$\beta = \gamma + (1/\gamma) - 2$

$\rho = (x_1 x_2 x_3)/(X_1 X_2 X_3)$

$\alpha = (\mu\sigma^3)/(\sigma_1 \sigma_2 x_3{}^3)$

The equations (1) and (2) now will serve to determine X_1 and X_2 using the value of W which is determined from (3) and (4). The Lagrange multiplier has already disappeared, and we have 2 equations for the variables W and X_3. The other quantities defined above are just convenient combinations of the input variables. It is useful to replace X_3 with another scaled variable

$$A = (X_3/x_3) - 1$$

so (3) and (4) become

$$(3') \qquad (1+A)^2 \; A = W \; \alpha$$

$$(4') \qquad (1-W^2)^2 = \rho(A+1)\Big((1+W^2)+\beta W\Big)$$

The solution to this pair of equations is actually done by Newton's method applied to (3'). That is we evaluate the function

$$f(W) = (1+A)^2 \; A = W \; \alpha$$

where A is found from

$$A(W) = \frac{1-(1-W^2)}{\rho*(\;(1+W)^2+\beta W)}$$

The derivative is also calculated as required. The function W is only meaningful for

$$\min(1/\gamma, \gamma) < W < 1$$

and the program requires appropriate protection from exceeding these bounds (which can lead to spurious solutions) as well as jumping around the solution. This is done by cutting the step given by the Newton formula.

The description above refers to the case of fitting all three of the variables. In case the angle is held fixed (NDIM=2), (3) is no longer useful and the function for which the zero in the physical region is sought is A(W).

The CALLIGRA input file for this note is F32BOB.S.GGFIT

Preliminary JADE Computer Note No. 54

24.11.81
E. Elsen

Detector Monte Carlo

A. Tracking

The 4-vector input format is described in JADE Computer Note 10. Different from JADE Computer Note 26 the main libraries used are

F22ELS. JMC.S (source)
F22ELS. JMC.L (load)

An example including full JCL is given in member #PRODUCT on the main library. The constants used in this step are given and explained in member BLDAT. Notice that the magnetic field is now negative as in data.

B. Reading

Tracked data are not yet in the final format. Special reading routines have to be called to take account of finite resolutions, broken I.D. cells, trigger conditions etc. This is automatically done when the supervisor - or TP-program is called (refer to J.C.N. 25 for other cases). The constants used in this step are held in member RDSTAT, which describes the detector status in 1981.

For other running periods different conditions apply. They are described in the following members, which have to be specially included in the reading program in the linkage step.

RDST0179	for	1979
RDST0180	"	1980
RDST0280	"	1980
RDST0181	"	1981

C. Radiative photons

A new tracking program is available that propagates photons radiated off electrons in the beam pipe, pressure vessel etc.. Together with a format extension of the VECT-bank (see below) it involves program changes affecting several routines including subroutine MCJADE, the steering routine. To minimize confusion and to allow for a gradual change-over this new feature is not yet made a standard. However, the new routines are called if the library

F22ELS.MCDEN.L

is linked in front of JMC.L. The new routines will be copied to JMC.L in the near future.

VECT, 1: 4-vectors and origins of particles produced in the detector (except for decays of K^0).

I * 4	1	length of header LØ' (=4)
"	2	length of particle data L1' (=12)
"	3	event no.
"	4	no. of particles following = nf_2

LØ'+1 … LØ'+L1' : data of 1. track

LØ'+L1'+1 … LØ'+2*L1' : data of 2. track

etc.

- data for a track:

R * 4	1	px	
"	2	py	
"	3	pz	
"	4	E	
"	5	m	
I * 4	6	q	
I * 4	7	type	
R * 4	8	x	of origin
"	9	y	of origin
"	10	z	of origin
I * 2	21	VECT bank no.	of mother particle
I * 2	22	particle no.	of mother particle
R * 4	12	path length of mother particle before decay/interaction	

JADE - Computer Note No. 54

17.5.1982
E. Elsen

Detector Monte Carlo

A. Tracking

The 4-vector input format is described in JADE Computer Note 10. Different from JADE Computer Note 26 the main libraries used are

F22ELS.JMC.S (source
F22ELS.JMC.L (load)

An example including full JCL is given in member #PRODUCT on the main library. The constants used in this step are given and explained in member BLDAT. Notice that the magnetic field is now negative in data.

#PRODUCT will create Monte Carlo Data without muon hits. Data including μ-chamber tracking are generated with the JCL stored in member #PRODMU (refer to J.C.N. 55 for further details of the μ-tracking).

B. Reading

Tracked data are not yet in the final format. Special reading routines have to be called to take account of finite resolutions, broken I.D. cells, trigger conditions etc. This is automatically done when the supervisor - or TP-program is called (refer to J.C.N. 25 for other cases). The constants used in this step are held in member RDTRG. RDTRG will look at the date as specified in the bank 'HEAD' and select the trigger conditions accordingly. The status of the inner detector (dead cells) is changed through this routine, too.

If the user wished to alter the detector status of the data he or she has to overwrite the time information in bank "HEAD" before RDTRG is called. This can be done most conveniently by supplying an own version of the routine RDDATE which is a dummy routine on JMC.L.

C. Radiative Photons

The tracking program propagates also photons radiated off electrons in the beam pipe, pressure vessel etc.. This addition initiated a format extension of the VECT-bank (see below).

VECT, 1: 4-vectors and origins of particles produced in the detector (except for decays of long-lived neutrals, where the decay products have to be provided).

I * 4	1	length of header LØ' (=4)
"	2	length of particle data L1' (=12)
"	3	event no.
"	4	no. of particles following = nf_2

LØ'+1
.
.
.
LØ'+L1' } data of 1. track

LØ'+L1'+1
.
.
.
LØ'+2*L1' } data of 2. track

repeated nf_2 times.

- data for a track:

R * 4	1	px	
"	2	py	
"	3	pz	
"	4	E	
"	5	m	
I * 4	6	q	
I * 4	7	type	
R * 4	8	x	of origin
"	9	y	
"	10	z	
I * 2	21	VECT bank no.	of mother particle
I * 2	22	particle no.	
R * 4	12	path length of mother particle before decay/interaction	

How to use the Muon Tracking Monte Carlo at DESY

JADE COMPUTER NOTE 55
Roger Barlow
November 1981

Particles are tracked through the muon filter using the Cascade program due to Alan Grant. The results produced look tolerably like the real thing, though no guarantee is given - various hilarious mistakes have been found in the original code, and I shouldn't think they've all been discovered yet.

