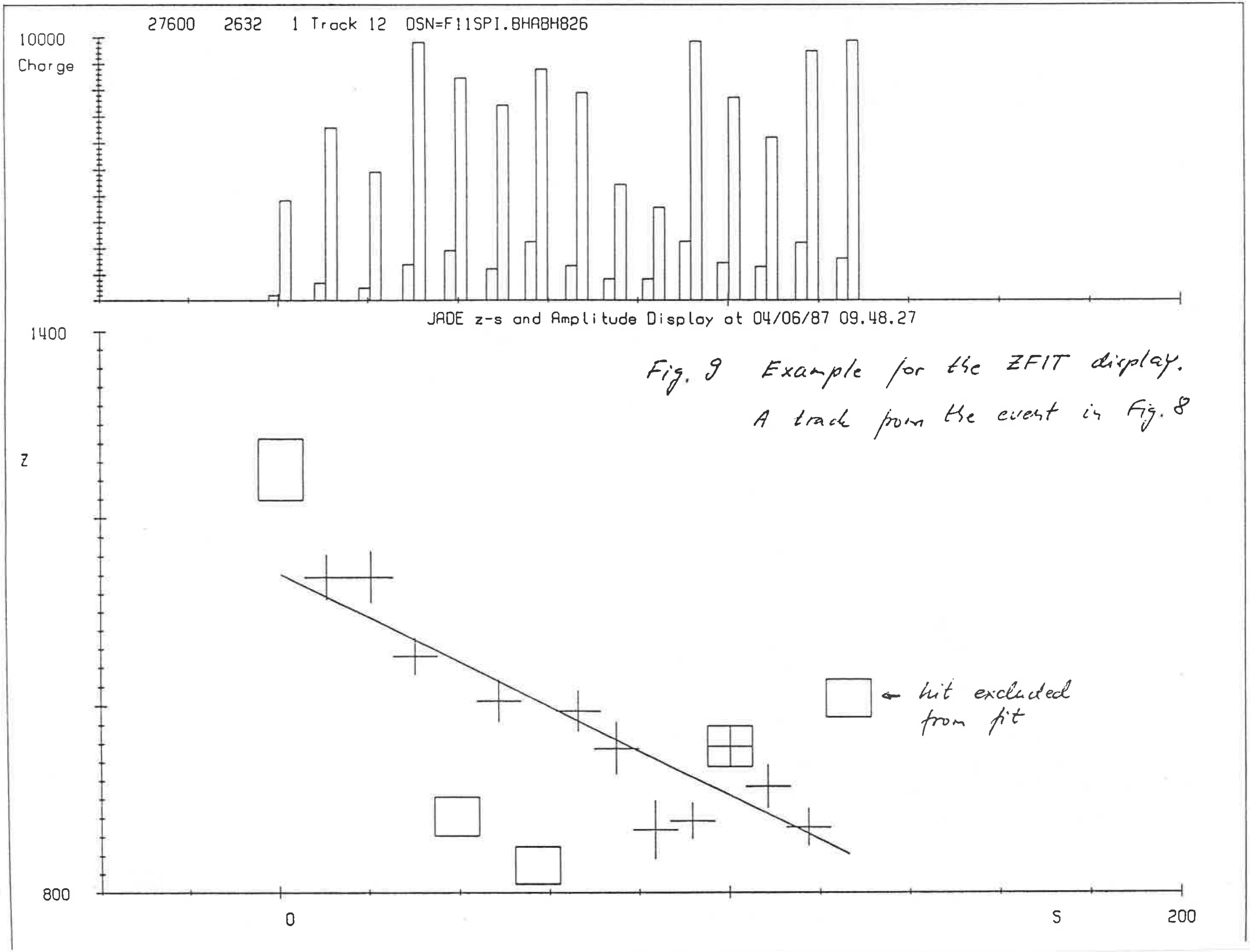


Fig. 9 Example for the ZFIT display.
