Usage of the fillwise runvertex determined with the vertex chamber

The subroutines needed are located on F22KLE.JVTXC.S/L. The runvertices are read from a sequential dataset. For this a FORTRAN unit 'nn' must be supplied:

```
//* In your JCL member :
//GO.FTnnF001 DD DSN=F22KLE.JVTXC.DATA(RUNVERTV),DISP=SHR,UNIT=FAST

C In your FORTRAN member :
C before the event loop, to initialize
      CALL VTXCRJ(nn)

C for each event, run or fill
      CALL VTXCRW( NRUN, RVX, RVY, RVDX2, RVDY2 )
```

The input to VTXCRW is the run number NRUN. RVX and RVY are the coordinates of the runvertex. RVDX2 and RVDY2 are the squares of the beam sizes.

G Eckerlin 14/12/87

TPed MH-Events at DESY

1) ALL MH-Datasets reanalysed with TP-Program from 12/86 are available under:

F11ECK.URZ.xxx

with xxx = TPM... for 1979-1984 Data

and xxx = G_____G____ for 1985 and 1986

- PATR/8 is created with Steffens r-φ and z-s Fits
- TP-Banks are taken from PATR/8
- Details see JCN83

List of Datasets see:

JADEPR.TEXT($MHTPLOG)

GE 14/12/87

2) 1986 Data were reanalysed with TP-version from 7/87 using J. Spitzers r-φ and z-s-Fit to create PATR/8 (replacing Steffens Fits) see JCN 94 & JCN 95 for details.

Name of datasets:

F11ECK.URZ.TP787.xxx

xxx = (see 1))

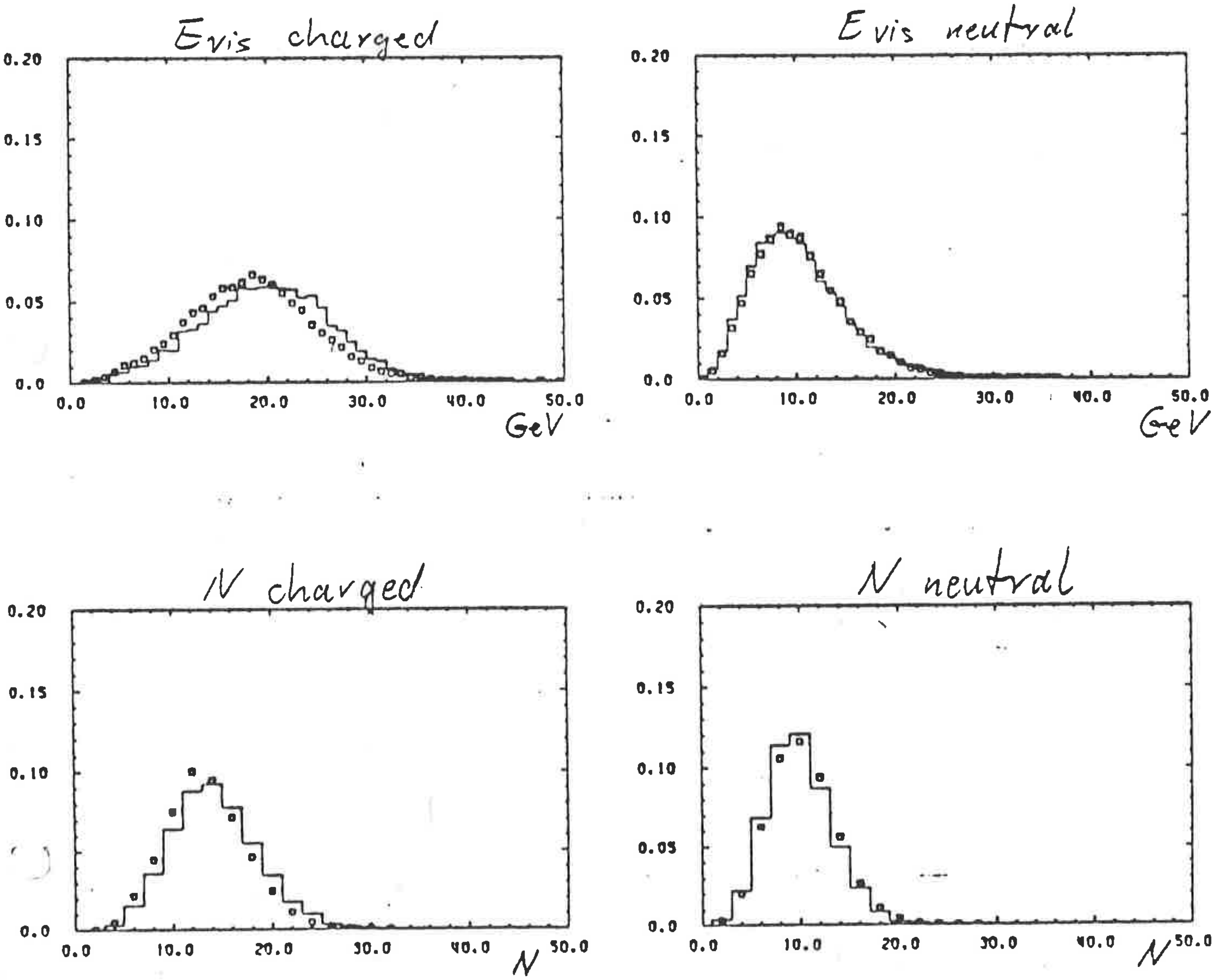
3) 1984 & 1985 MH-Events were TPed with a Version from 9/87 using z-chamber hits in addition in the z-s Fit (see JCN 95/1)

Datasetname : F11ECK.URZ.TP787.xxx

4) 1986 MH-Datasets with PATR/7 created with the Vertexchamber Software are stored in F.22HAG.TAPE.URZNEU.xxx

GE 14/12/87

Compare of : — MH '86
and ○ MH '82



↳ 1986 data has ~2 charged tracks more

GE 14/10157

MC-Events at DESY

1) LUND 5.2 and WEBBER-MC with different parameter settings done in Heidelberg are stored under
F11BET. MHTPxx. . . .   xx = CM-Energy.
Details see F11BET. MCINFO (FILEINFO)
Generator source Lib : F11BET. LUND. S

2) LUND 5.2 created at RAL are available under F22CHR. MHECM xx.
INFO in F22CHR. INFO(MCMHRAL)

All events are tracked with the bugs reported in JCN87

ALL INFO-Files are now copied to JADEPR.TEXT. Member MCTPOIR gives more information

GE 14/12/..

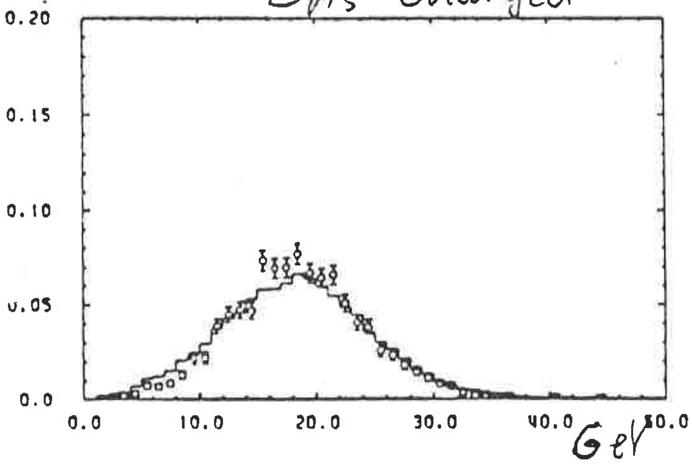
Compare of — MH '82

and o LUND5.2

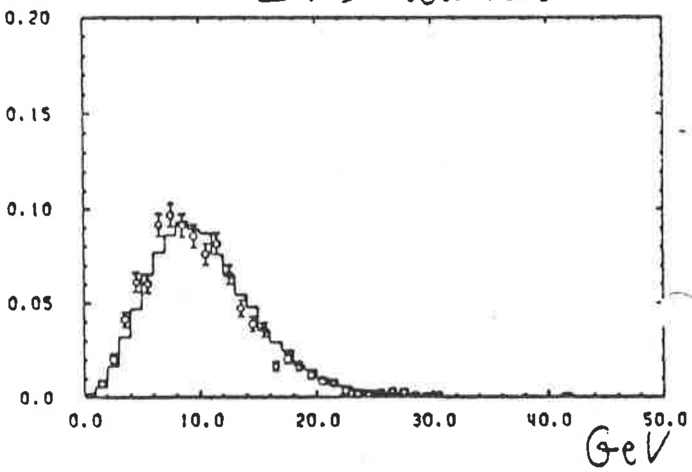
1982 tracking

generated at HD 1984

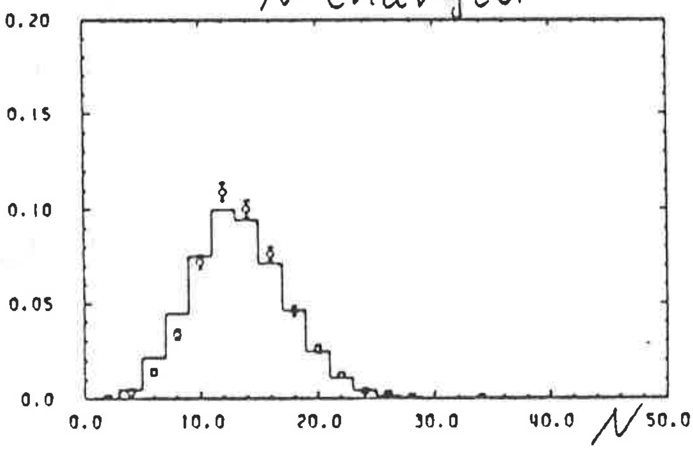
E_{vis} charged



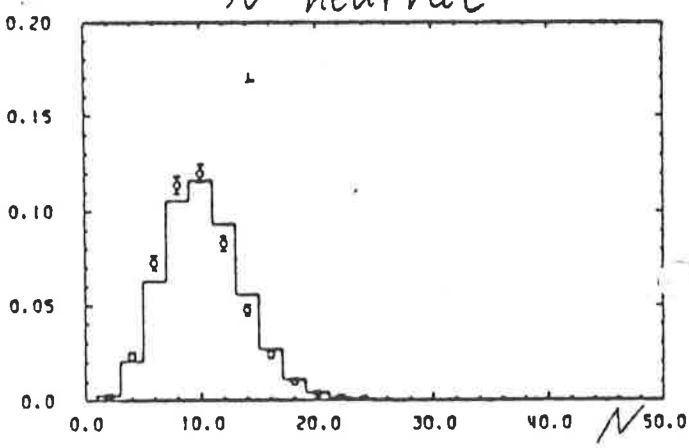
E_{vis} neutral



N charged



N neutral



→ LUND 5.2 tracked 1984 (with the bugs reported in JCN 87) describes the data well.

