

Observatorio Astronómico Nacional

CENTRO ASTRONÓMICO DE YEBES
Informe Técnico 1991.5

**Adaptación de los catálogos estelares PPM
al sistema de reducción de placas
astrométricas desarrollado en el CAY**

Pere Planesas

CENTRO ASTRONOMICO DE YEBES
Informe Técnico 1991.5

**Adaptación de los catálogos estelares PPM
al sistema de reducción de placas
astrométricas desarrollado en el CAY**

Pere Planesas

En este trabajo se describe la adaptación de los nuevos catálogos estelares PPM-Norte y PPM-Sur para su uso en el sistema de reducción de placas fotográficas con aplicaciones a Astrometría desarrollado en el Centro Astronómico de Yebes.

1. Introducción

Desde el inicio de esta década se está tendiendo a usar cada vez más el catálogo PPM (*Positions and Proper Motions*, de Röser y Bastian) como referencia para la determinación de posiciones precisas de asteroides y, en general, en trabajos de Astrometría. Ello se debe a la alta precisión de sus coordenadas, que supera a la de catálogos usados hasta ahora (como son el SAO y el AGK3), la de los movimientos propios (al menos para el hemisferio Norte, donde se dispone de datos durante mayor tiempo) y el hecho de usar las coordenadas referidas a la época J2000.0 y el sistema de referencia FK5. El catálogo de coordenadas estelares PPM esta dividido, por el momento, en dos ficheros. El primero de ellos (PPM-North) en su versión 1989 contiene las coordenadas de 181.731 estrellas con declinación mayor de -2.5° . Una versión preliminar del PPM-South ha aparecido en 1990, con 144.787 estrellas que completan el recubrimiento del cielo. Esta parte del catálogo va a ser completada y mejorada en los años siguientes.

Para información sobre dichos catálogos, se puede consultar la información adjunta a las cintas magnéticas o bien las comunicaciones presentadas por Röser y Bastian en el Symposium número 141 de la IAU (Leningrado, 1989).

2. Compresión de los catálogos PPM

Tal como se hizo anteriormente con los catálogos SAO y AGK3 (véase los Informes Técnicos del CAY 1986-4 y 1988-6), se ha procedido a la compresión de los dos catálogos PPM, a un formato compacto. Ello se hace por varios motivos:

- 1 para reducir considerablemente su tamaño, a fin de poder disponer de ellos en ordenadores personales sin que ocupen la mayor parte del disco;
- 2 tal reducción en tamaño lleva aparejada un aumento en la rapidez de consulta del catálogo, proceso siempre lento en un ordenador personal;
- 3 eliminación de diversas estrellas no adecuadas para la Astrometría que se lleva a cabo en el CAY, por ejemplo las estrellas múltiples o aquellas cuya declinación es menor de -60° .

Se ha partido de la versión 1989 del catálogo PPM-North y la versión 1990 del PPM-South, suministrados por el Centre de Données Stellaires de Strasbourg (Francia). Tales ficheros ocupan, respectivamente, 23.1 y 18.4 Mbyt, estructurados en registros de 133 bytes.

Las versiones obtenidas por nosotros, tras la compresión, ocupan 2.26 y 1.52 Mbyt respectivamente, lo cual corresponde a una reducción en un factor diez es espacio sobre disco. La compresión de los dos ficheros se ha llevado a cabo por separado por dos razones:

- 1 una banda de declinaciones (de 0° a -10°) está parcialmente compartida por las dos porciones del catálogo;
- 2 el PPM-South es una versión preliminar, que en su momento va a ser sustituida por otra mejor.

Los datos que se almacenan para cada estrella son:

- 1 número de identificación en el catálogo. En el caso de PPM-South estos números empiezan en el 500.001, a fin de no inducir a confusión cuando aparezca su nueva versión.
- 2 Las coordenadas ecuatoriales para la nueva época J2000.0.
3. Los movimientos propios correspondientes, cuando se conocen. En el caso del PPM-South se conocen pocos.
- 4 La magnitud fotográfica extraída del AGK3, para el hemisferio Norte, y la magnitud visual extraída del FOKAT, para el hemisferio Sur.
- 5 El tipo espectral (id. id.).
- 6 Bandera que indica la variabilidad.
- 7 Bandera que indica la multiplicidad.

El fichero resultante es de acceso directo, con registros de 1 kbyt (512 enteros de 2 bytes). Los doce primeros enteros constituyen la cabecera del buffer y los siguientes 498 contienen información relativa a 83 estrellas como máximo, a razón de 6 enteros por estrella. La estructura, tanto de la cabecera de cada registro como de la información relativa a cada estrella es idéntica a lo descrito para el catálogo SAO en el Informe Técnico del CAY 1988-6. Por ello no se van a repetir aquí.

