DR. 1036

LA-8271-MS

Informal Report

2 cy 4/11/80

MASTER

A Computer Program to Fit a Hyperellipse to a Set of Phase-Space Points in as Many as Six Dimensions

University of California

LOS ALAMOS SCIENTIFIC LABORATORY
Post Office Box 1663 Los Alamos, New Mexico 87545

A COMPUTER PROGRAM TO FIT A HYPERELLIPSE TO A SET OF PHASE-SPACE POINTS IN AS MANY AS SIX DIMENSIONS

E. Alan Wadlinger

ABSTRACT

We have written and tested a computer program that will fit a hyperellipse to a set of phase-space points in as many as 6 dimensions. The weight assigned to the phase-space points can be varied as a function of their distance from the centroid of the distribution. By varying the weight, we can determine whether there is a difference in ellipse orientation between inner and outer particles. This program should be useful in studying the effects of longitudinal and transverse phase-space couplings.

I. DERIVATION OF THE EQUATIONS

The equation of a hyperellipse in n dimensions can be written as[1]

$$X^T B X = (V_n/K_n)^{2/n} \tag{1}$$

where $\underline{B}$ is an n x n dimensional, symmetric matrix having unit determinant, X is the n-dimensional vector of the hyperellipse coordinates, V_n is the n dimensional hyperellipse volume, and K_n is the constant (Γ is the gamma function)

$$K_n = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2} + 1\right)} \tag{2}$$

Given a set of k phase-space points, we wish to find the shape parameters of a hyperellipse that characterizes the phase-space distribution; that is, the elements of matrix $\underline{B}$. Define the *function*

$$I = \sum_{i=1}^{k} \left(V_i/K\right)^m + \frac{nm}{2} \lambda \; (|B|-1)$$

$$= \sum_{i=1}^{k} \left(X_i^T \; B \; X_i\right)^{nm/2} + \frac{nm}{2} \; \lambda(|B|-1) \tag{3}$$

The subscript i denotes the ith phase-space point, m is an arbitrary power of the volume, and λ is a Lagrange multiplier. The matrix $\underline{B}$ is symmetric and its determinant is denoted by $|B|$.

Minimize I with respect to λ and the elements of B (B_{jk} with $j \leq k = 1, \ldots, n$).

$$\partial I/\partial B_{jk} = \frac{nm}{2} \left[\sum_i \left(X_i^T \; BX_i\right)^{\frac{nm}{2} -1} \; X_i^j \; X_i^k \; \left(2 - \delta_{jk}\right) + \lambda \left(\frac{\partial |B|}{\partial B_{jk}}\right)\right] = 0 \tag{4}$$

$$\partial I/\partial\lambda = \frac{nm}{2} \; (|B| \; -1) = 0 \tag{5}$$

A vector component of X is denoted by superscripts.

The derivative of the determinant of a matrix by one of its elements is the cofactor of that element.

$$\partial|B|/\partial B_{jk} = C_{jk} + \left(1- \delta_{jk}\right) C_{kj} = \left(2- \delta_{jk}\right) C_{jk}$$

where C_{jk} is the cofactor of B_{jk} and is symmetric with respect to an interchange of the indices j and k (because $B_{jk} = B_{kj}$, $C_{jk} = C_{kj}$).

Delta (δ) is the Kronecker delta symbol.

Define

$$b_i = X_i^T \; B \; X_i \tag{6}$$

then,

$$\partial I/\partial B_{jk} = \frac{nm}{2}(2-\delta_{jk})\left[\sum_i b_i^{\frac{nm}{2}-1} X_i^j X_i^k + \lambda C_{jk}\right] \tag{7}$$

$$\partial I/\partial\lambda = \frac{nm}{2}(|B|-1) \tag{8}$$

where ($j \leq k = 1, \ldots, n$ because of the symmetry of $\underline{B}$).

We eliminate λ by using the equation for $j = k = 1$

$$\lambda = -\frac{1}{C_{11}} \sum_i b_i^{\frac{nm}{2}-1}\left(X_i^1\right)^2 \tag{9}$$

Define

$$J_{jk} = \frac{C_{11}}{\frac{nm}{2}(2-\delta_{jk})}\,\partial I/\partial B_{jk} \tag{10}$$

$$J_\lambda = \frac{1}{nm/2}\,\partial I/\partial\lambda \tag{11}$$

then,

$$J_{jk} = \sum_i b_i^{\frac{nm}{2}-1}\left[C_{11} X_i^j X_i^k - C_{jk}\left(X_i^1\right)^2\right] \tag{12}$$

$$J_\lambda = |B|-1 \tag{13}$$

where $j \leq k = 1, \ldots, n$ and $j + k \neq 2$.