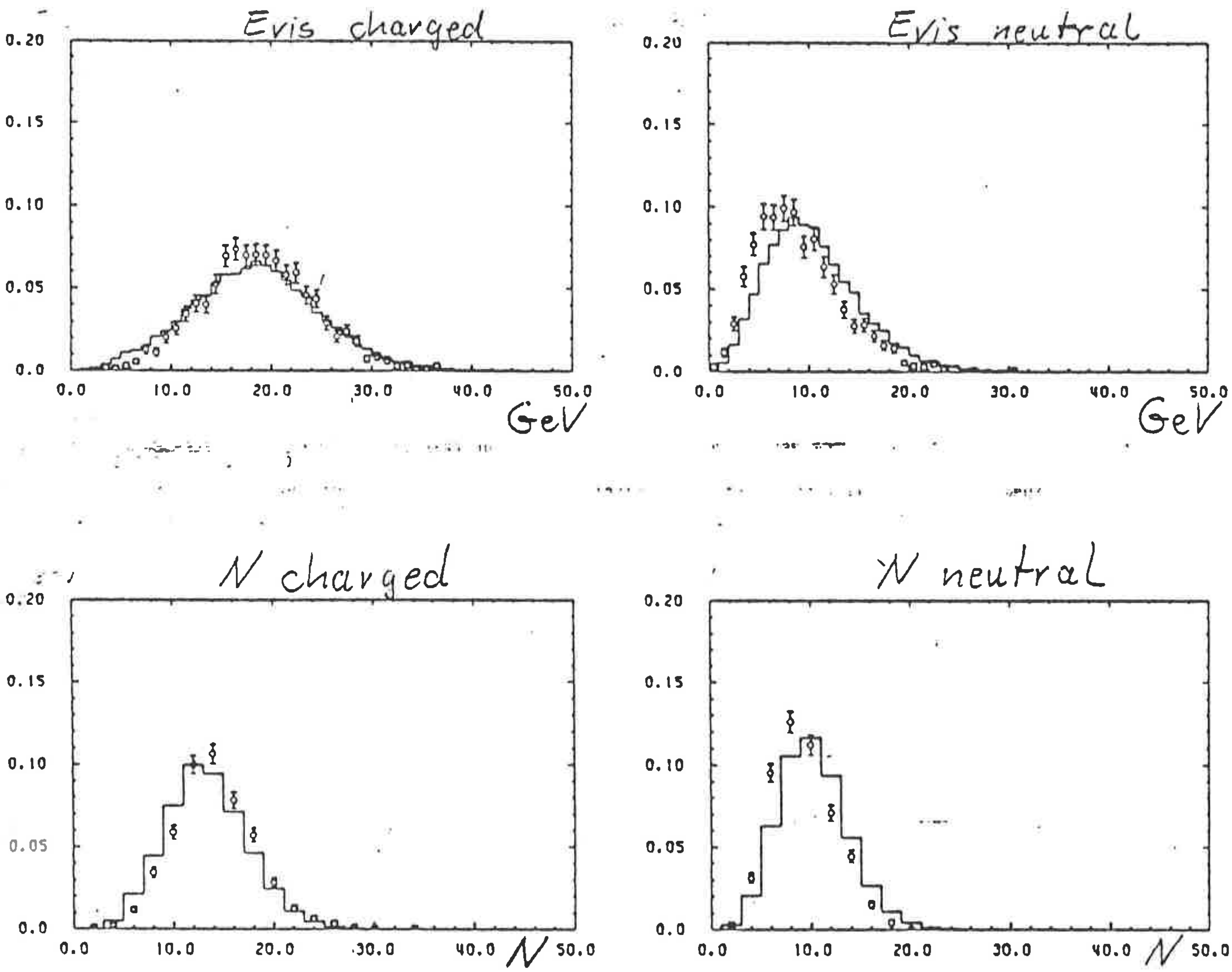
GE 29/10/84

Compare of : — MH '82

and ○ LUND5.2

1982 tracking (Version /57,

generated at Heidelberg



→ LUND5.2, but tracked with the ~~corrected~~ routines

is a little bit worse, especially the

neutral Energy and Multiplicity changes.

GE 14/12/87

3) LUND6.3

Generator source library with the JADE-Interface to the Tracking is

F11ECK.LUND63.S

original library: F11BET.LUND63.S

Tracked and TPed events will be at DESY around Jan 88.

The datasets will be named

F11ECK.URZMC.xxx

The datasetlist will be placed in

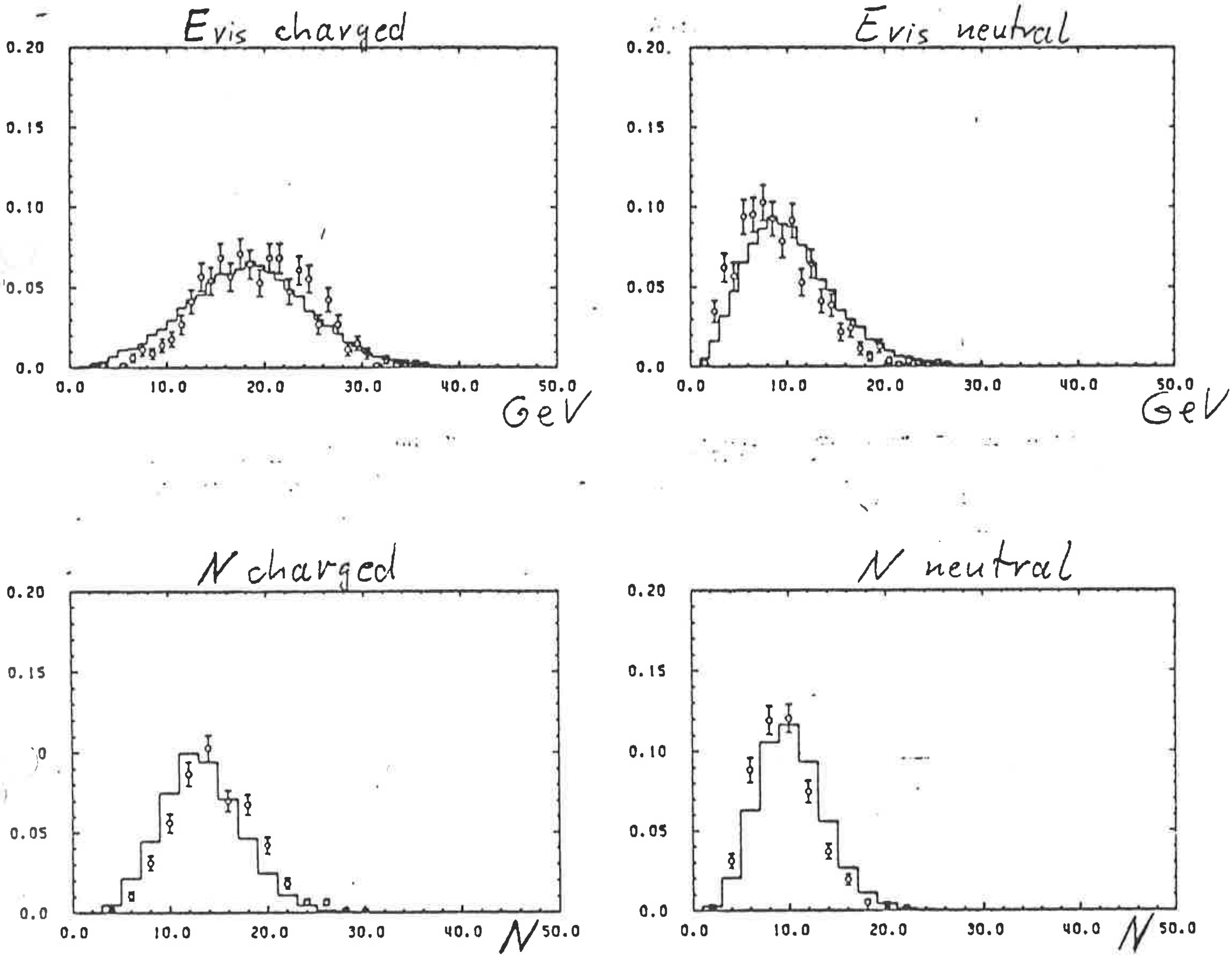
JADEPR.TEXT(MCTPLOG).

The DIRECTORY-file MCTPDIR gives a list of all MC-Infofiles in JADEPR.TEXT.

GE 14/1218

Compare of — MH '82

and o LUND6.3 shower
1982 tracking
generated at RAL



The agreement of the distributions from ~2000 MC-Event generated at RAL with the data gives no evidence for a bug in the Tracking or in the LUND6.3-Generator.

Status LUND6.3

At Heidelberg MC-Events will be generated for:

- 35 GeV 1986 detector ~50000 events
- 35 GeV 1982 detector ~50000 events
- 44 GeV 1985 detector ~25000 events

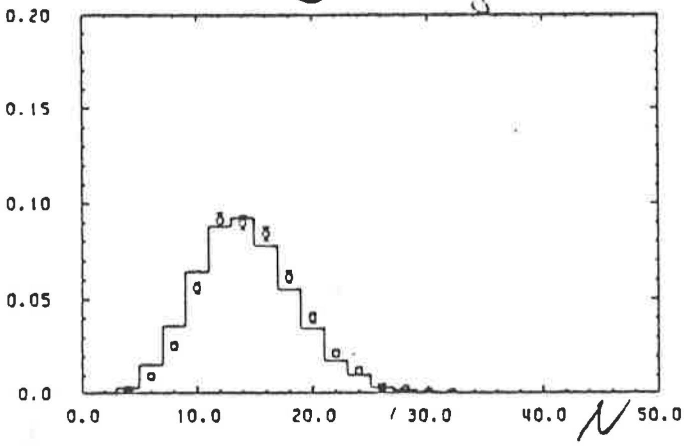
all generated with standard settings for LUND6.3 (except minor switches, see infofile for details)

A short comparison of ~5000 MC events with 35 GeV 1986 detector and ~10000 real MH-events from 1986 is shown in the appended figures

