JADE COMPUTER NOTE 72

MONTE CARLO DATA VALIDATION

C. BOWDERY
27/2/84

ABSTRACT. This note describes the M.C. Input Validation routines, what they check, what output they produce and how one can disable the tests if necessary.

1. Introduction.

It is a well-known law that if something can go wrong, it will. This is especially true for the business of tracking Monte Carlo 4-vector events through the JADE detector. It is all too easy for bugs to creep into 4-vector event datasets and cause a long-running tracking job to crash. Frequently the traceback points to an underflow in some routine that has long since been forgotten. Worse still, the bugs may go undetected for years. In a systematic attempt to trap these bugs before they do any damage, a set of 3 routines has been built into the standard JADE Tracking Program MCJADE. The principal routine has the name MCVALI and is called by BRVECT ("read next 4-vector event") which is itself called by JGETEV in MCJADE. When invalid data are detected in MCVALI, a call is made to MCVERR to print an error message and then the event is rejected. Finally, at the end of the tracking program, a call is made to MCVSUM to print a summary of the input errors detected.

2. The Tests made in MCVALI.

A total of 16 tests are performed on the input data by MCVALI. There are 9 tests of real values and 7 tests of integer values. These are as follows:

$$0 < E_{beam} < 1000\ GeV \qquad \text{(Test 1)}$$
$$10^{-20}\ GeV/c < |p_i| \leq E_{beam} \quad \text{or} \quad p_i = 0 \quad \text{for } i = x, y, z \qquad \text{(Test 2)}$$
$$10^{-20}\ GeV < E \leq 2E_{beam} \qquad \text{(Test 3)}$$
$$|M - M_{true}| < 0.01\ GeV/c^2 \qquad \text{(Test 4)}$$
$$|p^2 + M^2 - E^2| < 2.5 \times 10^{-8} E^2 \qquad \text{(Test 5)}$$
$$10^{-20}\ mm < |vertex_i| < 5000\ mm \quad \text{or} \quad vertex_i = 0 \quad \text{for } i = x, y, z \qquad \text{(Test 6)}$$
$$vertex < 8000\ mm \qquad \text{(Test 7)}$$
$$p \leq E_{beam} \qquad \text{(Test 8)}$$
$$10^{-20}\ GeV/c^2 < M \leq 2E_{beam} \quad \text{or} \quad M = 0 \qquad \text{(Test 9)}$$

$$particle\ code = 1, 2, \ldots, 6 \qquad \text{(Test 11)}$$
$$charge = 0 \text{ or } \pm 1 \qquad \text{(Test 12)}$$
$$event\ number = 0, 1, \ldots, 9999999 \qquad \text{(Test 13)}$$
$$\#\ particles = 1, 2, \ldots, 300 \qquad \text{(Test 14)}$$
$$\#\ charged\ particles = 1, 2, \ldots, 300 \qquad \text{(Test 15)}$$
$$\#\ neutral\ particles = 1, 2, \ldots, 300 \qquad \text{(Test 16)}$$
$$\#\ particles = \#\ charged + \#\ neutral \qquad \text{(Test 17)}$$

THE JADE SUPERVISOR 1/02/84
@
JADE COMPUTER NOTE 73

C. BOWDERY
J. OLSSON

```
*************************************************************************
*                                                                       *
*    The JADE SUPERVISOR Program :    An Introduction for Newcomers     *
*                                                                       *
*************************************************************************
```

A general description of the JADE SUPERVISOR program chain and its detailed working has often been requested, particularly by people who did not partake in the JADE experiment from the very beginning. The following note is an attempt to summarize the general scheme and the most important details.

What is the SUPERVISOR?

The SUPERVISOR is a subroutine (FORTRAN name: SUPERV) that handles most aspects of the standard JADE data analysis for the physicist user. It is the analysis steering routine inside REDUC1 and the graphics program JADEZ and is able to handle all types of JADE events, be they Monte Carlo, partially analysed real data, "fully" analysed TP events or whatever (but not 4-vector events in "CPROD" format !). The SUPERVISOR can also be used to simplify a user's own physics analysis routine by handling the I/O, initialisation and calibration.
@

What is the difference between the SUPERVISOR and the TP Program?

The SUPERVISOR is a more flexible routine than the TP program and has the possibilities to add additional analyses, make cuts and histogram quantities as required. The TP Program however includes analyses that are not done by the SUPERVISOR, can be steered by a small set of control cards and, of course, creates the TP Summary Banks.

How can it be used?

If you simply want to perform or complete the standard JADE analysis for every event on a dataset then life is very easy. To activate the SUPERVISOR you can use the program JDMAIN0, included either in source form by a %MACRO statement or in compiled form by an INCLUDE linkage editor command. Alternatively, you can write your own main program as follows:

```
     CALL SUPERV
     STOP
     END
```

Until voice recognition computers come into common use, we really can't make it any simpler! Additionally you must specify the names of the input and output event datasets (which must be different) and, in the case of real data, the name or names of the JADE calibration files you require . The execution step JCL is shown on page 3.
@

```
;
//*
//*        Default streams: input = 2   , output = 3
//*
//G.FT02F001 DD DISP=SHR,DSN=input.file
//G.FT03F001 DD DISP=NEW,DSN=output.file,SPACE=(TRK,(20,5),RLSE),etc...
//*
//*        Full Calibration Files including Spinning Block Data
//*        Default stream numbers: 21 and 22
//*
```



```
//G.FT21F001 DD DISP=SHR,UNIT=FAST,VOL=SER=STOR05,DSN=F11LHO.BUPDAT0
//G.FT22F001 DD DISP=SHR,UNIT=FAST,VOL=SER=STOR05,DSN=F11LHO.BUPDAT1
//*
;
```

If the events have already had the Lead Glass analysis performed, the short calibration file F11LHO.AUPDAT1 could be used instead of BUPDAT0/BUPDAT1, attached to stream 22. Specifying UNIT= and VOL= parameters may speed up the execution of the job incidentally. They are not essential though.

What can I do if I want something more sophisticated?

This is no problem. First of all though, a few words about how the SUPERVISOR works will help in answering the question.
The routine consists of a series of calls to analysis packages interspersed with calls to a routine called USER. The default version of USER simply passes control back to the SUPERVISOR after each call. By including your own version of USER into the SUPERVISOR job, the program can be tailored to your own needs. The easiest way to start this tailoring process is to copy the basic USER routine to your own library and add code to it where necessary. Alternatively ask around to see what is already available and use or develop an existing, more advanced version of USER. Be careful though that you do not pick up an old version which may not work any longer or may have incorrect comments.
@

At what stages of the analysis are the calls to USER?

The SUPERVISOR execution goes through a series of LEVELS. At each level a particular analysis is performed provided that it has not already been done. That is, if the appropriate result bank(s) already exist then nothing is done. At the end of execution of the code at a given LEVEL, there is a call to USER. The LEVELS are as follows:

LEVEL	Action Performed/Checked BEFORE Call To USER
0	SUPERVISOR initialised at program start
1	New event read in (USER only called if this is a new RUN; otherwise 1st USER call at Level 2)
2	Nothing further has been done
3	Lead Glass energies have been computed
4	Jet Chamber calibrated and Fast z vertex found
5	Jet Chamber Pattern Recognition done + MC Traceback
6	Lead Glass Shower Analysis done
7	Tracks and LG Clusters joined.
8	Nothing further done - empty level at present
9	Muon Chamber analysis done
10	Nothing further done - empty level at present
100	Last event has been written out (On completion of this level, program stops.)

The SUPERVISOR keeps track of the current execution level with a counter called INDEX which is passed to USER as the only argument. Normally, USER performs some operation at a given LEVEL and then increments INDEX
@
before returning control to the SUPERVISOR in order to continue the analysis. Two other levels also exist in the SUPERVISOR but a call to USER does NOT occur at the completion of these. They are:

LEVEL 11	The current event is written out and execution continues at LEVEL 1 or LEVEL 2.
LEVEL 12	Immediate termination of the analysis, statistics are printed out and execution continues at LEVEL 100