A track from the event in Fig. 8

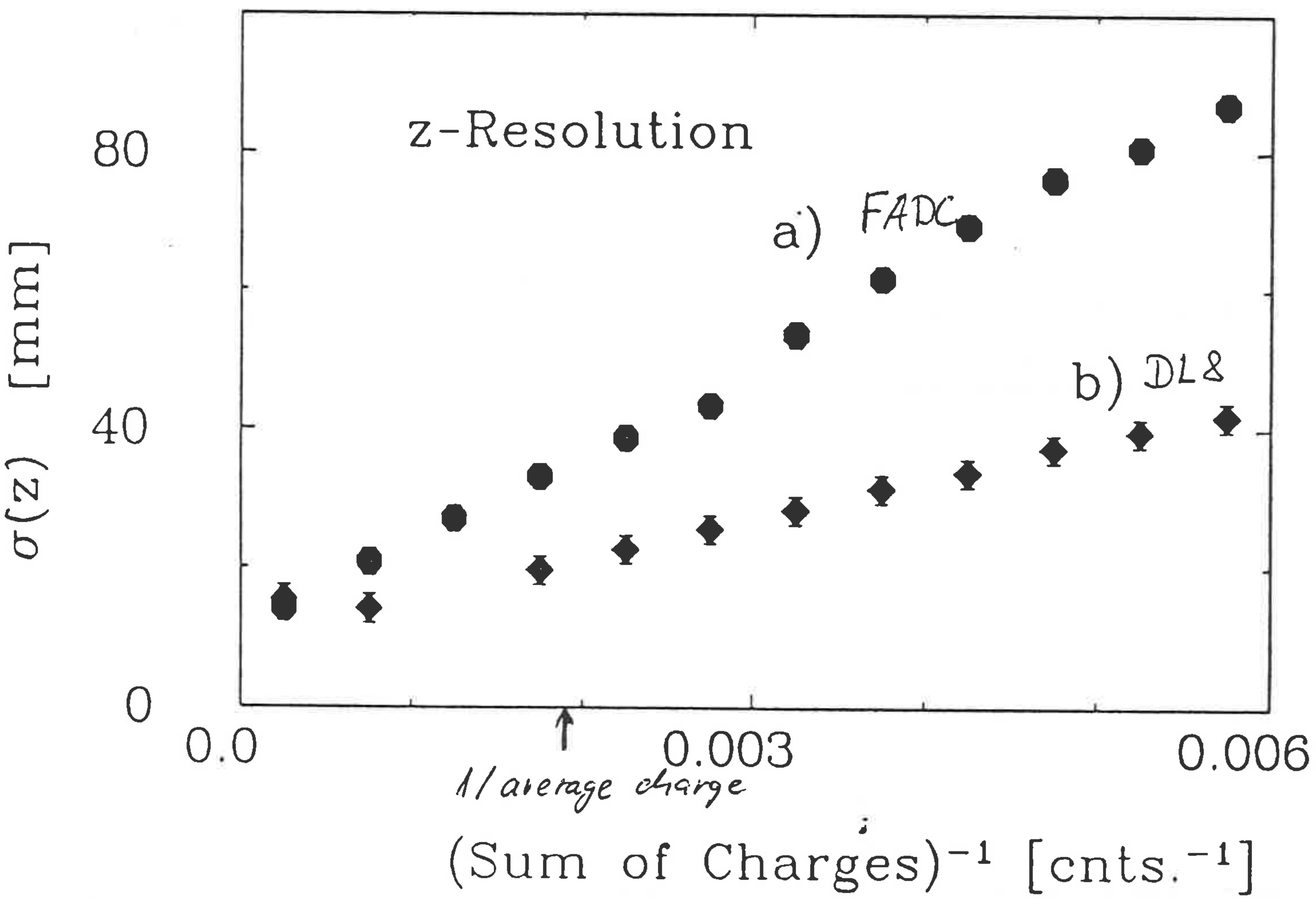
