

Informe Técnico DAY 1762-4.

UTILIZACION DE PROGRAMAS PARA EL TRATAMIENTO DE  
DATOS ESPECTRALES PROCEDENTES DE PICO DE VELETA  
(Junio 1985)

Pere Francesas Rigas

\*\*\*\*\*

INDICE I

\*\*\*\*\*

Trasvase y transformación de los datos espectrales  
procedentes de Pico de Veleta

MTDS	2
PICO	4

Información relativa al análisis de datos proceden-  
tes de Pico de Veleta correspondientes al banco de  
filtros de 512 canales (1 MHz)

ABE	5
EAP	6
EDICA	8
.CC	9

Representación gráfica de los espectros de Pico de  
Veleta correspondiente a Junio de 1985

ANEXO I

Listados de algunos de los programas citados	11
--	----

## TRASVASE Y TRANSFORMACION DE LOS DATOS ESPECTRALES PROCEDENTES DE PICO DE VELETA

En la primera ocasión que hemos realizado observaciones en el Observatorio de Pico de Veleta (IRAM-1GN) los datos espectrales nos han sido entregados en formato FITS, un estándar para observaciones espectrales con interferómetros, en código ASCII. El primer paso antes de proceder al análisis de tales datos en el Centro astronómico de Yepes consiste en su transformación al formato compacto que utilizan nuestros programas (ANAE, AEB), diez veces más denso, con el consiguiente ahorro de memoria.

=====  
 HTDS:  
 =====

El programa de transformación, denominado PICO, trabaja en el ordenador HP-1000, luego ante todo hay que pasar datos de la cinta magnética al disco magnético. Se utilizará el programa PICO ("Magnetic Tape to Disco") que corre en el HP-2100 con el sistema operativo RTE-11 y que fue realizado con propósitos más generales por Luis A. Belgado. El proceso de trasvase exige, en general, parar y apagar ambos ordenadores, conectar la unidad de disco al HP-2100, poner en marcha en éste el sistema operativo RTE-11, efectuar el trasvase, parar este ordenador, apagar ambos y reconfigurar el sistema. Una descripción detallada de todo este proceso puede encontrarse en los apartados 2.1 y 2.3 del Informe Técnico 1984-6.

Antes de proceder a tal trasvase es conveniente limpiar el cartridge número 13: en medio cartridge (4800 bloques) caben solamente 140 espectros en el formato FITS. Requisito inmediato: hay que traer los datos lo más analizados posible de Pico de Veleta a fin de tener que tratar aquí la menor cantidad posible de espectros.

Una vez determinado el tamaño máximo (en bloques) que puede

tener el fichero en disco, éste se crea mediante el comando:

```
CP,VELETA::13:4:tamaño en bloques
```

La razón de denominar VELETA a tal fichero se verá más adelante. Naturalmente, se le puede dar un nombre cualquiera y renombrarlo cuando sea necesario.

La compilación y carga del programa NTDS se lleva a cabo mediante el comando: `*,*NTDS`

Es puesto en marcha automáticamente. Si se va a utilizar de nuevo y ya ha sido cargado antes, bastará dar: `!RU,NTDS`

El programa NTDS pide el movimiento en ficheros y registros que hay que darle a la cinta. Cada espectro ocupa un fichero. Cuando se trate de leer los primeros espectros que hay en la cinta habrá que contestar: 0, 0 a tal cuestión, y a continuación dar:

```
*EO,NTDS
```

Si en una ocasión precedente han sido leídos 100 espectros, habrá que contestar: 100, 0.

El programa pide también el nombre del fichero (ya creado) donde deben colocarse los datos e ira colocandolos en él todos los que quepan, momento en el cual se detendrá, dando mensajes de error sin importancia. Es posible parar este proceso pulsando el bot 15 del switch register del ordenador. En la pantalla aparecerá el número de ficheros (espectros) que van siendo trasladados.

Una vez llevado a cabo este trasvase, si no hay más espacio donde colocar otro fichero grande de datos en formato FITS, se pasa a trabajar con el ordenador HP-1000.

=====  
PICO:  
=====

Este programa funciona en el HP-1000 y su misión es transformar un fichero de espectros de nombre VELETA:13 escrito en formato FITS a un fichero con el formato del CAY.

Según se transforme los datos del backend de 100 kHz o del de 1 MHz, generará un fichero de nombre VELENA:14 o bien VELEMB:14 que luego podrán ser renombrados por el usuario. El primero de ellos es analizable con AEE, el segundo con AEE.

A medida que los datos van siendo analizados, se actualiza una cuenta de los mismos. En el momento de salir de la impresora, a elección del usuario, se puede grabar o grabado automáticamente. Atención a la salida.

Este programa se pone en marcha mediante los comandos:

```
.,CC,PICO
:PICO
```

INFORMACION RELATIVA AL ANALISIS DE DATOS PROCEDENTES DE PICO DE VELETA CORRESPONDIENTES AL BANCO DE FILTROS DE 512 CANALES (1.412)

-----

AL:1

-----

El banco de filtros de anchura 1 MHz consta de 512 canales, luego no se puede proceder al análisis de tales datos mediante el programa ANAE. Es necesario utilizar su versión extendida a 1x512 o 2x256 canales, denominada AEE (análisis espectral extendido). Este programa es prácticamente idéntico a ANAE, por lo que su utilización no debe presentar ningún problema al usuario que conoce la versión estándar. El análisis resulta ligeramente más lento debido a que se trabaja con más datos y que el programa está dividido en un número mayor de segmentos.

En algunos casos puede resultar más cómodo utilizar ANAE, por ser más rápido, por tener más comandos y más actualizados, para comparar los resultados de lo que se está analizando con otros obtenidos previamente en Yebes o con el banco de 100 KHz, ...