Iterate to solve these equations, using Newton's method. Given the results for the tth iteration, find the results for the $(t+1)^{th}$.

$$J_{jk}^{t+1} = J_{jk}^t + \sum_{r\leq s=1}^{n} \partial J_{jk}/\partial B_{rs}\left(B_{rs}^{t+1} - B_{rs}^t\right) \tag{14}$$

$$J_\lambda^{t+1} = J_\lambda^t + \sum_{r\leq s=1}^{n} \partial J_\lambda/\partial B_{rs}\left(B_{rs}^{t+1} - B_{rs}^t\right) . \tag{15}$$

Find new values for the $\underline{B}$ matrix elements by setting J_{jk}^{t+1} and J_{λ}^{t+1} to zero. *We therefore solve the set of equations*

$$\sum_{r\leq s=1}^{n} \partial J_{jk}/\partial B_{rs} \Big|_{B=B^t} B_{rs}^{t+1} = \sum_{r\leq s=1}^{n} \partial J_{jk}/\partial B_{rs}\Big|_{B=B^t} - J_{jk}^{t} \tag{16}$$

$$\sum_{r\leq s=1}^{n} \partial J_{\lambda}/\partial B_{rs} \Big|_{B=B^t} B_{rs}^{t+1} = \sum_{r\leq s=1}^{n} \partial J_{\lambda}/\partial B_{rs}\Big|_{B=B^t} - J_{\lambda}^{t} \tag{17}$$

for B^{t+1}.

Define

$$C_{jk,rs} = \partial C_{jk}/\partial B_{rs},$$

which is the rs cofactor of C_{jk}. Because B is symmetric, we can interchange the row and column indices ($C_{jk,rs} = C_{kj,sr}$) but note that in general $C_{jk,rs} \neq C_{jk,sr}$. We calculate the derivatives in Eqs. (16) and (17) using Eqs. (12) and (13)

$$\partial J_{jk}/\partial B_{rs} = (\tfrac{nm}{2} -1)(2 - \delta_{rs}) \sum_i b_i^{\frac{nm}{2}-2} X_i^r X_i^s \left[C_{11}\, X_i^j X_i^k - C_{jk}\left(X_i^1\right)^2\right] \tag{18}$$

$$+ \sum_i b_i^{\frac{nm}{2}-1}\left[\left(2 - \delta_{rs}\right) C_{11,rs}\, X_i^j X_i^k - \left(C_{jkrs} + \left(1 - \delta_{rs}\right) C_{jksr}\right)\left(X_i^1\right)^2\right]$$

$$\partial J_{\lambda}/\partial B_{rs} = (2 - \delta_{rs})C_{rs} \quad . \tag{19}$$

Define the matrix $\underline{A}$ and vectors U and V by

$$\underline{A} = \begin{bmatrix} \partial J_{jk}/\partial B_{rs} \\ \partial J_{\lambda}/\partial B_{rs} \end{bmatrix} \tag{20}$$

$$U = \begin{bmatrix} B_{rs} \end{bmatrix} \tag{21}$$

$$V = \begin{bmatrix} J_{jk} \\ J_{\lambda} \end{bmatrix} \tag{22}$$

where $j \leqslant k = 1, \ldots, n$, $j + k \neq 2$ and $r \leqslant s = 1, \ldots, n$. For $\underline{A}$, jk is a row index and rs a column index. For U and V, both jk and rs are row indices. Equations (16) and (17) can be written as the matrix equations.

$$A^t \; U^{t+1} = A^t \; U^t - V^t \tag{23}$$

where the superscripts denote the iteration. Solve these equations by multiplying Eq. (23) on the left by the inverse of A^t, $(A^t)^{-1}$.

$$U^{t+1} = U^t - (A^t)^{-1} \; V^t \quad . \tag{24}$$

Figure 1 contains the computer listing that solves this problem. This routine requires the subroutine COFAC, which is listed in Figure 2.

Case $\frac{nm}{2} = 1$

If we force nm/2 to equal 1 in Eq. (3), the problem simplifies considerably. Equations (12) and (13) become

$$J_{jk} = \sum_i \left[C_{11} \; X_i^j X_i^k - C_{jk} \left(X_i^1 \right)^2 \right] \tag{25}$$

