PROGRAMAS DEL ALGORITMO DE CLASIFICACION DE POBLACIONES NORMALES TRIVARIANTES A PARTIR DE LA ENTROPIA DE MEZCLA [1]

Santiago Alcobé
BULL ESPAÑA, S.A., Barcelona, Spain

and

Rafael Cubarsi
Dept. Matemàtica Aplicada IV
Universitat Politècnica de Catalunya
Jordi Girona 1-3, E08034 Barcelona, Spain

[1] Programas asociados al modelo desarrollado en la Tesis Doctoral *Contribución al Estudio de la Dinàmica Galàctica: Superposición de Sistemas Estelares.*

Abstract

The complete FORTRAN codified programs of the segregation algorithm described in Alcobé (2001)[2] are described. The classification algorithm is applied to study the trivariate velocity distribution of stars from star catalogs, which can be locally approximated by a superposition of two or more normal components. An auxiliar sampling parameter P (such as the velocity module referred to a specific point, the absolute value of one peculiar velocity component alone, the distance to the galactic plane, etc.) is introduced in order to define the sample boundaries. The sampling parameter must induce a hyerarchical incorporation of stars to the population components, in the sense that the greater the P value, the greater the number of stars in each component. For a fixed P, a sample S(P) is drawn from the global catalog. Depending on the sampling parameter the population entropy H(P) of a two-component mixture is computed from the mixing proportions. The purpose is to find the optimal P value in order to maximize H(P). Then the algorithm is used recursively in order to segregate a global sample in more than two populations. Moreover, for each subsample S(P), the goodness of the superposition approximation is evaluated by reconstructing the sample central moments up to fourth-order from the population parameters. A chi-square test, taking into account the sampling distribution moments, is evaluated to measure the fitting error. For subsamples S(P) a total accordance between the minimum chi-square and the maximum population entropy H(P) is produced.

[2] *Contribución al Estudio de la Dinàmica Galàctica: Superposición de Sistemas Estelares*, Ph. D. Thesis, Universitat de Barcelona, Barcelona, 2001.

CONTENIDO

- Programas asociados al modelo desarrollado en la Tesis Doctoral *Contribución al Estudio de la Dinàmica Galàctica: Superposición de Sistemas Estelares.*
- Listado de programas por orden alfabético.

Barcelona, Enero de 2001.

Introducción

Este documento constituye un anexo a la memoria de tesis doctoral CONTRIBUCIÓN AL ESTUDIO DE LA DINÁMICA GALÁCTICA: SUPERPOSICIÓN DE SISTEMAS ESTELARES presentada por Santiago Alcobé en la Universidad de Barcelona. Comprende los programas asociados al modelo estadístico-numérico desarrollado en dicho trabajo.

Se describen los programas principales y rutinas utilizadas en el desarrollo del modelo de separación de poblaciones así como otros auxiliares. Estos programas han sido codificados en lenguaje FORTRAN.

El conjunto de programas y rutinas está constituido por un total de 28 archivos. Entre ellos hay 2 programas principales que consisten en las diferentes formas de atacar el problema a poblaciones ideales y muestras sintéticas y reales. Además hay 7 programas auxiliares que realizan tareas intermedias como generación de muestras sintéticas o selección de estrellas. Los 19 restantes constituyen las diferentes rutinas que son llamadas por los programas principales o auxiliares. Así mismo, se utilizan hasta 40 ficheros de datos. Procedamos a la descripción de cada uno de los elementos de cálculo.

Todas las referencias bibliográficas así como las que se hacen a expresiones matemáticas aparecen descritas en la mencionada memoria.

1.1 Programa principal de tratamiento de muestras sintéticas y reales

En primer lugar describimos el programa principal y rutinas que materializan el modelo numérico de aproximación por superposición de gausianas. Se utiliza para muestras reales y sintéticas.

1.1.1 SSE10

Los pasos que realiza son:

- Lectura de los valores de los momentos de la muestra: SINMO2.
- Lectura de los valores de los errores correspondientes a esos momentos: SINRO2.
- Cálculo de los cumulantes: CALNU.
- Cálculo de las d_i: CALDES.
- Cálculo de las variables auxiliares P,S,X,Y,Z,T: PSXYZT.
- Cálculo de la matriz de pesos a partir de los errores de las variables anteriores y los errores de las d_i. Se evalúan también los errores de las incógnitas: PESOER

- Aplicación de los pesos al sistema de ecuaciones: EAEB.
- Resolución del sistema de mínimos cuadrados: MINCUA.
- Cálculo del vector **D** y sus errores: CALDDE.
- Definición de los signos de los elementos del tensor $\mathbf{C}_2$ y sus errores según 3-13: CALCCE.
- Cálculo del parámetro q y errores: CALQUE.
- Definición de los elementos del tensor $\mathbf{C}_2$ y sus errores según 3-14: CESDES.
- Elección de la mejor de las soluciones anteriores
- Cálculo de los momentos y velocidades de las componentes parciales y errores: MOMPAE.
- Apertura y escritura del fichero RESSIN.TXT con los resultados.
- Cálculo de la cantidad χ^2 para estimar la bondad de la aproximación: RECALE

1.2 Programa principal de tratamiento de poblaciones ideales

1.2.1 SSE3

Mezcla y posterior separación de **poblaciones ideales**. Puesto que las poblaciones ideales se definen sin errores no tiene sentido en este caso hablar de cálculo de errores ni ponderación de ecuaciones. Los pasos que realiza son:

- Lectura de los valores de los momentos de las n poblaciones ideales que se mezclan: LECTU.
- Cálculo de los momentos de la muestra global a partir de las parciales leídas: CALDEL + CALMOM + CALNU.
- Cálculo de las d_i: CALDES.
- Cálculo de las variables auxiliares P,S,X,Y,Z,T: PSXYZT.
- Resolución del sistema de mínimos cuadrados: MINCUA.
- Cálculo de **D**: CALDD.
- Definición de los signos de los elementos y cálculo del tensor $\mathbf{C}_2$: CALCC.
- Cálculo del parámetro q: CALQU.
- Cálculo de los elementos del tensor $\mathbf{C}_2$: CESDES.
- Elección de la mejor de las soluciones anteriores.
- Cálculo de los momentos y velocidades de las componentes parciales: MOMPAR.
- Apertura y escritura del fichero RESU.TXT con los resultados.

1.3 Rutinas llamadas por los programas principales de separación

CALCC

Cálculo de los elementos del tensor $\mathbf{C}_2$. Permite Conocer el signo de cada elemento. Además, si no se reasignan valores a estos elementos, los valores calculados aquí son los que permanecen. Se utilizan las expresiones 3-13.

CALCCE

Igual que la anterior pero se evalúan también los errores de las variables que intervienen.

CALDD

A partir del vector obtenido en la resolución del sistema de mínimos cuadrados, se calculan tanto C_{22} como las tres componentes del vector **D**. La primera incógnita del sistema proporciona el valor de $\frac{1}{D_2^2}$ y la segunda $\frac{C_{22}}{D_2}$.

CALDDE

Igual que la anterior pero se evalúan también los errores de las variables que intervienen.

CALDEL

Calcula el valor de los vectores $\mathbf{w}^{(i)}$ según su definición 2-10 y el valor de la velocidad del centroide de la muestra total según 2-7. Se utiliza para generar muestras ideales a partir de poblaciones ídem.

CALDES

De acuerdo con las ecuaciones 3-4:

- Resuelve las ecuaciones cúbicas que proporcionan los valores de las variables auxiliares d_i (componentes del vector normalizado **d**) a partir de las dos primeras expresiones de 3-4.
- Utiliza las dos siguientes para tener dos nuevas soluciones.
- Se calculan las tangentes a las cuatro funciones en el punto hallado, con lo que se puede construir un sistema de mínimos cuadrados. De esta forma se promedian las dos soluciones obtenidas para cada d_i.

- La rutina permite escoger entre diferentes valores de las d_i para ajustarse a la mínima χ^2.
- Se ha establecido que, si al resolver las ecuaciones cúbicas, una de las d_i es cero la rutina se detiene ahí y considera que esas d_i son las válidas.

CALMOM

A partir de los momentos de las poblaciones parciales calcula los momentos de la muestra global utilizando las expresiones 2-19, 2-21 y 2-24.

Previamente se obtienen los momentos de orden 4 parciales de cada población ideal a partir de los de orden dos de las mismas según:

$$\mu_{ijkl} = \mu_{ij}\mu_{kl} + \mu_{ik}\mu_{jl} + \mu_{il}\mu_{jk}; i,j,k \in \{1,2,3\}$$

CALNU

Calcula los cumulantes de orden cuatro según $\mathbf{N}_4 = \mathbf{M}_4 - 3\mathbf{M}_2 * \mathbf{M}_2$

CALQU

A partir de los momentos de tercer orden, los cumulantes, C_{22} y D_2 se obtiene la variable q la cual está directamente relacionada con la fracción de cada población. de acuerdo con su definición 2-40, puesto que $n' = 1-n''$, esto se traduce en:

$$n' = \frac{1}{2}\left(1 + \sqrt{\left(1 - \frac{4}{(q^2+4)}\right)}\right)$$

CALQUE

Igual que la anterior pero se evalúan también los errores de las variables que intervienen.

CESDES

Se calcula C_{ij} utilizando las relaciones 3-14.

Las cuales se traducen en:

$$C_{ij} = \pm\sqrt{\frac{1}{3}\left(\nu_{iijj} - D_i^2 D_j^2\left(\frac{\nu_{2222} - 3C_{22}^2}{D_{22}^4}\right)\right)}$$

El signo de la raíz cuadrada viene dado por la rutina CALCC. Si el radicando es negativo, se conserva el valor originalmente calculado por CALCC para dicho elemento del tensor. La diferencia conceptual entre esta rutina y CALCC es que la presente calcula los elementos del tensor C según los momentos de cuarto orden y la otra según los de tercero.

EAEB

Realiza el producto de la matriz con los coeficientes de las ecuaciones por la de pesos. Así mismo, multiplica el vector de los términos independientes por la matriz de pesos. En definitiva, aplica los pesos a las ecuaciones.

EDSG

Se le dan los tres coeficientes de una ecuación de segundo grado y devuelve las raíces en un vector y un escalar indicando el número de soluciones. Es llamada desde la rutina CALDES.

EDTG

Resuelve una ecuación de tercer grado utilizando el algoritmo correspondiente (p.e. Bronshtein & Semendiaev 1982). Los casos triviales se resuelven aparte. Devuelve las raíces en un vector y un escalar indicando el número de soluciones. Es llamada desde la rutina CALDES.

LECTU

Rutina de lectura de momentos parciales de poblaciones ideales. Se aplica únicamente en el caso de poblaciones ideales. Permite definir con cuantas poblaciones se va a trabajar.

MINCUA

Resuelve un sistema de mínimos cuadrados que tenga 2 incógnitas y un máximo de 14 ecuaciones. Utiliza el clásico algoritmo: $\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{x} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{WA}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{Wb}$ donde **W** es la matriz de pesos.

MOMPAE

Tras haber calculado q y las fracciones de población de cada componente en la rutina CALQUE se calculan los momentos de orden dos de las poblaciones parciales y la propagación cuadrática de sus errores de cálculo. Se utiliza:

$$\mathbf{M}_2 = \mathbf{a}_2 + (\mathbf{D})^2 \Rightarrow \mathbf{a}_2 = \mathbf{M}_2 - (\mathbf{D})^2$$

$$\mathbf{b}_2 = \frac{1}{\sqrt{n'n''}}\left(\mathbf{C}_2 - q(\mathbf{D})^2\right)$$

$$\mathbf{M}_2' = \mathbf{a}_2 - n''\mathbf{b}_2; \mathbf{M}_2'' = \mathbf{a}_2 - n'\mathbf{b}_2$$

$$\mathbf{v}' = \mathbf{v} + n''\mathbf{w}; \mathbf{v}'' = \mathbf{v} - n'\mathbf{w}$$

MOMPAR

Tras haber calculado q y las fracciones de población de cada componente en la rutina CALQU se realizan las mismas operaciones que en la rutina anterior MOMPAE pero sin calcular la propagación de errores.

PESOER

De acuerdo con 3-18, calcula los errores de P, S, X, Y, Z, T y de las ecuaciones que entran en el sistema de mínimos cuadrados 3-11. La matriz de pesos es la inversa de la matriz de covarianza de los errores del sistema de ecuaciones. Las variables d se consideran sin error. Se utilizan errores absolutos.

PSXYZT

Calcula las variables auxiliares P, S, X, Y, Z, T definidas en 3-5 y 3-6. Estas variables entran en los cocientes que constituyen las ecuaciones el sistema de mínimos cuadrados.

RECALE

Cálculo de los residuos de los momentos de una población suponiendo que la muestra global está compuesta por dos poblaciones gausianas. Recompone los momentos globales según 2-30, 2-33 y 2-36. A partir de ahí evalúa la cantidad χ^2 según 3-21.

SINMO2

Lectura de los momentos globales de una muestra real o sintética.

SINRO2

Lectura de los errores correspondientes a los momentos globales leídos en la rutina SINMO2. Incluye un factor de eliminación de momentos. Si el error leído es mayor que el momento según un factor definido, el momento se considera nulo y el error igual al momento. Informa de cada momento que es anulado. Calcula también los errores de los cumulantes. Realiza este cálculo utilizando suma cuadrática de errores.

1.4 Programas adicionales de creación de muestras sintéticas y otros

GENMUE

Generación de 3n números aleatorios con distribución N(0,1). Utiliza las rutinas estándar (Press et al. 1992) *gasdev* de generación de números aleatorios y *ran1*, fuente de desviaciones uniformes que devuelve desviación normalmente distribuida con media cero y varianza unidad.

SSEGIR

Programa que aplica los correspondientes cambios de escala a las N(0,1) generadas con el programa anterior para obtener una distribución con media, dispersión y orientación determinada. Utiliza la rutina GIRMUE.

SSEMOM

Cálculo de los momentos y cumulantes de una muestra real o sintética mediante algoritmos estándar a partir de los valores de las velocidades de las estrellas de la muestra. Utiliza la rutina CALCMO (la cual a su vez llama a la rutina CALCUM). La versión M14000 de este programa es igual pero dimensionada a 14.000 estrellas.

MEZMUE

Mezcla de n muestras sintéticas en las proporciones que se indiquen para obtener una superposición de poblaciones sintéticas.

GLIEJAH

Selección de estrellas del catálogo según el valor del módulo de la velocidad. La versión G14000 de este programa es igual pero dimensionada a 14.000 estrellas.

DECIDE

Selección de estrellas de una población determinada en función de su probabilidad de pertenencia a la misma.

