556

MASTER

7-14-65

IOWA STATE UNIVERSITY

AMES LABORATORY

RESEARCH AND
DEVELOPMENT
REPORT

U.S.A.E.C.

PATENT CLEARANCE OBTAINED. RELEASE TO
THE PUBLIC IS APPROVED. PROCEDURES
ARE ON FILE IN THE RECEIVING SECTION.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, 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. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

IS-1072

Mathematics and Computers (UC-32)
TID-4500, May 1, 1965

UNITED STATES ATOMIC ENERGY COMMISSION
Research and Development Report

I. S. U. GUTS SYSTEM FOR KINEMATICAL
ANALYSIS OF BUBBLE CHAMBER DATA

by

William J. Kernan, William J. Higby and
Iva H. Boessenroth

November, 1964

Ames Laboratory

at

Iowa State University of Science and Technology
F. H. Spedding, Director
Contract W-7405 eng-82

This report is distributed according to the category Mathematics and Computers (UC-32) as listed in TID-4500, May 1, 1965.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Printed in USA. Price \$ 3.00 . . . Available from the
Clearinghouse for Federal Scientific and Technical Information.
National Bureau of Standards, U. S. Department of Commerce,
Springfield, Virginia.

CONTENTS

ABSTRACT	5
INTRODUCTION	5
MAIN PROGRAM	7
SUBROUTINE FILL	8
CONTROL AND COMMUNICATION	12
THE FIT AND ITERATION EQUATIONS	22
Analytic Constraints(definition)	23
Lagrange Multipliers(definition).	24
Computation of χ^2 (definition)	25
Step Size Cutting	29
Missing Mass	30
Calculation of Missing Variables	33
Calculation of Derivatives	36
Test on Rounding Errors.	46
DEFINITION OF CONVERGENCE	47
Convergence Criteria	47
CALCULATION OF MISSING VARIABLES WHEN NO FIT IS INVOLVED.	52
CALCULATION OF ERROR MATRICES	59
TRANSFORMATION TO CENTER OF MASS	67
Rotation of Axes	68
Lorentz Transformation	70
ACKNOWLEDGMENTS	77

APPENDIX I	78
APPENDIX II	79
APPENDIX III	80
REFERENCES	81

I. S. U. GUTS SYSTEM FOR KINEMATICAL ANALYSIS
OF BUBBLE CHAMBER DATA

William J. Kernan, William J. Higby and Iva H. Boessenroth

ABSTRACT

This paper is a description of the I. S. U. GUTS system for the kinematical analysis of bubble chamber data. As better methods of treating certain aspects of this problem are developed, we are occasionally updating this system. The present version of GUTS discussed here is Version III.

INTRODUCTION

Given the measured variables of all tracks from the interaction of two particles in a detection system such as a bubble chamber, the question has always been to determine what physically possible hypothesis or hypotheses yield good fits to the measured variables of this event. One well-known attempt at a least squares treatment of this problem is the program GUTS.¹

The present description is not a new departure of any sort, but simply a rewriting of GUTS into FORTRAN.

The physically possible final states for a given topology (i. e., number of prongs, vees and decaying tracks) change with the identity of the incoming and target particles and with the energy of the incoming particle. Since these change with each experiment, each topology is written as a subroutine which is changed from experiment to experiment. The topological program inspects the sign of the curvature of each particle

and then assigns the masses of a physically possible constrained final state. At this point the topological program calls VRTFIT. All of the routines containing the mathematics of the least squares fit are controlled by the call of VRTFIT.

The necessary input data to this version of GUTS are the following measured quantities for each track:

- azimuth in radians
- error in azimuth in radians
- dip in radians
- error in the dip in radians
- curvature (any units, only the sign is used by GUTS)
- momentum in MeV/c
- track length in thousandths of inches (a minus sign on this indicates a stopping track).

Also the identification of the event and the correct topological type must be supplied. Our read routine reads an input tape with more quantities than these and in a particular order, but the quantities GUTS must have to begin a fit are those listed above plus a mass assignment for each track.

The entire system can be divided into the following sections:

- a. main program(KGTSAN)
- b. topological routines
- c. control and communication(VRTFIT)
- d. least squares fit(GUTS)

- e. calculation of missing variables when no fit is involved (NOFIT)
- f. calculation of error matrices
- g. transformation to center of mass(CMQUAN).

MAIN PROGRAM

The purpose of the main program is to check on the use of a chained procedure, to check on any program hangs, to read in the data, assign alphameric names, assign particle masses, print preliminary information, and call the appropriate topological program for the event.

A brief description of various aspects of the main program follows:

1. IVA(1) is the type of incoming particle (see Appendix 1 for type numbers of particles).
2. If LATCH5 is not equal to zero this indicates that we are running in a chained procedure with DATPRO processing an event, followed by GUTS processing the same event.
3. The subroutine NOHANG initiates procedures to follow if a machine hang occurs during the processing of an event.
4. The subroutine ALPH sets up the alphameric notation for printing out the names of the particles.
5. The subroutine RUNGUT sets up the tape assignments and prints them if we are not running chained. (See FORTRAN comments at the beginning of subroutine RUNGUT for an explanation of the tape assignments.)

8..

6. The subroutine RDKMDT is a Fortran subroutine for reading the data associated with an event which is to be processed by GUTS. However we have found that FORTRAN read and write routines are very time consuming in lengthy production work. Therefore we are using a machine language subroutine for reading and writing data (called READWR). This is the reason for the input and output data being arranged in sequential storage locations.
7. Sense switch 1 when turned on causes a normal stop, but allows the program to finish the current event.
8. Sense switch 2 is used for two purposes. If turned on, PAUSE 102 occurs. Then, if desired, the location counter, LNO, can be interrogated to determine the event number or other information desired. Sense switch 2 is used in the same way if it becomes necessary to mount a new list tape (printed output on the fits).
9. PAUSE 101 can occur because of an end of file on the GUTS input tape. If there is no other input, sense switch 1 is turned on and start is pressed. However with the FORTRAN read routine RDKMDT, this PAUSE 101 would not occur.
10. The subroutine CHAIN gives control of the program to DATPRO if running chained.

SUBROUTINE FILL

The subroutine FILL has three main purposes and will be discussed in three blocks. It should be noted that the variable WALL, originally

the length of the chamber, is not now used; and that the momentum and the error in the momentum of the beam track, $P(1)$ and $EP(1)$, in MeV/C may change with the experiment.

The first block extends from FORTRAN statement 14 down to 12. This block transfers certain measured variables to another array since the original array will be overwritten and will later be the fitted variables.

Thus	original variable	becomes	new variable
Azimuth	AZ		SAZ
Error in azimuth	ERAZ		SERAZ
Dip	DIP		SDIP
Error in dip	ERDIP		SERDP
Momentum	P		SMOM
Error in momentum	EP		SERMOM

Although the original variables are not used in any other routine of GUTS except COPLN, they are saved in the arrays SERAZ, SDIP, etc. This is done so that they may be retrieved later if desired. This block also calculates an array SLNG(I), which is the track lengths. This array is the absolute value of the array BLNG(I). If an element of BLNG is negative, this denotes a stopping track.

Using the relationship that

$$k_i = \frac{1}{P_i \cos \lambda_i}$$

the error in the curvature ERRK or δk is expressed by

$$(\delta k_i)^2 = \left(\frac{\partial k_i}{\partial P_i} \delta P_i \right)^2 + \left(\frac{\partial k_i}{\partial \lambda_i} \delta \lambda_i \right)^2$$

or

$$(\delta k_i)^2 = \left(\frac{\delta P_i}{P_i^2 \cos \lambda_i} \right)^2 + \left(\frac{\tan \lambda_i \delta \lambda_i}{P_i \cos \lambda_i} \right)^2$$

The second block extends from below FORTRAN statement 12 to statement 7. The purpose of this block is to calculate certain variables between vertex positions. These variables are stored in positions 11-15 of the arrays SAZ, SDIP, SERAZ, SERDP, and SLNG. Referring to the coordinate system described under the section "Transformation to the Center of Mass" we find that (if subscripts ij mean "from vertex j to i") the dip is expressed by

$$\lambda_{ij} = \tan^{-1} \left(\frac{z_i - z_j}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \right) ;$$

the azimuth is expressed by

$$\phi_{ij} = \tan^{-1} \left(\frac{y_i - y_j}{x_i - x_j} \right) ;$$

the track length is expressed by

$$l_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} ;$$

the error in the azimuth is expressed by

$$\delta \phi_{ij} = \frac{40}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} ;$$

and the error in the dip is expressed by

$$\delta \lambda_{ij} = \frac{100 \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{(z_i - z_j)^2 + (y_i - y_j)^2 + (x_i - x_j)^2}$$

These variables are stored in the appropriate arrays as follows

vertex 1 to 2	in the 11 th element
vertex 1 to 3	12 th
vertex 1 to 4	13 th
vertex 2 to 3	14 th
vertex 2 to 4	15 th

The last block begins after FORTRAN statement 23. In this block the matrices SMEAS and ERMAT are calculated. SMEAS(I, J) is a matrix of the measured variables in which the first column contains the azimuth for each particle and the second column contains the tangent of the dip for each particle. The matrix SMEAS is further described in the section "Communications and Control". ERMAT(I, J, K) is the matrix of the associated uncertainties of the measured variables. All the elements of this matrix are zero except for*

$$\begin{aligned} \text{ERMAT}(J, 1, 1) &= (\delta \phi_j)^2 \\ \text{ERMAT}(J, 2, 2) &= [(1 + \tan^2 \lambda_j) \delta \lambda_j]^2 \\ \text{ERMAT}(J, 3, 3) &= (\delta k_j)^2 \end{aligned}$$

This matrix is also discussed in the section "COMMUNICATIONS AND CONTROL."

* This routine does not include any correlations between the errors assigned to the measured variables. If one desires to include error correlations in the array ERMAT, it would be quite valid at this point. The mathematics of GUTS is written in a general manner for the consideration of correlations if these terms appear in the array ERMAT.

In conclusion, since division by zero may occur in this routine, the overflow and divide check indicators are turned off in the last FORTRAN statements.

CONTROL AND COMMUNICATION

Subroutine VRTFIT is the main control section of the GUTS system. It is called by the TOP programs whenever a fit or missing mass calculation of a particular final state hypothesis is desired. The main function of this routine is to provide a communication link between the topological programs and subroutine GUTS. VRTFIT will examine certain variables set by the particular topological program by which it was called. These variables (see Ref. 2) provide the physical description of an event. We must then, on the basis of these variables, provide the subroutine GUTS with the proper input quantities that are to be used in the least squares fit.

VRTFIT has one fixed point parameter in the call list. We shall refer to this parameter by its FORTRAN name MM. This parameter determines if VRTFIT is called for a missing mass calculation or for a least squares fit. $MM \leq 0$ implies that a fit is desired, while $MM > 0$ implies that a missing mass calculation is called for.

The following discussion refers to specific blocks of Fortran coding in the subroutine. Figure 1 is a flow chart of these blocks.

BLOCK 01:

All WRITE OUTPUT TAPE statements refer to a variable tape number (IL). This list (output) tape number was read from the GUTS run card and is stored in NTPZ(5).

BLOCK 02:

The variables NCONST and NPART are saved upon entry to VRTFIT. Upon exit from this routine, they will be restored to their original value.

BLOCK 03:

The variables JA, and J are used as counters in BLOCK 06. They must be initially set to zero.

BLOCK 04:

GUTS will use the arrays PMASS(I), XMEAS(I, J), and GINA(I, J, K). They must be set to zero initially.

BLOCK 05:

DAMN determines which sign ($\pm$) to use in a later calculation in subroutine GUTS. The calculation involves the solution of a quadratic equation. In general, both the positive and negative solutions are desired. This feature is discussed thoroughly in the paper on writing topological programs.² An entry to VRTFIT with $MM \neq -3$ will cause DAMN to be set equal to +1, and MM will be unchanged. If $MM = -3$ upon entry, DAMN will be set to -1, and the value of MM will be changed to zero.

BLOCK 06:

The heading for the initial and final state particle configuration is established and written on the output tape at this point.

BLOCK 07:

We assign values to the two variables MISSL and MISSM to denote which particles have missing variables. The subroutine GUTS will need this information.

BLOCK 08:

Tracks that are relatively short are generally quite difficult to measure. As a result, the uncertainty in the track curvature may be larger than the curvature itself. If this is true for a given track, we may reduce constraints (if possible) and proceed to obtain a GUTS fit. Stopping tracks are an exception to this procedure. They will be treated in a special manner later in this program. Stopping tracks are denoted by a negative length. That is, if Track I is a stopping track the variable BLNG(I) will be negative in sign.

