CDF/MEMO/SEC VTX/GROUP/822
(Formerly CDF/VXS Note No. 6)
Aug. 6, 1986

TRACK EXPANSION ROUTINES

AESOOK BYON

Track will be expanded at the point close to the vertex point we store the information of such expansion and other necessary data in the proper work banks for later use.

This note will describe the existing track expansion routines which will receive the track selection results and fill the VSFO or VSF1 bank for selected track numbers.

1) SUBROUTINE VSOEXP (IEXP,VEXP)

This routine will do track expansion without steering track parameters and fill the VSFO bank which contains cartesian coordinates and weight matrix of expansion point and direction of track at the expansion point.
User should call this subroutine after track selection then it will check fitcode for each track number and call proper other subroutines (VSPARn and VSOCAL) to fill VSFO bank.

Input Arguements:

IEXP (I*4) = expansion code
VEXP (R*4) = initial expansion parameter

Output Arguements:

None

2) SUBROUTINE VS1EXP (IEXP,VEXP)

This routine will do track expansion with steering track and fill the VSF1 bank which contains expansion point coordinates, momentum coordiantes, weight matrix of track parameters and partial derivative matrices of track parameters with respect to expansion coordinates and the momentum coordinates.
User should call this subroutine after track selection then it will check fitcode for each track number and call proper other subroutines (VSPARn and VS1CAL) to fill VSF1 bank.

Input Arguements:

IEXP (I*4) = expansion code
VEXP (R*4) = initial expansion parameter

Output Arguements:

None

3) SUBROUTINE VSPAR1 (INDDAT,IDIMV,IEXP,VEXP,D,PR,XI,VI,QI,XO, YO,S,R,Q)

This routine will calculate cartesian coordinates and momentum coordinates at the expansion point and should be called when fitcode has standard polar coordinate helix parameters.
In this routine (or in any other VSPARn routines), protection for small R case has been set such that

```
        Without steering case
                if R<|D| and |D|>0.5cm then reset R=|D|
                if R<|D| and |D|<0.5cm then reset R=0.5cm

        With steering case
                if R<=|D| then R=0.5*|D|
```

Input Arguements:

```
        INDDAT(I*4) = data index
        IDIMV (I*4) = dimension of view
        IEXP  (I*4) = expansion code
        VEXP  (R*4) = initial expansion parameter
        D     (I*4) = flag which tells with(=1) or without(=0)
                      steering case
```

Output Arguements:

```
        PR(5) (R*4) = helix parameters
        XI(3) (R*4) = array for expansion point coordinate
        VI(3) (R*4) = array for direction of the track
                      at the expansion point
        QI(3) (R*4) = array for momentum at the expansion
                      point
        XO    (R*4) = x coordinate of circle center
        YO    (R*4) = y coordinate of circle center
        S     (R*4) = distance of center of circle from origine
        R     (R*4) = radius  of track circle in the transverse
                      plane
        Q     (I*4) = charge of the particle
```

4) SUBROUTINE VSOCAL(INDDAT;IDIMV,PR,XI,R,GIJ)

This routine calculates the weight matrix of the expansion point - without steering case.

Input Arguements:

```
        INDDAT(I*4) = data index
        IDIMV (I*4) = dimension of view
        PR(5) (R*4) = helix parameters
        XI(3) (R*4) = expansion point coordinates
        R     (R*4) = radius of track circle in transverse plane
```

Output Arguements:
```
        GIJ(6)(R*4) = weight matrix of expansion point in
                      triangular form
```

5) SUBROUTINE VS1CAL (INDDAT,IDIMV,PR,XI,VI,QI,XO,YO,S,R,Q,AIJ,
BIJ,CI,GIJ)

This routine will calculate weight matrix of track parameters, partial derivative of parameters with respect to expansion point coordinates and momentum coordinates and auxialiary parameters for chi-square expression.