LESCRIBE

Programa de cambio de formato de lectura-escritura.

```
C234567
C
C *** CALCULO DEL TENSOR C PARA CONOCER EL SIGNO DE CADA ELEMENTO
C
      SUBROUTINE CALCC(D,DD,P,S,C)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION P(3,3),C(3,3),DD(3),S(3),D(3)
C ***
      DO 100 I=1,2
      C(I,3)=1.D0/D(3)*(S(I)/DD(3)+D(I)*C(3,3))
 100  CONTINUE
C
      DO 200 I=1,2
      DO 210 J=I,2
      C(I,J)=1.D0/D(3)**2*(P(I,J)/DD(3)+D(J)*D(3)*C(I,3)+D(I)*D(3)*C(J,3
     #)-D(I)*D(J)*C(3,3))
 210  CONTINUE
 200  CONTINUE
C
      DO 300 I=1,3
      DO 310 J=I,3
      C(J,I)=C(I,J)
 310  CONTINUE
 300  CONTINUE
C
      DO 400 I=1,3
      DO 410 J=1,3
      WRITE(*,*)'TENSOR C2',I,J,C(I,J)
 410  CONTINUE
 400  CONTINUE
C
      RETURN
      END
```

```
C234567
C
C *** CALCULO DEL TENSOR C PARA CONOCER EL SIGNO DE CADA ELEMENTO
C *** SE CALCULAN TAMBIEN LOS ERRORES
C *** IGUAL QUE CALCC PERO CON CALCULO DE ERRORES
C
      SUBROUTINE CALCCE(D,DD,P,S,C,ES,EP,EDD,EC33,EC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION P(3,3),C(3,3),DD(3),S(3),D(3),EDD(3),EC(3,3)
      DIMENSION ES(3),EP(3,3)
C ***
      WRITE(*,*)'ACCESO A RUTINA CALCCE'
      EC(3,3)=EC33
      DO 100 I=1,2
      C(I,3)=1.D0/D(3)*(S(I)/DD(3)+D(I)*C(3,3))
 100  CONTINUE
C
      DO 200 I=1,2
      DO 210 J=I,2
      C(I,J)=1.D0/D(3)**2*(P(I,J)/DD(3)+D(J)*D(3)*C(I,3)+D(I)*D(3)*C(J,3
     #)-D(I)*D(J)*C(3,3))
 210  CONTINUE
 200  CONTINUE
C
      DO 300 I=1,3
      DO 310 J=I,3
      C(J,I)=C(I,J)
 310  CONTINUE
 300  CONTINUE
C
C *** SE CALCULAN LOS ERRORES DE LAS COMPONENTES DEL TENSOR C
C
      DO 500 I=1,2
      EC(I,3)=1.D0/D(3)*DSQRT((ES(I)/DD(3))**2+(S(I)*EDD(3)/DD(3)**2)
     #**2+(D(I)*EC(3,3))**2)
 500  CONTINUE
C
      DO 510 I=1,2
      DO 520 J=I,2
      EC(I,J)=1.D0/D(3)**2*DSQRT((EP(I,J)/DD(3))**2+(P(I,J)*EDD(3)/DD(3)
     #**2)**2+(D(J)*D(3)*EC(I,3))**2+(D(I)*D(3)*EC(J,3))**2+(D(I)*D(J)*
     #EC(3,3))**2)
 520  CONTINUE
 510  CONTINUE
C
      DO 600 I=1,3
      DO 610 J=I,3
      EC(J,I)=EC(I,J)
 610  CONTINUE
 600  CONTINUE
C
      DO 400 I=1,3
      DO 410 J=1,3
      WRITE(*,*)'C2',I,J,C(I,J),EC(I,J)
 410  CONTINUE
 400  CONTINUE
```

```
C
      RETURN
      END
```

```
C234567
C
C *** CALCULO DE LOS MOMENTOS DE SEGUNDO TERCER Y CUARTO ORDEN DE
C *** UNA MUESTRA DE ESTRELLAS
C *** CALCULA LOS MOMENTOS CENTRADOS Y NO CENTRADOS DE LAS VELOCIDADES
C *** RESIDUALES OBSERVADAS
C
      SUBROUTINE CALCMO(FICGIR,FICMOM)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*64 FICGIR,FICMOM
      DIMENSION U(10000),V(10000),W(10000),UMX(2),VMX(2),WMX(2),
     #C(9,9,9),VC(5,5,5),S(9,9,9),VS(5,5,5)
      WRITE(*,*)'ACCESO A LA RUTINA CALCMO'
C *** LECTURA DEL FICHERO DE DATOS
      OPEN(1,FILE=FICGIR,STATUS='OLD')
      OPEN(2,FILE=FICMOM,STATUS='NEW')
      NAA=0
      DO 100 I=1,10000
      READ(1,500,END=7777)U(I),V(I),W(I)
      NAA=NAA+1
 100  CONTINUE
 7777 CONTINUE
 500  FORMAT(F17.10,1X,F17.10,1X,F17.10)
      A=FLOAT(NAA)
      WRITE(*,*)'TOTAL ESTRELLAS',NAA
C *** CALCULO DE LAS MEDIAS DE LAS VELOCIDADES
      UMX(1)=0.D0
      VMX(1)=0.D0
      WMX(1)=0.D0
      KY=1
      DO 1030 II=1,NAA
C *** SE CONSIDERAN LAS VELOCIDADES U=UDD, V=VDD, W=WDD (DESCONTADA LA
C *** ROTACION GALACTICA)
      UMX(KY)=UMX(KY)+U(II)
      VMX(KY)=VMX(KY)+V(II)
      WMX(KY)=WMX(KY)+W(II)
 1030 CONTINUE
      UMX(KY)=UMX(KY)/A
      VMX(KY)=VMX(KY)/A
      WMX(KY)=WMX(KY)/A
C ***
      DO 1002 I=1,9
      DO 1003 J=1,9
      DO 1004 KA=1,9
      JJJJ=I+J+KA
      IF(JJJJ.GT.11) GO TO 1004
      S(I,J,KA)=0.D0
      C(I,J,KA)=0.D0
      DO 1005 II=1,NAA
      C(I,J,KA)=C(I,J,KA)+(((U(II)-UMX(KY))**(I-1))*((V(II)
     #-VMX(KY))**(J-1))*((W(II)-WMX(KY))**(KA-1)))
      S(I,J,KA)=S(I,J,KA)+((U(II))**(I-1))*(V(II))**(J-1)*
     #((W(II))**(KA-1))
 1005 CONTINUE
      C(I,J,KA)=C(I,J,KA)/A
      S(I,J,KA)=S(I,J,KA)/A
```

```
 1004 CONTINUE
 1003 CONTINUE
 1002 CONTINUE
C ***
      WRITE(*,*)'SE CALCULAN LOS MOMENTOS'
      DO 1006 I=1,5
      DO 1007 J=1,5
      DO 1008 KA=1,5
      JJJJ=I+J+KA
      IF(JJJJ.GT.7) GO TO 1008
      VS(I,J,KA)=DSQRT(DABS((S(2*I-1,2*J-1,2*KA-1)-S(I,J,KA)*S(I,J,KA))
     #/A))
 1008 CONTINUE
 1007 CONTINUE
 1006 CONTINUE
C ***
      WRITE(*,*)'SE CALCULAN LOS ERRORES DE LOS MOMENTOS'
      V1=DSQRT(DABS((C(3,1,1))/A))
      V2=DSQRT(DABS((C(1,3,1))/A))
      V3=DSQRT(DABS((C(1,1,3))/A))
C
      V11=DSQRT(DABS((C(5,1,1)-C(3,1,1)*C(3,1,1))/A))
      V12=DSQRT(DABS((C(3,3,1)-C(2,2,1)*C(2,2,1))/A))
      V13=DSQRT(DABS((C(3,1,3)-C(2,1,2)*C(2,1,2))/A))
      V22=DSQRT(DABS((C(1,5,1)-C(1,3,1)*C(1,3,1))/A))
      V23=DSQRT(DABS((C(1,3,3)-C(1,2,2)*C(1,2,2))/A))
      V33=DSQRT(DABS((C(1,1,5)-C(1,1,3)*C(1,1,3))/A))
C
      V111=DSQRT(DABS((C(7,1,1)-6*(C(5,1,1)*C(3,1,1))-C(4,1,1)*
     #C(4,1,1)+9*(C(3,1,1)**3))/A))
      V112=DSQRT(DABS((C(5,3,1)-4*(C(4,2,1)*C(2,2,1))-2*(C(3,3,1)*C(3
     #,1,1))-C(3,2,1)*C(3,2,1)+2*(C(3,1,1)*C(2,2,1)*C(2,2,1))+C(1,3,
     #1)*C(3,1,1)*C(3,1,1)+6*(C(3,1,1)*C(2,2,1)*C(2,2,1)))/A))
      V113=DSQRT(DABS((C(5,1,3)-4*(C(4,1,2)*C(2,1,2))-2*(C(3,1,3)*C(3
     #,1,1))-C(3,1,2)*C(3,1,2)+2*(C(3,1,1)*C(2,1,2)*C(2,1,2)+C(1,1,
     #2)*C(3,1,1)*C(3,1,1))+6*(C(3,1,1)*C(2,1,2)*C(2,1,2)))/A))
      V122=DSQRT(DABS((C(3,5,1)-2*(C(3,3,1)*C(1,3,1))-4*(C(2,4,1)*C(2
     #,2,1))-C(2,3,1)*C(2,3,1)+C(3,1,1)*C(1,3,1)*C(1,3,1)+2*(C(1,3,
     #1)*C(2,2,1)*C(2,2,1))+6*(C(2,2,1)*C(2,2,1)*C(1,3,1)))/A))
      V123=DSQRT(DABS((C(3,3,3)-2*(C(3,2,2)*C(1,2,2)+C(2,3,2)*C(2,1,2)+
     #C(2,2,3)*C(2,2,1))-C(2,2,2)*C(2,2,2)+C(3,1,1)*C(1,2,2)*C(1,2,2)
     #+C(1,3,1)*C(2,1,2)*C(2,1,2)+C(1,1,3)*C(2,2,1)*C(2,2,1)+6*
     #(C(2,2,1)*C(2,1,2)*C(1,2,2)))/A))
      V133=DSQRT(DABS((C(3,1,5)-2*C(3,1,3)*C(1,1,3)-4*C(2,1,4)*C(2
     #,1,2)-C(2,1,3)*C(2,1,3)+C(3,1,1)*C(1,1,3)*C(1,1,3)+2*C(1,1,
     #3)*C(2,1,2)*C(2,1,2)+6*C(2,1,2)*C(2,1,2)*C(1,1,3))/A))
      V222=DSQRT(DABS((C(1,7,1)-6*C(1,5,1)*C(1,3,1)-C(1,4,1)*
     #C(1,4,1)+9*(C(1,3,1)**3))/A))
      V223=DSQRT(DABS((C(1,5,3)-4*C(1,4,2)*C(1,2,2)-2*C(1,3,3)*C(1
     #,3,1)-C(1,3,2)*C(1,3,2)+8*C(1,3,1)*C(1,2,2)*C(1,2,2)+C(1,1,
     #4)*C(1,3,1)*C(1,3,1))/A))
      V233=DSQRT(DABS((C(1,3,5)-2*C(1,3,3)*C(1,1,3)-4*C(1,2,4)*C(1
     #,2,2)-C(1,2,3)*C(1,2,3)+C(1,3,1)*C(1,1,3)*C(1,1,3)+8*C(1,1,
     #3)*C(1,2,2)*C(1,2,2))/A))
      V333=DSQRT(DABS((C(1,1,7)-6*C(1,1,5)*C(1,1,3)-C(1,1,4)*
     #C(1,1,4)+9*(C(1,1,3)**3))/A))
```

```
C
      V1111=DSQRT(DABS((C(9,1,1)-C(5,1,1)*C(5,1,1)-8*C(6,1,1)*C(4,1,1)+
     #16*C(3,1,1)*C(4,1,1)*C(4,1,1))/A))
      V2222=DSQRT(DABS((C(1,9,1)-C(1,5,1)*C(1,5,1)-8*C(1,6,1)*C(1,4,1)+
     #16*C(1,3,1)*C(1,4,1)*C(1,4,1))/A))
      V3333=DSQRT(DABS((C(1,1,9)-C(1,1,5)*C(1,1,5)-8*C(1,1,6)*C(1,1,4)+
     #16*C(1,1,3)*C(1,1,4)*C(1,1,4))/A))
      V1112=DSQRT(DABS((C(7,3,1)-C(4,2,1)*C(4,2,1)-6*C(5,2,1)*C(3,2,1)-
     #2*C(4,3,1)*C(4,1,1)+6*C(3,1,1)*C(3,2,1)*C(3,2,1)+C(1,3,1)*
     #C(4,1,1)*C(4,1,1)+C(3,1,1)*C(4,1,1)*C(3,2,1)+2*C(3,1,1)*C(4,1,1)
     #*C(2,3,1)+6*C(2,2,1)*C(3,2,1)*C(4,1,1))/A))
      V1113=DSQRT(DABS((C(7,1,3)-C(4,1,2)*C(4,1,2)-6*C(5,1,2)*C(3,1,2)-
     #2*C(4,1,3)*C(4,1,1)+6*C(3,1,1)*C(3,1,2)*C(3,1,2)+C(1,1,3)*
     #C(4,1,1)*C(4,1,1)+C(3,1,1)*C(4,1,1)*C(3,1,2)+2*C(3,1,1)*C(4,1,1)
     #*C(2,1,3)+6*C(2,1,2)*C(3,1,2)*C(4,1,1))/A))
      V1122=DSQRT(DABS((C(5,5,1)-C(3,3,1)*C(3,3,1)-4*C(4,3,1)*C(2,3,1)-
     #4*C(3,4,1)*C(3,2,1)+2*C(3,1,1)*C(2,3,1)*C(2,3,1)+2*C(1,3,1)*
     #C(3,2,1)*C(3,2,1)+2*C(3,1,1)*C(3,2,1)*C(1,4,1)+8*C(2,2,1)*C(2,3,1)
     #*C(3,2,1)+2*C(1,3,1)*C(2,3,1)*C(4,1,1))/A))
      XX=C(5,3,3)-C(3,2,2)*C(3,2,2)-4*C(4,2,2)*C(2,2,2)-2*(
     #C(3,3,2)*C(3,1,2)+C(3,2,3)*C(3,2,1))+2*C(3,1,1)*C(2,2,2)*
     #C(2,2,2)+C(1,3,1)*C(3,1,2)*C(3,1,2)+C(1,1,3)*C(3,2,1)*C(3,2,1)
      V1123=DSQRT(DABS((C(3,1,1)*C(3,2,1)*C(1,3,2)+C(3,1,1)*C(3,1,2)*C(
     #1,3,2)+2*(C(2,2,1)*C(2,2,2)*C(3,1,2)+C(2,2,1)*C(3,2,1)*C(2,1,3)+
     #C(2,1,2)*C(3,1,2)*C(2,3,1)+C(2,1,2)*C(2,2,2)*C(3,2,1)+C(1,2,2)*
     #C(2,2,2)*C(4,1,1))+XX)/A))
C
      V1133=DSQRT(DABS((C(5,1,5)-C(3,1,3)*C(3,1,3)-4*(C(4,1,3)*C(2,1,3)
     #+C(3,1,4)*C(3,1,2))+2*(C(3,1,1)*C(2,1,3)*C(2,1,3)+C(1,1,3)*
     #C(3,1,2)*C(3,1,2)+C(3,1,1)*C(3,1,2)*C(1,1,4))+8*C(2,1,2)*C(2,1,3)
     #*C(3,1,2)+2*C(1,1,3)*C(2,1,3)*C(4,1,1))/A))
C
      V1222=DSQRT(DABS((C(3,7,1)-C(2,4,1)*C(2,4,1)-2*C(3,4,1)*C(1,4,1)-
     #6*(C(2,5,1)*C(2,3,1)-C(1,3,1)*C(2,3,1)*C(2,3,1)-C(2,2,1)*
     #C(2,3,1)*C(1,4,1))+C(3,1,1)*C(1,4,1)*C(1,4,1)+3*C(1,3,1)*
     #C(1,4,1)*C(3,2,1))/A))
      XX=C(3,5,3)-C(2,3,2)*C(2,3,2)-2*C(3,3,2)*C(1,3,2)-4*C(
     #2,4,2)*C(2,2,2)-2*C(2,3,3)*C(2,3,1)+C(3,1,1)*C(1,3,2)*C(1,3,2)+
     #2*C(1,3,1)*C(2,2,2)*C(2,2,2)+C(1,1,3)*C(2,3,1)*C(2,3,1)+
     #C(2,2,1)*C(2,3,1)*C(1,3,2)
      V1223=DSQRT(DABS((XX+2*C(2,2,1)*C(2,2,2)*C(1,3,2)+C(2,2,1)*C(2,3,
     #1)*C(1,2,3)+2*C(2,1,2)*C(2,2,2)*C(1,4,1)+C(1,3,1)*C(2,3,1)*
     #C(2,1,2)+C(1,3,1)*C(1,3,2)*C(3,1,2)+2*C(1,2,2)*C(2,2,2)*
     #C(2,3,1)+2*C(1,2,2)*C(1,3,2)*C(3,2,1))/A))
      XX=C(3,3,5)-C(2,2,3)*C(2,2,3)-2*C(3,2,3)*C(1,2,3)-2*
     #C(2,3,3)*C(2,1,3)-4*C(2,2,4)*C(2,2,2)+C(3,1,1)*C(1,2,3)*
     #C(1,2,3)+C(1,3,1)*C(2,1,3)*C(2,1,3)+2*C(1,1,3)*C(2,2,2)*
     #C(2,2,2)+2*C(2,2,1)*C(2,2,2)*C(1,1,4)
      V1233=DSQRT(DABS((XX+2*C(2,1,2)*C(2,1,3)*C(1,3,2)+2*(C(2,1,2)*
     #C(2,2,2)*C(1,2,3)+C(1,2,2)*C(2,2,2)*C(2,1,3)+C(1,2,2)*C(1,2,3)*
     #C(3,1,2))+C(1,1,3)*C(2,1,3)*C(2,3,1)+C(1,1,3)*C(1,2,3)*
     #C(3,1,2))/A))
      V1333=DSQRT(DABS((C(3,1,7)-C(2,1,4)*C(2,1,4)-2*C(3,1,4)*C(1,1,4)-
     #6*C(2,1,5)*C(2,1,3)+C(3,1,1)*C(1,1,4)*C(1,1,4)+6*(C(1,1,3)*
     #C(2,1,3)*C(2,1,3)+C(2,1,2)*C(2,1,3)*C(1,1,4))+3*C(1,1,3)*
     #C(1,1,4)*C(3,1,2))/A))
```

```
      V2223=DSQRT(DABS((C(1,7,3)-C(1,4,2)*C(1,4,2)-6*C(1,5,2)*C(1,3,2)-
     #2*C(1,4,3)*C(1,4,1)+6*C(1,3,1)*C(1,3,2)*C(1,3,2)+C(1,1,3)*
     #C(1,4,1)*C(1,4,1)+C(1,3,1)*C(1,4,1)*C(1,3,2)+2*C(1,3,1)*
     #C(1,4,1)*C(1,2,3)+6*C(1,2,2)*C(1,3,2)*C(1,4,1))/A))
      V2233=DSQRT(DABS((C(1,5,5)-C(1,3,3)*C(1,3,3)-4*(C(1,4,3)*C(1,2,3)-
     #C(1,3,4)*C(1,3,2))+2*(C(1,3,1)*C(1,2,1)*C(1,2,3)+C(1,1,3)*
     #C(1,3,2)*C(1,3,2)+C(1,3,1)*C(1,3,2)*C(1,1,4))+8*C(1,2,2)*
     #C(1,2,3)*C(1,3,2)+2*C(1,1,3)*C(1,2,3)*C(1,4,1))/A))
      V2333=DSQRT(DABS((C(1,3,7)-C(1,2,4)*C(1,2,4)-2*C(1,3,4)*C(1,1,4)-
     #6*C(1,2,5)*C(1,2,3)+C(1,3,1)*C(1,1,4)*C(1,1,4)+6*(C(1,1,3)*
     #C(1,2,3)*C(1,2,3)+6*C(1,2,2)*C(1,2,3)*C(1,1,4))+3*C(1,1,3)*
     #C(1,1,4)*C(1,3,2))/A))
C
      VC(2,1,1)=V1
      VC(1,2,1)=V2
      VC(1,1,2)=V3
      VC(3,1,1)=V11
      VC(2,2,1)=V12
      VC(2,1,2)=V13
      VC(1,3,1)=V22
      VC(1,2,2)=V23
      VC(1,1,3)=V33
      VC(4,1,1)=V111
      VC(3,2,1)=V112
      VC(3,1,2)=V113
      VC(2,3,1)=V122
      VC(2,2,2)=V123
      VC(2,1,3)=V133
      VC(1,4,1)=V222
      VC(1,3,2)=V223
      VC(1,2,3)=V233
      VC(1,1,4)=V333
      VC(5,1,1)=V1111
      VC(4,2,1)=V1112
      VC(4,1,2)=V1113
      VC(3,3,1)=V1122
      VC(3,2,2)=V1123
      VC(3,1,3)=V1133
      VC(2,4,1)=V1222
      VC(2,3,2)=V1223
      VC(2,2,3)=V1233
      VC(2,1,4)=V1333
      VC(1,5,1)=V2222
      VC(1,4,2)=V2223
      VC(1,3,3)=V2233
      VC(1,2,4)=V2333
      VC(1,1,5)=V3333
C
C *** SE PROCEDE A LA ESCRITURA DE LOS RESULTADOS: MOEMNTOS Y SUS ERRORE
S
C *** SE CALCA DE LA RUTINA DE XESCA PARA EVITAR COMPLICACIONES
C
      WRITE(*,*)'SE ESCRIBEN LOS MOMENTOS CENTRADOS Y SUS ERRORES'
      WRITE(*,*)'ASI COMO LOS MOMENTOS NO CENTRADOS Y SUS ERRORES'
 2013 FORMAT(2X,'INDICE',5X,'MOMENTO CENTRADO',5X,'ERROR',5X,'MOMENTO NO
```

```
     #CENTRADO',5X,'ERROR')
 2014 FORMAT(2X,3I1,5X,F12.2,5X,F12.2,5X,F12.2,5X,F12.2)
      DO 2039 LL=4,7
      LLL=LL-3
      WRITE(2,2015)LLL
      DO 2040 KA=1,5
      DO 2041 J=1,5
      DO 2042 I=1,5
      IJK=I+J+KA
      IF(IJK.NE.LL)GOTO 2042
      II=I-1
      JJ=J-1
      KK=KA-1
      WRITE(2,2014)II,JJ,KK,C(I,J,KA),VC(I,J,KA),S(I,J,KA),VS(I,J,KA)
 2042 CONTINUE
 2041 CONTINUE
 2040 CONTINUE
 2039 CONTINUE
 2015 FORMAT(2X,'ORDEN',I1)
C
C *** SE CALCULAN E IMPRIMEN LOS CUMULANTES
C
      CALL CALCUM(C,VC)
C
      CLOSE(1)
      CLOSE(2)
C
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LOS CUMULANTES Y SUS ERRORES
C
      SUBROUTINE CALCUM(C,VC)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL*8 NU4,M2,M4
      DIMENSION M2(3,3),M4(3,3,3,3),NU4(3,3,3,3),EM2(3,3)
      DIMENSION EM4(3,3,3,3),ENU4(3,3,3,3),C(9,9,9),VC(5,5,5)
C ***
      WRITE(*,*)'ACCESO A RUTINA CALCUM'
C ***
C
C *** IDENTIFICACION DE VARIABLES
C
      M2(1,1)=C(3,1,1)
      M2(1,2)=C(2,2,1)
      M2(2,2)=C(1,3,1)
      M2(1,3)=C(2,1,2)
      M2(2,3)=C(1,2,2)
      M2(3,3)=C(1,1,3)
      EM2(1,1)=VC(3,1,1)
      EM2(1,2)=VC(2,2,1)
      EM2(2,2)=VC(1,3,1)
      EM2(1,3)=VC(2,1,2)
      EM2(2,3)=VC(1,2,2)
      EM2(3,3)=VC(1,1,3)
C
      DO 400 I=1,3
      DO 410 J=I,3
      M2(I,J)=M2(I,J)
      M2(J,I)=M2(I,J)
      EM2(I,J)=EM2(I,J)
      EM2(J,I)=EM2(I,J)
 410  CONTINUE
 400  CONTINUE
C
C
      M4(1,1,1,1)=C(5,1,1)
      M4(1,1,1,2)=C(4,2,1)
      M4(1,1,2,2)=C(3,3,1)
      M4(1,2,2,2)=C(2,4,1)
      M4(2,2,2,2)=C(1,5,1)
      M4(1,1,1,3)=C(4,1,2)
      M4(1,1,2,3)=C(3,2,2)
      M4(1,2,2,3)=C(2,3,2)
      M4(2,2,2,3)=C(1,4,2)
      M4(1,1,3,3)=C(3,1,3)
      M4(1,2,3,3)=C(2,2,3)
      M4(2,2,3,3)=C(1,3,3)
      M4(1,3,3,3)=C(2,1,4)
      M4(2,3,3,3)=C(1,2,4)
      M4(3,3,3,3)=C(1,1,5)
      EM4(1,1,1,1)=VC(5,1,1)
      EM4(1,1,1,2)=VC(4,2,1)
      EM4(1,1,2,2)=VC(3,3,1)
```

```
      EM4(1,2,2,2)=VC(2,4,1)
      EM4(2,2,2,2)=VC(1,5,1)
      EM4(1,1,1,3)=VC(4,1,2)
      EM4(1,1,2,3)=VC(3,2,2)
      EM4(1,2,2,3)=VC(2,3,2)
      EM4(2,2,2,3)=VC(1,4,2)
      EM4(1,1,3,3)=VC(3,1,3)
      EM4(1,2,3,3)=VC(2,2,3)
      EM4(2,2,3,3)=VC(1,3,3)
      EM4(1,3,3,3)=VC(2,1,4)
      EM4(2,3,3,3)=VC(1,2,4)
      EM4(3,3,3,3)=VC(1,1,5)
C
      DO 450 I=1,3
      DO 460 J=I,3
      DO 470 K=J,3
      DO 480 L=K,3
      M4(I,J,K,L)=M4(I,J,K,L)
      M4(I,K,J,L)=M4(I,J,K,L)
      M4(I,L,K,J)=M4(I,J,K,L)
      M4(I,J,L,K)=M4(I,J,K,L)
      M4(I,K,L,J)=M4(I,J,K,L)
      M4(I,L,J,K)=M4(I,J,K,L)
      M4(J,I,K,L)=M4(I,J,K,L)
      M4(J,K,I,L)=M4(I,J,K,L)
      M4(J,L,K,I)=M4(I,J,K,L)
      M4(J,I,L,K)=M4(I,J,K,L)
      M4(J,K,L,I)=M4(I,J,K,L)
      M4(J,L,I,K)=M4(I,J,K,L)
      M4(K,I,J,L)=M4(I,J,K,L)
      M4(K,J,I,L)=M4(I,J,K,L)
      M4(K,L,J,I)=M4(I,J,K,L)
      M4(K,I,L,J)=M4(I,J,K,L)
      M4(K,J,L,I)=M4(I,J,K,L)
      M4(K,L,I,J)=M4(I,J,K,L)
      M4(L,I,J,K)=M4(I,J,K,L)
      M4(L,J,I,K)=M4(I,J,K,L)
      M4(L,K,J,I)=M4(I,J,K,L)
      M4(L,I,K,J)=M4(I,J,K,L)
      M4(L,J,K,I)=M4(I,J,K,L)
      M4(L,K,I,J)=M4(I,J,K,L)
      EM4(I,J,K,L)=EM4(I,J,K,L)
      EM4(I,K,J,L)=EM4(I,J,K,L)
      EM4(I,L,K,J)=EM4(I,J,K,L)
      EM4(I,J,L,K)=EM4(I,J,K,L)
      EM4(I,K,L,J)=EM4(I,J,K,L)
      EM4(I,L,J,K)=EM4(I,J,K,L)
      EM4(J,I,K,L)=EM4(I,J,K,L)
      EM4(J,K,I,L)=EM4(I,J,K,L)
      EM4(J,L,K,I)=EM4(I,J,K,L)
      EM4(J,I,L,K)=EM4(I,J,K,L)
      EM4(J,K,L,I)=EM4(I,J,K,L)
      EM4(J,L,I,K)=EM4(I,J,K,L)
      EM4(K,I,J,L)=EM4(I,J,K,L)
      EM4(K,J,I,L)=EM4(I,J,K,L)
      EM4(K,L,J,I)=EM4(I,J,K,L)
```

```
      EM4(K,I,L,J)=EM4(I,J,K,L)
      EM4(K,J,L,I)=EM4(I,J,K,L)
      EM4(K,L,I,J)=EM4(I,J,K,L)
      EM4(L,I,J,K)=EM4(I,J,K,L)
      EM4(L,J,I,K)=EM4(I,J,K,L)
      EM4(L,K,J,I)=EM4(I,J,K,L)
      EM4(L,I,K,J)=EM4(I,J,K,L)
      EM4(L,J,K,I)=EM4(I,J,K,L)
      EM4(L,K,I,J)=EM4(I,J,K,L)
  480 CONTINUE
  470 CONTINUE
  460 CONTINUE
  450 CONTINUE
C
C *** CALCULO DE NU4
C
      DO 100 I=1,3
      DO 110 J=1,3
      DO 120 K=1,3
      DO 130 L=1,3
      NU4(I,J,K,L)=M4(I,J,K,L)-(M2(I,J)*M2(K,L)+M2(I,K)*M2(J,L)+M2(I,L)*
     #M2(J,K))
  130 CONTINUE
  120 CONTINUE
  110 CONTINUE
  100 CONTINUE
C
C *** SE CONVIERTE EL ERROR DE EM4 A ENU4
C *** PROPAGACION CUADRATICA DEL ERROR
C
      DO 1300 I=1,3
      DO 1310 J=1,3
      DO 1320 K=1,3
      DO 1330 L=1,3
      ENU4(I,J,K,L)=DSQRT(EM4(I,J,K,L)**2+(DABS(M2(I,J)*EM2(K,L))+DAB
     #S(EM2(I,J)*M2(K,L)))**2+(DABS(M2(I,K)*EM2(J,L))+DABS(EM2(I,K)
     #*M2(J,L)))**2+(DABS(M2(I,L)*EM2(J,K))+DABS(EM2(I,L)*M2(J,K)))
     #**2)
 1330 CONTINUE
 1320 CONTINUE
 1310 CONTINUE
 1300 CONTINUE
C
C *** SE ESCRIBE EL FICHERO
C
      WRITE(*,*)'SE ESCRIBEN LOS CUMULANTES'
      WRITE(2,*)'CUMULANTES'
      DO 2000 I=1,3
      DO 2010 J=1,3
      DO 2020 K=1,3
      DO 2030 L=1,3
      IF(K.GT.L) GO TO 2030
      IF(J.GT.K) GO TO 2020
      IF(I.GT.J) GO TO 2010
