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

c. For each conversion:

1) Refit of electron - tracks in r - z with conversion vertex - constraint

2) Cuts

3) Photon fit with vertex - and invariant mass - constraint

4) Creation of a "PHOT" - bank

2. Options

Pattern recognition (rather time consuming), z - refit and photon fit are subject to the user's wishes and can be switched on/off in USCOGAM with the flags LPAT, LZF, LFIT. Default is LPAT, LZF, LFIT / 0, 1, 1 /.

```
LPAT = 1 / 0     perform modified PATREC  /  NO new PATREC
LZF  = 1 / 0     do the z - refit  /  do NOT the z - refit
LFIT = 1 / 0     do the photon fit  /  do NOT the photon fit
```

In addition to the parameters for the vertex search on BLGEO, there are cuts made in the user routine USCOGAM, which the user may change according to his needs.

The standard cuts applied are (for a description of the variables see "PHOT" - bank):

```
| DXY   | < 12.0 mm
| SDXY  | <  6.0
| SDPHI | <  4.0
| SDZ   | <  3.0
| SDTH  | <  3.0
        R < 50.0 mm
       SR > -5.0
      APV <  0.2
       XM < 30.0 MeV + EC / 50.0
momentum dependent cuts on # hits for electron tracks:
  # hits >  8    for                P <   75.0 MeV
  # hits > 16    for   75.0 MeV < P < 1000.0 MeV
  # hits > 25    for 2000.0 MeV < P
```

3. CPU time: 600 multihadron - events / minute with and 1500 multihadron - events / minute without pattern recognition.

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
IDATA(I + 25) = K2	in "PATR" bank
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	

Examples

The pointers KP1 and KP2 to the electron tracks K1 and K2 in the "PATR" - bank are given by the statements:

```
IP  = IDATA( IBLN( 'PATR' ) )
L0  = IDATA( IP + 1 )
LTR = IDATA( IP + 3 )
KP1 = IP + L0 + (K1-1) * LTR
KP2 = IP + L0 + (K2-1) * LTR
```

A loop over all "PHOT" - banks of an event could be (I is the pointer to the current bank):

```
      I = IDATA( IBLN( 'PHOT' ) )
101   CONTINUE
      IF( I .EQ. 0 )  GO TO 199
      .......
      .......
      I = IDATA( I - 1 )
      GO TO 101
199   CONTINUE
```