Input Arguements:
```
        INDDAT(I*4) = data index
        IDIMV  (I*4) = dimension of view
        PR(5)  (R*4) = helix parameters
        XI(3)  (R*4) = expansion point coordinates
        VI(3)  (R*4) = direction of the track
        QI(3)  (R*4) = momentum coordinates at expansion point
        XO     (R*4) = x coordinate of circle center
        YO     (R*4) = y coordinate of circle center
        S      (R*4) = distance of center of circle from origine
        R      (R*4) = radius of track circle in transverse plane
        Q      (I*4) = charge of the particle
```

Output Arguements:
```
        AIJ(5,3)     = partial  derivatives  of  parameters  with
                       respect to expansion point coordinates
        BIJ(5,3)     = partial  derivatives  of  parameters  with
                       respect to momentum coordinates
        CI(3)        = auxialiary parameters for chi-square
        GIJ(15)      = weight matrix for chi-square
```

6) SUBROUTINE VSPARn (n>1)

For different case of parameter expressions other than VSPAR1 can handle, more subroutines need to be written.
These subroutines should contain output arguements of PR, XI, VI, QI, XO, YO, S, R and Q.
In any of VSPARn routines, protection for small R case must be provided such that

```
        Without steering case
                if R<|D| and |D|>0.5cm then reset R=|D|
                if R<|D| and |D|<0.5cm then reset R=0.5cm

        With steering case
                if R<=|D| then R=0.5*|D|
```

Track expansion routines have a following structure.

```
| VS0EXP (VS1EXP) |

DO N=1,NTRACK
     ↓
CHECK FITCODE
     ↓
CALL VSPAR1, VSPAR2 or VSPAR3...
     ↓
CALL VS0CAL (VS1CAL)
     ↓
CALL VSF0CR (VSF1CR)
     ↓
N = NTRACK ?        no → back to DO N=1,NTRACK
     ↓ yes
CALL VSW0CR (VSW1CR)
     ↓
RETURN TO CALLER
```

APPENDIX : RESULTS OF TRACK EXPANSION CALCULATION

For curved tracks provided an expansion point fairly close to the vertex, tangents to the tracks at a given expansion point will be used as input for the vertex fitting algorithm. So track will be expanded at the point close to the vertex point and we store the information of such expansion and other necessary data in the work bank VSFO and VSF1 for later use.

This note containes the results of track expansion calculations, which will be used as the input for the vertex fitter, in two different methodes - with and without steering track parameters, and effects of multiple scattering contribution.

I. DEFINE SOME VARIABLES AND CONSTANTS

track parameters {Pi} = {P1, P2, P3, P4, P5}
= {COT, C, z0, D, PHIO}

COT = cotangent of polar angle
C = half curvature
z0 = z at the closest approach to beam axis
D = impact parameter
PHIO = azimuthal angle at D

R = radius of track circle in transverse plane
cov(P) = covariance matrix of track parameters
Q = charge of the particle
Bfield = magnetic field
K = conversion constant = 0.0015 * Bfield

expansion done at $\vec{x} = (x, y, z)$
with momentum $\vec{q} = (qx, qy, qz)$

then direction vector n is

$$\vec{n} = \vec{q} \ / \ qt = (qx/qt,\ qy/qt,\ qz/qt) = (nx,\ ny,\ nz)$$

$$\text{where } qt = \sqrt{qx^2 + qy^2}$$

$$A = [RC+(1+CD)D/R] \ / \ (1+2CD)$$

$$B = C \sqrt{(R^2-D^2)/(1+2CD)}$$

$$PSI = 2*SIN^{-1}(B)$$

II. RELATIONS BETWEEN PARAMETERS

C = Q*K/qt

D = Q(S-Q/2C)

S = $\sqrt{x0^2 + y0^2}$ = Q(1+2CD)/2C

x0 = -(1+2CD)sin(PHI0)/2C = x - Q*qy/2K

y0 = (1+2CD)cos(PHI0)/2C = y + Q*qx/2K

z0 = z - COT*$\sin^{-1}$(B)/C

tan(PHI0) = -x0/y0

qx = 2KQ(y0-y) = 2C*qt(y0-y)

qy = -2KQ(x0-x) = -2C*qt(x0-x)

COT = qz/qt

III. FAST VERTEX FINDING (without steering)

This method will fill VSFO bank.