      WRITE(2,3000)I,J,K,L,NU4(I,J,K,L),ENU4(I,J,K,L)
 2030 CONTINUE
```

```
 2020 CONTINUE
 2010 CONTINUE
 2000 CONTINUE
C
 3000 FORMAT(2X,4I1,5X,F12.2,5X,F12.2)
C
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LASS DD MAYUSCULAS
C
      SUBROUTINE CALDD(MM3,VV,D,DD,C33)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*8 MM3
      REAL*4 AUX
      DIMENSION D(3),DD(3)
      DIMENSION VV(2),MM3(3,3,3)
C ***
      WRITE(*,*)'ACCESO A RUTINA CALDD'
C ***
C *** VV(1)=(1/DD(3))**2 Y VV(2)=C33/DD(3)
      IF(VV(1).LT.0.D0)THEN
                       WRITE(*,*)'RAIZ NEGATIVA',VV(1)
      ENDIF
      WRITE(*,*)'MINIMOS 1 Y 2 **********',VV(1),VV(2)
      DD(3)=DSQRT(VV(1))
      SIGNO=3.D0*VV(2)-MM3(3,3,3)*VV(1)
      IF(SIGNO.GT.0.D0)THEN
                       DD(3)=-DD(3)
                       ELSE
                       DD(3)=DD(3)
      ENDIF
      DD(3)=1.D0/DD(3)
C *** C33 SE CALCULA A PARTIR DE DD(3)
C     AUX=SNGL(VV(2)*DD(3))
      C33=VV(2)*DD(3)
C ***
C *** SE CALCULAN LAS D MAYUSCULAS DD EN FUNCION DE LAS MINUSCULAS D
C *** RECORDAR QUE D(3)=1
      DD(1)=D(1)*DD(3)
      DD(2)=D(2)*DD(3)
      DD(3)=D(3)*DD(3)
C
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LASS DD MAYUSCULAS Y SUS ERRORES
C ***** ES IGUAL QUE CALDD PERO CON CALCULO DE ERRORES
C
      SUBROUTINE CALDDE(MM3,VV,D,DD,C33,EVEC,EDD,EC33)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*8 MM3
      REAL*4 AUX
      DIMENSION D(3),DD(3)
      DIMENSION VV(2),MM3(3,3,3),EVEC(2),EDD(3)
C ***
      WRITE(*,*)'ACCESO A RUTINA CALDDE'
C ***
C *** VV(1)=(1/DD(3))**2 Y VV(2)=C33/DD(3) ES DECIR VV=VEC
      IF(VV(1).LT.0.D0)THEN
                       WRITE(*,*)'RAIZ NEGATIVA',VV(1)
      ENDIF
      WRITE(*,*)'MINIMOS 1 Y 2 **********',VV(1),VV(2)
      DD(3)=DSQRT(VV(1))
      SIGNO=3.D0*VV(2)-MM3(3,3,3)*VV(1)
      IF(SIGNO.GT.0.D0)THEN
                       DD(3)=-DD(3)
                       ELSE
                       DD(3)=DD(3)
      ENDIF
      DD(3)=1.D0/DD(3)
C *** C33 SE CALCULA A PARTIR DE DD(3)
C     AUX=SNGL(VV(2)*DD(3))
      C33=VV(2)*DD(3)
C ***
C *** SE CALCULAN LAS D MAYUSCULAS DD EN FUNCION DE LAS MINUSCULAS D
C *** RECORDAR QUE D(3)=1
      DD(1)=D(1)*DD(3)
      DD(2)=D(2)*DD(3)
      DD(3)=D(3)*DD(3)
C
C *** Y AHORA SE CALCULAN LOS ERRORES DE LOS RESULTADOS
C
C *** PREVIAMENTE HACEN FALTA LOS ERRORES DE D3 Y C33, ED3 Y EC33
      ED3=1/2.D0*EVEC(1)/DSQRT(VV(1)**3)
      EC33=DSQRT((VV(2)*ED3)**2+(EVEC(2)*DD(3))**2)
      WRITE(*,*)'ERROR DE D3',ED3
      WRITE(*,*)'ERROR DE C33',EC33
C
      DO 100 I=1,3
      EDD(I)=D(I)*ED3
 100  CONTINUE
C
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LAS DELTAS Y VELOCIDAD DEL CENTROIDE DE LA MUESTRA TO
TAL
C
      SUBROUTINE CALDEL(F,V,NP,DELTA,VV)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION F(5),V(5,3),DELTA(5,3)
      DIMENSION VV(3)
C ***
      WRITE(*,*)'ACCESO A RUTINA CALDEL'
C ***
C
C ********** ESPECIFICACION DE CADA VARIABLE
C F=FRACCION DE POBLACION
C V=COMPONENTES DE LA VELOCIDAD DE LOS CENTROIDES DE LAS MUESTRAS PARCIA
LES
C DELTA=DIFERENCIAS DE LAS VELOCIDADES DE LOS CENTROIDES PARCIALES Y TOT
AL
C VV=VELOCIDAD DEL CENTROIDE DE LA MUESTRA TOTAL
C
C *** CALCULO DE LAS COMPONENTES DE LA VELOCIDAD
C *** DEL CENTROIDE DE LA MUESTRA TOTAL
      DO 200 I=1,3
      VV(I)=0.D0
      DO 210 N=1,NP
      VV(I)=VV(I)+F(N)*V(N,I)
 210  CONTINUE
 200  CONTINUE
C *** CALCULO DE LAS DELTAS
      DO 300 N=1,NP
      DO 310 I=1,3
      DELTA(N,I)=V(N,I)-VV(I)
 310  CONTINUE
 300  CONTINUE
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LAS D
C
      SUBROUTINE CALDES(MM3,D)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*8 MM3,MATRI
      DIMENSION MM3(3,3,3),D(3),D1(3),D2(3),ECU(2,3,3)
      DIMENSION MATRI(14,2),VECT(14),V(2)
C ***
      WRITE(*,*)'ACCESO A RUTINA CALDES'
C ***
C *** SE GUARDA LA D(1) DEL CASO DE SIMETRIA CILINDRICA
      D1SIM=D(1)
      WRITE(*,*)'*** D1 CON SIMETRIA CILINDRICA',D1SIM
C
C *** SE RESUELVEN LAS ECUACIONES CUBICAS
C
C *** RECORDAR D3=1. RECORDAR D MINUSCULAS SON D Y MAYUSCULAS DD
C *** CALCULO DE D1/D3
C *** LA SOLUCION SE DEJA EN EL VECTOR D1
      D(3)=1.D0
      A=-MM3(3,3,3)
      B=3.D0*MM3(1,3,3)
      C=-3.D0*MM3(1,1,3)
      DE=MM3(1,1,1)
      WRITE(*,*)'ABCD SON',A,B,C,DE
      CALL EDTG(A,B,C,DE,D1,N1)
C *** CALCULO DE D2/D3
C *** LA SOLUCION SE DEJA EN EL VECTOR D2
      A=-MM3(3,3,3)
      B=3.D0*MM3(3,3,2)
      C=-3.D0*MM3(3,2,2)
      DE=MM3(2,2,2)
      WRITE(*,*)'ABCD SON',A,B,C,DE
      CALL EDTG(A,B,C,DE,D2,N2)
C
C *** COMPROBACION DE CUAL DE LAS SOLUCIONES OBTENIDAS ES LA CORRECTA
C
C *** SE CALCULAN LAS DOS ECUACIONES SIGUIENTES CON LOS VALORES OBTENIDO
S
      DO 100 I=1,N1
      DO 110 J=1,N2
C
      ECU(1,I,J)=-MM3(3,3,3)*D1(I)**2*D2(J)+MM3(3,3,2)*D1(I)**2*D(3)
      ECU(1,I,J)=ECU(1,I,J)+2.D0*MM3(1,3,3)*D1(I)*D2(J)*D(3)
      ECU(1,I,J)=ECU(1,I,J)-2.D0*MM3(1,3,2)*D1(I)*D(3)**2
      ECU(1,I,J)=ECU(1,I,J)-MM3(1,1,3)*D(3)**2*D2(J)+MM3(1,1,2)*D(3)**3
      ECU(1,I,J)=DABS(ECU(1,I,J))
C
      ECU(2,I,J)=-MM3(3,3,3)*D1(I)*D2(J)**2+MM3(1,3,3)*D2(J)**2*D(3)
      ECU(2,I,J)=ECU(2,I,J)+2.D0*MM3(3,3,2)*D1(I)*D2(J)*D(3)
      ECU(2,I,J)=ECU(2,I,J)-2.D0*MM3(1,3,2)*D(3)**2*D2(J)
      ECU(2,I,J)=ECU(2,I,J)-MM3(3,2,2)*D1(I)*D(3)**2+MM3(1,2,2)*D(3)**3
      ECU(2,I,J)=DABS(ECU(2,I,J))
C
```

```
 110  CONTINUE
 100  CONTINUE
C * PRESENTACION DE LAS SOLUCIONES DE LAS ECUACIONES DE TERCER GRADO
      WRITE(*,*)' '
      WRITE(*,*)'**** LOS DOS TRIOS DE SOLUCIONES SON *****'
      DO 300 I=1,N1
      WRITE(*,*)'VALOR',I,' DE D1',D1(I)
 300  CONTINUE
      DO 310 I=1,N1
      WRITE(*,*)'VALOR',I,' DE D2',D2(I)
 310  CONTINUE
C *** SE COMPRUEBA CUAL DE LAS SOLUCIONES SE CUMPLE MEJOR
      CERO=1.D10
      DO 200 N=1,2
      DO 210 I=1,N1
      DO 220 J=1,N2
      IF(ECU(N,I,J).LT.CERO) THEN
                             CERO=ECU(1,I,J)
                             IND1=I
                             IND2=J
      ENDIF
 220  CONTINUE
 210  CONTINUE
 200  CONTINUE
C
      DE1=D1(IND1)
      DE2=D2(IND2)
      WRITE(*,*)'LAS D INICIALES SON',DE1,DE2,D(3)
C
      IF(DE1.NE.0.D0.AND.DE2.NE.0.D0)THEN
C
C *** SE INTENTA APROXIMAR LA SOLUCION DE D1 Y D2
C *** CON LAS DOS ECUACIONES SIGUIENTES SE OBTIENEN DOS NUEVOS VALORES
C *** DE D1 Y D2. D3 SE CORRESPONDE A D1 Y D4 A D2
C *** ESTOS NUEVOS VALORES SE CALCULAN MEDIANTE UN PROCESO ITERATIVO DE
C *** 20 ITERACIONES
C
      DE3=DE1
      DE4=DE2
      DO 400 I=1,20
      DIVID3=MM3(1,3,3)*DE4**2-2.D0*MM3(1,3,2)*DE4+MM3(1,2,2)
      DENOM3=MM3(3,3,3)*DE4**2-2.D0*MM3(3,3,2)*DE4+MM3(3,2,2)
      DIVID4=MM3(3,3,2)*DE3**2-2.D0*MM3(1,3,2)*DE3+MM3(1,1,2)
      DENOM4=MM3(3,3,3)*DE3**2-2.D0*MM3(1,3,3)*DE3+MM3(1,1,3)
      DE3=DIVID3/DENOM3
      DE4=DIVID4/DENOM4
C
C     WRITE(*,*)'*** ITERACION DE D1:',DE3
C     WRITE(*,*)'********** ITERACION DE D2:',DE4
 400  CONTINUE
C
C *** SE CALCULAN LAS TANGENTES A LAS CUATRO FUNCIONES EN EL PUNTO
C *** SE COMIENZA CALCULANDO LAS PENDIENTES
C
      P1=-3.D0*MM3(3,3,3)*DE1**2+6.D0*MM3(1,3,3)*DE1-3.D0*MM3(1,1,3)
      P2=-3.D0*MM3(3,3,3)*DE2**2+6.D0*MM3(3,3,2)*DE2-3.D0*MM3(3,2,2)
```

```
       P31=-2.D0*MM3(3,3,3)*DE3*DE4+2.D0*MM3(3,3,2)*DE3+
      #2.D0*MM3(1,3,3)*DE4-2.D0*MM3(1,3,2)
       P32=-MM3(3,3,3)*DE3**2+2.D0*MM3(1,3,3)*DE3-MM3(1,1,3)
       P41=-MM3(3,3,3)*DE4**2+2.D0*MM3(3,3,2)*DE4-MM3(3,2,2)
       P42=-2.D0*MM3(3,3,3)*DE3*DE4+2.D0*MM3(1,3,3)*DE4+
      #2.D0*MM3(3,3,2)*DE3-2.D0*MM3(1,3,2)
C
C *** LAS ECUACIONES DE LAS TANGENTES QUEDARAN
C *** P1*D(1)=P1*DE1
C *** P2*D(2)=P2*DE2
C *** P31*D(1)+P32*D(2)=P31*DE3+P32*DE4
C *** P41*D(1)+P42*D(2)=P41*DE3+P42*DE4
C *** CON TODO LO CUAL SE CONSTRUYE EL SISTEMA DE MINIMOS CUADRADOS
C *** MATRI Y VECT ESTAN DIMENSIONADOS A 14 PARA PODER UTILIIZAR LA
C *** RUTINA MINCUA PERO LA DIMENSION UTILIZADA ES 4
C
      DO 500 I=1,14
      VECT(I)=0.D0
      DO 510 J=1,2
      MATRI(I,J)=0.D0
 510  CONTINUE
 500  CONTINUE
C
      MATRI(1,1)=P1
      MATRI(1,2)=0.D0
      MATRI(2,1)=0.D0
      MATRI(2,2)=P2
      MATRI(3,1)=P31
      MATRI(3,2)=P32
      MATRI(4,1)=P41
      MATRI(4,2)=P42
      VECT(1)=P1*DE1
      VECT(2)=P2*DE2
      VECT(3)=P31*DE3+P32*DE4
      VECT(4)=P41*DE3+P42*DE4
      NDIM=4
      CALL MINCUA(MATRI,VECT,NDIM,V)
      D(1)=V(1)
      D(2)=V(2)
C
      WRITE(*,*)'LAS D FINALES SON',D(1),D(2),D(3)
      WRITE(*,*)'ASI PUES LAS D SON'
      WRITE(*,*)'DE O111 y O222',DE1,DE2
      WRITE(*,*)'DE O112 y O122',DE3,DE4
      WRITE(*,*)'PUEDE HACERSE UNA MEDIA ARITMETICA'
      WRITE(*,*)'DEL SISTEMA DE MINIMOS CUADRADOS',D(1),D(2)
      WRITE(*,*)'ELIJA LAS D:'
      WRITE(*,*)'1=O111 y O222, 2=O112 y O122, 3=MEDIA,4=MINCUA,5=CARTA'
      WRITE(*,*)'6=SIMETRIA CILINDRICA,7=Z=0'
      READ(*,*)KK
C *** SI SE ELIGE 4 O UN NUMERO MAYOR QUE 7 UTLIZA MINIMOS CUADRADOS
C     KK=4
      IF(KK.EQ.1)THEN
         D(1)=DE1
         D(2)=DE2
      ELSE
```

```
            IF(KK.EQ.2)THEN
               D(1)=DE3
               D(2)=DE4
            ELSE
               IF(KK.EQ.3)THEN
                  D(1)=(DE1+DE3)/2.D0
                  D(2)=(DE2+DE4)/2.D0
               ELSE
                  IF(KK.EQ.5)THEN
                     WRITE(*,*)'ENTRE D1'
                     READ(*,*)D(1)
                     WRITE(*,*)'ENTRE D2'
                     READ(*,*)D(2)
                  ELSE
                     IF(KK.EQ.6)THEN
                        WRITE(*,*)'*** SE ADOPTA SIMETRIA CILINDRICA ***'
                        D(1)=D1SIM
                     ELSE
                        IF(KK.EQ.7)THEN
                           WRITE(*,*)'*** SE ADOPTA Z=0 ***'
                           D(1)=MM3(1,2,2)/MM3(2,2,3)
                           D(2)=0.D0
                        ELSE
                           CONTINUE
         ENDIF
         ENDIF
         ENDIF
         ENDIF
         ENDIF
         ENDIF
C
         ELSE
            WRITE(*,*)'PUESTO QUE UNA D ES NULA NO PROCEDEN OTROS CALCULOS'
            WRITE(*,*)'ELIJA LAS D:'
            WRITE(*,*)'1=O111 y O222,2=SIMETRIA CILINDRICA,3=Z=0,4=CARTA'
            READ(*,*)KK
            IF(KK.EQ.2)THEN
               WRITE(*,*)'*** SE ADOPTA SIMETRIA CILINDRICA ***'
               D(1)=D1SIM
            ELSE
               IF(KK.EQ.3)THEN
                  WRITE(*,*)'*** SE ADOPTA Z=0 ***'
                  D(1)=MM3(1,2,2)/MM3(2,2,3)
                  D(2)=0.D0
               ELSE
                  IF(KK.EQ.4)THEN
                     WRITE(*,*)'ENTRE D1'
                     READ(*,*)D(1)
                     WRITE(*,*)'ENTRE D2'
                     READ(*,*)D(2)
                  ELSE
                     D(1)=DE1
                     D(2)=DE2
C
         ENDIF
         ENDIF
```

```
      ENDIF
      ENDIF
      WRITE(*,*)'LAS D DEFINITIVAS SON',D(1),D(2),D(3)
C
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LAS MAGNITUDES DE LA MUESTRA TOTAL
C
      SUBROUTINE CALMOM(M2,DELTA,F,NP,MM2,MM3,MM4)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*4 AUX
      REAL*8 M2,M4,MM2,MM3,MM4
      DIMENSION M2(5,3,3),M4(5,3,3,3,3),F(5),DELTA(5,3)
      DIMENSION MM2(3,3),MM4(3,3,3,3),MM3(3,3,3)
C ***
      WRITE(*,*)'ACCESO A RUTINA CALMOM'
C ***
C
C ********** DEFINICION DE LAS VARIABLES
C MX=MOMENTOS DE ORDEN X DE CADA MUESTRA
C F=FRACCION DE POBLACION DE CADA MUESTRA
C DELTA=DELTAS DE CADA POBLACION
C MMX=MOMENTOS DE ORDEN X DE LA MUESTRA TOTAL
C
C
C *** MOMENTOS DE SEGUNDO ORDEN DE LA MUESTRA TOTAL
C
      DO 110 I=1,3
      DO 120 J=1,3
      MM2(I,J)=0.D0
      DO 100 N=1,NP
      MM2(I,J)=MM2(I,J)+F(N)*(M2(N,I,J)+DELTA(N,I)*DELTA(N,J))
C     AUX=SNGL(MM2(I,J))
C     MM2(I,J)=AUX
 100  CONTINUE
      WRITE(*,*)'SEGUNDO ORDEN',I,J,MM2(I,J)
 120  CONTINUE
 110  CONTINUE
C
C *** MOMENTOS DE TERCER ORDEN
C
      DO 510 I=1,3
      DO 520 J=1,3
      DO 530 K=1,3
      MM3(I,J,K)=0.D0
      DO 500 N=1,NP
      MM3(I,J,K)=MM3(I,J,K)+F(N)*M2(N,I,J)*DELTA(N,K)
      MM3(I,J,K)=MM3(I,J,K)+F(N)*M2(N,J,K)*DELTA(N,I)
      MM3(I,J,K)=MM3(I,J,K)+F(N)*M2(N,K,I)*DELTA(N,J)
      MM3(I,J,K)=MM3(I,J,K)+F(N)*DELTA(N,I)*DELTA(N,J)*DELTA(N,K)
 500  CONTINUE
      WRITE(*,*)'TERCER ORDEN',I,J,K,MM3(I,J,K)
 530  CONTINUE
 520  CONTINUE
 510  CONTINUE
C
C *** MOMENTOS DE CUARTO ORDEN DE LA MUESTRA TOTAL
C
C *** MOMENTOS PARCIALES DE CUARTO ORDEN A PARTIR DE LOS DE SEGUNDO
      DO 200 N=1,NP
```

```
      DO 210 I=1,3
      DO 220 J=1,3
      DO 230 K=1,3
      DO 240 L=1,3
      M4(N,I,J,K,L)=M2(N,I,J)*M2(N,K,L)+M2(N,I,K)*M2(N,J,L)+
     #M2(N,I,L)*M2(N,J,K)
 240  CONTINUE
 230  CONTINUE
 220  CONTINUE
 210  CONTINUE
 200  CONTINUE
C *** MOMENTOS DE CUARTO ORDEN TOTALES
      DO 310 I=1,3
      DO 320 J=1,3
      DO 330 K=1,3
      DO 340 L=1,3
      MM4(I,J,K,L)=0.D0
      DO 300 N=1,NP
      MM4(I,J,K,L)=MM4(I,J,K,L)+F(N)*(M4(N,I,J,K,L)+M2(N,I,J)*DELTA(N,K)
     #*DELTA(N,L)+M2(N,J,K)*DELTA(N,L)*DELTA(N,I)+M2(N,K,L)*DELTA(N,I)*D
     #ELTA(N,J)+M2(N,L,I)*DELTA(N,J)*DELTA(N,K)+M2(N,I,K)*DELTA(N,L)*DEL
     #TA(N,J)+M2(N,J,L)*DELTA(N,I)*DELTA(N,K)+DELTA(N,I)*DELTA(N,J)*DELT
     #A(N,K)*DELTA(N,L))
 300  CONTINUE
      WRITE(*,*)'MOMENTOS DE 444444',I,J,K,L,MM4(I,J,K,L)
 340  CONTINUE
 330  CONTINUE
 320  CONTINUE
 310  CONTINUE
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LOS VALORES DE NUS
C
      SUBROUTINE CALNU(M2,M4,NU4)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL*8 NU4,M2,M4
      DIMENSION M2(3,3),M4(3,3,3,3),NU4(3,3,3,3)
C ***
      WRITE(*,*)'ACCESO A RUTINA CALNU'
C ***
      DO 100 I=1,3
      DO 110 J=1,3
      DO 120 K=1,3
      DO 130 L=1,3
      NU4(I,J,K,L)=M4(I,J,K,L)-(M2(I,J)*M2(K,L)+M2(I,K)*M2(J,L)+M2(I,L)*
     #M2(J,K))
 130  CONTINUE
 120  CONTINUE
 110  CONTINUE
 100  CONTINUE
C
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LOS VALORES DE Q SEGUN LAS DOS EXPRESIONES DISPONIBLE
S
C
      SUBROUTINE CALQU(MM3,NU4,C,DD,FP,Q)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL*8 MM3,NU4
      DIMENSION MM3(3,3,3),NU4(3,3,3,3),C(3,3),DD(3),FP(2)
C ***
      WRITE(*,*)'MM3 DE 3,3,3 VALE',MM3(3,3,3)
      WRITE(*,*)'NU4 DE 3,3,3,3 VALE',NU4(3,3,3,3)
C
      Q1=(MM3(3,3,3)-3.D0*C(3,3)*DD(3))/(2.D0*DD(3)**3)
      WRITE(*,*)'VALOR DE Q SEGUN MM3',Q1
      Q2=(3.D0*C(3,3)**2-NU4(3,3,3,3))/(2.D0*DD(3)**4)-1.D0
      IF(Q2.GE.0.D0)THEN
                    Q2=DSQRT(Q2)
                    ELSE
                    WRITE(*,*)'NO FUNCIONA PARA NU4 !!!!!'
                    Q2=Q1
      ENDIF
      WRITE(*,*)'VALOR DE Q SEGUN NU4',Q2
C
      WRITE(*,*)'*** ESCOJA VALOR DE Q (1 si MM3, 2 si NU4), 3 si PROMED
     #IO ***'
      READ(*,*)NQ
C     NQ=2
      IF(NQ.EQ.1)THEN
                 WRITE(*,*)'Q OBTENIDO DE MM3'
                 Q=Q1
                 ELSE
                 IF(NQ.NE.3)THEN
                            Q=Q2
                            ELSE
                            WRITE(*,*)'SE TOMA UNA Q PROMEDIO'
                            Q=(Q1+Q2)/2.D0
                 ENDIF
      ENDIF
C
C *** SE CALCULAN LAS FRACIONES DE POBLACION
C
      FP(1)=0.5D0*(1.D0+DSQRT(1.D0-4.D0/(Q**2+4.D0)))
      FP(2)=1.D0-FP(1)
      WRITE(*,100)FP(1),FP(2)
 100  FORMAT(1X,' *** FRACCIONES DE POBLACION *** ',F5.2,F5.2)
C
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LOS VALORES DE Q SEGUN LAS DOS EXPRESIONES DISPONIBLE
S
C ***** SE OBTIENE EL ERROR DE LA FRACCION DE POBLACION SEGUN LA Q DE
C ***** LOS MOMENTOS DE TERCER ORDEN
C ***** SE APARCA EL ERROR SEGUN LA Q PROVENIENTE DE LAS NU
C
      SUBROUTINE CALQUE(MM3,NU4,C,DD,FP,Q,EMM3,EC,EDD,EFP,EQ)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL*8 MM3,NU4
      DIMENSION MM3(3,3,3),NU4(3,3,3,3),C(3,3),DD(3),FP(2)
      DIMENSION EC(3,3),EDD(3),EMM3(3,3,3)
C ***
      WRITE(*,*)'MM3 DE 3,3,3 VALE',MM3(3,3,3)
      WRITE(*,*)'NU4 DE 3,3,3,3 VALE',NU4(3,3,3,3)
C
      Q1=(MM3(3,3,3)-3.D0*C(3,3)*DD(3))/(2.D0*DD(3)**3)
      WRITE(*,*)'VALOR DE Q SEGUN MM3',Q1
      Q2=(3.D0*C(3,3)**2-NU4(3,3,3,3))/(2.D0*DD(3)**4)-1.D0
      IF(Q2.GE.0.D0)THEN
                    Q2=DSQRT(Q2)
                    ELSE
                    WRITE(*,*)'NO FUNCIONA PARA NU4 !!!!!'
                    Q2=Q1
      ENDIF
      WRITE(*,*)'VALOR DE Q SEGUN NU4',Q2
C
      WRITE(*,*)'*** ESCOJA VALOR DE Q (1 si MM3, 2 si NU4), 3 si PROMED
     #IO ***'
      READ(*,*)NQ
C     NQ=2
      IF(NQ.EQ.1)THEN