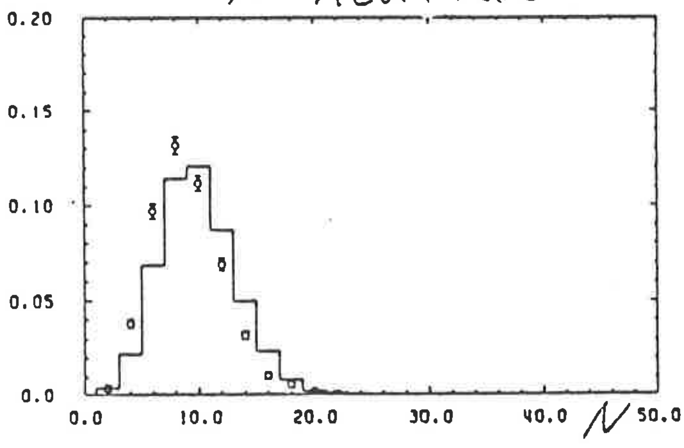
GE 14/12/87

comparing of – MH'86
and o LUND6.3, 1986 tracking

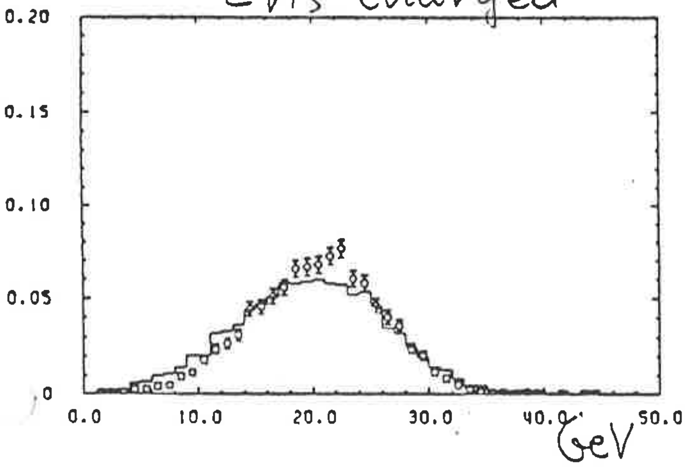
N charged



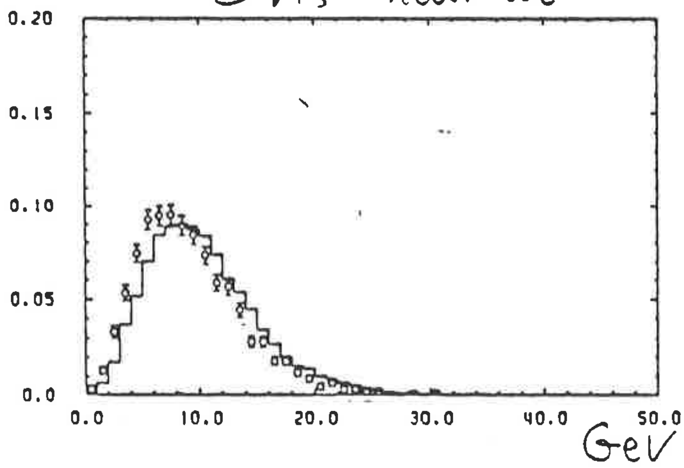
N neutral



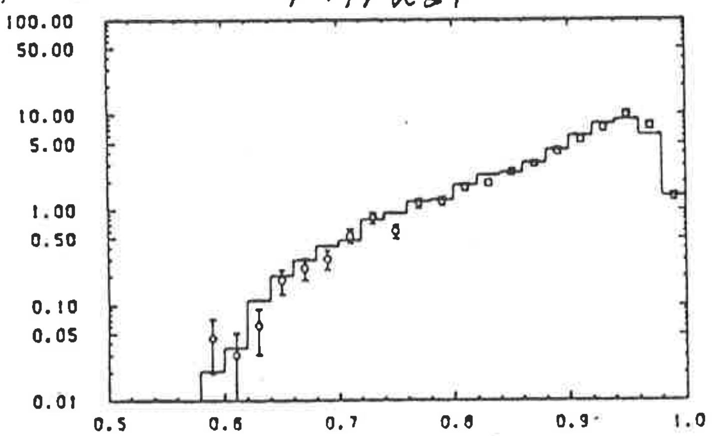
Evis charged



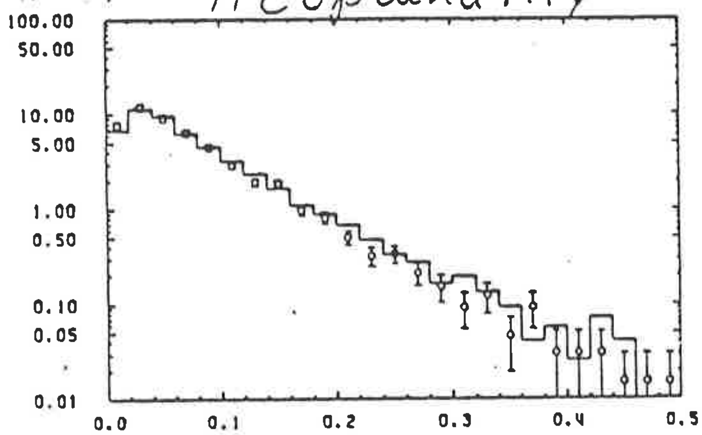
Evis neutral



Thrust



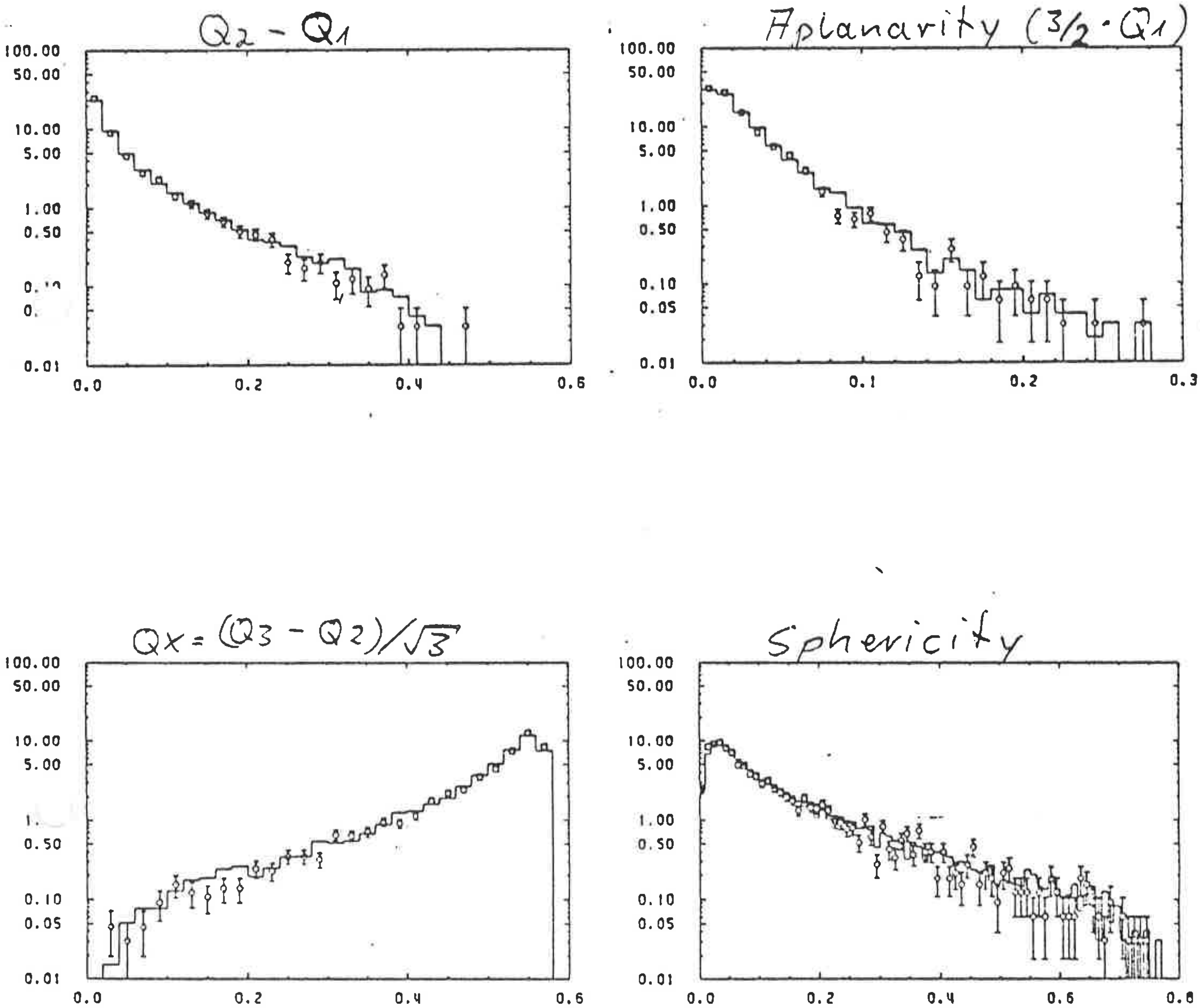
Acoplanarity



GE 14/12 1987

— MH '86

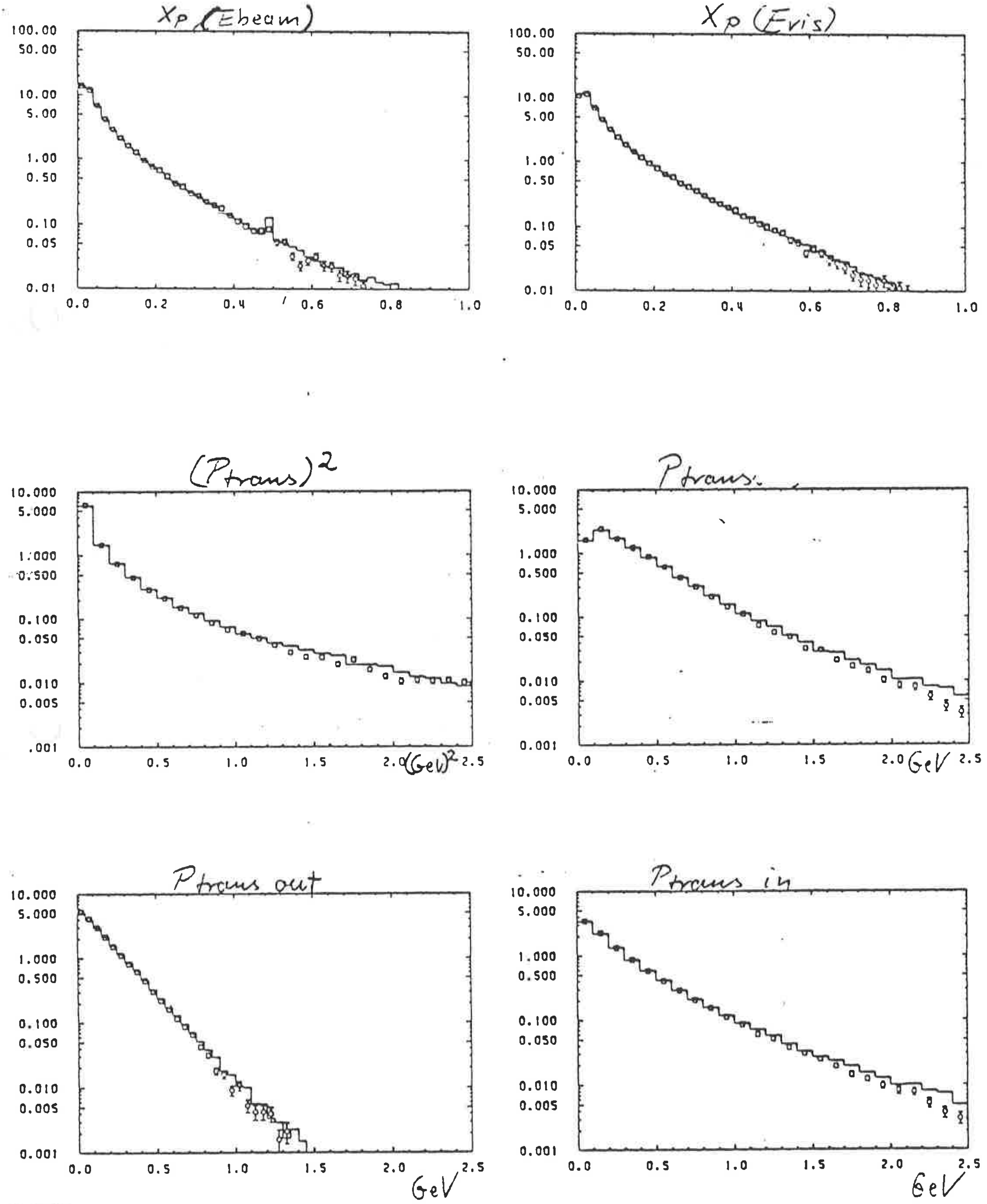
o LUND6.3, 1986 tracking



GE 14/12/87

— MH'86

o LUND6.3, 1986 tracking

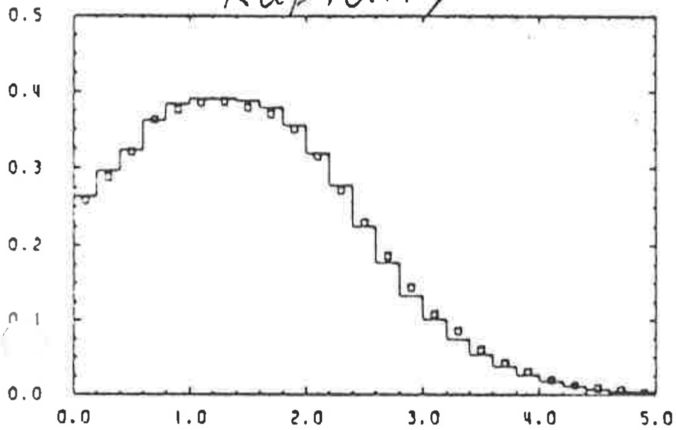


GE 14/12/87

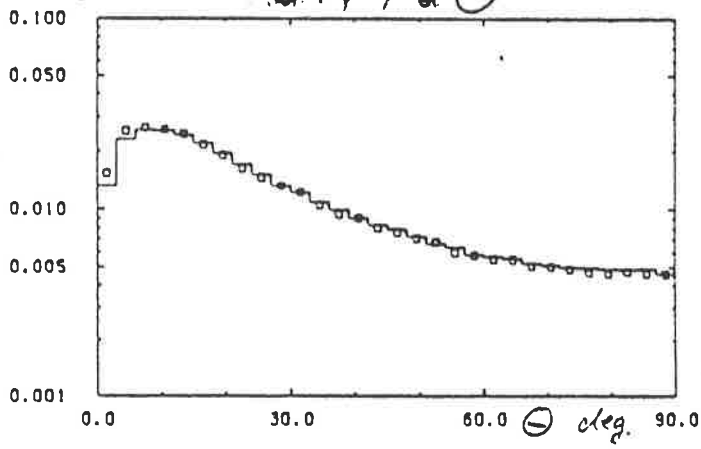
- MH '86

○ LUND6.3, 1986 tracking

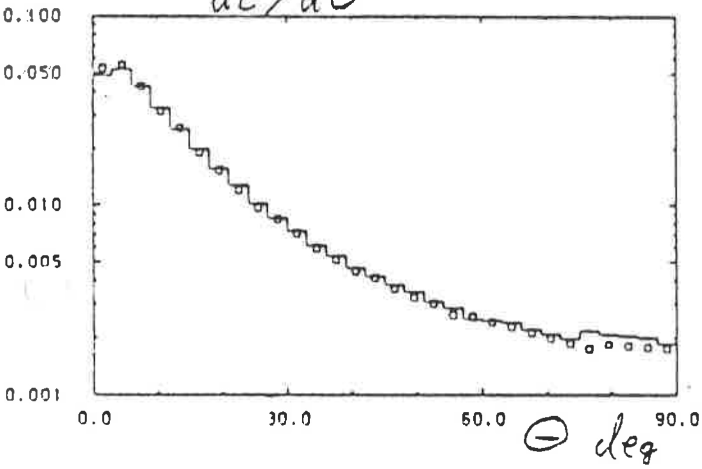
Rapidity



dN/dΘ



dE/dΘ



TP9 Status

14/12/87

- All the important routines are written and integrated.
- Most have been tested
- Bugs in TP8 have been corrected in TP9
  eg missing momentum had wrong direction
- Structure Chart drawn (best way of understanding the program)

To be done this week

- input options processor and options 'server'
- diagnostics code
- print out to be finalised
- testing and TP bank checking

Further work

- vertex chamber
- errors on direction cosines (charged tracks)

C. Bowdery
13/12/87

Which FORTRAN Compiler?

DESY

Language:	JADE uses:	DESY recommends:
FORTRAN 66	(IBM) HXE	(IBM) VS FORTRAN2 option LANGVL(66) or (IBM) HXE
FORTRAN 77	(Siemens) FORTRAN77	(IBM) VS FORTRAN2 LANGVL(77) → default

RAL

	JADE must use:
FORTRAN 66	(IBM) VS FORTRAN2 LANGVL (66)
FORTRAN 77	(IBM) VS FORTRAN2 LANGVL (77) → default

What is the problem?

LOAD modules are routinely copied from DESY to RAL (not source)

HXE load modules are compatible with RAL-compiled FORTRAN (from VS(2) compiler)

Siemens load modules are incompatible in principle. By maintaining a library of Siemens FORTRAN 77 functions, we can get programs to run. **THIS IS NOT IDEAL!** There could be problems in the future.

Proposal

Change from Siemens FORTRAN 77 to IBM VS FORTRAN(2) at DESY. Re-compile existing FORTRAN 77 members.

Consequences?

- Heidelberg will be affected (if Siemens load modules are copied from DESY) until VS FORTRAN (2) is available there.
- FORTRAN4 HXE compiled version of CERNLIB must be used until VS version available. (NAGLIB too.)

How much effort to change?

Since Siemens and VS FORTRAN load modules should not be mixed, changeover should occur all at once.

1) Members need changing:

Line 0 : from FORTRAN77
to FORTVS or FVS

Use JADE CLIST called REPLAC on a PS

eg define target line with string MEMBER & NAME
'change' string FORTRAN77
'replacement' string FORTVS

2) Re-compile members

Several options

a) COLOAD line commands on the LE screen.

b) COMP.PS JADE CLIST: (prompts for userid, for

c) Use Batch NEWLIB (when modified)

Link with: // EXEC NEWLIB PS= ...., PL= ...., FORTLIB='SYS1.FORT.LFVS'

d) Use COLOAD line commands on DIR screen

e) Other methods?

3) For new members in FORTRAN 77?

Use LANGUAGE FORTVS command (if necessary)
before creating new members

or

Use I Ø FORTVS on a member already created

4) JCL changes (and NEWLIB LINK page)

Change: JFORT → VFORT or FORT : check parameters with HELP PROCS

When should the change be made?

after full consultations with affected DESY + HD users (UK-based personnel and Maryland personnel probably not directly affected.)

- sooner rather than later
- during a 'quiet' period (batch jobs could be affected)

Who will do the change?

- dE/dx routines on JADEGS ?
- new TP 9 library (every member) Chris Bowdery
- old TP 8 library (a few members) Chris Bowdery
- JADE private code owners

Safety? Make back-ups!

JADE CLIST (SEARCH) can be used to check all standard libraries contain no remaining FORTRAN77 (ie Siemens) members. Could also check that all FORTVS members have been compiled. (?)

Too much bother?

- necessary for RAL
- Siemens FORTRAN 77 will disappear one day.