Thus LEVEL 11 can be reached simply by incrementing INDEX at the end of LEVEL 10 in USER or by setting INDEX = 11 at any LEVEL. This has the effect of terminating the analysis chain for this event, writing out the

```
event and starting the next. LEVEL 12 can only be reached  by  assigning
INDEX = 12 in the USER. This is the only valid way  of  terminating  the
SUPERVISOR program. Do not set INDEX = 100 as this  will  result  in  an
error message and loss of SUPERVISOR statistics.
----> Setting INDEX = 1 at any LEVEL in USER will result in the  current
      event being dropped and the analysis started on the next event.

What sort of operations can the USER be programmed to do?
---------------------------------------------------------

   One way the USER can influence the SUPERVISOR is by changing some  or
all the default values used during program execution, for  example,  the
input, output and calibration logical unit numbers\ or  the  destination
of the program printout. These operations need only be done usually once
per program run. Thus the necessary code could be placed in USER at  the
point set aside for servicing the  initialisation  call  (LEVEL  0  i.e.
after the SUPERVISOR has initialised itself). This level should be used
for booking histograms, reading  in  special  options,  printing  banner
headlines etc., etc.
@  Secondly, by manipulating the  INDEX  variable  that  the  SUPERVISOR
uses, certain analysis calls can be skipped or even done out of sequence
although the latter is not always feasible or desirable because  of  the
interdependence of some of the analysis steps.  Additionally,  decisions
about proceeding with the current event or rejecting it or stopping  the
analysis altogether can be made at any level and carried out by  setting
INDEX to one of the special values provided for these purposes.  A  word
of warning is necessary here. Care must be taken when changing INDEX  to
ensure that infinite loops do not occur or similar disasters.

   Thirdly, histograms  can  be  filled  at  various  levels  after  the
analysis  packages have  been called.  A call to  HISTDO at LEVEL 100 in
USER is an ideal way of printing the histograms at the end.

   Fourthly, by deleting results banks, a re-analysis of a detector part
can be performed. For example, by deleting the PATR and  JHTL  banks  at
the call to USER at the end of LEVEL 4 (or earlier), the PATREC  package
will create a new one with the latest constants at LEVEL 5. By  deleting
the muon results bank MUR1/0 and/or MUR2/0 before LEVEL 9,  a  new  muon
analysis will take place.

Where can I find the SUPERVISOR and the analysis packages it calls?
-------------------------------------------------------------------

   The  SUPERVISOR,  default  USER,  JDMAIN0  and  many  important  JADE
routines (in source/compiled form) reside on F11LHO.JADEGS/JADEGL. Other
routines can be found as follows:

   Lead Glass analysis        .......    JADELG.SOURCE/LOAD
   I.D. Pattern recognition   .......    F11GOD.PATRECSR/PATRECLD
   Muon analysis              .......    F22ALL.JADEMUS/JADEMUL
   BOS routines               .......    F1EBLO.BOSLIB.S/BOSLIB.L
@
Appendix A          SUPERVISOR Outline for the Inquisitive
----------          --------------------------------------

   The following is an outline of SUPERVISOR actions, in the order  they
occur between the various calls to USER.  Note  that  all  arguments  in
subroutine calls have been omitted for clarity. Further details  can  be
obtained from the source listings.

The JADE standard BLOCKDATA, with most of the geometrical constants,  is
linked via the statement:
                        EXTERNAL JADEBD

BOS initialisation (currently with COMMON / BCS / IDATA(40000) ) :

                        CALL BINT
                        CALL BWRO

Initialization of various analysis routines. Note that VTXINI is called,
although the vertex finding programs are not yet called in SUPERV.
```



```
                        CALL LGINIT        (Lead glass analysis)
                        CALL INITZV        (Z-vertex finding)
                        CALL MUINI         (Muon analysis)
                        CALL VTXINI        (Vertex finding routines)

                            ---->>  CALL USER(0)

Start event loop; first the remaining CPU-time is checked with the
function
                        IUHR(ISECLF)
@
ISECLF is found in COMMON /CSECLF/ ISECLF  and is BLOCKDATA set to 2. It
could be changed at level 0.

Read event:
                        CALL EVREAD

Note that EVREAD automatically handles the special constants records  in
the beginning of every MC data file (see JCN 66). After reading, a  copy
of the bank HEAD is stored in COMMON /CHEADR/ HEAD(108). Note here  that
the first word of the bank is found in HEAD(9). This is  an  Historische
Eigentuemlichkeit of JADE. Thus the Run and Event numbers are  found  in
HEAD(18) and HEAD(19) respectively. Note also the standard JADE  conven-
tion, with IMPLICIT INTEGER*2 (H).

IF the read event is the FIRST event of a NEW Run:

                        CALL KALIBR        (Calibration constants)
                        CALL INPATC        (Init. Pattern Recognition)
         and for the very first event:
                        CALL INPATR        (Init. Pattern Recognition)

Thus initialisation of PATREC takes place after  the  calibration  files
have been read. Note that INPATR is called only once,  while  INPATC  is
called at the beginning of each new run, to take account of e.g. changes
in Lorentz angle, etc..
                            ---->>  CALL USER(1)  (new run only)

For ALL events, before any analysis has been done/checked:

                            ---->>  CALL USER(2)
@
Lead glass calibration, if the bank ALGN does not yet exist:

                        CALL LGCALB
                            ---->>  CALL USER(3)

Jet chamber data calibration, if the calibrated JETC bank does not yet
exist; immediately followed by the fast Z-vertex finding (bank ZVTX):

                        CALL JETCAL
                        CALL ZVERTF
                            ---->>  CALL USER(4)

Pattern recognition, if no bank PATR exists or if only bank PATR 12 is
present (MC events). PATREC result banks PATR and JHTL are created. The
Monte Carlo backtrace facility is also prepared here, with the creation
of the bank TR4V (JCN 69):

                        CALL PATREC
                        IF( M.C. ) CALL MCTR4V
                            ---->>  CALL USER(5)

Lead glass cluster finding. The LGCL bank is created:

                        CALL LGANAL
                            ---->>  CALL USER(6)

Charged track - LG cluster connections and photon energy corrections;
The bank LGCL is partly overwritten with new results, see JCN 14C.
```

```
                        CALL LGCDIR
                            ---->>  CALL USER(7)
@
At level 8 nothing happens at present, free for later use:

                            ---->>  CALL USER(8)

Muon analysis. MUR1 and MUR2 banks created.

                        CALL MUANA
                            ---->>  CALL USER(9)

At level 10 nothing happens at present, free for later use:

                            ---->>  CALL USER(10)

At the end of the event loop, the event will be written out, unless  the
INDEX variable has been set to the value 1, with  subsequent  new  event
reading:
                        CALL EVWRIT

At the end of the job (EOF encountered, TIME LIMIT or USER forced),  the
last part of an event is written  out,  statistics  over  accessed  USER
levels as well as event statistics are printed  and  in  addition,  muon
statistics (if available) are also printed:

                        CALL BMLT
                        CALL BWRITE
                        CALL MUFINI
                            ---->>  CALL USER(100)

Finally a RETURN to the MAIN program is done.


    Type:  SUB  'JADEPR.TEXT(JADECN73)'    to get a copy of this note.
/*
```

```
74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74

==== UPDATE ==== THIS NOTE REPLACES OLD JADE C.N. 74=================

    JADE COMPUTER NOTE NUMBER 74
==============================================

   SUBJECT: Analysis Routines for Tagging System
   =============================================

                                             Author:A.J.Finch
                                             ================

THIS NOTE CAN BE FOUND IN 'F11LHO.TAGG.S(TAGNOTE2)'
```


Summary:
========

This note desribes briefly the analysis routines installed recently on F11LHO.TAGG.S and F11LHO.TAGG.L whose purpose is to produce the output banks 'ACLS', 'TAGG' /0,/1,/2 whose content is described in Jade Computer Note No.16.The input to these routines is the 'ATAG' bank which contains the raw adc contents from the tagging adcs.They also utilise a new set of calibration constants installed on the standard Jade calibration file,and obtained in the usual way by calling KALIBR for every run.The programs were originally written by H.Wriedt ,A.Finch, and J.Nye.

Introduction:
=============

One part of JADE that has undergone slightly more changes than most is the tagging system . In 1979 and 1980 there was the Mark 1 . For 1981 with the arrival of mini - betas Mark 2 was installed suspended on the muon chambers. In order to solve the problems of browning lead glass encountered with this location, Mark 3 was installed for 1983 running onwards . All these systems require different software to analyse them due to their differing geometries etc.

The Software:
=============

The minimum information a user needs to use the package that runs these analysis programs is:

```
 1) Have F11LHO.TAGG.L in his list of load libraries.
 2) CALL KALIBR for every run.
 3) CALL TAGAN once per event.

 4) For 1979/80 data only :
      He must also have the private calibration file
      'F22HOW.PEDESTAL.ALLSP80' attached to fortran
       stream 19.
```

The routine TAGAN


```
=================

     TAGAN(IERTAG,NRUN)

    Arguments:
    ==========

        IERTAG - OUTPUT RETURN CODE

        NRUN   - INPUT DUMMY RUN NUMBER TO OVERIDE CONTROL  OF  WHICH
                 ANALYSIS ROUTINE IS  CALLED  FOR  MONTE  CARLO
                 DATA
                +++++++++++++++++++++++++++++++++++++++++
                + IN 99% OF CASES THIS CAN AND SHOULD BE +
                + SET TO ZERO                            +
                +++++++++++++++++++++++++++++++++++++++++

    Description:
    ============

     This  routine  controls the  analysis  routines.   Its first job
     is to  check that the  input event can be  analysed.  Classes of
     events that can not be analysed are :
         a) Pedestal events.
         b) Events with no 'HEAD' bank.
         c) Events with no 'ATAG' bank.
     The  routine must then  decide which of the  three possible sets
     of routines to call (one for each version of the tagging sytem).
     It does this by  using the  information in the  HEAD  bank.  For
     real data it simply uses the run number. For Monte Carlo  events
     it is neccesary to tell the program which Mark of tagging system
     to expect this is done by one of two methods:-
          On encountering  Monte Carlo  data, the routine looks at
          the  input argument  'nrun' . If this number is not zero
          it takes,  and uses,  it as the  run number,  so -

  Use of NRUN:
  ============
              NRUN                ASSUMED SIMULATION
              ====                ==================

              <6000                Mark 1
         6000>    <12947           Mark 2
              >12948               Mark 3

        If nrun is set to zero there is a second line of attack which
        is to look at the 2nd half word of the bank 'ATAG'. The value
        of this determines which simulation  was  done  according  to
        following scheme.

                  Value of word           Simulation
                  =============           ==========

                        0                  Mark 1    (or real data !)
                        1                  Mark 2
                        2                  Mark 3

       (Provided this scheme has been adopted in the simulation then
         NRUN can be set to zero)

        N.B. Tagan issues a messsage the first time it is called saying
             what it thinks it is analysing, and again if it encounters
             a change from one tagger type to another.
```

```
     Meaning of return code 'IERTAG':
     ================================

                   Value of IERTAG             Meaning
                   ===============             =======

                         0             No Problems - and clusters found.

                         1            No Problems - but  no  clusters
                                      found ('ATAG',and 'HEAD' exist)

                         2             Analysis was completed but
                                        at least one of the output
                                       banks could not be created due
                                       to lack of space.

                         10           No 'ATAG' banks - no analysis done

                         11           No 'HEAD' bank - no analysis done


Stucture of the main anlysis routines:
======================================

      The routines that TAGAN calls are-

      ANATAG  for 1979/80 data
      TAGGTP  for 1981 onwards data (with a flag to tell whether data
                                     is from 1981/2 or 1983 onwards)

      The structure of TAGGTP is briefly described at the end of  this
      note for anyone who is interested, for more detail refer to  the
      commented version on F11LHO.TAGG.S.


Using analysis routines for other purposes:
===========================================

 Users may wish to use a small sample of the  routines for their own
 purposes. E.g. for fast selection routines. It is possible for 1981
 data onwards to use the following sheme:

          DATA THRESH/6000.0/

          CALL TAGSET(IMARK) - Force analysis to assume  IMARK  tagger,
                               IMARK = 2 or 3.  -  optional  for  monte
                               carlo data if simulation didn't set  flag
                               in 'ATAG

          CALL TAGINT(&100)  - Initialisation - to be called once  per
                              event ; RETURN 1 if can't work  out  which
                              tagger this is ( due to no head bank).

          CALL TAGADC(IWRITE,&100)     - Gets the ATAG data,if  IWRITE=
                                       1 writes out some debugging info
                                       Applies  nominal calibration to
                                       convert channel number to MeV.

          CALL TAGPED   - pedestal fixing - optional
                                    (no disaster if not done)
```



```
          CALL TAGKAL(IWRITE)     -   Calibration - optional
                                       (no disaster if not done)

          CALL TAGSUM(-1,SUMM,&100) - work out sum of -z and + z
          CALL TAGSUM(+1,SUMP,&100) - return 1 if sum has 'impossible'
                                                  value
C
C SUMM,SUMP,THRESH are in MeV
C
          IF((SUMM.GT.THRESH).OR.(SUMP.GT.THRESH))....
 100      CONTINUE

 | For more detailed information about these and other routines
 | in this package see 'F11LHO.TAGG.S(#TAGDOC)'

 ________________________________________________________________


 For completeness there now follows a brief description of
 the procedure adopted in the routine 'TAGGTP' for analysing
 all data from 1981 onwards.

---------------------------------------------------------------------
```