Hay que resaltar dos diferencias. Una de ellas implica una ligera variación en la subrutina de decodificación que se utilice para leer el catálogo: hay que restar $10''$ a las coordenadas de cada estrella. La otra implica el cambio de la subrutina de decodificación de los tipos espectrales por dos subrutinas, una para el hemisferio Norte y otra para el Sur.

Los ficheros PPMNORTE.CMP y PPMSUR.CMP contienen, además de la información sobre las estrellas, un primer registro en que se detalla en qué registro se inicia cada bloque de información relativa a una banda de declinación (de 10°) y a cada hora en Ascensión Recta. Ello permite realizar búsquedas mucho más rápidas. En particular, para la consulta relativa a una placa astrométrica, posiblemente sólo

haya que hacer un salto en el disco.

3. Programas usados para realizar la compresión de los catálogos PPM

COMPPM: Realiza la compresión de cualquiera de los ficheros PPM tras ser copiados en disco mediante, p.ej., TAPECOPY a partir de las cintas magnéticas originales. Las versiones con 133 bytes por registro han sido leídas y comprimidas satisfactoriamente mediante este programa. No ha sido comprobado el buen funcionamiento de la lectura de las versiones de 138 bytes/registro (correspondientes a una versión preliminar del catálogo que circuló en 1988 y que ahora se considera obsoleta).

PONERPPM: Lee uno de los ficheros generados por COMPPM a fin de construir el registro cabecera y lo graba en el primer registro del futuro catálogo comprimido.

Los listados de estos dos programas se encuentran al final de este informe. Otras informaciones generadas por COMPPM se encuentran almacenadas en el fichero COMPRESION.INFO. En particular, constan los movimientos propios máximos en cada coordenada ($0''7285$ en A.R. y $3''258$ en declinación).

4. Subrutinas para el acceso a los catálogos PPM

Son el mismo conjunto de subrutinas que se utilizaron para los catálogos SAO o AGK3, con las salvedades citadas antes relativas al tipo espectral (subrutinas DECODIF_N y DECODIF_S) y a una corrección en la ascensión recta (subrutina DEKOD). Las subrutinas que referencia el usuario son: LEERPPM (que utiliza BUSCA y CABE) y DEKOD (que utiliza DECODIF_N y DECODIF_S). Tales subrutinas se encuentran en el fichero LEEPPM.FOR, que contiene el programa utilizado para comprobar el contenido de los catálogos PPMNORTE y PPMSUR. Este programa puede asimismo ser usado para la consulta de tales ficheros en el VAX (ver Sección 5).

El listado del programa LEEPPM y sus subrutinas se encuentra al final de este informe.

5. Uso de los catálogos PPM en los VAXes del CAY

El programa LEEPPM se encuentra en [ALMACEN.UTIL] (subdirectorío que en estos momentos se encuentra en el \$DISK3:). Los catálogos PPMNORTE.CMP y PPMSUR.CMP se encuentran en cinta la magnética YEBES-170 (armario del despacho de Pere), en el 'save-set' denominado PPM.BCK. Para copiarlos sobre disco (en un área grande, como \$SELVA, pues se necesitan casi 7.800 bloques) basta con efectuar:

```
assign mub0: tape      (en el nodo ISIS)
mount/foreign tape:
back/log/label=pere tape:ppm.bck * * /select=(*.cmp)/
dismount tape:
```

Al ejecutar el programa LEEPPM, se obtienen listados simultáneos en pantalla y sobre disco (fichero FOR004.DAT).

6. Transferencia de los ficheros del VAX al ordenador personal

7. Listados de los programas y subrutinas referenciados

Aparecen por el siguiente orden:

Programa *COMPPM* con las subrutinas CODTS, WRII y otras auxiliares.

Programa *PONERPPM*

Programa *LEEPPM* con las subrutinas LEERPPM, BUSCA, CABE, DEKOD, DECODIF_N y DECODIF_S.

PROGRAMA COMPPM

* Programa COMPPM

* Junio 1991, Pere Planesas, Centro Astronomico de Yebes

* Programa para la compactificacion del catalogo PPM.