$$J_{\lambda} = |B| - 1 \tag{26}$$

```
      SUBROUTINE HELIPS(NP,NS,AEXP,X,B,E,XO)
C      WRITTEN BY E.A.WADLINGER 8-78
C      UPDATED BY E.A.W. 10/79, TO FIT ARBITRARY POWERS
C      OF THE VOLUME
C
C      FIT A HYPERELLIPSE TO A SET OF PHASE SPACE POINTS
C      MINIMIZE THE SUM (X(I,J)B(I,K)X(K,J))**(AEXP*NS/2)
C      - (E/K)**AEXP, WITH DET(B) - 1.
C      SEE IEEE TRANS ON NUCLEAR SCIENCE,
C      VOL. NS-26, NO. 3, P3511, (1979).
C      MAXIMUM OF 6 DIMENSIONS IN HYPERSPACE AND 2000 PARTICLES
C
C      INPUT PARAMETERS
C        NP - NO OF POINTS
C        NS - DIMENSION OF HYPERSPACE
C        X - COORDINATES OF POINTS;  DIMENSION X(NS,NP)
C        AEXP - POWER OF THE VOLUME USED IN THE FIT
C             - 0.0, POWER - 2/NS
C             SEE STARRED SECTION FOR LOGIC
C      OUTPUT PARAMETERS
C        B - ELLIPSE DEFINING MATRIX
C        E - VOLUME OF THE ELLIPSE
C        XO - CENTROID OF THE DISTRIBUTION
C
C      SUBROUTINES REQUIRED
C        COFAC - PARTIAL DERIVATIVES OF A DETERMINANT
C        DETERM - DETERMINANT OF A MATRIX
C        MATINV - MATRIX INVERSION
C
      DIMENSION X(6,2000),B(6,6),XO(6)
      DIMENSION AJ(21),BB(6,6),BX(2000),BXX(2000)
      DOUBLE PRECISION A(21,21)
C
C      DETERMINE THE CENTROID XO OF THE DISTRIBUTION
C      SHIFT THE PHASE POINTS BY XO
      DO 10 I - 1,NS
      SUM - 0.
      DO 20 J - 1,NP
   20 SUM - SUM + X(I,J)
   10 XO(I) - SUM / NP
      DO 30 I - 1,NP
      DO 30 J - 1,NS
   30 X(J,I) - X(J,I) - XO(J)
C
C      DETERMINE THE STARTING VALUES
      DO 110 I - 1,NS
      DO 110 J - I,NS
      B(I,J) - 0.0
      B(J,I) - 0.0
      IF(I .EQ. J)B(I,I) - 1.0
  110 CONTINUE
      EOLD - 0.0
C
C      ************
C      SOME OF THE NEXT SEVERAL STEPS ARE FOR PROGRAM PROTECTION
C      SEE STATEMENT LABLED BY 2000
C      WE AVOID THE POSSIBILITY OF RAISING A NEGATIVE NUMBER TO A
C      REAL POWER DURING AN INTERMEDIATE STEP IN THE SEARCH
      NTST - NS*AEXP - 2.0
      IF(AEXP .EQ. 0.0)NTST - 0
      POWR - (NS*AEXP - 4.0) * 0.5
      IF(POWR .LT. 0.0)POWR - 0.0
      NPOWR - POWR
      IF(NTST .NE. 0)NTST - (NPOWR + 1)*2
C      ************
C
C      NOW FIT
  200 IJ - 0
C      CALCULATE THE SUM(K,L) OF X(K,I)B(K,L)X(L,I)
      IF(NTST .EQ. 0)GO TO 405
      DO 410 I - 1,NP
      SUM - 0.0
      DO 420 K - 1,NS
      DO 420 L - 1,NS
  420 SUM - SUM + X(K,I)*B(K,L)*X(L,I)
 2000 BX(I) - SUM**NPOWR
  410 BXX(I) - BX(I) * SUM
      GO TO 4
  405 DO 425 I - 1,NP
  425 BXX(I) - 1.0
C      CALCULATE A AND AJ
  415 IJ - 0
      SUM1 - 0.0
      DO 435 M - 1,NP
  435 SUM1 - SUM1 + BXX(M)*X(1,M)**2
      C11 - COFAC(NS,1,1,B,0,0,1)
      DO 430 I - 1,NS
      DO 430 J - I,NS
C      FIRST EQUATION WAS USED TO ELIMINATE LAMBDA
      IF((I+J) .EQ. 2)GO TO 430
      IJ - IJ + 1
      CIJ - COFAC(NS,I,J,B,0,0,1)
      SUM2 - 0.0
      DO 445 M - 1,NP
  445 SUM2 - SUM2 + BXX(M)*X(I,M)*X(J,M)
      KL - 0
      DO 440 K - 1,NS
      DO 440 L - K,NS
      KL - KL + 1
      C11KL - COFAC(NS,1,1,B,K,L,2)
      CIJKL - COFAC(NS,I,J,B,K,L,2)
      IF(K .NE. L)C11KL - 2.0*C11KL
      IF(K .NE. L)CIJKL - CIJKL + COFAC(NS,I,J,B,L,K,2)
      A(IJ,KL) - C11KL*SUM2 - CIJKL*SUM1
      IF(NTST .EQ. 0)GO TO 470
      SUM3 - 0.0
      SUM4 - 0.0
      DO 450 M - 1,NP
      SUM - BX(M)*X(K,M)*X(L,M)
      SUM3 - SUM3 + SUM*X(I,M)*X(J,M)
  450 SUM4 - SUM4 + SUM*X(1,M)**2
      SUM - C11*SUM3 - CIJ*SUM4
      IF(K .NE. L)SUM - 2.0*SUM
      A(IJ,KL) - A(IJ,KL) + 0.5*SUM*NTST
  470 CONTINUE
  440 CONTINUE
      AJ(IJ) - C11*SUM2 - CIJ*SUM1
      K - IJ + 1
      A(KL,K) - CIJ
      IF(I .NE. J)A(KL,K) - A(KL,K)*2.0
  430 CONTINUE
      IJ - IJ + 1
      DO 235 I - 1,NS
      DO 235 J - 1,NS
  235 BB(I,J) - B(I,J)
      AJ(IJ) - DETERM(BB,NS) - 1.0
      A(IJ,1) - C11
C      INVERT THE A MATRIX
      CALL MATINV(A,IJ,SUM)
C      NEW ELLIPSE PARAMETER MATRIX
      KL - 0
      DO 240 I - 1,NS
      DO 240 J - I,NS
      SUM - 0.
      KL - KL + 1
      DO 250 K - 1,IJ
  250 SUM - SUM + A(KL,K) * AJ(K)
      B(I,J) - B(I,J) - SUM
  240 B(J,I) - B(I,J)
C      CALCULATE THE NEW VOLUME
      SUM - 0.
      DO 310 I - 1,NS
      DO 310 J - 1,NS
      DO 310 K - 1,NP
  310 SUM - SUM + X(I,K)*B(I,J)*X(J,K)
      ENEW - SUM
      IF(ENEW .LT. 0.0)GO TO 200
      TST - 1.0
      IF(EOLD .EQ. 0.0)GO TO 320
      TST - (EOLD - ENEW) / ENEW
      IF(ENEW .GT. EOLD)TST - (ENEW - EOLD) / EOLD
  320 EOLD - ENEW
      IF(TST .GT. .001)GO TO 200
      E - ENEW / NP
      AN - NS / 2.0
      E - E**AN
      IF(NS .EQ. 2)E - 3.142 * E
      IF(NS .EQ. 3)E - 4.189 * E
      IF(NS .EQ. 4)E - 4.935 * E
      IF(NS .EQ. 5)E - 5.264 * E
      IF(NS .EQ. 6)E - 5.168 * E
C      RETURN THE PHASE SPACE POINTS BACK TO
C      THE ORIGINAL VALUES
      DO 40 I - 1,NP
      DO 40 J - 1,NS
   40 X(J,I) - X(J,I) + XO(J)
      RETURN
      END
```

