SCN 69

MONTE CARLO TRACEBACK 28/10/83
@
JADE COMPUTER NOTE 69 C. BOWDERY

MONTE CARLO TRACEBACK

Introduction

In some fields of analysis, it is very important to be able to trace the history of a given Monte Carlo-generated Jet Chamber track. This JADE Computer Note will explain how this is now possible. Section A will explain an easy way to access the history information in an analysis program. Unfortunately this only applies to Monte Carlo events! For those interested in the implementation details, section B will explain the traceback from a PATR track to a VECT bank particle (a '4-vector') and section C will explain the traceback from a VECT bank particle to its origins in an e+e- collision and the conventions used in the new PALL bank. Please note that the word 'track' always means a PATR track and the word 'particle' always means a '4-vector' in this note.

IMPORTANT: The PATR Traceback will only work with events tracked since the 20th October 1983. Old events however will not cause the programs to crash. Also the 4-vector History Traceback will only work for events conforming to the PALL conventions that are explained in Section C.

@
A) Accessing the Monte Carlo Traceback

Two easy-to-use subroutines have been written to provide the user with the essential history information compiled during event generation, tracking and basic analysis. The first subroutine links a given PATR track to one or more 4-vector particles in the VECT banks.

```
      DIMENSION  IPART(3), IVECT(3), FRACT(3)
      .......
      .......
      CALL MCTRCB( ITRACK, NPART, IPART, IVECT, FRACT )
```

```
Input:  ITRACK =  Track number in the most recent PATR bank
Output: NPART  =  Number of 4-vectors associated (usually 1)
        IPART  =  Array of associated VECT particle numbers
        IVECT  =  Array of VECT bank numbers corresponding to IPART: 0/1
        FRACT  =  Array of hit fractions for each associated particle
```

If ITRACK is less than 1 or greater than the maximum number of tracks, MCTRCB will return with NPART set to -2 after printing a message on unit 6 (maximum of 10 messages). If NPART is -1 then there was a problem finding the TR4V bank (see Section B). This means that no traceback information was available which could mean that old Monte Carlo events are being processed. Otherwise NPART can take any value between 0 and 3. IVECT elements can be 0 or 1 signifying the traced 4-vector belongs to VECT/0 and VECT/1 respectively. FRACT values lie in the range 0.0 to
@
1.0 and signify the fraction of hits associated with ITRACK that were caused by the particular 4-vector. The sum of all FRACT's can exceed 1.0 if hits are 'claimed' by more than one 4-vector.

A second or third 4-vector will be associated with ITRACK if at least 8 hits belong to a second (or third) 4-vector and that 4-vector is not associated with another PATR track with a higher hit fraction. (Only 5

hits are required if the total number of hits on the PATR track is less than 20.) There will probably be some cases when MCTRCB only returns one associated 4-vector when there should have been 2 and vice versa but this should not be a big problem.

Examples

a) PATR track 5 was caused by 4-vector particle 7 in VECT/0.

```
   ==> NPART = 1  ,  IPART(1) = 7  ,  IVECT(1) = 0  ,  FRACT(1) = 1.00
```

b) PATR track 3 is a fit through the hits of 4-vector 2 in VECT/0 which decayed to 4-vector 9 in VECT/1.

```
   ==> NPART = 2  ,  IPART(1) = 2  ,  IVECT(1) = 0  ,  FRACT(1) = 0.39
                     IPART(2) = 9  ,  IVECT(2) = 1  ,  FRACT(2) = 0.61
```

c) PATR track 10 is the result of two very close particles, 7 in VECT/0 and 8 in VECT/0.

```
   ==> NPART = 2  ,  IPART(1) = 7  ,  IVECT(1) = 0  ,  FRACT(1) = 1.00
                     IPART(2) = 8  ,  IVECT(2) = 0  ,  FRACT(2) = 0.98
```