* Se carga mediante el comando: fc comppm

```

character filen*20, flags(5)*1, nts*2
dimension ibuf(512)
integer*2 jts, ts, mar, mdec, magn
integer*4 jars, jdec, nrec, fin
logical first, flag, full, seacabo, doble, mala
real*8 arcabe, ar, dec, pi, iar, idec, cf, num
PI=dacos(-1d0)

type *, 'QUE FORMATO QUIERES USAR?'
TYPE *, '1: nuevo (133 char/rec)' ; TYPE *, '2: viejo (138 char/rec)'
accept *, iform
if(iform.eq.1.or.iform.eq.2) then ; iform=iform*10 ; else ; stop ; endif

TYPE *, 'QUE FICHERO QUIERES PROCESAR?'
TYPE *, '1: PPMnorth' ; type *, '2: PPMsouth' ; accept *, ifich
ic=iform+ifich
if(ic.eq.11) then ; filen='ppmnorth.133'
      else if (ic.eq.12) then ; filen='ppmsouth.133'
      else if (ic.eq.21) then ; filen='ppmnorth.138'
      else if (ic.eq.22) then ; filen='ppmsouth.138'
      else ; stop ; endif
TYPE *, 'FICHERO DE DATOS: ', FILEN

OPEN (66, FILE=filen, STATUS='OLD', READONLY)
OPEN (77, FILE='COMPACTO.PPM', STATUS='NEW',
1   ACCESS='DIRECT', RECL=256)
OPEN (99, FILE='CABEcera.PPM', STATUS='NEW')

SEACABO=.FALSE. ; FIRST=.TRUE. ; FULL=.FALSE.
FLAG=.FALSE.   ; FIN=.FALSE.
NREC=2         ; NSTAR=0       ; NTOTSTAR=0

*-----
1   if(ic.lt.20) then

      READ(66, 200, END=999) numppm, Fmag, ts, arh, arm, ars, jsigndec,
1       decg, decm, decs, pma, pmd, flags
200  format (1x, i6, t20, f4.1, t25, a2, t28, f2.0, 1x, f2.0, 1x, f6.3, t42, a1,
1     f2.0, 1x, f2.0, 1x, f5.2, t56, f7.4, t64, f6.3, t127, 5a1)
      ar=arh+(arm+ars/6d1)/6d1
      dec=decg+(decm+decs/6d1)/6d1 ; if(jsigndec.eq.'-') dec=-dec

      else !----- ! NO TESTEADO

      read(66, 300, end=999) numppm, fmag, ts, flagS(1), ar, dec, pma, pmd
300  format
1     (4x, i6, f6.1, 2x, a2, 1x, a1, 1x, d15.8, 1x, d15.8, 1x, d12.5, 1x, d12.5)
      ar=ar/pi*12d0 ; dec=dec/pi*18d1