Fig. 1. Program listing of HELIPS

```
      FUNCTION COFAC(N,I,J,A,K,L,MODE)
C      WRITTEN BY E.A.WADLINGER 8-78
C
C      MODE = 1, DETERMINE THE VALUE OF THE DERIVATIVE OF THE DETERMINANT
C           OF THE MATRIX A WRT A(I,J).  THIS IS THE COFACTOR OF THE
C           ELEMENT A(I,J)
C      MODE = 2, DETERMINE THE VALUE OF THE DERIVATIVE OF THE COFACTOR OF
C           A(I,J) WRT THE ELEMENT A(K,L)
C      N - ORDER OF MATRIX A; DIMENSION A(N,N)
C      I,J - SUBSCRIPT FOR A(I,J)
C      K,L - SUBSCRIPT FOR A(K,L)
C
C      SUBROUTINE REQUIRED DETERM - DETERMINANT OF A MATRIX
      DIMENSION A(6,6),B(6,6)
C
      IF(MODE-2)10,100,10
C
C      MODE = 1 (NOT 2)
   10 M = N - 1
      I2 = 0
      DO 20 I1 = 1,N
      IF(I1 .EQ. I)GO TO 20
      I2 = I2 + 1
      I3 = 0
      DO 30 I4 = 1,N
      IF(I4 .EQ. J)GO TO 30
      I3 = I3 + 1
      B(I2,I3) = A(I1,I4)
   30 CONTINUE
   20 CONTINUE
      I1 = I + J
      COFAC = (-1.)**I1 * DETERM(B,M)
      RETURN
C
C      MODE = 2
  100 IF(I .EQ. K .OR. J .EQ. L)GO TO 200
      M = N - 2
      IF(M .EQ. 0)GO TO 210
      I2= 0
      DO 120 I1 = 1,N
      IF(I1 .EQ. I .OR. I1 .EQ. K)GO TO 120
      I2 = I2 + 1
      I3 = 0
      DO 130 I4 = 1,N
      IF(I4 .EQ. J .OR. I4 .EQ. L)GO TO 130
      I3 = I3 + 1
      B(I2,I3) = A(I1,I4)
  130 CONTINUE
  120 CONTINUE
      I1 = I + J + K + L
      IF(I .LT. K)I1 = I1 - 1
      IF(J .LT. L)I1 = I1 - 1
      COFAC = (-1.)**I1 * DETERM(B,M)
      RETURN
  200 COFAC = 0.
      RETURN
  210 COFAC = 1.0
      IF(I .NE. J)COFAC = -1.0
      RETURN
      END
```