PROCESS	ROUTINE NAME (if not done in TAGGTP)	NOTES
Initialisation	TAGINT	
Read data in 'ATAG'	TAGADC	1) An overall calibration that converts adc channel number to MeV is applied. 2) Software addresses are used from here on.
Subtract pedestals	TAGPED	These are caused by fluctuating pedestals due to 50HZ AC pickup on signal cables.It is only treated in those events where it exceeds the cut off at 20 channels applied by the Le Croy ADC controler.Ammount to be subtracted is estimated on an event by event basis.
Apply calibration factors.	TAGKAL	Factors obtained from Kalibr.
Work out the sum of energy in -Z and +Z tagger.	TAGSUM(JPART,SUM)	Works out SUM for JPART end (JPART = +/- 1)
LOOP1 <This section once for -Z then once for +Z>>>>>>>>>>>>>>>		
Sort adcs into order of decreasing energy contents.	TAGSR1	

```
  Use sorted list to| TAGCLS           |
  find clusters of  |                  |
  deposited energy. |                  |
    Store results in|                  |
   cluster map.     |                  |
                    |                  |
========================================================================
 PROCESS            | ROUTINE NAME     |   NOTES
                    | ( if not done    |
                    |  in TAGGTP)      |
=========================================================================
                    |                  |
 LOOP2 <start loop over all clusters (this end) >>>
                    |                  |
 Fill the cluster   |                  |
 map 'ACLS' + save  |                  |
 pointers for TAGG1 |                  |
                    |                  |
 Find position of   | TAGPOS           | This is the main routine where
   centre of each   |                  | geometrical differences effect
     cluster,in face|                  | the software.
  of blocks.        |                  |   The procedure adopted compares
                    |                  |   the ratio of energies in the
                    |                  |   hit block and its neighbours
                    |                  |   to known distributions of
                    |                  |   energy within e/m showers
                    |                  |   to estimate how far to move
                    |                  |   the centre of the shower from
                    |                  |   the centre of the block with
                    |                  |   the largest ammount of energy,
                    |                  |   towards its neighbours with the
                    |                  |   next largest ammounts of energy
                    |                  |   N.B. A large fraction of hits
                    |                  |   lose some significant fraction
                    |                  |   of their energy out of the
                    |                  |  inner or outer edges,which makes
                    |                  |   position determination
                    |                  |   difficult.
   Save information |                  |   (Hardware addresses are used
   on individual    |                  |   in the output)
   clusters to put  |                  |
   in TAGG/2        |                  |
                    |                  |
 <end of LOOP2 over each cluster>
                    |                  |
 repeat LOOP1 for +Z|                  |
                    |                  |
Calculate angle     |                  |
 between clusters   |                  |
 found above.Flag   |                  |
 colinear pairs of  |                  |
  clusters in TAGG2 |                  |
                    |                  |
 Create output      | TAGSTO           |
  banks             |                  |
                    |                  |
 Return to TAGAN    |                  |
=========================================================================
```

74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74

==== UPDATE ==== THIS NOTE REPLACES OLD JADE C.N. 74====================

JADE COMPUTER NOTE NUMBER 74

SUBJECT: Analysis Routines for Tagging System

Author:A.J.Finch

THIS NOTE CAN BE FOUND IN 'F11LHO.TAGG.S(TAGNOTE2)'

Summary:

This note desribes briefly the analysis routines installed recently on F11LHO.TAGG.S and F11LHO.TAGG.L whose purpose is to produce the output banks 'ACLS', 'TAGG' /0,/1,/2 whose content is described in Jade Computer Note No.16.The input to these routines is the 'ATAG' bank which contains the raw adc contents from the tagging adcs.They also utilise a new set of calibration constants installed on the standard Jade calibration file,and obtained in the usual way by calling KALIBR for every run.The programs were originally written by H.Wriedt ,A.Finch, and J.Nye.

Introduction:

One part of JADE that has undergone slightly more changes than most is the tagging system . In 1979 and 1980 there was the Mark 1 . For 1981 with the arrival of mini - betas Mark 2 was installed suspended on the muon chambers. In order to solve the problems of browning lead glass encountered with this location, Mark 3 was installed for 1983 running onwards . All these systems require different software to analyse them due to their differing geometries etc.

The Software:

The minimum information a user needs to use the package that runs these analysis programs is:

1) Have F11LHO.TAGG.L in his list of load libraries.
2) CALL KALIBR for every run.
3) CALL TAGAN once per event.

2	Mark 3

(Provided this scheme has been adopted in the simulation then NRUN can be set to zero)

N.B. Tagan issues a messsage the first time it is called saying what it thinks it is analysing, and again if it encounters a change from one tagger type to another.

Meaning of return code 'IERTAG':

Value of IERTAG	Meaning
0	No Problems - and clusters found.
1	No Problems - but no clusters found ('ATAG',and 'HEAD' exist)
2	Analysis was completed but at least one of the output banks could not be created due to lack of space.
10	No 'ATAG' banks - no analysis done
11	No 'HEAD' bank - no analysis done

Stucture of the main anlysis routines:

The routines that TAGAN calls are-

ANATAG for 1979/80 data
TAGGTP for 1981 onwards data (with a flag to tell whether data is from 1981/2 or 1983 onwards)

The structure of TAGGTP is briefly described at the end of this note for anyone who is interested, for more detail refer to the commented version on F11LHO.TAGG.S.

Using analysis routines for other purposes:

Users may wish to use a small sample of the routines for their own purposes. E.g. for fast selection routines. It is possible for 1981 data onwards to use the following sheme:

```
DATA THRESH/6000.0/

CALL TAGSET(IMARK) - Force analysis to assume  IMARK  tagger,
```

		channels applied by the Le Croy ADC controler.Ammount to be subtracted is estimated on an event by event basis.
Apply calibration factors.	TAGKAL	Factors obtained from Kalibr.
Work out the sum of energy in -Z and +Z tagger.	TAGSUM(JPART,SUM)	Works out SUM for JPART end (JPART = +/- 1)
LOOP1 <This section once for -Z then once for +Z>>>>>>>>>>>>>>>>		
Sort adcs into order of decreasing energy contents.	TAGSR1	
Use sorted list to find clusters of deposited energy. Store results in cluster map.	TAGCLS	

PROCESS	ROUTINE NAME (if not done in TAGGTP)	NOTES
LOOP2 <start loop over all clusters (this end) >>>		
Fill the cluster map 'ACLS' + save pointers for TAGG1		
Find position of centre of each cluster,in face of blocks.	TAGPOS	This is the main routine where geometrical differences effect the software. The procedure adopted compares the ratio of energies in the hit block and its neighbours to known distributions of energy within e/m showers to estimate how far to move the centre of the shower from the centre of the block with the largest ammount of energy, towards its neighbours with the next largest ammounts of energy N.B. A large fraction of hits lose some significant fraction of their energy out of the inner or outer edges,which makes position determination difficult. (Hardware addresses are used in the output)
Save information on individual clusters to put in TAGG/2		
<end of LOOP2 over each cluster>		

Software Information Note 8/2/84

A new version of the JADE Monte Carlo Tracking Program MCJADE has now been released. It features the Meier Lead Glass Shower Simulation (optional), K_L^0 and neutron tracking and improved multiple scattering of charged tracks.

The routines can be found on:

F22ELS:JMC.S/L

The old routines still exist on:

F22ELS.JMC.OLD.S/L

Standard JCL members exist as follows:

#MCJADE	standard tracking without muon tracking
#PRODUCT	as #MCJADE but with tape copy step
#PRODMU	as #PRODUCT but includes muon tracking

JADE Computer Note 75

16.05.84 Michael Kuhlen

ROUTINES FOR CONVERTED PHOTONS

The vertex routines written by Peter Dittmann (see JADE Computer Note 32) have been improved, and some new features were added. The new version is now available to the public. The whole package can be found on the libraries

```
        'F11KUH.CONVERT.S'   ( Source )
    and 'F11KUH.CONVERT.L'   ( Load   ).
```

Details concerning the vertex algorithm, cuts, fits, efficiency, resolution etc. can be found in: M. Kuhlen, Nachweis konvertierter Photonen aus der e^+e^- - Vernichtung im JADE - Detektor, Diplomarbeit, Hamburg 1984.

Usage

By submitting the job

```
        #COGAM     ( JCL )
```

with the MACROs

```
        USCOGAM    ( user routine )
        BLGEO      ( BLOCK DATA, parameters for vertex search ),
```

the user creates a "PHOT" - bank containing all the relevant information (description below) for each recognized photon conversion, which will be added to the other BOS - banks on the output file.

The program

1. The program consists of several steps:

   a. Pattern recognition for hits, which have not yet been assigned to tracks, resulting in some new, mainly low momentum ($p < 250$ MeV) tracks.

   b. Conversion search

The "PHOT" - bank

(I is the pointer to the "PHOT" - bank)

ADATA(I + 1) = PX	Photon 4 - vector
ADATA(I + 2) = PY	
ADATA(I + 3) = PZ	
ADATA(I + 4) = EG	
ADATA(I + 5) = XM	Invariant mass before photon fit
ADATA(I + 6) = EXM	Error of XM
ADATA(I + 7) = DXY	Track distance at conversion place in x-y
ADATA(I + 8) = SDXY	DXY in standard deviations
ADATA(I + 9) = DZ	Track distance at conversion place in r-z
ADATA(I + 10) = SDZ	DZ in standard deviations
ADATA(I + 11) = DPHI	Opening angle in x-y at conversion place
ADATA(I + 12) = SDPHI	DPHI in standard deviations
ADATA(I + 13) = DTH	Opening angle in r-z at conversion place
ADATA(I + 14) = SDTH	DTH in standard deviations
ADATA(I + 15) = CAPV	cos(APV)
ADATA(I + 16) = APV	φ - angle between photon- and vertex - direction
ADATA(I + 17) = VX	Vertex coordinates
ADATA(I + 18) = VY	
ADATA(I + 19) = VZ	
ADATA(I + 20) = RV	Radius of conversion vertex
ADATA(I + 21) = THV	Angle between vertex - direction and x-y - plane
ADATA(I + 22) = SR	Radial distance vertex - beam pipe (st. dev.)
ADATA(I + 23) = SIGZ	Error of DZ
IDATA(I + 24) = K1	Numbers of electron tracks in "PATR" bank
IDATA(I + 25) = K2	
IDATA(I + 26) = IFIT	Flag: 0 = no photon fit, 1 = photon fit
ADATA(I + 27) = CHI2	χ^2 for photon fit
IDATA(I + 28) = IUNI	Flag for track assignment to vertex: 4 = unambiguous 8 = ambiguous
IDATA(I + 29) = ITK	Flag for conversion place: 1 = tank wall 2 = beam pipe 3 = drift chamber
ADATA(I + 30) = 0.0	

JADE Computer Note No. 76
11.6.84
Karlheinz Meier

A Collection of Programs Used in the Analysis of Inclusive Photon Production

Introduction

This note describes a few routines which have been used for the analysis of LG-energy depositions and might be of some interest for other users as well. It should, however, be stressed that the special purpose of the analysis was the study of -mass spectra in multihadronic events. The programs are not general tools for any kind of photon analysis in JADE.

The aim here is to describe only the technical properties of the programs. Information about the (physical) background can be obtained from the DESY Internal Report F11/01 which will be available in July.

Any questions, suggestions or complaints should be directed to IBM-userid F11MEI or to Karlheinz Meier, CERN, EP-Division.

Source Files and compiled versions of the described routines can be found on

Source : F11MEI. SHOWS
Load : F11MEI. SHOWL

The Program JBPROD

JBPROD is a complete job ready for submission. It creates an output file containing a new bank named 'GAMR' with all photons used in the inclusive $\gamma(\gamma\gamma)$ - analysis. The following steps are being performed :

- recalibrate LG
- rerun modified cluster-analysis
- connect charged tracks with clusters
- analyse shape of neutral showers
- create output file with 'GAMR' - bank

12	X^2 from comparison with single shower
13	pointer to corresponding LGLC-cluster
14	not used
15	single shower fit
16	fitted impact point (1. coordinate)
17	fitted impact point (2. coordinate)
18	σ (1. coord.)
19	σ (2. coord.)
20	dx)
21	dy) from fit
22	dz)
23	X^2 from fit to single shower
24	not used
25	double shower fit
26	fitted impact point (1. coord., 1. photon)
27	fitted impact point (2. coord., 1. photon)
28	fitted impact point (1. coord., 2. photon)
29	fitted impact point (2. coord., 2. photon)
30	σ(1. coord., 1. photon
31	σ(2. coord., 1. photon)
32	σ(1. coord., 2. photon)
33	σ(2. coord., 2. photon)
40	Energy-ratio between the two photons from fit
41	(Energy-ratio)
42	Mass of double strucutre
43	X^2 from double shower fit
44	not used
45	not used

The Subroutine GEGAMM

Unpacking the information from the 'GAMR'-bank can (for example) be done with a routine like

GEGAMM (NGAM, N1, N2)

NGAM is the number of photons found in 'GAMR' (output variable). N1 and N2 are parameters used in a special analysis and without interest here.