1. Calculate tangent line at expansion point

$\vec{x} = \vec{n}i\ s + \vec{x}i$ tangent equation

where $\vec{n}i$ = tangent direction
$\vec{x}i$ = expansion point

then minimize χ^2

$$\chi^2 = (\vec{x}i - \vec{x}v + \vec{n}i\ si)^t\ Gij\ (\vec{x}i - \vec{x}v + \vec{n}i\ si)$$

where $Gij = cov(\vec{x}i)^{-1}$

So calculate $\vec{n}i$, $\vec{x}i$, Gij values then use them as input for the vertex fitter.

2. We have {Pi}, R (or Z0, PHI), cov(P), Q, and Bfield so first calculate A, B, S, PSI then

$$\vec{xi}\begin{cases} x = [\sin(\mathrm{PSI}+\mathrm{PHI0})-(1+2CD)\sin(\mathrm{PHI0})]/2C \\ \quad = [\cos(\mathrm{PHI0}+\mathrm{PSI}/2)\sin(\mathrm{PSI}/2)]/C-D\sin(\mathrm{PHI0}) \\ y = -[\cos(\mathrm{PSI}+\mathrm{PHI0})-(1+2CD)\cos(\mathrm{PHI0})]/2C \\ \quad = [\sin(\mathrm{PHI0}+\mathrm{PSI}/2)\sin(\mathrm{PSI}/2)]/C+D\cos(\mathrm{PHI0}) \\ z = z0 + \mathrm{COT}*\mathrm{PSI}/2C \end{cases}$$

$$x0 = -S\ \sin(\mathrm{PHI0})/Q$$
$$y0 = S\ \cos(\mathrm{PHI0})/Q$$

$$\vec{ni}\begin{cases} nx = 2C(y0-y) \\ ny = -2C(x0-x) \\ nz = \mathrm{COT} \end{cases}$$

$$[\mathrm{cov}(x)]ij = \sum_{k,l} (dxi/dPk)[\mathrm{cov}(P)]kl\ (dxj/dPl)$$

$$Gij = [\mathrm{cov}(x)]^{-1}$$

so need to find (dxi/dPk)

$$dx/dC = -[\cos(\mathrm{PHI0}+\mathrm{PSI}/2)\sin(\mathrm{PSI}/2)]/C^2$$

$$dx/dD = -\sin(\mathrm{PHI0})$$

$$dx/d\mathrm{PHI0} = -y$$

$$dx/dz0 = 0$$

$$dx/d\mathrm{COT} = 0$$

$$dy/dC = -[\sin(\mathrm{PHI0}+\mathrm{PSI}/2)\sin(\mathrm{PSI}/2)]/C^2$$

$$dy/dD = \cos(\mathrm{PHI0})$$

$$dy/d\mathrm{PHI0} = x$$

$$dy/dz0 = 0$$

$$dy/d\mathrm{COT} = 0$$

$$dz/dC = -\mathrm{COT}*\mathrm{PSI}/2C^2$$

$$dz/dD = 0$$

$$dz/d\mathrm{PHI0} = 0$$

$$dz/dz0 = 1$$

$$dz/d\mathrm{COT} = \mathrm{PSI}/2C$$

IV. WITH TRACK PARAMETER STEERING

This method will fill VSF1 bank.

1. For given R calculate x, q, and two (5,3) matrices A(i,j) and B(i,j)

$$A(i,j) = \frac{dPi}{dxj} \qquad B(i,j) = \frac{dPi}{dqj}$$

then true track parameters are approsimated by

$$\tilde{\vec{P}}i = \vec{P0}^{(i)} + A(i,j)(xj - x0^{(i)}) + B(i,j)(qj - q0^{(i)})$$

and minimize χ^2

$$\chi^2 = \sum_j (P0^{(i)} - \tilde{P}i)^t \, Gij \, (P0^{(i)} - Pi)$$

$$= \sum_j (Cj - A(i,j)xj - B(i,j)qj)^t \, Gij \, (Cj - A(i,j)xj - B(i,j)qj)$$

where $Cj = A(i,j)x0^{(i)} + B(i,j)q0^{(i)}$

So calculate $\vec{x}$, $\vec{q}$, A(i,j) B(i,j) and Cj values to use as input for the vertex fitter.