                 WRITE(*,*)'Q OBTENIDO DE MM3'
                 Q=Q1
                 ELSE
                 IF(NQ.NE.3)THEN
                            Q=Q2
                            ELSE
                            WRITE(*,*)'SE TOMA UNA Q PROMEDIO'
                            Q=(Q1+Q2)/2.D0
                 ENDIF
      ENDIF
C
C *** SE CALCULAN LAS FRACIONES DE POBLACION
C
      FP(1)=0.5D0*(1.D0+DSQRT(1.D0-4.D0/(Q**2+4.D0)))
      FP(2)=1.D0-FP(1)
      WRITE(*,100)FP(1),FP(2)
 100  FORMAT(1X,' *** FRACCIONES DE POBLACION *** ',F5.2,F5.2)
C
C *** AHORA EL ERROR DE LA FRACCION DE POBLACION SUPONIENDO LA
C *** Q OBTENIDA DE MM3
C
      EQ1=DSQRT((EMM3(3,3,3)/(2.D0*DD(3)**3))**2+(3.D0*EC(3,3)/(2.D0*
     #DD(3)**2))**2+((C(3,3)/DD(3)**3-MM3(3,3,3)/(2.D0*DD(3)**4))
```

```
     #*3.D0*EDD(3))**2)
      WRITE(*,*)'MM3 EMM3 DD3 EDD3 C33 EC33',MM3(3,3,3),EMM3(3,3,3),
     #DD(3),EDD(3),C(3,3),EC(3,3)
      EQ=EQ1
      WRITE(*,*)'ERROR DE Q',EQ
C *** Y AHORA EL ERROR DE FP1
      EFP1=EQ**2/16.D0*(Q**2+4)/Q**2*(8.D0*Q/(Q**2+4)**2)**2
      EFP=DSQRT(EFP1)
      WRITE(*,200)EFP
      EFP=0.D0
      WRITE(*,*)'*** SE ANULA EL ERROR DE N PRIMA ***'
 200  FORMAT(1X,'ERROR DE LA FRACCION DE POBLACION',F4.2)
C
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DEL TENSOR C2
C
      SUBROUTINE CESDES(NU4,DD,C)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*4 AUX
      REAL*8 NU4
      DIMENSION NU4(3,3,3,3),C(3,3),DD(3)
C ***
      WRITE(*,*)'ACCESO A RUTINA CESDES'
C ***
      VALOR=(NU4(3,3,3,3)-3.D0*C(3,3)**2)/DD(3)**4
C
      DO 200 I=1,2
      RAIZ=1.D0/3.D0*(NU4(I,I,I,I)-DD(I)**4*VALOR)
      IF(RAIZ.LT.0.D0)THEN
                      WRITE(*,*)'RAIZ DE NUMERO NEGATIVO',RAIZ
      ELSE
                      RAIZ=DSQRT(RAIZ)
      ENDIF
      IF(C(I,I).LT.0.D0)THEN
                        C(I,I)=-RAIZ
                        ELSE
                        C(I,I)=RAIZ
      ENDIF
 200  CONTINUE
      WRITE(*,*)'C2',I,I,C(I,I)
C
      AUX=SNGL(C(1,1))
      C(1,1)=AUX
      AUX=SNGL(C(2,2))
      C(2,2)=AUX
      AUX=SNGL(C(3,3))
      C(3,3)=AUX
C
      DO 100 I=1,2
      DO 110 J=2,3
      IF(I.EQ.J) GOTO 110
      AUX=SNGL(NU4(I,I,J,J))
      NU4(I,I,J,J)=AUX
      RAIZ=1.D0/2.D0*(NU4(I,I,J,J)-C(I,I)*C(J,J)-DD(I)**2*DD(J)**2*VALOR
     #)
      IF(RAIZ.LT.0.D0)THEN
                      WRITE(*,*)'RAIZ DE NUMERO NEGATIVO',RAIZ
      ELSE
                      RAIZ=DSQRT(RAIZ)
      ENDIF
      IF(C(I,J).LT.0.D0)THEN
                        C(I,J)=-RAIZ
                        ELSE
                        C(I,J)=RAIZ
      ENDIF
      C(J,I)=C(I,J)
 110  CONTINUE
 100  CONTINUE
```

```
      DO 300 I=1,3
      DO 310 J=1,3
      WRITE(*,*)'C2',I,J,C(I,J)
 310  CONTINUE
 300  CONTINUE
C
      RETURN
      END
```

```
C234567
C
C *** PROGRAMA DE SELECCION DE LAS ESTRELLAS SEGUN LA POBLACION
C *** A LA QUE PERTENECEN. A PARTIR DE LOS VALORES DE LAS DISPERSIONES
C *** QUE SE OBTIENEN EN EL PROGRAMA DE SEPARACION SE SELECCIONAN LAS
C *** ESTRELLAS SEGUN SU PROBABILIDAD DE PERTENECER A UNA POBLACION U OT
RA
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*8 M2
      DIMENSION FP(2),U(10000),V(10000),W(10000),SIGMA(2,3)
      DIMENSION M2(2,3,3),VEL(2,3),EM2(2,3,3),EV(2,3),P(2)
      WRITE(*,*)'PROGRAMA DE SELECCION DE ESTRELLAS'
C *** APERTURA DE LOS FICHEROS DE DATOS
      OPEN(1,FILE='RESSIN.TXT',STATUS='OLD')
      OPEN(2,FILE='FICMEZ.DAT',STATUS='OLD')
      OPEN(3,FILE='POBLA1.DAT',STATUS='NEW')
      OPEN(4,FILE='POBLA2.DAT',STATUS='NEW')
      PI=3.141592654D0
C
C *** LECTURA DEL FICHERO DE RESULTADOS
      READ(1,*)
      DO 400 N=1,2
      READ(1,*)M2(N,1,1),M2(N,3,3),M2(N,2,2)
      READ(1,*)VEL(N,1),VEL(N,3),VEL(N,2),FP(N)
      IF(N.EQ.1)THEN
         DO 910 I=1,6
         READ(1,*)
 910     CONTINUE
      ELSE
      ENDIF
 400  CONTINUE
 510  FORMAT(1X,F8.2,1X,F8.2,1X,F8.2,1X,F8.2,1X,F8.2,1X,F8.2,1X,F5.1,1X,
     #F5.1,1X,F5.1,1X,F4.2)
C
      DO 800 N=1,2
      DO 810 I=1,3
      SIGMA(N,I)=DSQRT(M2(N,I,I))
 810  CONTINUE
      WRITE(*,*)M2(N,1,1),M2(N,2,2),M2(N,3,3),FP(N)
      WRITE(*,*)SIGMA(N,1),SIGMA(N,2),SIGMA(N,3)
      WRITE(*,*)VEL(N,1),VEL(N,2),VEL(N,3)
 800  CONTINUE
C
C *** LECTURA DEL FICHERO DE ESTRELLAS
      NAA=0
      DO 100 I=1,10000
      READ(2,500,END=7777)U(I),V(I),W(I)
      NAA=NAA+1
 100  CONTINUE
 7777 CONTINUE
 500  FORMAT(F17.10,1X,F17.10,1X,F17.10)
      AA=FLOAT(NAA)
      WRITE(*,*)'TOTAL ESTRELLAS',NAA
C *** CALCULO DE LA PROBABILIDAD DE PERTENENCIA
C *** PUESTO QUE LA FUNCION DE DISTRIBUCION SE SUPONE DE SCHWARZSCHILD
```

```
C *** F(V)=N/((2.PI)**3/2.SIG1.SIG2.SIG3)*EXP(-1/2.SUMA((Vi-vi)/SIGi)**2
)
C
      WRITE(*,*)'INDIQUE FACTOR DE ESCALA'
      READ(*,*)FACT
      NEST1=0
      NEST2=0
      DO 200 I=1,NAA
      DO 300 N=1,2
      P(N)=FP(N)*AA/(DSQRT((2.D0*PI)**3)*SIGMA(N,1)*SIGMA(N,2)*
     #SIGMA(N,3))
      P(N)=P(N)*DEXP(-0.5D0*(((U(I)-VEL(N,1))/SIGMA(N,1))**2+((V(I)-
     #VEL(N,2))/SIGMA(N,2))**2+((W(I)-VEL(N,3))/SIGMA(N,3))**2))
 300  CONTINUE
      P(2)=P(2)*FACT
      IF(P(1).GE.P(2))THEN
         WRITE(3,500)U(I),V(I),W(I)
         NEST1=NEST1+1
      ELSE
         WRITE(4,500)U(I),V(I),W(I)
         NEST2=NEST2+1
      ENDIF
 200  CONTINUE
      WRITE(*,*)'TOTAL ESTRELLAS POBLACION 1',NEST1
      WRITE(*,*)'TOTAL ESTRELLAS POBLACION 2',NEST2
C
      STOP
      END
```

```
C234567
C
C *** PRODUCTO DE LA MATRIZ A Y EL VECTOR B POR LA MATRIZ DE PESOS
C
      SUBROUTINE EAEB(PESO,A,B,EA,EB)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION PESO(14,14),A(14,2),B(14)
      DIMENSION PESOT(14,14),EA(14,2),EB(14)
C ***
      WRITE(*,*)'ACCESO A RUTINA EAEB'
C ***
C
C *** SE ELIMINAN LAS ECUACIONES 0/0
C
      DO 500 I=1,14
      IF(A(I,1).EQ.0.D0.AND.A(I,2).EQ.0.D0)THEN
                                          PESO(I,I)=0.D0
      ENDIF
 500  CONTINUE
C
      DO 1100 I=1,14
      WRITE(*,*)'EL PESO ES',I,I,PESO(I,I)
 1100 CONTINUE
C ***
C
C *** TRASPUESTA DE LA MATRIZ DE PESOS
C
      DO 100 I=1,14
      DO 110 J=1,14
      PESOT(I,J)=PESO(J,I)
 110  CONTINUE
 100  CONTINUE
C
C *** CALCULO DE EA(I,J) MATRIZ A PONDERADA
C
      DO 300 I=1,14
      DO 310 J=1,2
      EA(I,J)=0.D0
      DO 320 K=1,14
      EA(I,J)=EA(I,J)+PESOT(I,K)*A(K,J)
 320  CONTINUE
 310  CONTINUE
 300  CONTINUE
C
C *** CALCULO DE EB(I) VECTOR B PONDERADO
C
      DO 400 I=1,14
      EB(I)=0.D0
      DO 410 K=1,14
      EB(I)=EB(I)+PESOT(I,K)*B(K)
 410  CONTINUE
 400  CONTINUE
C
      WRITE(*,*)'LAS MATRICES SON'
      DO 600 I=1,14
      WRITE(*,*)EA(I,1),EA(I,2),EB(I)
```

```
  600  CONTINUE
C ***
      RETURN
      END
```

```
C234567
C
C ***** RESOLUCION DE ECUACIONES DE SEGUNDO GRADO *****
C
      SUBROUTINE EDSG(A,B,C,X,N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(2)
      WRITE(*,*)'ACCESO A RUTINA EDSG'
C
C SE COMIENZA RESOLVIENDO LOS CASOS TRIVIALES
C
      IF(A.EQ.0.D0.AND.B.NE.0.D0) THEN
                                  X(1)=-C/B
                                  N=N+1
                                  GO TO 100
      ENDIF
      IF(C.EQ.0.D0.AND.B.EQ.0.D0) THEN
                                  X(1)=0.D0
                                  X(2)=0.D0
                                  N=1
                                  GO TO 100
      ENDIF
      IF(C.EQ.0.D0) THEN
                    X(1)=-B/A
                    X(2)=0.D0
                    IF(N.EQ.0) THEN
                               N=N+2
                               ELSE
                               N=N+1
                    ENDIF
                    GO TO 100
      ENDIF
C
C SE PASA AHORA A CALCULAR EL CASO GENERAL
C
      D=B**2-4.D0*A*C
      IF(D.LT.0.D0) THEN
                    N=N
      ENDIF
      IF(D.GT.0.D0) THEN
                    X(1)=1.D0/(2.D0*A)*(-B+DSQRT(D))
                    X(2)=1.D0/(2.D0*A)*(-B-DSQRT(D))
                    N=N+2
      ENDIF
      IF(D.EQ.0.D0) THEN
                    X(1)=-B/(2.D0*A)
                    X(2)=X(1)
                    N=N+1
      ENDIF
 100  CONTINUE
      RETURN
      END
```

```
C234567
C
C ********** RESOLUCION DE ECUACIONES DE  TERCER  GRADO **********
C
      SUBROUTINE EDTG(A,B,C,D,X,N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(3),X1(2)
      WRITE(*,*)'ACCESO A RUTINA EDTG'
C
C EN PRIMER LUGAR SE RESUELVEN LOS CASOS TRIVIALES
C
      IF(A.EQ.0.D0) THEN
                    N=0
                    CALL EDSG(B,C,D,X1,N)
                    X(1)=X1(1)
                    X(2)=X1(2)
                    GO TO 110
      ENDIF
      IF(D.EQ.0.D0.AND.C.EQ.0.D0.AND.B.EQ.0.D0) THEN
                                                X(1)=0.D0
                                                X(2)=0.D0
                                                X(3)=0.D0
                                                N=1
                                                GO TO 110
      ENDIF
      IF(D.EQ.0.D0.AND.C.EQ.0.D0) THEN
                                  X(1)=-B/A
                                  X(2)=0.D0
                                  X(3)=0.D0
                                  N=2
                                  GO TO 110
      ENDIF
      IF(D.EQ.0.D0) THEN
                    X(1)=0.D0
                    N=1
                    CALL EDSG(A,B,C,X1,N)
                    X(2)=X1(1)
                    X(3)=X1(2)
                    GO TO 110
      ENDIF
C
C SE PROCEDE AHORA A RESOLVER LOS CASOS NO PARTICULARES
C
      VAR=B/(3.D0*A)
      Q=1.D0/2.D0*((2.D0*B**3)/(27.D0*A**3)-(B*C)/(3.D0*A**2)+(D/A))
      P=1.D0/3.D0*((3.D0*A*C-B**2)/(3.D0*A**2))
      DIS=Q**2+P**3
C
C SI P ES 0 ENTONCES LAS TRES RAICES SON IGUALES
C
      IF(P.EQ.0.D0) THEN
                    Y1=RACU(-2.D0*Q)
                    Y2=Y1
                    Y3=Y2
                    N=1
                    GO TO 100
```

```
      ENDIF
C
C SI P NO ES 0 PUEDEN HABER RAICES DIFERENTES
C SI DIS ES MENOR O IGUAL QUE CERO HABRAN 3 RAICES REALES
C Y SI ES MAYOR QUE CERO SOLO HABRA UNA RAIZ REAL
C
      IF(DIS.GT.0.D0) THEN
                      N=1
                      CALL S1R(Q,DIS,Y1)
                      ELSE
                      N=3
                      CALL S3R(Q,P,DIS,Y1,Y2,Y3)
      ENDIF
  100 CONTINUE
      X(1)=Y1-VAR
      X(2)=Y2-VAR
      X(3)=Y3-VAR
  110 CONTINUE
      RETURN
      END
C
C *** RUTINA PARA EL CASO DE 3 SOLUCIONES REALES ***
C
      SUBROUTINE S3R(Q,P,DIS,Y1,Y2,Y3)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      WRITE(*,*)'ACCESO A RUTINA S3R'
      PI=DACOS(-1.D0)
      R=DSQRT(DABS(P))
      IF(Q.LT.0.D0) THEN
                    R=-R
                    ELSE
                    R=R
      ENDIF
      IF(DIS.EQ.0.D0) THEN
                      FI=0.D0
                      ELSE
                      FI=DACOS(Q/(R**3))
      ENDIF
      Y1=-2.D0*R*DCOS(FI/3.D0)
      Y2=2.D0*R*DCOS((PI-FI)/3.D0)
      Y3=2.D0*R*DCOS((PI+FI)/3.D0)
      RETURN
      END
C
C *** RUTINA PARA EL CASO DE 1 SOLUCION REAL ***
C
      SUBROUTINE S1R(Q,DIS,Y1)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      WRITE(*,*)'ACCESO A RUTINA S1R'
      U=RACU(-Q+DSQRT(DIS))
      V=RACU(-Q-DSQRT(DIS))
      Y1=U+V
      RETURN
      END
C
C                SOLUCION DE RAICES CUBICAS
```

```
C
      REAL*8 FUNCTION RACU(A)
      REAL*8 A
      IF(A.EQ.0.D0) THEN
                    RACU=0.D0
                    GO TO 100
      ENDIF
      IF(A.LT.0.D0) THEN
                    RACU=-(DEXP((1.D0/3.D0)*DLOG(-A)))
                    ELSE
                    RACU=DEXP((1.D0/3.D0)*DLOG(A))
      ENDIF
  100 CONTINUE
      RETURN
      END
```

```
C234567
C PROGRAMA PARA SELECCIONAR ESTRELLAS DEL CATALOGO HIPPARCOS
C                                                    CON 14.000 ESTRELLAS
C EN FUNCION DEL MODULO DE LA VELOCIDAD
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(1,FILE='HIPPAR.DAT',STATUS='OLD')
      OPEN(2,FILE='HIPSEL.DAT',STATUS='NEW')
C
      WRITE(*,*)'INDIQUE MODALIDAD DE CORTE'
      WRITE(*,*)' 0=UVW MAX, 1=UVW MIN Y MAX, 2=U Y UVW MAX'
      WRITE(*,*)' 3=V Y UVW MAX, 4=W Y UVW MAX'
      READ(*,*)MOD
C
      IF(MOD.EQ.0)THEN
         WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE SUPERIOR'
         READ(*,*)VELMAX
      ELSE
         IF(MOD.EQ.1)THEN
            WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE INFERIOR'
            READ(*,*)VELMIN
            WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE SUPERIOR'
            READ(*,*)VELMAX
         ELSE
            IF(MOD.EQ.2)THEN
               WRITE(*,*)'INDIQUE VELOCIDAD U DE CORTE'
               READ(*,*)ULMAX
               WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE SUPERIOR'
               READ(*,*)VELMAX
            ELSE
               IF(MOD.EQ.3)THEN
                  WRITE(*,*)'INDIQUE VELOCIDAD V DE CORTE'
                  READ(*,*)VLMAX
                  WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE SUPERIOR'
                  READ(*,*)VELMAX
               ELSE
                  WRITE(*,*)'INDIQUE VELOCIDAD W DE CORTE'
                  READ(*,*)WLMAX
                  WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE SUPERIOR'
                  READ(*,*)VELMAX
      ENDIF
      ENDIF
      ENDIF
      ENDIF
C
      N=0
      DO 100 I=1,14000
      READ(1,500,END=7777)U,V,W,UVW
      IF(MOD.EQ.0)THEN
         IF(UVW.LE.VELMAX)THEN
            WRITE(2,510)U,V,W
            N=N+1
         ELSE
            CONTINUE
         ENDIF
      ELSE
```

```
            IF(MOD.EQ.1)THEN
               IF(UVW.GE.VELMIN.AND.UVW.LE.VELMAX)THEN
                  WRITE(2,510)U,V,W
                  N=N+1
               ELSE
                  CONTINUE
               ENDIF
            ELSE
               IF(MOD.EQ.2)THEN
                  IF(DABS(U).LE.ULMAX.AND.UVW.LE.VELMAX)THEN
                     WRITE(2,510)U,V,W
                     N=N+1
                  ELSE
                     CONTINUE
                  ENDIF
               ELSE
                  IF(MOD.EQ.3)THEN
                     IF(DABS(V).LE.VLMAX.AND.UVW.LE.VELMAX)THEN
                        WRITE(2,510)U,V,W
                        N=N+1
                     ELSE
                        CONTINUE
                     ENDIF
                  ELSE
                     IF(DABS(W).LE.WLMAX.AND.UVW.LE.VELMAX)THEN
                        WRITE(2,510)U,V,W
                        N=N+1
                     ELSE
                        CONTINUE
                     ENDIF
      ENDIF
      ENDIF
      ENDIF
      ENDIF
100   CONTINUE
7777  CONTINUE
500   FORMAT(1X,F9.2,2X,F9.2,2X,F9.2,2X,F7.2)
510   FORMAT(F17.10,1X,F17.10,1X,F17.10)
      WRITE(*,*)'TOTAL ESTRELLAS SELECCIONADAS',N
      STOP
      END
```

```
C234567
C *** GENERACION DE UNA MUESTRA ALEATORIA DE ESTRELLAS
      REAL*8 RESU
      INTEGER I,NESTR,NCOMP,INUM
      CHARACTER*64 FICMUE
      DIMENSION RESU(60000),FICMUE(10)
      WRITE(*,*)'¨CUANTAS MUESTRAS SE QUIEREN GENERAR?'
      READ(*,*)NUMUE
      WRITE(*,*)'¨CUANTAS ESTRELLAS POR MUESTRA?'
      READ(*,*)NESTR
      NCOMP=3*NESTR
C ***
      FICMUE(1)='MOSTRA01.DAT'
      FICMUE(2)='MOSTRA02.DAT'
      FICMUE(3)='MOSTRA03.DAT'
      FICMUE(4)='MOSTRA04.DAT'
      FICMUE(5)='MOSTRA05.DAT'
      FICMUE(6)='MOSTRA06.DAT'
      FICMUE(7)='MOSTRA07.DAT'
      FICMUE(8)='MOSTRA08.DAT'
      FICMUE(9)='MOSTRA09.DAT'
      FICMUE(10)='MOSTRA10.DAT'
C
C *** SE GENERAN TANTAS MUESTRAS COMO LAS INDICADAS
C *** SE ABREN FICHEROS DE SALIDA PARA CADA POBLACION
C
      DO 1000 N=1,NUMUE
      OPEN(N,FILE=FICMUE(N),STATUS='NEW')
C
C     INUM=N
      DO 100 I=1,NCOMP
      RESU(I)=gasdev(INUM)
 100  CONTINUE
C
      DO 101 I=1,NESTR
      WRITE(N,500)RESU(I),RESU(I+NESTR),RESU(I+2*NESTR)
 101  CONTINUE
 500  FORMAT(F13.10,1X,F13.10,1X,F13.10)
      TOT=0.D0
      DO 200 I=1,NCOMP
      TOT=RESU(I)+TOT
 200  CONTINUE
      TOT=TOT/NCOMP
      WRITE(*,*)'LA MEDIA ES',TOT
 1000 CONTINUE
      STOP
      END
```

```
C234567
C *** SE LEE LA MUESTRA SINTETICA. A CONTINUACION SE DEFORMA LA ESFERA
C *** PARA CONVERTIRLA EN ELIPSOIDE, ES DECIR SE INTRODUCE UNA SIGMA
C *** DIFERENTE DE 1. A CONTINUACION SE ESTABLECEN ROTACIONES Y FINALMEN
TE
C *** SE DESPLAZA EL CENTROIDE PARA QUE LA MEDIA NO SEA CERO.
C
      SUBROUTINE GIRMUE(FICMUE,FICGIR,SIGMA,VALMED,ROT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*64 FICMUE,FICGIR