BLOCK 09:

The variables that GUTS will use for the least squares fit are the particle masses, input variables (AZIMUTH, DIP and CURVATURE), and the correlation error matrix associated with the input variables. Particle masses will be stored in an array PMASS(I), where I is 1, 2, 3, ... NPART. The input variables will be stored in a two dimensional array XMEAS(I, J). The first index refers to the Ith particle, and the second one refers to the Jth variable of the Ith particle. XMEAS will be of the form

$$\text{XMEAS}(I, J) = \begin{bmatrix} \varphi_1 & \tan \lambda_1 & k_1 \\ \varphi_2 & \tan \lambda_2 & k_2 \\ \varphi_3 & \tan \lambda_3 & k_3 \\ . & . & . \\ . & . & . \\ . & . & . \\ \varphi_n & \tan \lambda_n & k_n \end{bmatrix}$$

The correlation error matrix will be stored in a three dimensional array GINA(I, J, K). The first index refers to the Ith particle. Both indices J and K refer to correlation terms for the Ith particle. J and K will both take on the values 1, 2 and 3. We see that GINA will be a series of 3 x 3 matrices. A typical Ith 3 x 3 block will be of the form

$$\begin{bmatrix} \delta \varphi_\ell \delta \varphi_\ell & \delta \varphi_\ell \tan \lambda_\ell & \delta \varphi_\ell \delta k_\ell \\ \delta \tan \lambda_\ell \delta \varphi_\ell & \delta \tan \lambda_\ell \delta \tan \lambda_\ell & \delta \tan \lambda_\ell \delta k_\ell \\ \delta k_\ell \delta \varphi_\ell & \delta k_\ell \delta \tan \lambda_\ell & \delta k_\ell \delta k_\ell \end{bmatrix}$$

The variable NTR(J) is tested for each value of J. A negative NTR(J) specifies that the Jth particle is incoming to the vertex. Accordingly, the mass of the Jth particle will be set negative.

Recall that in BLOCK 08 we mentioned that tracks with negative BLNG denoted a stopping track. If a stopping track is found, we want to modify the measured curvature of this track

and the associated uncertainty in the curvature. The subroutine RTOP must be called, to determine the initial momentum (momentum at the vertex) of the stopping particle. RTOP is called with three arguments. The first two are input arguments GAP and RRM, where

GAP = mass of particle in MeV,

RRM = range of travel in cm,

and the third argument is PRM,

where

PRM = initial momentum in MeV/c.

This subroutine determines the initial momentum (PRM) of a particle with mass (GAP) which stops after traversing a distance (RRM). If the initial momentum calculated by RTOP satisfies the relationship

$$P_c \leq P_m - 2\delta P_m,$$

where

P_c is the initial momentum calculated by RTOP, P_m is the measured momentum and δP_m is the uncertainty in the measured momentum,

then the measured curvature is not modified. If the above relationship is not satisfied, we will modify the curvature using the momentum calculated by RTOP.

BLOCK 10:

IT is a variable set by the topological programs. This variable determines the mass of the target. Since the array SPMASS(I)

contains the particle masses, the topological program must assign a value to IT, say J. This will cause the target mass TMASS to be assigned a value corresponding to the J^{th} mass in the array SPMASS.

TEST 01:

VRTFIT may be called for one of two reasons. It is either called for a least squares fit of a particular hypothesis or it is called for a missing mass calculation. See p. 30 for a detailed discussion of the missing mass calculation. The variable MM, which is the call parameter of VRTFIT, will be less than or equal to zero if a fit is desired.

If GUTS is called for a missing mass calculation, VRTFIT outputs the results of this calculation, and returns to the calling program. If GUTS is called for a fit, the variable IRJCT is tested after the fit has been tried. If the fit has been rejected, IRJCT will be non-zero. After a successful fit has been made, the fitted variables and the correlation error matrix for the fitted variables will be stored in a doubly subscripted array XVERT, and a triply subscripted array GINA1, respectively. The format of the storage arrangement for these two arrays is identical to the format for the measured quantities. Recall that the measured variables and the correlation error matrix were stored in the arrays XMEAS and GINA before GUTS was called.

BLOCK 11:

For purposes of projecting a fit from one vertex to another, it is necessary to "SAVE" the fitted variables associated with given particles. The arrays SMEAS and ERMAT are used to save the fitted variable and their associated uncertainties. For fits that involve final states with a Λ^0 , K^0 , or K^0 as the incoming particle, the variables and uncertainties for the incoming particle will be saved. For fits that involve a Ξ^0 as one of the particles, the variables and uncertainties for this particle will be saved.

A variable NOP is set in the topological programs. The value of this variable will cause VRTFIT to initiate an additional variable saving procedure. If NOP is less than 1, the fitted variables and uncertainties for the last particle are saved. The fitted variables and uncertainties for the next to last particle are saved if $NOP > 1$. Nothing is saved if $NOP = 1$.

For a complete discussion of the necessity of these saving procedures and their implications, refer to Ref. 2.

BLOCK 12:

After the fit is made, and the desired variables have been saved, the fitted quantities are written on the output tape for later printing. It is desired to have the output in terms of degrees rather than radians, therefore, the factor of 57.3 is used in the conversion.

Recall that in the discussion of subroutine FILL we mentioned that the momentum of each track was converted to curvature by use of the relation

$$k = \frac{1}{P \cos \lambda}$$

Now we wish to convert the fitted curvature (k) to fitted momentum.

This conversion is trivial, since

$$P_i = \frac{1}{k_i \cos \lambda_i} = \left[\frac{1 + \tan^2 \lambda_i}{k_i^2} \right]^{\frac{1}{2}}$$

However, the conversion of δk to δP is not this straight forward.

We know that for a function of two variables

$$P_i = \left[\frac{1 + \tan^2 \lambda_i}{k_i^2} \right]^{\frac{1}{2}},$$

that

$$\delta P_i = \left[\left(\frac{\partial P_i}{\partial k_i} \delta k_i \right)^2 + \left(\frac{\partial P_i}{\partial \tan \lambda_i} \delta \tan \lambda_i \right)^2 + 2 \rho(k_i, \tan \lambda_i) \frac{\partial P_i}{\partial k_i} \frac{\partial P_i}{\partial \tan \lambda_i} \delta k_i \delta \tan \lambda_i \right]^{\frac{1}{2}}$$

Taking the partial derivatives, we have

$$\begin{aligned} \frac{\partial P_i}{\partial k_i} &= \left[\frac{k_i}{2(1 + \tan^2 \lambda_i)^{\frac{1}{2}}} \right] \times \left[-\frac{2k_i(1 + \tan^2 \lambda_i)}{k_i^4} \right] \\ &= -\frac{(1 + \tan^2 \lambda_i)^{\frac{1}{2}}}{k_i^2}, \end{aligned}$$

$$\frac{\partial P_i}{\partial \tan \lambda_i} = \left[\frac{k_i}{2(1+\tan^2 \lambda_i)^{\frac{1}{2}}} \right] \times \left[\frac{2 \tan \lambda_i}{k_i^2} \right]$$

The uncertainty in the momentum will be

$$\delta P_i = \left[\frac{(1+\tan^2 \lambda_i)(\delta k_i)^2}{k_i^4} + \frac{(\tan^2 \lambda_i)(\delta \tan \lambda_i)^2}{k_i^2(1+\tan^2 \lambda_i)} - 2 \frac{\tan \lambda_i}{k_i^3} [\rho(k_i, \tan \lambda_i) \delta k_i \delta \tan \lambda_i] \right]^{\frac{1}{2}}$$

The terms $(\delta k_i)^2$ and $(\delta \tan \lambda_i)^2$ will be diagonal elements of GINA1. The term $\rho(k_i, \tan \lambda_i) \delta k_i \delta \tan \lambda_i$ will be an off-diagonal element of GINA1.

A test is made after the fitted quantities have been written to determine if the track length is inconsistent with the fitted momentum. If this condition occurs for any track, a message to this fact is also written on the output tape.

After each GUTS fit is made, a record of the fitted quantities is written on a tape. This is the GUTS file. Subroutine WRKBN is called to write this record on tape.

Figure 1

THE FIT AND ITERATION EQUATIONS

Subroutine GUTS is the principle routine of the system. When this routine is called, the measured variables and their associated uncertainties for a given vertex serve as input quantities for a least squares fit. This fitting procedure will apply the conservation of energy and momentum to arrive at new values for the variables and for the associated uncertainties. For the following discussion, we will use the notation:

L	Number of constraints to be applied at this vertex	(NCONST)
I	Number of measured variables	(NVAR)
P	Number of particles	(NPART)
ϕ_i	Azimuth of the i^{th} particle in radians	(AZ(I))
$\tan \lambda_i$	tangent of the dip for the i^{th} particle	(DIP(I))
k_i	Radius of curvature for the i^{th} particle	(RCRV(I))
P_i	Momentum of the i^{th} particle	(FMOM(I))
E_i	Energy of the i^{th} particle	(ENERG(I))
M_i	Mass of the i^{th} particle	(PMASS(I))
M_T	Target mass	(TARGET)
x_i^{m}	Any measured quantity ($\phi_1, \tan \lambda_1, \phi_2, \dots, \phi_i, \dots$)	(XMEAS(I))
x_i	Any fitted quantity	(XVARN(I))
y_α	Any unmeasured variable	no special array
$(\pm)_i$	A notation to infer the value - 1 if particle i is incoming and +1 if particle i is outgoing.	(ONE(I))

ℓ	$\left\{ \begin{array}{ll} 3 \text{ Constraint:} & \text{Refers to track with missing momentum} \\ 2 \text{ Constraint:} & \text{Refers to one of the tracks with missing momentum} \\ 1 \text{ Constraint:} & \text{Refers to neutral particle, therefore, an unseen track} \\ 0 \text{ Constraint:} & \text{Refers to neutral particle, therefore, an unseen track} \end{array} \right\}$	MISSL
m	$\left\{ \begin{array}{ll} 2 \text{ Constraint:} & \text{Refers to the second track with missing momentum} \\ 0 \text{ Constraint:} & \text{Refers to the track with missing momentum} \end{array} \right\}$	MISSM

Any event is classed as a certain constraint type depending upon the number of constraints that can be applied at the vertex. These constraints are:

1. Conservation of momentum in the X direction (P_x).
2. Conservation of momentum in the Y direction (P_y).
3. Conservation of momentum in the Z direction (P_z).
4. Conservation of energy.

If an event is completely measured, therefore, no missing variables - then all four constraints can be applied for the fit. When variables are poorly measured or missing, then the number of constraints that can be applied will be four minus the number of poorly measured or missing variables.

Assuming that the variables x_i are normally distributed, we apply the usual principle of least squares. Our desire is to arrive at the optimum set of variables which minimize the function

$$\chi^2 = \sum_{i=1}^I \sum_{j=1}^I (x_i - x_i^m) G_{ij} (x_j - x_j^m) \quad (1.1)$$

When we introduce the conservation of energy and momentum in the form of constraint conditions $F_\lambda(x_i) = 0$ for $\lambda = 1, 2, \dots, L$ we have

$$M = \sum_{i=1}^I \sum_{j=1}^I (x_i - x_i^m) G_{ij} (x_j - x_j^m) + 2 \sum_{\lambda=1}^L \alpha_\lambda F_\lambda(x) \quad (1.2)$$

where

$$F_1 = \sum_{k=1}^P (\pm)_k \cos \phi_k / k_k = 0 \quad (1.3)$$

$$F_2 = \sum_{k=1}^P (\pm)_k \sin \phi_k / k_k = 0 \quad (1.4)$$

$$F_3 = \sum_{k=1}^P (\pm)_k \tan \lambda_k / k_k = 0 \quad (1.5)$$

$$F_4 = \sum_{k=1}^P (\pm)_k E_k - M_T = 0 \quad (1.6)$$

The α_λ in the Eq. (1.2) are the Lagrange multipliers. F_1, F_2 and F_3 are the sums of the x, y and z components of the momenta. F_4 is the conservation of energy.

The following iteration procedure was developed by Berge, Solmitz and Taft.¹ We will not give the derivation of the equations, but will only show how they are used in the least squares fit. For simplicity, we will give the equations in matrix notation. The subscripts denote the dimensions and the superscripts denote the iteration step. Essentially, we wish to define a vector $\vec{x}_i^{\nu+1}$ in terms of the previous iteration vector $\vec{x}_i^\nu$, the input error matrix G_{ij}^{-1} , the constraints $\vec{F}_\lambda^\nu$, and the derivatives of the constraints with respect to the variables, $F_{i\lambda}^\nu$. The definition of the elements of $F_{i\lambda}^\nu$ will be discussed later in this paper.