Be warned that this program is SLOW! Of the order of 1 second per track, increasing sharply with energy.

The source and load libraries are F22RJB.RLMC.S and F22RJB.RLMC.L. There is a sample program in F22RJB.RLMC.S(#EXAMPLE). It uses default values for various options. The defaults are fairly sensible, if you insist on changing them, read on.

Apart from the hadron cascade, there is also a different treatment of calibration constants. This is necessary because the muon filter studies need more control over the chamber constants being used.

Under the new system the date on which the data is supposed to have been taken is set by the user (not just taken as the day on which the Monte-Carlo job runs). Appropriate calibration data for the chosen date are then loaded from the O'Neill datasets. Note that for chamber constants KALIBR and co. still treat this as Monte-Carlo data and smearing etc. is performed as before.

The default date and time are 1030 am on 1 April 1980. To change this, set values in

COMMON/TODAY/HSEC,HMIN,HHOUR,HDAY,HMONTH,HYEAR

Another change is that input is now done by a seperate subroutine MCREAD This does not affect normal use in any way, but if you want to read a non-standard 4-vector tape, or use an internal 4-vector generator, you can put in your own version of MCREAD.

Control of the muon chamber noise, efficiency, and resolutions is done by the routine MUFIX. This is called by MCJADE, so parameters can be set by the arguments in the subroutine call, or by putting in your own version. The latter way of doing it is used if you want to alter the muon constants after they have been read in - to make some dead chambers actually work, for example. If this is done, then the muon finding in the supervisor should be told about it by calling the same version of MUFIX from USER, after the constants have been read in.

For simple use, the call in MCJADE is

```
   CALL MUFIX(SIGT,SIGL,EFF,RANDOM,DEBUG,SIGTEW)

     where SIGT   is the barrel transverse    resolution (default 10 mm)
           SIGL   is the barrel longitudinal resolution (default 500 mm)
           EFF    is the chamber efficiency              (default 95%)
           RANDOM is the random noise level              (default 1%)
           DEBUG  is a debugging print flag              (default .FALSE.)
           SIGTEW IS THE ENDWALL TRANSVERSE  RESOLUTION (DEFAULT 15 MM)
```

Note : Apart from the muon routines, the routines in this version that are different from the standard version are:

MCJADE DATEMC WRTMCB

JCL
===

You must include the extra libraries

F22RJB.RLMC.L
F11LHO.JADEGL
F22ALL.JADEMUL
SYS1.NAGLIB

F22RJB.RLMC.L must be before F22ELS.JMC.L

You also need the calibration datasets in the GO step for units 21 & 22

An example follows. It exists in F22RJB.RLMC.S(#EXAMPLE)

```
//F22RJBMC JOB 10622622,ROGER,MSGLEVEL=(0,0),CLASS=E,TIME=(00,30)
//*MAIN LINES=(10),ORG=EXT,RELPRI=MED
// EXEC FORTHCLG,PARM.FORT='NOSOURCE,XL',REGION.GO=912K
      IMPLICIT INTEGER*2 (H)
      COMMON / BCS / IW(25000)
      COMMON/TODAY/HDATE(6)
C  SET DATE AND TIME OF RUN  - 12 NOON ON 1ST MAY 1981
      HDATE(1)=0
      HDATE(2)=0
      HDATE(3)=12
      HDATE(4)=1
      HDATE(5)=5
      HDATE(6)=1981
      CALL BINT( 25000,5000,100,100)
      CALL MCJADE(10,100)
      CALL BSTA
      STOP
      END
//LKED.SYSLIB DD
//            DD
//            DD DSN=SYS1.NAGLIB,DISP=SHR,UNIT=FAST
//            DD DISP=SHR,UNIT=FAST,DSN=F22ALL.JADEMUL
//            DD DISP=SHR,UNIT=FAST,DSN=F11LHO.JADEGL
//            DD DISP=SHR,UNIT=FAST,DSN=JADELG.LOAD
//            DD DISP=SHR,UNIT=FAST,DSN=F22RJB.RLMC.L
//            DD DISP=SHR,UNIT=FAST,DSN=F22ELS.JMC.L
//            DD DSN=R02BJT.CERNLIB,DISP=SHR,UNIT=FAST
//            DD DSN=F1EBLO.BOSLIB.L,DISP=SHR,UNIT=FAST
//GO.FT03F001 DD DISP=SHR,UNIT=FAST,DSN=F22RJB.MU1
//GO.FT02F001 DD DISP=SHR,UNIT=FAST,DSN=F22RJB.MU2
//GO.FT21F001 DD UNIT=FAST,DISP=SHR,DSN=F11LHO.AUPDAT0
//GO.FT22F001 DD UNIT=FAST,DISP=SHR,DSN=F11LHO.AUPDAT1
```

Olsson

JADE COMPUTER NOTE 56
=====================

STATUS 14/01/82
P. STEFFEN

THIS IS JADEPR.TEXT(CPNOTE56)

SUBROUTINE MCTRCO
==

SEARCH FOR A TRACK IN 'PATR'-BANK 10 - OR ANY OTHER PATR BANK - WHICH CORRESPONDS TO A SELECTED MONTE CARLO GENERATED TRACK OF 'PATR'-BANK 12.

```
1.)  CALL   MCTRCO(IPTR1,IPPATR,JPTR2,CHSQTR)
     ------------------------------------------------

     - INPUT :
          IPTR1   : POINTER TO SELECTED TRACK IN PATR(12)-BANK
                    (E.G. IDATA(IPTR1+1) = TRACK # )
          IPPATR  : POINTER TO PATR(10)-BANK (BOS-POINTER)

     - OUTPUT:
          JPTR2   : POINTER TO FOUND TRACK IN PATR(10)-BANK
                    (E.G. IDATA(IPTR2+1) = TRACK # )
                    = 0 IF NO TRACK FOUND
          CHSQTR  : ARRAY OF 5 WORDS WITH RESULTS
                    CHSQTR(1) < 20. IF PROPER TRACK FOUND

     - TRACK IS SEARCHED FOR BY COMPARING POSITION + DIRECTION OF
       FIRST + LAST POINT OF TRACKS (ONLY IN THE X-Y-PLANE)
       CHI**2 = 1/4 * ( (DXY/.40)**2 + (DFI/.010)**2     : FIRST POINT
                      + (DXY/.40)**2 + (DFI/.010)**2  )  : LAST  POINT

       CHSQTR(1)   : CHI**2 ( < 20. IF PROPER TRACK FOUND)
       CHSQTR(2)   : DXY(FIRST POINT ON TRACK)
       CHSQTR(3)   : DXY(LAST  POINT ON TRACK)
       CHSQTR(4)   : DFI(FIRST POINT ON TRACK)
       CHSQTR(5)   : DFI(LAST  POINT ON TRACK)
```