      DIMENSION UVW(10000,3),UVWROT(10000,3),ROTA(3,3)
      DIMENSION SIGMA(3),VALMED(3),ROT(2)
C *** APERTURA DE LOS FICHEROS DE ENTRADA SALIDA
      WRITE(*,*)'ACCESO A LA RUTINA GIRMUE'
      PI=3.14159265
      OPEN(1,FILE=FICMUE,STATUS='OLD')
      OPEN(2,FILE=FICGIR,STATUS='NEW')
C
C *** SE LEE LA MUESTRA SINTETICA
C *** UVW ES LA VARIABLE QUE CONTIENE LAS VELOCIDADES U,V y W. N ES EL
C *** NUMERO DE POBLACION 1,2,3 ES LA COORDENADA U,V,W e I ES LA
C *** ESTRELLA
C
      NEST=0
      DO 100 I=1,10000
      READ(1,500,END=7777)UVW(I,1),UVW(I,2),UVW(I,3)
      NEST=NEST+1
 100  CONTINUE
 7777 WRITE(*,*)NEST
C
C *** SE DEFORMA LA ESFERA INTRODUCIENDO UNA SIGMA DIFERENTE DE 1
C
      DO 150 I=1,NEST
      DO 151 J=1,3
      UVW(I,J)=UVW(I,J)*SIGMA(J)
 151  CONTINUE
 150  CONTINUE
C
C *** SE ESTABLECEN ROTACIONES EN LAS COMPONENTES
C     (VER PAG. 19 ORUS)
C
C *** SE CONTRUYE LA MATRIZ DE ROTACION EN LA  DIRECCION ZETA
      ROT(1)=ROT(1)*PI/180.D0
      ROTA(1,1)=DCOS(ROT(1))
      ROTA(1,2)=-DSIN(ROT(1))
      ROTA(1,3)=0.D0
      ROTA(2,1)=DSIN(ROT(1))
      ROTA(2,2)=DCOS(ROT(1))
      ROTA(2,3)=0.D0
      ROTA(3,1)=0.D0
      ROTA(3,2)=0.D0
      ROTA(3,3)=1.D0
C *** SE GIRA EL ANGULO INDICADO
      DO 200 I=1,NEST
      DO 201 J=1,3
      UVWROT(I,J)=0.D0
```

```
      DO 202 K=1,3
      UVWROT(I,J)=UVWROT(I,J)+ROTA(J,K)*UVW(I,K)
 202  CONTINUE
 201  CONTINUE
 200  CONTINUE
C *** SE CONTRUYE LA MATRIZ DE ROTACION RESPECTO AL PLANO ECUATORIAL
      ROT(2)=ROT(2)*PI/180.D0
      ROTA(1,1)=DCOS(ROT(2))
      ROTA(1,2)=0.D0
      ROTA(1,3)=-DSIN(ROT(2))
      ROTA(2,1)=0.D0
      ROTA(2,2)=1.D0
      ROTA(2,3)=0.D0
      ROTA(3,1)=DSIN(ROT(2))
      ROTA(3,2)=0.D0
      ROTA(3,3)=DCOS(ROT(2))
C *** SE GIRA EL ANGULO INDICADO
      DO 300 I=1,NEST
      DO 301 J=1,3
      UVW(I,J)=0.D0
      DO 302 K=1,3
      UVW(I,J)=UVW(I,J)+ROTA(J,K)*UVWROT(I,K)
 302  CONTINUE
 301  CONTINUE
 300  CONTINUE
C
C *** SE DESPLAZA EL ELIPSOIDE: LA MEDIA DIFERENTE DE CERO
C
      DO 400 I=1,NEST
      DO 401 J=1,3
      UVW(I,J)=UVW(I,J)+VALMED(J)
 401  CONTINUE
 400  CONTINUE
C
C *** SE ESCRIBEN LOS RESULTADOS
C
      DO 450 I=1,NEST
      WRITE(2,510)UVW(I,1),UVW(I,2),UVW(I,3)
 450  CONTINUE
C
      CLOSE(1)
      CLOSE(2)
C
 500  FORMAT(F13.10,1X,F13.10,1X,F13.10)
 510  FORMAT(F17.10,1X,F17.10,1X,F17.10)
C
      RETURN
      END
```

```
C234567
C PROGRAMA PARA SELECCIONAR ESTRELLAS DEL CATALOGO DE GLIESE-JAHREISS
C EN FUNCION DEL MODULO DE LA VELOCIDAD
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      OPEN(1,FILE='CNS.DAT',STATUS='OLD')
      OPEN(2,FILE='GLIJAH.DAT',STATUS='NEW')
C
      WRITE(*,*)'INDIQUE MODALIDAD DE CORTE'
      WRITE(*,*)' 0=UVW MAX, 1=UVW MIN Y MAX, 2=U Y UVW MAX'
      WRITE(*,*)' 3=V Y UVW MAX, 4=W Y UVW MAX'
      READ(*,*)MOD
C
      IF(MOD.EQ.0)THEN
         WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE SUPERIOR'
         READ(*,*)VELMAX
      ELSE
         IF(MOD.EQ.1)THEN
            WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE INFERIOR'
            READ(*,*)VELMIN
            WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE SUPERIOR'
            READ(*,*)VELMAX
         ELSE
            IF(MOD.EQ.2)THEN
               WRITE(*,*)'INDIQUE VELOCIDAD U DE CORTE'
               READ(*,*)ULMAX
               WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE SUPERIOR'
               READ(*,*)VELMAX
            ELSE
               IF(MOD.EQ.3)THEN
                  WRITE(*,*)'INDIQUE VELOCIDAD V DE CORTE'
                  READ(*,*)VLMAX
                  WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE SUPERIOR'
                  READ(*,*)VELMAX
               ELSE
                  WRITE(*,*)'INDIQUE VELOCIDAD W DE CORTE'
                  READ(*,*)WLMAX
                  WRITE(*,*)'INDIQUE MODULO VELOCIDAD DE CORTE SUPERIOR'
                  READ(*,*)VELMAX
      ENDIF
      ENDIF
      ENDIF
      ENDIF
C
      N=0
      DO 100 I=1,10000
      READ(1,500,END=7777)U,V,W,UVW
      IF(MOD.EQ.0)THEN
         IF(UVW.LE.VELMAX)THEN
            WRITE(2,510)U,V,W
            N=N+1
         ELSE
            CONTINUE
         ENDIF
      ELSE
         IF(MOD.EQ.1)THEN
```

```
            IF(UVW.GE.VELMIN.AND.UVW.LE.VELMAX)THEN
               WRITE(2,510)U,V,W
               N=N+1
            ELSE
               CONTINUE
            ENDIF
         ELSE
            IF(MOD.EQ.2)THEN
               IF(DABS(U).LE.ULMAX.AND.UVW.LE.VELMAX)THEN
                  WRITE(2,510)U,V,W
                  N=N+1
               ELSE
                  CONTINUE
               ENDIF
            ELSE
               IF(MOD.EQ.3)THEN
                  IF(DABS(V).LE.VLMAX.AND.UVW.LE.VELMAX)THEN
                     WRITE(2,510)U,V,W
                     N=N+1
                  ELSE
                     CONTINUE
                  ENDIF
               ELSE
                  IF(DABS(W).LE.WLMAX.AND.UVW.LE.VELMAX)THEN
                     WRITE(2,510)U,V,W
                     N=N+1
                  ELSE
                     CONTINUE
                  ENDIF
      ENDIF
      ENDIF
      ENDIF
      ENDIF
100   CONTINUE
7777  CONTINUE
500   FORMAT(1X,F9.2,2X,F9.2,2X,F9.2,2X,F7.2)
510   FORMAT(F17.10,1X,F17.10,1X,F17.10)
      WRITE(*,*)'TOTAL ESTRELLAS SELECCIONADAS',N
      STOP
      END
```

```
C234567
C
C ***** APERTURA Y LECTURA DEL FICHERO DE DATOS
C
      SUBROUTINE LECTU(NP,M2,V,F,MA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*8 M2,MAYOR
      DIMENSION M2(5,3,3),V(5,3),F(5)
C ***
      WRITE(*,*)'ACCESO A RUTINA LECTU'
C
C NP=NUMERO DE POBLACIONES
C M2=MOMENTOS DE ORDEN 2 DE CADA POBLACION PARCIAL
C
C *** LECTURA DEL NUMERO DE POBLACIONES
      WRITE(*,*)'***** INTRODUZCA EL NUMERO DE POBLACIONES *****'
      READ(*,*)NP
C ***
C *** APERTURA DEL FICHERO ***
      OPEN(1,FILE='DATOS.DAT')
C *** LECTURA DE LOS DATOS DE LAS POBLACIONES PARCIALES
C V=VELOCIDAD DE CADA SUBCENTROIDE
C F=FRACCION DE POBLACION
C MA=INDICE DE LA VELOCIDAD MAYOR. SE TOMA COMO REFERENCIA
C
      DO 200 N=1,NP
      READ(1,500)M2(N,1,1),M2(N,1,2),M2(N,1,3),M2(N,2,2),M2(N,2,3),M2(N,
     #3,3),V(N,1),V(N,2),V(N,3),F(N)
 200  CONTINUE
 500  FORMAT(1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,
     #F7.1,1X,F7.1,1X,F4.2)
C
      DO 100 N=1,NP
      DO 110 I=1,3
      DO 120 J=I,3
      M2(N,J,I)=M2(N,I,J)
 120  CONTINUE
 110  CONTINUE
 100  CONTINUE
C
C *** LOCALIZACION DE LA POBLACION DE REFERENCIA
C
      MAYOR=0.D0
      DO 700 N=1,NP
      DO 710 I=1,3
      IF(V(N,I).GE.MAYOR) THEN
                          MAYOR=V(N,I)
                          MA=I
      ENDIF
 710  CONTINUE
 700  CONTINUE
      WRITE(*,*)'EL MAYOR ES',MA
      RETURN
      END
```

```
C234567
C
C *** PROGRAMA PARA CAMBIAR EL FORMATO DE ESCRITURA DE LOS MOMENTOS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*64 FICENT,FICSAL,FICERR
      REAL*8 MU2,MU3,MU4
      DIMENSION MU2(6),MU3(10),MU4(15),ERMU2(6),ERMU3(10),ERMU4(15)
C *** SE DEFINEN TODOS LOS FICHEROS A UTILIZAR
      WRITE(*,*)'NOMBRE DEL FICHERO DE MOMENTOS A REESCRIBIR'
C     READ(*,*)FICENT
      FICENT='MOMMEZ.DAT'
      FICSAL='SINTE.DAT'
      FICERR='SINER.DAT'
      OPEN(1,FILE=FICENT,STATUS='OLD')
      OPEN(2,FILE=FICSAL,STATUS='NEW')
      OPEN(3,FILE=FICERR,STATUS='NEW')
C
      READ(1,800)
      READ(1,510)UMED,ERUMED
      READ(1,510)VMED,ERVMED
      READ(1,510)WMED,ERWMED
      READ(1,800)
      DO 200 I=1,6
      READ(1,500)MU2(I),ERMU2(I)
 200  CONTINUE
      READ(1,800)
      DO 210 I=1,10
      READ(1,500)MU3(I),ERMU3(I)
 210  CONTINUE
      READ(1,800)
      DO 220 I=1,15
      READ(1,500)MU4(I),ERMU4(I)
 220  CONTINUE
C
 500  FORMAT(10X,F12.2,5X,F12.2,34X)
 510  FORMAT(10X,34X,F12.2,5X,F12.2)
C
C *** ESCRITURA DE LOS MOMENTOS TOTALES Y LA VELOCIDAD
C *** SE PERMUTAN LOS INDICES 2 Y 3, bzw V Y W
C
      WRITE(2,700)UMED,WMED,VMED
C
      WRITE(2,710)MU2(1),MU2(4),MU2(2),MU2(6),MU2(5),MU2(3)
C
      WRITE(2,720)MU3(1),MU3(5),MU3(2),MU3(8),MU3(6)
     #,MU3(3),MU3(10),MU3(9),MU3(7),MU3(4)
C
      WRITE(2,730)MU4(1),MU4(6),MU4(2),MU4(10),MU4(7),MU4(3),MU4(13),
     #MU4(11),MU4(8),MU4(4),MU4(15),MU4(14),MU4(12),MU4(9),MU4(5)
C
C *** ESCRITURA DE LOS ERRORES DE LOS MOMENTOS Y VELOCIDAD
C
      WRITE(3,700)ERUMED,ERWMED,ERVMED
C
      WRITE(3,710)ERMU2(1),ERMU2(4),ERMU2(2),ERMU2(6),ERMU2(5),ERMU2(3)
```

```
C
      WRITE(3,720)ERMU3(1),ERMU3(5),ERMU3(2),ERMU3(8),ERMU3(6)
     #,ERMU3(3),ERMU3(10),ERMU3(9),ERMU3(7),ERMU3(4)
C
      WRITE(3,730)ERMU4(1),ERMU4(6),ERMU4(2),ERMU4(10),ERMU4(7),ERMU4(3)
     #,ERMU4(13),ERMU4(11),ERMU4(8),ERMU4(4),ERMU4(15),ERMU4(14),
     #ERMU4(12),ERMU4(9),ERMU4(5)
C *** FORMATOS
 800  FORMAT(2X)
 700  FORMAT(1X,F12.2,1X,F12.2,1X,F12.2)
 710  FORMAT(1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2)
 720  FORMAT(1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F1
     #2.2,1X,F12.2,1X,F12.2,1X,F12.2)
 730  FORMAT(1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F1
     #2.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2
     #,1X,F12.2)
C
      STOP
      END
```

```
C234567
C
C *** CALCULO DE LOS MOMENTOS DE SEGUNDO TERCER Y CUARTO ORDEN DE
C *** UNA MUESTRA DE ESTRELLAS
C *** CALCULA LOS MOMENTOS CENTRADOS Y NO CENTRADOS DE LAS VELOCIDADES
C *** RESIDUALES OBSERVADAS
C
C *** FICHERO DE ENTRADA HIPSEL.DAT = SALIDA DE G14000.EXE
C *** FICHERO DE SALIDA  MOMMEZ.DAT = SALIDA DE MOMMEZ.EXE
C
C ********* SE HA REDIMENSIONADO A 14.000 ESTRELLAS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*64 FICGIR,FICMOM
      DIMENSION U(14000),V(14000),W(14000),UMX(2),VMX(2),WMX(2),
     #C(9,9,9),VC(5,5,5),S(9,9,9),VS(5,5,5)
      WRITE(*,*)'COMIENZA EL PROGRAMA M14000'
C *** LECTURA DEL FICHERO DE DATOS
      OPEN(1,FILE='HIPSEL.DAT',STATUS='OLD')
      OPEN(2,FILE='MOMMEZ.DAT',STATUS='NEW')
      NAA=0
      DO 100 I=1,14000
      READ(1,500,END=7777)U(I),V(I),W(I)
      NAA=NAA+1
 100  CONTINUE
 7777 CONTINUE
 500  FORMAT(F17.10,1X,F17.10,1X,F17.10)
      A=FLOAT(NAA)
      WRITE(*,*)'TOTAL ESTRELLAS',NAA
C *** CALCULO DE LAS MEDIAS DE LAS VELOCIDADES
      UMX(1)=0.D0
      VMX(1)=0.D0
      WMX(1)=0.D0
      KY=1
      DO 1030 II=1,NAA
C *** SE CONSIDERAN LAS VELOCIDADES U=UDD, V=VDD, W=WDD (DESCONTADA LA
C *** ROTACION GALACTICA)
      UMX(KY)=UMX(KY)+U(II)
      VMX(KY)=VMX(KY)+V(II)
      WMX(KY)=WMX(KY)+W(II)
 1030 CONTINUE
      UMX(KY)=UMX(KY)/A
      VMX(KY)=VMX(KY)/A
      WMX(KY)=WMX(KY)/A
C ***
      DO 1002 I=1,9
      DO 1003 J=1,9
      DO 1004 KA=1,9
      JJJJ=I+J+KA
      IF(JJJJ.GT.11) GO TO 1004
      S(I,J,KA)=0.D0
      C(I,J,KA)=0.D0
      DO 1005 II=1,NAA
      C(I,J,KA)=C(I,J,KA)+(((U(II)-UMX(KY))**(I-1))*((V(II)
     #-VMX(KY))**(J-1))*((W(II)-WMX(KY))**(KA-1)))
      S(I,J,KA)=S(I,J,KA)+((U(II))**(I-1))*(V(II))**(J-1)*
```

```
     #((W(II))**(KA-1))
 1005 CONTINUE
      C(I,J,KA)=C(I,J,KA)/A
      S(I,J,KA)=S(I,J,KA)/A
 1004 CONTINUE
 1003 CONTINUE
 1002 CONTINUE
C ***
      WRITE(*,*)'SE CALCULAN LOS MOMENTOS'
      DO 1006 I=1,5
      DO 1007 J=1,5
      DO 1008 KA=1,5
      JJJJ=I+J+KA
      IF(JJJJ.GT.7) GO TO 1008
      VS(I,J,KA)=DSQRT(DABS((S(2*I-1,2*J-1,2*KA-1)-S(I,J,KA)*S(I,J,KA))
     #/A))
 1008 CONTINUE
 1007 CONTINUE
 1006 CONTINUE
C ***
      WRITE(*,*)'SE CALCULAN LOS ERRORES DE LOS MOMENTOS'
      V1=DSQRT(DABS((C(3,1,1))/A))
      V2=DSQRT(DABS((C(1,3,1))/A))
      V3=DSQRT(DABS((C(1,1,3))/A))
C
      V11=DSQRT(DABS((C(5,1,1)-C(3,1,1)*C(3,1,1))/A))
      V12=DSQRT(DABS((C(3,3,1)-C(2,2,1)*C(2,2,1))/A))
      V13=DSQRT(DABS((C(3,1,3)-C(2,1,2)*C(2,1,2))/A))
      V22=DSQRT(DABS((C(1,5,1)-C(1,3,1)*C(1,3,1))/A))
      V23=DSQRT(DABS((C(1,3,3)-C(1,2,2)*C(1,2,2))/A))
      V33=DSQRT(DABS((C(1,1,5)-C(1,1,3)*C(1,1,3))/A))
C
      V111=DSQRT(DABS((C(7,1,1)-6*(C(5,1,1)*C(3,1,1))-C(4,1,1)*
     #C(4,1,1)+9*(C(3,1,1)**3))/A))
      V112=DSQRT(DABS((C(5,3,1)-4*(C(4,2,1)*C(2,2,1))-2*(C(3,3,1)*C(3
     #,1,1))-C(3,2,1)*C(3,2,1)+2*(C(3,1,1)*C(2,2,1)*C(2,2,1))+C(1,3,
     #1)*C(3,1,1)*C(3,1,1)+6*(C(3,1,1)*C(2,2,1)*C(2,2,1)))/A))
      V113=DSQRT(DABS((C(5,1,3)-4*(C(4,1,2)*C(2,1,2))-2*(C(3,1,3)*C(3
     #,1,1))-C(3,1,2)*C(3,1,2)+2*(C(3,1,1)*C(2,1,2)*C(2,1,2)+C(1,1,
     #2)*C(3,1,1)*C(3,1,1))+6*(C(3,1,1)*C(2,1,2)*C(2,1,2)))/A))
      V122=DSQRT(DABS((C(3,5,1)-2*(C(3,3,1)*C(1,3,1))-4*(C(2,4,1)*C(2
     #,2,1))-C(2,3,1)*C(2,3,1)+C(3,1,1)*C(1,3,1)*C(1,3,1)+2*(C(1,3,
     #1)*C(2,2,1)*C(2,2,1))+6*(C(2,2,1)*C(2,2,1)*C(1,3,1)))/A))
      V123=DSQRT(DABS((C(3,3,3)-2*(C(3,2,2)*C(1,2,2)+C(2,3,2)*C(2,1,2)+
     #C(2,2,3)*C(2,2,1))-C(2,2,2)*C(2,2,2)+C(3,1,1)*C(1,2,2)*C(1,2,2)
     #+C(1,3,1)*C(2,1,2)*C(2,1,2)+C(1,1,3)*C(2,2,1)*C(2,2,1)+6*
     #(C(2,2,1)*C(2,1,2)*C(1,2,2)))/A))
      V133=DSQRT(DABS((C(3,1,5)-2*C(3,1,3)*C(1,1,3)-4*C(2,1,4)*C(2
     #,1,2)-C(2,1,3)*C(2,1,3)+C(3,1,1)*C(1,1,3)*C(1,1,3)+2*C(1,1,
     #3)*C(2,1,2)*C(2,1,2)+6*C(2,1,2)*C(2,1,2)*C(1,1,3))/A))
      V222=DSQRT(DABS((C(1,7,1)-6*C(1,5,1)*C(1,3,1)-C(1,4,1)*
     #C(1,4,1)+9*(C(1,3,1)**3))/A))
      V223=DSQRT(DABS((C(1,5,3)-4*C(1,4,2)*C(1,2,2)-2*C(1,3,3)*C(1
     #,3,1)-C(1,3,2)*C(1,3,2)+8*C(1,3,1)*C(1,2,2)*C(1,2,2)+C(1,1,
     #4)*C(1,3,1)*C(1,3,1))/A))
      V233=DSQRT(DABS((C(1,3,5)-2*C(1,3,3)*C(1,1,3)-4*C(1,2,4)*C(1
```

```
     #,2,2)-C(1,2,3)*C(1,2,3)+C(1,3,1)*C(1,1,3)*C(1,1,3)+8*C(1,1,
     #3)*C(1,2,2)*C(1,2,2))/A))
      V333=DSQRT(DABS((C(1,1,7)-6*C(1,1,5)*C(1,1,3)-C(1,1,4)*
     #C(1,1,4)+9*(C(1,1,3)**3))/A))
C
      V1111=DSQRT(DABS((C(9,1,1)-C(5,1,1)*C(5,1,1)-8*C(6,1,1)*C(4,1,1)+
     #16*C(3,1,1)*C(4,1,1)*C(4,1,1))/A))
      V2222=DSQRT(DABS((C(1,9,1)-C(1,5,1)*C(1,5,1)-8*C(1,6,1)*C(1,4,1)+
     #16*C(1,3,1)*C(1,4,1)*C(1,4,1))/A))
      V3333=DSQRT(DABS((C(1,1,9)-C(1,1,5)*C(1,1,5)-8*C(1,1,6)*C(1,1,4)+
     #16*C(1,1,3)*C(1,1,4)*C(1,1,4))/A))
      V1112=DSQRT(DABS((C(7,3,1)-C(4,2,1)*C(4,2,1)-6*C(5,2,1)*C(3,2,1)-
     #2*C(4,3,1)*C(4,1,1)+6*C(3,1,1)*C(3,2,1)*C(3,2,1)+C(1,3,1)*
     #C(4,1,1)*C(4,1,1)+C(3,1,1)*C(4,1,1)*C(3,2,1)+2*C(3,1,1)*C(4,1,1)
     #*C(2,3,1)+6*C(2,2,1)*C(3,2,1)*C(4,1,1))/A))
      V1113=DSQRT(DABS((C(7,1,3)-C(4,1,2)*C(4,1,2)-6*C(5,1,2)*C(3,1,2)-
     #2*C(4,1,3)*C(4,1,1)+6*C(3,1,1)*C(3,1,2)*C(3,1,2)+C(1,1,3)*
     #C(4,1,1)*C(4,1,1)+C(3,1,1)*C(4,1,1)*C(3,1,2)+2*C(3,1,1)*C(4,1,1)
     #*C(2,1,3)+6*C(2,1,2)*C(3,1,2)*C(4,1,1))/A))
      V1122=DSQRT(DABS((C(5,5,1)-C(3,3,1)*C(3,3,1)-4*C(4,3,1)*C(2,3,1)-
     #4*C(3,4,1)*C(3,2,1)+2*C(3,1,1)*C(2,3,1)*C(2,3,1)+2*C(1,3,1)*
     #C(3,2,1)*C(3,2,1)+2*C(3,1,1)*C(3,2,1)*C(1,4,1)+8*C(2,2,1)*C(2,3,1)
     #*C(3,2,1)+2*C(1,3,1)*C(2,3,1)*C(4,1,1))/A))
      XX=C(5,3,3)-C(3,2,2)*C(3,2,2)-4*C(4,2,2)*C(2,2,2)-2*(
     #C(3,3,2)*C(3,1,2)+C(3,2,3)*C(3,2,1))+2*C(3,1,1)*C(2,2,2)*
     #C(2,2,2)+C(1,3,1)*C(3,1,2)*C(3,1,2)+C(1,1,3)*C(3,2,1)*C(3,2,1)
      V1123=DSQRT(DABS((C(3,1,1)*C(3,2,1)*C(1,3,2)+C(3,1,1)*C(3,1,2)*C(
     #1,3,2)+2*(C(2,2,1)*C(2,2,2)*C(3,1,2)+C(2,2,1)*C(3,2,1)*C(2,1,3)+
     #C(2,1,2)*C(3,1,2)*C(2,3,1)+C(2,1,2)*C(2,2,2)*C(3,2,1)+C(1,2,2)*
     #C(2,2,2)*C(4,1,1))+XX)/A))
C
      V1133=DSQRT(DABS((C(5,1,5)-C(3,1,3)*C(3,1,3)-4*(C(4,1,3)*C(2,1,3)
     #+C(3,1,4)*C(3,1,2))+2*(C(3,1,1)*C(2,1,3)*C(2,1,3)+C(1,1,3)*
     #C(3,1,2)*C(3,1,2)+C(3,1,1)*C(3,1,2)*C(1,1,4))+8*C(2,1,2)*C(2,1,3)
     #*C(3,1,2)+2*C(1,1,3)*C(2,1,3)*C(4,1,1))/A))
C
      V1222=DSQRT(DABS((C(3,7,1)-C(2,4,1)*C(2,4,1)-2*C(3,4,1)*C(1,4,1)-
     #6*(C(2,5,1)*C(2,3,1)-C(1,3,1)*C(2,3,1)*C(2,3,1)-C(2,2,1)*
     #C(2,3,1)*C(1,4,1))+C(3,1,1)*C(1,4,1)*C(1,4,1)+3*C(1,3,1)*
     #C(1,4,1)*C(3,2,1))/A))
      XX=C(3,5,3)-C(2,3,2)*C(2,3,2)-2*C(3,3,2)*C(1,3,2)-4*C(
     #2,4,2)*C(2,2,2)-2*C(2,3,3)*C(2,3,1)+C(3,1,1)*C(1,3,2)*C(1,3,2)+
     #2*C(1,3,1)*C(2,2,2)*C(2,2,2)+C(1,1,3)*C(2,3,1)*C(2,3,1)+
     #C(2,2,1)*C(2,3,1)*C(1,3,2)
      V1223=DSQRT(DABS((XX+2*C(2,2,1)*C(2,2,2)*C(1,3,2)+C(2,2,1)*C(2,3,
     #1)*C(1,2,3)+2*C(2,1,2)*C(2,2,2)*C(1,4,1)+C(1,3,1)*C(2,3,1)*
     #C(2,1,2)+C(1,3,1)*C(1,3,2)*C(3,1,2)+2*C(1,2,2)*C(2,2,2)*
     #C(2,3,1)+2*C(1,2,2)*C(1,3,2)*C(3,2,1))/A))
      XX=C(3,3,5)-C(2,2,3)*C(2,2,3)-2*C(3,2,3)*C(1,2,3)-2*
     #C(2,3,3)*C(2,1,3)-4*C(2,2,4)*C(2,2,2)+C(3,1,1)*C(1,2,3)*
     #C(1,2,3)+C(1,3,1)*C(2,1,3)*C(2,1,3)+2*C(1,1,3)*C(2,2,2)*
     #C(2,2,2)+2*C(2,2,1)*C(2,2,2)*C(1,1,4)
      V1233=DSQRT(DABS((XX+2*C(2,1,2)*C(2,1,3)*C(1,3,2)+2*(C(2,1,2)*