We will need to carry out the following calculations:

$$E_{i\lambda}^{\nu} = G_{ij}^{-1} F_{j\lambda}^{\nu} \quad (1.7)$$

$$H_{\lambda\mu}^{\nu} = (E_{i\lambda}^{\nu})^T F_{i\mu}^{\nu} \quad (1.8)$$

$$\vec{b}_{\lambda}^{\nu} = \vec{F}_{\lambda}^{\nu} + (\vec{x}_j^m - \vec{x}_j^{\nu})^T F_{j\lambda}^{\nu} \quad (1.9)$$

$$\vec{\alpha}_{\lambda}^{\nu+1} = (H_{\lambda\mu}^{\nu})^{-1} b_{\mu}^{\nu} \quad (1.10)$$

Now we arrive at the iterated values $\vec{x}_j^{\nu+1}$ as

$$\vec{x}_j^{\nu+1} = \vec{x}_j^m - E_{j\lambda}^{\nu} \vec{\alpha}_{\lambda}^{\nu+1} \quad (1.11)$$

We can compute χ^2 from two equations

$$\chi_a^2 = (\vec{x}_i^{\nu+1} - \vec{x}_i^m)^T G_{ij} (\vec{x}_j^{\nu+1} - \vec{x}_j^m) \quad (1.12)$$

$$\chi_b^2 = (\vec{\alpha}_{\lambda}^{\nu+1})^T \vec{b}_{\lambda}^{\nu+1} \quad (1.13)$$

where

$$\chi_a^2 \equiv \chi_b^2$$

We now have a test that can be applied on each iteration. This test will detect the presence of undesirably large rounding errors.

If on any iteration

$$\left[\frac{\chi_a^2 - \chi_b^2}{\chi_a^2} \right] < \epsilon_1 ,$$

where ϵ_1 is some preassigned error constant, we would discontinue the iteration.

Figure 2 is a flow diagram of the iteration section of GUTS. The following discussion will outline the function of each block shown on the flow chart.

BLOCK 01:

The curvatures of the tracks are stored in the array RCRV. GUTS will destroy the present contents of RCRV by various calculations. Other subroutines demand that RCRV should not be changed, so we must save the present values of RCRV.

BLOCK 02:

The constants that are set establish the error criteria, maximum number of iterations allowed, and other control factors.

BLOCK 03:

NVAR is the number of measured variables. It corresponds to I that has been referred to earlier.

BLOCK 04:

The call to subroutine SETCHK will set up the following arrays.

ONE(I)	$(\pm)_s \quad s = 1, P$ (Denotes particle s as incoming or outgoing)
PMASS(I)	$M_s \quad s = 1, P$ (Mass of particle s)
TARGET	M_T (Mass of target)
XMEASR(I)	$x_i^m \quad i = 1, I$ (Measured variables)
GINV(I, J)	$G_{ij}^{-1} \quad i, j = 1, 2 \cdots I$ (Input error matrix where G_{ij} is the variance covariance error matrix)

GINV and XMEASR are stored as

$$\text{GINV}(I, J) = \begin{bmatrix} \delta\phi_1 & \delta\phi_1 & 0 & 0 & 0 \\ 0 & \delta\tan\lambda_1 & \delta\tan\lambda_1 & 0 & 0 \\ 0 & 0 & \delta k_1 & \delta k_1 & 0 \\ 0 & 0 & 0 & \delta\phi_2 & \delta\phi_2 \\ 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

$$\text{XMEASR}(I) = \begin{bmatrix} \phi_1 \\ \tan\lambda_1 \\ k_1 \\ \phi_2 \\ \tan\lambda_2 \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}$$

If the number of constraints is not equal to four, there will not be, in general, three variables for each particle stored in GINV and XMEASR. For example, if the momentum of track two were missing (a three constraint case) GINV and XMEASR would be stored as

$$\text{GINV}(I, J) = \begin{bmatrix} \delta\phi_1 & \delta\phi_1 & 0 & 0 & 0 & . & . \\ 0 & \delta\tan\lambda_1 & \delta\tan\lambda_1 & 0 & 0 & . & . \\ 0 & 0 & \delta k_1 & \delta k_1 & 0 & . & . \\ 0 & 0 & 0 & \delta\phi_2 & \delta\phi_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta\tan\lambda_2 & \delta\tan\lambda_2 & 0 \\ . & . & . & . & 0 & \delta\phi_3 & \delta\phi_3 \\ . & . & . & . & . & 0 & . \end{bmatrix}$$

$$\text{SMEASR}(I) =$$

$$\begin{bmatrix} \phi_1 \\ \tan \lambda_1 \\ k_1 \\ \phi_2 \\ \tan \lambda_2 \\ \phi_3 \\ . \\ . \\ . \\ . \end{bmatrix}$$

BLOCK 05:

The iteration process of GUTS must have a starting value for the vector $\vec{x}_i^0$ to be able to calculate the next iterated variable $\vec{x}_i^0$. We set this initial $\vec{x}_i^0$ equal to the measured variables $\vec{x}_i^m$.

BLOCK 06:

Subroutine SETUP will set up variables from the iterated vector XNEW ($\vec{x}_i^{\nu+1}$).

AZ(I)	$\phi_i \quad i = 1, P$	Azimuth of track i
DIP(I)	$\tan \lambda_i \quad i = 1, P$	Tangent of the DIP for track i
RCRV(I)	$k_i \quad i = 1, P$	Curvature of track i
COSPHI(I)	$\cos \phi_i \quad i = 1, P$	
SINPHI	$\sin \phi_i \quad i = 1, P$	
SINLA(I)	$\sin \lambda_i \quad i = 1, P$	
COSLA(I)	$\cos \lambda_i \quad i = 1, P$	
FMOM(I)	$P_i \quad i = 1, P$	Momentum of particle i from the relation

$$P_i = \frac{1}{k_i \cos \lambda_i}$$

ENERG(I)	$E_i \quad i = 1, P$	Energy of particle i from the relation
----------	----------------------	---

$$E_i^2 = M_i^2 + P_i^2$$

Any of the above quantities that are functions of missing variables will not be computed in this routine. Locations will be reserved in the arrays for later computation of the functions of the missing variables.

A check is built into SETUP which enables us to maintain some control over the iteration process. It is possible in the fitting procedure for a track curvature to become negative. If this happens, we say that the new variables lie in an unphysical or undesirable region. The method that we are using to correct this situation is to cut down the step size. All iterated variables x_i^{v+1} will be replaced by

$$x_i^\nu + \frac{x_i^{\nu+1} - x_i^\nu}{f}.$$

The factor f is called CDFACT, and is initialized by GUTS in BLOCK 02. The subroutine that cuts the step size is CUTSTP. GCDCTR is a counter for the number of times that the step size has been cut. At no time during a fit is the step size allowed to be cut more than the maximum number of times specified by the variable FNCD. This variable is also initialized by GUTS in BLOCK 02. If the number of allowable cutsteps is exceeded, the fit is rejected by setting IRJCT equal to two (refer to Appendix 2 for reject types).

After the step size has been cut in an attempt to bring the variables into a physical region, they are again tested. If they still lie in an unphysical region, the step size is again cut. This procedure continues until the variables are acceptable or the number of cuts in the step size has exceeded the allowed value.

BLOCK 09.1

Given a vertex with a number of tracks that have completely measured variables, we can compute the missing mass at this vertex. Using the conservation of energy and momentum, we can find a unique mass that exactly balances energy-momentum.

We have

$$M^2 = E^2 - P^2 = F_4^2 - \sum_{i=1}^3 F_i^2.$$

So the missing mass M can be determined from this relation.

We also desire the uncertainty in the missing mass.

$$\begin{aligned}
\delta M &= \frac{\partial M}{\partial x_i} \delta x_i = \frac{1}{2M} \frac{\partial M}{\partial F_i} \frac{\partial F_i}{\partial x_j} \delta x_j \quad \text{over all } i \text{ and } j \\
&= \frac{1}{M} \left\{ F_4 \sum_i \frac{\partial F_4}{\partial x_i} \delta x_i - F_1 \sum_i \frac{\partial F_1}{\partial x_i} \delta x_i \right. \\
&\quad \left. - F_2 \sum_i \frac{\partial F_2}{\partial x_i} \delta x_i - F_3 \sum_i \frac{\partial F_3}{\partial x_i} \delta x_i \right\}
\end{aligned}$$

Considering the normally distributed variable

$$(\delta M)^2 = \frac{1}{M^2} \left[F_4 \sum_i \frac{\partial F_4}{\partial x_i} \delta x_i - F_1 \sum_i \frac{\partial F_1}{\partial x_i} \delta x_i - F_2 \sum_i \frac{\partial F_2}{\partial x_i} \delta x_i - F_3 \sum_i \frac{\partial F_3}{\partial x_i} \delta x_i \right]^2$$

if we define a vector of the constraint equation as

$$\vec{F}^* = \begin{bmatrix} -F_1 \\ -F_2 \\ -F_3 \\ +F_4 \end{bmatrix},$$

then the expression for $(\delta M)^2$ will reduce the

$$(\delta M)^2 = \frac{1}{M^2} \left[(\vec{F}^*)^T H \vec{F}^* \right],$$

where H is the matrix defined previously by Eq. (1.8).

When the missing mass is calculated and stored in UMASS, consideration is given to the sign of M^2 before taking the square root. We take the square root of the absolute value of M^2 , but the result will be signed the same as M^2 . In the calculation of $(\vec{F}^*)^T H \vec{F}^*$, there is a check for this quantity being negative. If this occurs, the missing mass calculation is rejected (Reject 16).

BLOCK 09:

The first convergence quantity is calculated at this point. As the iteration proceeds, the constraint functions should get smaller

in value. We therefore compute a quantity called GFSQRT defined as

$$\text{GFSQRT} = \left[\sum_{i=1}^L (F_i^\nu)^2 \right]^{\frac{1}{2}} .$$

No tests are made on this quantity if the fit is in the first iteration stage. After the first iteration has been made, GFSQRT will be tested for being less than one. If it is not less than one, and also not less than the value of GFSQRT from the previous iteration, the step size is cut.

BLOCK 07:

GUTS calls subroutine INITCR to initialize the analytic constraints and to calculate the missing variables. All the missing variables can be calculated by using the constraint equations. This is the reason for the terminology "3 constraint fit, 2 constraint fit", etc. We introduce the notation

$$\pi_x = \sum_s (\pm)_s \frac{\cos \phi_s}{k_s}$$

$$\pi_y = \sum_s (\pm)_s \frac{\sin \phi_s}{k_s}$$

$$\pi_z = \sum_s (\pm)_s \frac{\tan \lambda_s}{k_s}$$

$$\epsilon = \sum_s (\pm)_s E_s - M_T ,$$

where we have used $\sum_s$ to indicate the sum over all completely measured particles. The same convention applies to $(\pm)_s$ for these equations as it did for Eq. (1.3)-(1.6).

I. The 4 Constraint Case

All variables are completely measured so we can make use of all four constraint equations in the fit. We set the components of the constraint vector $\vec{F}_\lambda^\nu$ as

$$F_1 = \pi_x$$

$$F_2 = \pi_y$$

$$F_3 = \pi_z$$

$$F_4 = \epsilon$$

II. The 3 Constraint Case

For this case we have one curvature that is missing, k_ℓ . The energy constraint equation becomes

$$\sum_s (\pm)_s E_s - M_T + (\pm)_\ell E_\ell = 0$$

when the unknown energy E_ℓ is included. Clearly then, since

$$\sum_s (\pm)_s E_s - M_T = - (\pm)_\ell E_\ell,$$

we must have

$$\epsilon = - (\pm)_\ell E_\ell.$$

Considering the definition of k_ℓ as

$$k_\ell = \frac{1}{P_\ell \cos \lambda_\ell} = \left[\frac{1 + \tan^2 \lambda_\ell}{P_\ell^2} \right]^{\frac{1}{2}},$$

and since

$$\epsilon_\ell^2 = M_\ell^2 + P_\ell^2 ,$$

we have

$$k_\ell = \left[\frac{1 + \tan^2 \lambda_\ell}{\epsilon_\ell^2 - M_\ell^2} \right]^{\frac{1}{2}} .$$

The use of the energy constraint equation to solve for the missing curvature k_ℓ has eliminated its use in the iteration. Using the value of k_ℓ in the remaining constraint equations, we have a 3 vector $\vec{F}_\lambda^\nu$ as

$$F_1 = \pi_x + (\pm)_\ell \frac{\cos \phi_\ell}{k_\ell}$$

$$F_2 = \pi_y + (\pm)_\ell \frac{\sin \phi_\ell}{k_\ell}$$

$$F_3 = \pi_z + (\pm)_\ell \frac{\tan \lambda_\ell}{k_\ell} .$$

III. The 2 Constraint Case

Two curvatures are missing. They are k_ℓ and k_m . When these unknown curvatures are included in the first two constraint equations, we have

$$\pi_x + (\pm)_\ell \frac{\cos \phi_\ell}{k_\ell} + (\pm)_m \frac{\cos \phi_m}{k_m} = 0$$

$$\pi_y + (\pm)_\ell \frac{\sin \phi_\ell}{k_\ell} + (\pm)_m \frac{\sin \phi_m}{k_m} = 0 .$$

When these equations are solved for k_ℓ and k_m , we obtain

$$k_\ell = (\pm)_\ell \frac{\sin(\phi_\ell - \phi_m)}{\sum_s (\pm)_s \frac{\sin(\phi_m - \phi_s)}{k_s}}$$

$$k_m = (\pm)_m \frac{\sin(\phi_m - \phi_\ell)}{\sum_s (\pm)_s \frac{\sin(\phi_\ell - \phi_s)}{k_s}}$$

The $\sum_s$ notation still means the sum over all completely measured particles.