```

```
PMA=PMA/PI*12D0*36D0 ; PMD=PMD/PI*18D1*36D0
endif
```

```
*-----
if(abs(pma).gt.pmamax) pmamax=pma
if(abs(pmd).gt.pmdmax) pmdmax=pmd
if(numppm.lt.numant) goto 999 ; numant=numppm
```

```
* paso a formato SAO, para aprovechar programas y subrutinas
```

```
IAR=CF(AR)*1D7 ; IDEC=CF(DEC)*1D6
MAR=PMA*1E4 ; MDEC=PMD*1E3
NUM=NUMPPM
IF(FLAGS(2).EQ.'D') THEN ; DOBLE=.TRUE.
ELSE ; DOBLE=.FALSE. ; ENDIF
IF(FLAGS(1).EQ.'P'.OR.FLAGS(1).EQ.'C') THEN ; MALA=.TRUE.
ELSE ; MALA=.FALSE. ; ENDIF
FLAGS(1)=' ' ; FLAGS(2)=' '
MAGN=10*FMAG
IF(MAGN.GT.110) THEN ; MAGN=126 ; ELSE ; MAGN=MAGN+15 ; ENDIF
IF(MAGN.EQ.0) MAGN=127
```

```
CALL WRII (IAR, IDEC, MAR, MDEC, NUM, DOBLE, MALA, MAGN, TS)
```

```
GOTO 1
```

```
* se acabo
```

```
999 FULL=.TRUE. ; FIN=.TRUE. ; SEACABO=.TRUE. ; IAR=-999999.
CALL WRII (IAR, IDEC, MAR, MDEC, NUM, DOBLE, MALA, MAGN, TS)
```

```
99 CLOSE (88) ; CLOSE (77) ; CLOSE (99)
JTS=2HEN ; CALL CODTS(JTS)
write(55,*) filen
write(55,*) 'pmamax,pmdmax=', pmamax, pmdmax
write(55,*) 'numppm=', numant
END
```

```
*****
```

```
SUBROUTINE CODTS (I)
```

```
INTEGER*2 I,II, CODE(256) ; DATA N/0/
```

```
IF(I.EQ.0) RETURN
```

```
IF(I.EQ.2HEN) THEN ! fin
```

```
OPEN(99, FILE='CODIGOS.TS', STATUS='NEW')
```

```
DO J=1, N ; WRITE(99, '(1X, A2)') CODE(J) ; ENDDO ; RETURN
```

```
ENDIF
```

```
IF(N.EQ.0) THEN ! primer codigo
```

```
CODE(1)=I ; I=1 ; N=1 ; RETURN ; ENDIF
```

```
II=I ; DO I=1, N ; IF(II.EQ.CODE(I)) RETURN ; ENDDO ; N=N+1
```

```
if(n.gt.256) then
```

```
write(*,*) ' excedido el numero maximo de TS' ; return ; endif
```

```
CODE(N)=II ; WRITE(*,1) II, N ; I=N
```



```
1      FORMAT(' TS = ',A2,I5)
      END
```

```
*****
```

```
      SUBROUTINE WR11 (IAR, IDEC, MAR, MDEC, NUM, DOBLE, MALA, MAGN, TS)
```

```
      IMPLICIT NONE
```

```
      REAL*8 NUM, IAR, IDEC
```

```
      REAL*8 KNUM, KIAR, II, JJ, D20, D19, NSTAR, CERO, RNUM
```

```
      INTEGER*2 MAR, MDEC, MAGN, TS
```

```
      LOGICAL DOBLE, MALA, INIC
```

```
      INTEGER*2 IBUF(512) , jjj
```

```
      INTEGER*2 ISHIF, KONT, INEW, IOLD, I, J, N, IOR, IAND, ISIGN, NREC, III
```

```
      INTEGER*2 JDEC, JAR, KAR
```

```
      DATA CERO /0.D0/ , INIC /.TRUE./
```

```
      IF(IAR.EQ.-999999.) GOTO 4
```

```
      IF(INIC) THEN
```

```
        INIC=.FALSE. ; D20=1024D0**2 ; D19=D20/2D0
```

```
        OPEN (77, FILE='COMPACTO.SAO', STATUS='NEW', ACCESS='DIRECT'
```

```
*          , RECL=256, BLOCKSIZE=256, FORM='UNFORMATTED')
```

```
*          1      RNUM=CERO
```

```
        WRITE(77, REC=1) IBUF ; NREC=2
```

```
        N=0 ; IOLD=-1 ; NSTAR=0
```

```
        1      OPEN(88, FILE='COMPACTO.LIS', STATUS='NEW',
```

```
        ACCESS='SEQUENTIAL')
```

```
        WRITE (88,*) ' version 910327'
```

```
      ENDIF
```

```
1      RNUM=RNUM+1D0 ; IF(NUM.EQ.CERO) NUM=RNUM
```

```
10     IF(N.LE.0) THEN
```

```
      JDEC=IDEC/1D7 ! declinacion en decenas de grados
```

```
      IF(JDEC.LT.-5) GOTO 99 ! decl. limite -50 grados
```

```
      IF(IDEC.LT.0) THEN ; ISIGN='100000'0 ; JDEC=IABS(JDEC)
```

```
      ELSE ; ISIGN=0 ; ENDIF
```

```
      KNUM=NUM ; KIAR=IAR ; CALL CONV (KIAR)
```

```
      CALL CONV (JJ)
```

```
      JAR=IAR/1D7 ! asc. recta en horas
```

```
      KAR=JAR ; IBUF(1)=JDEC*256+JAR ; IBUF(1)=IOR(ISIGN, IBUF(1))
```

```
      IF(IOLD.EQ.-1) IOLD=IBUF(1)
```

```
      INEW=IBUF(1) ; IBUF(2)=NUM/128
```

```
      IBUF(3)=DMOD(NUM, 128D0)*256
```

```
      II=DMOD(IAR, 1D7) ; CALL CONV (II) ; CALL CONV (JJ)
```

```
      IBUF(4)=II/1D3 ; IBUF(5)=DMOD(II, 1D3)
```

```
      DO I=6, 12 ; IBUF(I)=0 ; ENDDO
```

```
      N=13
```

```
      kiar=dmax1(kiar-1d4, 0d0) ! 27 mar 91
```

```
      ENDIF
```

```
2      IF(TS.EQ.2HDE) THEN
```

```
        WRITE(88,*) ' estrella ', NUM, ' borrada'
```

```

DO I=0,4 ; IBUF(N+I)=0 ; ENDDO
IBUF(N+5)=168 ! codigo de estrella borrada
GOTO 30

ENDIF
JAR=IAR/1D7 ; KONT=-1 ; IF(JAR.NE.KAR) GOTO 3
KONT=0 ; I=NUM-KNUM ; IBUF(N)=ISHIF(I,8)
CALL CONV (IAR) ; II=IAR-KIAR ; IBUF(6)=II/1D3 ; IBUF(7)=DMOD(II,1D3)
I=II/16384D0 ; IBUF(N)=IOR(IBUF(N),I) ; I=DMOD(II,16384D0)
IBUF(N+1)=ISHIF(I,2) ; IF(IDECLT.CERO) IDEC=-IDEC
JJ=DMOD(IDECLT,1D7)
CALL CONV (JJ) ; J=JJ/D20 ; IBUF(N+1)=IOR(IBUF(N+1),J)
II=DMOD(JJ,D19) ; IBUF(N+2)=II/16D0 ; JJ=DMOD(JJ,D20)
IF(JJ.GT.D19) IBUF(N+2)=IOR(IBUF(N+2),'100000'O)
I=DMOD(II,16D0) ; CALL SGN (MAR,ISIGN)
IBUF(N+3)=IOR(ISHIF(I,12),MAR/4)
IF(ISIGN) 21,22,22
21 IBUF(N+3)=IOR(IBUF(N+3),'4000'O)
22 MAR=MOD(MAR,4) ; CALL SGN (MDEC,ISIGN)
IBUF(N+4)=IOR(ISHIF(MAR,14),MDEC)
IF(ISIGN) 23,24,24
23 IBUF(N+4)=IOR(IBUF(N+4),'20000'O)
24 IBUF(3)=IBUF(3)+1
IF(DOUBLE) IBUF(N)=IOR(IBUF(N),'100000'O)
IBUF(N+5)=ISHIF(MAGN,8)
CALL CODTS (TS)
IBUF(N+5)=IOR(IBUF(N+5),TS)
IF(MALA) IBUF(N+5)=IOR(IBUF(N+5),'100000'O)
30 N=N+6 ; IF(N.LE.507) RETURN

3 WRITE(77,REC=NREC) IBUF
N=(N-12)/6
write(99,*) NREC,KAR,JDEC,N,KNUM+N-1
IF(INEW.NE.IOLD) THEN
    III=IAND(IOLD,'77777'O)
    WRITE(88,601) IOLD,III/256,MOD(III,256),NSTAR
601 FORMAT(1X,O6,2I5,F9.0)
    IF(IAND(IOLD,'177400'O).NE.IAND(INEW,'177400'O))
1 WRITE(88,*) NREC
    IOLD=INEW ; NSTAR=0
ENDIF
NSTAR=NSTAR+N
DO I=1,512 ; IBUF(I)=0 ; ENDDO
NREC=NREC+1 ; N=0 ; IF(KONT.EQ.-1) GOTO 10
RETURN
99 IAR=CERO
4 IF(N.NE.0) WRITE(77,REC=NREC) IBUF
5 CLOSE(77) ; CLOSE(88) ; END