The Subroutine SHWFIT

The subroutine

SHWFIT(ICHOIC,VERT,PSSF,CHI22,PDSF,CHI24,ERATIO)

applies a single and/or double shower fit to a measured block topology.

Input variables :

ICHOIC	:	1: single shower fit only 2: double shower fit only 3: single and double shower fit
VERT(3)	:	Vertex (see description SHWCPR)

Output variables :

PSSF (2)	:	Optimized impact coordinates for single photon
DSSF (4)	:	Optimized impact coordinates for both photons in a double shower
CHI22	:	Optimized X^2 for single photon hypothesis
CHI24	:	Optimized X^2 for double photon hypothesis
ERATIO	:	Optimized energy ratio between the two photons in case of a double-shower fit.

As in the case of SHWCPR the measured cluster properties are being transferred via the

COMMON /CLDAT/ .

The fitting is done with the MINUIT algorithm SIMPLEX. Since a slightly modified version is used, the private MINUIT library

F11MEI.MINUIT.LOAD

has to be linked.

A warning : The double shower fit optimizes 5 parameters for a very complicated (z and E-dependent) shower function. It is therefore extremely slow. Only single clusters should be analysed !

The Subroutine EXPECT

The subroutine

EXPECT(IPART,E,VAR,VERT,NBLOCK,IBLIST,BLCFRC,AVGSUM)

is used by the routines SHWCPR and SHWFIT and might also be interesting for other purposes. It calculates the expected energy fraction in any LG-block

JADE Computer Note 77

H. Krehbiel

June 28, 1984

Copying of Source Files and Data Files from the JADE-NORD10 to the IBM and vice-versa.

Preface

The programs that can do such things were written by members of F58 to serve another purpose: the communication between the PADAC TMS9900 and various computers. The features which are useful to us are somewhat a by-product of the F58 efforts and their programs can do many things not described below. Since F58 clings to the principle of zero-documentation and hearsay-broadcast I could open a passable way through the F58 programs only pestering some members of the said group. Nevertheless I acknowledge the willingly given help and support from Messrs. Hochweller and Krechlock, F58; and from Mr. Butenschön, R2.

In the following note I shall first give a description of some programs, (to some length, but restricted as was said above), then give a short command list for quick reference. In the appendices I want to give some hints for the use of NORD data sets on the IBM.

Note: In the listings of interactive operations, the keys to be pressed by the user are underlined.
Lowercase letters in examples need replacement by the user's own codes, descriptors, etc.

1) The Service Program in the NORD10.

```
==============================================================================
=    Warning:  Do not use it when JADE experimental data are transmitted     =
=              to the IBM, i.e. when JDAS takes data with IBM transfer!!     =
==============================================================================
```

The program runs as a real-time program. To start it, log on under RT at the terminal No. 37 next to the Gould-Plotter in the corner of the software-room. (Nowhere else!) Proceed as follows:

```
        ΘFIX 63↓                    (Θ for @,  ↓ for return)
        ΘRT SERVC↓
        ΘLOG↓
```

Follows about a screen full of output: (see next page)

Note: In this program mistyped input can be corrected with ⟨ctrl⟩+A as before, but the response on the screen looks slightly different.

2) The IBM Programs

The msges to the IBM can only be understood (there) if the proper IBM modules are loaded The various modules with which we deal can be distinguished by their response to the empty msge or to HELP↓↓. Those modules are:

A) The R2 Online Monitor. It is activated and other modules are stopped by the m

CHANGE↓↓

Its response after starting and to the empty msge is the line

GIVE MESSAGE

and to HELP↓↓ is

M06 JAD F58SV EXPNORM1 OS 10622622

resp. JADE0

This is information about the on-line connection. The third word is the name of the running on-line job. JADE0 is the JADE online program, F58SV is the F58 service program, whither we proceed as follows:

START F58SV↓↓

B) Thus we enter the On-Line Job Starter Program. Its first response (after possibly some seconds) is

START jobnm REQUEST ACCEPTED.

Subsequent empty msges prompt the response

ONLINE JOB STARTING

until the job is started and th R2 Online Monitor takes over again responding

GIVE MESSAGE

Proceeding LOAD SS↓↓ with the response REQUEST ACCEPTED loads the

C) F58 Service module. Its response to the empty msge is

*** NO COMMAND TEXT ? ***

*** PROCESSING TERMINATED ***

HELP↓↓ brings forth a real list of helpful hints. Any other proper command list (some of them described below) when sent and executed gives

*** SUCCESSFULLY COMPLETED ***

When the user has finished his copying, he should enter

CHANGE↓↓ , and after the GIVE MESSAGE response of the R2 Online Monitor enter START JADE0↓↓ (Jade-Null) to restart the JADE program for data-taking, and only then press Ctrl+Z.

4) Quick Reference

A typical screen image is given with some abbreviations. Conventions as before.

Use terminal 37! and logon under user RT.

⊖FIX 63↓

⊖RT SERVC↓

⊖LOG↓

Enters a full page, ending with:

---> 1 > 1↓

MESSAGE ("CTRL+C" : TAKE LAST COMMANDS):
CHANGE↓↓

This line will be referred to as (m-line) below.

GIVE MESSAGE
(m-line)
START F58SV↓↓

START F58SV REQUEST ACCEPTED

(m-line)

↓ Press ↓ every few seconds. At first the

response is ON-LINE JOB STARTING

(m-line) When the response is "GIVE MESSAGE" in this

line, proceed LOAD SS↓↓

LOAD SS REQUEST ACCEPTED

(m-line) For a test,

press: ↓ The response should be:

*** NO COMMAND TEXT ? ***
*** PROCESSING TERMINATED ***

(m-line)

Now write a good message list as described in ch. 3. When it is sent off with ↓↓ and everything works well, then after some time, during which the terminal input is blocked, the message list is repeated and the line added

*** SUCCESSFULLY COMPLETED ***

Avoid typing errors by all means! They might lead you into mazes or wildly astray.

Appendix A:

Most important differences between NORD10-Fortran and Siemens FORTRAN 77:

	on the NORD	on the IBM
Type declarations:	INTEGER	INTEGER*2
	DOUBLE INTEGER	INTEGER*4
	REAL	REAL*4
	etc.	
Loop control	DO FOR ix =... ENDDO	not allowed. Use standard DO loop
Output statements	OUTPUT list	PRINT *, list
FORMAT characters:		
Integer with leading zeroes:	Jn	In.n
Bare X, one blank	X	1X
Quasi-binary	O (octal)	Z (hexadecimal)
Free real form	not allowed	G

JADE Computer Note 78 Nov. 9th, 1984
K.H. Hellenbrand

Decays of $K^{\pm}$ and KØL in MC

In the Tracking Monte Carlo a new routine TRKADC is implemented, which handles the decays of charged kaons and KØL's. (Remember: decays of KØS's are included in the 4 vector generators.) TRKADC foresees six decay channels for either particle type. All decay channels with more than 1% branching ratio are taken into account according to PDG 1982.

decay No.	decay channel	branching ratio
(1)	$K^0 \to \pi^+ e^- \nu$	19.35%
(2)	$\to \pi^- e^+ \nu$	19.35%
(3)	$\to \pi^+ \mu^- \nu$	13.55%
(4)	$\to \pi^- \mu^+ \nu$	13.55%
(5)	$\to \pi^+ \pi^- \pi^0$	12.6 %
(6)	$\to \pi^+ \pi^-$	21.6 %
		100.0 %
(7)	$K^+ \to \mu^+ \nu$	63.5 %
(8)	$\to \pi^+ \pi^0$	21.2 %
(9)	$\to \pi^+ \pi^+ \pi^-$	5.6 %
(10)	$\to \pi^+ \pi^0 \pi^0$	1.7 %
(11)	$\to \pi^0 \mu^+ \nu$	3.2 %
(12)	$\to \pi^0 e^+ \nu$	4.8 %
		100.0 %

For the antiparticle K^- the charge conjugate final states are taken. The KØL has no antiparticle. Therefore somehow artificially the K^0_{e3} decay and the $K^0_{\mu 3}$ decay are split into two parts to make sure that one finds an equal amount of e^+ and e^- (μ^+ and μ^- respectively) in the final state.

In the k_{e3} decays all terms with m_e^2 are neglected.

Fig. 2 shows the Dalitz plot and the lepton and the pion spectrum for the decay $k^{\pm} \rightarrow \pi^0\mu^{\pm}\nu$.

Performance

General: Only decays inside the inner surface of the lead glass are performed. Beyond the particles are handled by the lead glass routines TRLGL and ENDCLG. Muons, kaons and pions are tracked further by the muon routines which start at the inner surface of the lead glass.

Old scheme for charged kaons: 1) In subroutine TRCDET the routine PIKMUF is called in case of tracking at $k^{\pm}$ or $\pi^{\pm}$. PIKMUF determines the distance STPLEN the particles travel before decaying. STPLEN is exponentially distributed. 2) If the flightpath of a $k^{\pm}$ or $\pi^{\pm}$ becomes bigger than STPLEN the routine PIKDEC (P, PV2, STPLEN) is called with P(1...10) the usual parameters of the mother particle and PV2 (1...10) the equivalent set of parameters for the daughter particle. 3) SVECT1 (PV2, R) is called to store PV2 and R into the 'VECT', 1 bank. R(1...3) is the space point where the decay takes place and hence the starting point of the daughter particle. The old scheme foresees only one decay channel which has only one daughter particle to be tracked further on.

e.g. $\pi^+ \rightarrow \mu^+\nu$ Only the μ^+ is tracked.

$k^+ \rightarrow \pi^+\pi^0$ $(\rightarrow \pi^+\gamma\gamma)$
Only the π^+ is tracked, the two γ's are not.

New scheme for charged kaons: Step 1) is unchanged. 2) For the decay of a kaon the routine TRKADC (P,R) is called. 3) A list of the final state particles is set up in TRKADC. Neutrinos are omitted. If there are π^0's in the final state first the decay $\pi^0 \rightarrow \gamma\gamma$ is made and the two γ's are put into the list. (The decay $\pi^0 \rightarrow \gamma\, e^+e^-$ is neglected.) For all particles in this list SVECT1 (PV2 , R) is called. In this case the γ's from π^0 decays are tracked.