     #C(2,2,2)*C(1,2,3)+C(1,2,2)*C(2,2,2)*C(2,1,3)+C(1,2,2)*C(1,2,3)*
     #C(3,1,2))+C(1,1,3)*C(2,1,3)*C(2,3,1)+C(1,1,3)*C(1,2,3)*
     #C(3,1,2))/A))
```

```
      V1333=DSQRT(DABS((C(3,1,7)-C(2,1,4)*C(2,1,4)-2*C(3,1,4)*C(1,1,4)-
     #6*C(2,1,5)*C(2,1,3)+C(3,1,1)*C(1,1,4)*C(1,1,4)+6*(C(1,1,3)*
     #C(2,1,3)*C(2,1,3)+C(2,1,2)*C(2,1,3)*C(1,1,4))+3*C(1,1,3)*
     #C(1,1,4)*C(3,1,2))/A))
      V2223=DSQRT(DABS((C(1,7,3)-C(1,4,2)*C(1,4,2)-6*C(1,5,2)*C(1,3,2)-
     #2*C(1,4,3)*C(1,4,1)+6*C(1,3,1)*C(1,3,2)*C(1,3,2)+C(1,1,3)*
     #C(1,4,1)*C(1,4,1)+C(1,3,1)*C(1,4,1)*C(1,3,2)+2*C(1,3,1)*
     #C(1,4,1)*C(1,2,3)+6*C(1,2,2)*C(1,3,2)*C(1,4,1))/A))
      V2233=DSQRT(DABS((C(1,5,5)-C(1,3,3)*C(1,3,3)-4*(C(1,4,3)*C(1,2,3)-
     #C(1,3,4)*C(1,3,2))+2*(C(1,3,1)*C(1,2,1)*C(1,2,3)+C(1,1,3)*
     #C(1,3,2)*C(1,3,2)+C(1,3,1)*C(1,3,2)*C(1,1,4))+8*C(1,2,2)*
     #C(1,2,3)*C(1,3,2)+2*C(1,1,3)*C(1,2,3)*C(1,4,1))/A))
      V2333=DSQRT(DABS((C(1,3,7)-C(1,2,4)*C(1,2,4)-2*C(1,3,4)*C(1,1,4)-
     #6*C(1,2,5)*C(1,2,3)+C(1,3,1)*C(1,1,4)*C(1,1,4)+6*(C(1,1,3)*
     #C(1,2,3)*C(1,2,3)+6*C(1,2,2)*C(1,2,3)*C(1,1,4))+3*C(1,1,3)*
     #C(1,1,4)*C(1,3,2))/A))
C
      VC(2,1,1)=V1
      VC(1,2,1)=V2
      VC(1,1,2)=V3
      VC(3,1,1)=V11
      VC(2,2,1)=V12
      VC(2,1,2)=V13
      VC(1,3,1)=V22
      VC(1,2,2)=V23
      VC(1,1,3)=V33
      VC(4,1,1)=V111
      VC(3,2,1)=V112
      VC(3,1,2)=V113
      VC(2,3,1)=V122
      VC(2,2,2)=V123
      VC(2,1,3)=V133
      VC(1,4,1)=V222
      VC(1,3,2)=V223
      VC(1,2,3)=V233
      VC(1,1,4)=V333
      VC(5,1,1)=V1111
      VC(4,2,1)=V1112
      VC(4,1,2)=V1113
      VC(3,3,1)=V1122
      VC(3,2,2)=V1123
      VC(3,1,3)=V1133
      VC(2,4,1)=V1222
      VC(2,3,2)=V1223
      VC(2,2,3)=V1233
      VC(2,1,4)=V1333
      VC(1,5,1)=V2222
      VC(1,4,2)=V2223
      VC(1,3,3)=V2233
      VC(1,2,4)=V2333
      VC(1,1,5)=V3333
C
C *** SE PROCEDE A LA ESCRITURA DE LOS RESULTADOS: MOEMNTOS Y SUS ERRORE
S
C *** SE CALCA DE LA RUTINA DE XESCA PARA EVITAR COMPLICACIONES
C
```

```
      WRITE(*,*)'SE ESCRIBEN LOS MOMENTOS CENTRADOS Y SUS ERRORES'
      WRITE(*,*)'ASI COMO LOS MOMENTOS NO CENTRADOS Y SUS ERRORES'
 2013 FORMAT(2X,'INDICE',5X,'MOMENTO CENTRADO',5X,'ERROR',5X,'MOMENTO NO
     #CENTRADO',5X,'ERROR')
 2014 FORMAT(2X,3I1,5X,F12.2,5X,F12.2,5X,F12.2,5X,F12.2)
      DO 2039 LL=4,7
      LLL=LL-3
      WRITE(2,2015)LLL
      DO 2040 KA=1,5
      DO 2041 J=1,5
      DO 2042 I=1,5
      IJK=I+J+KA
      IF(IJK.NE.LL)GOTO 2042
      II=I-1
      JJ=J-1
      KK=KA-1
      WRITE(2,2014)II,JJ,KK,C(I,J,KA),VC(I,J,KA),S(I,J,KA),VS(I,J,KA)
 2042 CONTINUE
 2041 CONTINUE
 2040 CONTINUE
 2039 CONTINUE
 2015 FORMAT(2X,'ORDEN',I1)
C
C *** SE CALCULAN E IMPRIMEN LOS CUMULANTES
C
      CALL CALCUM(C,VC)
C
      CLOSE(1)
      CLOSE(2)
C
      STOP
      END
```

```
C234567
C
C *** PROGRAMA DE MEZCLA DE MUESTRAS SINTETICAS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*64 FICMEZ,FICGIR,MOMMEZ
      DIMENSION FICGIR(10),PORCEN(10),U(10000),V(10000),W(10000)
C *** SE DEFINEN TODOS LOS FICHEROS A UTILIZAR
C *** FICHEROS QUE CONTIENEN LAS MUESTRAS
C *** MUESTRAS GIRADAS
      FICGIR(1)='MOSGIR01.DAT'
      FICGIR(2)='MOSGIR02.DAT'
      FICGIR(3)='MOSGIR03.DAT'
      FICGIR(4)='MOSGIR04.DAT'
      FICGIR(5)='MOSGIR05.DAT'
      FICGIR(6)='MOSGIR06.DAT'
      FICGIR(7)='MOSGIR07.DAT'
      FICGIR(8)='MOSGIR08.DAT'
      FICGIR(9)='MOSGIR09.DAT'
      FICGIR(10)='MOSGIR10.DAT'
C
      FICMEZ='FICMEZ.DAT'
      MOMMEZ='MOMMEZ.DAT'
      OPEN(11,FILE=FICMEZ,STATUS='NEW')
C
C *** SE SUPONEN TODAS LAS MUESTRAS CON EL MISMO NUMERO DE ESTRELLAS
C *** SE PREGUNTA CUANTAS Y CUALES MUESTRAS SE VAN A MEZCLAR
C
      WRITE(*,*)'NUMERO DE MUESTRAS A MEZCLAR'
      READ(*,*)NUMUE
      WRITE(*,*)'NUMERO DE ESTRELLAS QUE CONTIENEN LAS MUESTRAS'
      READ(*,*)NEST
 9999 DO 100 I=1,NUMUE
      WRITE(*,*)'INDIQUE ORDINAL DE MUESTRA A MEZCLAR'
      READ(*,*)IMUE
      OPEN(I,FILE=FICGIR(IMUE),STATUS='OLD')
      WRITE(*,*)'PROPORCION DE LA MUESTRA ',FICGIR(IMUE),'(ENTRAR EN %)'
      READ(*,*)PORCEN(I)
 100  CONTINUE
      TOTAL=0.D0
      DO 110 I=1,NUMUE
      TOTAL=TOTAL+PORCEN(I)
 110  CONTINUE
      IF(TOTAL.NE.100)THEN
         WRITE(*,*)'PROPORCIONES INCORRECTAS'
         DO 800 IX=1,NUMUE
         CLOSE (IX)
 800     CONTINUE
         GO TO 9999
      ELSE
         CONTINUE
      ENDIF
C
C *** SE LEEN LAS ESTRELLAS CORRESPONDIENTES A LOS FICHEROS INDICADOS Y
C *** SE ESCRIBEN EN UN NUEVO FICHERO
C
```

```
      NAA=0
      DO 200 I=1,NUMUE
      NPART=PORCEN(I)*NEST/100.D0
      WRITE(*,*)'ESTRELLAS DE CADA MUESTRA',NPART
      DO 220 J=1,NPART
      READ(I,500,END=7777)U(J),V(J),W(J)
      GO TO 8888
 7777 WRITE(*,*)'NUMERO DE ESTRELLAS INCORRECTO'
      GO TO 9999
 8888 CONTINUE
      NAA=NAA+1
      WRITE(11,500)U(J),V(J),W(J)
 220  CONTINUE
 200  CONTINUE
 500  FORMAT(F17.10,1X,F17.10,1X,F17.10)
      WRITE(*,*)'ESTRELLAS GUARDADAS',NAA
C *** SE CALCULAN LOS MOMENTOS DE LA NUEVA MUESTRA MEZCLADA
      CLOSE(11)
      CALL CALCMO(FICMEZ,MOMMEZ)
      STOP
      END
```

```
C234567
C
C ***** RESOLUCION DEL SISTEMA POR MINIMOS CUADRADOS
C
      SUBROUTINE MINCUA(A,B,NDIM,V)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(14,2),AT(2,14),ATA(2,2),ATAI(2,2),ATAIAT(2,14),V(2)
      DIMENSION B(14)
C ***
      WRITE(*,*)'ACCESO A RUTINA MINCUA'
C ***
      DO 100 I=1,2
      DO 110 J=1,NDIM
      AT(I,J)=A(J,I)
 110  CONTINUE
 100  CONTINUE
C ***
      DO 200 I=1,2
      DO 210 J=1,2
      ATA(I,J)=0.D0
      DO 220 K=1,NDIM
C     AUX=AT(I,K)*A(K,J)
      ATA(I,J)=ATA(I,J)+AT(I,K)*A(K,J)
 220  CONTINUE
 210  CONTINUE
 200  CONTINUE
C ***
      DET=ATA(1,1)*ATA(2,2)-ATA(1,2)*ATA(2,1)
      ATAI(1,1) = ATA(2,2)/DET
      ATAI(1,2) = -ATA(1,2)/DET
      ATAI(2,1) = -ATA(2,1)/DET
      ATAI(2,2) = ATA(1,1)/DET
C ***
      DO 300 I=1,2
      DO 310 J=1,NDIM
      ATAIAT(I,J)=0.D0
      DO 320 K=1,2
      ATAIAT(I,J)=ATAIAT(I,J)+ATAI(I,K)*AT(K,J)
 320  CONTINUE
 310  CONTINUE
 300  CONTINUE
C ***
      DO 400 I=1,2
      V(I)=0.D0
      DO 410 J=1,NDIM
      V(I)=V(I)+ATAIAT(I,J)*B(J)
 410  CONTINUE
 400  CONTINUE
C ***
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LOS MOMENTOS PARCIALES SUPUESTAS DOS POBLACIONES
C ***** SE DAN TAMBIEN LOS ERRORES. ES COMO MOMPAR.FOR PERO CON ERRORES
C
      SUBROUTINE MOMPAE(MM2,C,DD,FP,Q,VV,M2,V,B2,W,EMM2,EM2,EVV,EV,EFP,E
     #Q,EC,EDD)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL*8 MM2,M2
      DIMENSION M2(5,3,3),C(3,3),DD(3),FP(2),MM2(3,3),A2(3,3),B2(3,3)
      DIMENSION VV(3),V(5,3),W(3),EMM2(3,3),EM2(5,3,3),EVV(3),EV(5,3)
      DIMENSION EB2(3,3),EW(3),EC(3,3),EDD(3),EA2(3,3)
C
      WRITE(*,*)'ACCESO A RUTINA MOMPAE'
C
C *** SE PROCEDE A CALCULAR EL TENSOR A2
C
      DO 100 I=1,3
      DO 110 J=1,3
      A2(I,J)=MM2(I,J)-DD(I)*DD(J)
 110  CONTINUE
 100  CONTINUE
C
C *** CALCULO DEL TENSOR B2
C
      DO 200 I=1,3
      DO 210 J=1,3
      B2(I,J)=1.D0/DSQRT(FP(1)*FP(2))*(C(I,J)+Q*DD(I)*DD(J))
 210  CONTINUE
 200  CONTINUE
C
C *** CALCULO DE LOS MOMENTOS PARCIALES
C
      DO 300 I=1,3
      DO 310 J=1,3
      M2(1,I,J)=A2(I,J)+FP(2)*B2(I,J)
      M2(2,I,J)=A2(I,J)-FP(1)*B2(I,J)
 310  CONTINUE
 300  CONTINUE
C
C *** CALCULO DE LAS VELOCIDADES
C
      DO 600 I=1,3
      W(I)=1.D0/DSQRT(FP(1)*FP(2))*DD(I)
      V(1,I)=VV(I)+FP(2)*W(I)
      V(2,I)=VV(I)-FP(1)*W(I)
 600  CONTINUE
C
C *** CALCULO DE ERRORES. EFP(1)=EFP(2)=EFP
C
      DO 760 I=1,3
      DO 770 J=1,3
      EA2(I,J)=DSQRT(EMM2(I,J)**2+(DD(I)*EDD(J)**2)+(EDD(I)*DD(J))**2)
 770  CONTINUE
 760  CONTINUE
C
```

```
      DO 700 I=1,3
      DO 710 J=1,3
      EB2(I,J)=DSQRT(1.D0/(FP(1)*FP(2))*(EC(I,J)**2+(EQ*DD(I)*DD(J))**2+
     #(Q*DD(I)*EDD(J))**2+(Q*EDD(I)*DD(J))**2)+0.25D0*(FP(1)*FP(2))**
     #(-3)*(C(I,J)+Q*DD(I)*DD(J))**2*(FP(1)**2+FP(2)**2)*EFP**2)
      WRITE(*,*)'B2',I,J,B2(I,J),EB2(I,J)
 710  CONTINUE
 700  CONTINUE
C
      DO 720 I=1,3
      EW(I)=EDD(I)**2/(FP(1)*FP(2))+(0.5D0*EFP*DD(I))**2*
     #(FP(1)*FP(2))**(-3)*(FP(2)**2+FP(1)**2)
      EW(I)=DSQRT(EW(I))
 720  CONTINUE
C
      DO 730 I=1,3
      EV(1,I)=DSQRT(EVV(I)**2+(EFP*W(I))**2+(EW(I)*FP(2))**2)
      EV(2,I)=DSQRT(EVV(I)**2+(EFP*W(I))**2+(EW(I)*FP(1))**2)
 730  CONTINUE
C
      DO 740 I=1,3
      DO 750 J=1,3
      EM2(1,I,J)=DSQRT(EA2(I,J)**2+(EFP*B2(I,J))**2+
     #(FP(2)*EB2(I,J))**2)
      EM2(2,I,J)=DSQRT(EA2(I,J)**2+(EFP*B2(I,J))**2+
     #(FP(1)*EB2(I,J))**2)
 750  CONTINUE
 740  CONTINUE
C***
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LOS MOMENTOS PARCIALES SUPUESTAS DOS POBLACIONES
C
      SUBROUTINE MOMPAR(MM2,C,DD,FP,Q,VV,M2,V,B2,W)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL*8 MM2,M2
      DIMENSION M2(5,3,3),C(3,3),DD(3),FP(2),MM2(3,3),A2(3,3),B2(3,3)
      DIMENSION VV(3),V(5,3),W(3)
C
      WRITE(*,*)'ACCESO A RUTINA MOMPAR'
C
C *** SE PROCEDE A CALCULAR EL TENSOR A2
C
      DO 100 I=1,3
      DO 110 J=1,3
      A2(I,J)=MM2(I,J)-DD(I)*DD(J)
 110  CONTINUE
 100  CONTINUE
C
C *** CALCULO DEL TENSOR B2
C
      DO 200 I=1,3
      DO 210 J=1,3
      B2(I,J)=1.D0/DSQRT(FP(1)*FP(2))*(C(I,J)+Q*DD(I)*DD(J))
 210  CONTINUE
 200  CONTINUE
C
C *** CALCULO DE LOS MOMENTOS PARCIALES
C
      DO 300 I=1,3
      DO 310 J=1,3
      M2(1,I,J)=A2(I,J)+FP(2)*B2(I,J)
      M2(2,I,J)=A2(I,J)-FP(1)*B2(I,J)
 310  CONTINUE
 300  CONTINUE
C
C *** CALCULO DE LAS VELOCIDADES
C
      DO 600 I=1,3
      W(I)=1.D0/DSQRT(FP(1)*FP(2))*DD(I)
      V(1,I)=VV(I)+FP(2)*W(I)
      V(2,I)=VV(I)-FP(1)*W(I)
 600  CONTINUE
C***
      RETURN
      END
```

```
C234567
C
C ******* CALCULO DE LAS VARIABLES EP, ES, EX, EY, EZ, ET OTRA VEZ
C ******* SON LOS ERRORES P,S,X,Y,Z,T Y AHORA SE CALCULAN PARA GENERAR
C ******* LA MATRIZ DE ERRORES DEL SISTEMA DE MINIMOS CUADRADOS
C ******* SE DA UNA ESTIMACION ESTADISTICA DEL ERROR SEGUN LA
C ******* EXPRESION DE QUE ES IGUAL AL INVERSO DE LA SUMA DE PESOS
C ******* EL ERROR DEL VECTOR SOLUCION SE GUARDA EN EVEC
C
      SUBROUTINE PESOER(EM3,ENU4,MA,H,P,S,X,Y,Z,T,PESO,EVEC,ES,EP)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION EM3(3,3,3),ENU4(3,3,3,3),D(3),EP(3,3),ES(3),EX(3,3,3,3)
      DIMENSION EY(3,3,3),EZ(3,3),ET(3),H(3,3)
      DIMENSION P(3,3),S(3),X(3,3,3,3),Y(3,3,3),Z(3,3),T(3)
      DIMENSION PESO(14,14),SIGMA(14),EVEC(2)
C ***
      WRITE(*,*)'ACCESO A RUTINA PESOER'
C ***
C
C ***** CALCULO DE LOS ERRORES DE P,S,X,Y,Z,T *****
C
C *** CALCULO DE EP(IA,IB)
C
      DO 100 IA=1,3
      DO 110 IB=1,3
      EP(IA,IB)=0.D0
      DO 120 I=1,3
      DO 130 J=1,3
      EP(IA,IB)=EP(IA,IB)+(H(IA,I)*H(IB,J)*EM3(I,J,MA))**2
 130  CONTINUE
 120  CONTINUE
      EP(IA,IB)=DSQRT(EP(IA,IB))
 110  CONTINUE
 100  CONTINUE
      WRITE(*,*)'ERROR DE P22 ********',EP(2,2)
C
C *** CALCULO DE ES(IA)
C
      DO 200 IA=1,3
      ES(IA)=0.D0
      DO 210 I=1,3
      ES(IA)=ES(IA)+(0.5D0*H(IA,I)*EM3(I,MA,MA))**2
 210  CONTINUE
      ES(IA)=DSQRT(ES(IA))
 200  CONTINUE
C
C *** CALCULO EX(IA,IB,IG,ID)
C
      DO 300 IA=1,3
      DO 310 IB=1,3
      DO 320 IG=1,3
      DO 330 ID=1,3
      EX(IA,IB,IG,ID)=0.D0
      DO 340 I=1,3
      DO 350 J=1,3
      DO 360 K=1,3
```

```
      DO 370 L=1,3
      EX(IA,IB,IG,ID)=EX(IA,IB,IG,ID)+(H(IA,I)*H(IB,J)*H(IG,K)*H(ID,L)*E
     #NU4(I,J,K,L))**2
  370 CONTINUE
  360 CONTINUE
  350 CONTINUE
  340 CONTINUE
      EX(IA,IB,IG,ID)=DSQRT(EX(IA,IB,IG,ID))
  330 CONTINUE
  320 CONTINUE
  310 CONTINUE
  300 CONTINUE
C
C *** CALCULO DE EY(IA,IB,IG)
C
      DO 400 IA=1,3
      DO 410 IB=1,3
      DO 420 IG=1,3
      EY(IA,IB,IG)=0.D0
      DO 430 I=1,3
      DO 440 J=1,3
      DO 450 K=1,3
      EY(IA,IB,IG)=EY(IA,IB,IG)+(H(IA,I)*H(IB,J)*H(IG,K)*ENU4(I,J,K,MA))
     #**2
  450 CONTINUE
  440 CONTINUE
  430 CONTINUE
      EY(IA,IB,IG)=DSQRT(EY(IA,IB,IG))
  420 CONTINUE
  410 CONTINUE
  400 CONTINUE
C
C *** CALCULO DE EZ(IA,IB)
C
      DO 500 IA=1,3
      DO 510 IB=1,3
      EZ(IA,IB)=0.D0
      DO 520 I=1,3
      DO 530 J=1,3
      EZ(IA,IB)=EZ(IA,IB)+(H(IA,I)*H(IB,J)*ENU4(I,J,MA,MA))**2
  530 CONTINUE
  520 CONTINUE
      EZ(IA,IB)=DSQRT(EZ(IA,IB))
  510 CONTINUE
  500 CONTINUE
C
C *** CALCULO DE ET(IA)
C
      DO 600 IA=1,3
      ET(IA)=0.D0
      DO 610 I=1,3
      ET(IA)=ET(IA)+(H(IA,I)*ENU4(I,MA,MA,MA))**2
  610 CONTINUE
      ET(IA)=DSQRT(ET(IA))
  600 CONTINUE
C ***
```

```
C
C ***** PREPARACION DE LA MATRIZ DE PESOS *****
C
C *** INICIALIZACION DE LA MATRIZ
      DO 700 I=1,14
      DO 710 J=1,14
      PESO(I,J)=0.D0
 710  CONTINUE
 700  CONTINUE
C
C *** CALCULO DE LOS CINCO PRIMEROS COEFICIENTES (1 - 5)
C
      N=0
      DO 800 I=1,2
      DO 810 J=I,2
      DO 820 K=J,2
      DO 830 L=K,2
      N=N+1
      AUX=P(I,J)*P(K,L)+P(I,K)*P(J,L)+P(I,L)*P(J,K)
      AUX1=DABS(P(I,J)*EP(K,L))+DABS(EP(I,J)*P(K,L))+DABS(P(I,K)*EP(J,L)
     #)+DABS(EP(I,K)*P(J,L))+DABS(P(I,L)*EP(J,K))+DABS(EP(I,L)*P(J,K))
      IF(AUX.NE.0.D0)THEN
             PESO(N,N)=(AUX1*X(I,J,K,L))**2/AUX**4+(EX(I,J,K,L)/AUX)**2
             PESO(N,N)=DSQRT(PESO(N,N))
             ELSE
             PESO(N,N)=0.D0
      ENDIF
 830  CONTINUE
 820  CONTINUE
 810  CONTINUE
 800  CONTINUE
C
C *** CALCULO DE LOS 4 SIGUIENTES COEFICIENTES (6 - 9)
C
      DO 900 I=1,2
      DO 910 J=I,2
      DO 920 K=J,2
      N=N+1
      AUX=DABS(P(I,J)*S(K)+P(I,K)*S(J)+P(J,K)*S(I))
      AUX1=DABS(P(I,J)*ES(K))+DABS(EP(I,J)*S(K))+DABS(P(I,K)*ES(J))+DABS
     #(EP(I,K)*S(J))+DABS(P(J,K)*ES(I))+DABS(EP(J,K)*S(I))
      IF(AUX.NE.0.D0)THEN
                PESO(N,N)=(EY(I,J,K)/AUX**2)+(AUX1*Y(I,J,K))**2/AUX**4
                PESO(N,N)=DSQRT(PESO(N,N))
                ELSE
                PESO(N,N)=0.D0
      ENDIF
 920  CONTINUE
 910  CONTINUE
 900  CONTINUE
C
C *** CALCULO DE LOS COEFICIENTES 10,11 y 12
C *** SE TOMA COMO REFERENCIA EL PRIMER COCIENTE (EMIN)
C
      EMIN=DABS(X(1,1,1,1)/(3.D0*P(1,1)**2))
C
```

```
      DEMIN=1.D0/DSQRT(EMIN)
      WRITE(*,*)'D3 DE LA PRIMERA ECUACION VALE',DEMIN
C
      DO 1000 I=1,2
      DO 1010 J=I,2
      N=N+1
      AUX=Z(I,J)-2.D0*EMIN*S(I)*S(J)
      AUX1=EZ(I,J)+2.D0*EMIN*(DABS(S(I)*ES(J))+DABS(ES(I)*S(J)))
      IF(AUX.NE.0.D0)THEN
                     PESO(N,N)=(AUX1*P(I,J))**2/AUX**4+EP(I,J)/AUX**2
                     PESO(N,N)=DSQRT(PESO(N,N))
                     ELSE
                     PESO(N,N)=0.D0
      ENDIF
 1010 CONTINUE
 1000 CONTINUE
C
C *** CALCULO DE LOS COEFICIENTES 13 y 14
C
      DO 1200 I=1,2
      N=N+1
      IF(T(I).NE.0.D0)THEN
                  PESO(N,N)=3.D0*(ET(I)*S(I))**2/T(I)**4+ES(I)/T(I)**2
                  PESO(N,N)=DSQRT(PESO(N,N))
                  ELSE
                  PESO(N,N)=0.D0
      ENDIF
 1200 CONTINUE
C
C *** SE EVALUA LA DISPERSION DE LOS RESULTADOS DE LAS NUEVE PRIMERAS
C *** ECUACIONES. SE LLAMA A LA RUTINA CTLDIS QUE NO ALTERA VALORES
C
C     WRITE(*,*)'COMO DESEA EVALUAR LOS PESOS? (1=MOMENT,2=DISPERSION)'
C     READ(*,*)NPI
C     IF(NPI.NE.1)THEN
C                 CALL CTLDIS(P,S,X,Y,Z,T,PESO)
C                 ELSE
C                 CONTINUE
C     ENDIF
C
C *** IMPRESION DE LA MATRIZ DE PESOS
C
      DO 1300 I=1,14
      WRITE(*,*)'EL ERROR ABSOLUTO ES',I,I,PESO(I,I)
 1300 CONTINUE
C
C *** NORMALIZACION DE ERRORES
C
      DO 1350 I=1,14
      IF(PESO(I,I).NE.0.D0)THEN
C *** SE GUARDA ERROR PARA CONOCER VARIANZA
                           SIGMA(I)=(1.D0/PESO(I,I))**2
                           PESO(I,I)=1.D0/PESO(I,I)
      ENDIF
1350  CONTINUE
C
```

```
      SUMA=0.D0
      DO 1400 I=1,14
      SUMA=SUMA+PESO(I,I)
 1400 CONTINUE
      WRITE(*,*)'LA SUMA DE INVERSOS DE ERRORES ES',SUMA
C
C *** SE ESTIMA EL ERROR DEL VECTOR SOLUCION COMO EL INVERSO DE LA
C *** SUMA DE INVERSOS DE LOS ERRORES AL CUADRADO.ESO SE GUARDO EN SIGMA
C
      VARIA1=0.D0
C *** PARA 1/D3**2 EL ERROR ES EL DE LAS 9 PRIMERAS ECUACIONES
      DO 1800 I=1,9
      VARIA1=VARIA1+SIGMA(I)
 1800 CONTINUE
      VARIA1=1.D0/VARIA1
      EVEC(1)=DSQRT(VARIA1)
      WRITE(*,*)'ERROR DE 1/D3**2 (PRIMERA INCOGNITA): ',EVEC(1)
C
C *** PARA C33/D3 EL ERROR ES DE LAS ECUACIONES 10 A 14
C *** PRIMERO LAS ECUACIONES 10 A 12
      N=9
      DO 1810 I=1,2
      DO 1820 J=I,2
      N=N+1
      AUX=EP(I,J)*(Z(I,J)-2.D0*EMIN*S(I)*S(J))
      AUX1=4.D0*(S(I)*ES(J)+ES(I)*S(J))*EMIN
      AUX2=2.D0*S(I)*S(J)*EVEC(1)
      IF(P(I,J).NE.0.D0)THEN
                SIGMA(N)=(AUX/P(I,J)**2)**2+(EZ(I,J)/P(I,J))**2+
     #          (AUX1/P(I,J))**2+(AUX2/P(I,J))**2