=====  
EAN:  
=====

Se puede proceder de distintas maneras a la transformación de los datos correspondientes a AEE (extendidos) al formato de ANAE (normales):

- a) si se dispone de resolución superflua, se puede realizar una compresión y a continuación cercenar la parte sobrante del espectro (que contendrá sólo dummies)
- b) si se dispone de línea de base más que suficiente, se pueden cercenar 128 canales en cada extremo del espectro, con lo que se acaban teniendo 256, que (para una sola vía) son analizables con ANAE
- c) si hay rayas en todo el rango cubierto, se puede tratar de partir el espectro en 2, generando no uno sino dos ficheros de espectros de 256 canales.

Para llevar a cabo lo anterior (salvo la compresión, que hay que realizarla con AEE, generando un fichero auxiliar de espectros comprimidos) se dispone del programa EAN ("Extendido A Normal"). Este programa considera solamente 256 datos, despreciando el resto. Comentaremos aquí sólo las características que nos interesan para el problema que nos ocupa, aunque el programa dispone de otras posibilidades. Su puesta en marcha se lleva a cabo mediante:

EAN, fichero origen, fichero destino:sc:cr, lpar1,par2,lpar3

En nuestro caso sólo nos interesa utilizar el parámetro opcional número 3 (par3). Este parámetro indica cuantos canales del principio del espectro no deben ser considerados. Por defecto, su valor es 0: ello indica que deben ser considerados los 256 primeros valores (caso (a)). Si se desea conservar los 256 valores centrales (caso (b)), habrá que despreciar los 128 primeros, luego habrá que comandar:

EAN, fich. orig. , fich. dest. ,,,128

Para el caso (c) hay que llevar a cabo tal transformación dos

veces:

EAN, fich. orig. , fich. dest. primero , , , corrim. primero  
 EAN, fich. orig. , fich. dest. segundo , , , corrim. segundo

Si no hay ninguna raya cerca del centro, el corrimiento primero puede ser simplemente cero (análogo al caso (a)) y el corrimiento segundo puede ser de 256 canales.

Nota adicional:

En el caso de la opción: EAN, fich. orig., fich. dest., el barrido es conservado a partir del primer valor no nulo. Esto permite, utilizando previamente el comando DIMI de AII, analizar un cierto número de canales (desde el 1 al n), creando un nuevo fichero extendido. A éste habrá que aplicarle EAN.



=====  
EDICA:  
=====

El caso (c) suele corresponder a una situación en que se obtienen varias rayas espectrales al mismo tiempo, correspondientes a distintas transiciones. (La ancha banda que cubre tal banco de filtros (512 MHz) corresponde a delta f / f = 0,005 a las frecuencias a las que hemos trabajado en junio de 1985.) Para poder observarlas simultáneamente se comandó una frecuencia en reposo que no corresponde a ninguna de ellas. Puede ser conveniente editar la cabecera de los espectros a fin de cambiar la frecuencia en reposo, la resolución y el canal en que se encuentra la raya.

Cualquier tipo de cambio en la cabecera, como los antes descritos, puede llevarse a cabo mediante el programa EDICA, que se encuentra en el fichero `&EDICA:xxii5`. Para utilizarlo, hay que editarlo, modificando la zona indicada (por favor, no modificar nada fuera de tal zona), donde se pueden encontrar algunos ejemplos (precedidos de \*\*). En un anexo incluimos un listado explicativo del contenido de la cabecera de un espectro en el formato de AEE ( y de AEE).

Tras la compilación y carga: `cc &EDICA`  
este programa se pone en marcha mediante:

`EDICA, fichero origen , fichero destino:sc:cr`

Como en el caso anterior, el fichero origen no es destruido. Si la transformación realizada es buena, se agradece de antemano su inmediato purgado.

====

,CC:

====

La carga de los programas EAN y EDICA se lleva a cabo mediante el comando ,CC puesto a punto en junio de 1985. Este comando lleva a cabo la compilación, realiza un UF del programa por si hubiera una versión anterior en funcionamiento, monta y carga la nueva versión y purga el objeto relocatable (fichero Mxxxxx). El programa queda listo para su ejecución.

En general este comando se pone en marcha mediante:

```
,CC,Mxxxxx
```

pero si se desea que se opere en el listado de carga o si se va a utilizar alguna librería hay que comandar:

```
,CC,Mxxxxx,LI
```

Cabe el caso de ABAE, que utiliza la librería de procesamiento de vectores:

## REPRESENTACION GRAFICA DE LOS ESPECTROS DE PICO DE VELETA CORRESPONDIENTES A JUNIO 85

Para el dibujo de espectros de Pico de Veleta, de 1 en 1, de 2 en 2 o de 3 en 3, se utilizan los programas: GRSPC y SCALA.

El programa GRSPC (GRáfica de eSPeCtros) permite la grafica de 1, 2 o 3 espectros, superpuestos, con un tamaño a escoger (por defecto hay un tamaño estándar).

Información relativa a la(s) gráfica(s) efectuada(s) es escrita en un fichero de nombre: .SCALA:\*\*:14, fichero editable. Tal edición suele ser necesaria para corregir el nombre de la fuente (RRaQL --> RR Aql).

El programa SCALA (eSCala) dibuja la escala de velocidades, y las escalas verticales (pues al tratarse de graficas con escalas autoajustadas, las escalas respectivas son distintas). La opción: SCALA,Cuadro permite el dibujo de un recuadro alrededor de la figura.