  Easier to make change now before more FORTRAN 77 language programs are written.

Format of bank ZEHD
can be found in zlib/ze4vpk.for
(J.O. 12.6.2008)

JADE Computer Note No. 99

A new compact Data-Format
Description of Bank ZE4V

G. Eckerlin, M. Zimmer

26. Januar 1988

Introduction A new compact Data-Format was developed in order to allow a fast access to all JADE-Multihadrons. For several analyses the complete event-record in all its details is not needed, and so it is often cumbersome to look through all JADE Data tapes to pick out only a few numbers. Although the new format was originally developed for electron analyses, it is general and detailed enough, that it can be used also for other applications.

The structure of the BOS - Bank ZE4V The event information is stored in a BOS Bank named ZE4V. Thus, the bank can be written out easily with the full event record. In the standard application, however, one creates a pure ZE4V dataset. The structure of the bank is as follows :
A global header part, which contains information about general event properties, is followed by the track section, where for all charged particles and photons details about momentum, energy, leadglass-clusters, dE/dx etc. are given. The track section is divided into two parts. The *general track section* (first 9 words), which contains mainly the fourvector and charge, is identical for charged particles and photons, while the following words (12 for charged particles and 5 for photons) are specific (*specific track section*). This format was chosen to accomodate even the minimal information of pure MC-fourvectors alone, where one can fill only the general track section and omit the additional words. To distinguish between the different filling modes (fourvector or photon or charged) the last halfword in the general track section is used (see example of a simple unpack routine).

$$\left.\begin{array}{l}\mathrm{RW}(\;\ldots+1)\\ \quad\ldots\\ \mathrm{RW}(\;\ldots+\mathrm{LH})\end{array}\right\}\textit{header section}$$

$$\left.\begin{array}{l}\left.\begin{array}{l}\mathrm{RW}(\;\ldots+\mathrm{LH}+1)\\ \quad\ldots\\ \mathrm{RW}(\;\ldots+\mathrm{LH}+\mathrm{LT})\end{array}\right\}\textit{general track section}\\ \left.\begin{array}{l}\mathrm{RW}(\;\ldots+\mathrm{LH}+\mathrm{LT}+1)\\ \quad\ldots\\ \mathrm{RW}(\;\ldots+\mathrm{LH}+\mathrm{LT}+\mathrm{Lch}\;(\mathrm{Lph}))\end{array}\right\}\begin{array}{l}\textit{specific track section}\\ \text{length for charged} = \mathrm{Lch}\\ \text{length for photons} = \mathrm{Lph}\end{array}\end{array}\right\}\begin{array}{l}\textit{track section}\\ \text{repeated for all particles}\end{array}$$

Contents of the ZE4V-Bank The header section contains information about the structure of the bank and quantities, which refer to the whole event. The first words give the sizes of the different sections of the bank. This is followed by run and eventnumber and basic event-quantities like thrust, sphericity and acoplanarity. The next words contain the coordinates of the main eventvertex from Bhabha calibration. In addition secondary vertices can be stored for analyses with reconstructed particles. The last section of the header is only filled for MC - events and holds the flavour and the fourvectors of the generated partons. The header is followed by the track section. For charged

Available ZE4V Datasets

All JADE Multihadrons are available in this data-format on two tapes. One tape contains the Multihadrons of the years 1979 - 1985, the other the 1986 data. The DS names are :

F22ELS.ZE4VPK.V987.DATA7985

F22ELS.ZE4VPK.V987.DATA86

The same events are presently also available on the following MSS datasets ordered in terms of energy. These datasets will be kept up to date, in order to have always the newest version available on MSS. The DS names give the version (V987 = version Sep. 87) , the energy range (E3237 = 32 to 37 GeV), and the year of data taking (DAT7985A = 79 to 85 Data / set A).