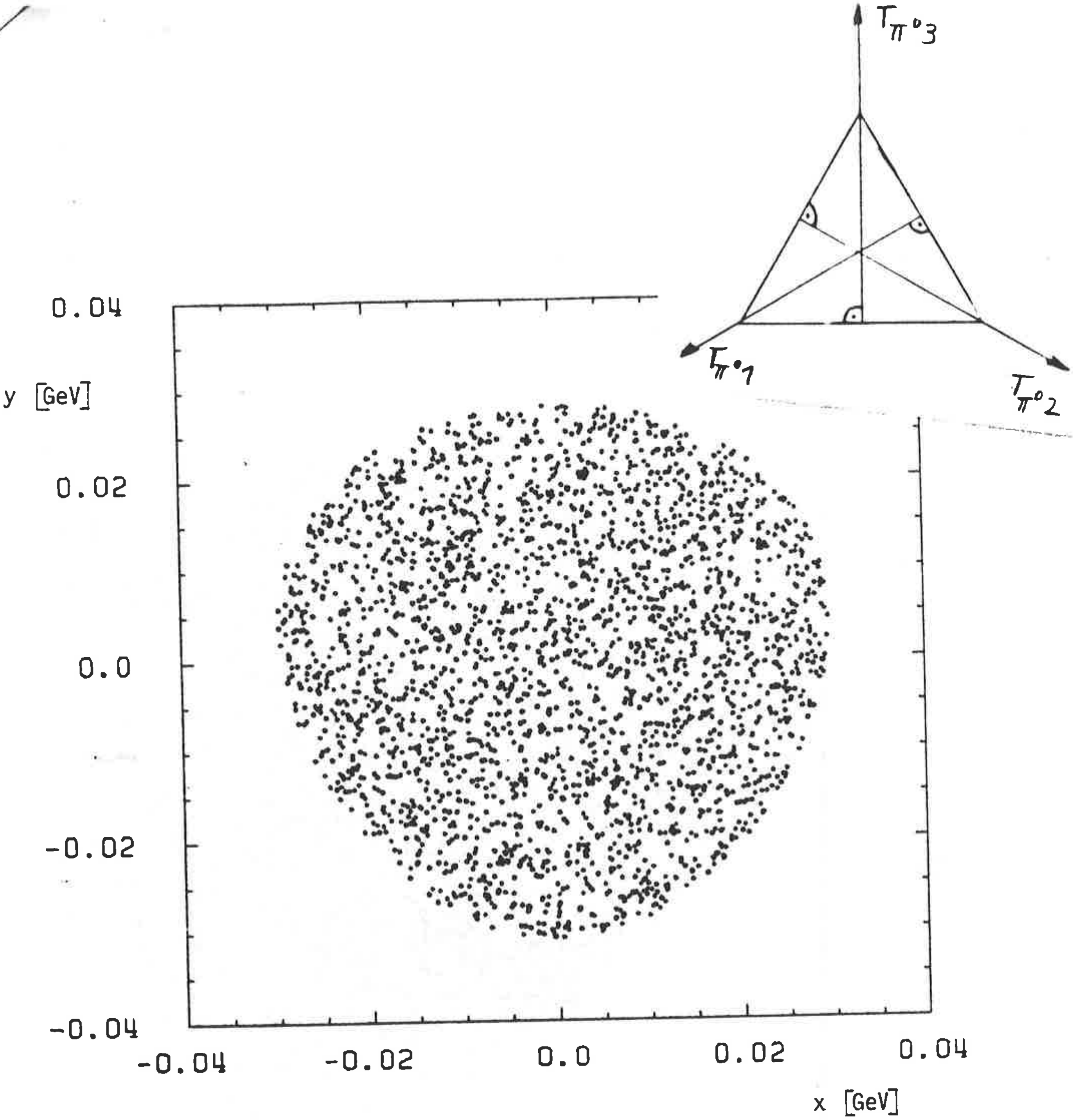

Fig. 1: Triangular Dalitz plot of the decay $K^0 \rightarrow \pi^0\pi^0\pi^0$.

$$x = \frac{(T_2 - T_1)}{\sqrt{3}}, \quad y = T_3 - \frac{Q}{3}, \quad Q = T_1 + T_2 + T_3.$$

$T_{1,2,3}$ are the kinetic energies of the π^0's.

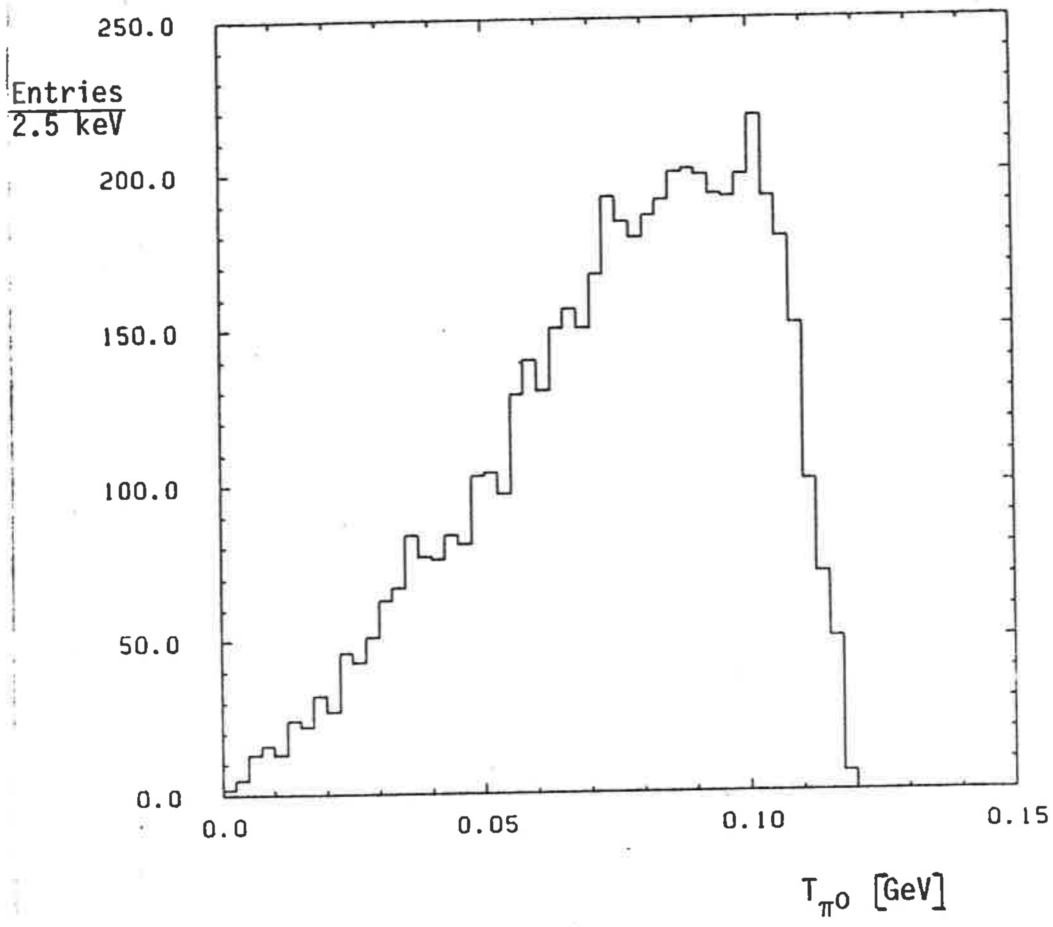

Fig. 2b: Spectrum of kinetic energy of the π^0 in the decay $k^\pm \to \pi^0 \mu^\pm \nu$ (y-projection of Fig. 2a)

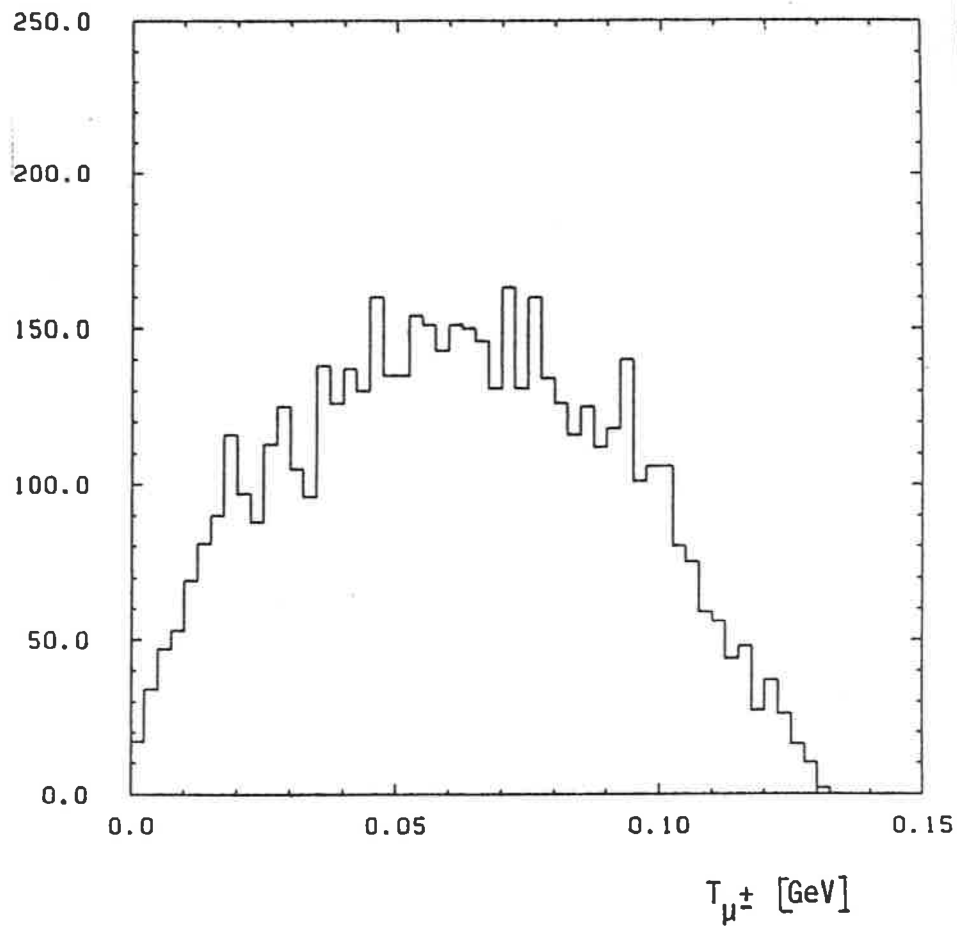

Fig. 2c: Spectrum of kinetic energy of the $\mu^\pm$ in the decay $k^\pm \to \pi^0 \mu^\pm \nu$ (x-projection of Fig. 2a)

Olsson

JADE COMPUTER NOTE 79

THE JADE TP PROGRAM

S. YAMADA, C. BOWDERY, E. ELSEN

7/12/84

ABSTRACT. This note describes the JADE TP Program, what it does, how it can be steered and what restrictions are imposed.

1. What is the TP Program?

The job of the JADE TP Program is to provide a simple means to analyse a JADE event and produce a set of banks summarising the results obtained. It is not so flexible for data reduction as the JADE SUPERVISOR Program[1] but it includes more analysis steps and is easier to control. The program reads a set of 'data cards' which the user must supply. These influence which analysis packages are called. A full set at present consists of 1 header 'card' and 19 TP flag 'cards'.

2. What do these Summary Banks contain?

The TP program produces 3 types of summary bank — TPEV, TPTR and TPVX. The TPEV bank contains a summary of the whole event such as the number of charged and neutral particles found, the number of vertices, the number of identified particles, etc. Thus there is only 1 TPEV bank per event. In contrast to this, there is a TPTR bank for every found particle in an event. Where present, charged particles occupy the TPTR banks with the lowest numbers with the photons and reconstructed V^0 particles following. The TP program does not take all charged tracks from the latest PATR bank because it attempts to eliminate those which are continuations of other ones, especially curling tracks. Also there is no attempt to match LG clusters to charged tracks other than by position. This implies that not all photons are found.