```

```

FUNCTION ISHIF (IDAT,ICORR)
IMPLICIT NONE ; INTEGER*2 ISHIF, IDAT, ICORR, I, J, N,IOR,IAND
I=IDAT ; N=ICORR ; IF(N) 10,30,20

```

```

10     N=IABS(N) ; IF(I) 11,30,13
11     I=IAND(I,'77777'O) ; I=I/2 ; I=IOR(I,'40000'O) ; N=N-1 ; IF(N) 30,30,13
13     I=I/2**N ; GOTO 30
20     IF(I) 22,30,22
22     DO 25 J=1,N
           I=IAND(I,'77777'O)
           IF(IAND(I,'40000'O)-'40000'O) 24,23,24
23     I=IAND(I,'37777'O)*2 ; I=IOR(I,'100000'O) ; GOTO 25
24     I=I*2
25     ENDDO
30     ISHIF=I ; END

```

```

SUBROUTINE SGN (M,ISIGN)
IMPLICIT NONE ; INTEGER*2 ISIGN,M
IF(M.LT.0) THEN ; ISIGN=-1 ; M=IABS(M) ; ELSE ; ISIGN=1 ; ENDIF ; END

```

```

REAL*8 FUNCTION CF(A) ; REAL*8 A,B,C
I1=A ; B=(A-I1)*60 ; I2=B ; C=(B-I2)*60 ; CF=I1+(I2+C/1D2)/1D2 ; END

```

```

SUBROUTINE CONV (D)
IMPLICIT NONE ; REAL*8 D,A ; INTEGER*2 I,J,N ; DATA N /0/

```

```

IF(N.NE.1) THEN ; A=DMOD(D,1D5) ; I=D/1D5 ; D=I*6D4+A
           ELSE ; A=DMOD(D,1D4) ; I=D/1D4 ; J=I/100
           I=MOD(I,100) ; D=((J*60)+I)*6D3+A ; ENDIF
N=MOD(N+1,2) ; END