Fig. 2. Program listing of COFAC.

```
      SUBROUTINE HELIPS(NP,NS,X,B,E,X0)
C      WRITTEN BY E.A.WADLINGER 8-78
C      FIT A HYPERELLIPSE TO A SET OF PHASE SPACE POINTS
C      MINIMIZE THE SUM X(I,J)B(I,K)X(K,J) = (E/K)**(2/NS)
C      WITH DET(B) = 1
C      MAXIMUM OF 6 DIMENSIONS IN HYPERSPACE AND 500 PARTICLES
C
C      INPUT PARAMETERS
C        NP = NO OF POINTS
C        NS = DIMENSION OF HYPERSPACE
C        X = COORDINATES OF POINTS;  DIMENSION X(NS,NP)
C      OUTPUT PARAMETERS
C        B = ELLIPSE DEFINING MATRIX
C        E = VOLUME OF ELLIPSE
C        X0 = CENTROID OF THE DISTRIBUTION
C      SUBROUTINES REQUIRED
C        COFAC = PARTIAL DERIVATIVES OF A DETERMINANT
C        DETERM = DETERMINANT OF A MATRIX
C        MATINV = MATRIX INVERSION
C
      DIMENSION X(6,500),B(6,6),X0(6)
      DIMENSION XIJ(6,6),AJ(21),BB(6,6)
      DOUBLE PRECISION A(21,21)
C
C      DETERMINE THE CENTROID X0 OF THE DISTRIBUTION AND OTHER FIXED
C      CONSTANTS
      DO 10 I = 1,NS
      SUM = 0.
      DO 20 J = 1,NP
   20 SUM = SUM + X(I,J)
   10 X0(I) = SUM / NP
      DO 30 I = 1,NS
      DO 30 J = 1,NS
      SUM = 0.
      DO 40 L = 1,NP
      FX1 = X(I,L) - X0(I)
      FX2 = X(J,L) - X0(J)
   40 SUM = SUM + FX1*FX2
      XIJ(I,J) = SUM
   30 XIJ(J,I) = SUM
C
C      DETERMINE THE STARTING VALUES FOR THE ELLIPSE MATRIX
      DO 110 I = 1,NS
      DO 110 J = 1,NS
      B(I,J) = 0.0
      B(J,I) = 0.0
      IF(I .EQ. J)B(I,I) = 1.0
  110 CONTINUE
      EOLD = 0.0
C
C      NOW FIT
  200 IJ = 0
      DO 210 I = 1,NS
      DO 210 J = 1,NS
      IF((I+J) .EQ. 2)GO TO 210
      IJ = IJ + 1
      KL = 0
      DO 220 K = 1,NS
      DO 220 L = K,NS
      KL = KL + 1
      C1 = COFAC(NS,1,1,B,K,L,2)
      IF(K .NE. L)C1 = 2.*C1
      C2 = COFAC(NS,1,J,B,K,L,2)
      IF(K .NE. L)C2 = C2 + COFAC(NS,I,J,B,L,K,2)
  220 A(IJ,KL) = XIJ(I,J) * C1 - XIJ(I,1) * C2
  210 CONTINUE
      IJ = IJ + 1
      KL = 0
      TMP = COFAC(NS,1,1,B,0,0,1)
      DO 230 K = 1,NS
      DO 230 L = K,NS
      KL = KL + 1
      A(IJ,KL) = COFAC(NS,K,L,B,0,0,1)
      AJ(KL) = -XIJ(1,1)*A(IJ,KL) + TMP*XIJ(K,L)
      IF(K .NE. L)A(IJ,KL) = 2. * A(IJ,KL)
  230 CONTINUE
      DO 235 I = 1,NS
      DO 235 J = 1,NS
  235 BB(I,J) = B(I,J)
      J = IJ - 1
C      FIRST EOUATION WAS USED TO ELIMINATE LAMBOA
      DO 237 I = 1,J
  237 AJ(I) = AJ(I+1)
      AJ(IJ) = DETERM(BB,NS) - 1.0
      CALL MATINV(A,IJ,DETERMM)
C      NEW ELLIPSE PARAMETER MATRIX
      KL = 0
      DO 240 I = 1,NS
      DO 240 J = I,NS
      SUM = 0.
      KL = KL + 1
      DO 250 K = 1,IJ
  250 SUM = SUM + A(KL,K) * AJ(K)
      B(I,J) = B(I,J) - SUM
  240 B(J,I) = B(I,J)
C      CALCULATE THE NEW VOLUME
      SUM = 0.
      DO 310 I = 1,NS
      DO 310 J = 1,NS
  310 SUM = SUM + B(I,J) * XIJ(I,J)
      ENEW = SUM
      IF(ENEW .LT. 0.0)GO TO 200
      TST = 1.0
      IF(EOLD .EQ. 0.0)GO TO 320
      TST = (EOLD - ENEW) / ENEW
      IF(ENEW .GT. EOLD)TST = (ENEW - EOLD) / EOLD
  320 EOLD = ENEW
      IF(TST .GT. .001)GO TO 200
      E = ENEW / NP
      AN = NS / 2.0
      E = E**AN
      IF(NS .EQ. 2)E = 3.142 * E
      IF(NS .EQ. 3)E = 4.189 * E
      IF(NS .EQ. 4)E = 4.935 * E
      IF(NS .EQ. 5)E = 5.264 * E
      IF(NS .EQ. 6)E = 5.168 * E
      RETURN
      END
```