F11ECK.ZE4V.V987.E1020.DAT7985A	with	3781	events
F11ECK.ZE4V.V987.E2026.DAT7985A	with	3119	events
F11ECK.ZE4V.V987.E2632.DAT7985A	with	1383	events
F11ECK.ZE4V.V987.E3237.DAT7985A	with	6604	events
F11ECK.ZE4V.V987.E3237.DAT7985B	with	6468	events
F11ECK.ZE4V.V987.E3237.DAT7985C	with	6341	events
F11ECK.ZE4V.V987.E3237.DAT7985D	with	6264	events
F11ECK.ZE4V.V987.E3237.DAT7985E	with	6195	events
F11ECK.ZE4V.V987.E3237.DAT7985F	with	6238	events
F11ECK.ZE4V.V987.E3237.DAT7985G	with	4890	events
F11ECK.ZE4V.V987.E3739.DAT7985A	with	3856	events
F11ECK.ZE4V.V987.E3739.DAT7985B	with	1939	events
F11ECK.ZE4V.V987.E3947.DAT7985A	with	3965	events
F11ECK.ZE4V.V987.E3947.DAT7985B	with	3751	events
F11ECK.ZE4V.V987.E3947.DAT7985C	with	3995	events
F11ECK.ZE4V.V987.E3947.DAT7985D	with	4006	events
F11ECK.ZE4V.V987.E3947.DAT7985E	with	3952	events
F11ECK.ZE4V.V987.E3947.DAT7985F	with	3125	events
F22ELS.ZE4V.V987.E35.DAT86A	with	16040	events
F22ELS.ZE4V.V987.E35.DAT86B	with	13393	events

Track Section

NPTR : *see unpack routine for definition*

General Track Section		
RW	(NPTR+ 1)	e_x
RW	(NPTR+ 2)	e_y Direction cosines of track at closest point to (x_v, y_v, z_v)
RW	(NPTR+ 3)	e_z
HW	(2NPTR+ 7)	0 or PATR number of PHOT partner (conversion)
HW	(2NPTR+ 8)	Particle code (JADE MC convention)*100 + identified type
HW	(2NPTR+ 9)	0 or Vertex number of particle origin
HW	(2NPTR+10)	0 or number of secondary vertex (if any)
RW	(NPTR+ 6)	P_{tot}
RW	(NPTR+ 7)	Charge
HW	(2NPTR+15)	0 or particle number in PALL Bank (MC only)
HW	(2NPTR+16)	Traceback code (MC only)
HW	(2NPTR+17)	0 or number of TPTR bank
HW	(2NPTR+18)	Filling mode flag for this track : -1 = LT words filled (4-vectors) 0 = LT+Lph words filled (photons) 1 = LT+Lch words filled (charged) 2 = LT+Lpr words filled (private)

Specific Track Section for Charged Particles (additional for charged particles)		
RW	(NPTR+ LT + 1)	Cluster-energy of connected LG-cluster (GeV)
RW	(NPTR+ LT + 2)	Error of cluster-energy
RW	(NPTR+ LT + 3)	0 or corrected cluster energy (K.H.Meier)
HW	(2NPTR+2LT + 7)	Detector flag (TP convention)
HW	(2NPTR+2LT + 8)	No. of 1. connected LG-cluster + 100*(No. of 2. LG-cluster)
HW	(2NPTR+2LT + 9)	Track number in PATR - Bank
HW	(2NPTR+2LT +10)	Number of hits in $(r - \phi) * 100$ + hits in $r - z$ vertauscht!?
RW	(NPTR+ LT + 6)	x_{track}
RW	(NPTR+ LT + 7)	y_{track} closest point to (x_v, y_v, z_v)
RW	(NPTR+ LT + 8)	z_{track}
RW	(NPTR+ LT + 9)	dE/dx
RW	(NPTR+ LT +10)	$\sigma_{dE/dx}$
RW	(NPTR+ LT +11)	R_{min}
IW	(NPTR+ LT +12)	Muon Quality or unused (= 0)

Specific Track Section for Photons (additional for Photons)		
RW	(NPTR+ LT + 1)	Cluster-energy in GeV
RW	(NPTR+ LT + 2)	Error of cluster-energy
RW	(NPTR+ LT + 3)	0 or corrected cluster energy (in future)
HW	(2NPTR+2LT + 7)	Lg-detector part 0=barrel or +/- 1 = +/- z endcap
HW	(2NPTR+2LT + 8)	No. of LG-cluster
HW	(2NPTR+2LT + 9)	Number of (LG-blocks - connected tracks)
HW	(2NPTR+2LT + 10)	Number of LG-blocks

JADE Computer Note No. 100
Vertex Chamber Software

J. Hagemann, C. Kleinwort, R. Ramcke

February 15, 1988

Introduction The vertex chamber software was been modified in accordance with the suggestions made at the last JADE software meeting. [1] All the necessary vertex chamber calibration data and run vertices are now on the standard JADE calibration files `AUPDAT1` and `BUPDAT1`. The vertex chamber software can be found on **'F22KLE.JVTXC.GS/GL'**, which is on the JADE volume `STOR05`. The user has to call two subroutines (**VTXCSV**, **VTXCRV**).

VTXCSV(MODE,NBPFLT,NBPCFT,IRET)

This subroutine steers the whole vertex chamber pattern recognition and the combined fit. A short description of this routines can be found in JCN 98. `VTXCSV` has to be called in the event loop of the users main program *after* the calls to `KALIBR` and `INPATC`.

Description of input arguments:

`MODE`	-1	To get final statistics printout `VTXCSV` has to be called at the end of the program with `MODE` = -1
	0	Dummy call
	1	Perform vertex chamber pattern recognition
	2	Perform combined fit in the $R\varphi$-plane
	3	Perform pattern recognition *and* combined fit
`NBPFLT`		Number of `PATR` bank to be used for the vertex chamber pattern recognition. To get the lowest bank `NBPFLT` has to be set negative.
`NBPCFT`		Number of `PATR` bank to be used for the combined fit. To get the lowest bank `NBPCFT` has to be set negative.

[1] See JCN 98

VTXCRV(RVX,RVY,RVDX2,RVDY2)

VTXCRV has to be called after KLREAD or KALIBR and delivers the position (RVX,RVY) of the fill-wise run vertex determined with the vertex chamber. RVDX2 and RVDY2 contain the squares of the beamsize in the X and Y direction. If this fill-wise run vertex is not available (e.g. run number < 20000), the run vertex determined with the ID and default beamsizes will be returned. For Monte Carlo events all four return parameters are zero.

Remark The standard JADE graphics module supports neither the vertex chamber calibration, pattern recognition nor the combined fit!

Description of **VPAT** Bank

Header	1	header length (6)
$I*2$	2	number of selected PATR bank
	3	generation date *((year-1986)*1000 + day-of-year)*
	4	track data length (12)
	5	total number of tracks with ID-link from VTXC-PATREC step 1 and from PASS I + PASS II of step 2
	6	number of tracks with ID-link only from PASS I of VTXC-PATREC step 2
Track 1	1	ID-track number of first linked track
$I*2$	2	distance of first VTXC-hit from ID-track in units of μm
	3	distance of last VTXC-hit from ID-track in units of μm
	4	link code : (1) PASS I (2) PASS II (10) VTXC-PATREC step 1
	5	number of associated VTXC-hits
	6	hit number in VTXC bank for 1^{st} hit
	7	hit number in VTXC bank for 2^{nd} hit
	8	hit number in VTXC bank for 3^{rd} hit
	9	hit number in VTXC bank for 4^{th} hit
	10	hit number in VTXC bank for 5^{th} hit
	11	hit number in VTXC bank for 6^{th} hit
	12	hit number in VTXC bank for 7^{th} hit
Track 2	13	ID-track number of second linked track
$I*2$	⋮	...

Remark The hit numbers are *positive* or *negative*. The *sign* of the hit number corresponds to the selection of *hit* or *mirror hit*.

$R*4$	32	σ_z		
$I*4$	~~32~~	# of used hits in z-fit		
	33	cells with		
7	⋮	ID-hits		
	39	of track		
	40	copied from	see JADE-	
	⋮	selected	computer	
	48	PATR bank	note 12	
	49	$d_0 - R$ X_0	ID-	
	50	φ Y_0	parametrisation	
	51	weight for covariance matrix		
	52	cov_4: error from cov. matrix x^4		
	53	cov_3: error from cov. matrix x^3		
	54	cov_2: error from cov. matrix x^2		
	55	cov_1: error from cov. matrix x^1		
	56	cov_0: error from cov. matrix x^0		
	57	error from cov. matrix angular		
	58	σ of VTXC-line fit		
	59	# of VTXC hits used in COMFIT		
	60	extended VTXC-hitmask		
	61	extended VTXC-hitmask		
	62	COMFIT return code		
	63	bank generation number		9,10
	64	track number in VPAT bank		
	65	same as words 53–56,		
	⋮	but for the		
	68	ID parametrisation		
TRACK 2	69	...		
	⋮	...		

COMFIT return codes:

0	successful fit with vertex chamber
-1	successful fit without vertex chamber (forced)
-2	successful fit without vertex chamber (no linked hits)
-3	ID refit failed
-4	error in number of ID hits
-5	χ^2 too bad for combined fit
-6	ID refit failed
-7	ID refit failed
-8	track has no intersection with vessel wall
-9	radial momentum too low

/

10

Reconstruction of cluster energies in the barrel lead glass

JADE Computer Note 101

D. Pitzl

19.5.1988

Introduction

Some new routines for the reconstruction of photon energies in the barrel lead glass have been introduced in JCN 98 and where on the standard library JADELG.SOURCE since 22.12.1987. Here I give a summary of how to use these routines for real data and Monte Carlo, for photons and electrons and for different reactions like multi hadrons, τ-pairs, Bhabhas and 2-photon collisions.
The analysis programs for the lead glass detector have been described in JCN 14D. A flowchart is given in the appendix.
It must be remembered that everything explained in this note refers to the barrel lead glass only.

Correction of the lead glass calibration

The calibration of the JADE lead glass detector is done with Bhabha events (see JN 86). In events selected to reduce Bremsstrählung the observed cluster energy is adjusted to the 'expected' energy by introducing a correction factor for each lead glass block. The 'expected' cluster energy is calculated from the beam energy by taking into account the following effects (see JCN 35):

- The mean energy loss of electrons in the material between the interaction point and the lead glass surface ($1.23X_0$ at $\Theta = 90^o$) is subtracted in the routine ENGLOS.
- The efficiency for the collection of Čerenkov light on the photocathode depends on the incident angle of the electron (routine ANGBAR).
- Routine BRLGN has correction factors depending on the electron impact point within one block.

In this procedure the leakage of the electromagnetic shower through the back face of the lead glass blocks is not taken into account. The block depth of 30 cm corresponds to $12X_0$ in SF5 and $18X_0$ in SF6. The average fraction of leaking shower energy is shown in fig 1. Since the

- For electrons below beam energy there exists no correction function. Their leakage is smaller than the leakage of Bhabhas and also smaller than the leakage of a photon shower of the same energy (in fig 3 and 4 Bhabhas are compared to photons). The present solution is to call BBLEAK and use the photon leakage correction LKCORR for electrons as well. This gives systematically overestimated cluster energies and the error is energy dependent.

Reconstruction of cluster energies

A photon or electron which has an energy E_0 at the interaction point produces a lead glass cluster whose observed energy is given by

$$E_{obs} = (1 + r_{calib})(E_0 - E_{loss})(1 - r_{Leak})(1 - r_{Th})f_{Clce} \tag{1}$$

The various effects of energy loss and signal reduction are in general functions of the polar angle Θ (symmetric around 90^o), the particle energy E_0, the particle type and the time period.

r_{calib} (Θ, year) : neglection of Bhabha leakage in the calibration.