```

PROGRAMA PONERPPM

* Programa PONERPPM

* Pere Planesas, Centro Astronomico de Yebes, junio 1991
* Sirve para rellenar el registro directorio de un catalogo
* PPM comprimido mediante COMPPM.FTN.

```
character filen*30,ns*2
integer*2 nrec(240),cabecera(512) ; equivalence (cabecera,nrec)
logical openf ; if(openf(11,'s.lis','or')) stop 'file not accesible'

*nombre del fichero comprimido
filen='ppmsur.cmp'      ! 'ppmnorte.cmp'
*banda de declinacion inicial
ns='S'                  ! 'N'
*numero de banjas de declinacion (de 10 grados)
ndec=6                  ! 10

if(ns.eq.'N') then ; ini=0 ; else ; ini=24*9 ; endif
nmax=ndec*24+ini ; k=ini

1  k=k+1 ; if(k.gt.nmax) goto 9
   read(11,*,end=9) nrec(k),ii,jj,m
2  read(11,*,end=9) n,i,jj,m ; if(m.ne.83) goto 1
   if(i.ne.ii) then ; k=k+1 ; nrec(k)=n ; ii=i ; endif
   goto 2
9  cabecera(512)='123456'0 ; type *,k,ii,jj,m

open(77,file=filen,status='OLD',access='DIRECT',recl=256)
type *,filen,' successfully opened'
write(77'1) cabecera ; close(77)
end
```

PROGRAMA LEEPPM

*Programa LEEPPM

*Pere Planesas, Centro Astronomico de Yebes, junio 1991

*Permite la lectura de uno de los ficheros PPM comprimidos.

*La salida es en pantalla y en FOR002.DAT

```
INTEGER*2 IBUF(512),NSTAR,CABECERA(512)
REAL*8 NUM(83),ALFA(83),DELTA(83),ar,dec
REAL*4 MOVA(83),MOVD(83),MAGN(83)
CHARACTER TS(83)*3,NS*1,LIN*1,FILEN*30
LOGICAL DOBLE(83),VARIABLE(83),FIRST
REAL*8 D
FIRST=.TRUE.

145  TYPE *,'nombre del fichero?'
      read(*,'(a)',end=99) filen

      TYPE *,'Norte o Sur?'
      read(*,'(a)',end=99) ns
      call str$upcase(ns,ns)

      OPEN(88,FILE=FILEN,STATUS='OLD',err=145,
1      ACCESS='DIRECT',RECL=256,BLOCKSIZE=256,readonly)
      READ(88'1) CABECERA
      LDISC=88

1  WRITE(*,*) ' Para acabar pulsar ^Z'
   PRINT *, ' ar,dec ?'
   READ (*,*,END=99) Ar,dec
   CALL LEERppm (AR,DEC,IBUF,CABECERA,NREG,MENOS10,LDISC)

   IF(FIRST) THEN
       WRITE(*,*) 'Encontraras el resultado en FOR002.DAT'
       FIRST=.FALSE.
   ENDIF

   CALL DEKOD (IBUF,NSTAR,NUM,ALFA,DELTA,MOVA,MOVD,TS,MAGN,
1           DOBLE,VARIABLE,NS) !!!! NS es una modificacion

   WRITE(4,*) 'num      ar              dec          mov ar  mov de',
1           ' tip Do Va magn'

   DO I=1,NSTAR
       I1=ALFA(I)
       A=(ALFA(I)-I1)*60
       I2=A
       A=(A-I2)*60
       NN=NUM(I)
       D=ABS(DELTA(I))
       J1=D
       D=(D-J1)*60
       J2=D
       DD=(D-J2)*60
       IF(DELTA(I).LT.0D0) J1=-J1
```

```

WRITE (4,33)
1      NN,I1,I2,A,J1,J2,DD,MOVA(I),MOVD(I),TS(I),
1      DOBLE(I),VARIABLE(I),MAGN(I)
WRITE (*,33)
1      NN,I1,I2,A,J1,J2,DD,MOVA(I),MOVD(I),TS(I),
1      DOBLE(I),VARIABLE(I),MAGN(I)
IF((I/20)*20.EQ.I) THEN
WRITE(*,'(A,$)') ' Pulsa RETURN para continuar:_'
READ (*,'(A)',END=1) LIN
ENDIF
ENDDO
PRINT *,' registro =',NREG

GOTO 1
99 STOP
33 FORMAT(' ',I6,2I3,F7.3,I4,I3,F6.2,F8.4,F7.3,1X,A3,2I3,F5.1)
END

```

SUBROUTINAS DE LECTURA DE CATALOGO

* Esta subrutina permite leer de COMPACTO.PPM
 * el buffer que contiene una cierta estrella,
 * de ascension recta y declinacion dadas.

SUBROUTINE LEERPPM (A,D,IBUF,CABECERA,NREG,MENOS10,LDISC) !no modif

* Entradas:

* A real*8 ascension recta, en horas y fraccion
 * D real*8 declinacion, en grados y fraccion
 * CABECERA int*2 cabecera del catalogo.
 * Si no ha sido leído previamente,
 * sera leído en esta ocasion.
 * Puede coincidir con IBUF.
 * LDISC int*4 unidad logica correspondiente al catalogo.
 * El catalogo debe estar abierto ya.