Fig. 3. Program Listing of HELIPS.

and Eq. (18) for the derivatives becomes

$$\partial J_{jk}/\partial B_{rs} = \sum_i \left\{ \left(2 - \delta_{rs}\right) C_{11,rs}\, X_i^j\, X_i^k - \left[C_{jkrs} + \left(1 - \delta_{rs}\right) C_{jksr}\right] \left(X_i^1\right)^2 \right\} \quad (27)$$

Equations (20) through (24) still apply. Figure 3 contains the listing of the computer program that utilizes these formula. This program is significantly faster than that of Figure 1. In two dimensions, the equations in this section reduce to solving the rms ellipse equations.[1]

II. RESULTS AND CONCLUSIONS

The appendix describes some tests of these computer routines. The routines in all cases perform as expected.

III. REFERENCES

1. "General Least-Squares Fitting Procedures to Minimize the Volume of a Hyperellipsoid," E. A. Wadlinger, 1979 Linear Accelerator Conf. Proc., Brookhaven.

APPENDIX

We made many tests to check the performance of the two different hyperellipse fitting routines. With one set of tests, we generated phase-space distributions with known shapes and then used the computer routines to fit hyperellipsoids to these distributions. The fitted hyperellipsoid shapes agreed with that expected within statistics. Figure A.1 lists the results for one particular case; Fig. A.2 lists the computer test routine. Various possible tests of the hyperellipse fitting routines are indicated in the listing.

The phase-space points were generated randomly in an n dimensional, rectangular volume with faces perpendicular to the axes of the hyperspace. The phase-space points were then rotated by an orthogonal matrix $\underline{U}$ to give the rectangular volume an arbitrary orientation in space. The orthogonal matrix was generated by using a random number generator to produce arbitrary vectors that were then orthogonalized by the Schmidt procedure.

We generated 2000 phase-space points for the run shown in Fig. A.1. We required that nm/2 = 2. We show in the figure the orthogonal matrix U that was used to generate the expected ellipse. The terms in the fitted ellipse shown in the figure should be compared to those of the expected ellipse. We multiplied the fitted ellipse by U^T (the transpose of the orthogonal matrix U) to obtain the transferred fitted ellipse. The matrix is diagonal for a perfect fit. The diagonal elements of the matrix can be compared with the list of expected diagonal terms. The off-diagonal elements are reasonably small compared to the diagonal terms. An estimate of the expected statistical accuracy can be inferred from the coordinates for the centroid of the distribution, which were expected to be zero.

ORTHOGONAL MATRIX (U)

.395E+00	-.431E-01	.675E+00	.558E+00	-.895E-01	.258E+00
.648E+00	-.228E+00	-.855E-01	-.227E-01	.158E+00	-.704E+00
.536E+00	.281E+00	-.585E+00	.113E+00	-.356E+00	.390E+00
.203E+00	.238E+00	.430E+00	-.708E+00	-.464E+00	-.239E-01
.309E+00	-.123E+00	.473E-01	-.399E+00	.700E+00	.488E+00
.427E-02	.892E+00	.872E-01	.119E+00	.368E+00	-.217E+00

EXPECTED ELLIPSE (U * D)

.835E+00	.552E-01	-.121E+00	-.272E+00	.104E+00	.183E+00
.552E-01	.181E+01	.121E+00	.976E-01	.244E+00	-.282E+00
-.121E+00	.121E+00	.766E+00	-.356E+00	.325E-01	-.143E+00
-.272E+00	.976E-01	-.356E+00	.108E+01	.201E-02	-.252E+00
.104E+00	.244E+00	.325E-01	.201E-02	.152E+01	.205E+00
.183E+00	-.282E+00	-.143E+00	-.252E+00	.205E+00	.999E+00

FITTED ELLIPSE (EF)

.896E+00	.568E-01	-.128E+00	-.236E+00	.119E+00	.168E+00
.568E-01	.180E+01	.105E+00	.998E-01	.253E+00	-.304E+00
-.128E+00	.105E+00	.762E+00	-.374E+00	.427E-01	-.123E+00
-.236E+00	.998E-01	-.374E+00	.106E+01	-.424E-02	-.265E+00
.119E+00	.253E+00	.427E-01	-.424E-02	.148E+01	.183E+00
.168E+00	-.304E+00	-.123E+00	-.265E+00	.183E+00	.988E+00

EXPECTED DIAGONAL TERMS (D)

.3340E+00	.6680E+00	.1002E+01	.1336E+01	.1670E+01	.2004E+01

TRANSFORMED FITTED ELLIPSE (U-TRANSPOSE * EF)

.335E+00	.274E-01	.165E-01	.304E-02	.221E-01	-.127E-01
.274E-01	.701E+00	.394E-01	-.200E-01	.899E-02	.284E-01
.165E-01	.394E-01	.101E+01	-.197E-01	-.345E-02	.866E-02
.304E-02	-.200E-01	-.197E-01	.130E+01	.224E-01	-.125E-02
.221E-01	.899E-02	-.345E-02	.224E-01	.164E+01	-.115E-01
-.127E-01	.284E-01	.866E-02	-.125E-02	-.115E-01	.200E+01