E_{loss} $(\Theta, E_0, \text{type, year})$: energy loss in the material between jet chamber and lead glass.

r_{Leak} $(\Theta, E_0, \text{type, year})$: shower leakage.

r_{TH} $(\Theta, E_0, \text{type, year})$: energy loss due to the lead glass readout threshold.

f_{Clce} $(\Theta, E_0, \text{type, year})$: Čerenkov light collection efficiency.

Since the correction factors depend on the incident energy this equation is solved for E_0 by iteration. This is done in the routine LGECOR which is called by LGCDIR. At this stage the calibration correction has already been applied to real data. Starting with the observed cluster energy as a first estimate for E_0 the particle is followed through the coil into the lead glass and all known effects of energy loss and signal reduction are taken into account. In the first iteration the resulting 'expected' cluster energy is in general smaller than the observed energy and the difference is added to the observed energy as a new estimate for E_0. The procedure is repeated until the difference between the 'expected' and the observed cluster energy is below 10 MeV. The cluster energy is stored in the LGCL bank, the block energies in the ALGN bank are not updated. Their sum (given by word #16 in the cluster data) is equal to the observed cluster energy after correction for the calibration.
Every effect has its own subroutine:

ENGLOS, ENLOSG : energy loss in the outer tank wall, z-chamber, TOF counters, coil etc. (1.04 X_0 at 90^o in 1986, see JCN 86) for electrons and photons respectively (see fig 2). The data for photons are from EGS4, the electron data are from the Tokyo group and were found to be in good agreement with EGS.

LKCORR : leakage of photon showers, values from EGS4 for 1.04 X_0 Aluminium and 30 cm lead glass SF5 or SF6 at 90^o (see fig 3 and 4).

Flowchart for the lead glass analysis routines mentioned in this note :

FORTRAN IV code on JADELG.SOURCE, compiled versions on JADELG.LOAD

Main program or SUPERVISOR

- LGINIT : block data for geometrical constants and cuts
- event loop
- LGCALB : lead glass calibration, creates ALGN bank from ALGL bank
- LGANAL : cluster search, creates LGCL bank
  - LGCLPB : process clusters in the barrel part
    - BBLEAK : calibration correction
- LGCDIR : links tracks to clusters, calculates direction cosines, writes onto LGCL bank
  - LGECOR : corrects photon and electron energies
    - ENGLOS, ENLOSG : energy loss of electrons and photons in coil
    - LKCORR : leakage of photon showers
    - THCORR : energy loss due to readout threshold
    - ANGBAR :angle dependence of Čerenkov light collection efficiency

next event

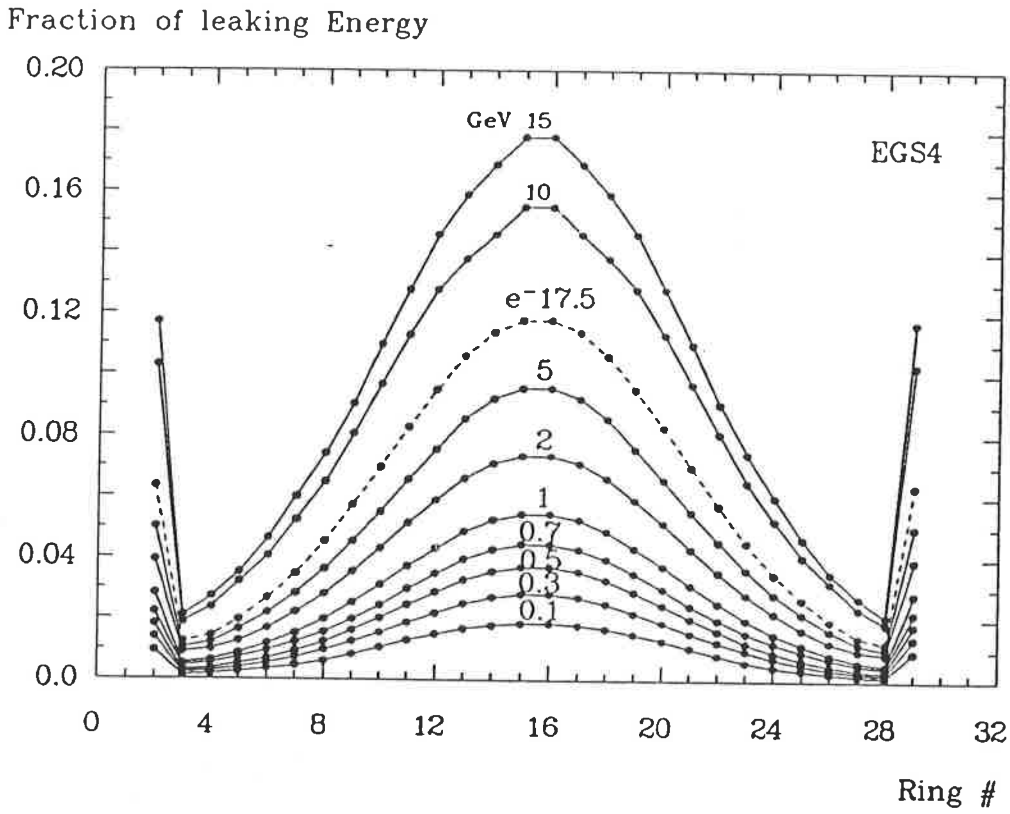


Figure 3: *Leakage of photon showers, 9 cm Al and 30 cm lead glass SF5*

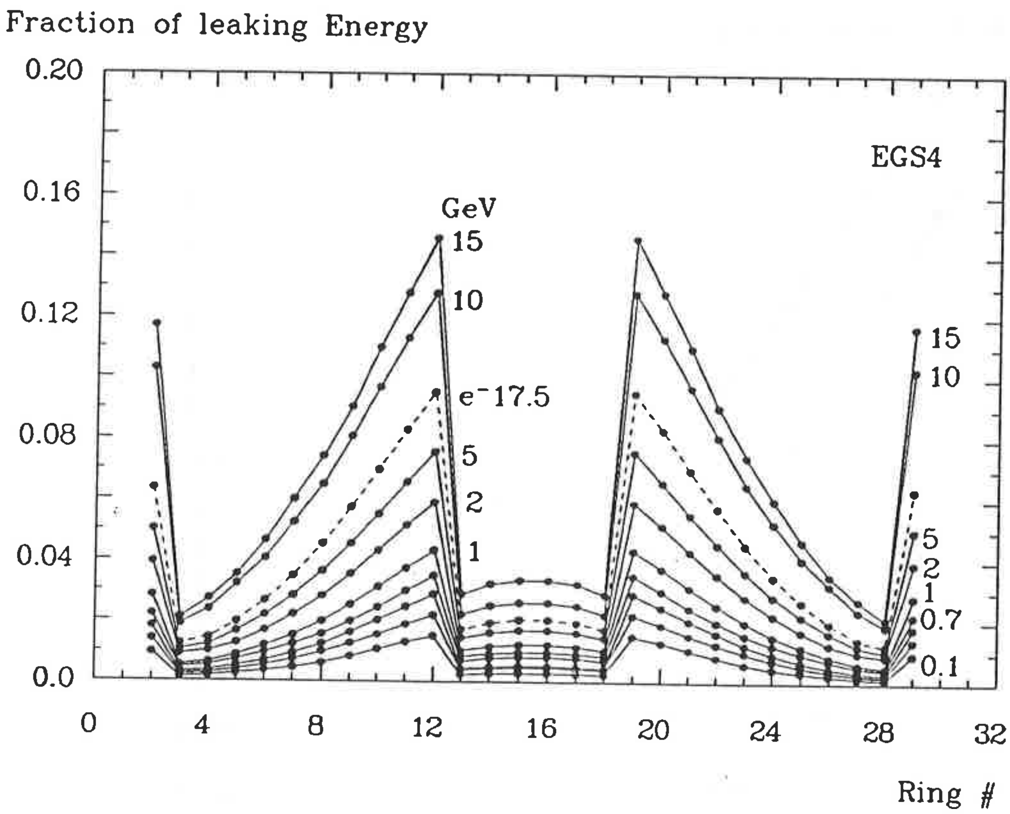


Figure 4: *Leakage of photon showers, SF6 in the 6 central rings*

JADE Computer Note 102

The New TP Program Version 9

C.K. Bowdery J.J. Pryce

August 11, 1988

1. Introduction

The TP program reconstructs and summarises JADE events, producing datasets containing a TPEV/1 bank (with global event information), a number of TPTR banks (each with details of one particle) and a number of TPVX banks (each with details of one vertex). The datasets will also contain raw and processed data banks as requested by the user.

The program has existed for many years. Version 8[1] was released in 1984 and has been the standard version up till now. It is documented in JCN 79 with the TP Banks being documented separately in JCN 80. A special Heidelberg version 8+ has been used recently, since version 8 has been virtually frozen during the development of a new version, which is the subject of this note.

Version 8 is a direct descendent of earlier versions of the program originally written by S. Yamada and E. Elsen. It was originally intended to be the ultimate version but various shortcomings in the program structure has made it unsuitable for adding new sections. Instead the program has been almost completely re-written along radically different lines.

The new program, to be known as Version 9 or TP9 for short, follows the philosophy of Structured Design (part of SASD) in consisting of a very large number[2] of small, 'weakly coupled' subroutines with 'strong cohesion'. However the nature of JADE programs, with the event data in a BOS common block, has meant that the

[1] The version numbers correspond roughly to the generation number of the multihadronic events the program produced.

[2] About 140, not counting the standard JADE packages called.

implementation is far from pure SASD with its afferent and efferent data flows. Nevertheless, it conforms to most of the 'design heuristics' that ensure reliability and ease maintenance problems.

Since it is a feature of SASD that a design can be evaluated using a number of graded criteria, it should be possible to evaluate the quality of TP9. Comments and criticism in this area (and all others) are thus welcome.

2. The Nature of the Program

The program normally runs as a batch job, reading events in from stream 1 and writing out 'accepted' events on stream 2 and 'rejected' events on stream 3. Whether events are rejected depends on whether the reduction option is specified (see later). Options, specified on data records, are read in from stream 5 (`G.SYSIN`) as with earlier versions of the program but their form is very different.

Printed output appears on stream 6 (`G.SYSOUT`), starting with a distinctive TP9 banner which can contain user-supplied text. This is followed by information about the sub-version number and date and recent TP news. Next the user options are listed. Assuming that there are no errors (which are clearly flagged and explained), the following pages contain printout pertaining to the event processing with special messages printed for the first event. The printout ends with an event termination message, various statistics and histograms and a final end-of-run message.

The TP bank output of TP9 is identical to that of TP8 so JCN 80 still applies. However note that some entries in the TPTR banks are still not filled, such as the errors on the particle direction cosines. This has been the case in all previous versions of the TP program but hopefully this will be rectified soon.