The two remaining constraint equations will be

$$F_1 = \pi_z + (\pm)_\ell \frac{\tan \lambda_\ell}{k_\ell} + (\pm)_m \frac{\tan \lambda_m}{k_m}$$

$$F_2 = \epsilon + (\pm)_\ell E_\ell + (\pm)_m E_m$$

IV. The 1 Constraint Case

All three variables ϕ_ℓ , $\tan \lambda_\ell$, and k_ℓ for one track are missing.

The first three constraint equations will be

$$\pi_x + (\pm)_\ell \frac{\cos \phi_\ell}{k_\ell} = 0$$

$$\pi_y + (\pm)_\ell \frac{\sin \phi_\ell}{k_\ell} = 0$$

$$\pi_z + (\pm)_\ell \frac{\tan \lambda_\ell}{k_\ell} = 0$$

Using the first two equations as

$$\frac{-\pi_y}{-\pi_x} = \frac{(\pm)_\ell \sin \phi_\ell}{(\pm)_\ell \cos \phi_\ell},$$

we arrive at

$$\phi_\ell = \tan^{-1} \left[\frac{-(\pm)_\ell \pi_y}{-(\pm)_\ell \pi_x} \right]$$

The $-(\pm)_\ell$ factors in the above expression cannot be cancelled, as they will be used to determine the proper quadrant in which to place ϕ_ℓ .

Squaring the first two constraint equations and summing will give us

$$\pi_x^2 + \pi_y^2 = \frac{\cos^2 \phi_\ell}{k_\ell^2} + \frac{\sin^2 \phi_\ell}{k_\ell^2},$$

so

$$k_\ell = (\pi_x^2 + \pi_y^2)^{-\frac{1}{2}}.$$

The third constraint equation can be solved directly for $\tan \lambda_\ell$ as

$$\tan \lambda_\ell = -(\pm)_\ell k_\ell \pi_z.$$

Checks are built into INITCR to detect calculated curvatures that are negative. If this occurs for a 3 constraint fit, a variable FLAG is set equal to 30 and the routine returns to GUTS. For this occurrence in a 2 constraint fit FLAG would be set equal to 20. It is impossible for GUTS to calculate a negative curvature for the 1 constraint fit. You will recall that subroutine SETUP did not calculate any functions of the missing variables (momentum, energy, etc). They are now calculated by INITCR.

When control is returned to GUTS, a test is made for the calculated curvature being negative. If this is the first iteration, the fit is rejected. If not the first iteration, the step size is cut.

BLOCK 08:

The derivatives of the constraint functions with respect to the measured variables ($F_{i\lambda} = \frac{\partial F_i}{\partial x_i}$) are computed in the subroutine DERIVS, and stored in the doubly subscripted array DERIV(I, J). This derivative matrix will be different for each constraint case.

The size of the matrix is $I \times L$ where I is the number of measured variables, and L is the number of constraints. We can think of this matrix as consisting of P blocks, where P is the total number of particles. The size of each of these blocks is of course a function of the number of constraints, and also the number of measured variables for each particle. We shall be using the notation $F_{ij}(s)$. This denotes the partial derivative of the j^{th} constraint with respect to the i^{th} variable of particle s . The index j can have the values of 1 to 4, and i the values 1 to 3.

The 4 Constraint Case

There will be P blocks in the F matrix, each block will be of size (3×4) . The matrix is defined as

$$F_{11}(s) = -(\pm)_s \frac{\sin \phi_s}{k_s}$$

$$F_{12}(s) = (\pm)_s \frac{\cos \phi_s}{k_s}$$

$$F_{13}(s) = 0$$

$$F_{14}(s) = 0$$

$$F_{21}(s) = 0$$

$$F_{22}(s) = 0$$

$$F_{23}(s) = (\pm)_s \frac{1}{k_s}$$

$$F_{24}(s) = (\pm)_s \frac{\tan \lambda_s}{E_s k_s^2}$$

$$F_{31}(s) = -(\pm)_s \frac{\cos \phi_s}{k_s^2}$$

$$F_{32}(s) = -(\pm)_s \frac{\sin \phi_s}{k_s^2}$$

$$F_{33}(s) = -(\pm)_s \frac{\tan \lambda_s}{k_s^2}$$

$$F_{34}(s) = -(\pm)_s \frac{P_s^2}{E_s k_s^3}$$

The index s will have the values 1 to P .

For the remaining constraint cases we will have to introduce a new matrix $V_{j\beta}(s, \ell) = (\pm)_s (\pm)_\ell \frac{\partial y_\beta}{\partial x_j}$. The factors $(\pm)_s$ and $(\pm)_\ell$ are included to simplify the equations for the derivatives. Here $\beta = 1, 2, \text{ or } 3$ refers to the missing variables of particle s and $j = 1, 2 \text{ or } 3$ refers to the measured variables of particle s . The matrix $V_{j\beta}(s, 1)$ is stored in a doubly subscripted array VMAT(I, J).

The 3 Constraint Case

In this case we have one unmeasured variable k_ℓ . There will be $P - 1$ blocks of the form

$$\begin{aligned} V_{11}(s, \ell) &= (\pm)_s (\pm)_\ell \frac{\partial k_\ell}{\partial \phi_s} = 0, \\ V_{21}(s, \ell) &= (\pm)_s (\pm)_\ell \frac{\partial k_\ell}{\partial \tan \lambda_s} = \frac{\tan \lambda_s}{E_s k_s^2} \times \frac{k_\ell E_\ell}{P_\ell^2}, \\ V_{31}(s, \ell) &= (\pm)_s (\pm)_\ell \frac{\partial k_\ell}{\partial k_s} = - \frac{P_s^2}{E_s k_s} \times \frac{k_\ell E_\ell}{P_\ell^2}, \end{aligned}$$

where $s = 1, 2, \dots, P - 1$. We have covered all particles with completely measured variables. We still must include particle ℓ with only the ϕ and $\tan \lambda$ measured. Replacing the index s by ℓ we obtain one block as

$$\begin{aligned} V_{11}(\ell, \ell) &= \frac{\partial k_\ell}{\partial \phi_\ell} = 0, \\ V_{21}(\ell, \ell) &= \frac{\partial k_\ell}{\partial \tan \lambda_\ell} = \frac{\tan \lambda_\ell}{k_\ell P_\ell^2}. \end{aligned}$$

The derivative matrix for the 3 constraint case may now be written as
 $P - 1$ blocks of the form

$$\begin{aligned}
 F_{11}(s) &= -(\pm)_s \frac{\sin \phi_s}{k_s}, \\
 F_{12}(s) &= (\pm)_s \frac{\cos \phi_s}{k_s}, \\
 F_{13}(s) &= 0, \\
 F_{21}(s) &= -(\pm)_s \frac{\cos \phi_\ell}{k_\ell^2} V_{21}(s, \ell), \\
 F_{22}(s) &= -(\pm)_s \frac{\sin \phi_\ell}{k_\ell^2} V_{21}(s, \ell), \\
 F_{23}(s) &= (\pm)_s \left[\frac{1}{k_s} - \frac{\tan \lambda_\ell}{k_\ell^2} V_{21}(s, \ell) \right], \\
 F_{31}(s) &= -(\pm)_s \left[\frac{\cos \phi_s}{k_s^2} + \frac{\cos \phi_\ell}{k^2} V_{31}(s, \ell) \right], \\
 F_{32}(s) &= -(\pm)_s \left[\frac{\sin \phi_s}{k_s^2} + \frac{\sin \phi_\ell}{k_\ell^2} V_{31}(s, \ell) \right], \\
 F_{33}(s) &= -(\pm)_s \left[\frac{\tan \lambda_s}{k_s^2} + \frac{\tan \lambda_\ell}{k_\ell^2} V_{31}(s, \ell) \right],
 \end{aligned}$$

and when ℓ is substituted for s there will be one block with

$$\begin{aligned}
 F_{11}(\ell) &= -(\pm)_\ell \frac{\sin \phi_\ell}{k_\ell}, \\
 F_{12}(\ell) &= (\pm)_\ell \frac{\cos \phi_\ell}{k_\ell}
 \end{aligned}$$

$$F_{13}(\ell) = 0 ,$$

$$F_{21}(\ell) = - (\pm)_\ell \frac{\cos \phi_\ell}{k_\ell^2} V_{21}(\ell, \ell) ,$$

$$F_{22}(\ell) = - (\pm)_\ell \frac{\sin \phi_\ell}{k_\ell^2} V_{21}(\ell, \ell) ,$$

$$F_{23}(\ell) = (\pm)_\ell \left[\frac{1}{k_\ell} - \frac{\tan \lambda_\ell}{k_\ell^2} V_{21}(\ell, \ell) \right]$$

The 2 Constraint Case

Both k_ℓ and k_m are missing. We will have $P - 2$ blocks of the form

$$V_{11}(s, \ell) = (\pm)_s (\pm)_\ell \frac{\partial k_\ell}{\partial \phi_s} = \frac{k_\ell^2 \cos(\phi_m - \phi_s)}{k_s \sin(\phi_\ell - \phi_m)} ,$$

$$V_{12}(s, m) = (\pm)_s (\pm)_m \frac{\partial k_m}{\partial \phi_s} = \frac{k_m^2 \cos(\phi_\ell - \phi_s)}{k_s \sin(\phi_m - \phi_\ell)} ,$$

$$V_{21}(s, \ell) = (\pm)_s (\pm)_\ell \frac{\partial k_\ell}{\partial \tan \lambda_s} = 0 ,$$

$$V_{22}(s, m) = (\pm)_s (\pm)_m \frac{\partial k_m}{\partial \tan \lambda_s} = 0 ,$$

$$V_{31}(s, \ell) = (\pm)_s (\pm)_\ell \frac{\partial k_\ell}{\partial k_s} = \frac{k_\ell^2 \sin(\phi_m - \phi_s)}{k_s^2 \sin(\phi_\ell - \phi_m)} ,$$

$$V_{32}(s, m) = (\pm)_s (\pm)_m \frac{\partial k_m}{\partial k_s} = \frac{k_m^2 \sin(\phi_\ell - \phi_s)}{k_s^2 \sin(\phi_m - \phi_\ell)} .$$

We will also have one block of the form

$$V_{11}(\ell, \ell) = \frac{\partial k_\ell}{\partial \phi_\ell} = k_\ell \cot(\phi_\ell - \phi_m),$$

$$V_{12}(\ell, m) = (\pm)_\ell (\pm)_m \frac{\partial k_m}{\partial \phi_\ell} = \frac{k_m^2}{k_\ell \sin(\phi_m - \phi_\ell)},$$

$$V_{21}(\ell, \ell) = \frac{\partial k_\ell}{\partial \tan \lambda_\ell} = 0,$$

$$V_{22}(\ell, m) = (\pm)_\ell (\pm)_m \frac{\partial k_m}{\partial \tan \lambda_\ell} = 0,$$

and another block of the form

$$V_{11}(m, \ell) = (\pm)_m (\pm)_\ell \frac{\partial k_\ell}{\partial \phi_m} = \frac{k_\ell^2}{k_m \sin(\phi_\ell - \phi_m)},$$

$$V_{12}(m, m) = \frac{\partial k_m}{\partial \phi_m} = k_m \cot(\phi_m - \phi_\ell),$$

$$V_{21}(m, \ell) = (\pm)_m (\pm)_\ell \frac{\partial k_\ell}{\partial \tan \lambda_m} = 0,$$

$$V_{22}(m, m) = \frac{\partial k_m}{\partial \tan \lambda_m} = 0.$$

The derivative matrix will consist of $P - 2$ blocks of the form

$$F_{11}(s) = -(\pm)_s \left[\frac{\tan \lambda_\ell}{k_\ell^2} V_{11}(s, \ell) + \frac{\tan \lambda_m}{k_m^2} V_{12}(s, m) \right],$$

$$F_{12}(s) = -(\pm)_s \left[\frac{P_\ell^2}{E_\ell k_\ell} V_{11}(s, \ell) + \frac{P_m^2}{E_m k_m} V_{12}(s, m) \right],$$