The TPVX banks give information about each vertex found by the Dittmann search program. A separate TPVX bank is created for each vertex with TPVX/1 being the bank for the primary vertex. All photons are assumed to originate from this vertex.

Full details of the TP banks can be found in JADE Computer Note 80.

[1] See JADE Computer Note 73.

The TP flags are:

1 Existing TP banks: 1 = scratch , 0 = keep

2 Pattern recognition: 1 = do if not already done, 2 = always do it, 0 = do nothing, < 0 = delete all PATR, JHTL and ZVTX banks[1] and then do pattern recognition

3 Vertex finding: 0 = omit, 1 = coplanar analysis only, 2 = 3-d Dittmann analysis

4 TOF analysis: 0 = omit, all other values = do analysis

5 TOF TP: 0 = omit, all other values = store results

6 dE/dx: 0 = omit, not 0 = do analysis and TP results

7 LG Cluster analysis: 0 = omit, 1 = do if not done, −1 = delete and re-analyse

8 LG matching to Tracks: 0 = omit, 1 = do it

9 LG Cluster TP: 0 = omit, 1 = store results

10 Muon analysis: 0 = omit, 1 = do if not done, −1 = delete old results and re-analyse

11 Unused

12 TP Muon results: 0 = omit, 1 = do it, −1 = as for 1 but MUR2 banks 4, 5 and 6 deleted at end of step,−2 = as for −1 but MUR2 banks 2 and 3 are also deleted. (Space saving option.)

13 Forward detector: 0 = omit, 1 = do if not done, −1 = as for 1 but old results deleted first

14 JETC calibration : 0 = omit, 1 = calibrate

15 Calculations: 0 = sphericity, > 0 = thrust[2] + sphericity

16 z recalibration: 0 = omit, 1 = yes, 2 = as 1 plus refit with no hit cleaning

17 Circle $r\phi$ refit: 0 = omit, 1 = yes, −1 = yes plus newest unrefitted PATR and JHTL banks deleted at end

18 Parabola $r\phi$ refit: 0 = omit, 1 = with no vertex constraint, 2 = with weak vertex constraint, 3 = with strong vertex constraint; add 10 for a common r-z refit; −ve option = +ve option but newest unrefitted PATR and JHTL banks deleted at end

19 TP Tracks: 0 = omit, 1 = choose circle fit PATR bank if present, 2 = choose parabola fit PATR bank if present, −ve options = +ve options but all JHTL banks deleted at end except for latest.(Space saving option)

[1]Except MC PATR/12

[2]Flag value = max. no. of tracks used in calculation

Title: Keep/Scratch Existing TP Banks

TP Flag 1: 1 ⇒ scratch, 0 ⇒ keep (see note)

Action: s/r TPSCRC called if TP flag 1 is non-zero. This deletes the TPEV bank and all the TPTR and TPVX banks.

Notes: Since partial TP updates cannot yet be done, it is essential that existing TP banks are deleted. To avoid mistakes, TP flag 1 is forced to be 1 and a warning message printed if the flag is set to another value. However it is the user's responsibility to set the correct flags to recreate the TP banks.

TP Error Messages: None.

Title: Vertex Finding and Fine Momentum Determination

TP Flag 3: 0 ⇒ omit, 1 ⇒ coplanar analysis only, 2 ⇒ analysis for all event types

Action: Existing GVTX banks are first deleted. If TP flag 3 is set to 2, s/r TPVTXD is called to find the primary and secondary vertices and compute the momentum of the associated particles at those vertices. The Dittmann 3 dimensional vertex search is used. If the event is coplanar, s/r TPVTXD abandons the vertex search and s/r TPVTX1 and s/r TPVTX2 are called to handle these cases. They perform a 2 dimensional vertex search in the x-y plane. If TP flag 3 is set to 1, these two subroutines are called immediately without calling s/r TPVTXD. In both cases, TPVX banks are created and filled for each vertex found.

TP Error Messages: Three error messages are possible from TP routines in this step. In addition there may be printout from the Dittmann routines if errors are found. Recovery action is taken.

- `**** Error in TPVTX1 ****` Vertex calculation does not converge
- `**** Error in TPVTX1 ****` No. of charged tracks (NTRK) exceeded 128
- `**** Error in TPVTX2 ****` No. of charged tracks (NTRK) exceeded 128

Title: Put TOF results into TP banks

TP Flag 5: 0 ⇒ omit, 1 ⇒ store results, −1 ⇒ as for 1 & delete TOFR bank

Action: s/r TPTOF called if TP flag 5 is non-zero. The most important TOF results are copied from the TOFR bank (if present) to the appropriate TPTR track summary banks. If TP flag 5 is set to −1 then the TOFR bank is deleted at the end of the step.

TP Error Messages: Two error messages are possible from the TP routine in this step. Recovery action is taken.

- `**** Error in TPTOF ****   Invalid inner detector track index found`
- `**** Error in TPTOF ****   'TOFR' bank is missing`

Title: LG Energy Cluster Finding

TP Flag 7: 0 ⇒ omit, 1 ⇒ do analysis if not done, −1 ⇒ delete old results and re-analyse

Action: s/r LGCALB (if needed) and s/r LGANAL are called if TP flag 7 is non-zero. LGCALB calibrates the ALGL bank to produce an ALGN bank if this is not already existing. LGANAL searches for clusters of electromagnetic energy deposited in the lead glass blocks (see JCN 14c). The results are stored in the LG cluster bank, LGCL.

TP Error Messages: None.

Title: LG Cluster Data Stored in TP Banks

TP Flag 9: $0 \Rightarrow$ omit. $\neq 0 \Rightarrow$ carry out step

Action: s/r TPLGCL is called if TP flag 9 is non-zero. Information from the LGCL bank (if present) is used to modify the TPTR banks for charged tracks and a new TPTR bank is created and filled for each photon cluster. The photons are assumed to originate from the primary vertex so the photon TPTR bank numbers are added to the TPVX/1 bank.

Notes: The first photon TPTR bank has the number following sequentially after the last TPTR charged track bank. If the TPTR bank for a photon already exists, then Step 9 terminates prematurely. This can lead to later error messages about missing TPVX banks.

TP Error Messages: There is only one TP error message, which could indicate that Step 3 was not done.

- `**** Error in TPLGCL **** , 'TPVX/1' bank is missing`

Warning: Step 9 requires that Step 8 is called in the same job since /CWORK/ is used to pass certain data values between the steps. If Step 9 is requested then Step 8 will be forced to be done.

Title: +++ Unused +++

TP Flag 11: reserved for future use

Action: None.

TP Error Messages: None.

Title: Tagging System Analysis and TP Summary

TP Flag 13: $0 \Rightarrow$ omit, $> 0 \Rightarrow$ do if not done, $< 0 \Rightarrow$ delete and re-analyse

Action: If ACLS/0 and TAGG banks 0 to 2 are present and TP flag 13 is positive, s/r TPFOWD is called to add new TPTR banks for each found tagging cluster. If any of the above banks are missing or TP flag 13 is negative, all the above banks are deleted and s/r TAGAN is called to analyse the Tagging Detector and then s/r TPFOWD is called.

TP Error Messages: None but error messages may be printed by the tagging analysis routines, possibly accompanied by a short dump.

TPSPRV: Step 15 Prog ID: 8.0 Date: 7/12/84

Title: Event Shape Calculations

TP Flag 15: 0 ⇒ sphericity only. > 0 ⇒ thrust as well (see below).

Action: s/r TPGNRL is called to compute the event momentum tensor eigenvectors and eigenvalues Q_i (related to sphericity). Note that the sum of the eigenvalues is normalised to 3. If TP flag 15 is positive, then thrust is also calculated using the value of TP flag 15 as the maximum number of particles to be used in the calculation.

tracks / photons ?