CENTROID OF THE DISTRIBUTION

.3709E-02	-.3175E-02	.1874E-01	.1164E-01	-.4445E-02	-.1958E-01

Fig. A-1. Results for fitting a hyperellipse to a set of phase space points.

```
      PROGRAM TESTE(INPUT,OUTPUT,DATA,TAPE6=DATA,TAPE5)
C      WRITTEN BY E.A.WADLINGER, 1979.
C
C      TEST ELLIPSE FITTING ROUTINES
      DIMENSION Y(6),B(6,6),XO(6),D(6),U(6,6),UU(6,6)
      DIMENSION INDX(6,6)
      DIMENSION X(6,2000)
C
      WRITE(5,999)
C      READ NO OF POINTS, DIMENSION OF THE PHASE SPACE,
C      POWER OF THE VOLUME,
C      ELLIPSE ROTATION METHOD, CUT FOR METHOD
C      ITYPE = 1; X = U*X
C              2; X = UT*X, UT = U TRANSPOSE
C              3; X = U*X OR UT*X FOR EVEN OR ODD PARTICLE INDEX
C              4; X = U*X FOR X .GT. CUT
C                 X = UT*X FOR X .LT. CUT
C              5; REVERSE OF 4
C              6...10 = 1...5 WITH U BLOCK DIAGONAL
C      ITYPE = 2,3,5,7,8,10 SHOULD GIVE BAD RESULTS
C      ITYPE = 4,5,9,10 SHOW THE EFFECTS OF DIFFERENT WEIGHTS
      READ(6,*)NP,NS,AEXP,ITYPE,CUT
      WRITE(5,1095)NP,NS,AEXP,ITYPE,CUT
C      READ EIGENVALUSE FOR UPRIGHT ELLIPSE
      READ(6,*)D
C      MAKE SURE D HAS UNIT DETERMINANT
      SUM = 1.0
      DO 5 I = 1,NS
    5 SUM = SUM * D(I)
      EXP = 1.0 / NS
      SUM = 1.0 / SUM**EXP
      DO 6 I = 1,NS
    6 D(I) = D(I)*SUM
C
      IF(ITYPE .LE. 5)GO TO 15
C      DETERMINE AN ORTHOGONAL BLOCK DIAGONAL MATRIX U.
C      INDX CREATES THE DESIRED BLOCK DIAGONAL FORM.
      READ(6,*)INDX
      WRITE(5,1097)
      WRITE(5,1005)INDX
      DO 35 I = 1,NS
      DO 35 J = 1,NS
      U(I,J) = RANF(DUMMY)*INDX(I,J)
   35 UU(I,J) = U(I,J)
      ITYPE = ITYPE - 5
      GO TO 45
C      DETERMINE AN ORTHOGONAL MATRIX U.
   15 DO 10 I = 1,NS
      DO 10 J = 1,NS
      U(I,J) = RANF(DUMMY)
   10 UU(I,J) = U(I,J)
C      ORTHOGONALIZE USING THE SCHMIDT PROCEDURE.
   45 NS1 = NS - 1
      DO 160 I = 1,NS1
      SUM = 0.0
      DO 100 J = 1,NS
  100 SUM = SUM + U(I,J)**2
      SUM = 1.0 / SQRT(SUM)
      DO 120 J = 1,NS
  120 U(I,J) = U(I,J)*SUM
      K = I + 1
      DO 150 L = 1,I
      SUM = 0.0
      DO 130 J = 1,NS
  130 SUM = SUM + U(L,J)*U(K,J)
      DO 140 J = 1,NS
  140 U(K,J) = U(K,J) - SUM * U(L,J)
  150 CONTINUE
  160 CONTINUE
      SUM = 0.0
      DO 170 J = 1,NS
      SUM = SUM + U(NS,J)**2
  170 CONTINUE
      SUM = 1.0 / SQRT(SUM)
      DO 180 J = 1,NS
  180 U(NS,J) = U(NS,J) * SUM
C
C      PRINT THE ORTHOGONAL MATRIX U
      WRITE(5,1010)
      WRITE(5,1000)U
C
C      NOW GENERATE THE PARTICLES. X(CORD. NO., PARTICLE NO.).
      NU = 0
      NUT = 0
      IF(ITYPE .GT. 5)ITYPE = 5
      IF(ITYPE .LT. 1)ITYPE = 1
      DO 200 I = 1,2000
  205 SUM = 0.0
      DO 210 J = 1,NS
      Y(J) = (1.0 - 2.0*RANF(DUMMY))
      Y(J) = Y(J) / SQRT(D(J))
  210 SUM = SUM + Y(J) * Y(J)
      GO TO(240,250,270,280,280),ITYPE
C
C      EVEN PARTICLES TRANSFERED BY U, ODD BY U TRANSPOSE
  270 J = I/2