Ambos programas se cargan con el comando CC y la opción LI, por ejemplo: ;CC,&GRSPC,LI

El proceso típico es, pues:

- Crear un fichero con los espectros (1, 2 o 3) a dibujar.
- Colocar una hoja de papel en el plotter.
- Correr el programa GRSPC para tal fichero, escogiendo el rango de velocidades adecuado.
- Editar el fichero .SCALA:\*\*:14 a fin de corregir el nombre de la fuente.
- Correr el programa SCALA o bien SCALA,CU

=====  
ANEXO  
=====

Listado de algunos de los programas citados  
y de la cabecera de los espectros tipo ANAE.

PICO

FTN

FILES 0,2

\* Programa para la transformacion de datos de Pico Veleta, \* escritos en el formato FLIS, al formato propio de ANAE.

\* Los datos son transmitidos desde la cinta magnetica \* al disco (conectadas ambas unidades al HP 2100) \* mediante el programa MTDS, que hay que cargar:

\* RU,FTN4,&MTDS,,- \* MR,XMTDS \* RU,LCADR,99,,,3 \* RU,MTDS } ,\*MTDS

\* El programa PICO transforma los datos de un fichero de \* nombre VELETA::13 al formato de ANAE generando un fichero \* de nombre VELEKH o bien VELEMH segun se trate de datos \* del banco de filtros de 100 kHz o de 1 MHz, respect.. \* El usuario debe encargarse de renombrar los ficheros \* con los nombres adecuados.

\* A medida que van siendo leidos los datos, se escribe \* una cierta informacion relativa a ellos en la consola \* o bien en la impresora. Cuando han sido grabados \* adecuadamente, aparece el mensaje : END-OF-RECORD .

```
PROGRAM PICO
INTEGER*4 FRECREST
REAL*6 FRECCOM
CHARACTER L*80, BUF*2560, BACK*2
DIMENSION IC(128), C(64), V(512), IV(1024), IBUF(1440)
COMMON L, BUF
EQUIVALENCE (IC,C), (C(12),FRECREST), (C(35),FRECCOM), (V,IV)
PARAMETER (DUM=-33333.)

CALL LGBUF(IBUF,1440)
LU=LOGLU(I)
WRITE(LU,*) 'Listado en impresora? (SI/NO): _' * BACK = 'NO'
READ(LU, '(A2)') BACK
IF(BACK.EQ.'SI') THEN * LL=6 * ELSE * LL=LU * ENDIF
WRITE(LU,*) 'Backend 100 kHz ? (SI/NO): _' * BACK=' '
READ(LU, '(A2)') BACK
IF(BACK.EQ.'SI') THEN * IDIM=256 * BACK='KH' * GO TO 4
ELSE IF (BACK.EQ.'NO') THEN * IDIM=512 * BACK='MH' * GOTO 4
ELSE * GO TO 3 * ENDIF

OPEN(91, FILE='VELETA::13', STATUS='OL', ERR=99, IOSTAT=IOS)
OPEN(92, FILE='VELE'//BACK//':0B:14:4:100',
& STATUS='NE', ERR=999, IOSTAT=IOS)

READ(91, END=100) BUF * CALL INIT
WRITE(LL,*) '-----'
CALL VMOV (0,,0,C,1,64) * C(39)=DUM
CALL LREAD(4) * READ(L(11:30),*) NCHAN * IC(20)=NCHAN * IC(21)=1
CALL LREAD(4) * READ(L(11:30),*) ESCALA * C(43)=ESCALA
CALL LREAD(1) * READ(L(11:30),*) OFFSET * C(45)=OFFSET
CALL LREAD(5) * READ(L(11:30),*) OFFRE
CALL LREAD(1) * READ(L(11:30),*) RESOF * C(13)=ABS(RESOF/1E6)
```

```

CALL LREAD(1) * READ(L(11:30),*) CANREF * IC(27)=CANREF
CALL LREAD(2) * READ(L(11:30),*) RA * C(8)=RA/18000.
CALL LREAD(1) * READ(L(11:30),*) OFRA * C(8)=C(8)+OFRA/18000. 1/COS(
CALL LREAD(3) * READ(L(11:30),*) DEC * C(9)=DEC/18000.
CALL LREAL(1) * READ(L(11:30),*) OFDEC * C(9)=C(9)+OFDEC/18000.
CALL LREAD(7) * READ(L(12:23),'(364)') (C(1),I=1,3)
WRITE(LL,*) ' ',L
CALL LREAD(2) * ANUL=32767.*ESCALA*OFFSET
CALL LREAD(2) * READ(L(11:30),*) FREQ * FRECREST=FREQ/1E3
WRITE(LL,*) ' ',L
CALL LREAD(1) * READ(L(11:30),*) VCANREF * C(20)=VCANREF/1E3

READ(91,ERR=333,END=100) BUF * CALL INIT

READ(L(11:20),*) DELTAV * C(40)=DELTAV/1E3
IF(FLOAT(IC(29)).NE.CANREF) THEN
  C(20)=C(20)-C(40)*(CANREF-IC(29))
  * +299792.5*OFFRA/FREQ
END IF
CALL LREAD(3) * READ(L(12:19),'(12,1X,12,1X,12)') ID,IM,IV
C(6)=(ID*100.+IM)*100.+MOD(IV,1900)
CALL LREAD(2) * READ(L(21:37),'(110,1X,18)') IDENT,IVER
WRITE(LL,*) ' ',L
IF(IVER.LT.0) IVER=0
IC(19)=IDENT * C(7)=IVER/12. * IC(31)=1
CALL LREAD(1) * READ(L(30:40),*) TOYS * IC(33)=TOYS * IC(18)=1
CALL LREAD(1) * READ(L(15:),*) TAU * C(29)=TAU
CALL LREAD(1) * READ(L(28:),*) TIME * IC(29)=TIME/2 * IC(31)=1
CALL LREAD(1) * READ(L(21:),*) ELEV * C(23)=ELEV/100.
CALL LREAD(1) * READ(L(27:),*) FRELIN
FREQ(0)=FREQ(1)
IF(FREQ/1E6.GT.FRELIN) THEN * IC(22)=1R0
  ELSE * IC(22)=1H1 * ENDIF
CALL LREAD(1) * IF(L(1:3).EQ.'END') GOTO 10
  WRITE(LL,*) ' ',L
CALL LREAD(1) * IF(L(1:3).EQ.'END') GOTO 10
  WRITE(LL,*) ' ',L
CALL LREAD(1) * IF(L(1:3).EQ.'END') GOTO 10
  WRITE(LL,*) ' ',L

* END

10 WRITE(LL,*) ' NCHAN=', NCHAN
READ(91) (V(I),I=1,NCHAN)
IF(NCHAN.LE.256) THEN * M=256 * ELSE * M=312 * ENDIF
READ(91,'(A14)') L(1:14)
IF(M.NE.IDIM) GO TO 1
CALL EXTEN (C,V)
IF(RESOF.LT.0.) THEN
  CALL VSWP (V,1,V(NCHAN),-1,NCHAN/2)
  IC(27)=NCHAN+1-IC(27)
END IF
IF (M.EQ.256) THEN
  CALL COMPA (C,V)
  M=M/2
END IF
WRITE(92,ERR=1000) (C(I),I=1,64),(V(I),I=1,M)
WRITE(LL,*) 'END-OF-RECORD'
GO TO 1