2.) EXEMPLE FOR APPLICATION OF MCTRCO:

1. SELECT TRACK FROM VECT-BANK

2. SEARCH FOR TRACK IN PATR(12)-BANK COMPARING TRACK# + TYP# (2. WORD IN TRACK ARRAY OF PATR(12)). THIS DOES NOT WORK FOR SOME MC-DATA CALCULATED IN TOKYO, BECAUSE WORD 2 OF THE TRACKS IN THE PATR-BANK(12) IS NOT SET CORRECTLY (TRACK # IN VECT-BANK = 0).

3. CALL MCTRCO: TRACK FOUND IF JPTR2 ◇ 0 .AND. CHSQTR(1) < 20.

4. 'F11PST.JADESR(MKLOTR)' IS AN EXEMPLE FOR MARKING PROTON + PION TRACKS FROM LAMBDA -> P, PI

3.) EFFICIENCY: ABOUT 3% OF THE TRACKS ARE NOT FOUND. THESE ARE IN GENERAL OF LOW MOMENTUM (ABOUT 50 MEV). OR SHORT TRACKS (10 HITS OR LESS)

4.) COMMENT: THE PROGRAM WILL BE IMPROVED IF BETTER TRACK PARAMETERS WILL BE STORED IN PATR-BANK(12) BY THE MC-TRACKING PROGRAM

```
       JADE COMPUTER NOTE 56                        STATUS  14/01/82
       =====================                        P. STEFFEN

       THIS IS JADEPR.TEXT(CPNOTE56)

       SUBROUTINE   MCTRCO
       ===============================================================

       SEARCH FOR A TRACK IN 'PATR'-BANK 10 - OR ANY OTHER PATR BANK -
       WHICH  CORRESPONDS TO A SELECTED MONTE CARLO GENERATED TRACK OF
       'PATR'-BANK 12.

 1.)   CALL   MCTRCO(IPTR1,IPPATR,JPTR2,CHSQTR)
       ---------------------------------------------

       - INPUT :
           IPTR1   : POINTER TO SELECTED TRACK IN PATR(12)-BANK
                     (E.G. IDATA(IPTR1+1) = TRACK # )
           IPPATR  : POINTER TO PATR(10)-BANK (BOS-POINTER)

       - OUTPUT:
           JPTR2   : POINTER TO FOUND TRACK IN PATR(10)-BANK
                     (E.G. IDATA(IPTR2+1) = TRACK # )
                     = 0 IF NO TRACK FOUND
           CHSQTR  : ARRAY OF 5 WORDS WITH RESULTS
                     CHSQTR(1) < 20. IF PROPER TRACK FOUND

       - TRACK IS SEARCHED FOR BY COMPARING POSITION + DIRECTION OF
         FIRST + LAST POINT OF TRACKS (ONLY IN THE X-Y-PLANE)
         CHI**2 = 1/4 * ( (DXY/.40)**2 + (DFI/.010)**2     : FIRST POINT
                        + (DXY/.40)**2 + (DFI/.010)**2 )   : LAST  POINT

         CHSQTR(1)   : CHI**2 ( < 20. IF PROPER TRACK FOUND)
         CHSQTR(2)   : DXY(FIRST POINT ON TRACK)
         CHSQTR(3)   : DXY(LAST  POINT ON TRACK)
         CHSQTR(4)   : DFI(FIRST POINT ON TRACK)
         CHSQTR(5)   : DFI(LAST  POINT ON TRACK)

 2.) EXEMPLE FOR APPLICATION OF MCTRCO:

     1.   SELECT TRACK FROM VECT-BANK

     2.   SEARCH FOR TRACK IN PATR(12)-BANK COMPARING
          TRACK# + TYP# (2. WORD IN TRACK ARRAY OF PATR(12)).
          THIS DOES NOT WORK FOR SOME MC-DATA CALCULATED IN
          TOKYO, BECAUSE WORD 2 OF THE TRACKS IN THE PATR-BANK(12)
          IS NOT SET CORRECTLY (TRACK # IN VECT-BANK = 0).

     3.   CALL MCTRCO: TRACK FOUND IF JPTR2 > 0 .AND. CHSQTR(1) < 20.

     4.   'F11PST.JADESR(MKL0TR)' IS AN EXEMPLE FOR MARKING
          PROTON + PION TRACKS FROM LAMBDA -> P, PI

 3.) EFFICIENCY: ABOUT 3% OF THE TRACKS ARE NOT FOUND.
                 THESE ARE IN GENERAL OF LOW MOMENTUM (ABOUT 50 MEV),
                 OR SHORT TRACKS (10 HITS OR LESS)

 4.) COMMENT: THE PROGRAM WILL BE IMPROVED IF BETTER TRACK PARAMETERS
              WILL BE STORED IN PATR-BANK(12) BY THE MC-TRACKIMG PROGRAM.
```

```
JADE Computer Note 57                     This is JADEPR.TEXT(NOTE57)
=========================

P. Steffen, 82/06/29


        How  to  Use  the  Improved  JET  Chamber  Calibration
      ==========================================================

1.)  INCLUDE: KALIBRSC,KLREADSC    (routines are on F11LHO.JADEGS)

2.)  use file
     FT32F001 DD DISP=SHR,DSN=F11PST.JCAL8292

3.)  Comments:
This is a preliminairy version for those who want to  try  the  improved
calibration constants, which have been determined for  the  spring  198200001700
data, starting from run 10269. The beam vertex position for this  period
is constant. With the improved calibration constants one  obtains
               xv = -1.25,         yv = -0.20.
This vertex can be used for a constraint track fit.
The new calibration constants can be tried also for the 1981  data.  But
it is not guarantied that there will be an improvement. Still  the  beam
vertices have to be determined.
Up  to  now  I  obtained  the  following  results  with  these  improved
calibration  constants  for  momentum  the  resolution   of   17.5   GeV
muon-pairs:
old calibration constants                      sigma(p) / p  =  53 %
new calibration constants                      sigma(p) / p  =  35 %
old calibration constants, vertex constraint   sigma(p) / p  =  41 %
new calibration constants, vertex constraint   sigma(p) / p  =  25 %
```