@ The second routine, MCHTRB, provides a traceback from a 4-vector in VECT/0 to its ancestors in the PALL bank (if filled according to the conventions given in Section C).

```
      DIMENSION  P9VECT(9,30) , IPALL(30)
      .......
      .......
      CALL MCHTRB( I4VECT, NFOUND, P9VECT, IPALL, IFLAVR, IQG, IPN )
```

```
Input:  I4VECT  = Particle number in VECT/0
Output: NFOUND  = No. of ancestors found including original particle
        P9VECT  = Array of 9-vectors for each of the NFOUND particles
        IPALL   = Array of pointers to PALL for each found particle
        IFLAVR  = Flavour of the event from PALL header (if relevant)
        IQG     = Index pointer to P9VECT for the parton ancestor
        IPN     = Parton parent order number (see Section C)
```

For each particle there is a 9-vector of information in P9VECT:
px, py, pz, E, m, charge, type, parent_number, parton_parent_number

The order of the particles in P9VECT is as follows:

```
1   original particle corresponding to VECT/0 entry
2   parent particle
3   grandparent particle
.     .......
.     .......
.   parton ancestor ( quark or gluon)               -- IQG points to here
.   virtual spin 1 boson ( photon or Z )            -- if present
.   initial particles ( e+,e-,radiated photon )     -- if present
```
@
Please note that ALL the components of P9VECT are REAL*4 (even the type, charge and pointers which are usually INTEGER*4) in order to simplify the interface. Full details of the 9-vectors are given in Section C. Please also note that the parent_numbers here are the pointers to the parents in the PALL bank. Obviously the position of a parent in P9VECT is just the next particle down in the list. The type values depend on which 4-vector generator was used to produce the Monte Carlo events.