* Salidas:

* IBUF int*2 Buffer que contiene hasta 83 estrellas,
 * entre las cuales se encuentra la de interes.
 * Para decodificarlo, utilizar DEKOD.
 * NREG int*4 Numero del buffer leído
 * < 0 en caso de lectura erronea
 * MENOS10 int*4 Numero del primer buffer correspondiente
 * a estrellas con la misma ascension recta
 * (en valor entero) y declinacion menor en
 * 10 grados.

INTEGER*2 IBUF (512) , CABECERA (512)
 INTEGER*4 NREG, MENOS10, LDISC
 REAL*8 A, D

INTEGER*2 IA, ID, DEC, N
 REAL*8 ARI, ARF

IA=A
 ID=(IDINT(D)/10)*10
 IF(D.LT.0) ID=ID-10
 CALL BUSCA (N, IA, ID, CABECERA, LDISC)
 DO K=0, 30
 NREG=N+K
 READ (LDISC'NREG,ERR=99) IBUF
 CALL CABE (IBUF, DEC, ARI, ARF)
 IF (A.LE.ARF) GOTO 1

ENDDO

1 MENOS10=IBUF(8)

 RETURN

99 NREG=-1

 END

* Esta subrutina determina el registro en que se encuentran las
 * primeras estrellas que corresponden a una cierta ascension recta

* y una declinacion.

* Entradas:

* A, D INT*2 asc.recta y declinacion
* IBUF INT*2 buffer auxiliar; solo es leido si no
* contiene el primer registro del fichero
* LDISC INT*4 unidad correspondiente al fichero
* COMPACTO.SAO

* Salida:

* N INT*2 numero del registro

```

SUBROUTINE BUSCA (N,A,D,IBUF,LDISC)                                !no modif
INTEGER*2 IBUF(512),N,A,D
IF (IBUF(512).NE.'123456'O) READ (LDISC'1) IBUF
N=IBUF((80-D)*2.4+A+1)
END
```

* Determina la banda de declinaciones (D) y las ascensiones rectas
* inicial y final correspondientes a un registro leido del catalogo.

```

SUBROUTINE CABE (IBUF,D,AI,AF)                                    !no modif
INTEGER*2 IBUF(512),D,ishft
REAL*8 AI,AF
IBUF1=IBUF(1)
ISIG=IAND (IBUF1,'100000'O)
IBUF1=IAND (IBUF1,'77777'O)
D=ISHFT (IBUF1,-8)*10
IF (ISIG.NE.0) D=-10-D
AI=IBUF (5)/36D5+IBUF (4)/36D2+MOD (IBUF1,256)
AF=IBUF (7)/36D5+IBUF (6)/36D2+AI
END
```

* Decodifica el contenido de un buffer leido del catalogo
* dando el numero de estrellas (NSTAR) y sus parametros de
* interes.

```

                                                                    !modificada
SUBROUTINE DEKOD (IBUF,NSTAR,NUM,ALFA,DELTA,MOVA,MOVD,TS,MAGN,
1                    DOBLE,VARIABLE,NS)   ! NS es una modificacion

IMPLICIT NONE
INTEGER*2 IBUF(512),NSTAR
REAL*8 NUM(83),ALFA(83),DELTA(83)
REAL*4 MOVA(83),MOVD(83),MAGN(83)
CHARACTER TS(83)*3,NS*1
LOGICAL DOBLE(83),VARIABLE(83)

REAL*8 AI,NU,DUM,D20
INTEGER*2 IJK,ICLS,ISHFT,IBUF1,ISIG,D,I,J1,J2,J3,N

ICLS(IJK)=IAND(IJK,'77777'O)
D20=1024D0**2
```



```

NSTAR=MOD (IBUF (3), 128)
NU=IBUF (2)*128D0+IBUF (3)/256
IBUF1=IBUF (1)
ISIG=IAND (IBUF1, '100000'O)
IBUF1=IAND (IBUF1, '77777'O)
D=ISHFT (IBUF1, -8)*10
AI=FLOAT (IBUF (5))/36D5+FLOAT (IBUF (4))/36D2+MOD (IBUF1, 256)
AI=ai-1d4/36d5 ! especial para PPM, marzo 1991

N=13
DO I=1, NSTAR
  NUM (I)=NU+ISHFT (IAND (IBUF (N), '77777'O), -8)
  J1=IAND (IBUF (N+5), '377'O)
  if (ns.eq.'N') CALL DECODIF_N (TS (I), J1)
  if (ns.eq.'S') CALL DECODIF_S (TS (I), J1)
  IF (TS (I).EQ.'DEL') GOTO 1
  IF (IAND (IBUF (N), '100000'O).NE.0) THEN
    DOBLE (I)=.TRUE.
  ELSE
    DOBLE (I)=.FALSE.
  ENDIF
  IBUF (N)=IAND (IBUF (N), '77777'O)
  DUM=MOD (IBUF (N), 256)*16384D0+ISHFT (IBUF (N+1), -2)
  ALFA (I)=AI+DUM/36D5
  J1=ICLS (IBUF (N+1))
  J2=IBUF (N+2)
  J3=ICLS (J2)
  DUM=MOD (J1, 4)*D20+J3*16D0+ISHFT (IBUF (N+3), -12)
  IF (J2.NE.J3) DUM=DUM+524288D0
  DELTA (I)=D+DUM/36D4
  IF (ISIG.NE.0) DELTA (I)=-DELTA (I)
  J1=ICLS (IBUF (N+3))
  J2=MOD (J1, 4096)
  J3=MOD (J2, 2048)
  MOVA (I)=(J3*4+ISHFT (IBUF (N+4), -14))/1E4
  IF (J3.NE.J2) MOVA (I)=-MOVA (I)
  J1=ICLS (IBUF (N+4))
  J2=MOD (J1, 16384)
  J3=MOD (J2, 8192)
  MOVD (I)=J3/1E3
  IF (J3.NE.J2) MOVD (I)=-MOVD (I)
  IF (IAND (IBUF (N+5), '100000'O).NE.0) THEN
    VARIABLE (I)=.TRUE.
  ELSE
    VARIABLE (I)=.FALSE.
  ENDIF
  J1=ishft (IAND (IBUF (N+5), '77400'O), -8)
  IF (J1.EQ.127) MAGN (I)=30
  IF (J1.EQ.126) MAGN (I)=20
  IF (J1.LE.125) MAGN (I)=(J1-15)*0.1
  N=N+6
ENDDO
END