Olsson

JADE Computer Note 57 This is JADEPR.TEXT(NOTE57)
========================

P. Steffen, 82/06/29

How to Use the Improved JET Chamber Calibration

1.) INCLUDE: KALIBRSC,KLREADSC (routines are on F11LHO.JADEGS)

2.) use file

```
FT32F001 DD DISP=SHR,DSN=F11PST.JCAL8292
```

3.) Comments:

This is a preliminairy version for those who want to try the improvec calibration constants, which have been determined for the spring 1982 data, starting from run 10269. The beam vertex position for this perioc is constant. With the improved calibration constants one obtains

xv = -1.25, yv = -0.20.

This vertex can be used for a constraint track fit.

The new calibration constants can be tried also for the 1981 data. But it is not guarantied that there will be an improvement. Still the bear vertices have to be determined.

Up to now I obtained the following results with these improvec calibration constants for momentum the resolution of 17.5 GeV muon-pairs:

old calibration constants	sigma(p) / p = 53 %
new calibration constants	sigma(p) / p = 35 %
old calibration constants, vertex constraint	sigma(p) / p = 41 %
new calibration constants, vertex constraint	sigma(p) / p = 25 %

JADE Computer Note 58 This is JADEPR.TEXT(NOTE58)

P. Steffen, 82/08/12

New Conventions for Calibration Data

With increasing data the calibration files F11LHO.AUPDAT0 and F11LHO.AUPDAT1 have grown larger and larger. Up to now about 240 tracks of disk storage are used. Also the cpu-time used to read and update the calibration constants is no more negligable, esp. if events are not always ordered with increasing run number.

Most of the data on the calibration files are the so called 'spinning block' data for the lead glass arrays and the tagging counters. These data are only used in the REDUC1 data reduction and in special cases of a recalibration of lead glass pulse heights. If one would drop these data from the standard calibration files one would need less than 40 tracks for the calibration files and certainly also less computer time.

Therefore the following changes are made on Wednesday, August, 18th, 17.00:

```
'F11LHO.AUPDAT0' )
                 ) ---> 'F11LHO.AUPDAT1' (without spinning block data)
'F11LHO.AUPDAT1' )                'F11LHO.AUPDAT0' will be obsolete

'F11LHO.BUPDAT0' = old 'F11LHO.AUPDAT0' (incl. spinning block data)
'F11LHO.BUPDAT1' = old 'F11LHO.AUPDAT1' (incl. spinning block data)
'F11LHO.KALWRK0' stays unchanged
```

The subroutine KLREAD will produce a warning printout, if no 'spinning' block constants are present and therefore no recalibration of leadglass data is possible.
The standard lead glass calibration routine LGCALB will be changed such, that the program stops with a printout, if called without available calibration data.

```
JADE Computer Note 58                        This is JADEPR.TEXT(NOTE58)
=========================

P. Steffen, 82/08/12

              New Conventions for Calibration Data
            ========================================

With  increasing  data  the  calibration  files  F11LHO.AUPDAT0  and
F11LHO.AUPDAT1 have grown larger and larger. Up to now about 240 tracks
of disk storage are used. Also the cpu-time used to read and update the
calibration constants is no more negligable,  esp.  if  events  are  not
always ordered with increasing run number.

Most of the data on the calibration files are the  so  called  'spinning
block' data for the lead glass arrays and the  tagging  counters.  These
data are only used in the REDUC1 data reduction and in special cases  of
a recalibration of lead glass pulse heights. If  one  would  drop  these
data from the standard calibration files one would  need  less  than  4000002000
tracks for the calibration files and certainly also less computer  time.

Therefore the following changes are made on
Wednesday, August, 18th, 17.00:

  'F11LHO.AUPDAT0' )
                   ) ---> 'F11LHO.AUPDAT1' (without spinning block data)
  'F11LHO.AUPDAT1' )                 'F11LHO.AUPDAT0' will be obsolete

  'F11LHO.BUPDAT0' = old 'F11LHO.AUPDAT0' (incl. spinning block data)
  'F11LHO.BUPDAT1' = old 'F11LHO.AUPDAT1' (incl. spinning block data)
  'F11LHO.KALWRK0' stays unchanged

The subroutine KLREAD will produce a warning printout, if  no 'spinning'
block constants are present and therefore no recalibration of  leadglass
data is possible.
The standard lead glass calibration routine LGCALB will be changed such,
that the program stops with a  printout,  if  called  without  available
calibration data.
```

JADE Computer Note 59 This is JADEPR.TEXT(NOTE59)

P. Steffen, 82/08/12

Run dependent Event Vertex on Calibration file

The event vertices of the different runs have been determined from Bhabha-events by S. Komamiya. Up to now this has been done for all the 1981 data and for the data from the spring period 1982 up to run 10972.

These run vertices have been included in the calibration files
'F11LHO.AUPDAT1', and
'F11LHO.BUPDAT0' , 'F11LHO.BUPDAT1'.
For the 1979 + 1980 data the old vertices of B. Naroska are used. For the later 1982 data the vertex from spring 1982 is used until the new ones are determined.

The run dependent event vertex can be used in the following way:

```
        Use  %MACRO CALIBR of macro library (F11GOD.PATRECSR)
        ICALIB(10)       :  Pointer = IPVTX
        ACALIB(IPVTX+1)  :  XV
                    +2   :  sigma(XV)     = sigma(bhabha-tracks)
                    +3   :  YV
                    +4   :  sigma(YV)     = sigma(bhabha-tracks)
                    +5   :  ZV           (= 0. at present)
                    +6   :  sigma(ZV)    (= 0. at present)
```

```
C-----------------------------------------------------------------------
C           MACRO CALIBR .... JADE CALIBRATION DATA COMMON
C-----------------------------------------------------------------------
      COMMON/CALIBR/ ACALIB(1000)
                     DIMENSION HCALIB(100),ICALIB(100)
                     EQUIVALENCE(ACALIB(1),HCALIB(1),ICALIB(1))
C----------------------- END OF MACRO CALIBR ------------------------
```

It should be noted that the run dependent vertices have been determined using the new jet chamber constants and that they are dependend on them.