N.B. If NFOUND is 0 then this may mean that the PALL bank does not conform to the convention defined in Section C. This will be the case for MC events which only have 'produced' particles stored in the first part of the event record (see Section C).

```
      NFOUND = -1     no VECT bank  (no warning message printed)
             = -2     no PALL bank  (no warning message printed)
```

```
           = -3     I4VECT particle number illegal (message printed)

Both these routines reside on 'F11LHO.JADEGS/L'.  MCHTRB needs s/r VZERO
                                                   from the CERN library.

B)  PATR Traceback
    --------------

   In order to systematically traceback a PATR  track  to  its  4-vector
origin, it is first necessary  to  store  the  association  of  the  jet
chamber hits with the 4-vector particle that produced  them.  Since  the
hit/track assignment of hits made in PATREC is stored in the  JHTL  (hit
label)  bank,  it  is  then  possible  to  associate  PATR  tracks  with
4-vectors. However, this simple outline is complicated  by  jet  chamber
smearing, random hits, measuring inefficiencies and overlapping  tracks.
We will consider the problem and its implemented solution in 3 stages.
@   Stage 1 :       4-vector particle / JETC hit association during
                    the Detector Simulation ("Tracking")
    Stage 2 :       Maintaining the association during JETC "Smearing"
    Stage 3 :       After PATREC, association of PATR tracks to 4-vector
                    particles in the VECT banks.

  Stage 1  (During Tracking)
  --------------------------

   The basic idea at stage 1 is to record the 4-vector  particle  number
in a BOS bank ('HITL') whenever a jet chamber hit is stored in the  JETC
bank during tracking. The implementation of  this  scheme  was  slightly
complicated by the fact that the jet chamber hits are first created  and
then  sorted  (by wire  numbers  and  drift  times).  To  overcome  this
problem, the hit label information is  simultaneously  sorted  with  the
hits.
          HITL Bank Structure
          -------------------

     Name of bank   :   HITL                         (= Hit Label)
     No. of bank    :    8                           (= JETC bank number)
     Length         :  No._of_JETC_hits + 1   half-words     (+1 if odd)

     Half-word         Contents                 Comment
     ---------         --------                 -------

         1               HITS          Number of (unsmeared) JETC hits
         2               HWORD1        Encoded 4-vector info for hit 1
         3               HWORD2           "         "      "   "   "  2
         .                 .                        .
         .                 .                        .
         .                 .                        .
     HITS + 1            HWORDn        Encoded 4-vector info for last hit
@
 The code used for the HWORDs is:      2 * particle_number + VECT_bank_no.

       e.g.   Particle  5 in VECT/0    would be code 10
              Particle 11 in VECT/1    would be code 23

 The following subroutines were modified to implement stage 1:

   JRING   :    Hit/particle association recorded here when hit created
   JHITIN  :    Hit/particle association information sorted here
   WRTMCB  :    HITL bank created and filled with the association info.

  Stage 2  (Reading Unsmeared Tracked Events)
  -------------------------------------------

   When the jet chamber hits are smeared in  RDMTCO,  the  corresponding
hit label information has to be updated. Basically, stage  2  creates  a
new BOS bank ('HTSL') which links  the  unsmeared  hits  with  the  new,
smeared ones. The following features had to be considered:

   Double hit resolution:   Hits lost due to the double hit resolution
```



```
                              cut are linked to the hit that absorbed them
                              but marked with a minus sign.
   Inefficiency losses:       Hits killed for this reason are marked with
                              a zero.
   Random hits:               These are not associated with any unsmeared
                              hits. They just alter the hit numbering.

@          HTSL Bank Structure
           -------------------

     Name of bank   :   HTSL                         (= Smeared Hit Label)
     No. of bank    :    8                           (= HITL bank number )
     Length         :  No._of_JETC_hits + 1   half-words    (+1 if odd)

     Half-word         Contents                 Comment
     ---------         --------                 -------

         1               HITS          Number of (unsmeared=old) JETC hits
         2               HLINK1        Pointer to smeared hit for old hit 1
         3               HLINK2           "     "    "     "   "   "   "  2
         .                 .                        .
         .                 .                        .
         .                 .                        .
     HITS + 1            HLINKn      Pointer to smeared hit for last old hit

   Thus:  HLINKi = 0      means unsmeared hit 'i' has been 'killed'
                 > 0      means that the unsmeared hit 'i' is linked to a
                          smeared hit with number HLINK
                 < 0      means that the unsmeared hit 'i' was lost due to
                          the double hit resolution. Smeared hit | HLINK |
                          was the hit that absorbed it.

 The following subroutines were modified to implement stage 2:

   RDMERG  :     Re-orders hits after creation of random hits in RDRDMH.
   RDDOUB  :     Double hit resolution cut applied here.
   RDPOIN  :     Edits out 'killed' and 'absorbed' hits.
   RDMODN  :     Now includes code to create the HTSL bank.
@ Stage 3  (Analysis of Smeared Data)
  ---------------------------------

   After PATREC has created the  PATR  and  JHTL  banks  containing  the
results of the track search, stage 3 of  the  traceback  scheme  can  be
carried out. This will normally be done in the SUPERVISOR but  users can
perform  this  task  themselves  by  calling  subroutine  MCTR4V.   This
processes the PATR, JHTL, HITL and HTSL banks in order to create  a  new
BOS bank ('TR4V') containing the desired traceback from every PATR track
to a 4-vector particle. Additionally, MCTR4V is called  in  MCTRCB  (see
Section A) to create the TR4V bank there if it does not already exist.

 The following FORTRAN statement invokes MCTR4V:
       ....
       ....
       CALL MCTR4V( IOPT, IERROR )
       ....
       ....

 where:   IOPT  is an input parameter:   0 = perform stage 3 if not done
                                             already for the latest PATR
                                             bank, i.e. create TR4V bank
                                         1 = delete old stage 3 results
                                             for latest PATR bank and redo
          IERROR is a return code:       0 = no errors
                                         1 = BOS error. TR4V not created
                                         2 = HITL bank not same length
                                             as HTSL bank. No TR4V bank
                                             was created.

   The information from the HTSL and HITL banks  are  unpacked  into  an
array holding up to 4 associated 4-vector particles for each smeared jet
```

chamber hit. Then the JHTL bank is decoded to determine the associated
@
PATR track (or 2 tracks). Finally each PATR track is considered in turn; its hits are examined and their 4-vector origins are histogrammed. The particle with the largest number of histogram entries is taken as the most likely cause of the PATR track. The particle number (encoded as 2 * particle_number + VECT_bank_number) is stored in the TR4V bank along with the number of hits that were associated with it.