```

```
99  WRITE(LU,*) 'ERROR ',IOS,' en la apertura del fichero de',' datos'  
100  STOP  
999  WRITE(LU,*) 'ERROR ',IOS,' en la apertura del fichero de',  
&      ' resultados'  
      STOP  
1000 WRITE(LU,*) 'ERROR ', 'en la escritura de',' resultados'  
      STOP  
333  WRITE(LU,*) 'ERROR ', 'en la lectura de datos'  
      STOP  
      END
```

EAN
-----

FTN  
 \*FILES 0,2

PROGRAM EAN ( ), >1955<

\* Conversion de un fichero analizabile con AEE (ANAE Extendido)  
 \* a un fichero analizabile con el programa ANAE normal.  
 \* Para ello cercena los vectores de datos dejandolos en 256 valores.  
 \* Antes traslada los datos de manera que no haya nulos al principio  
 \* del vector.  
 \* Admite el caso de dos vias.  
 \* Puesta en marcha:  
 \* EAN,fichero origen,fichero destino,(num. vias inicial,num. vias final)  
 \* Admite un quinto parametro: se pueda escoger el corrimiento del origen de  
 \* espectro dando el numero de nulos a colocar.

```
CHARACTER*12 FILE1,FILE2
CHARACTER VIA1*1,VIAF*1,NULO*3
INTEGER VIA,VIA1,VIA2
DIMENSION IBOF(1152),IC(128),C(64),V(256,2),W(512),U(128)
EQUIVALENCE (IC,C),(V,W),(V,U)
DATA FILE1/'',VIA1,VIAF/'0','0',NULO/'0'/
```

```
LU=LOGLO(L) * CALL LOGBOF(IBOF,1152)
CALL FFARM (FILE1,FILE2,VIA1,VIAF,NULO)
```

```
IF(FILE1(1:1).EQ.' ') THEN
  WRITE(LU,*)
  * 'EAN,fichero origen,fichero destino,'
  WRITE(LU,*)
  * '(num. vias inicial,num. vias final),'
  * 'lnulos]'
  STOP
END IF
```

```
READ(VIA1,*) VIA1
READ(VIAF,*) VIA2
READ(NULO,*) NUL * IF(NUL.GT.0) NUL=NUL+1
```

```
OPEN (31,FILE=FILE1,STATUS='OLD',ERR=99,IOSTAT=IOS)
OPEN (32,FILE=FILE2,STATUS='NEW',ERR=99,IOSTAT=IOS)
```

```
1 READ(31,ERR=99,IOSTAT=IOS,END=100) C,W
  VIA=IC(21) * N=IC(20) * DUM=C(39)
```

```
IF(VIA1.NE.0) VIA=VIA1
```

```
IF(VIA.EQ.1) THEN
  IC(20)=MIN0(256,N)
  IF(N.GT.IC(20)) THEN
    IF(NUL.NE.0) THEN * 1=NUL * GO TO 2 * END IF
    DO I=1,N * IF(W(I).NE.DUM) GO TO 2 * END DO
    ICORR=I-1 * IF(ICORR.EQ.0) GO TO 3
    CALL VMDV(W(I),1,W,1,N-ICORR) * IC(27)=IC(27)-ICORR
  END IF
ELSE
```



```
IC(20)=MIN0(128,N)
IF(N.GT.IC(20)) THEN
  DO IVIA=1,2
    IF(NUL.NE.0) THEN $ I=NUL $ GO TO 4 $ END IF
    DO I=1,N $ IF(V(I,IVIA).NE.DUM) GO TO 4 $ END DO
    ICORR=I-1 $ IF(ICORR.EQ.0) GO TO 5
    CALL VMOV(V(I,IVIA),1,V(I,IVIA),1,N-ICORR)
    IC(26+IVIA)=IC(26+IVIA)-ICORR
  END DO
ELSE
  CALL VMOV(W(257),1,W(129),1,N)
END IF
END IF

IF(VIA2.NE.0) IC(21)=VIA2
3 CALL COMPA (C,V)
WRITE(32,ERR=99,IOSTAT=IOS) C,U
GO TO 1

99 WRITE(LU,*) 'ERROR ',IOS,' en acceso a ficheros'

100 CLOSE (32) $ CLOSE (31)
END
$INCLUDE AC(MPC):15
```