      ELSE
         SIGMA(N)=0.D0
      ENDIF
 1820 CONTINUE
 1810 CONTINUE
C *** Y DESPUES LAS ECUACIONES 13 Y 14
      DO 1830 I=1,2
      N=N+1
      IF(S(I).NE.0.D0)THEN
              SIGMA(N)=1/9.D0*((T(I)*ES(I))**2/S(I)**4+(ET(I)/S(I))**2)
      ELSE
              SIGMA(N)=0.D0
      ENDIF
 1830 CONTINUE
C
      VARIA2=0.D0
      DO 1840 I=10,14
      IF(SIGMA(I).NE.0.D0)THEN
         VARIA2=VARIA2+1.D0/SIGMA(I)
      ELSE
         CONTINUE
      ENDIF
 1840 CONTINUE
      VARIA2=1.D0/VARIA2
      EVEC(2)=DSQRT(VARIA2)
      WRITE(*,*)'ERROR DE C33/D3 (SEGUNDA INCOGNITA): ',EVEC(2)
```

```
C
C *** ELECCION DEL CALCULO DEL PESO
C
      WRITE(*,*)'ELIJA COMO CALCULAR LOS PESOS (1=**2, 2=EXP)'
      READ(*,*)NPI
      IF(NPI.EQ.1)THEN
C
C *** SE TOMA COMO PESO EL INVERSO DEL ERROR AL CUADRADO Y NORMALIZADO
C
      DO 1500 I=1,14
      PESO(I,I)=PESO(I,I)/SUMA*14.D0
      PESO(I,I)=PESO(I,I)**2
 1500 CONTINUE
C
C *** SE TOMAN PESOS EXPONENCIALES
C
      ELSE
         WRITE(*,*)'INDIQUE FACTOR DE NORMALIZACION'
         READ(*,*)FACT
         DO 1510 I=1,14
         PESO(I,I)=PESO(I,I)/SUMA*FACT
         PESO(I,I)=DEXP(PESO(I,I))
 1510 CONTINUE
      ENDIF
C *** SE ELIMINAN ECUACIONES MOLESTAS
      WRITE(*,*)'SI DESEA MODIFICAR ALGUN PESO DIGA NUMERO (NO=>15)'
      READ(*,*)NEC
      IF(NEC.LE.14)THEN
                   WRITE(*,*)'INDIQUE NUEVO PESO PARA ECUACION',NEC
                   READ(*,*)PESNEC
                   WRITE(*,*)'SE HA MODIFICADO LA ECUACION',NEC
                   PESO(NEC,NEC)=PESNEC
      ELSE
                   WRITE(*,*)'NO SE MODIFICA NINGUN PESO'
      ENDIF
C ***
C
C *** SE BUSCA EL PESO MAYOR
C
      DMAX=0.D0
      DO 1600 I=1,14
      IF(PESO(I,I).GT.DMAX)THEN
                           DMAX=PESO(I,I)
      ENDIF
 1600 CONTINUE
C *** SE ASIGNA UN PESO IGUAL AL MAXIMO A LAS ECUACIONES DE ERROR CERO
      DO 1700 I=1,14
      IF((PESO(I,I).EQ.0.D0).OR.(NPI.EQ.2.AND.PESO(I,I).EQ.1.D0))THEN
                                                      PESO(I,I)=DMAX
      ENDIF
 1700 CONTINUE
C ***
      RETURN
      END
```

```
C234567
C
C ****** CALCULO DE LAS VARIABLES P, S, X, Y, Z, T, A y B
C
      SUBROUTINE PSXYZT(M3,NU4,D,MA,P,S,X,Y,Z,T,A,B,H)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL*8 M3,NU4
      DIMENSION M3(3,3,3),NU4(3,3,3,3),D(3),P(3,3),S(3),X(3,3,3,3)
      DIMENSION Y(3,3,3),Z(3,3),T(3),B(14),A(14,2),H(3,3)
C ***
      WRITE(*,*)'ACCESO A RUTINA PSXYZT'
C ***
C
C *** SE PROCEDE A DEFINIR EL TENSOR H
C
      H(1,1)=D(3)
      H(1,2)=0.D0
      H(1,3)=-D(1)
      H(2,1)=0.D0
      H(2,2)=D(3)
      H(2,3)=-D(2)
      H(3,1)=0.D0
      H(3,2)=0.D0
      H(3,3)=1.D0
      DO 1000 I=1,3
 1000 CONTINUE
C
C *** CALCULO DE P(IA,IB)
C
      DO 100 IA=1,3
      DO 110 IB=1,3
      P(IA,IB)=0.D0
      DO 120 I=1,3
      DO 130 J=1,3
      P(IA,IB)=P(IA,IB)+ H(IA,I)*H(IB,J)*M3(I,J,MA)
 130  CONTINUE
 120  CONTINUE
 110  CONTINUE
 100  CONTINUE
C
C *** CALCULO DE S(IA)
C
      DO 200 IA=1,3
      S(IA)=0.D0
      DO 210 I=1,3
      S(IA)=S(IA)+0.5D0*H(IA,I)*M3(I,MA,MA)
 210  CONTINUE
 200  CONTINUE
C
C *** CALCULO X(IA,IB,IG,ID)
C
      DO 300 IA=1,3
      DO 310 IB=1,3
      DO 320 IG=1,3
      DO 330 ID=1,3
      X(IA,IB,IG,ID)=0.D0
```

```
      DO 340 I=1,3
      DO 350 J=1,3
      DO 360 K=1,3
      DO 370 L=1,3
      X(IA,IB,IG,ID)=X(IA,IB,IG,ID)+H(IA,I)*H(IB,J)*H(IG,K)*H(ID,L)*NU4(
     #I,J,K,L)
  370 CONTINUE
  360 CONTINUE
  350 CONTINUE
  340 CONTINUE
  330 CONTINUE
  320 CONTINUE
  310 CONTINUE
  300 CONTINUE
C
C *** CALCULO DE Y(IA,IB,IG)
C
      DO 400 IA=1,3
      DO 410 IB=1,3
      DO 420 IG=1,3
      Y(IA,IB,IG)=0.D0
      DO 430 I=1,3
      DO 440 J=1,3
      DO 450 K=1,3
      Y(IA,IB,IG)=Y(IA,IB,IG)+H(IA,I)*H(IB,J)*H(IG,K)*NU4(I,J,K,MA)
  450 CONTINUE
  440 CONTINUE
  430 CONTINUE
  420 CONTINUE
  410 CONTINUE
  400 CONTINUE
C
C *** CALCULO DE Z(IA,IB)
C
      DO 500 IA=1,3
      DO 510 IB=1,3
      Z(IA,IB)=0.D0
      DO 520 I=1,3
      DO 530 J=1,3
      Z(IA,IB)=Z(IA,IB)+ H(IA,I)*H(IB,J)*NU4(I,J,MA,MA)
  530 CONTINUE
  520 CONTINUE
  510 CONTINUE
  500 CONTINUE
C
C *** CALCULO DE T(IA)
C
      DO 600 IA=1,3
      T(IA)=0.D0
      DO 610 I=1,3
      T(IA)=T(IA)+H(IA,I)*NU4(I,MA,MA,MA)
  610 CONTINUE
  600 CONTINUE
C ***
C
C *** PREPARACION DEL SISTEMA DE MINIMOS CUADRADOS
```

```
C
      DO 2000 I=1,9
      A(I,2)=0.D0
 2000 CONTINUE
      DO 2100 I=13,14
      A(I,1)=0.D0
 2100 CONTINUE
      A(1,1)=3.D0*P(1,1)**2
      A(2,1)=3.D0*P(1,1)*P(1,2)
      A(3,1)=P(1,1)*P(2,2)+2.D0*P(1,2)**2
      A(4,1)=3.D0*P(1,2)*P(2,2)
      A(5,1)=3.D0*P(2,2)**2
C
      A(6,1)=3.D0*P(1,1)*S(1)
      A(7,1)=P(1,1)*S(2)+2.D0*P(1,2)*S(1)
      A(8,1)=P(2,2)*S(1)+2.D0*P(1,2)*S(2)
      A(9,1)=3.D0*P(2,2)*S(2)
C
      A(10,1)=2.D0*S(1)**2
      A(10,2)=P(1,1)
      A(11,1)=2.D0*S(1)*S(2)
      A(11,2)=P(1,2)
      A(12,1)=2.D0*S(2)**2
      A(12,2)=P(2,2)
C
      A(13,2)=3.D0*S(1)
      A(14,2)=3.D0*S(2)
C
      B(1)=X(1,1,1,1)
      B(2)=X(1,1,1,2)
      B(3)=X(1,1,2,2)
      B(4)=X(1,2,2,2)
      B(5)=X(2,2,2,2)
C
      B(6)=Y(1,1,1)
      B(7)=Y(1,1,2)
      B(8)=Y(1,2,2)
      B(9)=Y(2,2,2)
C
      B(10)=Z(1,1)
      B(11)=Z(1,2)
      B(12)=Z(2,2)
C
      B(13)=T(1)
      B(14)=T(2)
C
      WRITE(*,*)'LAS MATRICES SON'
      DO 700 I=1,14
C     WRITE(*,*)A(I,1),A(I,2),B(I)
 700  CONTINUE
C ***
      RETURN
      END
```

```
C234567
C
C ***** SE RECALCULAN LOS MOMENTOS PARA UNA DE LAS POBLACIONES
C ***** Y SE HACE UNA EVALUACION DEL ERROR DEL MODELO
C
      SUBROUTINE RECALE(FP,W,V,M2,VV,MM2,MM3,NU4,EMM2,EMM3,ENU4)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*8 M2,MM2,MM3,MM4,NU4
      DIMENSION M2(5,3,3),V(5,3),FP(2),VV(3),NU4(3,3,3,3)
      DIMENSION MM2(3,3),MM3(3,3,3)
      DIMENSION W(3),TM2(3,3),TM3(3,3,3)
      DIMENSION TM4(3,3,3,3),EMM2(3,3),EMM3(3,3,3),ENU4(3,3,3,3)
C ***
      WRITE(*,*)'ACCESO A RUTINA RECALE'
C ***
      WRITE(*,*)'FP1 Y FP2 SON',FP(1),FP(2)
C     WRITE(*,*)'QUIERE BARRERLAS ? 1=SI 2=NO'C
C     READ(*,*)NENO
C     IF(NENO.EQ.1)THEN
C        WRITE(*,*)'DIGA FP1'
C        READ(*,*)FP(1)
C        FP(2)=1.D0-FP(1)
C        WRITE(*,*)'PUES AHORA SON',FP(1),FP(2)
C     ELSE
C        CONTINUE
C     ENDIF
      FP12=FP(1)*FP(2)
C ***** RECALCULO DE MOMENTOS DE ORDEN 2 TOTALES
      DO 100 I=1,3
      DO 110 J=1,3
      TM2(I,J)=FP(1)*M2(1,I,J)+FP(2)*M2(2,I,J)+FP12*W(I)*W(J)
C     WRITE(*,*)'*** TM2',I,J,TM2(I,J)
 110  CONTINUE
 100  CONTINUE
C ***** RECALCULO DE MOMENTOS DE ORDEN 3 TOTALES
      DO 200 I=1,3
      DO 210 J=1,3
      DO 220 K=1,3
      TM3(I,J,K)=FP12*((M2(1,I,J)-M2(2,I,J))*W(K)+(M2(1,K,I)-M2(2,K,I))
     #*W(J)+(M2(1,J,K)-M2(2,J,K))*W(I))+FP12*(FP(2)-FP(1))
     #*W(I)*W(J)*W(K)
C     WRITE(*,*)'*** TM3',I,J,K,TM3(I,J,K)
 220  CONTINUE
 210  CONTINUE
 200  CONTINUE
C ***** RECALCULO DE MOMENTOS DE ORDEN 4 TOTALES
      DO 300 I=1,3
      DO 310 J=1,3
      DO 320 K=1,3
      DO 330 L=1,3
      TM4(I,J,K,L)=FP(1)*(M2(1,I,J)*M2(1,K,L)+M2(1,I,K)*M2(1,J,L)+
     #M2(1,I,L)*M2(1,J,K))+FP(2)*(M2(2,I,J)*M2(2,K,L)+M2(2,I,K)*
     #M2(2,J,L)+M2(2,I,L)*M2(2,J,K))
     #+FP12*((FP(2)*M2(1,I,J)+FP(1)*M2(2,I,J))*W(K)*W(L)+
     #(FP(2)*M2(1,I,K)+FP(1)*M2(2,I,K))*W(J)*W(L)+(FP(2)*M2(1,I,L)+
     #FP(1)*M2(2,I,L))*W(J)*W(K)+(FP(2)*M2(1,J,K)+FP(1)*M2(2,J,K))
```

```
     #*W(I)*W(L)+(FP(2)*M2(1,K,L)+FP(1)*M2(2,K,L))*W(I)*W(J)+(FP(2)*
     #M2(1,J,L)+FP(1)*M2(2,J,L))*W(I)*W(K))
     #+FP12*(1.D0-3.D0*FP12)*W(I)*W(J)*W(K)*W(L)
C     WRITE(*,*)'TM4  (MM4)',I,J,K,L,TM4(I,J,K,L)
      TM4(I,J,K,L)=TM4(I,J,K,L)-(TM2(I,J)*TM2(K,L)+TM2(I,K)*TM2(J,L)+
     #TM2(I,L)*TM2(J,K))
C     WRITE(*,*)'*** TM4  (NU4)',I,J,K,L,TM4(I,J,K,L)
  330 CONTINUE
  320 CONTINUE
  310 CONTINUE
  300 CONTINUE
C
C *** SE UTILIZA PARA EVALUAR EL ERROR DEL MODELO
C
      ERROR2=0.D0
      DO 1200 I=1,3
      DO 1210 J=I,3
      IF(EMM2(I,J).NE.0.D0)THEN
         ERROR2=ERROR2+((TM2(I,J)-MM2(I,J))/EMM2(I,J))**2
      ELSE
         CONTINUE
      ENDIF
 1210 CONTINUE
 1200 CONTINUE
      ERROR2=ERROR2/6.D0
      ERROR2=DSQRT(ERROR)
      WRITE(*,*)'CONTRIBUCION DEL ERROR DE MM2',ERROR2
C
      ERROR3=0.D0
      DO 1100 I=1,3
      DO 1110 J=I,3
      DO 1120 K=J,3
      IF(EMM3(I,J,K).NE.0.D0)THEN
         ERROR3=ERROR3+((TM3(I,J,K)-MM3(I,J,K))/EMM3(I,J,K))**2
      ELSE
         CONTINUE
      ENDIF
C     AVER=(TM3(I,J,K)-MM3(I,J,K))/EMM3(I,J,K)
C     WRITE(*,*)'JI MOMENTO',I,J,K,AVER
 1120 CONTINUE
 1110 CONTINUE
 1100 CONTINUE
      ERRJI2=ERROR3
      ERROR3=ERROR3/10.D0
      ERROR3=DSQRT(ERROR3)
      WRITE(*,*)'CONTRIBUCION DEL ERROR DE MM3',ERROR3
C
      ERROR4=0.D0
      DO 1000 I=1,3
      DO 1010 J=I,3
      DO 1020 K=J,3
      DO 1030 L=K,3
      IF(ENU4(I,J,K,L).NE.0.D0)THEN
         ERROR4=ERROR4+((TM4(I,J,K,L)-NU4(I,J,K,L))/ENU4(I,J,K,L))**2
      ELSE
         CONTINUE
```

```
      ENDIF
 1030 CONTINUE
 1020 CONTINUE
 1010 CONTINUE
 1000 CONTINUE
      ERRJI2=ERRJI2+ERROR4
      ERROR4=ERROR4/15.D0
      ERROR4=DSQRT(ERROR4)
      WRITE(*,*)'CONTRIBUCION DEL ERROR DE NU4',ERROR4
C
      ERROR=ERROR2+ERROR3+ERROR4
      WRITE(*,*)'***** ERROR TOTAL',ERROR
C *** SE CALCULA JI2 DE DOS FORMAS MAS ADECUADAS
C *** COMO HAY 15 CONDICIONES DE LIGADURA, ESOS SON LOS GRADOS DE LIBERT
AD
C *** PERO TAMBIEN ES VERDAD QUE HAY 10+15=25 SUMANDOS. LUEGO SE DA JI2
C *** DIVIDIDO POR 25 Y LUEGO LA RAIZ.
      ERRJ25=ERRJI2/25.D0
      ERRJ25=DSQRT(ERRJ25)
      ERROJI=ERRJI2/15.D0
      WRITE(*,*)'JI CUADRADO/15 (P) ====',ERROJI
      ERROJI=DSQRT(ERROJI)
      ERRJI2=DSQRT(ERRJI2)
      ERRJI2=ERRJI2/15.D0
      WRITE(*,*)'(RAIZ DE JI CUADRADO)/15 ====',ERRJI2
      WRITE(*,*)'RAIZ DE (JI CUADRADO/15) ====',ERROJI
      WRITE(*,*)'RAIZ DE (JI CUADRADO/25) ======',ERRJ25
C
      RETURN
      END
```

```
C234567
C
C ***** APERTURA Y LECTURA DEL FICHERO DE DATOS PREPARADO PAR FORMATO
C ***** DE UNA MUESTRA SINTETICA
C
      SUBROUTINE SINMO2(VV,MM2,MM3,MM4,MA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*8 MM2,MM3,MM4,MAYOR
      DIMENSION MM2(3,3),MM3(3,3,3),MM4(3,3,3,3),VV(3)
C ***
      WRITE(*,*)'ACCESO A RUTINA SINMO2'
C ***
C *** APERTURA DE LOS FICHEROS ***
      OPEN(1,FILE='SINTE.DAT')
C *** LECTURA DE LOS MOMENTOS TOTALES
C VV=VELOCIDAD DEL CENTROIDE
C
      READ(1,500)VV(1),VV(2),VV(3)
C
      READ(1,510)MM2(1,1),MM2(1,2),MM2(1,3),MM2(2,2),MM2(2,3),MM2(3,3)
      DO 400 I=1,3
      DO 410 J=I,3
      MM2(I,J)=MM2(I,J)
      MM2(J,I)=MM2(I,J)
 410  CONTINUE
 400  CONTINUE
C
      READ(1,520)MM3(1,1,1),MM3(1,1,2),MM3(1,1,3),MM3(1,2,2),MM3(1,2,3)
     #,MM3(1,3,3),MM3(2,2,2),MM3(2,2,3),MM3(2,3,3),MM3(3,3,3)
      DO 420 I=1,3
      DO 430 J=I,3
      DO 440 K=J,3
      MM3(I,J,K)=MM3(I,J,K)
      MM3(J,I,K)=MM3(I,J,K)
      MM3(K,J,I)=MM3(I,J,K)
      MM3(I,K,J)=MM3(I,J,K)
      MM3(J,K,I)=MM3(I,J,K)
      MM3(K,I,J)=MM3(I,J,K)
 440  CONTINUE
 430  CONTINUE
 420  CONTINUE
C
      READ(1,530)MM4(1,1,1,1),MM4(1,1,1,2),MM4(1,1,1,3),MM4(1,1,2,2),MM
     #4(1,1,2,3),MM4(1,1,3,3),MM4(1,2,2,2),MM4(1,2,2,3),MM4(1,2,3,3),MM
     #4(1,3,3,3),MM4(2,2,2,2),MM4(2,2,2,3),MM4(2,2,3,3),MM4(2,3,3,3),MM
     #4(3,3,3,3)
      DO 450 I=1,3
      DO 460 J=I,3
      DO 470 K=J,3
      DO 480 L=K,3
      MM4(I,J,K,L)=MM4(I,J,K,L)
      MM4(I,K,J,L)=MM4(I,J,K,L)
      MM4(I,L,K,J)=MM4(I,J,K,L)
      MM4(I,J,L,K)=MM4(I,J,K,L)
      MM4(I,K,L,J)=MM4(I,J,K,L)
      MM4(I,L,J,K)=MM4(I,J,K,L)
```

```
      MM4(J,I,K,L)=MM4(I,J,K,L)
      MM4(J,K,I,L)=MM4(I,J,K,L)
      MM4(J,L,K,I)=MM4(I,J,K,L)
      MM4(J,I,L,K)=MM4(I,J,K,L)
      MM4(J,K,L,I)=MM4(I,J,K,L)
      MM4(J,L,I,K)=MM4(I,J,K,L)
      MM4(K,I,J,L)=MM4(I,J,K,L)
      MM4(K,J,I,L)=MM4(I,J,K,L)
      MM4(K,L,J,I)=MM4(I,J,K,L)
      MM4(K,I,L,J)=MM4(I,J,K,L)
      MM4(K,J,L,I)=MM4(I,J,K,L)
      MM4(K,L,I,J)=MM4(I,J,K,L)
      MM4(L,I,J,K)=MM4(I,J,K,L)
      MM4(L,J,I,K)=MM4(I,J,K,L)
      MM4(L,K,J,I)=MM4(I,J,K,L)
      MM4(L,I,K,J)=MM4(I,J,K,L)
      MM4(L,J,K,I)=MM4(I,J,K,L)
      MM4(L,K,I,J)=MM4(I,J,K,L)
 480  CONTINUE
 470  CONTINUE
 460  CONTINUE
 450  CONTINUE
C
 500  FORMAT(1X,F12.2,1X,F12.2,1X,F12.2)
 510  FORMAT(1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2)
 520  FORMAT(1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F1
     #2.2,1X,F12.2,1X,F12.2,1X,F12.2)
 530  FORMAT(1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F1
     #2.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2
     #,1X,F12.2)
C
C *** LOCALIZACION DE LA POBLACION DE REFERENCIA
C
      MAYOR=-10000000.D0
      DO 700 I=1,3
      IF(VV(I).GE.MAYOR) THEN
                         MAYOR=VV(I)
                         MA=I
      ENDIF
 700  CONTINUE
      MA=3
      WRITE(*,*)'EL MAYOR ES',MA
C
      RETURN
      END
```

```
C234567
C
C ***** APERTURA Y LECTURA DEL FICHERO DE DATOS DE ERRORES PARA UNA
C ***** MUESTRA SINTETICA
C
      SUBROUTINE SINRO2(VV,MM2,MM3,MM4,EVV,EMM2,EMM3,ENU4)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*8 MM2,MM3,MM4
      DIMENSION EMM2(3,3),EMM3(3,3,3),EMM4(3,3,3,3),EVV(3)
      DIMENSION MM2(3,3),MM3(3,3,3),MM4(3,3,3,3),VV(3),ENU4(3,3,3,3)
C ***
      WRITE(*,*)'ACCESO A RUTINA SINRO2'
C ***
C *** APERTURA DEL FICHERO
      OPEN(1,FILE='SINER.DAT')
C *** LECTURA DE LOS ERRORES
C EVV=ERROR DE LA VELOCIDAD DEL CENTROIDE
C
      READ(1,500)EVV(1),EVV(2),EVV(3)
C
      READ(1,510)EMM2(1,1),EMM2(1,2),EMM2(1,3),EMM2(2,2),EMM2(2,3),EMM2(
     #3,3)
      DO 400 I=1,3
      DO 410 J=I,3
      EMM2(I,J)=EMM2(I,J)
      EMM2(J,I)=EMM2(I,J)
 410  CONTINUE
 400  CONTINUE
C
      READ(1,520)EMM3(1,1,1),EMM3(1,1,2),EMM3(1,1,3),EMM3(1,2,2),EMM3(1,
     #2,3),EMM3(1,3,3),EMM3(2,2,2),EMM3(2,2,3),EMM3(2,3,3),EMM3(3,3,3)
      DO 420 I=1,3
      DO 430 J=I,3
      DO 440 K=J,3
      EMM3(I,J,K)=EMM3(I,J,K)
      EMM3(J,I,K)=EMM3(I,J,K)
      EMM3(K,J,I)=EMM3(I,J,K)
      EMM3(I,K,J)=EMM3(I,J,K)
      EMM3(J,K,I)=EMM3(I,J,K)
      EMM3(K,I,J)=EMM3(I,J,K)
 440  CONTINUE
 430  CONTINUE
 420  CONTINUE
C
      READ(1,530)EMM4(1,1,1,1),EMM4(1,1,1,2),EMM4(1,1,1,3),EMM4(1,1,2,2)
     #,EMM4(1,1,2,3),EMM4(1,1,3,3),EMM4(1,2,2,2),EMM4(1,2,2,3),EMM4(1,2
     #,3,3),EMM4(1,3,3,3),EMM4(2,2,2,2),EMM4(2,2,2,3),EMM4(2,2,3,3),EMM
     #4(2,3,3,3),EMM4(3,3,3,3)
      DO 450 I=1,3
      DO 460 J=I,3
      DO 470 K=J,3
      DO 480 L=K,3
      EMM4(I,J,K,L)=EMM4(I,J,K,L)
      EMM4(I,K,J,L)=EMM4(I,J,K,L)
      EMM4(I,L,K,J)=EMM4(I,J,K,L)
      EMM4(I,J,L,K)=EMM4(I,J,K,L)
```

```
      EMM4(I,K,L,J)=EMM4(I,J,K,L)
      EMM4(I,L,J,K)=EMM4(I,J,K,L)
      EMM4(J,I,K,L)=EMM4(I,J,K,L)
      EMM4(J,K,I,L)=EMM4(I,J,K,L)
      EMM4(J,L,K,I)=EMM4(I,J,K,L)
      EMM4(J,I,L,K)=EMM4(I,J,K,L)
      EMM4(J,K,L,I)=EMM4(I,J,K,L)
      EMM4(J,L,I,K)=EMM4(I,J,K,L)
      EMM4(K,I,J,L)=EMM4(I,J,K,L)
      EMM4(K,J,I,L)=EMM4(I,J,K,L)
      EMM4(K,L,J,I)=EMM4(I,J,K,L)
      EMM4(K,I,L,J)=EMM4(I,J,K,L)
      EMM4(K,J,L,I)=EMM4(I,J,K,L)
      EMM4(K,L,I,J)=EMM4(I,J,K,L)
      EMM4(L,I,J,K)=EMM4(I,J,K,L)
      EMM4(L,J,I,K)=EMM4(I,J,K,L)
      EMM4(L,K,J,I)=EMM4(I,J,K,L)
      EMM4(L,I,K,J)=EMM4(I,J,K,L)
      EMM4(L,J,K,I)=EMM4(I,J,K,L)
      EMM4(L,K,I,J)=EMM4(I,J,K,L)
  480 CONTINUE
  470 CONTINUE
  460 CONTINUE
  450 CONTINUE
C
  500 FORMAT(1X,F12.2,1X,F12.2,1X,F12.2)
  510 FORMAT(1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2)
  520 FORMAT(1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F1
     #2.2,1X,F12.2,1X,F12.2,1X,F12.2)
  530 FORMAT(1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F1
     #2.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2,1X,F12.2
     #,1X,F12.2)
C
C     DO 9990 I=1,3
C     DO 9991 J=1,3
C     DO 9992 K=1,3
C     DO 9993 L=1,3
C     WRITE(*,*)'MOMENTO DE ORDEN 4',I,J,K,L,MM4(I,J,K,L)
C     WRITE(*,*)'ERRORES DE ORDEN 4',I,J,K,L,EMM4(I,J,K,L)
C9993 CONTINUE
C9992 CONTINUE
C9991 CONTINUE
C9990 CONTINUE
C
C *** SE CONVIERTEN EN CERO LOS MOMENTOS SUPUESTAMENTE NULOS
C *** SE TOMA COMO CRITERIO MMX < TN*EMMX TN TOMA EL VALOR QUE SE ASIGNE
C *** EL ERROR DE UN MOMENTO ASI SE CONSIDERA NULO
C
      WRITE(*,*)'INDIQUE VALOR DE DISCRIMINACION DEL ERROR'
      READ(*,*)TN
      WRITE(*,*)'SE ANULAN LOS VALORES <',TN,'* ERROR'
C
      DO 1000 I=1,3
      DO 1010 J=I,3
      IF(DABS(MM2(I,J)).LT.TN*EMM2(I,J))THEN
                                        WRITE(*,*)MM2(I,J)
```

```
                                        MM2(I,J)=0.D0