TP Error Messages: Three TP error messages are possible from this step.

```
- **** Error in TPGNRL ****   'TPVX'/I bank is missing
- **** Error in TPGNRL ****   'TPTR'/I bank is missing although ...
- **** Error in TPGNRL ****   Momentum buffer overflow
```

```
TP Bank Format
@
;
   JADE Computer Note 80                        12/12/84        C. Bowdery
   ---------------------                                         S. Yamada


                  Format of the Generation 8 TP Banks
                  -----------------------------------
;

   This note explains the contents of the TP summary banks produced
   by the TP program version 8. (This version is the one that prod-
   uces Generation 8 MH datasets but the program is able to process
   all types of events as well.)  The output of earlier versions of
   the TP program can be found in JADE Computer Note 24A.


   Changed lines are indicated by <<

   This note can be listed by submitting the job
                    'JADEPR.TEXT(JADECN80)'


     There are 3 banks to store the fully analysed results. They are

'TPEV'      which contains summary information for the whole event,

'TPTR'/n    a series of banks, each of which contains information about
            a single particle. (Neutral particles included.)

'TPVX'/n    a series of banks, each of which contains information about
            a found vertex. 'TPVX'/1 summarises the primary vertex and
            subsequent 'TPVX' banks summarise other vertices found.


@
     Variables in each bank are described in the following, where IDATA,
ADATA  and  HDATA are INTEGER*4,  REAL and INTEGER*2 words respectively,
equivalenced to the common /BCS/.  The index in the brackets is counted
from the  BOS  pointer.   Notice  that  the index for the I*2 variables
increases  twice  as fast as the others  and that the  BOS  pointer for
the I*2  variables  must be multiplied by 2.  In the following the  two
kinds of indices are treated separately.

 IDATA,ADATA:   'NAME',  NO.,NEXTP,LNGTH,    1,    2,    3,    4,....
 HDATA      :   'NAME',  NO.,NEXTP,LNGTH, 1, 2, 3, 4, 5, 6, 7, 8,....



;
                 'TPEV'/1 Bank     (Event Summary)
                 -------------------------------
;


IDATA( 1)  The version number - currently 8 .
  "  ( 2)  The production date and time. (This is fixed  at the
           beginning of the TP job.)
HDATA( 5)  # of recorded particles
  "  ( 6)  # of positive recorded particles
  "  ( 7)  # of negative recorded particles
  "  ( 8)  # of ambiguous charged tracks (i.e. sigma(rho) > rho)
  "  ( 9)  # of neutral particles (includes gammas)
```



```
  "  (10)  # of clusters in the - z tagging detector
  "  (11)  # of clusters in the + z tagging detector

@
  "  (12)  # of vertices in the event including the primary vertex
               = no. of 'TPVX' banks.
  "  (13)  # of neutral vertices (not including the primary vertex).
  "  (14)  # of charged vertices.

  "  (15)  # of  gammas in the 'TPTR' banks
  "  (16)  # of  e+                   "
  "  (17)  # of  e-                   "
  "  (18)  # of  mu+                  "
  "  (19)  # of  mu-                  "
  "  (20)  # of  pi+                  "
  "  (21)  # of  pi-                  "
  "  (22)  # of  pi0                  "
  "  (23)  # of  K+                   "
  "  (24)  # of  K-                   "
  "  (25)  # of  K0/s                 "
  "  (26)  # of  eta-0                "
  "  (27)  # of  protons              "
  "  (28)  # of  anti-protons         "
  "  (29)  # of  lambda-0             "
  "  (30)  # of  unknown particles"

  "  (31)  A flag of seen particle types (additive)
           ( 1 = gamma, 10 = e+-, 100 = mu+-, 1000 = hadrons )
  "  (32)  Not used
  "  (33)   "   "
  "  (34)   "   "
@
           N.B. The following energies and momenta are measured in GeV.

ADATA(18)  Visible charged particle energy (E_visM-Ach)
  "  (19)  Sigma(E_visM-Ach)
  "  (20)  Visible neutral particle energy (E_visM-Aneu)
  "  (21)  Sigma(E_visM-Aneu)

  "  (22)  Missing momentum (p_miss) x
  "  (23)      "       "             y
  "  (24)      "       "             z
  "  (25)  Sigma(p_miss) x
  "  (26)    "           y
  "  (27)    "           z

HDATA(55)  Charge sphericity flag
              = 0, if all tracks are included.
              = 1, if only charged tracks are used.
  "  (56)  # of tracks used for the sphericity calculation


ADATA(29)  Smallest momentum tensor ellipse eigenvalue, alpha1 = S
  "  (30)  Middle       "        "       "        "     , alpha2
  "  (31)  Largest      "        "       "        "     , alpha3

           Q1 = 1 - 2/3 * alpha3           alpha1 + alpha2 + alpha3 = 3
           Q2 = 1 - 2/3 * alpha2               Q1  +   Q2   +   Q3  = 1
           Q3 = 1 - 2/3 * alpha1

           Sphericity   (S) = 1/2 * ( Q1 + Q2 ) = alpha1    ( 0..1 )
           Acoplanarity (A) = 1/2 * Q1                      ( 0..1 )
@
ADATA(32)  Direction cosine x  of the eigenvector corresponding to Q3
  "  (33)      "        "   y                 "
  "  (34)      "        "   z                 "

              This is the alpha1 or sphericity axis (for p(long.) ).

  "  (35)  Direction cosine x  of the eigenvector corresponding to Q2
  "  (36)      "        "   y                 "
```

```
  "  (37)     "        "    z            "

           This is the alpha2 or major axis (for p(t)_in ).

  "  (38)  Direction cosine x  of the eigenvector corresponding to Q1
  "  (39)    "           "    y            "
  "  (40)    "           "    z            "

           This is the alpha3 or minor axis (for p(t)_out ).

HDATA(81)  # of tracks used for the thrust calculation
  "  (82)  Max. # of tracks accepted by the thrust program
ADATA(42)  Thrust    ( T : 0.5 ... 1.0 )           (if calculation done)
  "  (43)  Direction cosine x of the thrust axis ( "        "          " )
  "  (44)    "           "    y            "     ( "        "          " )
  "  (45)    "           "    z            "     ( "        "          " )
  "  (46)  Not used
  "  (47)   "
  "  (48)   "
  "  (49)   "
  "  (50)   "
@
ADATA(51)  TOF of the beam counter
  "  (52)  Hit time difference for 2-prong events
  "  (53)  Collinearity of 2-prong events
  "  (54)  Acoplanarity of 2-prong events
  "  (55)  Not used


           The following 10 I*2 words are error flags for each step.
           General error codes (additive):
                     10000  the necessary raw data is missing.
                      4000  corresponding TP-subroutine was not called
                      2000  /BCS/ space was not enough for a new bank.
                      1000  the necessary result bank is missing or
                            it has an error.
                        -1  analysis or TP program is not ready yet.

HDATA(111)   Error flag for Jet Chamber pattern recognition
  "  (112)       "              vertex fit
  "  (113)       "              TOF
  "  (114)       "              dE/dx
  "  (115)       "              Lead Glass
                        100  LGCDIR was not called or ended prematurely
                         10  ID track-LG connection was not done.
                          1  LG-cluster energy correction was not done.
  "  (116)       "              Muon detector
  "  (117)       "              Tagging detector
  "  (118)       "              pairs and vees
  "  (119)       "              jet analysis
  "  (120) Not used

@
;
              'TPTR'  Bank    (One Particle Summary)
              --------------------------------------

;

      A 'TPTR' bank is made for each particle.  The length
      of the bank is different for each type, e.g. for gammas
      the TOF and dE/dx information is omitted.
      PATR tracks: 80 words,  LG Photons: 50 words,    Others: 40 words

HDATA( 1)  The index( = the TPVX bank #) of the vertex for the track
  "  ( 2)  The index( = the TPVX bank #) of the secondary vertex if any
  "  ( 3)  Flag of the detectors where the track is seen
           1000  Jet Chamber
            100  Lead Glass shower counters
             10  Muon Filter
              1  Tagging Detector
```



```
  "  ( 4)  The index of the track in the 'PATR' bank, if seen there,
              (otherwise 0)
  "  ( 5)  Number of LG clusters connected to the track
             0  not detected by the LG and it is not expected to be
            -1    "        "     "    "   although it is expected.
                   (The extrapolated hit position is near the edge)
            -2  not detected by the LG although LG hit is expected
                within the fiducial detection region.
  "  ( 6)  The index of the 1-st connected LG cluster
  "  ( 7)   "   "     "   "  2-nd     "         "    "
  "  ( 8)  Not used       <<
  "  ( 9)   "   "         <<       This was not filled even
  "  (10)   "   "         <<       by the old program.
  "  (11)   "   "         <<
  "  (12)   "   "         <<
@
HDATA(13)  The index of the Tagging Detector cluster
  "  (14)  Not used
  "  (15)   "   "
  "  (16)   "   "
  "  (17)  Type of the stored track origin
              1  first observed point in the Jet Chamber
              2  the fixed point (x,y,z) = (0,0,0)
              3  the closest point from the beam axis on the track
                 extrapolation
             20  on the beam beam axis,at z = z_vertex of the event
                 (used for gammas)
             30  closest point on the track from the fitted event vertex
  "  (18)  Original track flag.
              1  if the track is used to fit the event vertex.
              0  otherwise.
ADATA(10)  x coordinate of the track origin   = x(origin)
  "  (11)  y           "                      = y(origin)
  "  (12)  z           "                      = z(origin)
  "  (13)  Sigma( x(origin) )
  "  (14)  Sigma( y(origin) )
  "  (15)  Sigma( z(origin) )
  "  (16)  The shortest distance from the vertex to the track=abs(delta)
  "  (17)  d(r)/sigma(r)    ( not filled )
  "  (18)  d(z)/sigma(z)    ( not filled )

  "  (19)  Chi-squared of the (r-phi) fit
IDATA(20)  Number of degrees of freedom of the (r-phi) fit
ADATA(21)  Chi-squared of the (r-z) fit
IDATA(22)  Number of degrees of freedom of the (r-z) fit
@
ADATA(23)  Electric charge   ( = 100 if not known )
  "  (24)  Momentum (GeV/c)    = p
  "  (25)  Sigma(p)

IDATA(26)  Type of the stored track direction
             1 the line direction from the vertex to the first hit point
             2 the tangent direction on the track at the closest point
               from the vertex
ADATA(27)  x-component of the direction cosine  alpha-x
  "  (28)  y-component  "  "        "        "   alpha-y
  "  (29)  z-component  "  "        "        "   alpha-z
  "  (30)  Sigma( alpha-x )
  "  (31)  Sigma( alpha-y )
  "  (32)  Sigma( alpha-z )

  "  (33)  Not used

HDATA(67)  Input mass type (available only for MC test data)
           The mass code is given below.
  "  (68)  Most likely particle type
             0 = unknown,           1 = gamma,
             2 = electron,          3 = muon,
             4 = pion,              5 = kaon,
             6 = proton/neutron,    7 = lamda
ADATA(35)  Most likely mass of the particle in GeV/c**2 = AMASS
```

```
  "  (36)  Total energy       = ETOT = SQRT( p**2 + AMASS**2 )

  "  (37)  Shower energy      = ESH
  "  (38)  Sigma(ESH)
@
HDATA(77)  Quality of the shower energy measurement
             -2  not detected by the LG although LG hit is expected
                 within the fiducial detection region.
             -1  not detected by the LG although hit is expected
                 near the detector edge.
              0  not detected and a hit is not expected due to
                 the gap in the LG detector or absorption in the coil.
              1  the connected LG cluster is near the detector edge.
                 (ESH may not be correct)
              2  the connected LG cluster is in the fiducial region.

  "  (78)  Uniqueness of the cluster assignment
           = number of other tracks which share the same connected
             clusters. =0 , if the connection is unique.
ADATA(40)  Chi-squared deviation of the ESH and p for a shower
           =( (ESH-p)/sigma(ESH) )**2
ADATA(41)  Chi-squared deviation of ESH and expected ESH for a
           non-showering track
           =(( ESH-(Expected_ESH) / sigma(Expected_ESH) )**2
           Temporarily  Expected_ESH = 0.25, sigma(Expected_ESH) = 0.125
  "  (42)  Not used

HDATA(85)  Number of associated muon hits with the Jet Chamber track
  "  (86)  The Muon Filter acceptance flag          <<
             0  track is safely inside the Muon Filter acceptance
             1  track is at edge of Muon Filter acceptance
             2  track is definitely outside the Muon Filter acceptance

  "  (87)  Muon track quality flag (see JCN 22).
             <  1  track did not satisfy basic muon criteria
           1..99  track satisfied basic muon criteria(1 & 2 = good)
            > 99  probably a penetrating hadron
@
HDATA(88)  Not used.      <<
  "  (89)   "   "         <<
  "  (90)   "   "         <<      These were not filled even
  "  (91)   "   "         <<      with earlier versions of
  "  (92)   "   "         <<      the TP program.
  "  (93)   "   "         <<
  "  (94)   "   "         <<
  "  (95)   "   "         <<
  "  (96)  Muon selection flag  ("MUCUTS")     (see JCN 22).
             0  not a good muon candidate
             1  good muon candidate
             5  better muon candidate
             9  best muon candidate

ADATA(49)  Probability of chi-squared for the muon candidate.(0.0..1.0)
           = 0 if muon quality flag is not greater than zero
  "  (50)  Integral material thickness in mm
  "  (51)  Integral material thickness in g/cm**2
  "  (52)  Total energy loss in the material (GeV)
  "  (53)  Integral absorption length for a pion
  "  (54)  Energy at outermost hit assuming particle is a muon (GeV).
  "  (55)  Probability of being a muon ( = 1.0  if  ADATA(49) > 0 )
  "  (56)  Probability of being a punchthrough pion
  "  (57)  Not used

IDATA(58)  Quality of TOF
             1  one hit and unique solution
             2  two tracks hit the same counter but resolved
            -1  only one hit but left/right TOF does not agree
            10  >=2 hits and can not be resolved
@
ADATA(59)  TOF in nsec (after all corrections)
```



```
  "  (60)  Path length to the TOF counter hit point
  "  (61)  Beta
  "  (62)  Sigma( beta )
  "  (63)  Calculated mass  (GeV/c**2)
  "  (64)  Sigma( mass )
  "  (65)  Chi-squared to be a proton   from TOF data
  "  (66)       "       "     kaon        "   "   "
  "  (67)       "       "     pion/muon   "   "   "
  "  (68)       "       "     electron    "   "   "
  "  (69)  dE/dx in the TOF counter
  "  (70)  Not used
  "  (71)   "   "

IDATA(72)  Quality of the Jet Chamber dE/dx measurement
ADATA(73)  dE/dx
  "  (74)  Sigma( dE/dx )
  "  (75)  Chi-squared to be a proton   from the dE/dx data
  "  (76)       "       "     kaon        "   "    "    "
  "  (77)       "       "     pion/muon   "   "    "    "
  "  (78)       "       "     electron    "   "    "    "
IDATA(79)  Particle type with lowest chi**2 ( numbered as above )
  "  (80)  Not used

@

;
                'TPVX'  Bank    (One Vertex Summary)
                -----------------------------------
;

        A 'TPVX' bank is made for each vertex. The first bank 'TPVX'/1 is
        used for the event vertex.  The length of the bank is different
        for each bank.


HDATA( 1)  The bank no. of the primary particle which makes this vertex.
           For 'TPVX'/1 it is 0.
  "  ( 2)  Flag of the vertex calculation
           = 10 * ( number_of_used_tracks) + fitting_mode
           Fitting mode = 0,1     the vertex is not calculated.
                        = 2       closest point from the beam axis
                        = 3       fitted in the (x,y) projection
                        = 4       fitted in the 3-dim. space
ADATA( 2)  x coordinate of the vertex       = xvtx
  "  ( 3)  y     "                          = yvtx
  "  ( 4)  z     "                          = zvtx
  "  ( 5)  Sigma( xvtx )
  "  ( 6)  Sigma( yvtx )
  "  ( 7)  Sigma( zvtx )
  "  ( 8)  Chi-squared of the vertex fit

IDATA( 9)  Number of degrees of freedom in the fit

           ****** Important caution ******                              <<
           If fitting mode is 2 or 3, this word is REAL; ADATA(9).     <<
           = dev**2 / (sigma**2 * number_of_d.o.f.)                     <<
@
ADATA(10)  COST =  < cos( angle_between_tracks ) >

HDATA(21)  Charge of the vertex ( = charge of the original track)
  "  (22)  # of the secondary tracks emitted from the vertex + mulsec
  "  (23)  # of the positive secondary tracks
  "  (24)  # of the negative secondary tracks
  "  (25)  # of the neutral secondary tracks including gammas
  "  (26)  # of ambiguous secondary tracks
  "  (27)  # of gammas
  "  (28)  # of electron/positrons
  "  (29)  # of muons
  "  (30)  # of hadrons (including ambiguous tracks)
```

```
*  (31)  The bank no. of the 1-st secondary track
*  (32)        "            2-nd    "
*  (33)        "            3-rd    "
*
******
*
*  (30+mulsec) "            last    "
```