2. Results for xi and qi will be same as in without steering case.

A(1,1) = dCOT/dx = 0
A(1,2) = dCOT/dy = 0
A(1,3) = dCOT/dz = 0

A(2,1) = dC/dx = 0
A(2,2) = dC/dy = 0
A(2,3) = dC/dz = 0

$$A(3,1) = \frac{dz0}{dx} = \frac{RCA*COT*A(4,1)}{B\sqrt{1-B^2}\,(1+2CD)}$$

A(3,2) = dz0/dy = A(3,1)*y0/x0
A(3,3) = dz0/dz = 1

A(4,1) = dD/dx = -sin(PHI0)
A(4,2) = dD/dy = cos(PHI0)
A(4,3) = dD/dz = 0

A(5,1) = dPHI0/dx = $-y0/S^2$
A(5,2) = dPHI0/dy = $x0/S^2$
A(5,3) = dPHI0/dz = 0

B(1,1) = dCOT/dqx = -nx*COT*C/(QK)
B(1,2) = dCOT/dqy = B(5,1)*ny/nx
B(1,3) = dCOT/dqz = C/(QK)

B(2,1) = dC/dqx = $-nx*C^2/(QK)$
B(2,2) = dC/dqy = B(2,1)*ny/nx
B(2,3) = dC/dqz = 0

B(3,1) = dz0/dqx

$$= (z0-z)\left[\frac{B(1,1)}{COT} - \frac{B(2,1)}{C}\right] - \frac{COT}{C\sqrt{1-B^2}}\left[\frac{B}{C}\left(1-\frac{QD}{2S}\right)B(2,1) - \frac{QRCA}{2SB}\,B(4,1)\right]$$

B(3,2) = dz0/dqy

$$= (z0-z)\left[\frac{B(1,2)}{COT} - \frac{B(2,2)}{C}\right] - \frac{COT}{C\sqrt{1-B^2}}\left[\frac{B}{C}\left(1-\frac{QD}{2S}\right)B(2,2) - \frac{QRCA}{2SB}\,B(4,2)\right]$$

B(3,3) = dz0/dqz = (z0-z)*B(1,3)/COT

B(4,1) = dD/dqx = [A(4,2)-nx]*Q/(2K)
B(4,2) = dD/dqy = -[A(4,1)+ny]*Q/(2K)
B(4,3) = dD/dqz = 0

B(5,1) = dPHI0/dqx = A(5,2)*Q/(2K)
B(5,2) = dPHI0/dqy = -A(5,1)*Q/(2K)
B(5,3) = dPHI0/dqz = 0

then

$$Ci = \sum_{j}^{3} A(i,j)xj + B(i,j)qj$$

and

$$Gij = [\mathrm{cov}(P)]^{-1}$$

* Protection of small R

The value of R can not be less than distance of impact parameter(|D|) so protection of the case when R < |D| is needed.

1. Without steering case (filling VSF0 bank)

None of calculating values are singular. But inverse of the covariance matrix (COVX) needs to be calculated and since determinant of COVX is a order of D**2, if |D| is very small, inverse of COVX becomes very large(-10**12).
To prevent this,

1) when R<|D| and |D|>0.5cm, reset R=|D|
2) when R<|D| and |D|<0.5cm, reset R=0.5cm

2. With steering case (filling VSF1 bank)

If R=|D| then values of AIJ(3,1),AIJ(3,2),BIJ(3,1),BIJ(3,2) becomes singular. To prevent this with protection of small R

when R<|D| or R=|D|, reset R=1.5*|D|