Differences between old and new vertices are

```
                 < 0.4 mm  for spring 81 data
                 < 1.8 mm  for summer 81 data
                 < 0.7 mm  for fall   81 data
                 < 0.3 mm  for spring 82 data
```

In order to obtain improved momentum and space resolution in the r-phi plane it is necessary to fit the tracks again, e.g. by using the subroutine REFITV (see JADE Computer Note 60).

```
JADE Computer Note 60                    This is JADEPR.TEXT(NOTE60)
=========================

P. Steffen, 82/08/12


         Refit Tracks with Run Vertex Constraint
       ===========================================
            SUBROUTINE REFITV(IPTR,IPJHTL,ERRFAC)

This subroutine fits a parabola to the hits of a track in the  PATR-bank
in the x-y-projection. The run vertex from the common CALIBR  (see  JADE
Computer Note 59) is used as an additional constraint.

Arguments:
    IPTR   : pointer to th 0-th word of the track data in the PATR-bank,
             (e.g. idata(iptr+1) = number of the track)
    IPJHTL : BOS-pointer to JHTL-bank
    ERRFAC : factor to increase or decrease the standard  error  of  the
             vertex :
             =   1.: full vertex constraint
             =1000.: essentially no vertex constraint

Concerning the errors the following assumptions give reasonable results:
    sigma(hits  ) = 0.200 mm
    sigma(vertex) = sqrt( 0.300**2 + sig(mult.scatt.)**2 ) mm

The error of the vertex can be increased by the external factor  ERRFAC:
    ERRFAC =     1. --> full vertex constraint and best chamber
                        calibration is used:
                        momentum resolution (mu-pairs, spring 82)
                        improved by a factor of 2
    ERRFAC = 1000. --> no vertex constraint, but best chamber
                        calibration is used:
                        momentum resolution (mu-pairs, spring 82)
                        improved by a factor of 1.4

The measured hits are subject  to  all  corrections  described  in  JADE
Computer Note 45.

The track data with the results of the  fit  are  stored  in  the  array
WRK(HPTR0+0,1,2,...) / IWRK( ) in the common  /CWORK/  (use  the  macros
CWORKPR and CWORKEQ). The track data in CWORK is a  copy  of  the  track
data in the PATR-bank. Only the fit parameters, start and end point  and
directions are replaced by the new results. The program identifier (2-nd
word of track data) is set to 32 in order to identify tracks fitted with
this program.

Exemple for use of this program:
         Use  %MACRO CWORKPR and %MACRO CWORKEQ
         of macro library (F11GOD.PATRECSR)
C-----------------------------------------------
C  MACRO CWORKPR .... PATTERN RECOGNITION CWORK
C-----------------------------------------------
      COMMON /CWORK/ HPLAST,HPFREE,HPWRK(30),ADWRK(600),
     ,               HPRO,HNTR,HNTCEL(98),IPCL(200),NRHT(200),
     ,               NWR1(200),DS1(200),SL1(200),
     ,               NWR2(200),DS2(200),SL2(200),
     ,               LBL(200),NTREL(200),ICRO(200),
     ,               NTR,HNREL(100),HISTR(9,100),HRES(168),
     ,               NTRLM,RLMTR(3,5),
     ,               WRK(7000)
                     DIMENSION TRKAR(200,11),ITRKAR(200,11),
     ,                         LMRTR(3,5)
                     EQUIVALENCE (IPCL(1),TRKAR(1,1),ITRKAR(1,1))
                     EQUIVALENCE (LMRTR(1,1),RLMTR(1,1))
         DIMENSION IWRK(7000),HWRK(14000),IDWRK(600),HDWRK(1200)
                     EQUIVALENCE (IWRK(1),WRK(1),HWRK(1))
                     EQUIVALENCE (IDWRK(1),ADWRK(1),HDWRK(1))
C---------- END OF MACRO CWORKPR --------------
C------------------------------------------------------
```



```
C  MACRO CWORKEQ .... PATTERN RECOGNITION CWORK POINTERS
C--------------------------------------------------------
        EQUIVALENCE
C                   POINTERS FOR FXYZ HIT ARRAY .. PRIMARY L/R SOLUTION
     #             (HPHT0,HPWRK( 1)),(HPHT9,HPWRK( 2)),(HLDHT,HPWRK( 3))
C                   POINTERS FOR CWORK SINGLE TRACK PATR BANK
     #            ,(HPTR0,HPWRK( 4)),(HPTR9,HPWRK( 5)),(HLDTR,HPWRK( 6))
C                   POINTERS FOR TRACK ELEMENT HIT LABEL ARRAY
     #            ,(HPHL0,HPWRK( 7)),(HPHL9,HPWRK( 8)),(HLDHL,HPWRK( 9))
C                   POINTERS FOR FXYZ HIT ARRAY .. OPPOSITE L/R SOLUTION
     #            ,(HPHT0A,HPWRK(10)),(HPHT9A,HPWRK(11)),(HLDHTA,HPWRK(12))
C                  POINTER LIMIT ON FXYZ HIT ARRAY
     #            ,(HPHTLM,HPWRK(13))
C                  POINTERS FOR
     #            ,(HPTE0,HPWRK(14)),(HPTE9,HPWRK(15)),(HLDTE,HPWRK(16))
C------------- END OF MACRO CWORKEQ ------------------


C         GET RUN VERTEX
C                                                USE MACRO CALIBR
          IPVTX = ICALIB(10)
          XV    = ACALIB(IPVTX+1)
          YV    = ACALIB(IPVTX+3)
C         GET POINTER TO JHTL_BANK
          IPJHTL= IDATA(IBLN('JHTL'))

          loop over all tracks (and select special ones)
C
C            CHECK IF MOMENTUM >100 MEV
             IF(ABS(ADATA(IPTR+25)).GT..00150) GOTO 900
C
C            GET DISTANCE OF TRACK FROM VERTEX
             CALL DRTRCK(IPTR,XV,YV,DR0)
C
C            CHECK IF DISTANCE TO VERTEX IS < 25 MM
             IF(ABS(DR0).LT.25.) GOTO 900
C
C            FIT TRACK WITH VERTEX CONSTRAINT
             CALL REFITV(IPTR,IPJHTL,1.0)
             IP     = HPTR0 - 1
             SIG    =  WRK(IP+23)
             SIG0   = ADATA(IPTR+23)
             ANHT   = IWRK(IP+24)
             IF(LAND(IWRK(IP+ 4),16).NE.0) ANHT = ANHT * 1.5
             ANHT0 = IDATA(IPTR+24)
             IF(SIG0.LT.SIG .AND. SIG.GT..35) GOTO 900
 C
C            REPLACE OLD RESULT IN PATR_BANK
             CALL MVC(IDATA(IPTR+1),0,IWRK(HPTR0),0,192)
C
  900     CONTINUE
```