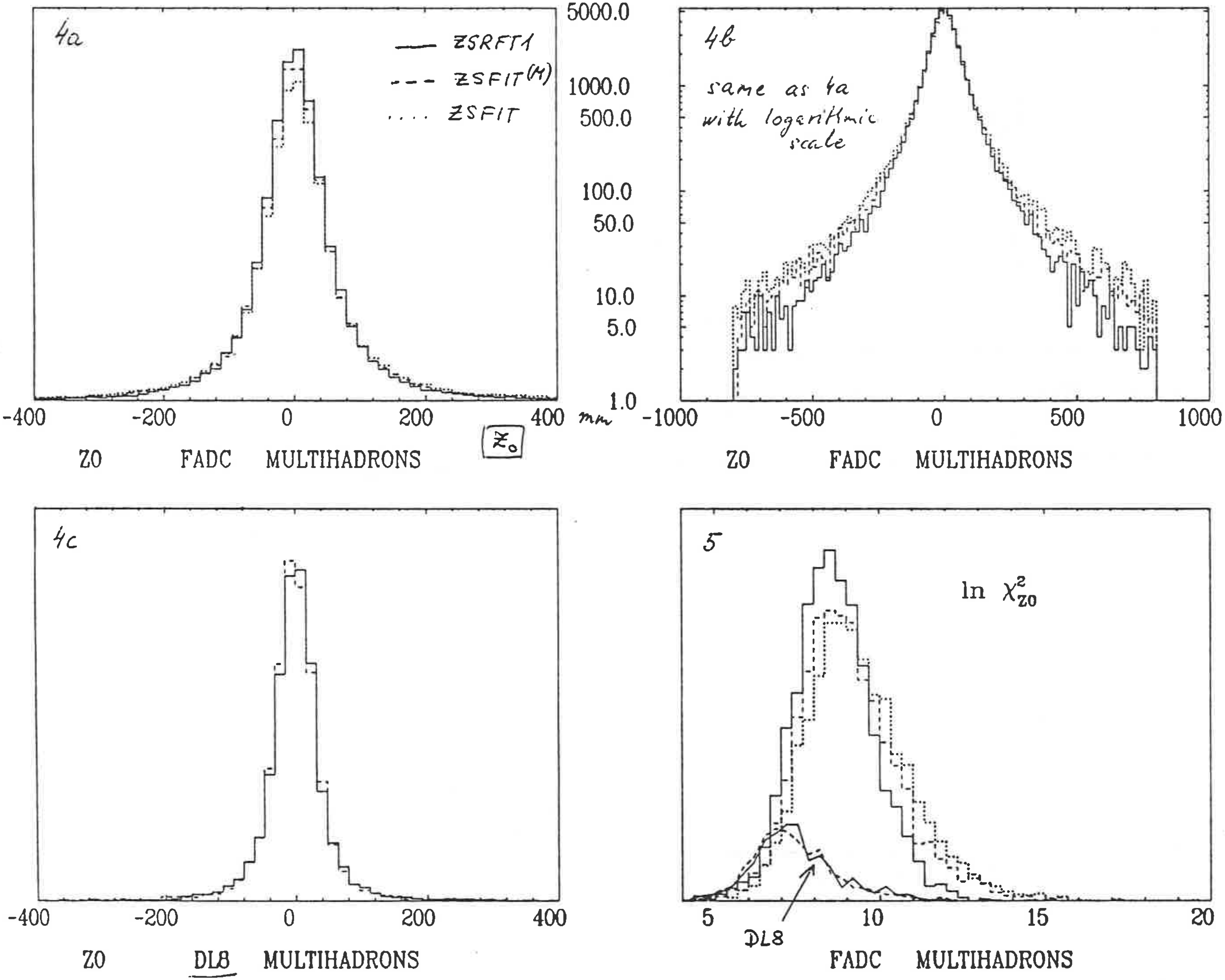
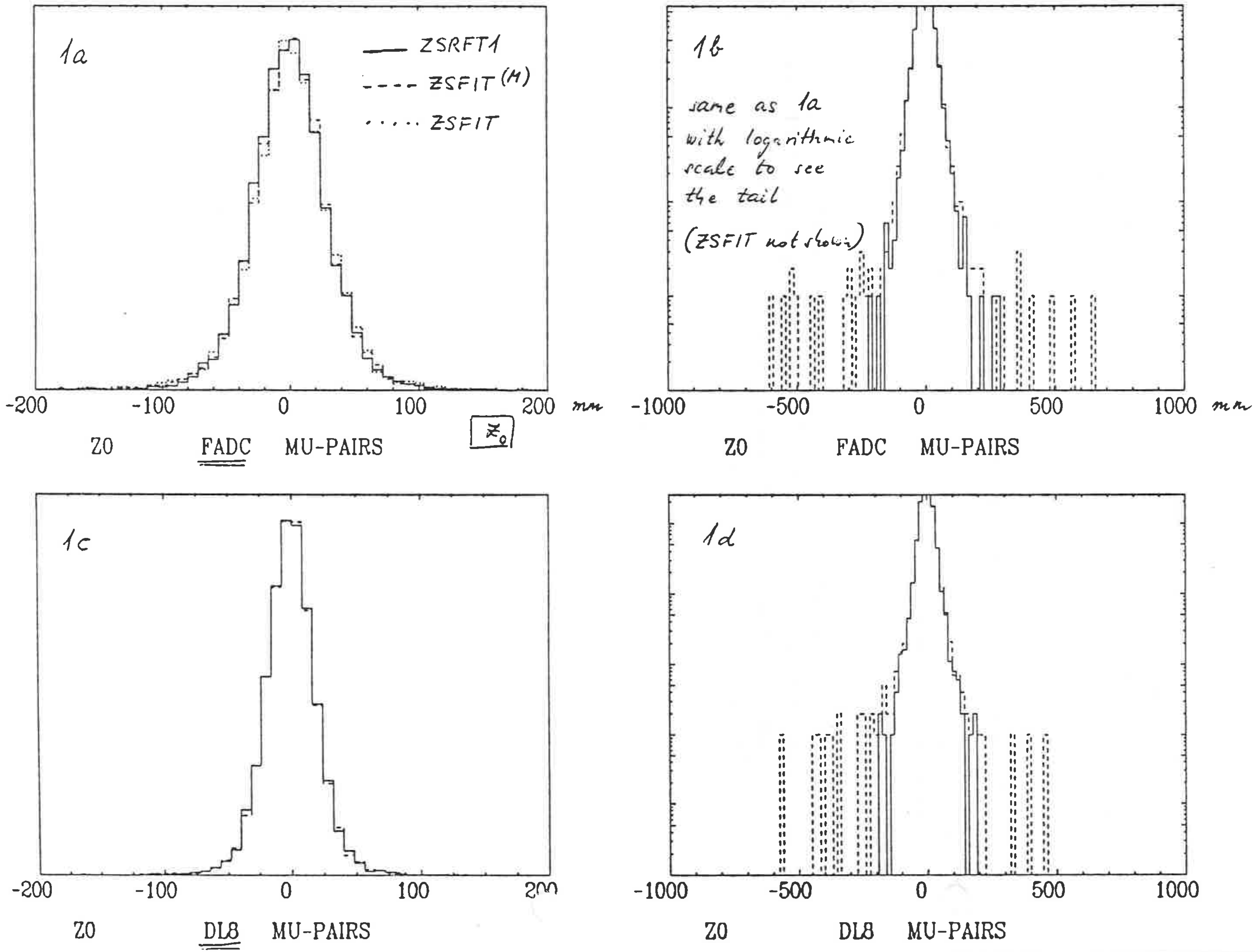