Additionally, if there exists a second (or even a third) candidate 4-vector particle that has at least 8 hits associated (5 if the total number of hits for the track is less than 20), a second (and third) entry is made in TR4V for that PATR track.

```
        TR4V Bank Structure
        -------------------

    Name of bank   :  TR4V              (= Track_to_4-Vector Link)
    No. of bank    :  = latest PATR bank number
    Length         :  8 * (No._of_PATR_tracks) + 1  words

     Full-word          Contents        Comment
     ---------          --------        -------

         1               NTRKS       Number of PATR tracks
       2...9             NRES1       8 words with results for track 1
      10...17            NRES2       8 words with results for track 2
      18...25            NRES3       "   "     "     "     "     "  3
         .                 .           .      .      .      .
NTRKS*8-6...NTRKS*8+1    NRESn       8 words with results for last track
@
where NRESi stands for 8 words as follows:

        1       NAPART      No. of associated 4-vector particles
        2       NTOTHT      Total no. of hits along the PATR track
                            as recorded in the JHTL bank by PATREC
        3       IPART1      Encoded word for most probable 4-vector
        4       NHITS1      No. of smeared hits belonging to IPART1
        5       IPART2      Encoded word for 2nd most probable 4-V
        6       NHITS2      No. of smeared hits belonging to IPART2
        7       IPART3      Encoded word for 3rd most probable 4-V
        8       NHITS3      No. of smeared hits belonging to IPART3
```

The code used for the IPARTi words is (as for the HITL bank):

    2 * particle_number + VECT_bank_number

Subroutine MCTR4V resides on 'F11LHO.JADEGS/L'.

C) 4-Vector History Traceback

The PATR Traceback described in section B allows one to trace the origin of a PATR track to the VECT banks. (Pointers inside VECT/1 allow traceback to VECT/0.) However, it is often vital to know the 4-vector history of a particle in the VECT/0 bank. This is now possible using the new 'PALL' bank.
Changes have been made in the JADE Tracking Program MCJADE in order to create the PALL bank from information stored in the input 4-vectors.
@
This information is only meaningful if set up correctly by the 4-vector event generator. At the time of writing (October 1983), only the LUND 4.3 and 5.1 generators maintained by Alfred Petersen are able to output the required data but other generators could be modified to conform with the scheme to be outlined below.

JADE Computer Note 10 describes the format of JADE 4-vector events

that are acceptable to the tracking program. This is the so-called CPROD format, named after the COMMON block used in the tracking program. (Actually the /CPROD/ common in the tracking program defines the 'shape' and maximum size of the 4-vector records; the events themselves are always smaller, with no trailing zeros. The tracking routine BRVECT unpacks this compact structure into the /CPROD/ common.) The CPROD format is split into 2 parts, the first part for 'produced' particles and the last part for 'final' particles that are stable or meta-stable. It is the 'final' particles from an event generator that are tracked by MCJADE. Up till now, the 'produced' particles have simply been thrown away by the tracking program but, for the purposes of what might be called '4-vector history traceback', this is no longer so.

The 'produced' particles list of the CPROD format will now be known as the 'all' particles list and expanded from 30 particles to 500. As its name suggests, this list now provides space for storing information about every particle produced in the event simulation. When the events are tracked, this information is copied into a PALL bank in the same way as the 'final' particle information is copied into the VECT/0 bank. Since there is an (almost) identical entry in PALL for every entry in VECT/0, it is possible to associate a particle in the VECT/0 bank with one in PALL. Since it is also part of the scheme that each PALL particle has a pointer to its parent, a complete traceback of the history of any VECT/0 particle is thus possible.
@

```
      COMMON / CPROD / NEV, BEAM, PT, PHI, THETA, IFLAVR,
     +                 NP, NC, NN, PP(4,500), XM(500),
     +                 JCH(500), JTP(500), JP(500,2),
     +
     +                 NF, NCF, NNF, PF(4,300), XMF(300),
     +                 ICF(300), ITF(300), PSTRT(3,300)


NP        =  total number of 'all' particles
NC        =  total number of 'all' charged particles
NN        =  total number of 'all' neutral particles

PP(i,k)   =  i'th component of the 4-vector for the k'th  particle
XM(k)     =  mass in GeV of the k'th particle
JCH(k)    =  charge of the k'th particle
JTP(k)    =  type of the k'th particle (see below)
JP(k,1)   =  pointer to PARENT of the k'th particle in the 'all' list
JP(k,2)   =  order no. of parton parent for the k'th particle (see below)
```

Thus for each particle k, there is a so-called 9-vector of data incorporating the energy-momentum 4-vector.