$$F_{21}(s) = (\pm)_s \frac{1}{k_s},$$

$$F_{22}(s) = (\pm)_s \frac{\tan \lambda_s}{E_s k_s^2},$$

$$F_{31}(s) = -(\pm)_s \left[\frac{\tan \lambda_s}{k_s^2} + \frac{\tan \lambda_\ell}{k_\ell^2} V_{31}(s, \ell) + \frac{\tan \lambda_m}{k_m^2} V_{32}(s, m) \right],$$

$$F_{32}(s) = -(\pm)_s \left[\frac{P_s^2}{E_s k_s} + \frac{P_\ell^2}{E_\ell k_\ell} V_{31}(s, \ell) + \frac{P_m^2}{E_m k_m} V_{32}(s, m) \right],$$

with one block of the form

$$F_{11}(\ell) = -(\pm)_\ell \left[\frac{\tan \lambda_\ell}{k_\ell^2} V_{11}(\ell, \ell) + \frac{\tan \lambda_m}{k_m^2} V_{12}(\ell, m) \right],$$

$$F_{12}(\ell) = -(\pm)_\ell \left[\frac{P_\ell^2}{E_\ell k_\ell} V_{11}(\ell, \ell) + \frac{P_m^2}{E_m k_m} V_{12}(\ell, m) \right],$$

$$F_{21}(\ell) = (\pm)_\ell \frac{1}{k_\ell},$$

$$F_{22}(\ell) = \frac{\tan \lambda_\ell}{E_\ell k_\ell^2}.$$

And one block of the form

$$F_{11}(m) = -(\pm)_m \left[\frac{\tan \lambda_\ell}{k_\ell^2} V_{11}(m, \ell) + \frac{\tan \lambda_m}{k_m^2} V_{12}(m, m) \right],$$

$$F_{12}(m) = -(\pm)_m \left[\frac{P_\ell^2}{E_\ell k_\ell} V_{11}(m, \ell) + \frac{P_m^2}{E_m k_m} V_{12}(m, m) \right],$$

$$F_{21}(m) = (\pm)_m \frac{1}{k_m},$$

$$F_{22}^{(m)} = \frac{\tan \lambda_m}{E_m k_m^2}.$$

The 1 Constraint Case

We will have P-1 blocks of the form

$$V_{11}(s, \ell) = \frac{\tan \phi_\ell}{k_\ell} V_{13}(s, \ell) (\pm)_\ell \frac{\cos \phi_s}{\pi_x k_s^2},$$

$$V_{12}(s, \ell) = \frac{\tan \lambda_\ell}{k_\ell} V_{13}(s, \ell),$$

$$V_{13}(s, \ell) = (\pm)_\ell \frac{k_\ell^3}{k_s} [\pi_x \sin \phi_s - \pi_y \cos \phi_s],$$

$$V_{21}(s, \ell) = 0, \quad V_{22}(s, \ell) = -k_\ell/k_s, \quad V_{23}(s, \ell) = 0,$$

$$V_{31}(s, \ell) = \frac{\tan \phi_\ell}{k_\ell} V_{33}(s, \ell) - (\pm)_\ell \frac{\sin \phi_s}{\pi_x k_s^2},$$

$$V_{32}(s, \ell) = \frac{\tan \lambda_\ell}{k_\ell} V_{33}(s, \ell) + \frac{k_\ell \tan \lambda_s}{k_s^2},$$

$$V_{33}(s, \ell) = (\pm)_\ell \frac{k_\ell^3}{k_s^2} [\pi_y \sin \phi_s + \pi_x \cos \phi_s].$$

The constraint derivative will be P-1 blocks of the form

$$F_{11}(s) = (\pm)_s \left[\frac{P_\ell^2}{E_\ell k_\ell} V_{13}(s, \ell) + \frac{\tan \lambda_\ell}{E_\ell k_\ell^2} V_{12}(s, \ell) \right],$$

$$F_{21}(s) = (\pm)_s \left[\frac{\tan \lambda_s}{E_s k_s^2} + \frac{\tan \lambda_\ell}{E_\ell k_\ell^2} V_{22}(s, \ell) \right],$$

$$F_{31}(s) = (\pm)_s \left[- \frac{P_s^2}{E_s k_s} - \frac{P_\ell^2}{E_\ell k_\ell} V_{33}(s, \ell) + \frac{\tan \lambda_\ell}{E_\ell k_\ell^2} V_{32}(s, \ell) \right] .$$

BLOCK 10:

Later calculations will be functions of variables computed on a previous iteration. This is the reason for the FORTRAN statements $CGSQRT = CFSQRT$, $XOLD(I) = XNEW(I)$, and $FILOLD(I, J) = DERIV(I, J)$.

BLOCKS 11 and 12:

The matrices $E_{i\lambda}^\nu$ and $H_{\lambda\mu}^\nu$ defined by Eqs. (1.7) and (1.8) are computed in these BLOCKS. The result of the matrix multiplication performed to arrive at these functions is stored in the doubly subscripted arrays (EMAT(I, J) and HMAT(I, J)).

BLOCK 13:

The vector b_λ^ν defined by Eq. (1.9) is computed in this BLOCK. The result of the calculation is stored in the singly subscripted variable B(I).

At this point we are now ready to compute the Lagrange multipliers define by Eq. (1.10) as

$$\vec{\alpha}_\lambda^{\nu+1} = (H_{\lambda\mu}^\nu)^{-1} \vec{b}_\mu^\nu .$$

H^ν has been computed in BLOCK 12, and it is a symmetric matrix of order $L \times L$ where L is the number of constraints to be applied to this fit. The matrix inversion routine (MATRX) that we are using is a modified version of the Busing and Levy method. Not

only will this routine invert a symmetric matrix, but it will provide the solution to the matrix equation

$$A\vec{x} = \vec{b},$$

where the upper triangular half of the matrix A and the vector $\vec{b}$ must be specified. The subroutine MATRX will compute A^{-1} and the vector $\vec{x}$, where

$$\vec{x} = A^{-1} \vec{b}.$$

To save storage space, MATRX will replace the upper triangular portion (including the diagonal) of A with the upper triangular portion of A^{-1} . The vector $\vec{b}$ will also be replaced by the solution vector $\vec{x}$. The call to MATRX is

CALL MATRX(A, IORD, IMAX, B, 1, DET, W1, W2, W3),

where

A is the symmetrix matrix to be inverted

IORD is the order of the matrix

IMAX is the value that A is dimensioned for in both subscripts (IMAX by IMAX)

B is a vector corresponding to the $\vec{b}$ to which we have referred the 5th parameter must always be specified as 1

DET is set equal to zero by MATRX to denote that A is a singular matrix

W1, W2, W3 are work areas used by MATRX. All three are singly subscripted variables. They must be dimensioned equal to or greater than the dimension of a single subscript of the matrix A. For example if A was previously dimensioned as 20x20; W1, W2, and W3 must be dimensioned 20 or greater.

You will note that before the matrix solution is attempted, the variable NCONST is tested to determine if this fit is 1 constraint. If so, the H matrix will consist of only 1 element in the H(1, 1) position. The solution for the single Lagrange multiplier can be done for the case that H is a scalar in a direct manner. If H is indeed a singular matrix, the variable IRJCT is set equal to 3, and the fit is discontinued. The storage variable for H^{-1} is the doubly subscripted array HINV(I, J). The Lagrange multipliers are stored in the singly subscripted array GRANGE(I).

BLOCK 14:

We are now ready to compute the new variables $\vec{x}_j^{\nu+1}$ as given by Eq. (1.11). These variables will be stored in a singly subscripted array XNEW(I).

BLOCK 15:

Use is made here of the two Eqs. (1.12 and 1.13) that define χ^2 . Both χ_a^2 and χ_b^2 are computed and stored in GMFCN and PRIMEM.

BLOCK 16:

There are certain checks that are made at this point to improve the iteration or reject the fit. If, on the first step, $\chi^2 \geq 500$, the fit is rejected. The factor of 500. can be changed if the user so desires. It is referred to by the variable GMMAX, and is set equal to 500 at the beginning of GUTS (BLOCK 02).

A test is made to determine if the relative difference between the two expressions for χ^2 is less than 10^{-3} . If this is not true, the

rounding errors are too large and the fit is rejected. The factor of 10^{-3} can be changed by assigning a different value to CHECK at the beginning of GUTS. If, after 15 iterations, χ^2 is not less than 25, the fit is rejected. The final iteration test will not allow the fit to exceed 50 steps. Also, if the number of iterations has reached 20, every other following iteration will have the step size cut.

DEFINITION OF CONVERGENCE

After each step of the iteration we have an approximation to the variables x_i that minimize the function given by Eq. (1.2). We desire convergence criteria which, if satisfied, will determine the step that yields the best solution to the problem in the sense of least squares. The tests on convergence will be made on each step after completion of the first iteration. Before BLOCK 10 is entered, a test on GCONTR is made to determine if the first iteration has been completed. If it has, then the convergence tests are employed.

From the conservation of energy and momentum, we expect the constraint equations, when summed over all particles, to yield a relatively small value. This criteria is defined by

$$(\sum_i F_i^2)^{\frac{1}{2}} < 1.$$

The constant 1 is previously set by GUTS in BLOCK 02, and is referred to by the FORTRAN variable GFOK. We further expect that the relationship between the constraint equations of a given iteration and the following iteration satisfy

$$(\sum_i F_i^{\nu+1})^2)^{\frac{1}{2}} < (\sum_i F_i^{\nu})^2)^{\frac{1}{2}}.$$

If both of the above criteria are not satisfied, the step size is cut. The remaining convergence criteria that must be satisfied is

$$\delta M^{\nu+1} = (\psi_i^{\nu+1})^T G_{ij}^{-1} \psi_j^{\nu+1})^{\frac{1}{2}} < \epsilon,$$

where

$$\psi_i^{\nu+1} = 2(F_{i\lambda}^{\nu+1} - F_{i\lambda}^{\nu}) \alpha_{\lambda}^{\nu+1},$$

and ϵ is some constant (presently we are using 10^{-3}).

We further require that this last criteria be satisfied for two successive iterations before we say the fit has converged.

After the fit has converged, GUTS will call the appropriate routines to compute the correlation error matrices of the fitted variables. Subroutine CMQUAN will be called for the conversion of the laboratory quantities to the center of mass system. The array RCRV(I) will be restored to its original value, and control will be returned to VRTFIT.

Fig. 2.

Fig. 2 (Cont.)

Fig. 2. (Cont.)

CALCULATION OF MISSING VARIABLES WHEN NO FIT IS INVOLVED

The physical situation is such that the ℓ^{th} particle is completely unmeasured, and the m^{th} particle has an uncertain momentum. Since this is a zero constraint case, no fit can be made. That is, it is impossible to iterate and obtain better values for the variables. All the constraint equations will be used just to calculate the missing variables, leaving no Lagrange multipliers in the function that we desire to minimize. We can only calculate the missing variables consistent with the measured variables, and the associated error correlations of these variables.

Consider the summation of the momentum and energy over all the completely measured particles:

$$\begin{aligned}\pi_x &= \sum_s (\pm)_s \frac{\cos \phi_s}{k_s} \\ \pi_y &= \sum_s (\pm)_s \frac{\sin \phi_s}{k_s} \\ \pi_z &= \sum_s (\pm)_s \frac{\tan \lambda_s}{k_s} \\ \epsilon &= \sum_s (\pm)_s E_s - M_T.\end{aligned}\tag{1}$$

Recall that $(\pm)_s$ corresponds to the FORTRAN variable ONE(I), where a positive value denotes that the particle is outbound, and a negative value denotes a particle is inbound.

The quantities π_x , π_y , π_z and ϵ are calculated in the first section of the subroutine NOFIT and are referred to by the FORTRAN variables

PIX, PIY, PIZ and EPSILN respectively. If we now calculate the curvature of the m^{th} particle, the formalism of the 1 constraint case may be used to find the other missing variables. The reader should refer to Ref. 1 for the development of the following mathematics.

