```
JADE Computer Note 60                          This is JADEPR.TEXT(NOTE60)
=========================

P. Steffen, 82/08/12


         Refit Tracks with Run Vertex Constraint
        ===========================================
          SUBROUTINE REFITV(IPTR,IPJHTL,ERRFAC)

This subroutine fits a parabola to the hits of a track in the  PATR-bank
in the x-y-projection. The run vertex from the common CALIBR  (see  JADE
Computer Note 59) is used as an additional constraint.

Arguments:
    IPTR   : pointer to th 0-th word of the track data in the PATR-bank,
             (e.g. idata(iptr+1) = number of the track)
    IPJHTL : BOS-pointer to JHTL-bank
    ERRFAC : factor to increase or decrease the standard  error  of  the
             vertex :
             =   1.: full vertex constraint
             =1000.: essentially no vertex constraint

Concerning the errors the following assumptions give reasonable results:
    sigma(hits  ) = 0.200 mm
    sigma(vertex) = sqrt( 0.300**2 + sig(mult.scatt.)**2 ) mm

The error of the vertex can be increased by the external factor  ERRFAC:
    ERRFAC =    1. --> full vertex constraint and best chamber
                       calibration is used:
                       momentum resolution (mu-pairs, spring 82)
                       improved by a factor of 2
    ERRFAC = 1000. --> no vertex constraint, but best chamber
                       calibration is used:
                       momentum resolution (mu-pairs, spring 82)
                       improved by a factor of 1.4

The measured hits are subject  to  all  corrections  described  in  JADE
Computer Note 45.

The track data with the results of the  fit  are  stored  in  the  array
WRK(HPTR0+0,1,2,...) / IWRK( ) in the common  /CWORK/  (use  the  macros
CWORKPR and CWORKEQ). The track data in CWORK is a  copy  of  the  track
data in the PATR-bank. Only the fit parameters, start and end point  and
directions are replaced by the new results. The program identifier (2-nd
word of track data) is set to 32 in order to identify tracks fitted with
this program.

Exemple for use of this program:
         Use  %MACRO CWORKPR and %MACRO CWORKEQ
         of macro library (F11GOD.PATRECSR)
C-------------------------------------------------
C  MACRO CWORKPR .... PATTERN RECOGNITION CWORK
C-------------------------------------------------
      COMMON /CWORK/ HPLAST,HPFREE,HPWRK(30),ADWRK(600),
     ,               HPRO,HNTR,HNTCEL(98),IPCL(200),NRHT(200),
     ,               NWR1(200),DS1(200),SL1(200),
     ,               NWR2(200),DS2(200),SL2(200),
     ,               LBL(200),NTREL(200),ICRO(200),
     ,               NTR,HNREL(100),HISTR(9,100),HRES(168),
     ,               NTRLM,RLMTR(3,5),
     ,               WRK(7000)
                     DIMENSION TRKAR(200,11),ITRKAR(200,11),
     ,                         LMRTR(3,5)
                     EQUIVALENCE (IPCL(1),TRKAR(1,1),ITRKAR(1,1))
                     EQUIVALENCE (LMRTR(1,1),RLMTR(1,1))
          DIMENSION IWRK(7000),HWRK(14000),IDWRK(600),HDWRK(1200)
                     EQUIVALENCE (IWRK(1),WRK(1),HWRK(1))
                     EQUIVALENCE (IDWRK(1),ADWRK(1),HDWRK(1))
C---------- END OF MACRO CWORKPR --------------
C-------------------------------------------------------
```



```
C  MACRO CWORKEQ .... PATTERN RECOGNITION CWORK POINTERS
C-------------------------------------------------------
      EQUIVALENCE
C                 POINTERS FOR FXYZ HIT ARRAY .. PRIMARY L/R SOLUTION
     #           (HPHT0,HPWRK( 1)),(HPHT9,HPWRK( 2)),(HLDHT,HPWRK( 3))
C                  POINTERS FOR CWORK SINGLE TRACK PATR BANK
     #          ,(HPTR0,HPWRK( 4)),(HPTR9,HPWRK( 5)),(HLDTR,HPWRK( 6))
C                  POINTERS FOR TRACK ELEMENT HIT LABEL ARRAY
     #          ,(HPHL0,HPWRK( 7)),(HPHL9,HPWRK( 8)),(HLDHL,HPWRK( 9))
C                  POINTERS FOR FXYZ HIT ARRAY .. OPPOSITE L/R SOLUTION
     #          ,(HPHT0A,HPWRK(10)),(HPHT9A,HPWRK(11)),(HLDHTA,HPWRK(12))
C                 POINTER LIMIT ON FXYZ HIT ARRAY
     #          ,(HPHTLM,HPWRK(13))
C                  POINTERS FOR
     #          ,(HPTE0,HPWRK(14)),(HPTE9,HPWRK(15)),(HLDTE,HPWRK(16))
C-------------- END OF MACRO CWORKEQ ------------------


C        GET RUN VERTEX
C                                                 USE MACRO CALIBR
         IPVTX = ICALIB(10)
         XV    = ACALIB(IPVTX+1)
         YV    = ACALIB(IPVTX+3)
C        GET POINTER TO JHTL_BANK
         IPJHTL= IDATA(IBLN('JHTL'))

         loop over all tracks (and select special ones)
C
C           CHECK IF MOMENTUM >100 MEV
            IF(ABS(ADATA(IPTR+25)).GT..00150) GOTO 900
C
C           GET DISTANCE OF TRACK FROM VERTEX
            CALL DRTRCK(IPTR,XV,YV,DR0)
C
C           CHECK IF DISTANCE TO VERTEX IS < 25 MM
            IF(ABS(DR0).LT.25.) GOTO 900
C
C           FIT TRACK WITH VERTEX CONSTRAINT
            CALL REFITV(IPTR,IPJHTL,1.0)
            IP    = HPTR0 - 1
            SIG   =  WRK(IP+23)
            SIG0  = ADATA(IPTR+23)
            ANHT  = IWRK(IP+24)
            IF(LAND(IWRK(IP+ 4),16).NE.0) ANHT = ANHT * 1.5
            ANHT0 = IDATA(IPTR+24)
            IF(SIG0.LT.SIG .AND. SIG.GT..35) GOTO 900
 C
C           REPLACE OLD RESULT IN PATR_BANK
            CALL MVC(IDATA(IPTR+1),0,IWRK(HPTR0),0,192)
C
  900    CONTINUE
```

One can simplify the procedure by just calling FITEVR (see JADE Computer Note 61). This subroutine tries to refit all tracks which originate from a position of <25mm from the run vertex. The new result replaces the old one in the PATR bank.