Particle Types

Where defined, JADE types are used as in JADE Computer Note 10. Otherwise private types are used such as the LUND table. Although this is unsatisfactory in principle, no other solution yet exists. In practice this may not matter as particle masses are also available and the LUND table may well last for the lifetime of JADE at PETRA. (Currently the LUND 4.3 and 5.1 generators held by Alfred Petersen use (1000 + LUND_type) * ISIGN or (JADE_type) * ISIGN, whichever is
@
appropriate. ISIGN is +ve for particles and -ve for anti-particles. Since the type is signed, only the absolute values should be used when comparing entries in the VECT/0 and PALL banks.)

Parent Pointers

JADE Computer Note 10 says that JP(k,1) is a pointer to the daughter particle in the 'final' list. In this new scheme, it is now a pointer to the parent particle in the 'all' list. This follows the LUND simulation scheme and is more useful in practice anyway. It is a convention that the parent number of the 'first' particle(s) is zero. There could be more than one 'first' particle of course but what type this particle(s)

has is not fixed in this scheme. It could be an initial electron or positron (as in the LUND generator) but does not have to be.

Parton Parent (Order Number)

JP(k,2) used to be the decay multiplicity but has now been redefined to be the parton parent number. However note that this does not give the flavour of the parton. It can however be used to make a fast 'jet' assignment. The first parton is assigned number 1, the second (which may be a gluon) is assigned number 2 and so on. The flavour of an event (where meaningful) can be found in IFLAVR in /CPROD/ and in the 9th word of the PALL and VECT/0 banks. It is another convention that the parton number of the partons is -100 and the initial e+, e-, virtual photon (or virtual Z) have a 'parton number' of -1000. Radiated photons in the initial state have a 'parton number' of -2.
@

```
          PALL Bank Structure
          -------------------

     Name of bank   :   PALL                    (= All Particles)
     No. of bank    :     0
     Length         :  L0 + L1 * NP   words


     Full-word        Contents              Comment
     ---------        --------              -------

         1               L0          Length of header = 9 words
         2               L1          Number of words per particle = 9
         3             IEVNT         Event number
         4               NP          Total number of 'all' particles
         5               NC          Number of 'all' charged particles
         6               NN          Number of 'all' neutral particles
         7              PHI          Phi of event axis      (if relevant)
         8             THETA         Cos Theta of event axis( "      "    )
         9             IFLAVR        Flavour of event       ( "      "    )
      L0 + 1           9VECT1        9-vector of 1st particle  (9 words)
   L0 + L1 + 1         9VECT2        9-vector of 2nd particle  (9 words)
  L0 + 2*L1 + 1        9VECT3        "    "     " 3rd    "     ("    "  )
         .               .              .       .     .         .
         .               .              .       .     .         .
  L0 + L1*(NP-1)+1     9VECTn        9-vector of last particle (9 words)
```

where 9VECTi stands for the 9-vectors as explained above:
  px, py, pz, E, m, charge, type, parent_number, parton_parent_number
@
Notes
-----

a) The first 5 components of P9VECTi are REAL*4 while the other 4 are INTEGER*4. This is different from the output of MCHTRB.

b) If a quark or anti-quark appears in the chain, its electric charge value will probably be undefined.

c) The PALL bank should not be confused with the subroutine PALL in the BOS library.

d) For Multi-hadronic events, there has been an unofficial practice of storing the partons at the end of the VECT/0 bank with an illegal type code, e.g. -100,-101 etc. This is now obsolete with the new PALL bank but may still happen for the time being. However this practice may cease in the near future. Another unofficial feature which may persist is the allocation of a sign to the IFLAVR word in VECT/0 to indicate whether the quark or anti-quark is stored first in the bank. A negative value implies the anti-quark comes first.

e) Subroutine MCTRCB calls subroutine SCTR4V. Both are stored on the member MCTRCB on F11LHO.JADEGS/L.

I would like to acknowledge the contribution of Karl-Heinz Hellenbrand to the design and testing of the traceback scheme outlined in this note.

Copies of this JADE Computer Note can be obtained by the following NEWLIB command: SUB 'JADEPR.TEXT(JADECN69)'

▸ ausschließlich relevant für MCJADE. (siehe MCVAL1)

1 = γ
2 = e
3 = μ
4 = π
5 = K
6 = N