## EDICA

```
FTN
%FILES 0,2
PROGRAM EDICA
```

```
* Programa destinado a la modificacion de la cabecera
* de los registros de un fichero de datos tipo ANAD.

* El fichero de partida NO es destruido y se crea uno
* nuevo, cuyo nombre hay que dar en la forma:
*
*      nomfch:cs:cr

* Las modificaciones a realizar en la cabecera deben
* incluirse en este programa editando la zona indicada.
```

```
INTEGER*4 FREQR
DIMENSION IBUF(384),IC(128),V(128),C(64)
CHARACTER FILEN*12,TIPO*2
EQUIVALENCE (IC,C),(FREQR,C(12))
LU=LOGLO(L) * CALL LGRUF(IBUF,384)
WRITE(LU,*) 'Fichero partida: _'
READ(LU,'(A)') FILEN
OPEN (33,FILE=FILEN,STATUS='OLD',ERR=99,IOSTAT=IOS)
WRITE(LU,*) 'Fichero llegada: _'
READ(LU,'(A)') FILEN
OPEN (44,FILE=FILEN,STATUS='NEW',ERR=99,IOSTAT=IOS)

DO I=1,10000
    READ(33,ERR=99,END=100,IOSTAT=IOS) IC,V

*** edicion de la cabecera *****

** ejemplos:
**      IF(IC(20).EQ.128) IC(20)=127           ! cambio numero canal
**      IC(19)=100+IC(19)                     ! cambio del numero de identificacio
**      IC(27)=NINT(IC(27)/60.)               ! paso del tiempo a minutos y segundos
**      FREQR=87654321                        ! freq. repaso en kr
**      C(40)=C(13)*1000.*299792.5/FREQR     ! resolucio

*****

10      WRITE(44,ERR=99,IOSTAT=IOS) IC,V
        ENDDO

100     CLOSE(33) * CLOSE(44) * STOP
99      WRITE(LU,*) 'Error ',IOS,' en acceso a ficheros'
        END
```

```

RES=C(40) * NCF=IC(26) * NCI=1
VELC=C(21) * IF (VELC.EQ.0) VELL=C(20)
VI=(IC(27)-NCF-.5)*RES+VELC
VF=(IC(27)-.5)*RES+VELC
*definicion de la escala en velocidades, antes de la 1a grafica*
IF (VIR.EQ.VFR) THEN * RESR=RES * VIF=VI * VFR=VF
  WRITE (LU,*) 'Velocidad inicial y final ',VIR,VFR,' '
  READ (LU,*) VIR,VFR
  RKMM=(VFR-VIR)/DX
  WRITE(88,'(6A20)') (IC(I),I=1,6) ! nombre de la fuente
  WRITE(88,'(2F12.3)') VIR,VFR ! intervalo graficado
END IF
*ejecucion de la grafica*
IF (VI.LT.VFR.AND.VF.GT.VIR) THEN
  DO WHILE (VI.LT.VIR) * VI=VI+RES * NCF=NCF-1 * END DO
  DO WHILE (VF.GT.VFR) * VF=VF-RES * NCI=NCI+1 * END DO
  SX=(VI-VIR)/RKMM * DX=(VF-VI)/RKMM
  IF (YMAX.EQ.YMIN) THEN * CALL VMD (YI,YF)
    ELSE * YI=YMIN * YF=YMAX
  END IF
  WRITE(88,'(2F12.6,18,112)') YI,YF,FECHA,FRE
  CALL HISTO (15,V,NCF,NCI,X+SX,Y,DX,DY,YI,YF)
  WRITE (LU,*) 'X+5X,DX:',X+5X,DX
END IF
Y=Y-AY ! nueva posicion
*controles para no salirse de la hoja*
IF (Y.LE.0) THEN * Y=Y0 * X=X+AX * END IF
IF (X+DX.GE.255) THEN * WRITE (LU,*) 'Cambiar escala' * GO TO 1
END IF
GO TO 2 * END

```

\*\*\*\*\*

#### SUBROUTINE EXTEN

```

COMMON /C/ C(64),V(256),DUM
DIMENSION IC(128),IV(512)
EQUIVALENCE (V,IV),(C,IC)

```

```

FE=C(43) * OF=C(45)
DO I=IC(20),1,-1
  IF (IV(I).EQ.-32768) THEN * V(I)=DUM
  ELSE * V(I)=IV(I)*FE+OF
  END IF
END DO * DO I=43,46 * C(I)=0 * END DO
END

```

\*\*\*\*\*

#### SUBROUTINE VMD (VMIND,VMAXD)

```

COMMON /C/ IC(128),V(256),DUM

```

```

VMIND=DUM * VMAXD=DUM
DO L=1,IC(20)
  IF (V(L).NE.DUM) THEN
    IF (VMIND.EQ.DUM.OR.VMIND.GT.V(L)) VMIND=V(L)
    IF (VMAXD.EQ.DUM.OR.VMAXD.LT.V(L)) VMAXD=V(L)
  END IF
END DO * END

```

# GRSPC

#FILES 0,2,3

PROGRAM GRSPC ( ), (850918) L.A. Delgado

- \* Dibujo conjunto de espectros ajustados por velocidades.
- \* Solo dibuja via 1.
- \* Si no damos valores a YMIN e YMAX, la escala de cada grafica quedara ajustada a su minimo y su maximo.
- \* Escribe informacion en el fichero .SCALA:\*\*:14 para programas de rotulacion posteriores (p. ej., el programa SCALA).

```

INTEGER*4 FRQ
CHARACTER NAM*15,R*1
COMMON /C/ IC(120),V(256),DUM
DIMENSION C(64),LBUF(384),U(128)
EQUIVALENCE (C,IC),(V,U),(C(12),FRQ),(C(6),FECHA)