z-Resolution
a) FADC
b) DL8
80
40
0
0.0
0.003
0.006
σ(z) [mm]
1/average charge
(Sum of Charges)⁻¹ [cnts.⁻¹]

Fig. 7


4a
ZSRFT1
ZSFIT(M)
ZSFIT
-400
-200
0
200
400
mm
z_0
ZO FADC MULTIHADRONS
5000.0
1000.0
500.0
100.0
50.0
10.0
5.0
1.0
4b
same as 4a with logarithmic scale
-1000
-500
0
500
1000
ZO FADC MULTIHADRONS
4c
-400
-200
0
200
400
ZO DL8 MULTIHADRONS
5
ln χ^2_{ZO}
DL8
5
10
15
20
FADC MULTIHADRONS


1a
ZSRFT1
ZSFIT (M)
ZSFIT
-200
-100
0
100
200
mm
Z0
FADC
MU-PAIRS
z_0
1b
same as 1a
with logarithmic
scale to see
the tail
(ZSFIT not shown)
-1000
-500
0
500
1000
mm
Z0
FADC
MU-PAIRS
1c
-200
-100
0
100
200
Z0
DL8
MU-PAIRS
1d
-1000
-500
0
500
1000
Z0
DL8
MU-PAIRS

The experimental ratio of the σ's, by our construction, is the same for DL8 and FADC. Quite independently from the value of n wich we take to be 44, Eq.(2) gives $r_{mid} = 550\ mm$ (instead of the half radius of 496 mm of the sensitive part of the jet chamber). With these values, the single hit resolution can then be estimated by using Eq.(1).

The performance of three single track fitting routines have been compared: the standard ZSFIT routine, its new version ZSFIT$^{(M)}$ with modified parameters for the FADC and the new routine ZSRFT1 used by ZSRFTV. (The routine ZRFIT in the pattern recognition package uses unweighted hits and gives much worse, upto factor 2, results.) All three routines do equally well in simple topological configurations such as μ- or Bhabha-pairs where the tracks have many hits and are little disturbed by other tracks. Their common results are listed in Table 1 for DL8 and FADC μ-pairs.

The single hit resolutions derived are in agreement with the results obtained during the calibration of the z-coordinate in late 86. The resolutions as function of the amplitude for DL8 and FADC data are given in Fig.7. At the working point (average amplitude 550 cnts.) the single hit resolutions are in rough agreement with the values in Table 1, indicating that systematic effects possibly left in the calibration do not have a dominant contribution to the track resolution.