3. The TP9 Options

An interactive program has been developed to prompt the user for the required options for TP9. The output of the interactive program is the input for the batch job, which must be placed immediately after the `//G.SYSIN DD *` card of the JCL. The interactive program is very easy to use, being self-explanatory. A description of how to run the program will be given in a forthcoming JADE Computer Note.

For those users who choose to specify the TP9 options manually, the format of the options, as read by TP9, will now be given. The first point to note is that no data records need be given but the program will then only delete all existing TPEV, TPTR and TPVX banks and create a new TPEV/1 bank for each event.

Secondly, the data records can be specified in any order since they identify themselves. Thirdly, the information in a record can appear anywhere and in any order except for the identification keyword which must come first. It can be preceded by blanks if so desired. All 80 columns of the record can be used.[3] Keywords must be separated by at least one blank. Fourthly, there are two basic types of data records: special records and group option records.

Special Records

There are four of these. All are optional and sensible defaults are taken if these records are not specified. One is identified with the word `TPCONTROL`, the second by `SMEARINGDATE`, the third by `COMMENT` and the last by `TPEND`. These words must appear as the first sequence of non-blank letters in the record. Except for `COMMENT`, only one of each special record may appear.

The `TPCONTROL` record specifies which event records are to be processed by TP9 and has the form:

```
TPCONTROL  { START i | SKIP j } { END k | PROCESS m }
```

where keywords in curly brackets, {}, are optional and can be omitted. The alternatives are separated by vertical bars. The letters i,j,k,m represent strings of integer digits. Examples of valid `TPCONTROL` records are:

```
TPCONTROL  START 15   PROCESS 100
```

This requests TP9 to start by processing event 15 on the input file (that is, the 15th event according to its position in the file) and finish after processing event 114 such that 100 events are processed. (Assuming there are at least 114 events to be read.)

```
TPCONTROL  END 2000
```

This causes a start at the first event and an end after event 2000 while the one below asks TP9 to skip over the first 100 events on the input file and process all the rest, starting at event 101.

```
TPCONTROL  SKIP 100
```

[3] If a number spans columns 73 to 80, it will be ignored.

If no `TPCONTROL` record is given or if the information is incomplete, the default start record is 1 and the default end record is 9999999.

The `SMEARINGDATE` record is a means of specifying a smearing date for Monte Carlo events in order to override the usage of the configuration date as the default smearing date. The format is:

`SMEARINGDATE` { *day* *month* *year* }

where *day* and *month* are 1 or 2 digit numbers and *year* is a 4 digit number. All 3 arguments must be given or none at all (the latter case being equivalent to omitting this record). An example is:

```
SMEARINGDATE     14 10     1983
```

The `COMMENT` record provides a means of printing 4 lines of comments on the header pages of the TP output. If more than 4 `COMMENT` records (with non-blank information following) are found, they are ignored. The format is:

`COMMENT` { *text* }

Examples are:

```
   COMMENT  Output dataset:  F22BOW.OUTPUT.A1
```

```
COMMENT       This is a test job.  Discard me.
```

```
COMMENT  --> Top secret output.  Not to be seen by MARK-J. <--
```

The first example shows that the word `COMMENT` does not have to start in column 1. The text here would be useful if the output is first written to a temporary dataset before copying to tape since the program will not know what the tape dataset is to be called. Please note that blanks appearing after the word `COMMENT` and before the comment text are copied to the output except for the first which must be there for syntax reasons. This allows comments to be aligned in columns if wanted, as shown in the second and third examples.

The `TPEND` record is useful for providing an end-of-file marker for the case when further records are appended to `G.SYSIN` for use by a later job step. If absent, option records will be read until a FORTRAN EOF occurs. The format is:

```
TPEND
```

Clearly a `TPEND` record, if present, must be the last option record to be specified, otherwise the following option records will not be read by the program.

Group Option Records

Read this section carefully. The program only performs the actions you ask for. That is, there are no default actions.

The Group Option records all have the same format, with the first keyword identifying which 'group' of options is being referred to. Basically a group corresponds to a subdetector but the options for refitting, charged track processing and overall event analysis are incorporated into their own groups. All the options for a particular group must be given on one record. The format is:

group { *option_keyword* } { *option_keyword* } { *option_keyword* } ...

with different *option_keywords* appearing any number of times (including zero), separated by at least one blank. For example:

```
MUON    ANALYSE    IFNOTDONE    TP    DELETE
```

where `MUON` is the *group* and `ANALYSE`, `IFNOTDONE`, `TP` and `DELETE` are *option_keywords*. The allowed values of *group* are:

`JETC` `VTXC` `TRACK` `TOF` `LG` `REFIT` `MUON` `TAGG` `DEDX` `CALCS`

which are self-explanatory except possibly `REFIT` (which deals with a refit of the Jet Chamber), `TRACK` (which summarises the latest PATR bank) and `CALCS` (which deals with global event calculations like sphericity and thrust plus event reduction).

The allowed values of *option_keyword* depend to some extent on the *group* but the basic list (common to almost all the groups) is:

`ANALYSE`	perform the relevant reconstruction task
`IFNOTDONE`	qualifier to `ANALYSE`; prevents re-analysis
`TP`	perform the summary; create or modify the TP banks
`DELETE`	delete detector results banks after summarising

The exact action for each group is specified later in the note. The important point to emphasise again is that the absence of these *option_keywords* implies that the corresponding actions are NOT done. In addition, please note that qualifiers, like `IFNOTDONE`, are meaningless without their associated verbs.[4]

[4]The qualifiers can appear anywhere in the record but it is good practice to place them immediately after the relevant verb.

A summary of all the legal *option_keywords* now follows. Note that a few of them are mutually exclusive.

```
JETC    RFICAL IFNORFICAL ZCAL OLDDEL ANALYSE IFNOTDONE ZVTX
VTXC    ANALYSE  COMFIT
REFIT   RFI IFNORFI STRONGVTX WEAKVTX ZS IFNOZS COMMONZVTX OLDPATR
TRACK   TP  VPROCESS  DELETE
DEDX    ANALYSE  TP
TOF     ANALYSE  IFNOTDONE  TP  DELETE
LG      LGCAL  IFNOLGCAL  ANALYSE  IFNOTDONE  TRACKMATCH  TP DELETE
MUON    ANALYSE  IFNOTDONE  TP  DELETE  FULL
TAGG    ANALYSE  IFNOTDONE  TP  DELETE
CALCS   SPHERICITY  THRUST  REDUCTION
```

4. What TP9 Does In Detail

The first action performed, after an event is read in, is the deletion of all existing TP banks followed by the creation of the TPEV/1 bank. Next the program performs (if requested) the following tasks, in the order given below:

- jet chamber basic processing including 'calibration'
- jet chamber track refitting (Spitzer routines)
- vertex chamber processing and combined fit with jet chamber
- copying charged tracks from lowest the PATR bank into TPTR banks
- primary vertex finding, the creation of the TPVX/1 bank and the adjustment of the direction cosines in the TPTR banks for tracks from the primary vertex
- deletion of some or all PATR banks
- TOF analysis, summary of results in TPTR banks and deletion of the TOFR bank
- LG 'calibration', analysis and cluster-track matching
- creation of photon summary TPTR banks and addition of cluster information to charged track TPTR banks
- deletion of LGCL bank

- muon filter analysis, addition of muon information to TPTR banks and MUR2 bank deletion
- tagging system analysis, creation of new TPTR banks and deletion of tagging results banks
- dE/dx analysis and addition of results to TPTR banks
- identification of particles in TPTR banks
- finding of charged decays, V^0's and converted photons, and creation of new TPVX and TPTR banks
- compilation of event statistics and copying them into the TPEV/1 bank
- computation of visible energies, sphericity and thrust, and histogramming them and other quantities
- flagging events that fail the reduction cuts
- writing out the event to the relevant dataset

4.1 Jet Chamber Processing

The actions performed are: calibration in $r\phi$ and z, deletion of all existing PATR banks,[5] PATRCO analysis and a fast z vertex search. All of these actions are optional with the option group name being `JETC`.

`RFICAL`	do $r\phi$ calibration of the Jet Chamber
`IFNOFICAL`	qualifier to `RFICAL`; prevents recalibration
`ZCAL`	do z calibration of Jet Chamber (no qualifier)
`ANALYSE`	call PATRCO to perform pattern recognition
`IFNOTDONE`	qualifier to `ANALYSE`; prevents re-analysis if a PATR exists
`OLDDEL`	qualifier to `ANALYSE`; delete existing PATR banks[5] first
`ZVTX`	do fast z vertex search

Note that `RFICAL IFNORFICAL` is a sensible choice if `ANALYSE` is specified.

[5]Except PATR/12

4.2 Charged Track Refitting

The actions performed are: Spitzer $r\phi$ and Spitzer zs refitting. All of the actions are optional with the option group name being REFIT. As always, the default actions are no actions, i.e. no vertex constraints are used in fits unless asked for explicitly. See Jade Computer Notes 94 and 95 for details about the refitting programs.

RFI	perform an $r\phi$ refit (defaults: *unconstrained* and *create new* PATR *bank*)
IFNORFI	qualifier to RFI; prevents refitting
STRONGVTX	qualifier to RFI; with *strong* vertex constraint
WEAKVTX	qualifier to RFI; with *weak* vertex constraint, i.e. vertex errors multiplied by 100
ZS	perform zs refit (defaults: *no common z vertex constraint* and *create new* PATR *bank*)
IFNOZS	qualifier to ZS; prevents refitting
COMMONZVTX	qualifier to ZS; with *common z vertex constraint*
OLDPATR	qualifier to RFI and ZS; *overwrite* existing PATR bank

4.3 Vertex Chamber Processing

The actions performed are: vertex chamber analysis (using the latest PATR bank, possibly created in the JETC or REFIT steps) followed by a combined fit with the jet chamber (COMFIT). Standard routines VTXCSF and VTXCSV are used. Both of these actions are optional with the option group name being VTXC.

ANALYSE	do vertex chamber analysis
COMFIT	perform a combined fit with jet chamber to make HWDS bank

4.4 Charged Track Summarising

The actions performed are: TP track summarising using the PATR bank with the lowest number to create a TPTR bank for every track found, vertex processing of every TPTR bank and the deletion of all PATR banks.[6] All of these actions are optional

[6]Not implemented in version 9.1

with the option group name being TRACK. The vertex processing action results in the PATR bank being analysed to find the primary vertex. A TPVX bank 1 is then created. The corresponding TPTR track numbers are added to the TPVX bank and the starting positions and direction cosine information in the TPTR banks are set accordingly. No correction for energy loss for each track is made, at present. All later vertex related tasks, such as decay finding, V^0 searches and converted photon searches are also dependent on the VPROCESS option in this section.

TP	perform a summary of the PATR bank with the lowest number; create TPTR banks as explained above
VPROCESS	perform vertex processing as explained above
DELETE	delete all PATR banks[7] after summarising (except PATR/12)

4.5 TOF Processing

The actions performed are: TOFINT analysis, summarising the results into the charged track TPTR banks (if present) and deletion of the TOFR bank. All of these actions are optional with the option group name being TOF.