```

```
DATA X0,Y0,DX,DY,AX,AY/40.,120.,90.,25.,80.,30./
```

```
DDN=-33333,
LU=LOCLU(I) * CALL LGBUF (LBUF,384)
```

```

*parametros basicos y logicos de las graficas*
WRITE (LU,*) 'X0,Y0,DX,DY,AX,AY (40 120 90 25 80 30)? _'
! X0,Y0: Posicion inicial para la primera grafica
! DX,DY: tamaño del cuadro de la grafica
! AX,AY: Incrementos para posicionar las siguientes graficas
READ (LU,*) X0,Y0,DX,DY,AX,AY
WRITE (LU,*) 'YMIN, YMAX y IVS: _' * IVS=E
READ (LU,*) YMIN,YMAX,IVS * CALL ZVS (IVS)

```

```

*bucle de graficas
100 CLOSE (Y0) ! apertura de nuevo fichero. Alas fin
WRITE (LU,*) 'Nombre del fichero? _' * READ (LU,'(A)') NAM

```

```

*velocidad*
IF (NAM(17).EQ.' ') THEN * CALL ZEND
WRITE(LU,*) 'Tienes que cargar el fichero .SCALA ?'
STOP * END IF

```

```

*apertura fichero de datos 3 espectros*
OPEN (Y7,FILE=NAM,STATUS='OLD',IOSTAT=IOS)
IF (IOS.NE.0) THEN
WRITE (LU,*) IOS, ' I/O ERROR' * GO TO 100
END IF
CLOSE(OS)

```

```

*apertura fichero salida informacion*
OPEN (S8,FILE='.SCALA:**:14',STATUS='OL')
WRITE(S8,'(Y14)') X0,Y0,DX,DY,AX,AY,IVS

```

```

*lectura y dibujos*
IF (X.NE.0) GO TO 2
1 X=X0 * Y=Y0 ! inicio del dibujo o nuevo papel
2 READ (Y7,IOSTAT=IOS) C,W ! lectura de espectros
IF (IOS.NE.0) THEN ! error o EOF
WRITE (LU,*) IOS, ' I/O ERROR' * GO TO 100
END IF * CALL EXTEN
! N=no ; A=cambio fichero ; RETURN=si

```

```

WRITE (LU,'(I5,EX,6A2,"N,X,RTN? _")') IC(19),(IC(12),I=1,5)
READ (LU,'(A)') R
IF (R.EQ.'N') GO TO 2 ! espectro anulado
IF (R.EQ.'X') GO TO 100 ! cambio de fichero

```

\*informacion relativa a velocidades y resolucion\*

```

RES=C(40) * NCF=IC(28) * NCI=1
VELC=C(21) * IF (VELC.EQ.0) VELL=C(20)
VI=(IC(27)-NCF-.5)*RES+VELC
VF=(IC(27)-.5)*RES+VELC
*definicion de la escala en velocidades, antes de la 1a grafica*
IF (VIR.EQ.VFR) THEN * RESK=RES * VIR=VI * VFK=VF
  WRITE (LU,*) 'Velocidad inicial y final ',VIR,VFR,'/1'
  READ (LU,*) VIR,VFR
  RKMM=(VFR-VIR)/DX
  WRITE(86,'(6A2)') (IC(I),I=1,6) ! nombre de la fuente
  WRITE(86,'(2F12.3)') VIR,VFR ! intervalo graficado
END IF
*ejecucion de la grafica*
IF (VI.LT.VFR.AND.VF.GT.VIR) THEN
  DO WHILE (VI.LT.VIR) * VI=VI+RES * NCF=NCF+1 * END DO
  DO WHILE (VF.GT.VFR) * VF=VF-RES * NCI=NCI+1 * END DO
  SX=(VI-VIR)/RKMM * DX=(VF-VI)/RKMM
  IF (YMAX.EQ.YMIN) THEN * CALL VMD (YI,YF)
    ELSE * YI=YMIN * YF=YMAX
  END IF
  WRITE(86,'(2F12.5,18,1)E') YI,YF,FECHA,FRO
  CALL HISTO (15,V,NCF,NCI,*,SX,Y,DX,DY,YI,YF)
  WRITE (LU,*) 'X+5X,DX:',X+5X,DX
END IF
Y=Y-AY ! nueva posicion
*controles para no salirse de la hoja*
IF (Y.LE.0) THEN * Y=Y0 * X=X+AX * END IF
IF (X+DX.GE.255) THEN * WRITE (LU,*) 'Cambiar papel' * GO TO 1
END IF
GO TO 2 * END

```

\*\*\*\*\*

SUBROUTINE EXTER

```

COMMON /C/ C(64),V(256),DUM
DIMENSION IC(126),IV(512)
EQUIVALENCE (V,IV),(C,IC)

```

```

FE=C(43) * OF=C(45)
DO I=IC(20),1,-1
  IF (IV(I).EQ.-32768) THEN * V(I)=DUM
  ELSE * V(I)=IV(I)*FE+OF
  END IF
END DO * DO I=43,46 * C(I)=0 * END DO
END

```

\*\*\*\*\*

SUBROUTINE VMD (VMIND,VMAXD)

```

COMMON /C/ IC(126),V(256),DUM

```

```

VMIND=DUM * VMAXD=DUM
DO L=1,IC(26)
  IF (V(L).NE.DUM) THEN
    IF (VMIND.EQ.DUM.OR.VMIND.GT.V(L)) VMIND=V(L)
    IF (VMAXD.EQ.DUM.OR.VMAXD.LT.V(L)) VMAXD=V(L)
  END IF
END DO * END