      J = I - 2*J
      IF(J .EQ. 0)GO TO 240
      GO TO 250
C
C      VOLUMES GREATER THAN CUT GET ROTATED BY
C      (U FOR ITYPE=4), (U TRANSPOSE FOR ITYPE=5),
C      ROTATION IN OPPOSITE DIRECTION FOR VOLUMES LESS THAN CUT
  280 SUM = 0.0
      DO 282 J = 1,NS
  282 SUM = SUM + D(J)*Y(J)**2
      IF(SUM .GT. CUT)GO TO 286
      IF(ITYPE .EQ. 4)GO TO 250
      GO TO 240
C      EXTEND THE OUTER PARTICLES TO EMPHASIZE THE DESIRED EFFECT.
  286 DO 287 J = 1,NS
  287 Y(J) = 2.0 * Y(J)
      IF(ITYPE .EQ. 4)GO TO 240
      GO TO 250
C
C      TRANSFORM THE VECTOR Y BY U.
  240 DO 220 J = 1,NS
      SUM = 0.0
      DO 230 K = 1,NS
  230 SUM = SUM + U(J,K)*Y(K)
  220 X(J,I) = SUM
      NU = NU + 1
      GO TO 200
C
C      TRANSFORM THE VECTOR Y IN THE OPPOSITE DIRECTION
C      WITH U TRANSPOSE.
  250 DO 260 J = 1,NS
      SUM = 0.0
      DO 255 K = 1,NS
  255 SUM = SUM + U(K,J)*Y(K)
  260 X(J,I) = SUM
      NUT = NUT + 1
  200 CONTINUE
      WRITE(5,1096)NU,NUT
C
C      NOW FIT THE ELLIPSE
      CALL HELIPS(2000,NS,AEXP,X,B,E,XO)
C      CALCULATE THE EXPECTED ELLIPSE
  340 DO 345 I = 1,NS
      DO 345 L = 1,NS
      SUM = 0.0
      DO 346 J = 1,NS
  346 SUM = SUM + U(I,J)*D(J)*U(L,J)
  345 UU(I,L) = SUM
C      WRITE THE EXPECTED ELLIPSE.
      WRITE(5,1035)
      WRITE(5,1000)UU
C      WRITE THE FITTED ELLIPSE.
      WRITE(5,1040)
      WRITE(5,1000)B
C      TRANSFORM B BY U TO SEE IF THE EIGENVALUES
C      D(I) APPEAR
C      D = U(J,I) B(J,K) U(K,L)
      DO 300 I = 1,NS
      DO 300 L = 1,NS
      SUM = 0.0
      DO 310 J = 1,NS
      DO 310 K = 1,NS
      SUM = SUM + U(J,I)*B(J,K)*U(K,L)
  310 CONTINUE
      UU(I,L) = SUM
  300 CONTINUE
C
C      WRITE THE DIAGONAL TERMS
      WRITE(5,1060)D
C      WRITE THE TRANSFORMED B MATRIX UU
C      FOR ITYPE = 1 AND 6 OR WITH A HEAVY WEIGHT ON THE OUTER
C      PARTICLES FOR ITYPE = 4 AND 9, THE DIAGONAL ELEMENTS OF
C      THE TRANSFORMED MATRIX SHOULD BE CLOSE THE THE VALUES OF D.
C      THE OFF DIAGONAL ELEMENTS OF THIS MATRIX SHOULD BE SMALL
C      COMPARED TO THE DIAGONAL TERMS.
      WRITE(5,1050)
      WRITE(5,1000)UU
C      WRITE THE CENTROID OF THE DISTRIBUTION
C      THE DEVIATION FROM ZERO INDICATES THE STATISTICAL
C      ACCURACY EXPECTED FOR THE FIT.
      WRITE(5,1070)XO
      CALL EXIT
C
  999 FORMAT(1H1)
 1000 FORMAT(1H ,6E11.3)
 1005 FORMAT(1H ,6I6)
 1010 FORMAT(/  ORTHOGONAL MATRIX U )
 1035 FORMAT(/  EXPECTED ELLIPSE )
 1040 FORMAT(/  FITTED ELLIPSE )
 1050 FORMAT(/  TRANSFORMED MATRIX )
 1060 FORMAT(/  DIAGONAL TERMS = /1H ,6E11.4)
 1070 FORMAT(/  CENTROID OF THE DISTRIBUTION /
     1(1H ,6E11.4)
 1090 FORMAT(/  SUBROUTINE HELIPS RESULTS )
 1095 FORMAT(  NP,NS,AEXP,ITYPE,CUT /1H ,2I6,E12.3,I6,E12.3)
 1096 FORMAT(/  NO. PARTICLES TRANSFORMED BY U = ,I6/
     1  NO. PARTICLES TRANSFORMED BY UT = ,I5)
 1097 FORMAT(/  CORRELATION CONTROL MATRIX )
      END
```

Fig. A-2. Program for testing HELIPS.