Fit results with common z constraint are also given in the Table. The old routine FITEVR is slightly worse than our ZSRFTV because unweighted hits are used and, in addition, cut parameters have not been tuned to the worse resolution of the FADC resulting in rejection of too many hits (Fig.3).

In the simple case of two track events, the track resolution is limited by the single hit resolution and depends neither on the method, nor on the cuts applied to fit the tracks. In a multiparticle environment, however, the hit cleaning technique and starting value search may have some importance, and it is this application where the three routines can be distiguished. Fig.4a shows the distributions of the track intersection with the beam line in FADC multihadron events. The width of this distribution includes the effects of interactions in the beam pipe and the resolution. ZSRFT1 gives a slightly smaller width of this distribution than the other routines and also a smaller tail as seen on the logarithmic display in Fig.4b. The quantity

$$\chi_n^2 \equiv \frac{1}{n-1} \sum_{k=1}^{n} (z_{0k} - \bar{z}_0)^2$$

constructed on an event with n tracks, would follow a known distribution under certain conditions which do not satisfy for multihadronic events. (They do for the μ-pairs and the same resolution can be extracted as from the method described above.) Still, the distribution of χ^2 measuring the spread of the track intersections at the beam line within events, is a good indicator of the quality of the different fits. Fig.5 gives a comparison of $\log \chi^2$ for the fits considered (all n included). The optimized ZSFIT$^{(M)}$ gives a better result than the old standard ZSFIT, ZSRFT1 is still better. Fig.6 indicates that ZSRFT1 rejects the least number of hits, while the old ZSFIT clearly rejects too many hits. This does not apply to the DL8 data and correspondingly we do not observe a similar improvement there.

In the single track fitting routine ZSRFT1, it is first checked whether the very wide ± 600 mm road around the original fit contains 75% of the available (marked as usable) hits. In the small fraction of cases when it does not, starting values are searched for in the following way. Fit parameters are calculated from hit pairs taken in the following sequence: (1,n), (1,$n-1$), (2,n), || (2,$n-1$), (2,$n-2$), (3,$n-1$), || .... (The hits are ordered in s so that the hits 1 and n are at the two ends of the track. The sequence does not include all possible pairs.) The search is terminated if the "75% test" is satisfied for some hit pair. If that never happens, the parameters taken correspond to that hit pair for which the largest number of hits is included in the road (and for which the unweighted χ^2 is the smallest in case of equal number of hits.)

A sequence of straight line fits are then performed. From the $k+1$-th fit, k hits having the largest weighted residuals to the k-th fit are excluded. These hits, if their residuals happen to decrease, may be reincluded later on. On the other hand, the hits which, in any step, are further than 600 mm from the fit, will not be reconsidered. The termination of the sequence is based on two quantities, the (weighted) χ^2/dof and its rate of decrease to the next fit, similarily to the procedure implemented for the $r-\varphi$ fit in XYRFT1. The number of hits k allowed to be discarded (the number of iterations, more precisely) is limited to 25% of the available hits when the iteration is started.

For the common z fits in ZSRFTV those and only those hits are taken which were included in the fit in ZSRFT1.

Resolution in the fit parameters.

For the statistical error of the fit parameters $z_0 \equiv z(r \approx 0)$ and $\cot\vartheta = dz/ds$, one can easily derive the following formulae:

$$\sigma^2_{int}(z_0) = \frac{\sigma^2_{hit}}{n}\left[1 + \left(\frac{r_{mid}}{d}\right)^2 \frac{12}{n^2-1}\right] \tag{1a}$$

$$\sigma^2_{int}(\cot\vartheta) = \frac{\sigma^2_{hit}}{n}\frac{12}{d^2(n^2-1)}, \tag{1b}$$

where n is the number hits measured with the (same) error σ_{hit} at equal distances d from each other in r, and r_{mid} is the radius of the middle of the track. Both errors are independent of the track position and orientation if one neglects the z-dependence of σ_{hit}.

The resolutions can be measured experimentally by making use of the back-to-back property of μ-pairs (or Bhabha's). While the width of the z_0 plot includes the beam spread, the quantity $(z_{01} - z_{02})/\sqrt{2}$ is not affected by the uncertainty in the interaction point and its width $\sigma_{int}(z_0)$ is a direct measure of the intrinsic extrapolation error at the beam line. From comparing the two quantities, the spread in z of the interaction point can be extracted as well as from the comparison of $\sigma(z_0)$'s in fits with and without common z constraint. (See Table 1 for more details.)