JADE Computer Note 81

LUMI and RANDOM events on Tape

JANUARY 22, 1987
J. OLSSON

In order to determine trigger efficiencies it is sometimes necessary to study the background conditions during the actual data taking time. For this purpose randomly triggered events are available in the data. These are T1 accept triggers (bit 16, see Jade Note 32 and supplements) with only beam crossing and a clock rate as trigger conditions. For background studies one can also use the small angle Bhabha scattering events, detected in the forward tagging counters. These are the Lumi triggers (T1 accept, bit 1), with no inner detector, TOF, lead glass or muon filter conditions required in the trigger.

Since there are about 250 000 beam crossings per second, a large number of such events have to be studied in order to obtain significant information. This requires the reading of a large number of tapes. For the convenience of such studies, all random and lumi triggers in the data since 1982 (random triggers are only available since 1982) have been collected from the original data tapes (Reform Tapes). This note gives information on the name of tapes, number of events, etc. The total number of lumi and random events (1982-1984) is $\sim$ 2 millions, corresponding to $\sim$ 8 seconds of beam time.

The collected Lumi and Random events are still uncalibrated, i.e. they are "Reform"-data. Any analysis that uses lead glass results requires the use of the "B-files", i.e. `F11LHO.BUPDAT0` and `F11LHO.BUPDAT1`.

Use the following code to select either LUMI or RANDOM (or both) triggers:

```
C
      INTEGER*2 (H)
      COMMON /BCS/ IDATA(40000)
      DIMENSION HELP(2), HDATA(80000)
      EQUIVALENCE (ICAMWD,HELP(1)), (IDATA(1),HDATA(1))
      DATA HELP /0,0/, MSKLUM/Z0001/, DATA MSKRAN/Z8000/, MSKMIX/Z8001/
C                         READ STEERING PARAMETER AND SET SELECTION MASK
      READ 3810, IRNSEL
3810  FORMAT(I10)
      MSKSEL = MSKLUM
      IF(IRNSEL.EQ.2) MSKSEL = MSKRAN
      IF(IRNSEL.EQ.3) MSKSEL = MSKMIX
      WRITE(6,3811) IRNSEL,MSKSEL
3811  FORMAT(' IRNSEL ',I5,'  MASK = ',Z8)
C         EVENT LOOP:                  READ EVENT AND LOCATE TRIG 1 BANK.
      IPTRIG = IDATA(IBLN('TRIG'))
      IF(IPTRIG.LE.0) GO TO 1
C                    NO TRIG BANK, SKIP AND READ NEXT EVENT  ( GO TO 1 )
      HELP(2) = HDATA(2*IPTRIG+8)
C                                        HELP(2) NOW HOLDS T1 ACCEPT WORD
      IF(LAND(MSKSEL,ICAMWD).NE.0) GO TO 11
C                 IF BIT SET, WRITE EVENT AND READ NEXT EVENT (GO TO 11)
      GO TO 1
```

F11LHO.JDATA09.REFORM . G0xxxV00 (1983)			
Raw Data (Reform)	Lumi and Random Tape	Run nr.	Nr. of events
xxx = 001 – 030	F11OLS.LUMI09 . T001T030	13598 – 13767	39990
xxx = 031 – 070	F11OLS.LUMI09 . T031T070	13768 – 13918	40939
xxx = 071 – 115	F11OLS.LUMI09 . T071T115	13919 – 14088	41166
xxx = 116 – 175	F11OLS.LUMI09 . T116T175	14089 – 14295	39658
xxx = 176 – 205	F11OLS.LUMI09 . T176T205	14296 – 14399	31615
xxx = 206 – 230	F11OLS.LUMI09 . T206T230	14400 – 14483	28768
xxx = 231 – 244	F11OLS.LUMI09 . T231T244	14484 – 14548	21078

F11LHO.JDATA10.REFORM . G0xxxV00 (1983 -- 1984)			
Raw Data (Reform)	Lumi and Random Tape	Run nr.	Nr. of events
xxx = 001 – 050	F11OLS.LUMI10 . T001T050	14549 – 14794	37709
xxx = 051 – 085	F11OLS.LUMI10 . T051T085	14795 – 14980	35732
xxx = 086 – 115	F11OLS.LUMI10 . T086T115	14981 – 15095	37147
xxx = 116 – 155	F11OLS.LUMI10 . T116T155	15096 – 15276	38397
xxx = 157 – 190	F11OLS.LUMI10 . T157T190	15282 – 15459	30012
xxx = 191 – 220	F11OLS.LUMI10 . T191T220	15460 – 15619	24166
xxx = 221 – 249	F11OLS.LUMI10 . T221T249	15565 – 15770	21823

F11LHO.JDATA11.REFORM . G0xxxV00 (1984)			
Raw Data (Reform)	Lumi and Random Tape	Run nr.	Nr. of events
xxx = 001 – 025	F11OLS.LUMI11 . T001T025	15771 – 15934	40032
xxx = 026 – 055	F11OLS.LUMI11 . T026T055	15935 – 16079	37205
xxx = 056 – 085	F11OLS.LUMI11 . T056T085	16080 – 16291	37958
xxx = 086 – 110	F11OLS.LUMI11 . T086T110	16292 – 16530	41004
xxx = 111 – 170	F11OLS.LUMI11 . T111T170	16531 – 17012	39263
xxx = 171 – 190	F11OLS.LUMI11 . T171T190	17013 – 17182	42251
xxx = 191 – 205	F11OLS.LUMI11 . T191T205	17183 – 17290	35512
xxx = 206 – 220	F11OLS.LUMI11 . T206T220	17291 – 17388	42697
xxx = 221 – 235	F11OLS.LUMI11 . T221T235	17389 – 17475	36677
xxx = 236 – 245	F11OLS.LUMI11 . T236T245	17455 – 17525	28596
xxx = 246 – 251	F11OLS.LUMI11 . T246T251	17526 – 17562	18820

F11LHO.JDATA14.REFORM . G0xxxV00 (1985)			
Raw Data (Reform)	Lumi and Random Tape	Run nr.	Nr. of events
xxx = 001 – 025	F110LS.LUMI14 . T001T025	22124 – 22296	36440
xxx = 026 – 055	F110LS.LUMI14 . T026T055	22297 – 22476	36057
xxx = 056 – 080	F110LS.LUMI14 . T056T080	22477 – 22649	35795
xxx = 081 – 105	F110LS.LUMI14 . T081T105	22650 – 22810	35479
xxx = 106 – 130	F110LS.LUMI14 . T106T130	22811 – 22946	28381
xxx = 131 – 165	F110LS.LUMI14 . T131T165	22947 – 23156	34272
xxx = 166 – 190	F110LS.LUMI14 . T166T190	23157 – 23302	33721
xxx = 191 – 215	F110LS.LUMI14 . T191T215	23303 – 23451	45354
xxx = 216 – 230	F110LS.LUMI14 . T216T230	23449 – 23570	48138
xxx = 231 – 245	F110LS.LUMI14 . T231T245	23571 – 23685	45926
xxx = 246 – 255	F110LS.LUMI14 . T246T255	23686 – 23758	26065

F11LHO.JDATA15.REFORM . G0xxxV00 (1985 -- 1986)			
Raw Data (Reform)	Lumi and Random Tape	Run nr.	Nr. of events
xxx = 001 – 020	F110LS.LUMI15 . T001T020	23759 – 23903	49245
xxx = 021 – 040	F110LS.LUMI15 . T021T040	23904 – 24046	44560
xxx = 041 – 060	F110LS.LUMI15 . T041T060	24047 – 24160	51423
xxx = 061 – 085	F110LS.LUMI15 . T061T085	24161 – 24434	49065
xxx = 086 – 095	F110LS.LUMI15 . T086T095	24435 – 24578	41307
xxx = 096 – 115	F110LS.LUMI15 . T096T115	24579 – 24755	51978
xxx = 116 – 130	F110LS.LUMI15 . T116T130	24756 – 24866	45623
xxx = 131 – 140	F110LS.LUMI15 . T131T140	24867 – 24955	36581
xxx = 141 – 155	F110LS.LUMI15 . T141T155	24960 – 25064	34591
xxx = 156 – 170	F110LS.LUMI15 . T156T170	25065 – 25194	42425
xxx = 171 – 185	F110LS.LUMI15 . T171T185	25195 – 25303	53787
xxx = 186 – 200	F110LS.LUMI15 . T186T200	25304 – 25430	55133
xxx = 201 – 215	F110LS.LUMI15 . T201T215	25431 – 25545	49354
xxx = 216 – 230	F110LS.LUMI15 . T216T230	25546 – 25673	57849
xxx = 231 – 245	F110LS.LUMI15 . T231T245	25674 – 25798	50677
xxx = 246 – 255	F110LS.LUMI15 . T246T255	25799 – 25872	27569

86 :
> 24214