```

* Decodificación de los tipos espectrales

*

```

SUBROUTINE DECODIF_N(ST,NUM)                ! nueva
IMPLICIT NONE
INTEGER*2 NUM,NMAX
PARAMETER (NMAX=78)
CHARACTER*3 ST, TIPO(NMAX)
DATA TIPO / 'K5 ','G5 ','F5 ','K0 ','A2 ','F0 ','A0 ','F2 ','
1          ','B8 ','K2 ','M1 ','G0 ','A5 ','A3 ','F8 ','G2 ','
2          'K3 ','A ','B9 ','G6 ','M8 ','F ','K ','B5 ','G ','
3          'M0 ','M2 ','M3 ','M ','M4 ','B3 ','M5 ','B2 ','K8 ','
4          'R0 ','R3 ','A4 ','B0 ','M9 ','N4 ','M6 ','B1 ','N1 ','
5          'B ','K7 ','A7 ','N5 ','R2 ','S4 ','R8 ','M7 ','F9 ','
6          'N ','O5 ','O1 ','G5 ','N3 ','R5 ','O ','N2 ','S ','
7          'O4 ','F6 ','K1 ','N6 ','S5 ','N0 ','K6 ','K4 ','A8 ','
8          'O8 ','R ','F7 ','S2 ','O7 ','O0 ','F3 ' /

```

```

ST=' '
IF (NUM.GE.1.AND.NUM.LE.NMAX) ST = TIPO(NUM)
END

```

```

SUBROUTINE DECODIF_S(ST,NUM)                !nueva
IMPLICIT NONE
INTEGER*2 NUM,NMAX
PARAMETER (NMAX=74)
CHARACTER*3 ST, TIPO(NMAX)
DATA TIPO / 'G5 ','K5 ','K2 ','F8 ','A5 ','K0 ','F5 ','
1          'G0 ','A3 ','F2 ','A2 ','F0 ','      ','B8 ','G1 ','
2          'M0 ','B9 ','M5 ','A0 ','M1 ','F7 ','G8 ','M3 ','
3          'M2 ','K1 ','K8 ','G3 ','G2 ','F6 ','G9 ','K3 ','
4          'G6 ','R0 ','R3 ','F9 ','F7 ','A7 ','M4 ','F3 ','
5          'A8 ','N0 ','M8 ','E5 ','B3 ','B0 ','B1 ','M6 ','
6          'B2 ','F1 ','R5 ','K6 ','K4 ','A9 ','G4 ','A4 ','
7          'A1 ','F4 ','K9 ','A6 ','N7 ','B7 ','O9 ','B4 ','
8          'M9 ','N9 ','R4 ','N2 ','S3 ','R7 ','M7 ','G7 ','
9          'O4 ','R2 ','R9 ' /

```

```

ST=' '
IF (NUM.GE.1.AND.NUM.LE.NMAX) ST = TIPO(NUM)
END

```