For determining the resolution in $\cot\vartheta$, the quantity $(\cot\vartheta_1 + \cot\vartheta_2)/\sqrt{2}$ can be plotted. (The sum of $\cos\vartheta$, for example, would have an error depending on ϑ and one would mix different resolutions in a plot if ϑ is left unconstrained.) The contribution of the effect of initial

In addition to the results in the 'PATR' bank, the routine provides you with the following information:

KTSTZS(·,2,·) are flags with the following meaning:

- 2 Track was used in a common z fit.
- 1 Single track fit was successful but the track was not used in a common fit.
- 0 Track was not considered for refit.
- <0 Single track fit was attempted but failed.

(Please, for the exact conditions, look at the code.)

SQCHZS(2,·) is set to the total number of hits (R*4 (!)) used in the common fit; SQCHZS(1,·) will hold the sum of weighted residual squares (average weight close to 1) divided by the number of degrees of freedom $n_{hits} - (n_{tracks} + 1)$. The normalization of the latter to get the χ^2/dof is left to the user. It should be done differently for the DL8 and FADC data. SQCHZS(3,·) will contain the common z at the common (x, y) point. All three values are set to 0.0 if the common fit was actually not performed.

In case IOPT=1 or 2, you need not care about the COMMON unless you want to get the information just described. For these values of IOPT, the track flags are given in KTSTZS(·,2,1) (KTSTZS(ITR,1,1)=ITR is implicitly assumed without being set) and, in addition for IOPT=2, SQCHZS(·,1) holds the corresponding quantities for the common fit.

The options IOPT $\rightarrow$ IOPT+8 are also valid; when used, the single track fitting routine ZSRFT1 creates a bank 'ZSPD' with certain hit data for the graphics program (only if the fit is successful and there is enough space in /BCS/). If such banks are already present, you may delete them to allow for creating new ones, on LEVEL 2. The new graphics command ZFIT has been provided to look at the single track $s - z$ fit results together with the hit coordinates with symbols indicating the measurement error and whether or not the hit was included in the fit.

The cut parameters mentioned in the explanation of the IOPT-parameter for selecting tracks for the common fits are in COMMON /CCMZCT/ DIMPCT, ZCUTV, ZCUTVV with the values 15.0, 800.0, 1600.0, respectively and hence can easily be changed. The track selection should in fact be carefully designed for the different applications. In particular, the 15 mm cut on the impact in $r - \varphi$ might be too lose in many cases (the corresponding cut in the old routine FITEVR is 25 mm), and the z cuts should be chosen appropriately depending on the track quality requirements employed in the analysis. We encourage you to consider changing the parameters and even the code, as far as the selection is concerned, for your actual needs. (The routine is written in SHELTRAN but you need not know more about the language than you see around the code you are going to modify.) A cut on $z - \bar{z}|_{\text{common } x,y}$ is coded on the lines starting with 'COMIT' and may be activated by removing the comments.

direction the distance along the y-axis is minimized.

The vertex is not included in the above procedure (it could only disturbe it). Instead, at the end, if vertex constraint was requested, it is checked whether the track is consistent with the run vertex by comparing the contribution of the vertex to the χ^2 with the final fit parameters and the χ^2/d.o.f. If a certain (safe) cut condition is satisfied, the fit is repeated (once, no more hits thrown away) with the vertex constraint otherwise the fit without it will be kept.

It has been checked using μ-pairs that this procedure gives slightly better results for the DL8 and definitely better results for the FADC than FRFITV in its current, not optimized for the FADC, form. Concerning low momentum tracks, they are in general longer when fitted with this routine, apparently the picture is better recognized. An example is included in the following figures. I have not seen a failure of the routine (but I have not tried very hard). I have looked at the cases in which a problem indicated by loss of too many hits occurred, the result was always reasonable and better than the earlier fit. If you detect a problem or have some suggestion, please, tell me. Good luck!

These routines however were written for special applications and although the majority of physically interesting tracks gets refitted, for some tracks the old fit remains. Only parabola fit is provided and in case of a bad fit or/and large curvature, the third ring or even the second ring is omitted. This may be reasonable for tracks coming indeed from the centre of the detector (for which these routines were written), but may fail for V^0's originating in the second ring, for instance. The residual cut in these routines has not been optimized for the FADC resolution and in case a vertex constraint is requested it may happen that the residual cut removes so many hits that the number of degrees of freedom becomes zero and a divide check occours.