ANALYSE	perform the TOF analysis (TOFINT)
IFNOTDONE	qualifier to ANALYSE; prevents re-analysis
TP	perform the summary; modify TPTR banks for charged tracks (if any)
DELETE	delete TOFR bank after summarising

4.6 LG Processing

The actions performed are: LG calibration (LGCALB), cluster finding (LGANAL), track-to-cluster matching by position (LGCDIR) and TP summarising which modifies some charged track TPTR banks and creates new neutral particle TPTR banks. All these actions are optional with the option group name being LG.

LGCAL	calibrate the LG system (LGCALB)
IFNOLGCAL	qualifier to LGCAL; prevents recalibration

[7]Not implemented in version 9.1

ANALYSE	perform the LG cluster finding analysis (LGANAL)
IFNOTDONE	qualifier to ANALYSE; prevents re-analysis
TRACKMATCH	join tracks and clusters (LGCDIR)
TP	perform the summary; modify TPTR banks for charged tracks (if any) and create new TPTR banks for neutrals
DELETE	delete the LGCL bank[8]

4.7 Muon Filter Processing

The actions performed are: muon analysis[9] (MUANA), TP summarising which modifies charged track TPTR banks and two levels of muon results bank deletion. All these actions are optional with the option group name being MUON.

ANALYSE	perform the muon filter calibration and analysis (MUANA)
IFNOTDONE	qualifier to ANALYSE; prevents recalibration and re-analysis
TP	perform the summary; modify TPTR banks for charged tracks (if any)
DELETE	delete MUR2 banks 5 and 6 at end
FULL	qualifier to DELETE; delete banks 2, 3 and 4 as well

4.8 Tagging System Processing

The actions performed are: tagging analysis (TAGAN), TP summarising which creates new TPTR banks and deletion of the ATAG and TAGG banks after summarising. All these actions are optional. The option group name is TAGG.

ANALYSE	perform the tagging system calibration and analysis (TAGAN)
IFNOTDONE	qualifier to ANALYSE; prevents recalibration and re-analysis
TP	perform the summary; create new TPTR banks
DELETE	delete ATAG and TAGG banks

[8] Not implemented in version 9.1

[9] Including (re)calibration

4.9 dE/dx Processing

The actions performed are: dE/dx analysis and TP summarising which modifies the TPTR banks of charged tracks. Since there is no DEDX bank, both actions must be performed or neither of them. The option group name is DEDX.

ANALYSE	perform the dE/dx analysis of Jet Chamber tracks (DEDXBN)
TP	perform the summary; modify TPTR banks for charged tracks

4.10 Particle Identification

An attempt is made to identify the particles in the event using information determined from the muon filter, LG detector, Jet Chamber (dE/dx) and the TOF counters. Searches for V^0 particles, converted photons and tracks with kinks are also performed, depending on the VPROCESS keyword specified as part of the TRACK group of options. The TPEV and TPTR banks are updated and new TPTR and TPVX banks created when new particles are found.

4.11 Event Analysis

Two actions occur that are not optional. They are the calculation of visible energy (charged and neutral) from the TPTR banks (which is written into the TPEV/1 bank and the histogramming of certain quantities.

The optional actions (grouped together under the name CALCS) are the calculations of sphericity and thrust and determining whether an event is to be accepted or rejected. This reduction process depends on the code incorporated into the reduction routine. The default code is for multihadronic event selection (MCREDU).

SPHERICITY	calculate sphericity (SPHERI)
THRUST	calculate thrust (THRUST)
REDUCTION	perform the desired reduction

5. Warning and Error Messages

At present the messages printed by the TP routines[10] are rather terse and give only a short traceback from the routine that detected the error or generated the warning. This is often enough to determine the cause of the problem. However this should be improved in later versions when experience provides information on why the errors occur. Since there are some routines from version 8 linked in, there may be error messages printed in the old TP8 style.

Please report occurrences of errors and warnings so that the program can be changed to remove unnecessary messages or provide more user-friendly ones.

6. Running the Program

At DESY, a standard job exists to run the program which only requires the addition of the option records ('cards'). As stated before, these can be added by hand or by using the output from the interactive options generator. The standard JCL member is called `#RUNTP9` and lives on `F22BOW.TP9.S`, the source library for TP9. Another member, `#RUNTP9C`, includes a tape-to-disk copy step before the TP step. The Siemens FORTRAN77 load library is called `F22BOW.TP9.L` and there is also an IBM FORTVS load library called `F22BOW.TP9.VSL`.

Note that there is a problem with FORTVS `BLOCK DATA` linking so members `TPDICT` and `TPBD01` should be explicitly `INCLUDE`d in the link step if the FORTVS library is used.

Running TP9 at RAL will involve using copied FORTVS load modules from DESY. When the program is ready an announcement will be made. Note, a VAX version of the interactive options program exists and can be used in conjuction with remote job submission to RAL from the Manchester and Lancaster VAXes. Usage at Heidelberg is still to be decided. Please ask for the latest information.

Finally, as always, be aware that future improvements might entail larger region sizes or library changes so always check the standard JCL members before submitting long production jobs.

[10]With the exception of the options handling routines

JADE Computer Note 102

The New TP Program Version 9

C.K. Bowdery J.J. Pryce

August 11, 1988

1. Introduction

The TP program reconstructs and summarises JADE events, producing datasets containing a TPEV/1 bank (with global event information), a number of TPTR banks (each with details of one particle) and a number of TPVX banks (each with details of one vertex). The datasets will also contain raw and processed data banks as requested by the user.

The program has existed for many years. Version 8[1] was released in 1984 and has been the standard version up till now. It is documented in JCN 79 with the TP Banks being documented separately in JCN 80. A special Heidelberg version 8+ has been used recently, since version 8 has been virtually frozen during the development of a new version, which is the subject of this note.

Version 8 is a direct descendent of earlier versions of the program originally written by S. Yamada and E. Elsen. It was originally intended to be the ultimate version but various shortcomings in the program structure has made it unsuitable for adding new sections. Instead the program has been almost completely re-written along radically different lines.

The new program, to be known as Version 9 or TP9 for short, follows the philosophy of Structured Design (part of SASD) in consisting of a very large number[2] of small, 'weakly coupled' subroutines with 'strong cohesion'. However the nature of JADE programs, with the event data in a BOS common block, has meant that the

[1]The version numbers correspond roughly to the generation number of the multihadronic events the program produced.

[2]About 140, not counting the standard JADE packages called.

implementation is far from pure SASD with its afferent and efferent data flows. Nevertheless, it conforms to most of the 'design heuristics' that ensure reliability and ease maintenance problems.

Since it is a feature of SASD that a design can be evaluated using a number of graded criteria, it should be possible to evaluate the quality of TP9. Comments and criticism in this area (and all others) are thus welcome.

2. The Nature of the Program

The program normally runs as a batch job, reading events in from stream 1 and writing out 'accepted' events on stream 2 and 'rejected' events on stream 3. Whether events are rejected depends on whether the reduction option is specified (see later). Options, specified on data records, are read in from stream 5 (`G.SYSIN`) as with earlier versions of the program but their form is very different.

Printed output appears on stream 6 (`G.SYSOUT`), starting with a distinctive TP9 banner which can contain user-supplied text. This is followed by information about the sub-version number and date and recent TP news. Next the user options are listed. Assuming that there are no errors (which are clearly flagged and explained), the following pages contain printout pertaining to the event processing with special messages printed for the first event. The printout ends with an event termination message, various statistics and histograms and a final end-of-run message.

The TP bank output of TP9 is identical to that of TP8 so JCN 80 still applies. However note that some entries in the `TPTR` banks are still not filled, such as the errors on the particle direction cosines. This has been the case in all previous versions of the TP program but hopefully this will be rectified soon.

3. The TP9 Options

An interactive program has been developed to prompt the user for the required options for TP9. The output of the interactive program is the input for the batch job, which must be placed immediately after the `//G.SYSIN DD *` card of the JCL. The interactive program is very easy to use, being self-explanatory. A description of how to run the program will be given in a forthcoming JADE Computer Note.

For those users who choose to specify the TP9 options manually, the format of the options, as read by TP9, will now be given. The first point to note is that no data records need be given but the program will then only delete all existing TPEV, TPTR and TPVX banks and create a new TPEV/1 bank for each event.

Secondly, the data records can be specified in any order since they identify themselves. Thirdly, the information in a record can appear anywhere and in any order except for the identification keyword which must come first. It can be preceded by blanks if so desired. All 80 columns of the record can be used.[3] Keywords must be separated by at least one blank. Fourthly, there are two basic types of data records: special records and group option records.

Special Records

There are four of these. All are optional and sensible defaults are taken if these records are not specified. One is identified with the word `TPCONTROL`, the second by `SMEARINGDATE`, the third by `COMMENT` and the last by `TPEND`. These words must appear as the first sequence of non-blank letters in the record. Except for `COMMENT`, only one of each special record may appear.

The `TPCONTROL` record specifies which event records are to be processed by TP9 and has the form:

TPCONTROL { START *i* | SKIP *j* } { END *k* | PROCESS *m* }

where keywords in curly brackets, {}, are optional and can be omitted. The alternatives are separated by vertical bars. The letters i,j,k,m represent strings of integer digits. Examples of valid `TPCONTROL` records are:

```
TPCONTROL   START 15    PROCESS 100
```

This requests TP9 to start by processing event 15 on the input file (that is, the 15th event according to its position in the file) and finish after processing event 114 such that 100 events are processed. (Assuming there are at least 114 events to be read.)

```
TPCONTROL   END 2000
```

This causes a start at the first event and an end after event 2000 while the one below asks TP9 to skip over the first 100 events on the input file and process all the rest, starting at event 101.

```
TPCONTROL   SKIP 100
```

[3]If a number spans columns 73 to 80, it will be ignored.

If no TPCONTROL record is given or if the information is incomplete, the default start record is 1 and the default end record is 9999999.

The SMEARINGDATE record is a means of specifying a smearing date for Monte Carlo events in order to override the usage of the configuration date as the default smearing date. The format is:

```
SMEARINGDATE     { day   month   year }
```

where *day* and *month* are 1 or 2 digit numbers and *year* is a 4 digit number. All 3 arguments must be given or none at all (the latter case being equivalent to omitting this record). An example is:

```
SMEARINGDATE      14 10      1983
```

The COMMENT record provides a means of printing 4 lines of comments on the header pages of the TP output. If more than 4 COMMENT records (with non-blank information following) are found, they are ignored. The format is:

```
COMMENT    { text }
```

Examples are:

```
   COMMENT  Output dataset:  F22BOW.OUTPUT.A1
```

```
COMMENT        This is a test job.  Discard me.
```

```
COMMENT   --> Top secret output.  Not to be seen by MARK-J. <--
```

The first example shows that the word COMMENT does not have to start in column 1. The text here would be useful if the output is first written to a temporary dataset before copying to tape since the program will not know what the tape dataset is to be called. Please note that blanks appearing after the word COMMENT and before the comment text are copied to the output except for the first which must be there for syntax reasons. This allows comments to be aligned in columns if wanted, as shown in the second and third examples.

The TPEND record is useful for providing an end-of-file marker for the case when further records are appended to G.SYSIN for use by a later job step. If absent, option records will be read until a FORTRAN EOF occurs. The format is:

```
TPEND
```

Clearly a TPEND record, if present, must be the last option record to be specified, otherwise the following option records will not be read by the program.