```



```

READ(88,30,END=99) YI,YF,FECHA,FRQ ! END=solo hay un espectro
CALL SATIS(FECHA)
CALL TRANSICION,FRQ)
A=ESCA(YF) * Z(2,3)=A * Z(2,4)=A
WRITE(LU,*) 'espectro 2 ',A,' K'
Y0=Y0-AY ! nuevo origen vertical
CALL ZLIT (6,IGION,1,X0+DX-4*CSX,Y0+DY-4*CSY,LSA,CSY)
WRITE(LIT, '(16)') FECHA * IF(LIT(1:1).EQ.' ') LIT(1:1)=0'
CALL ZLIT (6,LIT,1,X0+DX-4*CSX,Y0+DY-2*CSY,CSX,CSY)
CALL NUBE (15,Z,1,4,X0,Y0,BX,DY,XI,XF,YI,YF,0)
IF (A.GT.1.) THEN * ASSIGN 10 TO FMT
ELSE * ASSIGN 20 TO FMT * A=A*10
CALL ZPUN (X0+1,Y0+DY/1.5,CO,CDY)
END IF
WRITE (LIT,FMT=FMT) A
CALL ZLIT (4,LIT,1,X0+1,Y0+DY/1.5,CSX,CSY)

```

\*estructura datos 3er espectro\*\*\*\*\*

```

READ(88,30,END=99) YI,YF,FECHA,FRQ ! END=solo hay 2 espectros
CALL SATIS(FECHA)
CALL TRANSICION,FRQ)
CLOSE(88) ! se cierra el fichero de datos (.50VCM;14)
Y0=Y0-AY ! nuevo origen vertical
A=ESCA(YF) * Z(2,3)=A * Z(2,4)=A
WRITE(LU,*) 'espectro 3 ',A,' K'
CALL ZLIT (6,IGION,1,X0+DX-4*CSX,Y0+DY-4*CSY,CSX,CSY)
WRITE(LIT, '(16)') FECHA * IF(LIT(1:1).EQ.' ') LIT(1:1)=0'
CALL ZLIT (6,LIT,1,X0+DX-4*CSX,Y0+DY-2*CSY,CSX,CSY)
CALL NUBE (15,Z,1,4,X0,Y0,BX,DY,XI,XF,YI,YF,0)
IF (A.GT.1.) THEN * ASSIGN 10 TO FMT
ELSE * ASSIGN 20 TO FMT * A=A*10
CALL ZPUN (X0+1,Y0+DY/1.5,CO,CDY)
END IF
WRITE (LIT,FMT=FMT) A
CALL ZLIT (4,LIT,1,X0+1,Y0+DY/1.5,CSX,CSY)

```

\*velocidad en unidades\*\*\*\*\*

```

99 X0=X0+5 * Y1=CSY/2 ! reposicionamiento del origen en X, tamaño
V1=ESCA(VF-VI)/Z ! semianchura redondeada.
V2=ESCA(VF+VI)/Z ! valor medio redondeado
Z(1,1)=V2-V1 * Z(2,1)=1 ! vector para dibujar la escala horizontal
Z(1,2)=V2-V1 * Z(2,2)=0 ! se dibujan a velocidades
Z(1,3)=V2 * Z(2,3)=1 ! una central y 2 laterales
Z(1,4)=V2 * Z(2,4)=1 ! cubriendo un rango lo mas
Z(1,5)=V2 * Z(2,5)=0 ! amplio posible siendo 300
Z(1,6)=V2+V1 * Z(2,6)=0 ! números redondeados.
Z(1,7)=V2+V1 * Z(2,7)=1
CALL NUBE (15,Z,1,7,X0,Y0-5,DX,BY,V1,V2,0,1,0) ! dibujo
Y0=Y0-CSY-A ! posicionamiento vertical para el dibujo

```

\*velocidad media

```

V=V2-V1 * X0=X0+DX*(V-V1)/(VF-VI) ! idea horizontal
WRITE(LIT, '(14)') V ! se escribe V en una variable caracter
CALL ZLIT (4,LIT,0,XV,YV,CSX,CSY) ! se rotula la primera veloci

```

\*velocidad media

```

V=V2    * XV=X0+DX*(V-V1)/(VF-V1)    ! posicionamiento siguiente V
WRITE(LIT,'(13)') V
CALL ZLIT(3,LIT,0,XV,YV,CSX,CSY)      !                               -1
CALL KMS (XV+2*CSX,YV,CSX,CSY)       ! rotulo: h s

```

\* velocidad mayor

```

V=V2+V1 * XV=X0+DX*(V-V1)/(VF-V1)
WRITE(LIT,'(13)') V
CALL ZLIT(3,LIT,0,XV,YV,CSX,CSY)

```

\* cuadrado\*\*\*\*\*

```

IF(ICUADRO.EQ.1) THEN
  Z(1,1)=0 * Z(2,1)=0                    ! coordenadas que definen el
  Z(1,2)=1 * Z(2,2)=0
  Z(1,3)=1 * Z(2,3)=1
  Z(1,4)=0 * Z(2,4)=1
  Z(1,5)=0 * Z(2,5)=0
  X0=X0-1-19*CSX/4 * Y0=YV-15*CSX/4      ! posicion fisica
  BX=DX+5.5*CSX+1 * BY=YSUP-YV+8*CSY-15 ! tamaño fisico
  CALL NUBE(15,Z,1,5,X0,Y0,BX,BY,0.,1.,0.,1.,0) ! dibujo
END IF
CALL ZEND                                ! despedida del plotter
WRITE(LU,('(STOP ESCALA)')              ! mensaje de despedida del pr

```

\* algunos formatos\*\*\*\*\*

```

10  FORMAT(2.0"K ")                    ! formato escala vertical cuando valor<
20  FORMAT(" (1)" K")                  ! " " " " " valor<
30  FORMAT(2F12.6,I8,I1P)              ! formato de lectura del fichero .SCALEP
END