For the purpose of refitting essentially all tracks in $r-\varphi$, I've written a fitting routine that provides circle fit, a better, unified for DL8 and FADC hit cleaning procedure and a more appropriate handling of the vertex constraint. The vertex chamber hits have not been included but it is forseen to develope, together with the vertex chamber group, a routine using all the experience we have. (They are already using the new calibration provided in JFTNEW together with the vertex chamber hits in their private fitting routine.) The common r-z fit which is very important indeed, especially for the FADC data, has to be done with FITEVR. I'm planning to incorporate it into the new routine sometime in the future. In what follows, I describe first how the new routine(s) are called and then give certain details concerning the fit itself.

Subroutine XYRFTV(MODE).

This routine contains a loop over the tracks and selects the 'PATR' bank to be updated by the single track fitting routine. The input argument MODE has a similar meaning as the one for FITEVR:

- (= 0) Overwrite old 'PATR' bank with new results
- (= 1) Create new 'PATR' bank with new results
- (+ 2) Not used (for FITEVR: also common z fit)
- (+ 4) Vertex weakly constrained (ERRFAC = 100.0)
- (+ 8) Not used (for FITEVR: rerun patrec in case of bad 'JHTL' bank)
- (+16) No vertex constraint (ERRFAC = 1000.0)

In the most likely case, you don't need vertex contraint and you need to call with MODE = 16 or 17. Even if you ask for a vertex constraint (MODE = 0 or 1) in which case ERRFAC is set to 1.0, the vertex will not be used if it is incompatible with the track fitted without vertex constraint first (see later).

, LDTR

Subroutine XYRFT1(IPTR,IPJHTL,ERRFAC).

This routine performs the $r-\varphi$ fit for a single track (without the vertex chamber hits). IPTR is the pointer for the track in the pattern bank to be updated, IPJHTL is the pointer to the 'JHTL' bank (passed to JFETCH) and the meaning of the real parameter ERRFAC determining the strength of the vertex constraint has been indicated above. The routine i called from XYRFTV for each track in a loop but you may as well call it yourself for selecte tracks only, with possibly different requests concerning the vertex constraint. Like any rou

```
C NV = TOTAL NUMBER OF VERTICES
C IV(1) = FLAG (0 = NO VERTEX FIT
C                1 = BAD VERTEX FIT
C                2 = VERTEX OF 1- OR COLLINEAR 2-PRONG
C                3 = GOOD VERTEX FIT
C                4 = E+E- PAIR VERTEX
C                5 = ISOLATED SINGLE TRACK VERTEX)
C V( 2) = X .
C V( 3) = Y . VERTEX COORDINATES
C V( 4) = Z .
C V( 5) = DX .
C V( 6) = DY . ERROR OF X,Y,Z
C V( 7) = DZ .
C IV(8) = NTR NUMBER OF TRACKS USED IN VERTEX FIT
C V( 9) = CHI2 CHISQARE OF FIT (N.D.F. = 2NTR-3)
C IV(10)= NTRALL NUMBER OF TRACKS BELONGING TO THIS VERTEX
C
C NEW VERTEX RESULTS ( 11 - 13 ) BY KLE
C -------------------
C V(11) = COVXY FOR VERTEX
C V(12) = COVXZ FOR VERTEX
C V(13) = COVYZ FOR VERTEX
C
C (13-26) 2.  VERTEX
C ......  ..........
C
C INTERNAL PARAMETERS
C ===================
C IT(1) IS SET NEGATIV TEMPORARILY IF TRACK BELONGS TO VERTEX
C T(16) = COULOMB SCATTERING ERROR ( TANKWALL )
C T(17) = COULOMB SCATTERING ERROR ( NEW BEAMPIPE ONLY )
C T(18) = S TO TANKWALL NEAR
C T(19) = S TO BEAMPIPE NEAR
C T(20) = PROJ. TRACKLENGTH IN RPHI
C T(21) = SIN(PHIO)
C T(22) = COS(PHIO)
C T(23) = TAN(THETA)
C T(24) = COS(THETA)
C T(25) UNUSED
C T(26) = S TO TANKWALL FAR
C T(27) = S TO TANKWALL NEAR
C T(28) = S.D. X
C T(29) = S.D. Y
C T(30) = S.D. Z
C
C NEW INTERNAL PARAMETERS ( 31 - 40 ) BY KLE
C ----------------------
```

Also available are routines for packing the common `/CKOMAK/` into a BOS bank KOKS (subroutine BKKOKS),and also unpacking the bank KOKS into the common `/CKOMAK/` (subroutine KOKSRE(IERROR)).

Disclaimer.