                                        EMM2(I,J)=0.D0
                                    WRITE(*,*)'** ELIMINADO MM2',I,J
      ENDIF
 1010 CONTINUE
 1000 CONTINUE
C
      DO 1100 I=1,3
      DO 1110 J=1,3
      DO 1120 K=1,3
      IF(DABS(MM3(I,J,K)).LT.TN*EMM3(I,J,K))THEN
                                              WRITE(*,*)MM3(I,J,K)
                                              MM3(I,J,K)=0.D0
                                              EMM3(I,J,K)=0.D0
                                    WRITE(*,*)'*** ELIMINADO MM3',I,J,K
      ENDIF
 1120 CONTINUE
 1110 CONTINUE
 1100 CONTINUE
C
      DO 1200 I=1,3
      DO 1210 J=1,3
      DO 1220 K=1,3
      DO 1230 L=1,3
      IF(DABS(MM4(I,J,K,L)).LT.TN*EMM4(I,J,K,L))THEN
                                                WRITE(*,*)MM4(I,J,K,L)
                                                MM4(I,J,K,L)=0.D0
                                                EMM4(I,J,K,L)=0.D0
                                    WRITE(*,*)'**** ELIMINADO MM4',I,J,K,L
      ENDIF
 1230 CONTINUE
 1220 CONTINUE
 1210 CONTINUE
 1200 CONTINUE
C
C *** SE CONVIERTE EL ERROR DE EMM4 A ENU4
C
      DO 1300 I=1,3
      DO 1310 J=1,3
      DO 1320 K=1,3
      DO 1330 L=1,3
C     ENU4(I,J,K,L)=EMM4(I,J,K,L)+DABS(MM2(I,J)*EMM2(K,L))+DABS(EMM2(I,J
C    #)*MM2(K,L))+DABS(MM2(I,K)*EMM2(J,L))+DABS(EMM2(I,K)*MM2(J,L))+DABS
C    #(MM2(I,L)*EMM2(J,K))+DABS(EMM2(I,L)*MM2(J,K))
C
      ENU4(I,J,K,L)=DSQRT(EMM4(I,J,K,L)**2+(DABS(MM2(I,J)*EMM2(K,L))+DAB
     #S(EMM2(I,J)*MM2(K,L)))**2+(DABS(MM2(I,K)*EMM2(J,L))+DABS(EMM2(I,K)
     #*MM2(J,L)))**2+(DABS(MM2(I,L)*EMM2(J,K))+DABS(EMM2(I,L)*MM2(J,K)))
     #**2)
C
 1330 CONTINUE
 1320 CONTINUE
 1310 CONTINUE
 1300 CONTINUE
C
      WRITE(*,*)'***** COMPROBACION DE LECTURA DE DATOS *****'
```

```
C
      WRITE(*,*)'1 1',MM2(1,1),EMM2(1,1)
      WRITE(*,*)'1 2',MM2(1,3),EMM2(1,3)
      WRITE(*,*)'1 3',MM2(1,2),EMM2(1,2)
      WRITE(*,*)'2 2',MM2(3,3),EMM2(3,3)
      WRITE(*,*)'2 3',MM2(3,2),EMM2(3,2)
      WRITE(*,*)'3 3',MM2(2,2),EMM2(2,2)
      WRITE(*,*)' ***** '
      WRITE(*,*)'1 1 1',MM3(1,1,1),EMM3(1,1,1)
      WRITE(*,*)'1 1 2',MM3(1,1,3),EMM3(1,1,3)
      WRITE(*,*)'1 1 3',MM3(1,1,2),EMM3(1,1,2)
      WRITE(*,*)'1 2 2',MM3(1,3,3),EMM3(1,3,3)
      WRITE(*,*)'1 2 3',MM3(1,3,2),EMM3(1,3,2)
      WRITE(*,*)'1 3 3',MM3(1,2,2),EMM3(1,2,2)
      WRITE(*,*)'2 2 2',MM3(3,3,3),EMM3(3,3,3)
      WRITE(*,*)'2 2 3',MM3(3,3,2),EMM3(3,3,2)
      WRITE(*,*)'2 3 3',MM3(3,2,2),EMM3(3,2,2)
      WRITE(*,*)'3 3 3',MM3(2,2,2),EMM3(2,2,2)
      WRITE(*,*)' ***** '
      WRITE(*,*)'1 1 1 1',MM4(1,1,1,1),EMM4(1,1,1,1)
      WRITE(*,*)'1 1 1 2',MM4(1,1,1,3),EMM4(1,1,1,3)
      WRITE(*,*)'1 1 1 3',MM4(1,1,1,2),EMM4(1,1,1,2)
      WRITE(*,*)'1 1 2 2',MM4(1,1,3,3),EMM4(1,1,3,3)
      WRITE(*,*)'1 1 2 3',MM4(1,1,3,2),EMM4(1,1,3,2)
      WRITE(*,*)'1 1 3 3',MM4(1,1,2,2),EMM4(1,1,2,2)
      WRITE(*,*)'1 2 2 2',MM4(1,3,3,3),EMM4(1,3,3,3)
      WRITE(*,*)'1 2 2 3',MM4(1,3,3,2),EMM4(1,3,3,2)
      WRITE(*,*)'1 2 3 3',MM4(1,3,2,2),EMM4(1,3,2,2)
      WRITE(*,*)'1 3 3 3',MM4(1,2,2,2),EMM4(1,2,2,2)
      WRITE(*,*)'2 2 2 2',MM4(3,3,3,3),EMM4(3,3,3,3)
      WRITE(*,*)'2 2 2 3',MM4(3,3,3,2),EMM4(3,3,3,2)
      WRITE(*,*)'2 2 3 3',MM4(3,3,2,2),EMM4(3,3,2,2)
      WRITE(*,*)'2 3 3 3',MM4(3,2,2,2),EMM4(3,2,2,2)
      WRITE(*,*)'3 3 3 3',MM4(2,2,2,2),EMM4(2,2,2,2)
C
C
      RETURN
      END
```

```
C234567
C
C ***** CALCULO DE LOS MOMENTOS PARCIALES DE UNA MUESTRA SINTETICA *****
C ***** TENIENDO PRESENTES LOS ERRORES DE CADA MOMENTO             *****
C ***** CASO QUE SOLO CAMBIAN LAS RUTINAS LEEMOM Y LERROR          *****
C ***** DADO QUE TIENEN UN FORMATO DE LECTURA DIFERENTE DEL        *****
C ***** CASO DE ERICKSSON. AQUI ES UNA MUESTRA SINTETICA           *****
C ***** EN ESTE CASO, ADEMAS, LAS D MINUSCULAS NO TIENEN ERROR     *****
C ***** LOS ERRORES CONSIDERADOS SON LOS ABSOLUTOS Y SE CALCULAN   *****
C ***** CON SUMA CUADRATICA                                       *****
C ***** ADEMAS PERMITE REALIZAR CALCULO DE RESIDUS                *****
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL*8 M2,MM2,MM3,MM4,NU4,M111,M222
      DIMENSION NU4(3,3,3,3)
      DIMENSION DD(3),MM2(3,3),MM3(3,3,3),MM4(3,3,3,3),VV(3),C(3,3)
      DIMENSION D(3),FP(2),H(3,3),M2(5,3,3),V(5,3),W(3),B2(3,3)
      DIMENSION P(3,3),S(3),X(3,3,3,3),Y(3,3,3),Z(3,3),T(3)
      DIMENSION B(14),A(14,2),VEC(2),EA(14,2),EB(14),PESO(14,14)
      DIMENSION EMM2(3,3),EMM3(3,3,3),ENU4(3,3,3,3),EVV(3),F(2)
      DIMENSION EDD(3),EVEC(2),ES(3),EP(3,3),EC(3,3),EM2(5,3,3),EV(5,3)
C ***
      WRITE(*,*)'SE CALCULAN LOS MOMENTOS DE UNA MUESTRA SINTETICA'
C ***
C
C LECTURA DE LOS DATOS DE LAS POBLACIONES PARCIALES
C
      CALL SINMO2(VV,MM2,MM3,MM4,MA)
      CALL SINRO2(VV,MM2,MM3,MM4,EVV,EMM2,EMM3,ENU4)
C
C *** CALCULO DE LAS NU
C
      CALL CALNU(MM2,MM4,NU4)
C
C *** CONOCIDOS LOS TODOS LOS MOMENTOS DE LA MUESTRA TOTAL SE PROCEDE
C *** A RECONSTRUIR LOS MOMENTOS PARCIALES COMO SI HUBIESEN SOLO DOS
C *** POBLACIONES
C
C *** CALCULO DE LAS VARIABLES AUXILIARES D
C *** LAS D MINUSCULAS SE ESCRIBEN D Y LAS MAYUSCULAS DD
C
      D(1)=MM2(1,3)*NU4(1,1,3,3)**2*(NU4(1,1,1,1)*MM2(2,3)**4-NU4(2,2,2,
     #2)*MM2(1,3)**4)/(MM3(1,1,3)**2*3.D0*(NU4(1,1,3,3)**2*MM2(2,3)**4-N
     #U4(2,2,3,3)**2*MM2(1,3)**4))
 9999 CALL CALDES(MM3,D)
C
C *** CALCULOS DE P, S, X, Y, Z, T, A , B y SUS ERRORES
C
      CALL PSXYZT(MM3,NU4,D,MA,P,S,X,Y,Z,T,A,B,H)
      CALL PESOER(EMM3,ENU4,MA,H,P,S,X,Y,Z,T,PESO,EVEC,ES,EP)
      CALL EAEB(PESO,A,B,EA,EB)
C
C *** RESOLUCION DEL SISTEMA DE MINIMOS CUADRADOS
C
      NDIM=14
      CALL MINCUA(EA,EB,NDIM,VEC)
```

```
C
C *** CALCULO DE LAS DD MAYUSCULAS
C
      CALL CALDDE(MM3,VEC,D,DD,C33,EVEC,EDD,EC33)
C
C *** CALCULO DEL TENSOR C2 PARA OBTENER LOS SIGNOS
C
      C(3,3)=C33
      WRITE(*,*)'C33 VALE ',C33
C
      CALL CALCCE(D,DD,P,S,C,ES,EP,EDD,EC33,EC)
C
C *** CALCULO DE Q SEGUN DOS EXPRESIONES
C
      CALL CALQUE(MM3,NU4,C,DD,FP,Q,EMM3,EC,EDD,EFP,EQ)
C
C *** CALCULO DE LAS C Y D RESTANTES
C
C     CALL CESDES(NU4,DD,C)
C
      NP=2
C
C *** CALCULO DE LOS MOMENTOS PARCIALES
C
      CALL MOMPAE(MM2,C,DD,FP,Q,VV,M2,V,B2,W,EMM2,EM2,EVV,EV,EFP,EQ,EC,
     #EDD)
C
C*******************
C
C *** SE ENTRA EN EL BUCLE DE RECALCULO
C
      WRITE(*,*)'*** DESEA ITERAR LA ALGUNA POBLACION ? (1=PRIMERA 2=SEG
     #UNDA 3=NINGUNA)'
C     READ(*,*)NSI
      NSI=3
      IF(NSI.EQ.1)THEN
                  WRITE(*,*)'SE ITERA LA POBLACION MAYORITARIA'
                  CALL RECAL2(B2,FP,W,V,M2,VV,MM2,MM3,NU4)
                  GOTO 9999
      ELSE
         IF(NSI.EQ.2)THEN
                  WRITE(*,*)'SE ITERA LA POBLACION MINORITARIA'
                  CALL RECAL(B2,FP,W,V,M2,VV,MM2,MM3,NU4)
                  GOTO 9999
         ELSE
            CONTINUE
         ENDIF
      ENDIF
      CALL RECALE(FP,W,V,M2,VV,MM2,MM3,NU4,EMM2,EMM3,ENU4)
C
C*******************
      OPEN(2,FILE='RESSIN.TXT',STATUS='NEW')
      WRITE(*,*)'MOMENTOS DE ORDEN 2 GLOBALES'
      WRITE(*,*)MM2(1,1),MM2(2,2),MM2(3,3),VV(1),VV(2),VV(3)
      WRITE(*,*)'MOMENTOS DE ORDEN 2 SEPARADOS'
      DO 200 N=1,NP
```

```
      WRITE(2,*)'******** NOTACION MATEMATICA ************** POBLACION'
     #,N,' **********'
      WRITE(2,*)M2(N,1,1),M2(N,2,2),M2(N,3,3),V(N,1),V(N,2),V(N,3),FP
     #(N),M2(N,1,2),M2(N,1,3),M2(N,2,3)
      WRITE(2,*)EM2(N,1,1),EM2(N,2,2),EM2(N,3,3),EV(N,1),EV(N,2),EV(N,3)
     #,EFP,EM2(N,1,2),EM2(N,1,3),EM2(N,2,3)
C
      WRITE(*,*)'******** NOTACION MATEMATICA ************** POBLACION'
     #,N,' **********'
      WRITE(*,*)M2(N,1,1),M2(N,2,2),M2(N,3,3),V(N,1),V(N,2),V(N,3),FP
     #(N),M2(N,1,2),M2(N,1,3),M2(N,2,3)
      WRITE(*,*)EM2(N,1,1),EM2(N,2,2),EM2(N,3,3),EV(N,1),EV(N,2),EV(N,3)
     #,EFP,EM2(N,1,2),EM2(N,1,3),EM2(N,2,3)
 200  CONTINUE
C
      DO 210 N=1,NP
      WRITE(2,*)' '
      WRITE(2,*)' '
      WRITE(2,*)' '
      WRITE(2,*)'******** NOTACION ASTRONOMICA ************** POBLACION'
     #,N,' **********'
      WRITE(2,*)' '
      WRITE(2,1000)
      WRITE(2,1010)M2(N,1,1),M2(N,3,3),M2(N,2,2),M2(N,1,3),M2(N,1,2),
     #M2(N,2,3),V(N,1),V(N,3),V(N,2),FP(N)
      WRITE(2,1010)EM2(N,1,1),EM2(N,3,3),EM2(N,2,2),EM2(N,1,3),
     #EM2(N,1,2),EM2(N,2,3),EV(N,1),EV(N,3),EV(N,2),EFP
 210  CONTINUE
C
 1000 FORMAT(1X,'   M11      M22       M33       M12      M13      M23
     #  V1     V2     V3    FP')
 1010 FORMAT(1X,F8.2,1X,F8.2,1X,F8.2,1X,F8.2,1X,F8.2,1X,F8.2,1X,F5.1,1X,
     #F5.1,1X,F5.1,1X,F4.2)
 720  FORMAT(1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F4.2)
 700  FORMAT(1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F4.2)
 710  FORMAT(1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1)
C******************
C
      STOP
      END
```

```
C234567
C
C ***** CALCULO DE LOS MOMENTOS PARCIALES DE N POBLACIONES *****
C ***** RECOMPONIENDO DE NUEVO LAS DOS POBLACIONES OBTENIDAS *****
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL*8 M2,M4,MM2,MM3,MM4,NU4
      DIMENSION F(5),V(5,3),M2(5,3,3),M4(5,3,3,3,3),NU4(3,3,3,3)
      DIMENSION DD(3),MM2(3,3),MM3(3,3,3),MM4(3,3,3,3),VV(3),C(3,3)
      DIMENSION D(3),DELTA(5,3),FP(2),H(3,3),B2(3,3),W(3)
      DIMENSION P(3,3),S(3),X(3,3,3,3),Y(3,3,3),Z(3,3),T(3)
      DIMENSION B(14),A(14,2),VEC(2)
C ***
      WRITE(*,*)'INICIO DE PROGRAMA'
      WRITE(*,*)'CALCULO DE LOS MOMENTOS PARCIALES DE N POBLACIONES'
      WRITE(*,*)'DA LA OPCION DE RECOMPONER PARA ESTIMAR EL ERROR'
C ***
C
C SE PROCEDE A INICIALIZAR LAS VARIABLES M2 Y NP Y A LA
C LECTURA DE LOS DATOS DE LAS POBLACIONES PARCIALES
C NP=NUMERO DE POBLACIONES
C M2=MOMENTOS DE ORDEN 2 DE CADA POBLACION PARCIAL
C
      NSI=0
      CALL LECTU(NP,M2,V,F,MA)
C
C *** CALCULO DE LOS MOMENTOS PARA LA MUESTRA TOTAL
C
 9999 CALL CALDEL(F,V,NP,DELTA,VV)
      CALL CALMOM(M2,DELTA,F,NP,MM2,MM3,MM4)
      CALL CALNU(MM2,MM4,NU4)
C
C *** CONOCIDOS LOS TODOS LOS MOMENTOS DE LA MUESTRA TOTAL SE PROCEDE
C *** A RECONSTRUIR LOS MOMENTOS PARCIALES COMO SI HUBIESEN SOLO DOS
C *** POBLACIONES
C
C *** CALCULO DE LAS VARIABLES AUXILIARES D
C *** LAS D MINUSCULAS SE ESCRIBEN D Y LAS MAYUSCULAS DD
C
      CALL CALDES(MM3,D)
C
C *** CALCULOS DE P, S, X, Y, Z, T, A y B
C
      CALL PSXYZT(MM3,NU4,D,MA,P,S,X,Y,Z,T,A,B,H)
C
C *** RESOLUCION DEL SISTEMA DE MINIMOS CUADRADOS
C
      NDIM=14
      CALL MINCUA(A,B,NDIM,VEC)
C
C *** CALCULO DE LAS DD MAYUSCULAS
C
      CALL CALDD(MM3,VEC,D,DD,C33)
C
C *** CALCULO DEL TENSOR C2 PARA OBTENER LOS SIGNOS
C
```

```
      C(3,3)=C33
      WRITE(*,*)'C33 VALE ',C33
C
      CALL CALCC(D,DD,P,S,C)
C
C *** CALCULO DE Q SEGUN DOS EXPRESIONES
C
      CALL  CALQU(MM3,NU4,C,DD,FP,Q)
C
C *** CALCULO DE LAS C Y D RESTANTES
C
C     CALL CESDES(NU4,DD,C)
C
C*******************
      OPEN(2,FILE='RESU.TXT',STATUS='NEW')

      WRITE(2,*)' '
      WRITE(2,510)
      DO 100 N=1,NP
      WRITE(2,500)M2(N,1,1),M2(N,1,2),M2(N,1,3),M2(N,2,2),M2(N,2,3),M2(N
     #,3,3),V(N,1),V(N,2),V(N,3),F(N)
      WRITE(*,500)M2(N,1,1),M2(N,1,2),M2(N,1,3),M2(N,2,2),M2(N,2,3),M2(N
     #,3,3),V(N,1),V(N,2),V(N,3),F(N)
 100  CONTINUE
      WRITE(2,*)'           **********'
C*******************
C
C *** CALCULO DE LOS MOMENTOS PARCIALES
C
C *** A PARTIR DE ESTE MOMENTO YA SOLO HAY DOS POBLACIONES
C
      NP=2
C
      CALL MOMPAR(MM2,C,DD,FP,Q,VV,M2,V,B2,W)
C
C*******************
      WRITE(*,*)' '
      WRITE(*,510)
      DO 200 N=1,NP
      WRITE(2,500)M2(N,1,1),M2(N,1,2),M2(N,1,3),M2(N,2,2),M2(N,2,3),M2(N
     #,3,3),V(N,1),V(N,2),V(N,3),FP(N)
      WRITE(*,500)M2(N,1,1),M2(N,1,2),M2(N,1,3),M2(N,2,2),M2(N,2,3),M2(N
     #,3,3),V(N,1),V(N,2),V(N,3),FP(N)
 200  CONTINUE
 500  FORMAT(1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,F7.1,1X,
     #F7.1,1X,F7.1,1X,F4.2)
 510  FORMAT(1X,'M2(1,1)',1X,'M2(1,2)',1X,'M2(1,3)',1X,'M2(2,2)',1X,'M2(
     #2,3)',1X,'M2(3,3)',3X,'V(1)',4X,'V(2)',4X,'V(3)',4X,'F')
C*******************
C
C *** SE ENTRA EN EL BUCLE DE RECALCULO
C
      WRITE(*,*)'********** DESEA ESTIMAR EL ERROR DE D3 ? (1=SI)'
      READ(*,*)NSI
      IF(NSI.EQ.1)THEN
                  F(1)=FP(1)
```

```
                    F(2)=FP(2)
                    GOTO 9999
      ENDIF
C
      STOP
      END
```

```
C234567
C *** EN ESTA APLICACION SE GENERAN MUESTRAS ALEATORIAS DE ESTRELLAS
C *** A CONTINUACION SE REALIZAN UNAS ROTACIONES DE LAS COMPONENTES
C *** DE LA VELOCIDAD DE LAS MUESTRAS Y SE CALCULAN LOS MOMENTOS
C *** ESTA PARTE  DE LA APLICACION SOLO EFECTUA LA DESN(0,1)ZACION
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*64 FICMUE,FICGIR,FICMOM
      DIMENSION FICMUE(10),FICGIR(10),FICMOM(10)
      DIMENSION SIGMB(3),VALMEB(3),ROTB(2),SIGMA(10,3),VALMED(10,3),
     #ROTA(10,2)
C *** SE DEFINEN TODOS LOS FICHEROS A UTILIZAR
C *** FICHEROS QUE CONTIENEN LAS MUESTRAS
      FICMUE(1)='MOSTRA01.DAT'
      FICMUE(2)='MOSTRA02.DAT'
      FICMUE(3)='MOSTRA03.DAT'
      FICMUE(4)='MOSTRA04.DAT'
      FICMUE(5)='MOSTRA05.DAT'
      FICMUE(6)='MOSTRA06.DAT'
      FICMUE(7)='MOSTRA07.DAT'
      FICMUE(8)='MOSTRA08.DAT'
      FICMUE(9)='MOSTRA09.DAT'
      FICMUE(10)='MOSTRA10.DAT'
C *** MUESTRAS   GIRADAS
      FICGIR(1)='MOSGIR01.DAT'
      FICGIR(2)='MOSGIR02.DAT'
      FICGIR(3)='MOSGIR03.DAT'
      FICGIR(4)='MOSGIR04.DAT'
      FICGIR(5)='MOSGIR05.DAT'
      FICGIR(6)='MOSGIR06.DAT'
      FICGIR(7)='MOSGIR07.DAT'
      FICGIR(8)='MOSGIR08.DAT'
      FICGIR(9)='MOSGIR09.DAT'
      FICGIR(10)='MOSGIR10.DAT'
C *** FICHEROS DE MOMENTOS
      FICMOM(1)='MOMMOS01.DAT'
      FICMOM(2)='MOMMOS02.DAT'
      FICMOM(3)='MOMMOS03.DAT'
      FICMOM(4)='MOMMOS04.DAT'
      FICMOM(5)='MOMMOS05.DAT'
      FICMOM(6)='MOMMOS06.DAT'
      FICMOM(7)='MOMMOS07.DAT'
      FICMOM(8)='MOMMOS08.DAT'
      FICMOM(9)='MOMMOS09.DAT'
      FICMOM(10)='MOMMOS10.DAT'
C
C *** SE INDICA CUANTAS MUESTRAS SE VAN A UTILIZAR
      WRITE(*,*)'INDIQUE NUMERO DE MUESTRAS'
      READ(*,*)NMUE
      OPEN(1,FILE='SIGMA.DAT',STATUS='OLD')
      OPEN(2,FILE='MEDIA.DAT',STATUS='OLD')
      OPEN(3,FILE='ROTACION.DAT',STATUS='OLD')
      DO 1100 N=1,NMUE
      READ(1,500)SIGMA(N,1),SIGMA(N,2),SIGMA(N,3)
      READ(2,500)VALMED(N,1),VALMED(N,2),VALMED(N,3)
      READ(3,510)ROTA(N,1),ROTA(N,2)
```

```
 1100 CONTINUE
C
      DO 1000 N=1,NMUE
      DO 1010 I=1,3
      SIGMB(I)=SIGMA(N,I)
      VALMEB(I)=VALMED(N,I)
 1010 CONTINUE
      DO 1020 I=1,2
      ROTB(I)=ROTA(N,I)
 1020 CONTINUE
      CALL GIRMUE(FICMUE(N),FICGIR(N),SIGMB,VALMEB,ROTB)
 1000 CONTINUE
 500  FORMAT(F5.1,1X,F5.1,1X,F5.1)
 510  FORMAT(F5.1,1X,F5.1)
C
      STOP
      END
```

```
C234567
C *** EN ESTA APLICACION SE GENERAN MUESTRAS ALEATORIAS DE ESTRELLAS
C *** A CONTINUACION SE REALIZAN UNAS ROTACIONES DE LAS COMPONENTES
C *** DE LA VELOCIDAD DE LAS MUESTRAS Y SE CALCULAN LOS MOMENTOS
C *** ESTA PARTE SOLO REALIZA EL CALCULO DE MOMENTOS
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*64 FICMUE,FICGIR,FICMOM
      DIMENSION FICMUE(10),FICGIR(10),FICMOM(10)
C *** SE DEFINEN TODOS LOS FICHEROS A UTILIZAR
C *** FICHEROS QUE CONTIENEN LAS MUESTRAS
      FICMUE(1)='MOSTRA01.DAT'
      FICMUE(2)='MOSTRA02.DAT'
      FICMUE(3)='MOSTRA03.DAT'
      FICMUE(4)='MOSTRA04.DAT'
      FICMUE(5)='MOSTRA05.DAT'
      FICMUE(6)='MOSTRA06.DAT'
      FICMUE(7)='MOSTRA07.DAT'
      FICMUE(8)='MOSTRA08.DAT'
      FICMUE(9)='MOSTRA09.DAT'
      FICMUE(10)='MOSTRA10.DAT'
C *** MUESTRAS   GIRADAS
      FICGIR(1)='MOSGIR01.DAT'
      FICGIR(2)='MOSGIR02.DAT'
      FICGIR(3)='MOSGIR03.DAT'
      FICGIR(4)='MOSGIR04.DAT'
      FICGIR(5)='MOSGIR05.DAT'
      FICGIR(6)='MOSGIR06.DAT'
      FICGIR(7)='MOSGIR07.DAT'
      FICGIR(8)='MOSGIR08.DAT'
      FICGIR(9)='MOSGIR09.DAT'
      FICGIR(10)='MOSGIR10.DAT'
C *** FICHEROS DE MOMENTOS
      FICMOM(1)='MOMMOS01.DAT'
      FICMOM(2)='MOMMOS02.DAT'
      FICMOM(3)='MOMMOS03.DAT'
      FICMOM(4)='MOMMOS04.DAT'
      FICMOM(5)='MOMMOS05.DAT'
      FICMOM(6)='MOMMOS06.DAT'
      FICMOM(7)='MOMMOS07.DAT'
      FICMOM(8)='MOMMOS08.DAT'
      FICMOM(9)='MOMMOS09.DAT'
      FICMOM(10)='MOMMOS10.DAT'
C
C *** SE INDICA CUANTAS MUESTRAS SE VAN A UTILIZAR
      WRITE(*,*)'INDIQUE NUMERO DE MUESTRAS'
      READ(*,*)NMUE
C *** SE CALCULAN LOS MOMENTOS DE LAS MUESTRAS
C *** APERTURA DEL FICHERO DE DATOS Y LECTURA DEL MISMO
      DO 2000 N=1,NMUE
      CALL CALCMO(FICGIR(N),FICMOM(N))
 2000 CONTINUE
C
      STOP
      END
```