One can simplify the procedure by just calling FITEVR (see JADE Computer Note 61). This subroutine tries to refit all tracks which originate from a position of <25mm from the run vertex. The new result replaces the old one in the PATR bank.

P. Steffen, 82/08/18

Refit all Tracks of an Event with Run Vertex Constraint

SUBROUTINE FITEVR(NTRVTX,MODE)

This subroutine selects all tracks with more than 16 hits, which originate from the run vertex within a distance of 25mm. For these selected tracks a parabola fit with vertex constraint is performed in the x-y-plane. (A modification of SUBROUTINE REFITV is used; see JADE Computer Note 60.) If the mean deviation of the hits from the fitted curve is <0.35mm or less than the one of the previous fit, the new result is stored in the PATR-bank. In addition (MODE = 2,3) all those tracks, which are assumed to originate from the vertex are then simultaneously fitted in the r-z-plane to different slopes and a common vertex in z.

Arguments:

```
        Input : MODE   = 0 : overwrite old PATR-bank with new results
                MODE   = 1 : create new PATR-bank with new results
                MODE   + 2 : do also common z-fit
        Output: NTRVTX = number of tracks from the run vertex
```

The measured hits are subject to all corrections described in JADE Computer Note 45.

The track data with new fit results in the PATR-bank can be identified by the program identifier (2-nd word of track data). Bit 32 is set if the r-phi coordinates of the track are refitted. Bit 64 is set if the track coordinates also participated in the common r-z-fit.

If a z-r-fit with a common vertex is requested, it is recommended to use the recalibration of z-coordinates by P. Dittmann (CALL ZSFIT with MODE=1) before this program.

JADE - Computer Note No. 62
W. Bartel / L. Becker
12.10.1982

JADE REFORMAT PROGRAM

The JADE experimental data coming from the NORD 10/50 computer are written to raw data tapes (F22YEN.JADE.EXDATA07.GO....VOO)

I. Reform step

The raw data serve as input to the reformatting step, where BOS banks are rearranged and put into proper format.

The REFORM program produces the following output data sets :

i. F11LHO.JDATA08. REFORM.GO....VOO
contains reformatted events

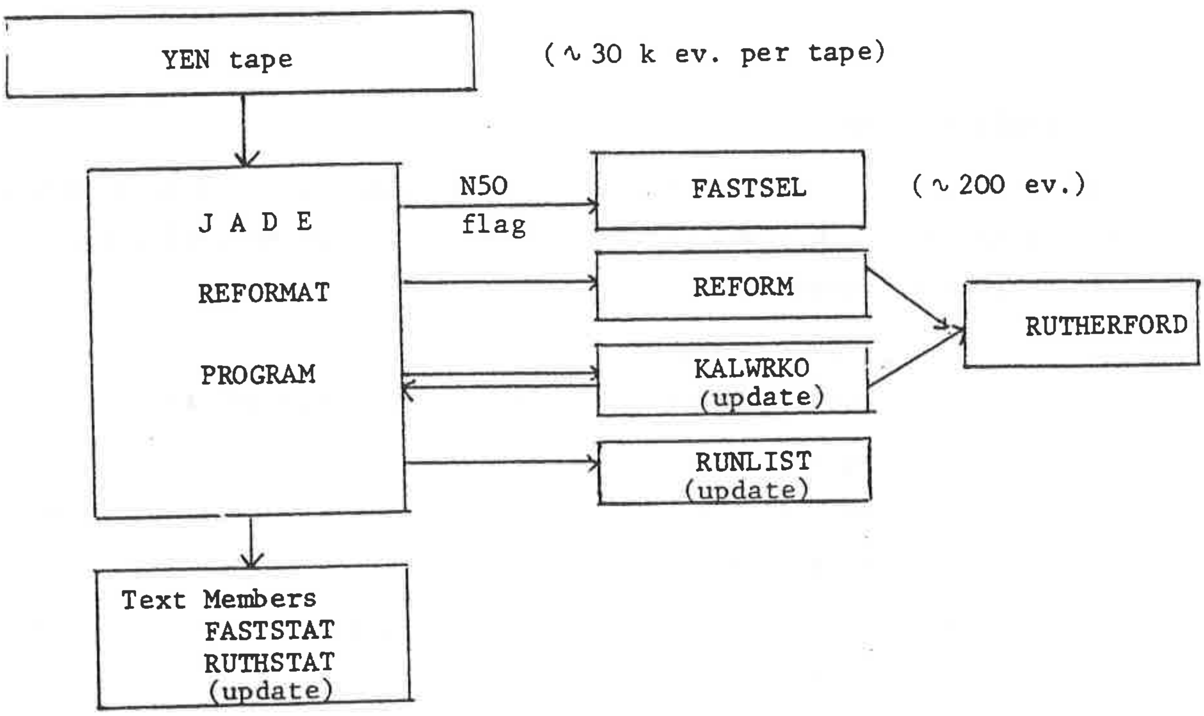
ii. F11LHO.KALWRKO
contains updated calibration file (bad lead glass blocks)

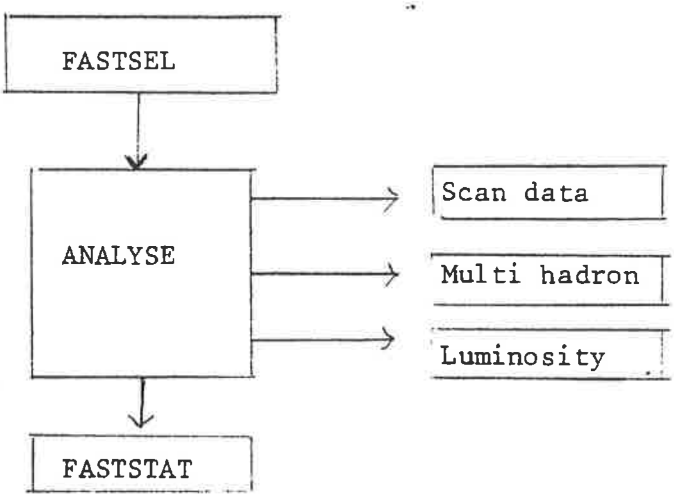
iii. F11LHO.DSKTI001
contains updated runlist

iv. F11BEK.FASTSEL.DS...
contains events flagged by the N50

v. Rutherford tape
contains a copy of REFORM and KALWRKO

vi. The text members FASTSTAT and RUTHSTAT on the library JADEOL.REFORM.S. are updated.