Define:

$$A = (a^2 - \epsilon^2 M_m^2)$$

$$B = a b$$

$$C = b^2 - \epsilon^2 (1 + \tan^2 \lambda_m)$$

$$R = (a k_m + b)$$

$$S = -\epsilon (1 + m_m^2 k_m^2 + \tan^2 \lambda_m)$$

where

$$a = \frac{1}{2}(M_\ell^2 - M_m^2 + \pi^2 - \epsilon^2)$$

$$\pi^2 = \pi_x^2 + \pi_y^2 + \pi_z^2$$

$$b = (\pm)_m (\pi_x \cos \phi_m + \pi_y \sin \phi_m + \pi_z \tan \lambda_m) \quad (2)$$

The expression for the missing curvature of the m^{th} particle is

$$k_m = \frac{1}{A} \left\{ -B \pm \sqrt{B^2 - AC} \right\} \quad (3)$$

Subroutine KVEC is called by NOFIT to make this calculation. There are two rejects that may occur in this calculation of k_m , and they are

$$\begin{array}{ll} \text{reject \# 15} & B^2 - AC < 0 \quad (\text{imaginary momentum in zero constraint}) \\ \text{reject \# 17} & k_m \leq 0 \quad (\text{negative momentum or energy in zero constraint}). \end{array}$$

The sign of the radicand in the expression for k_m is set by subroutine VRTFIT. The procedure necessary to obtain both fits for the positive and negative solution of this quadratic is described in the publication,

"Writing Topological Programs for I. S. U. GUTS".²

In the determination of the error correlations we will need the derivatives $\frac{\partial k_m}{\partial x_i}$, where the x_i are the measured variables. These derivatives are defined as

$$\frac{\partial k_m}{\partial x_i} = - \frac{R(k_m \frac{\partial a}{\partial x_i} + \frac{\partial b}{\partial x_i}) + S \frac{\partial \epsilon}{\partial x_i}}{Ak_m + B}, \quad (4)$$

For the definition of the $\frac{\partial a}{\partial x_i}$, $\frac{\partial b}{\partial x_i}$ and $\frac{\partial \epsilon}{\partial x_i}$, we will substitute for the x_i , the actual variables.

$$x_1 = \phi_1,$$

$$x_2 = \tan \lambda_1,$$

$$x_3 = k_3,$$

$$x_4 = \phi_2,$$

.

.

.

etc.

(5)

For a:
$$\frac{\partial a}{\partial \phi_s} = \frac{(\pm)_s}{k_s} [-\pi_x \sin \phi_s + \pi_y \cos \phi_s],$$

$$\frac{\partial a}{\partial \tan \lambda_s} = \frac{(\pm)_s}{k_s} \left[-\frac{\epsilon}{E_s k_s} \tan \lambda_s + \pi_z \right],$$

$$\frac{\partial a}{\partial k_s} = \frac{(\pm)_s}{k_s} \left[\frac{\epsilon}{E_s} P_s^2 - \frac{1}{k_s} (\pi_x \cos \phi_s + \pi_y \sin \phi_s + \pi_z \tan \lambda_s) \right],$$

$$\frac{\partial a}{\partial \phi_m} = 0,$$

$$\frac{\partial a}{\partial \tan \lambda_m} = 0 . \quad (6)$$

For b:

$$\frac{\partial b}{\partial \phi_s} = (\pm)_m (\pm)_s \left[-\frac{\sin \phi_s}{k_s} \cos \phi_m + \frac{\cos \phi_s}{k_s} \sin \phi_m \right] ,$$

$$\frac{\partial b}{\partial \tan \lambda_s} = (\pm)_m (\pm)_s \frac{1}{k_s} \tan \lambda_m ,$$

$$\frac{\partial b}{\partial k_s} = -(\pm)_m (\pm)_s \frac{1}{k_s^2} [\cos \phi_s \cos \phi_m + \sin \phi_s \sin \phi_m + \tan \lambda_s \tan \lambda_m] ,$$

$$\frac{\partial b}{\partial \phi_m} = (\pm)_m [-\pi_x \sin \phi_m + \pi_y \cos \phi_m] ,$$

$$\frac{\partial b}{\partial \tan \lambda_m} = (\pm)_m \pi_z . \quad (7)$$

For ϵ :

$$\frac{\partial \epsilon}{\partial \phi_s} = 0 ,$$

$$\frac{\partial \epsilon}{\partial \tan \lambda_s} = (\pm)_s \frac{\tan \lambda_s}{E_s k_s^2} ,$$

$$\frac{\partial \epsilon}{\partial k_s} = -(\pm)_s \frac{P_s^2}{E_s k_s} ,$$

$$\frac{\partial \epsilon}{\partial \phi_m} = 0 ,$$

$$\frac{\partial \epsilon}{\partial \tan \lambda_m} = 0 . \quad (8)$$

When KVEC finishes its calculation and returns control to NOFIT, the k_m will be stored in the calling parameter CURVE of NOFIT. The derivative vector $\frac{\partial k_m}{\partial x_i}$ will be stored in the singly subscripted variable VEC(I), which is in common. The form of VEC(I) will be

$$\text{VEC(I)} = \begin{bmatrix} \frac{\partial k_m}{\partial \phi_1} \\ \frac{\partial k_m}{\partial \tan \lambda_1} \\ \frac{\partial k_m}{\partial k_1} \\ \frac{\partial k_m}{\partial \phi_2} \\ \frac{\partial k_m}{\partial \tan \lambda_2} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \quad (9)$$

When control is returned to subroutine NOFIT, we have the curvature of the m^{th} particle and its derivative vector. Using this calculated curvature we may compute the momentum and energy of this m^{th} particle. We have now arrived at a situation comparable to the 1 constraint case. The ℓ^{th} particle is the only one with missing variables. Two routines must now be called to obtain variables and derivatives for

the particle that is completely unmeasured. Before these routines may be called, the defining factors that communicate the physical situation to all routines must be changed. The FORTRAN statements that change these defining factors are

MISSM = 0 (the m^{th} particle now has all known variables)

NCONST = 1 (the formalism has been changed to a 1 constraint case)

NVAR = NVAR + 1 (the number of known variables has increased by one, since we have just computed k_m).

Subroutine INITCR will calculate the constraint functions using the additional information of the momentum and energy of the m^{th} particle. VDERIV will compute the derivative matrix $U^{\nu+1}$, which is needed for the error correlation. After these two routines have been called, the formalism of the 1 constraint case is finished. We therefore reset the defining factors for the zero constraint case to their original values.

Before the calculation of the error matrices can take place, there is a certain amount of reordering and modification that must be done.

VDERIV has calculated a derivative matrix

$$U_{i\alpha} = \frac{\partial y_\alpha}{\partial x_i} \quad (10)$$

which contains the derivatives of the unmeasured variables with respect to the measured. Remember that this matrix was computed under the 1 constraint situation. It is necessary to modify this matrix to a new

$$\text{matrix, } \left(\frac{\partial y_\alpha}{\partial x_i} \right)' = \frac{\partial y_\alpha}{\partial x_i} + \frac{\partial y_\alpha}{\partial k_m} \frac{\partial k_m}{\partial x_i} \quad (11)$$

The $\frac{\partial k_m}{\partial x_i}$ were calculated by KVEC, and stored in the array VEC(I).

The matrix $\frac{\partial y_\alpha}{\partial x_i}$ is stored in the doubly subscripted array VMAT(I, J) in the form

$$\begin{bmatrix} \frac{\partial \phi_\ell}{\partial \phi_1} & \frac{\partial \tan \lambda_\ell}{\partial \phi_1} & \frac{\partial k_\ell}{\partial \phi_1} \\ \frac{\partial \phi_\ell}{\partial \tan \lambda_1} & \frac{\partial \tan \lambda_\ell}{\partial \tan \lambda_1} & \frac{\partial k_\ell}{\partial \tan \lambda_1} \\ \frac{\partial \phi_\ell}{\partial k_1} & \frac{\partial \tan \lambda_\ell}{\partial k_1} & \frac{\partial k_\ell}{\partial k_1} \\ \vdots & \vdots & \vdots \\ \frac{\partial \phi_\ell}{\partial \phi_m} & \frac{\partial \tan \lambda_\ell}{\partial \phi_m} & \frac{\partial k_\ell}{\partial \phi_m} \\ \frac{\partial \phi_\ell}{\partial \tan \lambda_m} & \frac{\partial \tan \lambda_\ell}{\partial \tan \lambda_m} & \frac{\partial k_\ell}{\partial \tan \lambda_m} \\ \frac{\partial \phi_\ell}{\partial k_m} & \frac{\partial \tan \lambda_\ell}{\partial k_m} & \frac{\partial k_\ell}{\partial k_m} \\ \vdots & \vdots & \vdots \end{bmatrix} \quad (12)$$

We see that the $\frac{\partial y_\ell}{\partial k_m}$ are contained in the $3m^{\text{th}}$ row as

$$\begin{bmatrix} \frac{\partial \phi_\ell}{\partial k_m} & \frac{\partial \tan \lambda_\ell}{\partial k_m} & \frac{\partial k_\ell}{\partial k_m} \end{bmatrix}$$

These elements are transferred to the array ROW(I). For the error matrix calculations, this row must not be part of the matrix $\frac{\partial y_\alpha}{\partial x_i}$.

The reason is quite obvious. When we compute the error correlations under the zero constraint formalism, k_m is not a measured variable. If the particle with the poorly measured momentum is ordered next to last, then we may simply ignore the last row of the matrix. If the particle with poorly measured momentum is not ordered next to last, we must shift the remaining portion of the matrix one row so that $\frac{\partial y_\alpha}{\partial \phi_{m+1}}$ is moved into $\frac{\partial y_\alpha}{\partial k_m}$, $\frac{\partial y_\alpha}{\partial \tan \lambda_{m+1}}$ is moved into $\frac{\partial y_\alpha}{\partial \phi_{m+1}}$, etc. After the necessary shifting has been done, the matrix must be modified by using Eq. (11).

Now we are ready to move the vector $\frac{\partial k_m}{\partial x_i}$ into its proper position in the $\frac{\partial y_\alpha}{\partial x_i}$ matrix. The vector $\frac{\partial k_m}{\partial x_i}$ must occupy the first column, and the original columns of $\frac{\partial y_\alpha}{\partial x_i}$ must be moved one column to the right.

Before control is returned to GUTS, the correlation error matrix of the measured variables vs. measured variables is set equal to the input error matrix G^{-1} .

CALCULATION OF ERROR MATRICES

After the fit has converged, we desire expressions for the uncertainty in the fitted variables. This uncertainty is given in the form of error matrices. The first error matrix that we will consider is the correlation error matrix of the fitted measured variables defined as

$$\left\langle \delta x_i^{\nu+1} \delta x_j^{\nu+1} \right\rangle_{av} = G_{ij}^{-1} - E_{i\lambda}^\nu (H^\nu)^{-1}_{\lambda\mu} (E^\nu)_{\mu j}^T. \quad (1)$$

This error matrix is computed by subroutine COREL1 and is stored in a doubly subscripted array `ERMAT1(I, J)` in the form

$$\begin{bmatrix} \delta\phi_1 & \delta\phi_1 & \delta\phi_1 & \delta\tan\lambda_1 & \delta\phi_1 & \delta k_1 & \cdot & \cdot & \cdot \\ \delta\tan\lambda_1 & \delta\phi_1 & \delta\tan\lambda_1 & \delta\tan\lambda_1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \delta k_1 & \delta\phi_1 & \cdot & \cdot & \delta k_1 & \delta k_1 & \delta\phi_2 & \delta\phi_2 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (2)$$

We can think of this matrix as the correlation error matrix of the measured variables vs. the measured variables. We must now correlate the measured with the unmeasured variables, and the unmeasured with the unmeasured. These remaining error matrices are

$$\begin{aligned} \left\langle \delta x_i^{\nu+1} \delta y_\alpha^{\nu+1} \right\rangle_{av} &= \left\langle \delta x_i^{\nu+1} \delta x_j^{\nu+1} \right\rangle_{av} U_{j\alpha}^{\nu+1}, \\ \left\langle \delta y_\alpha^{\nu+1} \delta y_\beta^{\nu+1} \right\rangle_{av} &= (U^{\nu+1})_{\alpha i}^T \left\langle \delta x_i^{\nu+1} \delta x_j^{\nu+1} \right\rangle_{av} U_{j\beta}^{\nu+1} \end{aligned}$$

$$\text{where } U_{i\alpha}^{\nu+1} = \left. \frac{\partial y_\alpha}{\partial x_i} \right|_{x_i = x_i^{\nu+1}}$$

These remaining error matrices are computed by subroutines COREL2 and COREL3. The matrices will be stored in the doubly subscripted arrays ERMAT2(I, J) and ERMAT3(I, J).

EXAMPLE

Consider the following $\pi^- + P \rightarrow \pi^- + P + \pi^0$. The π^0 is unseen, therefore, unmeasured. This makes our fit a 1 constraint case. If the tracks are numbered

our error matrices will look like

$$\text{ERMAT1(I, J)} = \begin{bmatrix} \delta\phi_1 \delta\phi_1 & \delta\phi_1 \delta\tan\lambda_1 & \delta\phi_1 \delta k_1 & \dots & \delta\phi_1 \delta k_3 \\ \delta\tan\lambda_1 \delta\phi_1 & & & & \\ \delta k_1 \delta\phi_1 & & & & \\ \vdots & & & & \\ \vdots & & & & \\ \delta k_3 \delta\phi_1 & \dots & & & \delta k_3 \delta k_3 \end{bmatrix} \quad (4)$$

(measured vs. measured)

$$\text{ERMAT2(I, J)} = \begin{bmatrix} \delta\phi_4 \delta\phi_4 & \delta\phi_4 \delta\tan\lambda_4 & \delta\phi_4 \delta k_4 \\ \delta\tan\lambda_4 \delta\phi_4 & \delta\tan\lambda_4 \delta\tan\lambda_4 & \delta\tan\lambda_4 \delta k_4 \\ \delta k_4 \delta\phi_4 & \delta k_4 \delta\tan\lambda_4 & \delta k_4 \delta k_4 \end{bmatrix}$$

(unmeasured vs. unmeasured)

$$\text{ERMAT3(I, J)} = \begin{bmatrix} \delta\phi_1 \delta\phi_4 & \delta\phi_1 \delta\tan\lambda_4 & \delta\phi_1 \delta k_4 \\ \delta\tan\lambda_1 \delta\phi_4 & \delta\tan\lambda_1 \delta\tan\lambda_4 & \\ \delta k_1 \delta\phi_4 & & \\ \delta\phi_2 \delta\phi_4 & & \\ \vdots & & \\ \vdots & & \\ \delta k_3 \delta\phi_4 & & \delta k_3 \delta k_4 \end{bmatrix} \quad (6)$$

(unmeasured vs. measured)

When the error matrices have been calculated, the fit is completed. Only two things remain that must be done before the subroutine GUTS can return control to VRTFIT. We must first arrange the variables and the correlation of the errors in a manner designed to make the logic of VRTFIT as simple as possible. This rearrangement is done in subroutine SORTG. The variables and the error correlations are stored in XVERT(I, J) and GINA1(I, J, K) in the same arrangement as the input quantities were stored in XMEAS(I, J) and GINA(I, J, K). The storing of AZ, DIP, and RCRV in XVERT is trivial, so there is no need for a detailed explanation of the FORTRAN statements. The method of transferring the proper elements of the error matrices to GINA1 is not quite straight forward. Rather than discuss the individual FORTRAN steps involved, we will present an explanation of what must be done logically for each constraint case. Recall that when we mention GINA1(I, J, K), the subscript I refers to the i^{th} particle, and for each particle there is a 3×3 matrix whose elements correspond to the J and K indices of GINA1.

Consider breaking up the matrix ERMAT1 into a 3×3 block of the following form for each completely measured particle

$$\begin{bmatrix} \delta\phi_i \delta\phi_i & \delta\phi_i \delta\tan\lambda_i & \delta\phi_i \delta k_i \\ \delta\tan\lambda_i \delta\phi_i & \delta\tan\lambda_i \delta\tan\lambda_i & \delta\tan\lambda_i \delta k_i \\ \delta k_i \delta\phi_i & \delta k_i \delta\tan\lambda_i & \delta k_i \delta k_i \end{bmatrix} \quad (7)$$

For completely measured particles, all that is necessary is to transfer the i^{th} 3×3 block along the diagonal of ERMAT1 to the i^{th} layer of the

3 dimensional array GINA1. For a 4 constraint fit, there are no unmeasured variables, so GINA1 will consist entirely of 3×3 blocks from ERMAT1. Note that the elements not included in the diagonal 3×3 blocks of ERMAT1 are ignored. In the present version of GUTS we assume no correlation between the variables of one particle and the variables of another.*

Now that the elements for the measured variables have been transferred to GINA1, we must consider the error correlations for the unmeasured variables, if the fit is not 4 constraint.

3 Constraint Case

In this case, the ℓ^{th} particle has an unmeasured curvature. The block along the diagonal of ERMAT1 for this particle will be of order 2×2 and will look like

$$\begin{bmatrix} \delta\phi_\ell \delta\phi_\ell & \delta\phi_\ell \delta\tan\lambda_\ell \\ \delta\tan\lambda_\ell \delta\phi_\ell & \delta\tan\lambda_\ell \delta\tan\lambda_\ell \end{bmatrix}. \quad (8)$$

ERMAT2 will consist of only one element in the (1, 1) position as

$$\delta k_\ell \delta k_\ell, \quad (9)$$

and the block of ERMAT3 that corresponds to the ℓ^{th} particle will be of the form

$$\begin{bmatrix} \delta\phi_\ell \delta k_\ell \\ \delta\tan\lambda_\ell \delta k_\ell \end{bmatrix}. \quad (10)$$

*It would not be very difficult to change this limitation of the present program.

If we refer to the these specific sections of ERMAT1, ERMAT2, and ERMAT3 as ϵ_1 , ϵ_2 and ϵ_3 we see that after these sections have been transfered to the ℓ^{th} layer of GINA1, it will be of the form

$$\begin{array}{|c|c|} \hline \epsilon_1 & \epsilon_3 \\ \hline \epsilon_3^T & \epsilon_2 \\ \hline \end{array},$$

where ϵ_3^T is the transpose of the vector ϵ_3 .

2 Constraint Case

Two curvatures k_ℓ and k_m are unmeasured. The storage of the error correlations will be exactly the same as the 3 constraint case except for the fact that there is one more particle with missing curvature to be considered.

1 Constraint Case

Here we have all variables associated with the ℓ^{th} particle missing. There will be no block along the diagonal of ERMAT1 for this particle. ERMAT2 will be of the form

$$\begin{bmatrix} \delta\phi_\ell \delta\phi_\ell & \delta\phi_\ell \delta\tan\lambda_\ell & \delta\phi_\ell \delta k_\ell \\ \delta\tan\lambda_\ell \delta\phi_\ell & \delta\tan\lambda_\ell \delta\tan\lambda_\ell & \delta\tan\lambda_\ell \delta k_\ell \\ \delta k_\ell \delta\phi_\ell & \delta k_\ell \delta\tan\lambda_\ell & \delta k_\ell \delta k_\ell \end{bmatrix} \quad (11)$$

All that is necessary for this case is to transfer this 3 x 3 matrix to the ℓ^{th} 3 x 3 layer of GINA1.

0 Constraint Case

This case involves a particle, m , that has a missing curvature, and a particle, l , that has all variables missing. ERMAT1 will have no elements for the l^{th} particle, and the block along the diagonal for the m^{th} particle will look like

$$\begin{bmatrix} \delta\phi_m & \delta\phi_m & \delta\phi_m \delta\tan\lambda_m \\ \delta\tan\lambda_m \delta\phi_m & \delta\tan\lambda_m \delta\tan\lambda_m \end{bmatrix}, \quad (12)$$

which we will refer to as ϵ_1 . ERMAT2 will be of the form

$$\begin{bmatrix} \delta k_m \delta k_m & \delta k_m \delta\phi_l & \delta k_m \delta\tan\lambda_l & \delta k_m \delta k_l \\ \delta\phi_l \delta k_m & \delta\phi_l \delta\phi_l & \delta\phi_l \delta\tan\lambda_l & \delta\phi_l \delta k_l \\ \delta\tan\lambda_l \delta k_m & \delta\tan\lambda_l \delta\phi_l & \delta\tan\lambda_l \delta\tan\lambda_l & \delta\tan\lambda_l \delta k_l \\ \delta k_l \delta k_m & \delta k_l \delta\phi_l & \delta k_l \delta\tan\lambda_l & \delta k_l \delta k_l \end{bmatrix}. \quad (13)$$

Since we are only considering correlations of variables for each particle, we will ignore the elements of ERMAT2 that have subscripts of both l and m . This leaves us with ERMAT2(1,1), which we will refer to as

$$\epsilon_2^{(m)}.$$

Also we have the lower right hand 3×3 section of ERMAT2 which we will refer to as

$$\epsilon_2^{(l)}.$$

We will use one block of ERMAT3 corresponding to the m^{th} particle.

This block will be of the form

$$\begin{bmatrix} \delta\phi_m & \delta k_m \\ \delta \tan \lambda_m & \delta k_m \end{bmatrix}, \quad (14)$$

which will be referred to as ϵ_3 .

The m^{th} layer of GINA1 will become

ϵ_1	ϵ_3
ϵ_3^T	$\epsilon_2^{(m)}$

The ℓ^{th} layer of GINA1 will become the entire 3×3 block

$$\epsilon_2^{(\ell)}.$$

TRANSFORMATION TO CENTER OF MASS

After the GUTS fit is completed, the variables are in terms of the laboratory system. It is desirable to transform these quantities to the center of mass system.

For purposes of clarity, we will refer to the X, Y and Z coordinates as X_1 , X_2 and X_3 respectively.

The first step in the transformation to C. M. is to transform the coordinate system so that the X_1 axis will coincide with the vector OP. This is really the rotation of the chamber coordinate system to a system whose X_1 axis is aligned with the beam track. The components in different coordinate systems are related by

$$X'_1 = a_{11} X_1 + a_{12} X_2 + a_{13} X_3,$$

$$X'_2 = a_{21} X_1 + a_{22} X_2 + a_{23} X_3,$$

$$X'_3 = a_{31} X_1 + a_{32} X_2 + a_{33} X_3;$$

or, in general,

$$X'_\alpha = a_{\alpha\beta} X_\beta,$$

where the $a_{\alpha\beta}$ are the direction cosines between the α^{th} primed axis, and the β^{th} unprimed axis. For a rotation of the primed axis to the unprimed, we rotate around the X_3 axis by the azimuth angle. The direction cosines for this rotation will be

$$a_{11} = \cos \alpha, \quad a_{12} = \cos\left(\frac{\pi}{2} - \alpha\right) = \sin \alpha,$$

$$a_{13} = \cos\left(\frac{\pi}{2}\right) = 0,$$

$$a_{21} = \cos\left(\frac{\pi}{2} + \alpha\right) = -\sin \alpha, \quad a_{22} = \cos \alpha,$$

$$a_{23} = \cos\left(\frac{\pi}{2}\right) = 0,$$

$$a_{31} = \cos\left(\frac{\pi}{2}\right) = 0, \quad a_{32} = \cos\left(\frac{\pi}{2}\right) = 0,$$

$$a_{33} = \cos(0) = 1.$$

And so, the rotation matrix about X_3 is

$$A = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

If we now rotate the system about the X'_2 axis by the dip angle, the rotation matrix will be

$$B = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} .$$

The product rotation matrix is

$$\begin{aligned} C = BA &= \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \beta \cos \alpha & \cos \beta \sin \alpha & \sin \beta \\ -\sin \alpha & \cos \alpha & 0 \\ -\sin \beta \cos \alpha & -\sin \beta \sin \alpha & \cos \beta \end{bmatrix} . \end{aligned}$$

To visualize the physical significance of this rotation, consider a particle with momentum P along the directed line segment $\overrightarrow{OP}$. The components of the momentum along the unprimed axes are

$$P_1 = P \cos \beta \cos \alpha ,$$

$$P_2 = P \cos \beta \sin \alpha$$

$$P_3 = P \sin \beta .$$

Representing these components as a 3 vector

$$\vec{P}^{\circ} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Transforming this vector to a new vector $\vec{P}''$ in the double prime system,

$$\begin{aligned} \vec{P}'' = C\vec{P}^{\circ} &= \begin{bmatrix} \cos\beta\cos\alpha & \cos\beta\sin\alpha & \sin\beta \\ -\sin\alpha & \cos\beta & 0 \\ -\sin\beta\cos\alpha & -\sin\beta\sin\alpha & \cos\beta \end{bmatrix} \begin{bmatrix} P\cos\beta\cos\alpha \\ P\cos\beta\sin\alpha \\ P\sin\beta \end{bmatrix} \\ &= \begin{bmatrix} P\cos^2\beta\cos^2\alpha + P\cos^2\beta\sin^2\alpha + P\sin^2\beta \\ -P\cos\beta\cos\alpha\sin\alpha + P\cos\beta\sin\alpha\cos\alpha \\ -P\cos\beta\cos^2\alpha\sin\beta - P\cos\beta\sin^2\alpha\sin\beta + P\sin\beta\cos\beta \end{bmatrix} \\ &= \begin{bmatrix} P\cos^2\beta(\cos^2\alpha + \sin^2\alpha) + P\sin^2\beta \\ 0 \\ -P\cos\beta\sin\beta(\cos^2\alpha + \sin^2\alpha) + P\cos\beta\sin\beta \end{bmatrix} \\ &= \begin{bmatrix} P \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

Thus, we have transformed our system so that a particle has momentum only along the X_1'' axis.

Now the relativistic transformation from the laboratory to the C. M. can be accomplished by multiplying the momentum 4 vector of a particle by the Lorentz transformation matrix

$$\begin{bmatrix} \Gamma & 0 & 0 & -iN \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ iN & 0 & 0 & \Gamma \end{bmatrix}.$$

Consider a two body interaction in the laboratory system with the incoming particle having mass M_I and the target particle having mass M_T . Expressing the momentum of each in the double prime system

$$P_I'' = \begin{bmatrix} P \\ 0 \\ 0 \\ \frac{-iE_I}{c} \end{bmatrix} \quad P_T'' = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{-iM_T}{c} \end{bmatrix}.$$

We have introduced here the momentum-energy 4 vector, where the first 3 elements are the X_1 , X_2 and X_3 components of the momentum. For the following discussion, we will use lower case letters for C. M. quantities and capital letters for laboratory quantities.

The transformation to the C. M. system for the momentum-energy vectors is

$$\begin{bmatrix} \Gamma & 0 & 0 & -iN \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ iN & 0 & 0 & \Gamma \end{bmatrix} \begin{bmatrix} P \\ 0 \\ 0 \\ -\frac{i}{c} E_I \end{bmatrix} = \begin{bmatrix} \Gamma P - \frac{N}{c} E_I \\ 0 \\ 0 \\ iNP - \frac{i}{c} \Gamma E_I \end{bmatrix} = \begin{bmatrix} p_I \\ 0 \\ 0 \\ -\frac{i}{c} c_I \end{bmatrix},$$

$$\begin{bmatrix} \Gamma & 0 & 0 & -iN \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ iN & 0 & 0 & \Gamma \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{i}{c} M_T c^2 \end{bmatrix} = \begin{bmatrix} -NM_T c \\ 0 \\ 0 \\ -i\Gamma M_T c \end{bmatrix}.$$

By definition of the C.M. system,

$$\sum_i \vec{p}_i = 0, \quad \text{so} \quad (1)$$

$$\Gamma P - \frac{N}{c} E_I - N M_I c = 0.$$

This is one equation in two unknowns; however, relations of Γ and N can be derived for $P_I^2 + M_I^2 c^2 = \frac{e_I^2}{c^2}$, remembering that the mass is a scalar, therefore invariant to the Lorentz transformation, or $M_I = m_I$. So, in the C.M. system

$$P_I^2 + M_I^2 c^2 = \frac{e_I^2}{c^2}. \quad (2)$$

Upon substitution of the lab quantities, we have

$$\begin{aligned} \left(\Gamma P - \frac{N}{c} E_I \right)^2 + M_I^2 c^4 &= \left(\frac{\Gamma}{c} E_I - N P \right)^2, \\ \Gamma^2 P^2 + \frac{N^2}{c^2} E_I^2 - 2 \frac{\Gamma}{c} N E_I P + M_I^2 c^2 &= \frac{\Gamma^2}{c^2} E_I^2 + N^2 P^2 - 2 \frac{\Gamma}{c} N E_I P, \\ \Gamma^2 P^2 - \frac{\Gamma^2}{c^2} E_I^2 &= N^2 P^2 - \frac{N^2}{c^2} E_I^2 - M_I^2 c^2, \\ \Gamma^2 \left(P^2 - \frac{1}{c^2} E_I^2 \right) &= N^2 \left(P^2 - \frac{1}{c^2} E_I^2 \right) - M_I^2 c^2, \\ \Gamma^2 \left(\frac{E_I^2}{c^2} - P^2 \right) &= N^2 \left(\frac{E_I^2}{c^2} - P^2 \right) + M_I^2 c^2, \end{aligned}$$

substitution of $M_I^2 c^2 = \frac{E_I^2}{c^2} - P^2$ gives $\Gamma^2 M_I^2 c^2 = N^2 M_I^2 c^2 + M_I^2 c^2$,

$$\text{or } \Gamma^2 = N^2 + 1$$

From Eq. 1 $\Gamma P = N \left(\frac{E_I}{c} + M_T c \right),$

$$\Gamma^2 P^2 = N^2 \left(\frac{E_I^2}{c^2} + M_T^2 c^2 + 2 E_I M_T \right);$$

and, since $\Gamma^2 - 1 = N^2$, we have

$$\frac{E_I^2}{c^2} + M_T^2 c^2 + 2 E_I M_T = \Gamma^2 \left[\frac{E_I^2}{c^2} - P^2 + M_T^2 c^2 + 2 E_I M_T \right],$$

$$E_I^2 + M_T^2 c^4 + 2 E_I M_T c^2 = \Gamma^2 [E_I^2 - P^2 c^2 + M_T^2 c^4 + 2 E_I M_T c^2],$$

$$(E_I + M_T c^2)^2 = \Gamma^2 [-P^2 c^2 + (E_I + M_T c^2)^2].$$

So,

$$\Gamma = \frac{E_I + M_T c^2}{\sqrt{(E_I + M_T c^2)^2 - P^2 c^2}}.$$

Now, expressing the mass of any particle in terms of its energy $M c^2$ in MeV, that is

$$M^* = M c^2,$$

and substituting this in the expression relating momentum and energy,

$$p^2 c^2 + (M c^2)^2 = e^2,$$

we have

$$p^2 c^2 + M^{*2} = e^2.$$

Let us further express the momentum in terms of MeV/c, that is

$$\frac{p^*}{c} = p.$$

When this substituted in the previous equation, we have

$$(p^*)^2 + (M^*)^2 = e^2.$$

Our equation for Γ now becomes

$$\Gamma = \frac{E_I + M_T^*}{\sqrt{(E_I + M^*)^2 - (P_I^*)^2}}$$

We will drop the * notation remembering that mass will always be in terms of MeV and momentum in MeV/c.

The transformation matrix is now completely defined for a particle with momentum in the X_1'' direction only.

We can now use the Lorentz transformation to transform all the variables from the GUTS fit to the C.M. system. For the following discussion:

1. All variables referring to the incident particle are indexed by (I).
2. All variables referring to the target particle are indexed by (T).
3. All variables referring to any particle are indexed by (j),

for $j=1, 2, \dots, n$ where n is the total number of incoming plus outgoing particles.

We have derived

$$\Gamma = \frac{M_T + E_I}{\sqrt{(M_T + E_I)^2 - P_I^2}}$$

$$N = \sqrt{\Gamma^2 - 1}$$

The 4 vector of particle j is given by

$$\begin{bmatrix} \Gamma & 0 & 0 & -iN \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ iN & 0 & 0 & \Gamma \end{bmatrix} \begin{bmatrix} \cos\beta(1)\cos\alpha(1) & \cos\beta(1)\sin\alpha(1) & \sin\beta(1) & 0 \\ -\sin\alpha(1) & \cos\alpha(1) & 0 & 0 \\ -\sin\beta(1)\cos\alpha(1) & -\sin\beta(1)\sin\alpha(1) & \cos\beta(1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X(P(j))$$

$$= \begin{bmatrix} \Gamma\cos\beta(1)\cos\alpha(1) & \Gamma\cos\beta(1)\sin\alpha(1) & \Gamma\sin\beta(1) & -iN \\ -\sin\alpha(1) & \cos\alpha(1) & 0 & 0 \\ -\sin\beta(1)\cos\alpha(1) & -\sin\beta(1)\sin\alpha(1) & \cos\beta(1) & 0 \\ iN\cos\beta(1)\cos\alpha(1) & iN\cos\beta(1)\sin\alpha(1) & iN\sin\beta(1) & \Gamma \end{bmatrix}$$

$$X \begin{bmatrix} P_j \cos\beta(j)\cos\alpha(j) \\ P_j \cos\beta(j)\sin\alpha(j) \\ P_j \sin\beta(j) \\ -iE_j \end{bmatrix} = \begin{bmatrix} P_x(j) \\ P_y(j) \\ P_z(j) \\ -ie_j \end{bmatrix}$$

$$\text{where } E_j = \sqrt{P_j^2 + M_j^2}$$

After multiplication we have the components of the 4 vector in the C, M. system as

$$\begin{aligned} P_x(j) &= P_j \cos\beta(j) \cos\alpha(j) \Gamma\cos\beta(1)\cos\alpha(1) \\ &+ P_j \cos\beta(j) \sin\alpha(j) \Gamma\cos\beta(1)\sin\alpha(1) \\ &+ P_j \sin\beta(j) \Gamma\sin\beta(1) - NE_j, \end{aligned}$$

$$P_y(j) = -P_j \cos\beta(j) \cos\alpha(j) \sin\alpha(1) + P_j \cos\beta(j) \sin\alpha(j) \cos\alpha(1),$$

$$P_z(j) = -P_j \cos\beta(j) \cos\alpha(j) \sin\beta(1) \cos\alpha(1) \\ - P_j \cos\beta(j) \sin\alpha(j) \sin\beta(1) \sin\alpha(1) + P_j \sin\beta(j) \cos\beta(1),$$

$$e_j = \Gamma E_j - P_j \sin\beta(j) N \sin\beta(1) \\ - P_j \cos\beta(j) \sin\alpha(j) N \cos\beta(1) \sin\alpha(1) \\ - P_j \cos\beta(j) \cos\alpha(j) N \cos\beta(1) \cos\alpha(1).$$

Recall that in the above formulae, α 's are azimuth angles and β 's are dip angles. From the relations

$$P_j = \sqrt{P_x(j)^2 + P_y(j)^2 + P_z(j)^2},$$

$$\cos \theta(j) = \frac{P_x(j)}{P(j)},$$

$$\sin \theta(j) = \frac{P_y(j)}{P(j)},$$

$$\cos \phi(j) = \frac{P_z(j)}{P(j) \sin \theta(j)},$$

$$\sin \phi(j) = \frac{P_y(j)}{P(j) \sin \theta(j)},$$

we may compute the C.M. quantities $\theta(j)$, $\cos \theta(j)$, $P(j)$, $\phi(j)$.

The center of mass quantities are computed by subroutine CMQUAN. In general, the variables for the incoming particle will not be stored first in the variable arrays such as AZ, DIP, etc. CMQUAN must first determine the proper index for the incoming particle by testing the sign

of elements in the array ONE(I). The element that is negative, for example, the j th, indicates that the variables of the incoming particle must be referred to by the index j . The elements of the Lorentz transformation matrix Γ and N are then calculated by CMQUAN, and are stored in GAMMA and FN respectively. Calculation of the C.M. quantities use the equations as we have defined them here. The results for $\theta(j)$, $\cos \theta(j)$, $p(j)$, and $\phi(j)$ are stored in CMTHTA(J), CMCOS(J), PCM(J), and PSI(J) respectively.

ACKNOWLEDGMENTS

We wish to express our appreciation to Dr. Lawrence Leipuner of B. N. L. and his staff for their cooperation and help in testing this system. The test events and machine language listings were a major factor in the completion of our system.

Also we wish to thank Ann McDonald of the Computer Services Dept. here who contributed so much of her time and effort to this project.

APPENDIX I

Assignment No.	Particle
----------------	----------

1 -----	π^+
2 -----	π^-
3 -----	π^0
4 -----	K^+
5 -----	K^-
6 -----	K^0
7 -----	γ
8 -----	P
9 -----	N^0
10 -----	Λ^0
11 -----	Σ^+
12 -----	Σ^-
13 -----	Σ^0
14 -----	Ξ^-
15 -----	Ξ^0
16 -----	e^+
17 -----	e^-
18 -----	μ^+
19 -----	μ^-
20 -----	ν
21 -----	$\bar{K}^0$
26 -----	Ω^-

APPENDIX II - Reject Numbers and Messages

Reject No.	Message
1	Energy or momentum negative
2	Too many cutdowns
3	CHI^2 too large on first step
4	Too many iterations
5	HLM singular
6	MPRIME not equal to GMFCN
7	GYFCN imaginary
8	FNORM imaginary
9	Missing mass of V imaginary
10	CHI^2 large after 15 steps
11	GYFCN singular
12	Divide check
14	GINA singular
15	Imaginary momentum in zero constraint
16	Error in missing mass imaginary
17	Negative momentum or energy in zero constraint
18	A diagonal element of GINA1 is less than or equal to zero
19	Error in momentum is imaginary
20	CHI^2 too large, $\text{CHI}^2 = \underline{\hspace{1cm}}$
21	Negative curvature on initial step, two constraint case
30	Poor momentum measurement
80	Missing mass for type <u> </u> particle rejected in VFIT (no.)
90	VFIT, three body decay
111	Zero particles in VRTFIT

APPENDIX III

K-Experiment

TOP 1

Vertex I - Interaction

Particle Conf. No.	Track	Target	Track 3	Track 2	Neutral Track
201(302)	K	+	P	+	(π^0)
303			π^+	+	N
301			P	+	$\bar{K}^0$
305			π^+	+	Λ^0
319			K^+	+	Ξ^0
313			π^+	+	Σ^0
315			K^+	+	Σ^0
307			K^+	+	Λ^0

Notation: Parentheses around last particle indicate that it may or may not occur. If it does occur, its particle configuration number is also in parentheses.

REFERENCES

- ¹J. P. Berge and F. T. Solmitz, "Kinematical Analysis of Interaction Vertices from Bubble Chamber Data", Rev. Sci. Instr. 32, 538-547 (1951).
- ²W. J. Kernan, W. J. Higby and I. H. Boessenroth, "Writing Topological Programs for I. S. U. GUTS", in preparation.