Most of the routines in this package were written a long time ago and to the remaining authors of this note who have not prudently emigrated to California, resemble something of a black box. While a certain amount of experience with the package has been gained, and willing as we may be to help the user with any problems that may be encountered, we do not consider ourselves to be legally liable for the results produced by KOMAKS. For any really interesting problems the only solution may be to have a look at the source code. However be warned, this is not a pretty sight and in some of the routines the rare comments are in Japanese!

Happy hunting.

Failure to meet these requirements will probably result in junk results, 0C4's and all sorts of nasty retributions. The call to KOMAKS has the form

`CALL KOMAKS(IPRINT,IKOMA,ITRY,IERRK,IEEFG)`

where the arguments are of essentially an administrative nature and will be described later.

There are several parameters available to the user to enable her to steer the V° selection by KOMAKS, and although suggested values will be provided in this note, it is recommended that the user experiments a little in order to tune these parameters to her particular case. The values given here would be appropriate in a high multiplicity scenario, e.g. multihadronic events.

The communication of steering parameters and results between the user and KOMAKS is made via the common `/CKOMAK/`. For historical reasons the maximum number of vertices which can be stored in the common is limited to 50.

```
COMMON/CKOMAK/NMAS,PCOMB1(4,50),PCOMB2(4,50),HTR1(50),
      HTR2(50),VMAS(50),DMAS(50),DVLKO(50),DKO(50),XYZKO(3,50),
      STDEV,ARLIM,IZRFIT,MNHIT,PCUTT,DVLMIN,DKOMAX,IEEFG3
```

The steering parameters are the following (recommended values in parentheses).

`STDEV` (3.0): This has a somewhat complicated meaning. As stated previously it is possible to exclude those V° candidates which have one or both tracks being consistent, within scattering errors, with coming from the event vertex. `STDEV` is a factor which can be used to increase (`STDEV` $>$ `1.0`), or decrease (`STDEV` $<$ `1.0`) the size of the errors on each track. Thus the larger the value of `STDEV` the 'clearer' the separation between event vertex and secondary vertex.

`ARLIM` (4.0): Minimum closest approach in $r\phi$ of a track to the run vertex (mm).

`IZRFIT` (1): Flag to choose whether or not to perform the common z fit (1= perform, 0= do not perform).

`MNHIT` (24): Minimum number of hits required to be associated with a track in the $r\phi$ plane.

`PCUTT` (0.1): Minimum track momentum (GeV).

`DVLMIN` (10.0): Minimum $r\phi$ decay length of a V° candidate (mm).

`DKOMAX` (10.0): Maximum of the minimum distance between the event vertex and the V° flight path in the $r\phi$ plane (mm).

`IEEFG3` (1): Flag to select only those vertices whose tracks are inconsistent with coming from the run vertex (1= select, 0= no selection). If `IEEFG3` `= 0` then the value of `STDEV` is irrelevant.

The results of the V° search are given by:

`NMAS` : The number of secondary vertices.

Run	REDUCONE Tape	JETC Cal.	LG Cal.
24214 - 26371	JADEOL.RED1HH.G1244V00 - JADEOL.RED1HH.G1347V00	final	extr.
26372 - 28738	JADEPR.REDUCONE.G1365V00 - JADEPR.REDUCONE.G1519V00	prel.	extr.
28739 - 28746	JADEPR.REDUCONE.G1520V00	final	final
28747 - 29187	JADEPR.REDUCONE.G1521V00 - JADEPR.REDUCONE.G1548V00	prel.	extr.
29188 - 29339	JADEPR.REDUCONE.G1549V00 - JADEPR.REDUCONE.G1559V00	final	extr.
29340 - 29899	JADEPR.REDUCONE.G1567V00 - JADEPR.REDUCONE.G1599V00	final	final
29900 - 30397	JADEPR.REDUCONE.G1608V00 - JADEPR.REDUCONE.G1642V00	final	extr.

Tapes 1549 - 1559 and 1608 - 1642 were directly produced at Heidelberg. No effort was made to eliminate non-physics runs from these tapes. However, routines RUNFIX and EBEAM have been updated to inform the user of such runs. The summary information on REDUC1 output tapes is kept in member REDUCONE on 'JADEPR.TEXT'.

REDUC2 REDUC2 for 1986 had been started on the REDUC1 output tapes with the wrong calibration and had to be redone completely. The tapes that were produced from the calibrated files above are stored as F11OLS.REDUC22.G0xxxV00. They are subject to the same mix of calibrations as the REDUCONE output tapes.