Like for the REFORM-step the analysis program should be submitted by JADE-members on shift via the self-explanatory CLIST #ANALYSE on JADEOL.REFORM.S. The CLIST is started by typing

(# ANALYSE)

into the command field. The user is prompted for the number of the FASTSEL-file, which should be processed. The file number can be inferred from a listing of the text member FASTSTAT, which will appear on the screen. The number of the FASTSEL-file is automatically transferred to the output files SCAN and MUHA so that there is a one to one correspondence of file numbers.

Like in the case of the REFORM step, the CLIST may be aborted without further consequences latest after the 'SUB' command, when the user is asked whether he really wants to submit the analysis job.

Note:

Both CLIST's contain LDS-commands in order to find out whether data sets exist or not. If a data set does not exist the following line will appear on the screen :

```
***     ERROR  ***       DSN                NOT IN CATALOG
```

The output 'ERROR' is an IBM-feature. It is irrelevant in the present context.

L. Becker
G. Heinzelmann
13.5.1983

JADE COMPUTER NOTE NO. 63

z-Recalibration

It is known that the z-calibration based on pulser data is not sufficient to obtain a good z-coordinate. This is essentially due to systematic effects which have been studied by T. Nozaki and P. Dittmann. They found e.g. that the correction for the z-coordinate depends on the φ angle of a track element.

To correct for the systematic effects a method was developed by P. Dittmann described in a note dated Oct. 81. This first version however changes the amplitudes in the jetchamber bank and influences therefore the dE/dx - dedetermination. To avoid this, P. Dittmann afterwards set up a slightly different version which improves the z-coordinate

$$z = \frac{L_{eff}}{2} \quad \frac{A_L - A_R}{A_L + A_R}$$

by correcting the amplitudes in the following way :

$$A_{L,R} \rightarrow A_{L,R} \pm C$$

Now the correction C (which depends on the orientation and position of a track element in drift space) drops out in the sum $A_L + A_R$ used for dE/dx calculation. P. Dittmann started to create new calibration files for 1979 - beginning of 1981. This was now continued up to the end of 1982. The method also determines new pedestals and gaines using a combined set of multihadron and Bhabha events for the first hit and a larger set of multihadron events for the second hit. The constants have been obtained by an iterative least squares fit. The calibration constants are slightly time dependent and were computed for the following run periods :

79A:	0-1400	79B:	1400-2600		
80A:	2600-3730	80B:	3730-4900	80C:	4900-6000
81A:	6000-7600	81B:	7600-10000		
82A:	10000-11800	82B:	11800-13000		

IJHTL = pointer to hit label bank

IPATR = Pointer to PATR bank

MODE = 0 : does everything (z-calibration, hit cleaning, track fit)

= 1 : z calibration only

= 2 : z calibration and track fit (no hit cleaning)

MODE 0,1,2 overwrites the amplitudes in JETC bank

MODE 0,2 overwrites all z correlated information in PATR and JHTL bank (z-bit)

If problems occur, please contact L. Becker or G. Heinzelmann.

FIGURE CAPTIONS

Fig. 1 : Distribution of $\sigma_z = [\Sigma(z_i^{meas} - z_i^{fit})^2/(n-2)]^{1/2}$ before and after z recalibration, refit and hit cleaning.

Fig. 2 : K^0 mass spectrum before and after z recalibraiton, refit and hit cleaning.