```

\* Subrutina para el dibujo de un punto mas a la derecha  
\* de lo que lo hace, por defecto, el plotter.  
\* Entradas: posicion inferior izquierda del caracter (X0,Y0)  
\* y tamaño de los caracteres usados (CSX,CSY).  
\* Salida: dibujo en el plotter.

```

SUBROUTINE ZPUN (X0,Y0,CSX,CSY) ,(<850911>
CHARACTER LIT*1
LIT='.'
CALL ZLIT (1,LIT,1,X0+CSX/2,Y0,CSX,CSY)
END

```

\* Subrutina para pasar una fecha del formato Yeee  
\* (donde e es el formato internacional (aeeedd)).  
\* Entrada y salida: FECHA  
\* (con solo cambiar LT por UT en el IF(), se consigue  
\* el efecto contrario: internacional->Yeee)

```

SUBROUTINE SATIS (FECHA) ,(<850911>
IF(FECHA.LT.4E5) THEN
  I=FECHA/1E4 * J=FECHA/100-I*100 * K=ANG/(FECHA,100.)
  FECHA=I+100*(J+100.*K)
END IF
END

```



- \* Subrutina para obtener una lista de caracteres
- \* que informan de la transición de que se trata.
- \* Entrada: FRQ      frecuencia en reposo de la transición en kHz (10<sup>4</sup>)
- \* Salida:    ICION    lista de 8 caracteres, justificadas a la derecha

```

SUBROUTINE TRANS (ICION,FRQ)  ,(850910)
CHARACTER*8 ICION
INTEGER*4 FRQ
I=FRQ/1.E6
IF(I.EQ.43)            THEN # ICION=' Si0 v=1'
ELSE IF(I.EQ.86) THEN # ICION=' Si0 v=0'
ELSE IF(I.EQ.115) THEN # ICION='CO J=1,0'
ELSE                   # ICION='            ' # ENDIF
END

```

- \* FUNCTION para la determinación de una escala.
- \* Funciona por recorte a una cifra significativa.
- \* Cambiando el parametro FACTOR se puede controlar
- \* el tipo de redondeo.

```

FUNCTION ESCA (Y)  ,(850910)
PARAMETER (FACTOR=1.00)
Z=FACTOR*Y # IA=ALOG(Z)
IF((IA.LT.0).OR.(IA.EQ.0.AND.Z.LT.1.)) IA=IA-1
A=10.**IA # B=Z/A # ESCA=I*B*A
END

```

- \* Subrutina para la rotulación de km s<sup>-1</sup>
- \* Entradas: posición inferior izquierda (X0,Y0) y
- \*            tamaño de los caracteres (CSX,CSY).
- \* Salida:    impresión del rotulo en el plotter.

```

SUBROUTINE KMS (X0,Y0,CSX,CSY),(850911)
CHARACTER LIT*2
LIT='km'
CALL ZLIT (2,LIT,1,X0,Y0,CSX,CSY)
LIT='s '
CALL ZLIT (1,LIT,1,X0+2.3*CSX,Y0,CSX,CSY)
LIT='-1'
CALL ZLIT (2,LIT,1,X0+3.1*CSX,Y0+CSY/2,CSX/2,CSY/2)
END

```

\*\* CAECERA \*\* 128 palabras: C(64) = IC(128)

IDENTIFICACION

C(1) a C(5)	nombre de la fuente	id. del 1er espectro
C(6)	dia-mes-año	id. "
C(7)	T.U. inicial	id. "
C(8)	A.R.	id. "
C(9)	declinacion	id. "
C(10) 1a p.	No. identif. espectro	No. ident. 1er espectro

CANALES,....

C(10) 2a p.	No. de canales (256,128,64,40)	No. mayor de canales usado
C(11) 1a p.	No. de vias o receptores (1,2)	No. mayor de vias usado
2a p.	banda lateral (S,I) (byte der.)	id. 1er espectro (INTEGER*4)
C(12)	frecuencia raya en reposo (kHz)	id. "
C(13)	ancho de los canales (MHz)	id. "

PARAMETROS PRACTICOS DE LA OBSERVACION

C(14) 1a p.	canal raya en via 1	id. espectro promedio
2a p.	" " " 2	id. "
C(15) 1a p.	tiempo integracion via 1 (seg)	id. total
2a p.	" " " 2 (seg)	id. "
C(16) 1a p.	1	No. espect. promediados via
2a p.	1	" " via
C(17) 1a p.	temperatura receptor via 1 (K)	id. media
2a p.	" " " 2 (K)	id. media
C(18)	temp. calibrador via 1 (K y fr)	id. media (K y fraccion)
C(19)	" " " 2 "	id. media "

VALORES ASTRONOMICOS

C(20)	velocidad respecto al L.S.R.	id. 1er espectro
C(21)	" " al Sol	id. "
C(22)	tiempo siderico inicial	id. "
C(23)	elevacion final	id. media via 1
C(24)	" "	id. " " 2
C(25)	acimut final	id. " " 1
C(26)	" "	id. " " 2
C(27)	angulo paralactico	id. " " 1
C(28)	" "	id. " " 2

ATMOSFERA,....

C(29)	opacidad	id. media
C(30)	correccion a la opacidad	id. "
C(31)	temp. ambiente (K)	id. "
C(32)	temp. atmosfera (K)	id. "
C(33)	GI	id. "
C(34)	acoplamiento a la atmosfera	id. "

FRECUENCIAS

C(35) a C(37)	frecuencias F1 Y F2 (REAL*6) comandadas al receptor	id. 1er espectro
C(38) 1a p.	armónico N1	id. "
2a p.	" N2	id. "

OTR05

C(39)	dummy (valor nulo)	id.
C(40)	resolucion (km/s)	id.
C(41)	sigma via 1	total
C(42)	sigma via 2	total