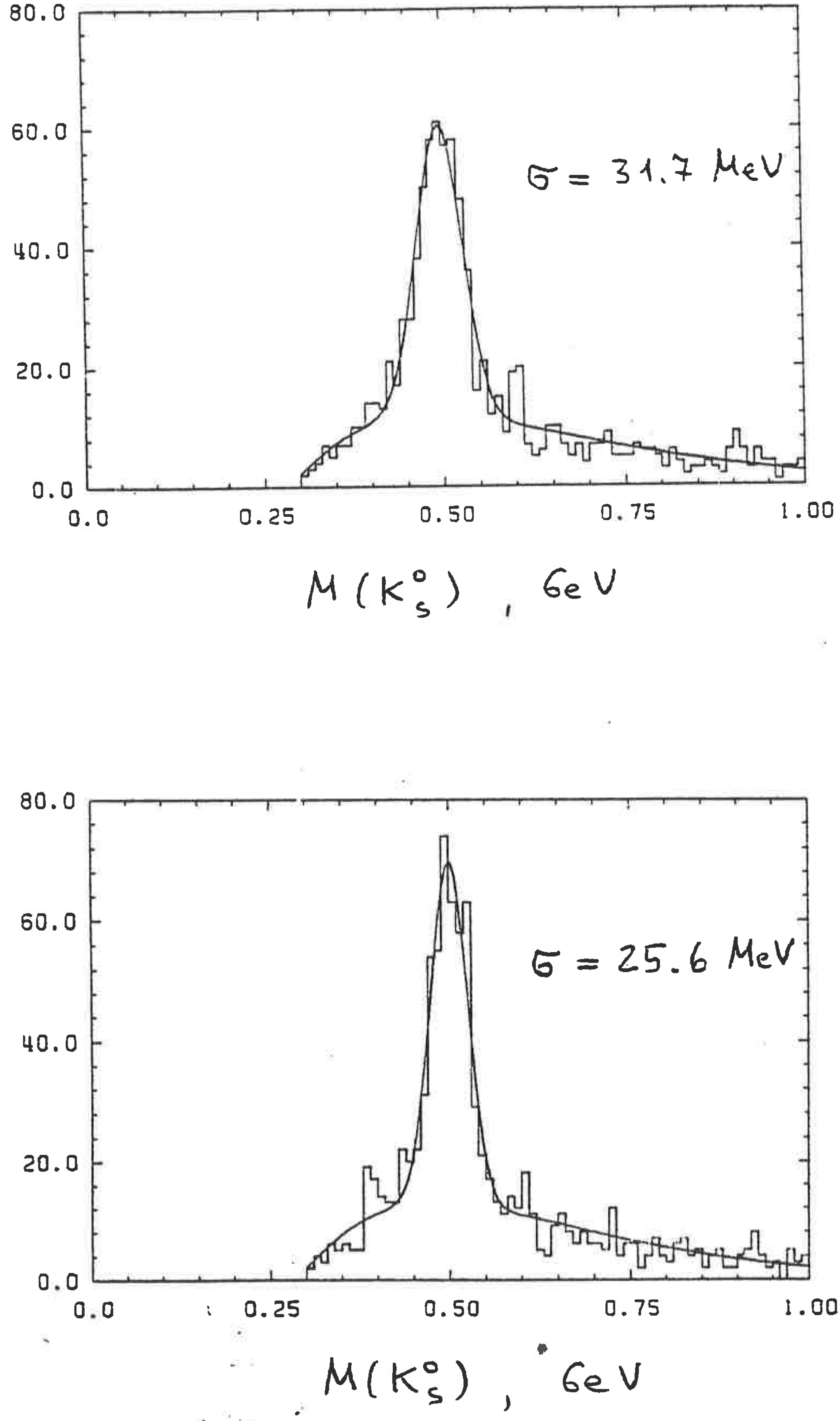

80.0
60.0
40.0
20.0
0.0
0.0
0.25
0.50
0.75
1.00
σ = 31.7 MeV
M(K°s) , GeV
σ = 25.6 MeV
M(K°s) , GeV

Fig. 2

Olsson

JADE COMPUTER NOTE 63 a L.Becker 5.12.83

The z calibration constants are now available from the general calibration files F11LHO.AUPDAT1/BUPDAT0,1. Therefore, the call of ZSREAD is no more necessary.

Further, the calibration status of the 'JETC' bank can be obtained from the second half word in this bank. If the z calibration has been performed it containes the version number (e.g. 8305), else it is zero.

Now, the user has to make one call for each event:

```
CALL  ZSFIT(MODE)
```

The parameter MODE has the following meaning:

- MODE = 0 : z calibration, hit cleaning and track fit
- MODE = 1 : z calibration only
- MODE = 2 : z calibration and track fit (no hit cleanig)
- MODE = 3 : corresponds to MODE = 0 without updating the 'PATR' bank
- MODE = 4 : corresponds to MODE = 2 without updating the 'PATR' bank

```
******************************
 I                             I
I   JADE COMPUTER NOTE NR. 64   I
 I                             I
******************************
```

HELP for JADE (on the computer)

R.G.GLASSER   JULY 27, 1983

I have created the skeleton of a HELP system on the DESY IBM. This is modelled on the successful version in use at PLUTO. The system is available for use but it is currently limited in utilit by the paucity of information contained in the file. Preparation of additional members by either the experts or by anyone who has the information on general programs in use at JADE would definitely improve the accessibility of information for all, especially the newer members of JADE.

To use the system you only need ONCE IN EACH LIBRARY from which you want access to HELP to enter the command:

'f32BOB.HELP(##INSTAL)'

This will link the JADE HELP on behind the system help.

The command

HELP JADE
or   H JADE

gives the list of available HELPs. Putting an H in column 1 of the line containing the item of interest will display the help member. If you know the name under which the item is filed just type:

H name

If there are common names for JADE and NEWLIB, the system always wins. So in such cases one must add a * to the name to allow it to be found. For that reason the help listed below appears as *GUIDE in the list of available helps.

MEMBER #*GUIDE = HELP *GUIDE

GUIDELINES FOR JADE.HELP

1. Mixed mode (upper and lower case) is permitted.
2. Write your documentation with large margins -- the HELP processor truncates the first two columns.
3. For a HELP member which gives the format of a bank, please use the established format. (See the member f32BOB.HELP(#HEAD) for an example.)
4. If you wish people to be able to display your HELP member at their teminals, its name should begin with the character '#'.
5. Brevity is a virtue, clarity is a greater virtue, but

the imperative virtue is accessiblity.

Contact R.G.Glasser (in his absence Steve Wagner) F32 to have your HELP member installed.

a copy of this information can be obtained by submitting the job 'JADEPR.TEXT(JBJCN64)'

Bob Glasser

SUBROUTINE GGFIT(EMAS,PROB)

R.G.Glasser

PROGRAM DESCRIPTION

The subroutine GGFIT was written to do a kinematic fit to the hypothesis that two observed photons were produced from the decay of a particle of fixed mass, e.g. π^0. The main purpose of the fit is to test consistency with the mass hypothesis so there is no calculation of the fitted particle directions. The fit uses the kinematics of massless particles for the photons. For this case the constraint in any Lorenz frame can be written:

$$M^2 = 4\ E_1\ E_2 \sin^2(\delta/2)$$

where δ is the space angle between the two photons and M is the parent mass. The variables for the fit are the two energies and $\sin^2(\delta/2)$. in the laboratory frame. No attempt is made to deal with correlations in the input data.

The program has been tested on a large sample of events. If it fails to find a fit within 30 iterations it reports failure. (Typical events converge in two to six iteraticns.) It is safe to regard such rare failures as though the chi-squared were very large.

After setting the values of the input kinematic variables in the common GGFITC (available in the MACRO 'F32BOB.JGAMS(¢GGFIT)), the program call is simply

CALL GGFIT (EMAS,PROB)

where EMAS is the value of the mass and PROB is the returned value of the probability of chi-squared (1 degree of freedom).

Input variables in GGFITC are:

- XXM(1-3) - e1,e2,a; measured energies and $\sin^2(\delta/2)$
- VAR(1-3) - variance of the measured values (VAR=σ^2)
- NDIM = number of variables allowed in the fit (2 to fit only energies, 3 to also fit the angle) - preset to 3
- LITER = maximum number of iterations - preset to 30

Output variables are: