

Informe Técnico CAY 1983-1.

PROGRAMA INTERACTIVO PARA EL ANALISIS  
DE OBSERVACIONES RADIO ESPECTRALES

Pere Planesas

Dentro del proyecto de investigación  
"ESTUDIO DE LA ESTRUCTURA Y COMPOSICION  
QUIMICA DE LA GALAXIA" y en vista de los  
nuevos medios de cálculo disponibles en  
en Centro Astronómico de Yebes, se han  
realizado, tal como se describe en la  
introducción, programas para análisis  
de las observaciones espectrales

## INDICE

=====

1. INTRODUCCION .....	3
2. PROGRAMA 'RRL' DE ANALISIS DE OBSERVACIONES ESPECTRALES .....	4
2.1 DESCRIPCION GENERAL .....	4
2.2 INICIALIZACIONES .....	5
2.3 LECTURA DE COMANDOS .....	6
2.4 DESCRIPCION DETALLADA DE LOS COMANDOS .....	7
2.5 MENSAJES DE ERROR .....	12
2.6 DESCRIPCION DE LOS FICHEROS .....	14
2.7 DESCRIPCION DE LAS VARIABLES CONTENIDAS EN LAS SENTENCIAS DIMENSION Y EQUIVALENCE .....	16
2.8 CARGA DEL PROGRAMA .....	17
2.9 SUBROUTINAS UTILIZADAS POR 'RRL' .....	18
1. Subrutinas de ajuste por polinomios de Chebyshev	
2. Subrutinas de interés general que incluyen el uso de 'dummies'	
3. Subrutinas que hacen gráficas de derecha a izquierda	
4. Subrutinas generales	
5. Subrutinas específicas de RRL	
3. PROGRAMAS COMPLEMENTARIOS .....	20
3.1 PROGRAMAS RELATIVOS A OBSERVACIONES EN LA 'EEM' .....	20
3.1.1 Programa TRASL	
3.1.2 Programa TRAS2	
3.1.3 Programa MCH2	
3.2 PROGRAMAS RELATIVOS A OBSERVACIONES EN EL 'MPIFR' .....	25
3.2.1 Programa DATEF	
3.3 PROGRAMAS PARA EL ANALISIS POSTERIOR .....	26
3.3.1 Programa AREA	
3.3.2 Programa LINEA	
ANEXOS .....	28
A Listado del programa RRL .....	29
B Listados de las subrutinas utilizadas por 'RRL' .....	43
C Listados de los programas complementarios .....	55

## 1. INTRODUCCION

=====

El hecho de disponer de una unidad de disco magnético para los ordenadores del Centro Astronómico de Yebes (CAY) permite analizar los datos relativos a las observaciones espectrales con una comodidad y rapidez muchísimo mayores que anteriormente. A largo plazo hemos proyectado disponer de un buen programa de análisis instalado en el HP-1000F adquirido el presente año, entendiéndose por programa bueno uno versátil, cómodo, rápido y lo más automático posible, que convierta la hasta ahora dura y larga tarea de analizar las grandes cantidades de espectros que se obtienen en una sesión cualquiera de observaciones en una tarea más rápida y cómoda, y, por lo tanto, eficaz. Sin embargo, un tal programa no es fácil de realizar: hay que planearlo a largo plazo, construyendo sus secciones poco a poco y con cuidado. Es necesario, también, conocer bien el ordenador a fin de aprovechar al máximo sus posibilidades, lo que no se consigue hasta después de un largo periodo de uso. Además hay que tener en cuenta la experiencia de todos los astrónomos del CAY adquirida en el uso de distintos tipos de programas de análisis en sus estancias en observatorios extranjeros (Nançay, Effelsberg, ...). Y por último hay que disponer de los periféricos necesarios.

Dado que el único terminal gráfico conectado inicialmente al HP-1000F era un Plotter, mientras que no se sabía cuando podría disponerse de un monitor de TV y por otra parte faltaba un cable especial de conexión de dicho ordenador al terminal Tektronix, no nos pareció de interés iniciar ya la realización de tal programa de análisis. Ahora bien, el tener acumulados muchos espectros para analizar (procedentes de observaciones en la estación de INTA-NASA en Robledo de Chavela y en el radiotelescopio de 100 m de Effelsberg) nos ha obligado a desarrollar un programa de análisis espectral que permita llevar a cabo dicho análisis aprovechando el disponer de la unidad de disco magnético. Para su confección hemos buscado o realizado y luego probado diversas subrutinas que van a formar parte del futuro programa de análisis.

El presente programa se ha denominado RRL pues todos los datos a analizar que tenemos acumulados en estos momentos corresponden a Rayas

de Recombinacion a frecuencias Radio (Radio Recombination Lines, en inglés). Se trata de un programa conversacional pero con un número de instrucciones bastante limitado, pero suficiente, por la limitada capacidad del ordenador que se ha usado, el HP-2100S pues es el que tiene conectado actualmente el terminal gráfico Tektronix. No se basa en una estructura en segmentos como el programa ANA de análisis de observaciones en el continuo (Ver Informe Tecnico 1983-2.), sino de un único programa, que está en todo momento completo dentro del ordenador. En la unidad de disco se encuentran dos ficheros: uno de ellos contiene los DATOS a analizar y el otro contiene los RESULTADOS obtenidos. Ambos ficheros tienen igual formato.

Han debido realizarse además diversos programas adicionales que convierten los datos del formato como están grabados en las cintas magnéticas procedentes de cada estación de observación a un formato único adecuado para ser analizados con el programa RRL.

## 2. PROGRAMA 'RRL' DE ANALISIS DE OBSERVACIONES ESPECTRALES

### 2.1 DESCRIPCION GENERAL

Permite el análisis (limpieza, promedio, ajuste) de espectros calculados y calibrados anteriormente y que se encuentran en el fichero de datos como dos secuencias de 128 valores reales correspondientes a espectros obtenidos simultáneamente mediante dos receptores o dos vias distintas.

A medida que van siendo leídos los espectros (ambas vias simultáneamente) individualmente o por grupos, se eliminan de manera automático o por el usuario aquellos canales que corresponden a valores claramente erróneos (interferencias, canales que no funcionan bien). A continuación pueden acumularse en unos vectores que, tras promediar en ellos los que se desee, pueden ser grabados en el fichero de resultados. Antes quizás deba recurrirse a extraer una línea de base o realizar un suavizado (Hanning) e incluso es posible promediar las dos vias.

Naturalmente también es posible analizar espectros con una sola vía o receptor, la primera. Basta con que conste que se trata de espectros de tal característica en la cabecera del registro correspondiente. Sin embargo, el generar un fichero de datos con una sola vía o receptor hace que se desperdicie espacio en el disco; puede ser conveniente construir tal fichero de manera que estén ocupadas ambas vías a pesar de corresponder a espectros tomados en distintos momentos.

El fichero de resultados generado por RRL puede a su vez ser analizado, para lo que es necesario en primer lugar salirse del programa y reiniciarlo.

## 2.2 INICIALIZACIONES

Al iniciarse el programa se piden los nombres de los dos ficheros, de datos y de resultados, y son abiertos. Si no es posible abrir el fichero de resultados por no existir aparece en pantalla el mensaje FICHERO NO EXISTENTE, se pide un código de seguridad (RETURN o bien 0 significa que no lo hay) y es creado en el mismo cartridge que el de datos, apareciendo el mensaje correspondiente. Su tamaño es de 300 bloques, lo que corresponde a 50 registros (50 espectros de dos vías).

A continuación se inicializan tres parámetros que es posible cambiar posteriormente en cualquier momento mediante el comando >MODE. (Los comandos tienen todos 4 letras que hay que escribir tras el símbolo > que es escrito por el propio programa a fin de indicar que está listo para recibir nuevos comandos.) Tales parámetros son el receptor a usar que en principio son el 1 y el 2, el modo de ponderar los promedios (inicialmente, por tiempo) y el número de sigmas al que se va a recortar el espectro (inicialmente 10).

Tras ello se pide si se desea perder las partes extremas no comunes de espectros que tienen distinta longitud o que están decalados unos respecto de otros en velocidad. Sólo es deseable no perderlos (respuesta: -1) cuando se están promediando espectros ya ajustados. En caso contrario, el hecho de que tengan distintos niveles hace que se produzcan saltos en el espectro promedio. En resumen, dado que en general se promedian espectros antes de ajustarlos, hay que

responder con un RETURN a esta pregunta, indicando que se desea perder tales partes no comunes.

Por último, el programa entra automáticamente en el comando >MODE a fin de poder cambiar los 3 parámetros antes definidos. Para mayor información consultar la información relativa a tal comando.

### 2.3 LECTURA DE COMANDOS

Tras un pitido y la aparición en pantalla del símbolo > (mayor que), el ordenador queda listo para recibir un comando. Este está compuesto en general por una palabra de cuatro letras (ninguna de ellas en blanco) y un conjunto de caracteres que en general son números, blancos y comas (,). En un determinado caso (comando >DUMP) se usan letras, pero éstas son tratadas como números por el programa. La instrucción es colocada en una variable real y los demás caracteres en el primer byte de cada palabra del vector entero L(40).

En primer lugar se efectúa la comprobación de la existencia de la instrucción mediante comparación con una tabla. Los comandos posibles son:

MODE, READ, BASE, HANN, PROM, SAVE, DUMP, HELP, STOP .

y serán descritos más adelante. Si la entrada no es válida, se repite la pregunta.

A continuación se realiza la traducción de los números ASCII a binario. Para tal transformación se utiliza la operación:

$$IAND ( L(I) , 7400B ) / 256$$

Tal traducción maneja los blancos y las comas siguiendo las siguientes reglas:

a) un blanco inmediatamente detrás del comando no es tenido en cuenta, pero no es obligatorio. Es decir, READ 7 y READ7 son equivalentes.

b) en cualquier otro caso, un blanco equivale a una coma. Es decir,

READ 7 9 equivale a READ 7,9 y a READ7,9, y tambien a READ7 9. De todas formas se recomienda el uso de comas, para evitar confusiones.

c) dos (o más) comas seguidas equivale a no dar el valor de un parámetro. Ejemplos: DUMP 1,10,D significa listar la cabecera de los 10 primeros registros del fichero de datos; DUMP,,D significa listar todo el fichero de datos; BASE,4 significa calcular la linea de base con un polinomio de grado 4, sin especificar el receptor o via: si hay confusión posible el programa pedira más información.

Si el número de parámetros es excesivo, es truncado en 20 y aparece un mensaje de error. A continuación se ejecuta el comando.

## 2.4 DESCRIPCION DETALLADA DE LOS COMANDOS

### MODE

Permite escoger:

a) el receptor accesible por el programa: 1 = el primero, 2 = el segundo, 12 = ambos, RETURN = el escogido anteriormente o por inicialización. Si se escoge 2 y no existe, al intentar leerlo aparecerá un mensaje de error. Si se escoge 12 y sólo es accesible el 1, sólo será leído y accesible éste, naturalmente. Cualquier numero distinto de 1 y 2 implica 12.

b) tipo de promedio: puede ser ponderando a partir del número de espectros (SPEC), por medio de la desviación típica (SIGM) o por tiempo de integración (TIME). Como antes, RETURN significa el antes escogido o inicializado. En caso de entrada errónea se repite la pregunta.

c) recorte a N sigmas: N?: Por defecto toma el valor antes definido. Sólo se admiten numeros enteros positivos.

### READ



Este comando permite la lectura en el disco de los datos a analizar.

Es el comando más complejo y que realiza mayor cantidad de operaciones. En primer lugar realiza un control y limpieza de los valores leídos, desechando los valores negativos, los que suponen un decrecimiento, los que correspondan a un intervalo que se solapa con el precedente y se recorta el número total de registros a analizar si supera el centenar. Hay un mensaje que avisa tal acción.

Por defecto se lee el registro siguiente al último leído.  
Ejemplos:

```
>READ 1,8,12,12 lee los registros del 1 al 8 y el 12
>READ 23       lee el registro 23
>READ         lee el registro siguiente (en este caso, el 24).
```

A continuación se permite al usuario eliminar el resultado de la acumulación de los espectros analizados con anterioridad, o no eliminarlos a fin de poder añadir al resultado de la acumulación anterior uno o varios más.

Tras dicha cuestión, en caso afirmativo se inicializan una serie de variables y se inicia un largo bucle, con representación gráfica inevitable, de cada uno de los espectros, con colocación de 'dummies' (valores a no considerar) y posible acumulación de espectros.

En primer lugar se lee el registro y se genera una cabecera que es escrita mediante una subrutina (CABEC) a tal efecto. A continuación una llamada a la subrutina CLIP permite recortar (a un nivel de sigmas antes establecido) el espectro y se calculan los valores máximo y mínimo a fin de realizar las gráficas. La representación gráfica será de ambos receptores cuando ambos contengan datos y se desee analizar los dos. En caso contrario sólo se representa uno de ellos. Las gráficas se autoajustan, de manera que sus escalas en general serán ligeramente distintas. Se representa asimismo un recuadro con subdivisiones cada 2 canales y cada 10 canales, éstas mayores, con indicación de los valores máximo y mínimo, del número de canal de 10 en 10 y de las velocidades correspondientes a los canales 0, N y N/2, siendo N el número de canales representado, típicamente 64 o 128.

A continuación es posible colocar dummies independientemente a las vías 1 y 2, tanto en puntos aislados como en intervalos completos.

Por último aparece la cuestión: ACUMULAR ESPECTROS?. Tiene muchas posibles respuestas:

1 = acumular sólo el receptor 1

2 = acumular sólo el receptor 2, si es accesible

12 = acumular ambos receptores, si son accesibles

RETURN = acumular los receptores accesibles (el 1 o ambos)

0 = no acumular

-1 = interrupción del bucle de lectura. Para continuar la lectura será necesario un nuevo comando READ.

En caso de acumulación aparece un mensaje diciendo diciendo qué receptor o receptores lo han sido. En la acumulación se tiene en cuenta la posible diferencia de velocidad del canal central (o sea, el corrimiento relativo en velocidades entre los distintos espectros). La subrutina que ejecuta tal acción se denomina ACUMU. En caso de usar ponderado en sigmas, se calcula la sigma a partir de los 10 primeros valores (canales) distintos de dummy.

Al acumular se calculan al mismo tiempo los valores medios (sin ponderado, pues son meramente indicativos) de la elevación, la temperatura de sistema y la del calibrador, así como el tiempo total de integración y el número total de espectros integrados, ambas cantidades para cada vía.

El comando >READ puede dar lugar a otros dos mensajes de error. Uno en el caso de que el receptor seleccionado por el análisis no coincida con el disponible en los datos leídos. Otro en el caso de que el número de puntos a analizar sea menor que 1 o mayor que 128, lo que probablemente sea indicativo de que en tal registro no hay datos por analizar.

BASE

Permite ajustar una línea de base.

Se pueden introducir dos parámetros, a saber:

- 1) el receptor cuyo espectro deseamos ajustar, y
- 2) el grado del polinomio de Chebyshev a utilizar en el ajuste.

Por defecto se toma el receptor dado en >MODE. Si son accesibles ambos, se hace un pregunta para discernir. Por defecto, se toma grado 0 para el polinomio. Dicho grado debe estar comprendido entre 0 y 7. Por exceso se toma 7. Ejemplos:

```
BASE 2,5   receptor 2, grado del polinomio = 5
BASE,4     receptor dado por >MODE, grado = 4
BASE 1     receptor 1, grado = 0.
```

Si se desea ajustar una línea de base a un receptor no considerado, aparece un mensaje de error.

Pasados todos los controles aparece una cabecera (subrutina CABEC) y una gráfica tal como la descrita en el comando >READ. A continuación se piden los intervalos a evitar en el ajuste. Si se responde 0, se ajusta todo el espectro. Si se contesta RETURN se toman los mismos que en la ocasión anterior. Como en el comando >READ, se procede a continuación a un control y limpieza de los valores introducidos, con iguales criterios. Seguidamente se generan unos vectores X, Y que contienen los puntos a ajustar y se recurre a una subrutina, CHEBY, que, utilizando un par de subrutinas del Subroutine Software Package (SSP) del IBM-360, realiza el ajuste del polinomio de Chebyshev considerado. A continuación se calculan los residuos a fin de determinar el error cuadrático medio del ajuste y se genera un vector línea de base que es representado sobre la gráfica anterior.

Si la respuesta a la pregunta: AJUSTE DEFINITIVO? es afirmativa, la línea de base es sustraída de los datos y se ejecuta la gráfica de los residuos en la misma pantalla.

HANN

Realiza el suavizado hanning de los receptores accesibles. Si no se presenta ningún problema en su ejecución, aparece el mensaje

"SUAVIZADO HANNING".

## PROM

Realiza el promedio ponderado de los espectros resultantes de los receptores 1 y 2 colocando el resultado en el receptor 1, con el numero total de espectros y el tiempo total de integracion. Aparece asimismo la relación de pesos: peso via 2 / peso via 1.

En caso de no haber nada acumulado en alguno de los dos receptores, aparece el mensaje DATOS DISPONIBLES SOLO PARA UN RECEPTOR. Si el disponible es el 1, la relación de pesos sera 0. Si el disponible es el 2, es trasladado al 1 y la relación de pesos es 1E37 (para el ordenador, 1/0).

## DUMP

Permite listar las cabeceras de los registros ocupados del fichero de datos (D) o de resultados (R).

Por defecto toma el de datos. Se pueden fijar los registros inicial y final que se desea listar. Por defecto lista todos los que reunan ciertas condiciones (número de puntos entre 1 y 128, y otras).

Ejemplos:

DUMP 10,20,R lista los registros del 10 al 20 del fichero de result.  
DUMP 10,,D lista el registro 10 del fichero de datos  
DUMP ,100,R lista los registros 1 al 50 (máximo)  
DUMP ,,R lista todos los registros del fichero de resultados  
DUMP ,10000 lista todos los registros del fichero de datos.

En la simple cabecera listada aparecen el número del registro, nombre de la fuente, numero de identificación del espectro, hora y minutos, frecuencia en reposo, ascensión recta y declinación para 1950.0 .

## SAVE

Permite grabar un espectro acumulado en el fichero de resultados.  
Ejemplos:

SAVE 5 coloca el espectro en el registro 5

SAVE lo coloca tras el último grabado (el 6, en este caso). Si es la primera vez, lo coloca en el primer registro.

Sus acciones son: generar una cabecera, lo que se hace a partir de la última leída y los datos acumulados y escribir en el fichero. Aparece un mensaje indicando el nombre del fichero y el número del registro donde ha sido grabado.

Es posible que al primer intento de escritura encuentre un error en el código de seguridad, si el fichero ha sido creado con antelación al uso de RRL.

Si se supera el número máximo de registros (50) aparece un mensaje indicando que el fichero está lleno.

#### HELP

Es un comando de ayuda. Escribe en la pantalla una información sucinta, con ejemplos, de los comandos utilizables.

#### STOP

Cierra los ficheros y detiene el programa.

#### 2.5 MENSAJES DE ERROR

Hay diversos tipos de mensajes de error. Los hay que detienen el programa, los que simplemente dan un aviso y unos terceros que además de dar un aviso piden información suplementaria.

Si se da un mensaje desconocido, el programa responde:

## COMANDO 'cccccc' NO VALIDO

y el programa queda listo para la entrada de otro comando.

En cuanto a los parametros que siguen a cada instruccion, se dan varias posibilidades. En la mayoría de casos se toman valores por defecto. En ciertos casos, como en >READ o en los intervalos ajustar o en los intervalos a colocar dummies, hay unos controles que desprecian intervalos que no tienen sentido. En otros casos se pide más información. Éste es el caso de >BASE cuando hay dos posibles receptores y no se ha indicado sobre cual hay que realizar el ajuste. En general hay suficiente cantidad de controles como para que el programa no realice ninguna acción desastrosa por despiste del usuario.

Hay una serie de errores derivados del manejo de ficheros ('FMP errors'), ante los que el programa responde de la forma:

```
nnnnn   FMP ERROR   CODE : -ccc   PAUSE 00000
```

donde '-ccc' da el tipo de error y donde 'nnnnn' es un número entero que permite localizar en qué lugar del programa ha tenido lugar.

En el Informe Técnico CAY 1983-2, titulado "PROGRAMAS INTERACTIVOS PARA EL ANALISIS DE OBSERVACIONES RADIO EN EL CONTINUO" puede encontrarse una lista de los errores FMP que pueden aparecer con mayor posibilidad (Vease el apartado 2.1.6, paginas 12 y 13, de tal informe).

A continuación damos una lista de los numeros 'nnnnn' de localización y el momento en que aparecen:

<u>nnnnn</u>	<u>acción</u>	<u>comando</u>
1	apertura del fichero de datos	inicializaciones
3	creación del fichero de resultados	"
4	al posicionarse en el fichero de datos	READ
5	lectura del fichero de datos	READ
6	al posicionarse en el fichero de resultados	SAVE
7	escritura en el fichero de resultados	SAVE
8	apertura del fichero de resultados	SAVE

9	al posicionarse en cualquiera de los ficheros	DUMP
10	al leer cualquiera de los ficheros	DUMP
11	al cerrar el fichero de resultados	STOP
12	al cerrar el fichero de datos	STOP

Tras un error de este tipo aparece PAUSE a fin de poder realizar, si es posible, algún tipo de manipulación que permita proseguir con el programa simplemente mediante:

\*GD,RRL

En algún caso ello no es posible, siendo necesario abortar el programa mediante:

\*AB,RRL

Hay algún caso, no incluido en la lista anterior, en que el propio programa detecta el error y pide al usuario mas información a fin de subsanarlo. Tal es el caso de creación de un fichero de resultados cuando se trata de abrir sin que exista. O cuando intenta escribir en el fichero de resultados y el código de seguridad está mal, en cuyo caso pide el nuevo código de seguridad.

## 2.6 DESCRIPCION DE LOS FICHEROS

Los ficheros de datos y resultados tienen idéntica estructura, a fin de poder ser usados éstos a posteriori en el lugar de aquéllos.

Se trata de ficheros de tipo 2 (registros de tamaño definido) con registros de 768 palabras (6 bloques de 128 palabras) con un tamaño de, por lo menos, 300 bloques, o sea 50 registros, lo que equivale a usar 100 espectros. Esta es la estructura que supone el programa RRL para el fichero de resultados. Un fichero de datos creado a partir de otro programa puede ser mucho mayor.

El programa supone asimismo que en cada registro hay dos espectros de como máximo 128 puntos. La estructura es en tres grupos de dos bloques:

cabecera

espectro via 1

espectro via 2

En la lectura, los espectros se denominan en el programa V1 y V2, y en la escritura (espectro acumulado) se denominan W1 y W2.

La cabecera es, evidentemente la parte cuya estructura hay que analizar cuidadosamente. Se ha basado su construcción en la de las cintas que se obtienen en el MPIFR, en Effelsberg (Alemania), adaptándose y simplificándola. Consta de:

5 enteros, 1 real, 2 enteros, 12 caracteres, 3 enteros y 114 reales

Vamos a suponer que se hallan integrados en un buffer entero de nombre IBF(256). De entre los 5 primeros enteros interesan dos:

IBF(3) = longitud de los datos (típicamente 64, 128)

IBF(4) = número de receptores (1 o 2)

El real es el valor del 'dummy'. El programa RRL estandariza este valor al valor 6666.6670. Los 12 caracteres ASCII constituyen el nombre de la fuente. Entre los 3 enteros siguientes se usan:

IBF(16) = número de identificación del espectro

IBF(17) = número comprendido entre 0 y 3, usado como control por >DUMP

A continuación hay 114 reales que vamos a suponer contenidos en un vector real R(114) tal que existe la equivalencia EQUIVALENCE (IBF(19), R(1)). Los más interesantes son:

R(1) = 1 Valor utilizado como código por >DUMP

R(18) = tiempo universal

R(19) = tiempo sidereo

R(20) = acimut

R(22) = elevación

R(25) = ángulo paraláctico

R(26) = ascensión recta 1950.0

R(27) = declinación 1950.0

R(29) = velocidad LSR en km/s

R(37) = año mes dia, en la forma 831103



R(38) = peso  
 R(45) = tiempo total de integración del receptor 1  
 R(46) = idem 2  
 R(49) = temperatura del calibrador 1  
 R(50) = idem 2  
 R(51) = número total de espectros promediados en vía 1  
 R(52) = idem 2  
 R(73) = temperatura de sistema en el receptor 1  
 R(74) = idem 2  
 R(85) = ancho de banda en kHz  
 R(89) = velocidad LSR en el canal central en km/s  
 R(93) = resolución en velocidad, en km/s  
 R(105) = frecuencia en reposo, en MHz.

Todos los ángulos y tiempos astronómicos vienen dados en mediasvueltas (o sea, radianes/pi).

## 2.7 DESCRIPCIÓN DE LAS VARIABLES CONTENIDAS EN LAS SENTENCIAS DIMENSION Y EQUIVALENCE

Van a ser descritas sólo las de mayor interés.

NAME se usa para abrir ambos ficheros. A lo largo del programa contiene el nombre del fichero de resultados  
 IDCB, IDCC contienen la identificación y buffer auxiliar para acceder a uno y otro ficheros  
 IBF, IBUF buffers de lectura de los ficheros de datos y resultados respectivamente  
 V1, V2 valores de los espectros 1 y 2 leídos  
 V vector auxiliar que contiene los dos anteriores  
 W1, W2 valores de los espectros 1 y 2 almacenados, antes y después del ajuste y que van a ser grabados por >SAVE  
 NOM, NAM contienen los caracteres de la cabecera de IBF e IBUF respectivamente  
 R, RP contienen los valores reales de las cabeceras en IBF e IBUF respectivamente  
 COMND contiene los 9 posibles comandos  
 L, L4 vectores inseparables. El primero se usa para leer los parámetros que siguen a la instrucción. El segundo sirve para, en

casos límite, indicar el final del anterior

LA vector que incluye L y L4

LL, LL2 vectores inseparables. El primero contiene los valores numéricos de los caracteres ASCII de L. El segundo tiene la misma función que L4

LLA vector que incluye LL y LL2

M vector usado en la traducción de números ASCII a binario. Su dimensión indica que se admiten números de hasta 5 cifras

LP vector que almacena los intervalos a evitar en el ajuste

LREAD vector que contiene los números de los registros a analizar. Se rellena mediante >READ. Su tamaño es 100

X, Y, Z vectores usados en el ajuste de la línea de base. X e Y contienen las parejas de puntos a ajustar. Z contiene la línea de base generada

COE contiene los coeficientes del polinomio de Chebyshev ajustado en sus 8 primeras componentes. Sus componentes 9 y 10 son, respectivamente, el origen y el factor de escala que permiten normalizar el intervalo de ajuste.

## 2.8 CARGA DEL PROGRAMA

Para una información amplia sobre la carga de programas con el RTE II/III del HP-2100S ver el Anexo A del Informe Técnico CAY 1983-2, titulado PROGRAMAS INTERACTIVOS PARA EL ANALISIS DE OBSERVACIONES RADIO EN EL CONTINUO, donde se describe la carga de programas simples y la de aquellos que usan librerías y son, además, segmentados, que es el caso más general posible en nuestro sistema

Vamos a completar aquí la información allí existente, no tanto en cuanto al método, sino en lo referente a la capacidad, dado que al tratarse en esta ocasión de un programa no segmentado ha habido problemas al rebasarse en repetidas ocasiones la capacidad del ordenador.

El RTE II/III se reserva las 12 primeras K-palabras (desde la dirección de memoria 00000B hasta 27777B) para el sistema. Los programas y subrutinas se cargan en el resto (desde 30000B hasta casi 77777B). En total son unas 20 Kpalabras. Realmente hay algunas palabras al final (64) que permanecen reservadas al ordenador, de

manera que hay realmente disponibles 20414 palabras (algo más de 19.9 K).

Las subrutinas del sistema operativo utilizadas por un programa típico con entradas y salidas por pantalla y disco, modo gráfico y todo tipo de operaciones (aritméticas, trigonométricas, lógicas) ocupan de 5900 a 6000 palabras (sólo el FORMATTER ocupa ya casi 2000 palabras). Si además se utiliza la librería %FLIB, que permite realizar con mayor rapidez (comparable a la del MTS, Magnetic Tape System) las operaciones aritméticas en coma flotante, hay que tener en cuenta que ésta ocupa 41 palabras adicionales. Quedan para el programa y subrutinas del usuario poco más de 14 K (unas 14440 palabras).

En el caso que nos ocupa tenemos la siguiente distribución:

%RRL	10235
%FLIB	41
%RRLS	4213
S.O.	5925
total	20414

donde RRLS es el conjunto de subrutinas que utiliza el programa y que pasamos a describir someramente a continuación.

## 2.9 SUBRUTINAS UTILIZADAS POR 'RRL'

Vamos a dividir las en 5 grupos:

### 2.9.1 Subrutinas de ajuste por polinomios de Chebyshev:

Una llamada desde RRL a la subrutina CHERY hace que se ejecute tal ajuste mediante las subrutinas APCH1 y APFS del SSP del IBM-360. También pertenece a este grupo de subrutinas CNPS, utilizada por RRL para el cálculo de residuos. Han sido alteradas ligeramente a fin de introducir variables de doble precisión. En conjunto las 4 subrutinas ocupan 1194 palabras.

Son subrutinas de interés general para futuros programas de

análisis.

#### 2.9.2 Subrutinas de interés general que incluyen el uso de 'dummies':

AMAXD y AMIND calculan los valores máximo y mínimo de un vector prescindiendo de los valores dummy. HANND realiza un suavizado de von Hann de un conjunto de valores consecutivos algunos de los cuales pueden ser dummies. Los valores extremos pasan a su vez a ser dummies. CLIP asigna el valor dummy a todos aquellos valores que difieren de su promedio en más de un cierto número de veces su desviación típica. Sus tamaños respectivos son: 119, 118, 375 y 204 palabras. En total son 816.

#### 2.9.3 Subrutinas que hacen gráficas de derecha a izquierda:

O sea, en velocidades. Se trata de GRAF3, NUMBI y AXISI, de tamaños respectivos: 413, 178 y 290 palabras (en total son 881). Las tres subrutinas han sido extraídas de versiones más antiguas que realizan gráficas de izquierda a derecha. GRAF3 se basa en GRAFE (25-10-82). Además no dibuja los valores dummy. NUMBI se basa en NUMBR (16-8-82), pero numera a la inversa. AXISI ha sido extradida de AXIS (26-8-82) pero ha requerido muchos cambios.

#### 2.9.4 Subrutinas generales:

La subrutina PLOT es una antigua subrutina (25-8-82) utilizada por casi todas las subrutinas que realizan gráficas y no merece ninguna explicación adicional. La subrutina ERFMP analiza si ha habido error despues de usar una llamada al FMP. Si existe, imprime un mensaje y detiene momentáneamente la ejecución del programa con un PAUSE. Sus tamaños respectivos son 184 y 42 palabras, respectivamente.

#### 2.9.5 Subrutinas específicas de RRL:

CABEC es la subrutina que escribe la cabecera antes de las gráficas generadas en >READ y en >BASE. Escribe en una primera fila: nombre de la fuente, número de identificación del espectro, instante (año, mes, día, hora y minuto), ascensión recta y declinación para 1950.0 y la elevación. En la segunda fila escribe: receptor, frecuencia en reposo, resolución en km/s, velocidad LSR, temperaturas

de sistema y del calibrador, y tiempo de integración en segundos (hasta 9999, casi 3 horas). Ocupa 360 palabras.

ACUMU promedia un espectro con un promedio de espectros con los pesos adecuados y con el corrimiento en velocidades que sea necesario. Para mayor información ver el comando >READ. Ocupa 716 palabras.

En los anexos A y B pueden encontrarse listados completos y comentados del programa RRL y del conjunto de subrutinas (RRLS) respectivamente.

### 3. PROGRAMAS COMPLEMENTARIOS

=====

#### 3.1 PROGRAMAS RELATIVOS A OBSERVACIONES EN LA 'EEM'

Dado que el formato con el que se obtiene la cinta magnética que contiene los datos resultantes de una sesión de observaciones en dicha Estación no es el más adecuado para ahorrar espacio en la cinta y como que se trata de realizar muchas observaciones y acumularlas en pocas cintas, hemos desarrollado un programa que traslada la información de la cinta particular de cada observación a una única cinta y con un formato más compacto. Anteriormente realizábamos esta tarea relleno el ordenador con el máximo número de datos posible. Sin embargo su capacidad sólo permite el trasladar de una vez 87 fases de una cinta a otra, lo que corresponde a unos 21 ciclos de 4 fases; ello obligaba a repetir varias veces el proceso.

Actualmente el trasvase de datos de una cinta a otra se hace por medio del disco, con lo que se puede hacer de una sola vez. El programa que lleva a cabo tal tarea se denomina, como antes, TRASL.

La cinta magnética es el mejor método para almacenar con seguridad grandes cantidades de datos, pero no para proceder a su análisis, por lo que hemos desarrollado otro pequeño programa, TRAS2, cuya tarea es simplemente trasladar los datos de la cinta donde están escritos en binario y sin formato (cinta comprimida) a un fichero en disco que

tiene su mismo formato.

Dado que el programa RRL analiza directamente espectros, hemos debido realizar un programa, ya bastante más complejo que los dos anteriores, para el cálculo de espectros. No es esta su única tarea, sino que genera un fichero de datos adecuado en base a diversas cuestiones que plantea al usuario. Dicho programa se denomina MCH2.

Pasamos a describir a continuación los tres programas anteriores.

### 3.1.1 Programa TRASL:

Este programa permite el traslado de los datos obtenidos con el receptor multicanal en la Estación Espacial de Madrid y grabados en ASCII a otra cinta magnética con formato binario, por lo tanto mucho más compacta. Se utiliza como intermediario un fichero en disco que es creado por el propio programa y posteriormente purgado por él.

La unidad de cinta magnética se utiliza de forma equivalente a un fichero disco habiéndole dado, previamente al uso de este programa, un nombre de la forma:

CR,MAGTAP,8,WR,BO,EO,BI

El nombre del fichero que equivale a la cinta es, pues, MAGTAP y se encuentra localizado en el cartridge 2. No es purgado, por lo que a menudo permanece creado de una utilización anterior.

Gracias a este proceso, el conjunto de fases puede dividirse en varios ficheros en cinta magnética, correspondiendo cada uno de ellos a los ciclos consecutivos realizados en un mismo punto del cielo o en unas mismas condiciones. Ello facilita su localización posterior.

Hay mensajes de error en caso de estar la unidad de cinta en LOCAL o en el caso en que la cinta a grabar no tenga colocada la anilla. En el proceso de generación de ficheros en cinta, las preguntas a responder son:

- número de fases en el próximo fichero en cinta ? (0=FIN)

Tras un mensaje en que aparece la cabecera de la ultima fase leida:  
- end-of-file ? (1=SI)

Este programa utiliza diversas llamadas al FMP y la subrutina MOVE para control de la cinta magnética.

### 3.1.2 Programa TRAS2:

El número maximo de fases que permite trasladar es, en su versión del 83.04.12, de 200. Las cuestiones que plantea el programa son:

- Nombre del fichero
- Hay que crearlo o machacar uno ya existente?
- Número de fases por ciclo
- Número de ciclos a leer (FIN = 0)

El número de fases por ciclo utilizadas en una observacion en Cebreros o Robledo es de 3 o 4, pero dado que casi desde el principio de la campaña de observaciones decidimos hacer 4, el programa MCH2 esta preparado para tratar 4 fases únicamente.

Terminado el proceso, este programa permite listar las posiciones de cada ciclo, así como el nombre de la fuente.

Este programa utiliza para leer la subrutina LEE, que a su vez hace uso de la subrutina MOVE para producir desplazamientos en la cinta magnética.

### 3.1.3 Programa MCH2:

Utiliza un fichero generado por TRAS2 y que se supone que va a ser "purgado" a continuación, aunque es posible conservarlo, así como es posible realizar el proceso de llenado de un fichero de espectros en varias sesiones. Se realizan una serie de cálculos y se definen parámetros supuesta la observación en Cebreros o Robledo de la raya H142alfa.

Los espectros se calculan supuestas observaciones en conmutación

de frecuencia de 4 fases, a saber:

fase 1	frecuencia 1	
fase 2	"	1 y tubo de ruido conectado
fase 3	"	2
fase 4	"	2 y tubo de ruido conectado

y utilizando las calibraciones:

$$\frac{1 - 3}{2 - 1} \quad \text{y} \quad \frac{3 - 1}{4 - 3}$$

correspondiendo, en este orden, a los espectros 1 y 2 del fichero. Se supone que el número de canales usado es 128 y que el corrimiento en frecuencias equivale a 64 canales.

El programa plantea las siguientes cuestiones de inicialización:

- datos tomados por segundo, tanto por tiempo efectivo en la toma de datos, ancho de banda por canal (en kHz).

Por defecto toma: 20, 80, 50, que son los valores habitualmente usados.

- nombre del fichero de datos.
- nombre del fichero de resultados.
- crearlo o abrirlo?

No hay creación automática si se dice "abrirlo" y no existe.

En su caso es creado en el mismo cartridge que el de datos.

- Código de seguridad? (Return = no)
- Registro inicial del fichero de resultados (en caso de abrirlo)

A continuación hay una serie de cuestiones ya relativas a un cierto grupo de ciclos. Algunos de los parámetros a entrar se refieren al cálculo del espectro. Otros se necesitan para generar una cabecera bastante más completa que la que se tiene a partir de la EEM. La respuesta "-1" a cualquiera de las preguntas siguientes (salvo VLSR y canal) reinicia este ciclo de preguntas.

- temperatura de sistema (K) y del tubo de ruido (K). (0 = FIN)

Por defecto toma como temperatura del tubo la grabada en la cinta



magnética.

- ciclos inicial y final a tratar.  
Admite hasta 5 intervalos. NO hay control.
- ciclos aislados a eliminar.  
Dentro del grupo anterior. Hasta 10.
- AR (H, M, S) y DEC (G, ', ").  
Return = anteriores valores.
- VLSR (km/s).  
Return = anterior valor.
- canal para el que la velocidad es VLSR.  
Return = valor anterior.
- deseas corregir los datos anteriores ?  
En caso afirmativo se reinicia este ciclo de preguntas.

En caso de dar por válidas las respuestas a las preguntas anteriores se entra en un largo bucle donde se analiza cada ciclo. En pantalla aparecen el nombre de la fuente, ciclo y temperatura del tubo de ruido usada tan pronto como han sido calculados los espectros. Estos se determinan de tal manera que se da valor mudo (dummy) en el caso de un canal que tome sistemáticamente el mismo valor en las fases 1 y 2 o 3 y 4 (o sea, al que no le afecta la conexión del tubo de ruido. Ello puede producirse cuando es nula su respuesta o cuando está saturado el canal.)

A continuación, y dentro de este bucle, se calcula la cabecera del registro, determinándose diversos ángulos (acimut, elevación, ángulo paraláctico) y tiempos (universal, sidereo, fecha), que son reducidos a medias vueltas (radianes/pi). Por último, se graba en el disco en un registro cuyo número aparece en pantalla.

Finalizado el bucle, se limpia la pantalla y se vuelve al ciclo de preguntas. La respuesta "0" a la pregunta "Temperatura de sistema, ..." hace que sea cerrado el fichero de datos y listado todo el de espectros a partir de un registro inicial que se escoge (por defecto toma el 1). Se listan simplemente registro y nombre de fuente. Habiendo llegado al final o a un registro no rellenado con datos, es cerrado tal fichero y se pregunta si se desea purgar el fichero de datos, cosa que SI es aconsejable hacer.

Este programa utiliza varias subrutinas, que son:

LEE62 lee el fichero generado mediante TRSS2.  
Supone ciclos de 4 fases y 275 palabras.

TSGOH calcula el tiempo sidéreo en Greenwich a las 0 horas de TU para el día de observación.

MESDI dados el año y el día del año, calcula el mes y el día del mes. Tiene en cuenta los años bisiestos.

ERFMP da cuenta de los errores en el manejo de ficheros.

Asimismo, se utilizan un par de funciones:

VM reduce a medias vueltas (de -1 a +1).

ASIN calcula el arcoseno, dando un resultado comprendido entre  $-\pi/2$  y  $+\pi/2$ .

### 3.2 PROGRAMAS RELATIVOS A OBSERVACIONES EN EL 'MPIFR'

#### 3.2.1 Programa DATEF

Este programa permite leer las cintas de datos en el formato del Max-Planck Institut fur Radioastronomie y grabarlos en un fichero disco. Las preguntas que este programa plantea al usuario son:

- nombre del fichero disco
- código de seguridad
- hay que crearlo (1) o abrrirlo (0) ?
- número del cartridge (por defecto=15)
- número de bloques
- número de escanes a leer (0=FIN, -1=REWIND, 9999= TODOS)

Tras ello se inicia un bucle en el que son leídos los escanes uno a uno, generada una ~~cabecera~~ cabecera compatible con el programa RRL, transformados los datos de ASCII a binario y grabado todo ello en un registro de 768 palabras de longitud.

Este programa utiliza tres subrutinas:

HEADR genera la cabecera, simplificando la original

MOVE controla los movimientos de la cinta magnética

ERFMP avisa de errores del FMP y permite su localización

### 3.3 PROGRAMAS PARA EL ANALISIS POSTERIOR

#### 3.3.1 Programa AREA

Permite el cálculo aproximado del área de una raya. El espectro es leído en un fichero disco del mismo tipo que los analizados o generados mediante el programa RRL.

Las preguntas que efectua al usuario son:

- nombre del fichero
- registro a leer (0=FIN)

Por defecto utiliza el anteriormente leído.

- canales entre los que quieres calcular el area (0=LISTAR)

El listado sirve para orientar en la elección de dichos canales.

En el caso de calcular el area, la da para ambas vias y en K km/s.

El cálculo del area se efectua mediante la subrutina ITRAP, que efectua el cálculo por el método del trapecio.

Este programa tiene otra opción que es la de perforar una cinta de papel con los valores del espectro (solo la via 1). Para ello basta responder a la pregunta "registro a leer" con el numero del registro con signo negativo. En este caso no es calculada el área ni es posible listar. Ello es útil para conservar los valores de los espectros en un soporte no magnético, pero sobretodo para la utilización del programa LINEA, que pasamos a describir a continuación.

#### 3.3.2 Programa LINEA

Este programa permite el ajuste de una gausiana mediante una subrutina denominada AJGAU debida también al autor de este informe, pero que no vamos a describir. Dicha gausiana se ajusta a un conjunto de puntos igual o menor que 100 y mayor o igual que 10, de abscisas consecutivas equidistantes, y que pueden introducirse desde el teclado o bien desde una cinta de papel. En ambos casos aparece un listado con

la máxima precisión posible en pantalla en unidades arbitrarias a fin de poder corregir algún valor si es necesario.

Este programa utiliza las siguientes subrutinas:

AJGAU para el ajuste de la gaussiana y un polinomio de grado hasta 3  
GRAF realiza la gráfica del espectro y la gaussiana en la pantalla.

ANEXOS

=====

ANEXO A

=====

LISTADO DEL PROGRAMA 'RRL'

```
00001 FTN
00002 PROGRAM RRL
00003 C PROGRAMA DE ANALISIS DE ESPECTROS
00004 C 83.10.27 PERE PLANESAS
00005 DIMENSION NAME(3),IDCB(784),ISIZE(2),IDCC(784),LA(42),LLA(42),
00006 ,L(40),L4(2),LL(20),LL2(2),COMND(9),M(5),W1(128),W2(128),
00007 ,X(128),Y(128),Z(128),COE(10),SIG(2),
00008 ,IBF(768),IBUF(768),LREAD(100),LP(42),
00009 ,V(256),V1(128),V2(128),NOM(6),R(114),NAM(6),RP(114),E(2),
00010 ,AR2(3),DE2(3)
00011 C
00012 EQUIVALENCE (V(1),V1(1)),(IBF(257),V(1)),(V(129),V2(1)),
00013 ,(R(1),IBF(19)),(IBUF(257),W1(1)),(IBUF(513),W2(1)),
00014 ,(RP(1),IBUF(19)),(L(1),LA(1)),(LL(1),LLA(1)),(LLA(21),LL2(1)),
00015 ,(L4(1),LA(41)),(DUMMI,IBUF(6)),(IBF(6),DUMMY)
00016 C
00017 DATA COMND/4HMODE,4HREAD,4HBASE,4HHANN,4HPROM,4HDUMP,4HSAVE,
00018 ,4HSTOP,4HHELP/, L4/0,0/, LL2/0,0/, DUM/6666.6670/
00019 C
00020 C -- INICIO DEL PROGRAMA --
00021 C
00022 C APERTURA DE FICHEROS
00023 WRITE(6,210)
00024 210 FORMAT(/"RRL"7X"83.10.27"//
00025 "FICHERO DE DATOS?:"6X"_")
00026 READ(1,211) NAME
00027 211 FORMAT(3A2)
00028 CALL OPEN(IDCB,IE,NAME)
00029 CALL ERFMP(1,IE)
00030 WRITE(6,212)
00031 212 FORMAT("FICHERO DE RESULTADOS?: _")
00032 READ(1,211) NAME
00033 CALL OPEN(IDCC,IE,NAME)
00034 IF(IE.GE.0) GO TO 215
00035 WRITE(6,1141)
00036 1141 FORMAT(4X"FICHERO NO EXISTENTE")
00037 ICR=IAND(IDCB(1),77B)
00038 ISIZE(1)=300
00039 ISIZE(2)=768
00040 WRITE(6,214)
00041 214 FORMAT("CODIGO DE SEGURIDAD? (RETURN=NO) _")
00042 READ(1,211) KODE
00043 IF(KODE.EQ.20040B) KODE=0
00044 CALL CREAT(IDCC,IE,NAME,ISIZE,2,KODE,ICR)
00045 CALL ERFMP(3,IE)
00046 IF(IE.GE.0) WRITE(6,213) NAME,ICR
00047 213 FORMAT("FICHERO '3A2' CREADO EN CARTRIDGE"13)
00048 C INICIALIZACIONES
00049 215 LRID=0
00050 C NUMERO DE REGISTROS DEL FICHERO DE RESULTADOS LIMITADO A 50.
00051 NRTOT=300/6
00052 NREGF=0
00053 C PARA QUE ANULE LOS ACUMULADORES
00054 IANU=1
00055 IRECM=12
00056 PROM=4HTIME
00057 PROMA=4HTIME
00058 ICLIP=10
00059 ICODE=1
00060 DUMMI=DUM
```

```
00061      DO 216 I=1,42
00062 216   LP(I)=0
00063      WRITE(6,2463)
00064 2463  FORMAT("AL PROMEDIAR, DESEAS PERDER LAS PARTES NO COMUNES ("
00065      "RETURN=SI) O NO (-1)?_")
00066      READ(1,x) ICODE
00067      ICODE=ICODE/IABS(ICODE)
00068 C VAMOS A 'MODE' PARA INICIALIZAR EL PROGRAMA
00069      GO TO 10
00070 C
00071 C -- COMANDOS --
00072 C
00073 9      DO 116 I=1,20
00074 116   LL(I)=0
00075 C PITIDO Y MARCA DE LECTURA
00076      CALL EXEC(2,506B,7,1)
00077      WRITE(6,200)
00078 200   FORMAT(")_")
00079      READ(1,201) COMAN,L
00080 C SOLO SE RELLENA EL PRIMER BYTE DE CADA PALABRA DE L
00081 201   FORMAT(A4,40A1)
00082 C COMPROBACION DEL COMANDO
00083      DO 6 ICOMN=1,9
00084      IF(COMAN.EQ.COMND(ICOMN)) GO TO 7
00085 6      CONTINUE
00086      WRITE(6,202) COMAN
00087 202   FORMAT(30X"COMANDO 'A4' NO VALIDO")
00088      GO TO 9
00089 C TRADUCCION DE LOS NUMEROS DE ASCII A BINARIO.
00090 C SE RELLENA LA PARTE FINAL DE L CON 'SS' PARA PODER DETERMINAR
00091 C MAS ABAJO EL FINAL DE LOS DATOS
00092 7      DO 1 I=1,40
00093      J=41-I
00094      IF(L(J).NE.2H ) GO TO 171
00095 1      L(J)=2HSS
00096 171   NL=1
00097      NM=0
00098      M(1)=0
00099 C SOLO SE ADMITEN BLANCOS, COMA Y NUMEROS.
00100 C LOS DEMAS SIMBOLOS SON TRATADOS COMO NUMEROS
00101      DO 4 I=1,40
00102 C UN BLANCO TRÁS EL COMANDO NO ES TENIDO EN CUENTA
00103      IF(L(I).EQ.2H .AND.I.EQ.1) GO TO 4
00104 C EN CUALQUIER OTRO CASO, UN BLANCO EQUIVALE A UNA COMA
00105      IF(L(I).EQ.2H, .OR.L(I).EQ.2H .OR.L(I).EQ.2HSS) GO TO 2
00106 C TRANSFORMACION DEL NUMERO ASCII DEL PRIMER BYTE A BINARIO
00107      NM=NM+1
00108      M(NM)=IAND(L(I),7400B)/256
00109      GO TO 4
00110 2      DO 3 J=1,NM
00111 3      LL(NL)=LL(NL)*10+M(J)
00112      IF(L(I).EQ.2HSS) GO TO 5
00113      NL=NL+1
00114      NM=0
00115 C PARA QUE SE PUEDAN DAR COMAS SEGUIDAS
00116      M(1)=0
00117 C CONTROL
00118      IF(NL.GT.20) GO TO 118
00119 4      CONTINUE
00120 C EJECUCION DEL COMANDO ESCOGIDO
```



```
00121 5      GO TO (10,20,30,40,50,60,70,80,90), ICOMN
00122 C NUMERO DE PARAMETROS EXCESIVO: SE TRUNCA EN 20
00123 118    WRITE(6,119) COMND(ICOMN),LL
00124 119    FORMAT("NUMERO DE PARAMETROS EXCESIVO: "/">"A4,1X,16(I3",""/
00125        &6X,3(I3","")I3)
00126      GO TO 5
00127 C
00128 C ** MODE **
00129 C
00130 10      WRITE(6,18) IRECM,PROM,ICLIP
00131 18      FORMAT("* RECEPTOR: "I3,5X" PROMEDIO: "A4,5X" RECORTE A "I3" SIGMAS")
00132      WRITE(6,11)
00133 11      FORMAT("RECEPTOR? (1,2,12=AMBOS): _")
00134      READ(1,*) IRECM
00135 C RETURN=EL ANTERIOR
00136 C CUALQUIER COSA DISTINTA DE 1, 2 O RETURN IMPLICA 12.
00137      IF(IRECM.NE.1.AND.IRECM.NE.2) IRECM=12
00138 14      WRITE(6,12)
00139 12      FORMAT("TIPO DE PROMEDIO? (SPEC,SIGM,TIME): _")
00140      READ(1,201) PROM
00141 C RETURN= EL ANTERIOR
00142      IF(PROM.EQ.4H      ) PROM=PROMA
00143      IF(PROM.NE.4HSPEC.AND.PROM.NE.4HSIGM.AND.PROM.NE.4HTIME) GOTO 14
00144      PROMA=PROM
00145      WRITE(6,153)
00146 153     FORMAT("RECORTE A 'N' SIGMAS: N= _")
00147      ICL=0
00148      READ(1,*) ICL
00149      IF(ICL.GE.1) ICLIP=ICL
00150      GO TO 9
00151 C
00152 C ** READ **
00153 C
00154 20      K=0
00155      LREAD(1)=0
00156 C CONTROL Y LIMPIEZA DE LOS VALORES LEIDOS
00157      DO 235 I=1,20,2
00158 2353   IF(LLA(I).EQ.0) GO TO 236
00159      IF(LLA(I).LT.0) GO TO 271
00160      IF(LLA(I).GT.LLA(I+1).AND.LLA(I+1).NE.0) GO TO 271
00161      IF(I.EQ.1) GO TO 235
00162      IF(LLA(I).LT.LLA(I-1)) GO TO 271
00163      GO TO 235
00164 271    DO 237 J=I,20,2
00165      LLA(J)=LLA(J+2)
00166 237    LLA(J+1)=LLA(J+3)
00167      GO TO 2353
00168 235    CONTINUE
00169 C
00170 C ALMACENAMIENTO DE LOS NUMEROS DE LOS REGISTROS A LEER
00171 236    DO 21 I=1,20,2
00172      IF(LL(I)) 23,23,22
00173 22     DO 21 J=LL(I),LL(I+1)
00174      K=K+1
00175      LREAD(K)=J
00176      IF(K.GT.100) GO TO 251
00177 21     CONTINUE
00178 23     IF(LREAD(1)) 24,24,26
00179 C NUMERO DE REGISTROS EXCESIVO
00180 251    WRITE(6,252)
```

```

00181 252  FORMAT("NUMERO DE REGISTROS > 100. TRUNCACION EN 100.")
00182      GO TO 26
00183 C POR DEFECTO LEE EL REGISTRO SIGUIENTE AL ULTIMO LEIDO
00184 24    LREAD(1)=LRID+1
00185      K=1
00186 C INICIALIZACIONES
00187 26    NREAD=MINO(K,100)
00188      NRID=0
00189      IF(IANU.EQ.1) GO TO 2589
00190      WRITE(6,2599)
00191 2599  FORMAT("DESEAS PERDER LOS ANTES ACUMULADOS? (SI=1): _")
00192      READ(1,*) IANU
00193      IF(IANU.NE.1) GO TO 25
00194 2589  IANU=0
00195      WI=0
00196      WII=0
00197      DO 266 I=1,128
00198      W1(I)=0
00199 266   W2(I)=0
00200      CON=0
00201      SELEV=0
00202      STCAL=0
00203      STSYS=0
00204      TIME1=0
00205      TIME2=0
00206      SPEC1=0
00207      SPEC2=0
00208 C
00209 C BUCLE DE LECTURA.
00210 C INCLUYE LA COLOCACION DE 'DUMMIES' Y LA ACUMULACION DE ESPECTROS
00211 25    NRID=NRID+1
00212      IF(NRID.GT.NREAD) GO TO 9
00213      LRID=LREAD(NRID)
00214      CALL POSNT(IDCIB,IE,LRID,1)
00215      CALL ERFMP(4,IE)
00216      CALL READF(IDCIB,IE,IBF)
00217      CALL ERFMP(5,IE)
00218 C GENERACION DE LAS PALABRAS QUE COMPONEN LA CABECERA
00219      N=IBF(3)
00220      IF(N.LE.0.OR.N.GT.128) GO TO 254
00221      NCODE=N*ICODE
00222 C IBF(4) = NUMERO DE RECEPTORES. SI IBF(4)=1 ENTONCES ES IREC=1.
00223 C SI IBF(4)=2 ENTONCES ES IREC=12
00224      IREC=11*IBF(4)-10
00225      IF(IREC.EQ.1.AND.IRECM.EQ.2) GO TO 255
00226      IREC=MINO(IREC,IRECM)
00227      NIREC=1
00228      IF(IREC.EQ.12) NIREC=2
00229      IF(DUM.EQ.DUMMY) GO TO 1937
00230 C UNIFICACION DE DUMMIES
00231      DO 193 I=1,256
00232      IF(V(I).EQ.DUMMY) V(I)=DUM
00233 193    CONTINUE
00234 1937  DO 2521 J=1,6
00235 2521  NOM(J)=IBF(9+J)
00236      NIESP=IBF(16)
00237      TIMEA=R(45)
00238      TIMEB=R(46)
00239      SPECA=R(51)
00240      SPECB=R(52)

```

```

00241 IF(SPECA+SPECB.NE.0.0) GO TO 6114
00242 SPECA=1
00243 SPECB=1
00244 6114 ELEV=R(22)
00245 HORA=R(18)
00246 HOR=HORA*12
00247 IF(HOR.LE.0.) HOR=HOR+24
00248 TCAL=R(49)
00249 TSYS=R(73)
00250 VLSR=R(29)
00251 VCEN=R(89)
00252 BANDA=R(93)
00253 C PASO DE PAGINA. SE GRAFICAN UNO Y/U OTRO RECEPTORES
00254 CALL EXEC(2,506B,6924,1)
00255 TIMEI=TIMEA
00256 IF(IREC.EQ.2) TIMEI=TIMEB
00257 CALL CABEC(NOM,NIESP,R(37),R(26),R(27),ELEV,HOR,IREC,R(105),
00258 ,BANDA,VLSR,TSYS,TCAL,TIMEI)
00259 K=2
00260 IF(IREC.NE.2) K=1
00261 C PARAMETROS PARA LAS GRAFICAS
00262 KK=(K-1)*128+1
00263 CALL CLIP(V(KK),N,DUM,ICLIP)
00264 CALL AMAXD(V(KK),N,VMAX1,DUM)
00265 CALL AMIND(V(KK),N,VMIN1,DUM)
00266 IF(IREC.EQ.12) CALL AMAXD(V2,N,VMAX2,DUM)
00267 IF(IREC.EQ.12) CALL AMIND(V2,N,VMIN2,DUM)
00268 CALL NUMBI(1005.,705.,900.,N,10)
00269 AX=9000./N
00270 AX5=AX/5
00271 AX10=AX/10
00272 XSD=985-AX10
00273 DXT=900-AX10
00274 C GRAFICA
00275 CALL AXISI(985,705,AX,900,25.,250,10,10)
00276 CALL AXISI(985,705,AX5,900,5.,250,4,4)
00277 FE=250/(VMAX1-VMIN1)
00278 CALL GRAF3(V(KK),N,XSD,455.,DXT,250.,FE,VMIN1,DUM)
00279 CALL PLOT(1,535,453,535,473)
00280 CALL PLOT(0,0,680)
00281 WRITE(6,302) VMAX1
00282 302 FORMAT(F6.3)
00283 CALL PLOT(0,0,455)
00284 WRITE(6,302) VMIN1
00285 IF(IREC.NE.12) GO TO 3043
00286 C SI SE ANALIZAN DOS RECEPTORES, SEGUNDA GRAFICA
00287 CALL AXISI(985,455,AX,900,25.,250,10,10)
00288 CALL AXISI(985,455,AX5,900,5.,250,4,4)
00289 FE=250/(VMAX2-VMIN2)
00290 CALL GRAF3(V2,N,XSD,205.,DXT,250.,FE,VMIN2,DUM)
00291 CALL PLOT(1,535,203,535,223)
00292 CALL PLOT(0,0,435)
00293 WRITE(6,302) VMAX2
00294 CALL PLOT(0,0,205)
00295 WRITE(6,302) VMIN2
00296 C VELOCIDADES
00297 3043 KK=430
00298 IF(IREC.EQ.12) KK=180
00299 C VELOCIDADES CORRESPONDIENTES A LOS CANALES 'N' Y '0'.
00300 RANV=BANDA*FLOAT(N/2)

```

```
00301      VIN=VCEN+RANV
00302      VFIN=VCEN+RANV
00303      CALL PLOT(0,53, KK)
00304      WRITE(6,2592) VIN
00305 2592  FORMAT(I4)
00306      CALL PLOT(0,490, KK)
00307      WRITE(6,2592) VCEN
00308      CALL PLOT(0,953, KK)
00309      WRITE(6,2592) VFIN
00310      CALL PLOT(0,0, KK-25)
00311      K=2
00312      IF(IREC.NE.2) K=1
00313 1270  WRITE(6,27) K
00314 27    FORMAT("RECEPTOR" I2, 3X "DUMMIES? (RETURN=NO): _")
00315      J=-1
00316      READ(1,*) J
00317      IF(J.LE.0) GO TO 123
00318      DO 127 I=1,40
00319 127    L(I)=0
00320      WRITE(6,28)
00321 28    FORMAT(5X "DUMMIES EN LOS PUNTOS AISLADOS (RETURN=NING.): _")
00322      READ(1,*) L
00323 C DADOS EN UN ORDEN CUALQUIERA
00324      I=1
00325 125    IF(LA(I)) 128,128,126
00326 126    J=(K-1)*128+LA(I)
00327      V(J)=DUM
00328      I=I+1
00329      GO TO 125
00330 128    WRITE(6,129)
00331 129    FORMAT(5X "DUMMIES EN LOS INTERVALOS (RETURN=NING.): _")
00332      DO 122 I=1,40
00333 122    L(I)=0
00334      READ(1,*) L
00335      I=1
00336 1241   IF(LA(I)) 123,123,124
00337 124    IF(LA(I).GT.LA(I+1)) GO TO 1217
00338      DO 121 JLA=LA(I),LA(I+1)
00339      J=(K-1)*128+JLA
00340 121    V(J)=DUM
00341 1217   I=I+2
00342      GO TO 1241
00343 123    IF(IREC.NE.12.OR.K.NE.1) GO TO 1250
00344      K=2
00345      GO TO 1270
00346 C ACUMULACION DE LOS ESPECTROS (PARA CADA RECEPTOR)
00347 1250   WRITE(6,1251)
00348 1251   FORMAT("ACUMULAR ESPECTROS? (0=NO; -1=FIN): _")
00349      READ(1,1252) J
00350 1252   FORMAT(A2)
00351      IF(J.EQ.2H-1) GO TO 9
00352      K=0
00353      IF(J.EQ.2H ) K=IREC
00354      IF(J.EQ.2H12.AND.IREC.EQ.12) K=12
00355      IF(J.EQ.2H1 .AND.IREC.NE.2) K=1
00356      IF(J.EQ.2H2 .AND.IREC.NE.1) K=2
00357      IF(J.EQ.2H0 .OR.K.EQ.0) GO TO 2525
00358      IF(CON.EQ.0.) HORAI=HORA
00359      IF(CON.EQ.0.) NESP=NIESP
00360      WPA=TIMEA
```

```
00361 WPB=TIMEB
00362 IF(PROM.EQ.4HSPEC) WPA=SPECA
00363 IF(PROM.EQ.4HSPEC) WPB=SPECB
00364 IF(K.NE.2) SPEC1=SPEC1+SPECA
00365 IF(K.NE.1) SPEC2=SPEC2+SPECB
00366 IF(K.NE.2) TIME1=TIME1+TIMEA
00367 IF(K.NE.1) TIME2=TIME2+TIMEB
00368 CON1=CON+1
00369 SELEV=(SELEV*CON+ELEV)/CON1
00370 STSYS=(STSYS*CON+TSYS)/CON1
00371 STCAL=(STCAL*CON+TCAL)/CON1
00372 CON=CON1
00373 WRITE(6,1265)
00374 1265 FORMAT("ACUMULADO RECEPTOR _")
00375 C CORRIMIENTO: LO REDONDEA A UN NUMERO ENTERO DE CANALES
00376 IF(CON.EQ.1.) WCEN=VCEN
00377 ICORR=(VCEN-WCEN)/BANDA+0.49999
00378 INI=1
00379 IFI=N
00380 IF(ICORR.LT.0) INI=INI-ICORR
00381 IF(ICORR.GT.0) IFI=IFI-ICORR
00382 C SELECCION DEL RECEPTOR O RECEPTORES
00383 IF(K.EQ.2) GO TO 1261
00384 C RECEPTOR 1:
00385 IF(PROM.NE.4HSIGM) GO TO 1260
00386 C DETERMINACION, EN SU CASO, DE SIGMA, A PARTIR DE 10 VALORES
00387 C DE UN EXTREMO DEL ESPECTRO, DISTINTOS DE 'DUM'.
00388 J=0
00389 SX=0
00390 SY=0
00391 SXY=0
00392 SX2=0
00393 SY2=0
00394 DO 1259 I=1,10
00395 1264 J=J+1
00396 IF(V1(J).EQ.DUM) GO TO 1264
00397 T=V1(J)
00398 SX=SX+J
00399 SY=SY+T
00400 SXY=SXY+J*T
00401 SX2=SX2+J*J
00402 1259 SY2=SY2+T*T
00403 SIGY=SY2-0.1*SY*SY
00404 R2=SXY-0.1*SX*SY
00405 R2=R2*R2/SIGY/(SX2-0.1*SX*SX)
00406 WPA=1/SIGY/(1-R2)
00407 1260 CALL ACUMU(W1,V1,WI,WPA,INI,IFI,ICORR,DUM,CON,X,NCODE)
00408 IF(K.EQ.1) WRITE(6,12671)
00409 12671 FORMAT("1")
00410 IF(K.EQ.12) WRITE(6,12672)
00411 12672 FORMAT("1, 2")
00412 1261 IF(K.EQ.1) GO TO 25251
00413 C RECEPTOR 2:
00414 IF(PROM.NE.4HSIGM) GO TO 1263
00415 J=0
00416 SX=0
00417 SY=0
00418 SXY=0
00419 SX2=0
00420 SY2=0
```

```
00421 DO 1269 I=1,10
00422 1268 J=J+1
00423 IF(V2(J).EQ.DUM) GO TO 1268
00424 T=V2(J)
00425 SX=SX+J
00426 SY=SY+T
00427 SXY=SXY+J*T
00428 SX2=SX2+J*J
00429 1269 SY2=SY2+T*T
00430 SIGY=SY2-0.1*SY*SY
00431 R2=SXY-0.1*SX*SY
00432 R2=R2*R2/SIGY/(SX2-0.1*SX*SX)
00433 WPA=1/SIGY/(1-R2)
00434 1263 CALL ACUMU(W2,V2,WII,WPA,INI,IFI,ICORR,DUM,CON,X,NCODE)
00435 IF(K.EQ.2) WRITE(6,12673)
00436 12673 FORMAT("2")
00437 C PERDIDA DE 1/2 SEGUNDO PARA LEER EL MENSAJE
00438 25251 DO 25252 J=1,200
00439 25252 ZZZ=SIN(3.39)
00440 GO TO 25
00441 C
00442 C MENSAJES DE ERROR
00443 254 WRITE(6,2541) N
00444 2541 FORMAT("NUMERO DE PUNTOS N="I5" ( 1 O BIEN ) 128")
00445 GO TO 9
00446 255 WRITE(6,2551)
00447 2551 FORMAT("RECEPTOR SELECCIONADO NO COINCIDENTE CON EL DISPONIBLE")
00448 GO TO 9
00449 2525 WRITE(6,2526)
00450 2526 FORMAT("RECEPTOR NO ACUMULADO")
00451 GO TO 25
00452 C
00453 C ** BASE **
00454 C
00455 C POR DEFECTO SE TOMA EL RECEPTOR DADO POR 'IREC'.
00456 C POR DEFECTO SE TOMA GRADO 0. EN GENERAL, 0=<GRADO=<7.
00457 30 IF(LL(1).EQ.0.AND.IREC.EQ.12) GO TO 38
00458 IF(LL(1).EQ.0.AND.IREC.NE.12) I=IREC
00459 IF(LL(1).NE.0) I=LL(1)
00460 IF(I.NE.1.AND.I.NE.2) GO TO 38
00461 IF(I.EQ.1.AND.WI.EQ.0.) GO TO 3021
00462 IF(I.EQ.2.AND.WII.EQ.0.) GO TO 3021
00463 IGRAD=LL(2)
00464 IF(LL(2).GT.7) IGRAD=7
00465 IF(LL(2).LT.0) IGRAD=0
00466 SIG(I)=0
00467 GO TO 31
00468 C CASO EN QUE SE DESEA AJUSTAR UN RECEPTOR NO CONSIDERADO
00469 3021 WRITE(6,32) I
00470 32 FORMAT("RECEPTOR" I2 " NO ACCESIBLE")
00471 GO TO 9
00472 C GRAFICA DEL ESPECTRO A AJUSTAR
00473 31 CALL EXEC(2,506B,6924,1)
00474 TIMEI=TIME1
00475 IF(I.EQ.2) TIMEI=TIME2
00476 HOR=HORAI*12
00477 IF(HOR.LE.0.) HOR=HOR+24
00478 CALL CABEC(NOM,NIESP,R(37),R(26),R(27),SELEV,HOR,I,R(105),
00479 ,BANDA,VLSR,STSYS,STCAL,TIMEI)
00480 IF(I.EQ.1) CALL AMAXD(W1,N,VMAX,DUM)
```

```
00481 IF(I.EQ.1) CALL AMIND(W1,N,VMIN,DUM)
00482 IF(I.EQ.2) CALL AMAXD(W2,N,VMAX,DUM)
00483 IF(I.EQ.2) CALL AMIND(W2,N,VMIN,DUM)
00484 FE=250./(VMAX-VMIN)
00485 CALL NUMBI(1005.,705.,900.,N,10)
00486 CALL AXISI(985,705,AX,900,25.,250,10,10)
00487 CALL AXISI(985,705,AX5,900,5.,250,4,4)
00488 IF(I.EQ.1) CALL GRAF3(W1,N,XSD,455.,DXT,250.,FE,VMIN,DUM)
00489 IF(I.EQ.2) CALL GRAF3(W2,N,XSD,455.,DXT,250.,FE,VMIN,DUM)
00490 CALL PLOT(1,535,453,535,473)
00491 CALL PLOT(0,0,683)
00492 WRITE(6,302) VMAX
00493 CALL PLOT(0,0,455)
00494 WRITE(6,302) VMIN
00495 VIN=WCEN-RANV
00496 VFIN=WCEN+RANV
00497 CALL PLOT(0,53,432)
00498 WRITE(6,2592) VIN
00499 CALL PLOT(0,490,432)
00500 WRITE(6,2592) WCEN
00501 CALL PLOT(0,953,432)
00502 WRITE(6,2592) VFIN
00503 CALL PLOT(0,0,135)
00504 C LECTURA DE PARAMETROS PARA EL AJUSTE
00505 DO 339 J=2,40
00506 339 L(J)=0
00507 WRITE(6,33)
00508 33 FORMAT("INTERVALOS A EVITAR (0=NING.) (RETURN=MISMOS): _")
00509 L(1)=-1
00510 READ(1,*) L
00511 C RETURN=LOS MISMOS QUE ANTES
00512 IF(L(1).EQ.-1) GO TO 392
00513 C CONTROL Y LIMPIEZA DE LOS VALORES ENTRADOS
00514 DO 35 II=1,40,2
00515 351 IF(LA(II).EQ.0) GO TO 36
00516 IF(LA(II).LT.0) GO TO 371
00517 IF(LA(II).GT.LA(II+1)) GO TO 371
00518 IF(II.EQ.1) GO TO 35
00519 IF(LA(II).LT.LA(II-1)) GO TO 371
00520 GO TO 35
00521 371 DO 37 J=II,40,2
00522 LA(J)=LA(J+2)
00523 37 LA(J+1)=LA(J+3)
00524 GO TO 351
00525 35 CONTINUE
00526 36 NPAR=II-1
00527 DO 391 J=1,40
00528 391 LP(J)=L(J)
00529 392 WRITE(6,361) (LP(J),J=1,NPAR)
00530 361 FORMAT((I3,17("","I3"))/)
00531 C GENERACION DE LOS VECTORES X, Y A PARTIR DE LP Y LOS DUMMIES.
00532 C CALCULO DEL NUMERO DE PUNTOS A AJUSTAR (NPA).
00533 LI=1
00534 KI=1
00535 DO 331 J=1,N
00536 K=(I-1)*128+J
00537 IF(W1(K).EQ.DUM) GO TO 331
00538 IF(J.GT.LP(LI+1)) LI=LI+2
00539 IF(LP(LI).NE.0) GO TO 3321
00540 LP(LI)=N+1
```

```
00541 LP(LI+1)=N+1
00542 3321 IF(J.GE.LP(LI)) GO TO 331
00543 X(KI)=J
00544 Y(KI)=W1(K)
00545 KI=KI+1
00546 331 CONTINUE
00547 LP(LI)=0
00548 LP(LI+1)=0
00549 NPA=KI-1
00550 CALL CHEBY(X,Y,NPA,IGRAD,IRES,COE)
00551 C CALCULO DE LOS RESIDUOS ==> SIG(I)
00552 X0=COE(9)
00553 XD=COE(10)
00554 SRES=0
00555 DO 332 J=1,NPA
00556 T=X(J)*XD+X0
00557 CALL CNPS(VALOR,T,COE,IRES)
00558 332 SRES=SRES+(Y(J)-VALOR)**2
00559 SIG(I)=SQRT(SRES/IRES)
00560 C GENERACION DEL VECTOR Z = LINEA DE BASE
00561 DO 333 J=1,N
00562 VALOR = DUM
00563 K=(I-1)*128+J
00564 IF(W1(K).EQ.DUM) GO TO 333
00565 T=J*XD+X0
00566 CALL CNPS(VALOR,T,COE,IRES)
00567 333 Z(J)=VALOR
00568 C GRAFICA DE LA LINEA DE BASE SUPERPUESTA A LA GRAFICA ANTERIOR
00569 CALL GRAF3(Z,N,XSD,455.,DXT,250.,FE,VMIN,DUM)
00570 CALL PLOT(0,870,725)
00571 WRITE(6,137) SIG(I)
00572 137 FORMAT("SIG="F7.4)
00573 CALL PLOT(0,0,75)
00574 WRITE(6,139)
00575 139 FORMAT("AJUSTE DEFINITIVO? (SI=1): _")
00576 READ(1,*) K
00577 IF(K.NE.1) CALL EXEC(2,506B,6924,1)
00578 IF(K.NE.1) GO TO 9
00579 C CONSIDERADO UN AJUSTE BUENO, LA LINEA DE BASE ES SUSTRADA Y
00580 C SE EJECUTA LA GRAFICA DE LOS RESIDUOS
00581 DO 136 J=1,N
00582 K=J+(I-1)*128
00583 IF(W1(K).NE.DUM) W1(K)=W1(K)-Z(J)
00584 136 CONTINUE
00585 IF(I.EQ.1) CALL AMAXD(W1,N,VMAX,DUM)
00586 IF(I.EQ.1) CALL AMIND(W1,N,VMIN,DUM)
00587 IF(I.EQ.2) CALL AMAXD(W2,N,VMAX,DUM)
00588 IF(I.EQ.2) CALL AMIND(W2,N,VMIN,DUM)
00589 FE=250/(VMAX-VMIN)
00590 CALL AXISI(985,430,AX,900,25.,250,10,10)
00591 KK=(I-1)*128+1
00592 CALL GRAF3(W1(KK),N,XSD,180.,DXT,250.,FE,VMIN,DUM)
00593 CALL PLOT(1,535,178,535,198)
00594 CALL PLOT(0,0,412)
00595 WRITE(6,302) VMAX
00596 CALL PLOT(0,0,180)
00597 WRITE(6,302) VMIN
00598 C VELOCIDADES
00599 CALL PLOT(0,53,163)
00600 WRITE(6,2592) VIN
```



```

00601      CALL PLOT(0,490,163)
00602      WRITE(6,2592) WCEN
00603      CALL PLOT(0,953,163)
00604      WRITE(6,2592) VFIN
00605      CALL PLOT(0,0,25)
00606      GO TO 9
00607 C CASO EN QUE IREC=12 Y NO SE HA SELECCIONADO RECEPTOR
00608 38      WRITE(6,39)
00609 39      FORMAT("RECEPTOR 1 O RECEPTOR 2? : _")
00610      READ(1,*) LL(1)
00611      GO TO 30
00612 C
00613 C ** HANN **
00614 C
00615 40      MM=N
00616      IF(IREC.NE.2) CALL HANND(W1,MM,DUM)
00617      IF(MM.EQ.-1) GO TO 9
00618      IF(IREC.NE.1) CALL HANND(W2,MM,DUM)
00619      IF(MM.EQ.-1) GO TO 9
00620      WRITE(6,41)
00621 41      FORMAT(20X"SUAVIZADO HANNING")
00622      GO TO 9
00623 C
00624 C ** PROM **
00625 C
00626 C SE PROMEDIAN, CON PESO, LOS ESPECTROS RESULTANTES PARA AMBOS RECEPTOR
00627 50      NOREC=1
00628      SPEC1=SPEC1+SPEC2
00629      TIME1=TIME1+TIME2
00630      IF(WII.EQ.0..OR.WI.EQ.0.) WRITE(6,54)
00631      WM1=1./(WI+WII)
00632      DO 51 I=1,N
00633      IF(W1(I).NE.DUM.AND.W2(I).NE.DUM)
00634      &      W1(I)=(WI*W1(I)+WII*W2(I))*WM1
00635      IF(W1(I).EQ.DUM.OR.W2(I).EQ.DUM) W1(I)=DUM
00636 51      CONTINUE
00637      WW=1E37
00638      IF(WI.NE.0.) WW=WII/WI
00639      WRITE(6,52) WW
00640 52      FORMAT("PROMEDIADOS RECEPTORES 1 Y 2"10X"RELACION PESOS 2/1 ="
00641      &1PE10,2)
00642 54      FORMAT("DATOS DISPONIBLES SOLO PARA UN RECEPTOR")
00643      GO TO 9
00644 C
00645 C ** DUMP **
00646 C
00647 C SE LISTAN UNAS SIMPLES CABECERAS DEL FICHERO DE DATOS ('D')
00648 C O DE RESULTADOS ('R'), POR DEFECTO TOMA EL DE DATOS.
00649 C SE PUEDEN FIJAR LOS REGISTROS INICIAL Y FINAL.
00650 60      NI=LL(1)
00651      NF=LL(2)
00652      IDR=LL(3)/2
00653      IF(IDR.NE.1.AND.IDR.NE.2) IDR=2
00654      IF(NI.EQ.0.AND.NF.EQ.0) NF=NRTOT
00655      IF(NI.LE.0) NI=1
00656      IF(NF.LT.NI) NF=NI
00657      IF(IDR.EQ.1.AND.NF.GT.NRTOT) NF=NRTOT
00658      WRITE(6,601)
00659 601      FORMAT("REGISTRO"2X"FUENTE"4X"IDENT."1X"FECHA"3X"HORA"
00660      &2X"FRECUENCIA"5X"A.R."6X"DECLIN."/)

```

```

00661 IF(IDR.EQ.2) CALL POSNT(IDC B,IE,NI,1)
00662 IF(IDR.EQ.1) CALL POSNT(IDCC,IE,NI,1)
00663 CALL ERFMP(9,IE)
00664 DO 61 I=NI,NF
00665 IF(IDR.EQ.2) CALL READF(IDC B,IE,IBF,768,LEN)
00666 IF(IDR.EQ.1) CALL READF(IDCC,IE,IBF,768,LEN)
00667 CALL ERFMP(10,IE)
00668 IF(LEN.EQ.-1) GO TO 63
00669 C CONTROL DE LECTURA
00670 IF(IBF(3).LE.0.OR.IBF(3).GT.128) GO TO 61
00671 IF(((IBF(17)-2)/3.NE.0.AND.R(1).NE.1.) GO TO 61
00672 DO 611 J=1,6
00673 611 NAM(J)=IBF(J+9)
00674 NISP=IBF(16)
00675 HOR=R(18)*12
00676 IF(HOR.LE.0.) HOR=HOR+24
00677 FREQ=R(105)
00678 AR1=R(26)
00679 DE1=R(27)
00680 AD=12*AR1
00681 IF(AD.LE.0) AD=AD+24
00682 AR2(1) = IFIX(AD)
00683 AD=(AD-AR2(1))*60
00684 AR2(2)=IFIX(AD)
00685 AR2(3)=(AD-AR2(2))*60
00686 AD=180*DE1
00687 DE2(1)=IFIX(AD)
00688 AD=(AD-DE2(1))*60
00689 DE2(2)=IFIX(AD)
00690 DE2(3)=ABS((AD-DE2(2))*60)
00691 DE2(2)=ABS(DE2(2))
00692 E(1)=IFIX(HOR)
00693 E(2)=(HOR-E(1))*60
00694 FECH=R(37)
00695 WRITE(6,62) I,NAM,NISP,FECH,E,FREQ,AR2,DE2
00696 62 FORMAT(I5,2X,6A2,2X,I4,2X,I6,2X,2I2,2X,F9.3,2X,2I3,F5.1,1X,3I3)
00697 61 CONTINUE
00698 63 GO TO 9
00699 C
00700 C ** SAVE **
00701 C
00702 C SE COLOCA EN EL REGISTRO ESCOGIDO DEL FICHERO DE RESULTADOS.
00703 C POR DEFECTO, ESCRIBE A CONTINUACION DEL ULTIMO EN QUE SE HA ESCRITO
00704 70 IF(LL(1).EQ.0) LL(1)=NREGF+1
00705 IF(LL(1).GT.NRTOT) GO TO 751
00706 CALL POSNT(IDCC,IE,LL(1),1)
00707 CALL ERFMP(6,IE)
00708 NREGF=LL(1)
00709 C GENERACION DE LA NUEVA CABECERA
00710 71 DO 711 I=1,256
00711 711 IBUF(I)=IBF(I)
00712 IBUF(4)=NOREC
00713 IBUF(16)=NESP
00714 RP(45)=TIME1
00715 RP(46)=TIME2
00716 RP(51)=SPEC1
00717 RP(52)=SPEC2
00718 RP(22)=SELEV
00719 RP(35)=HORAI
00720 RP(49)=STCAL

```

```

00721      RP(73)=STSYS
00722      RP(89)=VCEN
00723 C ESCRITURA EN EL FICHERO
00724 76    CALL WRITF(IDCC,IE,IBUF)
00725      IF(IE.EQ.-7) GO TO 77
00726      CALL ERFMP(7,IE)
00727      WRITE(6,766) NAME,NREGF
00728 766   FORMAT(20X"ESPECTRO GRABADO EN FICHERO '3A2'" REGISTRO "I3)
00729      GO TO 9
00730 C CASO EN QUE AL PRIMER INTENTO DE ESCRITURA ENCUENTRA UN ERROR EN
00731 C EL CODIGO DE SEGURIDAD.
00732 77    WRITE(6,78)
00733 78    FORMAT("FMP ERROR -007: BAD SECURITY CODE")
00734      WRITE(6,214)
00735      READ(1,211) KODE
00736 C POR DEFECTO, TOMA CERO (CASO DE CREAR EL FICHERO)
00737      IF(KODE.EQ.20040B) KODE=0
00738      CALL OPEN(IDCC,IE,NAME,0,KODE)
00739      CALL ERFMP(8,IE)
00740      GO TO 70
00741 751   WRITE(6,75) NAME
00742 75    FORMAT(10X"FICHERO '3A2'" LLENO")
00743      GO TO 9
00744 C
00745 C ** HELP **
00746 C
00747 90    CALL EXEC(2,506B,6924,1)
00748      WRITE(6,91)
00749 91    FORMAT(">HELP"//">MODE"/5X"RECEPTOR (1,2,12=AMBOS):"/5X
00750      &"TIPO DE PROMEDIO ? (SPEC,SIGM,TIME):"/
00751      &5X"RECORTE A 'N' SIGMAS: N="//
00752      &">READ 1,8,12,12"/10X"(==> 1 AL 8, EL 12)"
00753      &/5X"RECEPTOR 1  DUMMIES ? (RETURN=NO):"/
00754      &5X"RECEPTOR 2  DUMMIES ? (RETURN=NO):"/
00755      &10X"(DUMMIES EN PUNTOS SUELTOS O BIEN EN INTERVALOS)"/
00756      &5X"ACUMULAR ESPECTROS ? (1,2,12,0=NO,-1=FIN) :"/
00757      &10X"(RETURN=RECEP.ESCOG.EN MODE)"/
00758      &">BASE 1,4"/10X"(==> RECEPTOR=1, GRADO POL. CHEBY=4)"/
00759      &10X"(0=<GRADO=<7) (POR DEFECTO: GRADO=0)"/
00760      &10X"(POR DEFECTO: RECEPTOR ESCOGIDO EN 'MODE')"/
00761      &">HANN"//">PROM"//">SAVE 43"/10X"(==> EN REGISTRO 43 DEL FICH"
00762      &"ERO DE RESULTADOS)"/10X"(POR DEFECTO: TRAS EL ULTIMO GRABADO"
00763      &//">DUMP 12,25,R"/10X"(==> REG. INICIAL, REG. FINAL Y "
00764      &"'D'=DATOS"/10X"O BIEN 'R'=RESULTADOS)"/10X"(POR DEFECTO: "
00765      &"LISTA TODOS LOS REG. DEL FICH. DE DATOS)"/">STOP"/)
00766      GO TO 9
00767 C
00768 C ** STOP **
00769 C
00770 80    CALL CLOSE(IDCC,IE)
00771      CALL ERFMP(11,IE)
00772      CALL CLOSE(IDC8,IE)
00773      CALL ERFMP(12,IE)
00774      WRITE(6,82)
00775 82    FORMAT("/END RRL"/)
00776      END

```

ANEXO B

=====

LISTADOS DE LAS SUBROUTINAS UTILIZADAS POR 'RRL'

```

0001 FTN
0002 C *****
0003 C CONJUNTO DE SUBROUTINAS PARA EL PROGRAMA DE ANALISIS DE ESPECTROS 'RRL'.
0004 C
0005 C * SUBROUTINAS DE AJUSTE POR POLINOMIOS DE CHEBYSHEV:
0006 C     CHEBY, APCH1, APFS, CNPS
0007 C * SUBROUTINAS DE INTERES GENERAL, PERO QUE INCLUYEN EL USO DE 'DUMMIES':
0008 C     AMAXD, AMIND, HANND, CLIP
0009 C * SUBROUTINAS QUE HACEN GRAFICAS DE DERECHA A IZQUIERDA ( EN VELOCIDADES)
0010 C     GRAF3, NUMBI, AXISI
0011 C * SUBROUTINAS ESPECIFICAS DE 'RRL':
0012 C     CABEC, ACUMU
0013 C * SUBROUTINAS GENERALES:
0014 C     ERFMP, PLOT
0015 C *****
0016     SUBROUTINE CABEC(NOM,NIESP,FECHA,AR,DEC,ELEV,HORA,IRC,FREQ,
0017     ,DELTU,VLSR,TSYS,TTDR,TINT)      ,83.07.16
0018     DIMENSION NOM(6),A(3),D(3),E(2)
0019     B=12*AR
0020     IF(B.LE.0.) B=B+24
0021     A(1)=IFIX(B)
0022     B=(B-A(1))*60
0023     A(2)=IFIX(B)
0024     A(3)=(B-A(2))*60
0025     B=180*DEC
0026     D(1)=IFIX(B)
0027     B=(B-D(1))*60
0028     D(2)=IFIX(B)
0029     D(3)=ABS((B-D(2))*60)
0030     D(2)=ABS(D(2))
0031     E(1)=IFIX(HORA)
0032     E(2)=(HORA-E(1))*60
0033     ELV=180*ELEV
0034     WRITE(6,2) NOM,NIESP,FECHA,E,A,D,ELV
0035 2     FORMAT(6A2,I5,I8,I3,I2,3X"AR="I2" "I2" "F4.1,3X"DEC="I3" "I2" "
0036     ,I2,3X"ELEV="I3)
0037     WRITE(6,3) IRC,FREQ,DELTU,VLSR,TSYS,TTDR,TINT
0038 3     FORMAT("REC" I3 FRQ="F9.3" RES.V="F5.2" VLSR="F5.1
0039     "" TSYS="F5.1" TTR="F4.1" TIME="I4)
0040     END
0041 C *****
0042     SUBROUTINE AMAXD(A,N,VMAX,D)      ,83.07.15
0043 C DETERMINA EL VALOR MAXIMO 'VMAX' DE LOS 'N' PRIMEROS
0044 C VALORES ALMACENADOS EN 'A', PRESCINCIENDO DE AQUELLOS
0045 C CUYO VALOR SEA 'D' (LOS DUMMIES).
0046     DIMENSION A(1)
0047     DO 1 I=1,N
0048     IF(A(I).NE.D) GO TO 2
0049 1     CONTINUE
0050     WRITE(6,99)
0051 99     FORMAT("TODOS LOS VALORES CONSIDERADOS SON 'DUMMIES'")
0052     RETURN
0053 2     VMAX=A(I)
0054     DO 3 J=I,N
0055     IF(A(J).EQ.D) GO TO 3
0056     IF(VMAX-A(J)) 4,3
0057 4     VMAX=A(J)
0058 3     CONTINUE
0059     RETURN
0060     END

```

```
00061 C *****
00062 SUBROUTINE AMIND(A,N,VMIN,D) ,83.07.15
00063 C DETERMINA EL VALOR MINIMO 'VMIN' DE LOS 'N' PRIMEROS
00064 C VALORES ALMACENADOS EN 'A', PRESCINDIENDO DE AQUELLOS
00065 C CUYO VALOR SEA 'D' (LOS DUMMIES).
00066 DIMENSION A(1)
00067 DO 1 I=1,N
00068 IF(A(I).NE.D) GO TO 2
00069 1 CONTINUE
00070 WRITE(6,99)
00071 99 FORMAT("TODOS LOS VALORES CONSIDERADOS SON 'DUMMIES'")
00072 RETURN
00073 2 VMIN=A(I)
00074 DO 3 J=I,N
00075 IF(A(J).EQ.D) GO TO 3
00076 IF(A(J)-VMIN) 4,3
00077 4 VMIN=A(J)
00078 3 CONTINUE
00079 RETURN
00080 END
00081 C *****
00082 SUBROUTINE CHEBY(X,Y,N,IGRAD,IRES,COE) ,83.07.15
00083 C SUBROUTINA PARA AJUSTE DE POLINOMIOS DE CHEBYSHEV.
00084 C SE HAN ELIMINADO LOS POSIBLES MENSAJES DE ERROR.
00085 DIMENSION X(1),Y(1),COE(1),WORK(45)
00086 IP=IGRAD+1
00087 CALL APCH1(X,Y,N,IP,XD,X0,WORK,IER)
00088 EPS=1E-6
00089 ETA=1E-6
00090 IOP=-1
00091 CALL APFS(WORK,IP,IRES,IOP,EPS,ETA,IER)
00092 IPOSN=(IRES-1)*IRES/2
00093 DO 6 I=1,IRES
00094 6 COE(I)=WORK(I+IPOSN)
00095 COE(9)=X0
00096 COE(10)=XD
00097 RETURN
00098 END
00099 C *****
00100 SUBROUTINE HANND(V,N,D) ,83.07.15
00101 C SUAVIZADO HANNING DE 'N' VALORES CONSECUTIVOS COLOCADOS EN 'V'
00102 C ALGUNOS DE LOS CUALES PUEDEN SER DUMMIES, CON VALOR 'D'.
00103 C LOS EXTREMOS PASAN A SER A SU VEZ DUMMIES.
00104 C EN CASO DE ERROR, A LA SALIDA N=-1.
00105 DIMENSION V(1)
00106 IF(N.LT.3) GO TO 99
00107 NF=N-1
00108 DO 1 I=1,N
00109 IF(V(I).NE.D) GO TO 2
00110 1 CONTINUE
00111 GO TO 99
00112 2 NI=I+1
00113 A=V(I)
00114 V(I)=D
00115 DO 3 I=NI,NF
00116 B=V(I)
00117 IF(V(I).EQ.D) GO TO 3
00118 IF(V(I+1).EQ.D.AND.A.NE.D) V(I)=(A+2*V(I))/3
00119 IF(V(I+1).NE.D.AND.A.EQ.D) V(I)=(2*V(I)+V(I+1))/3
00120 IF(V(I+1).NE.D.AND.A.NE.D) V(I)=(A+2*V(I)+V(I+1))/4
```

```

00121      IF(V(I+1).EQ.D.AND.A.EQ.D) V(I)=V(I)
00122 3     A=B
00123      V(N)=D
00124      RETURN
00125 99    WRITE(6,100) N
00126 100   FORMAT("ERROR EN HANN:  N="IS" <3, O BIEN  V(I)=DUMMY PARA "
00127      " "TODO I=<N")
00128      N=-1
00129      RETURN
00130      END
00131 C *****
00132      SUBROUTINE GRAF3 (Y,NP,X0,Y0,DX,DY,FE,YMIN,DUM) ,83.07.28
00133 C
00134 C BASADA EN GRAFE (25-10-82).
00135 C GRAFICA LOS 'NP' VALORES CONTENIDOS EN 'Y' DE DERECHA A IZQUIER-
00136 C DA.
00137      DIMENSION Y(1)
00138 C
00139 C ESTA SUBROUTINA DIBUJA EN LA PANTALLA LA GRAFICA CORRESPONDIENTE
00140 C A LOS VALORES DE LA VARIABLE DIMENSIONADA <Y>. SE DIFERENCIA DE
00141 C <GRAF> EN QUE EL FACTOR DE ESCALA <FE> ES PARAMETRO DE ENTRADA
00142 C Y DAMOS EL VALOR QUE DESEAMOS SE CONSIDERE MINIMO PARA TENER LA
00143 C COLOCACION DESEADA. NO DIBUJA LAS LINEAS QUE SE SALGAN DEL RE-
00144 C CUADRO LIMITADO POR <X0,Y0,DX,DY>. NO DIBUJA BARRAS VERTICALES,
00145 C PERO DIBUJA EL NIVEL CERO SI CAE DENTRO DEL RANGO DEFINIDO.
00146 C NO DIBUJA LOS VALORES CONSIDERADOS 'DUMMIES'.
00147 C
00148      N=NP-1
00149      AX=-DX/N
00150      IY0=Y0
00151      IYF0=Y0+DY
00152      IYOF=Y0-YMIN*FE
00153      DO 30 I=1,N
00154      IF(Y(I+1).EQ.DUM) GO TO 30
00155      IF(Y(I).EQ.DUM) GO TO 30
00156      IXI=(I-1)*AX+X0
00157      IXF=I*AX+X0
00158      IYI=Y(I)*FE+IYOF
00159      IYF=Y(I+1)*FE+IYOF
00160      AY=IYF-IYI
00161      IF (IYI-IY0) 13,10
00162 10      IF (IYF0-IYI) 12,11
00163 11      IF (IYF-IY0) 7,14
00164 14      IF (IYF0-IYF) 4,1
00165 12      IF (IYF-IY0) 8,15
00166 15      IF (IYF0-IYF) 30,5
00167 13      IF (IYF-IY0) 30,16
00168 16      IF (IYF0-IYF) 9,6
00169 4      NXF=AX*(IYF0-IYI)/AY+IXI
00170      CALL PLOT (1,IXI,IYI,NXF,IYF0)
00171      GO TO 30
00172 7      NXF=AX*(IY0-IYI)/AY+IXI
00173      CALL PLOT (1,IXI,IYI,NXF,IY0)
00174      GO TO 30
00175 1      CALL PLOT (1,IXI,IYI,IXF,IYF)
00176      GO TO 30
00177 5      NXI=AX*(IYF0-IYI)/AY+IXI
00178      CALL PLOT (1,NXI,IYF0,IXF,IYF)
00179      GO TO 30
00180 8      NXI=AX*(IYF0-IYI)/AY+IXI

```

```

00181 NXF=AX*(IY0-IYI)/AY+IXI
00182 CALL PLOT (1,NXI,IYF0,NXF,IY0)
00183 GO TO 30
00184 6 NXI=AX*(IY0-IYI)/AY+IXI
00185 CALL PLOT (1,NXI,IY0,IXF,IYF)
00186 GO TO 30
00187 9 NXI=AX*(IY0-IYI)/AY+IXI
00188 NXF=AX*(IYF0-IYI)/AY+IXI
00189 CALL PLOT (1,NXI,IY0,NXF,IYF0)
00190 30 CONTINUE
00191 IF (IYOF.GT.IYF0.OR.IYOF.LT.IY0) RETURN
00192 IXI=X0
00193 CALL PLOT (1,IXI,IYOF,IXF,IYOF)
00194 RETURN
00195 END
00196 C *****
00197 SUBROUTINE ACUMU(W1,V1,WI,WP,INI,IFI,ICORR,DUM,CON,X,NCODE),
00198 , 83.08.01
00199 C ACUMULA LOS VALORES DE 'V1' CON PESO 'WP' EN 'W1' CON PESO 'WI'
00200 C PARA LOS VALORES COMPREDIDOS ENTRE 'INI' Y 'IFI' Y CON UN
00201 C CORRIMIENTO 'ICORR'. SE INTERPOLAN O SE EVITAN LOS VALORES
00202 C IGUALES A 'DUM'. 'CON' ES UN CONTADOR, 'X' UN VECTOR AUXILIAR
00203 C Y 'NCODE' INDICA SI HAN DE CONSERVARSE LOS VALORES EXTREMOS
00204 C NO PROMEDIADOS (NCODE=-N) O DEBEN DESECHARSE (NCODE=N).
00205 DIMENSION W1(1),V1(1),X(1)
00206 WE=W1+WP
00207 IF(CON.NE.1) GO TO 1818
00208 C PRIMER ESPECTRO A PROMEDIAR
00209 DO 1817 I=INI,IFI
00210 1817 X(I)=V1(I)
00211 GO TO 1919
00212 C LOS DEMAS ESPECTROS; PUEDEN ESTAR CORRIDOS
00213 1818 DO 1262 I=INI,IFI
00214 IMC=I+ICORR
00215 T=DUM
00216 IF(V1(IMC).NE.DUM.AND.W1(I).NE.DUM) GO TO 12631
00217 IF(V1(IMC).EQ.DUM.AND.W1(I).NE.DUM) GO TO 12632
00218 IF(V1(IMC).NE.DUM.AND.W1(I).EQ.DUM) GO TO 12633
00219 C VALOR DUMMY EN AMBOS ESPECTROS: DA LUGAR A DUMMY
00220 GO TO 1262
00221 C CASO NORMAL
00222 12631 T=(WI*W1(I)+WP*V1(IMC))/WE
00223 GO TO 1262
00224 C VALOR DUMMY EN EL ESPECTRO INDIVIDUAL
00225 12632 IF(IMC.NE.1) GO TO 12634
00226 IF(V1(IMC+1).NE.DUM) T=(WI*W1(I)+WP*V1(IMC+1))/WE
00227 GO TO 1262
00228 12634 IF(V1(IMC-1).NE.DUM.AND.V1(IMC+1).NE.DUM)
00229 & T=(WI*W1(I)+WP*(V1(IMC-1)+V1(IMC+1)))/2)/WE
00230 IF(V1(IMC-1).EQ.DUM.AND.V1(IMC+1).NE.DUM)
00231 & T=(WI*W1(I)+WP*V1(IMC+1))/WE
00232 IF(V1(IMC-1).NE.DUM.AND.V1(IMC+1).EQ.DUM)
00233 & T=(WI*W1(I)+WP*V1(IMC-1))/WE
00234 GO TO 1262
00235 C VALOR DUMMY EN EL ESPECTRO PROMEDIO
00236 12633 IF(I.NE.1) GO TO 12635
00237 IF(W1(I+1).NE.DUM) T=(WI*W1(I+1)+WP*V1(IMC))/WE
00238 GO TO 1262
00239 12635 IF(W1(I-1).NE.DUM.AND.W1(I+1).NE.DUM)
00240 & T=(WI*(W1(I-1)+W1(I+1)))/2+WP*V1(IMC))/WE

```



```

00241      IF(W1(I-1).EQ.DUM.AND.W1(I+1).NE.DUM)
00242      & T=(W1*W1(I+1)+WP*V1(IMC))/WE
00243      IF(W1(I-1).NE.DUM.AND.W1(I+1).EQ.DUM)
00244      & T=(W1*W1(I-1)+WP*V1(IMC))/WE
00245 1262  X(I)=T
00246 C
00247 1919  WI=WE
00248      IF(NCODE) 1,2
00249 C SE CONSERVAN LOS VALORES EXTREMOS; SOLO PUEDE HACERSE ESTO CUANDO
00250 C SE ESTAN PROMEDIANDO ESPECTROS AJUSTADOS (VALOR MEDIO NULO EN LOS
00251 C EXTREMOS)
00252 1      DO 12667 I=INI,IFI
00253 12667  W1(I)=X(I)
00254      RETURN
00255 C LOS VALORES EXTREMOS NO CONSIDERADOS SON CONVERTIDOS EN DUMMIES.
00256 C (CASO EN QUE SE PROMEDIAN ESPECTROS ANTES DE AJUSTARLOS)
00257 2      DO 12666 I=1,NCODE
00258      W1(I)=DUM
00259      IF(I.GE.INI.AND.I.LE.IFI) W1(I)=X(I)
00260 12666  CONTINUE
00261      RETURN
00262      END
00263 C *****
00264      SUBROUTINE CLIP(A,N,D,ICLIP) ,83.08.04
00265 C ASIGNA EL VALOR 'D' (DUMMY) A TODOS AQUELLOS VALORES
00266 C ENTRE LOS 'N' PRIMEROS DEL VECTOR 'A' QUE DIFIERAN DE
00267 C SU PROMEDIO EN MAS DE 'ICLIP' VECES LA DESVIACION
00268 C TIPICA.
00269      DIMENSION A(1)
00270      K=0
00271      S=0
00272      S2=0
00273      DO 1 I=1,N
00274      T=A(I)
00275      IF(T.EQ.D) GO TO 1
00276      S=S+T
00277      S2=S2+T*T
00278      K=K+1
00279 1      CONTINUE
00280      IF(K.EQ.0) RETURN
00281      AVE=S/K
00282      SIG=SQRT(S2/K-AVE*AVE)
00283      COTAS=AVE+SIG*ICLIP
00284      COTAI=AVE-SIG*ICLIP
00285      DO 2 I=1,N
00286      IF(A(I).EQ.D) GO TO 2
00287      IF(A(I).GT.COTAS.OR.A(I).LT.COTAI) A(I)=D
00288 2      CONTINUE
00289      RETURN
00290      END
00291 C *****
00292      SUBROUTINE NUMBI (X0,Y0,DX,NP,IBR) ,83.07.21
00293 C
00294 C BASADA EN NUMBR (16-8-82), NUMERA DE DERECHA A IZQUIERDA
00295 C
00296 C NUMERACION DE LAS BARRAS DIBUJADAS
00297 C CON GRAF. PARA CONSEGUIR LA POSICION
00298 C DESEADA PODEMOS JUGAR MODIFICANDO X0
00299 C E Y0. EL SIGNIFICADO DE LOS PARAMETROS
00300 C ES EL MISMO QUE EN GRAF3.

```

```
00301 C
00302 IXF=0
00303 F=-DX/(NP-1)
00304 IY=Y0
00305 I=NP/IBR*IBR
00306 20 IF (I.LT.1) RETURN
00307 LOG=ALOGT(FLOAT(I)+.5)+1
00308 IX=X0+(I-1)*F-LOG*14-9
00309 IF (IX.LE.IXF) GO TO 10
00310 IXF=IX+LOG*14+4
00311 CALL PLOT (0,IX,IY)
00312 GO TO (1,2,3,4),LOG
00313 1 WRITE (6,101) I
00314 101 FORMAT (I1)
00315 GO TO 10
00316 2 WRITE (6,102) I
00317 102 FORMAT (I2)
00318 GO TO 10
00319 3 WRITE (6,103) I
00320 103 FORMAT (I3)
00321 GO TO 10
00322 4 WRITE (6,104) I
00323 104 FORMAT (I4)
00324 10 I=I-IBR
00325 GO TO 20
00326 END
00327 C *****
00328 SUBROUTINE PLOT (K,IXI,IYI,IXF,IYF) ,25-8-82
00329 C
00330 DIMENSION ICAR(5)
00331 C
00332 C SI K#0 DIBUJA LA LINEA DADA POR IXI, IYI, IXF, IYF.
00333 C
00334 C SI K=0 POSICIONA EL CURSOR EN EL PUNTO IXI,IYI.
00335 C
00336 ICAR(1)=IOR(7424,32+IYI/32)
00337 ICAR(2)=IOR((96+IYI-32*(IYI/32))*256,32+IXI/32)
00338 ICAR(3)=(64+IXI-32*(IXI/32))*256
00339 IF (K.EQ.0) GO TO 1
00340 ICAR(3)=IOR(ICAR(3),32+IYF/32)
00341 ICAR(4)=IOR((96+IYF-32*(IYF/32))*256,32+IXF/32)
00342 ICAR(5)=IOR((64+IXF-32*(IXF/32))*256,31)
00343 CALL EXEC (2,506B,ICAR,5)
00344 RETURN
00345 1 ICAR(3)=IOR(ICAR(3),31)
00346 CALL EXEC (2,506B,ICAR,3)
00347 RETURN
00348 END
00349 C *****
00350 SUBROUTINE ERFMP (I,IERR), 23-12-82
00351 C
00352 C ANALIZA <IERR> DESPUES DE USAR ALGUNA <FMP CALLS>.
00353 C <I> ES UNA ETIQUETA QUE PONEMOS PARA ENCONTRAR LA
00354 C POSICION DEL ERROR EN EL PROGRAMA.
00355 C
00356 IF (IERR) 1,2
00357 1 WRITE (6,100) I,IERR
00358 100 FORMAT (/I5,5X"FMP ERROR CODE:"I4,5X"_"")
00359 PAUSE
00360 2 RETURN
```

```

00361      END
00362 C *****
00363      SUBROUTINE AXISI (IX,IY,AX,LX,AY,LY,LTX,LTY)      ,83.07.21
00364 C
00365 C EXTRAIDA DE 'AXIS' (26-8-82), TOMA COMO ORIGEN EL PUNTO
00366 C SUPERIOR DERECHO DEL RECUADRO Y DIBUJA EJES Y TRAZOS EN
00367 C LOS CUATRO LADOS.
00368 C
00369 C IX,IY: ORIGEN DE LOS EJES (PUNTO SUPERIOR DERECHO)
00370 C AX,AY: DISTANCIA ENTRE LAS DIVISIONES EN LOS EJES X E Y
00371 C LX,LY: LONGITUD DE LOS EJES
00372 C LTX,LTY: LONGITUD DE LOS TRAZOS EN LOS EJES X E Y
00373 C
00374      CALL PLOT(1,IX,IY,IX-LX,IY)
00375      CALL PLOT(1,IX-LX,IY,IX-LX,IY-LY)
00376      CALL PLOT(1,IX-LX,IY-LY,IX,IY-LY)
00377      CALL PLOT(1,IX,IY-LY,IX,IY)
00378      NX=LX/AX
00379      NY=LY/AY
00380      DX=-AX
00381      DY=-AY
00382      IYF=IY-LTX
00383      DO 10 I=1,NX
00384      IXI=IX+DX*I-.49999
00385 10    CALL PLOT (1,IXI,IY,IXI,IYF)
00386      IXF=IX+LTY-LX
00387      DO 20 I=1,NY
00388      IYI=IY+DY*I-.49999
00389 20    CALL PLOT (1,IX-LX,IYI,IXF,IYI)
00390      IYF=IY-LY+LTX
00391      IOF=IX+DX*(NX+1)
00392      DO 30 I=1,NX
00393      IXI=IOF-DX*I+.49999
00394 30    CALL PLOT(1,IXI,IY-LY,IXI,IYF)
00395      IXF=IX-LTY
00396      IOF=IY+DY*(NY+1)
00397      DO 40 I=1,NY
00398      IYI=IOF-DY*I+.49999
00399 40    CALL PLOT(1,IX,IYI,IXF,IYI)
00400      RETURN
00401      END
00402 C *****
00403      SUBROUTINE APCH1(X,Y,N,IP,XD,X0,W,IER)
00404 C -----
00405 C
00406 C GENERA LAS ECUACIONES NORMALES DE UN AJUSTE POR MINIMOS CUADRADOS
00407 C DE UNA FUNCION DISCRETA DADA POR POLINOMIOS DE CHEBYSHEV.
00408 C
00409 C X      VECTOR DE DIMENSION N QUE CONTIENE LAS ABSCISAS DE LOS PUNTOS
00410 C Y      VECTOR DE DIMENSION N QUE CONTIENE LAS ORDENADAS DE LOS PUNTOS
00411 C N      NUMERO DE PUNTOS A AJUSTAR
00412 C IP     DIMENSION DEL AJUSTE, O SEA NUMERO DE POLINOMIOS DE CHEBYSHEV
00413 C        UTILIZADOS. IP NO PUEDE SER MAYOR QUE N
00414 C XD     CONSTANTE MULTIPLICATIVA RESULTANTE PARA LA TRANSFORMACION
00415 C        LINEAL DEL RANGO DEL ARGUMENTO
00416 C X0     CONSTANTE ADITIVA RESULTANTE PARA LA TRANSFORMACION LINEAL
00417 C        DEL RANGO DEL ARGUMENTO
00418 C W      VECTOR AUXILIAR DE DIMENSION (IP+1)*(IP+2)/2. A LA SALIDA
00419 C        CONTIENE LA MATRIZ SIMETRICA DE COEFICIENTES DE LAS ECUA-
00420 C        CIONES NORMALES EN FORMA COMPRIMIDA SEGUIDOS POR LOS TER-

```

```

00421 C      MINUS DE LA DERECHA Y LA SUMA DE LOS CUADRADOS DE LOS VA-
00422 C      LORES DE LA FUNCION.
00423 C IER  = -1  ERRORES FORMALES EN LA DIMENSION
00424 C      =  0  NO HAY ERRORES
00425 C      = +1  ARGUMENTOS COINCIDENTES
00426 C
00427 C OBSERVACION: NO SE UTILIZA PONDERADO DE LOS VALORES
00428 C
00429 C ORIGEN: SE TRATA DE UNA VERSION ESPECIAL USADA EN EL MPIFR ('POPS')
00430 C      DE LA SUBROUTINA 'APCH' DEL "SUBROUTINE SOFTWARE PACKAGE"
00431 C      DEL IBM-360.
00432 C
00433 C -----
00434 C
00435      DOUBLE PRECISION SUM
00436      DIMENSION X(1),Y(1),W(1)
00437 C BUSQUEDA DE ERRORES FORMALES EN LA DIMENSION
00438      IF(N-1) 19,20,1
00439 1      IF(IP) 19,19,2
00440 C ARGUMENTOS MENOR Y MAYOR
00441 2      IF(IP-N) 3,3,19
00442 3      XA=X(1)
00443      X0=X(N)
00444      XE=0
00445 C INICIALIZACIONES PARA EL CALCULO DE LAS ECUACIONES NORMALES
00446      XD=X0-XA
00447      M=(IP*(IP+1))/2
00448      IEND=M+IP+1
00449      MT2=IP+IP
00450      MT2M=MT2-1
00451 C PUESTA A CERO DEL VECTOR DE TRABAJO Y RESULTADOS
00452      DO 8 I=1,IP
00453          J=MT2-I
00454          W(J)=0
00455          W(I)=0
00456          K=M+I
00457 8      W(K)=0
00458 C CONTROL DEL RANGO DE ARGUMENTOS
00459      IF(XD) 20,20,9
00460 C CALCULO DE LAS CONSTANTES PARA LA REDUCCION DE ARGUMENTOS
00461 9      X0=-(X0+XA)/XD
00462      XD=2/XD
00463      SUM=0
00464 C INICIO DEL BUCLE PARA TODOS LOS PUNTOS DADOS
00465      DO 15 I=1,N
00466          T=X(I)*XD+X0
00467          DF=Y(I)
00468 C CALCULO Y ALMACENAMIENTO DE LOS VALORES DE LOS POLINOMIOS DE
00469 C CHEBYSHEV PARA EL ARGUMENTO T
00470      XA=1
00471      XM=T
00472 C (PONDERADO ELIMINADO RESPECTO DE 'APCH')
00473 11      T=T+T
00474          SUM=SUM+DF*DF
00475          DF=DF+DF
00476          J=1
00477 12      K=M+J
00478          W(K)=W(K)+DF*XA
00479 13      W(J)=W(J)+XA
00480          IF(J-MT2M) 14,15,15

```

```

00481 14      J=J+1
00482      XE=T*XM-XA
00483      XA=XM
00484      XM=XE
00485      IF(J-IP) 12,12,13
00486 15      CONTINUE
00487      W(IEND)=SUM+SUM
00488 C CALCULO DE LA MATRIZ DE LAS ECUACIONES NORMALES
00489      LL=M
00490      KK=MT2M
00491      JJ=1
00492      K=KK
00493      DO 18 J=1,M
00494          W(LL)=W(K)+W(JJ)
00495          LL=LL-1
00496          IF(K-JJ) 16,16,17
00497 16      KK=KK-2
00498          K=KK
00499          JJ=1
00500          GO TO 18
00501 17      JJ=JJ+1
00502          K=K-1
00503 18      CONTINUE
00504      IER=0
00505      RETURN
00506 C RETORNO EN CASO DE ERRORES FORMALES
00507 19      IER=-1
00508      RETURN
00509 C RETORNO EN CASO DE ARGUMENTOS COINCIDENTES
00510 20      IER=1
00511      RETURN
00512      END
00513 C *****
00514 C
00515      SUBROUTINE APFS(W,IP,IRES,IOP,EPS,ETA,IER)
00516 C
00517 C-----
00518 C
00519 C REALIZA LA FACTORIZACION SIMETRICA DE LA MATRIZ DE LAS ECUACIONES
00520 C NORMALES SEGUIDA, OPCIONALMENTE, DEL CALCULO DEL AJUSTE POR
00521 C MINIMOS CUADRADOS.
00522 C
00523 C W      MATRIZ SIMETRICA DE COEFICIENTES DADA. SU DIMENSION TOTAL ES
00524 C      (IP+1)*(IP+2)/2
00525 C IP     NUMERO DE FUNCIONES UTILIZADAS PARA EL AJUSTE
00526 C IRES   NUMERO DE FUNCIONES REALMENTE UTILIZADAS EN EL AJUSTE
00527 C      (DEPENDE DE IP,ETA,EPS,IOP)
00528 C IOP = 0 FACTORIZACION TRIANGULAR, DIVISION DEL MIEMBRO DERECHO
00529 C      POR LA TRANSPUESTA DE LA MATRIZ Y CALCULO DE LA SUMA DE
00530 C      LOS CUADRADOS DE LOS ERRORES
00531 C      =+-1 ADEMAS SE CALCULA LA SOLUCION DE DIMENSION IRES
00532 C      =+-2 ADEMAS SE CALCULAN TODAS LAS SOLUCIONES DE DIMENSION
00533 C      1 HASTA IRES.
00534 C EPS   TOLERANCIA RELATIVA PARA EL TEST DE PERDIDA DE SIGNIFICADO
00535 C      (1E-3< <1E-6)
00536 C ETA   TOLERANCIA RELATIVA PARA LA SUMA DE LOS CUADRADOS DE LOS
00537 C      CUADRADOS DE LOS ERRORES TOLERADA (1E0< <1E-6)
00538 C IER = -1 IP<0
00539 C      = 0 NO SE HA DETECTADO PERDIDA DE SIGNIFICACION Y LA TOLERANCIA
00540 C      ESPECIFICADA PARA LOS ERRORES HA SIDO ALCANZADA

```

```

00541 C      = +1  SE HA DETECTADO PERDIDA DE SIGNIFICACION O LA TOLERANCIA
00542 C          ESPECIFICADA PARA LOS ERRORES NO HA SIDO ALCANZADA.
00543 C
00544 C  ORIGEN: "SUBROUTINE SOFTWARE PACKAGE" DEL IBM-360
00545 C -----
00546 C
00547          DOUBLE PRECISION SUM,TOL,PIV
00548          DIMENSION W(1)
00549          IRES=0
00550 C  TEST DE LA DIMENSION ESPECIFICADA
00551          IF(IP.GT.0) GO TO 2
00552          IER=-1
00553          RETURN
00554 C  INICIO DEL PROCESO DE FACTORIZACION
00555 2          IPIV=0
00556          IPP1=IP+1
00557          IER=1
00558          ITE=IP*IPP1/2
00559          IEND=ITE+IPP1
00560          TOL=ABS(EPS*W(1))
00561          TEST=ABS(ETA*W(IEND))
00562 C  BUCLE SOBRE TODAS LAS FILAS DE W
00563          DO 11 I=1,IP
00564              IPIV=IPIV+I
00565              JA=IPIV-IRES
00566              JE=IPIV-1
00567 C  PRODUCTO ESCALAR NECESARIO PARA MODIFICAR LOS ELEMENTOS COLUMNA
00568              JK=IPIV
00569              DO 9 K=I,IPP1
00570                  SUM=0
00571                  IF(IRES) 5,5,3
00572 3                  JK=JK-IRES
00573                  DO 4 J=JA,JE
00574                      SUM=SUM+W(J)*W(JK)
00575 4                      JK=JK+1
00576 5                      IF(JK-IPIV) 6,6,8
00577 C  TEST PARA PERDIDA DE SIGNIFICANCIA
00578 6                      SUM=W(IPIV)-SUM
00579                      IF(SUM-TOL) 12,12,7
00580 7                      SUM=DSQRT(SUM)
00581                      W(IPIV)=SUM
00582                      PIV=1./SUM
00583                      GO TO 9
00584 C  ACTUALIZAR LOS TERMINOS DE FUERA DE LA DIAGONAL
00585 8                      SUM=(W(JK)-SUM)*PIV
00586                      W(JK)=SUM
00587 9                      JK=JK+K
00588 C  ACTUALIZAR LA SUMA DE LOS CUADRADOS DE LOS ERRORES
00589                      W(IEND)=W(IEND)-SUM*SUM
00590 C  DIRECCION DE MEMORIA DEL ULTIMO ELEMENTO PIVOT
00591                      IRES=IRES+1
00592                      IADR=IPIV
00593 C  TEST PARA EL ERROR TOLERABLE, SI FUE ESPECIFICADO
00594                      IF(IOP) 10,11,11
00595 10                     IF(W(IEND)-TEST) 13,13,11
00596 11                     CONTINUE
00597                      IF(IOP) 12,22,12
00598 C  RESOLUCION, SI SE ESPECIFICO
00599 12                     IF(IOP) 14,23,14
00600 13                     IER=0

```

```

00601 14      IPIV=IRES
00602 15      IF(IPIV) 23,23,16
00603 16      SUM=0
00604        JA=ITE+IPIV
00605        JJ=IADR
00606        JK=IADR
00607        K=IPIV
00608        DO 19 I=1,IPIV
00609          W(JK)=(W(JA)-SUM)/W(JJ)
00610          IF(K-1) 20,20,17
00611 17      JE=JJ-1
00612          SUM=0
00613          DO 18 J=K,IPIV
00614            SUM=SUM+W(JK)*W(JE)
00615            JK=JK+1
00616 18      JE=JE+J
00617            JK=JE-IPIV
00618            JA=JA-1
00619            JJ=JJ-K
00620 19      K=K-1
00621 20      IF(IOP/2) 21,23,21
00622 21      IADR=IADR-IPIV
00623        IPIV=IPIV-1
00624        GO TO 15
00625 C RETORNO NORMAL
00626 22      IER=0
00627 23      RETURN
00628        END
00629 C *****
00630 C
00631        SUBROUTINE CNPS(Y,X,C,N)
00632 C-----
00633 C
00634 C CALCULA EL VALOR Y DEL N-ESIMO TERMINO DE UNA DESARROLLO EN
00635 C POLINOMIOS DE CHEBYSHEV CON VECTOR DE COEFICIENTES C PARA
00636 C UN ARGUMENTO X
00637 C
00638 C ORIGEN: "SUBROUTINE SOFTWARE PACKAGE" DEL IBM-360
00639 C-----
00640 C
00641        DIMENSION C(1)
00642 C TEST PARA LA DIMENSION
00643        IF(N-2) 3,4,4
00644 3        Y=C(1)
00645        RETURN
00646 C INICIALIZACION
00647 4        ARG=X+X
00648        H1=0
00649        H0=0
00650 C
00651        DO 5 I=1,N
00652          K=N-I
00653          H2=H1
00654          H1=H0
00655 5        H0=ARG*H1-H2+C(K+1)
00656          Y=0.5*(C(1)-H2+H0)
00657        RETURN
00658        END
00659        END$

```

ANEXO C

=====

LISTADOS DE LOS PROGRAMAS COMPLEMENTARIOS



```

00001 FTN,L,B
00002 C   PROGRAMA QUE PERMITE EL TRASLADO DE LOS DATOS OBTENIDOS
00003 C   CON EL RECEPTOR MULTICANAL ('MCHRCVR') EN LA ESTACION
00004 C   ESPACIAL DE MADRID Y GRABADOS EN 'ASCII' A OTRA CINTA
00005 C   MAGNETICA CON FORMATO BINARIO.
00006 C   SE UTILIZA COMO INTERMEDIARIO UN FICHERO EN EL DISCO
00007 C   QUE ES CREADO POR EL MISMO PROGRAMA.
00008 C   EL CONJUNTO DE FASES PUEDE DIVIDIRSE EN VARIOS FICHEROS
00009 C   GRACIAS A ESTE PROCESO, CORRESPONDIENDO CADA UNO DE
00010 C   ELLOS A LOS CICLOS CONSECUTIVOS REALIZADOS EN UN MISMO
00011 C   PUNTO DEL CIELO O EN UNAS MISMAS CONDICIONES.
00012 C   LA UNIDAD DE CINTA SE UTILIZA DE FORMA EQUIVALENTE A
00013 C   UN FICHERO EN DISCO HABIENDOLE DADO, PREVIAMENTE AL USO
00014 C   DE ESTE PROGRAMA, UN NOMBRE, DE LA FORMA:
00015 C       CR,MAGT,8,WR,BO,EO,BI
00016 C       =====
00017 C
00018 C   PROGRAM TRASL (),1 JULIO 1982
00019 C   DIMENSION IDCB(291),NAMED(3),IDCC(291),NAMEC(3),ISIZE(2)
00020 C   DIMENSION NDAT(17),TTR(1),BUFF(128),IDAT(275)
00021 C   WRITE(6,10)
00022 10  FORMAT(// " COLOCADA CINTA 'NASA' ? (1=SI): _")
00023 C   READ(1,*) K
00024 C   IF(K.NE.1) PAUSE 1
00025 3   IF(LOCAL(8)) 1,2
00026 1   WRITE(6,20)
00027 20  FORMAT(" UNIDAD DE CINTA EN <LOCAL> _")
00028 C   PAUSE
00029 C   GO TO 3
00030 2   WRITE(6,32)
00031 32  FORMAT(" NOMBRE FICHERO DISCO A CREAR ? (<= 6 CARACT) : _")
00032 C   READ(1,31) NAMED
00033 31  FORMAT(3A2)
00034 C   ISIZE(1) = 24
00035 C   CALL CREAT(IDCB,IERR,NAMED,ISIZE,3,0,11)
00036 C   IF(IERR.LT.0) GO TO 99
00037 C   CALL MOVE(1)
00038 C   WRITE(6,43)
00039 43  FORMAT(// " NUMERO TOTAL DE FASES A LEER ? : _")
00040 C   READ(1,*) NTF
00041 C   DO 4 JF=1,NTF
00042 C   READ(8,40) NDAT,TTR
00043 40  FORMAT(3X,I3,1X,I1,I2,6A2,2I5,I4,I3,1X,3I2,I1,F4.1)
00044 C   READ(8,41) BUFF
00045 41  FORMAT(16F8.4)
00046 C   DO 5 I=1,275
00047 5   IDAT(I)=NDAT(I)
00048 C   CALL WRITF(IDCB,IERR,IDAT,275)
00049 C   IF(IERR.LT.0) GO TO 99
00050 4   CONTINUE
00051 C   CALL RWNDF(IDCB)
00052 C   REWIND 8
00053 C   WRITE(6,50)
00054 50  FORMAT(// " QUITAR CINTA 'NASA'." /
00055 C   * " COLOCAR CINTA 'CAY' CORRESPONDIENTE." /
00056 C   * " ATENCION!: 'WRITE ENABLE' DEBE ESTAR ENCENDIDO."//)
00057 C   PAUSE
00058 C   WRITE(6,30)
00059 30  FORMAT(" NOMBRE DEL FICHERO CINTA YA CREADO ? : _")
00060 C   READ(1,31) NAMEC

```

```
00061      CALL MOVE(0)
00062      CALL OPEN(IDCC,IERR,NAMED)
00063      IF(IERR.LT.0) GO TO 99
00064 6      WRITE(6,55)
00065 55     FORMAT(" NUMERO DE FASES EN EL PROXIMO FICHERO EN CINTA ?"
00066      1" (0=FIN): _")
00067      READ(1,*) NF
00068      IF(NF) 9,9,61
00069 61     DO 7 K=1,NF
00070      CALL READF(IDCIB,IERR,IDAT)
00071      IF(IERR.LT.0) GO TO 99
00072      CALL WRITF(IDCC,IERR,IDAT,275)
00073      IF(IERR.LT.0) GO TO 99
00074 7      CONTINUE
00075      WRITE(6,60) (IDAT(I),I=1,13)
00076 60     FORMAT(" ULTIMA FASE: "3I4,2X,6A2,2X,4I5///
00077      *" END-OF-FILE ? (1=SI): _")
00078      READ(1,*) K
00079      IF(K.EQ.1) ENDFILE 8
00080      GO TO 6
00081 9      CALL CLOSE(IDCIB,IERR)
00082      IF(IERR.LT.0) GO TO 99
00083      CALL PURGE(IDCIB,IERR,NAMED)
00084      CALL CLOSE(IDCC,IERR)
00085      IF(IERR.LT.0) GO TO 99
00086      REWIND 8
00087      STOP
00088 99     WRITE(6,999) IERR
00089 999     FORMAT(///" IERR = " I6///)
00090      END
00091      END$
```

```
00001 FTN,L
00002 PROGRAM TRAS2
00003 C
00004 C TRaslADO DE DATOS DE LA CINTA "OBREROS -S/F"
00005 C (OBSERVACIONES DE RRR, FORMATO BINARIO)
00006 C A UN FICHERO "SCRATCH" EN DISCO PARA SU ANALISIS.
00007 C
00008 C NUMERO MAXIMO DE FASES = 200.
00009 C
00010 C 83.04.12
00011 DIMENSION IDAT(275),NAME(3),ISIZE(2)
00012 DIMENSION IDCB(291)
00013 WRITE(6,900)
00014 900 FORMAT("NUMERO MAXIMO DE FASES : 200"///"NOMBRE DEL"
00015 "" FICHERO ? : _")
00016 READ(1,910) NAME
00017 910 FORMAT(3A2)
00018 7 WRITE(6,920)
00019 920 FORMAT("HAY QUE CREAR UNO NUEVO ( 1 ) O MACHACAR UNO"
00020 "" ANTIGUO ( 0 )? _")
00021 READ(1,*) ICR
00022 IF(ICR.NE.1.AND.ICR.NE.0) GO TO 7
00023 ISIZE(1) = 600
00024 ISIZE(2) = 275
00025 IF(ICR.EQ.1) CALL CREAT(IDCB,IE,NAME,ISIZE,2)
00026 IF(ICR.EQ.0) CALL OPEN(IDCB,IE,NAME)
00027 CALL ERFMP(1,IE)
00028 WRITE(6,200)
00029 200 FORMAT("NUMERO DE FASES/CICLO =_")
00030 READ(1,*) NFNC
00031 NCICT = 0
00032 3 CALL MOVE(1)
00033 WRITE(6,100)
00034 100 FORMAT("NUMERO DE CICLOS A LEER ? (FIN = 0) :_")
00035 READ(1,*) NCIC
00036 IF(NCIC.EQ.0) GO TO 99
00037 NFAS = NCIC*NFNC
00038 NCICT = NCICT + NCIC
00039 IF(NCICT*NFNC.GT.200) GO TO 919
00040 DO 1 I=1,NFAS
00041 CALL LEE(IDAT)
00042 CALL WRITF(IDCB,IE,IDAT,275)
00043 1 CONTINUE
00044 GO TO 3
00045 919 WRITE(6,38)
00046 38 FORMAT("NUMERO DE CICLOS SUPERIOR A LOS QUE CABEN EN EL"
00047 "" FICHERO DISCO"/"ESTOS ULTIMOS NO SON TENIDOS EN"
00048 "" CUENTA"//)
00049 99 WRITE(6,400)
00050 400 FORMAT("REWIND ? (SI = 1) : _")
00051 READ(1,*) IRW
00052 IF(IRW.EQ.1) REWIND 8
00053 WRITE(6,500)
00054 500 FORMAT("LISTADO DEL FICHERO ? (SI = 1) : _")
00055 READ(1,*) ILIST
00056 IF(ILIST.NE.1) GO TO 98
00057 WRITE(6,501)
00058 501 FORMAT("PUEDES CONECTAR EL TELETIPO. CONTESTA CUALQUIER COSA")
00059 READ(1,910) NAME
00060 CALL RWNDF(IDCB,IE)
```

```

00061      CALL ERFMP(2,IE)
00062      DO 21 I=1,NCICT
00063      K = (I-1)*NFNC+1
00064      DO 20 J=1,NFNC
00065      CALL READF(IDCIB,IE,1DAT)
00066 20    CALL ERFMP(3,IE)
00067 21    WRITE(6,600) K,(1DAT(J),J=4,9),I
00068 600    FORMAT("REGISTRO "I4,5X,6A2,5X"CICLO "I4)
00069 98    CALL CLOSE(IDCIB,IE)
00070      CALL ERFMP(4,IE)
00071      END
00072 C*****
00073      SUBROUTINE LEE(1DAT)
00074      DIMENSION 1DAT(275),IDUM(2)
00075      K = -1
00076 1      CALL EXEC(1,200,1DAT,275)
00077      WRITE(6,200) 1DAT(1),1DAT(2)
00078 200    FORMAT("CICLO "I3" FASE "I2)
00079      IF(IEOF(8)) 2,3
00080 2      IF(K) 23,5
00081 23     CALL PTAPE(8,-2,0)
00082      K = 0
00083      GO TO 1
00084 5      WRITE(6,100)
00085 100    FORMAT("CICLO O FASE NO ENCONTRADO")
00086      CALL PTAPE(8,0,1)
00087 3      RETURN
00088      END
00089 $
    
```

```
00001 FTN
00002 PROGRAM MCH2 (),83.10.27
00003 C PROGRAMA PARA EL CALCULO DE ESPECTROS A PARTIR DE LOS DATOS
00004 C PREVIAMENTE GRABADOS EN DISCO A PARTIR DE LA CINTA "CEBREROS
00005 C S/F" MEDIANTE EL PROGRAMA "TRAS2" (VERSION 83.04.12).
00006 C EL FORMATO DEL FICHERO DE ESPECTROS ES EL QUE SE UTILIZA
00007 C CON EL PROGRAMA "RRL" DE ANALISIS DE RAYAS DE RECOMBINACION,
00008 C QUE A SU VEZ SE HA HECHO EN FUNCION DEL FORMATO EN QUE VIENEN
00009 C LOS DATOS DE TAL TIPO DE OBSERVACION PROVENIENTES DE
00010 C "EFFELSBURG".
00011 C SE SUPONE HABITUALMENTE QUE EL FICHERO DE DATOS VA A SER
00012 C PURGADO A CONTINUACION, AUNQUE ES POSIBLE CONSERVARLO,
00013 C ASI COMO ES POSIBLE REALIZAR EL PROCESO DE LLENADO DE UN
00014 C FICHERO DE ESPECTROS EN VARIAS SESIONES.
00015 C SE REALIZAN UNA SERIE DE CALCULOS Y SE DEFINEN PARAMETROS
00016 C SUPUESTA LA OBSERVACION EN CEBREROS O ROBLEDO DE LA RAYA
00017 C H142ALFA.
00018 C SE CALCULAN LOS ESPECTROS SUPUESTAS OBSERVACIONES EN
00019 C CONMUTACION DE FRECUENCIA, DE FASES:
00020 C 1=F1 2=F1+TDR 3=F2 4=F2+TDR
00021 C Y USANDO LAS CALIBRACIONES: (1-3)/(2-1) Y (3-1)/(4-3),
00022 C CORRESPONDIENDO, EN ESTE ORDEN, A LOS ESPECTROS 1 Y 2 DEL
00023 C FICHERO. SE SUPONE QUE EL NUMERO DE CANALES ES 128 Y
00024 C QUE EL CORRIMIENTO ES 64.
00025 DIMENSION NAME(3),NAMER(3),IDCC(784),ISIZE(2)
00026 DIMENSION NOMDB(6),CIF(10),CAE(10)
00027 DIMENSION IBF(768),S1(128),S2(128),R(112)
00028 COMMON IDCB(291),ICLO,IFAS,IEST,NOMBR(6),INTN,INTR
00029 COMMON IDAT(5),ITIPO,TTR,V1(128),IDUM2(19),V2(128)
00030 COMMON IDUM3(19),V3(128),IDUM4(19),V4(128)
00031 EQUIVALENCE (IBF(257),S1),(IBF(385),S2),(IBF(19),R),(IBF(6),DUM)
00032 EQUIVALENCE (IBF(10),NOMDB)
00033 C
00034 WRITE(6,535)
00035 535 FORMAT("MCH2 VERSION 83.10.27"/)
00036 MPS=20
00037 TP100=80
00038 BANDA = 50
00039 WRITE(6,999)
00040 999 FORMAT("DATOS TOMADOS POR SEGUNDO"/"PORCENTAJE DE TIEMPO"
00041 "" EFECTIVO EN DICHA TOMA DE DATOS"/
00042 ""ANCHO DE BANDA POR CANAL (KHZ)")
00043 READ(1,*) MPS,TP100,BANDA
00044 C
00045 C ENTRADA DE DATOS REFERENTES A LOS FICHEROS.
00046 C NO HAY CREACION AUTOMATICA DE FICHERO DE RESULTADOS.
00047 C EL FICHERO DE RESULTADOS ES CREADO EN EL MISMO FICHERO
00048 C QUE EL DE DATOS, LUEGO SU NOMBRE DEBE SER DISTINTO.
00049 9184 WRITE(6,998)
00050 998 FORMAT("NOMBRE DEL FICHERO DE DATOS : _")
00051 READ(1,4561) NAME
00052 4561 FORMAT(3A2)
00053 CALL OPEN(IDCB,IE,NAME)
00054 CALL ERFMP(1,IE)
00055 WRITE(6,3391)
00056 3391 FORMAT("NOMBRE DEL FICHERO DE RESULTADOS _")
00057 READ(1,4561) NAMER
00058 WRITE(6,4562)
00059 4562 FORMAT("HAY QUE CREARLO (1) O SOLO ABRIRLO (0) ? _")
00060 ICR=0
```

```
00061 READ(1,*) ICR
00062 WRITE(6,3393)
00063 3393 FORMAT("CODIGO DE SEGURIDAD (2 LETRAS) : _")
00064 READ(1,3394) ISECU
00065 3394 FORMAT(A2)
00066 IF(ISECU.EQ.20040B) ISECU=0
00067 ISIZE(1) = 600
00068 ISIZE(2) = 768
00069 C CARTRIDGE
00070 ICAR=IAND(IDCB(1),77B)
00071 IF(ICR.EQ.1) CALL CREAT(IDCC,IE,NAMER,ISIZE,2,ISECU,ICAR)
00072 IF(ICR.EQ.0) CALL OPEN(IDCC,IE,NAMER,0,ISECU)
00073 CALL ERFMP(0,IE)
00074 IREG = 1
00075 IF(ICR.EQ.1) GO TO 2321
00076 WRITE(6,2320)
00077 2320 FORMAT("REGISTRO INICIAL FICH. RESULT.? _")
00078 READ(1,*) IREG
00079 CALL POSNT(IDCC,IE,IREG,1)
00080 2321 CALL EXEC(2,506B,6924,1)
00081 C INICIALIZACIONES
00082 VLSR=0
00083 10 TTDR = 0.0
00084 ICORR=0
00085 DUM=6666.667
00086 DO 1 I=1,10
00087 CIF(I) = 0.0
00088 1 CAE(I) = 0.0
00089 C** LECTURA DE DATOS
00090 WRITE(6,6542)
00091 6542 FORMAT("TEMPERATURA DEL SISTEMA (K) Y "
00092 ""TEMPERATURA DEL TUBO DE RUIDO (K)"/
00093 ""(0=FIN) _")
00094 READ(1,*) TSYS,TTDR
00095 C SI TTDR=0, ENTONCES TOMA EL VALOR GRABADO EN LA CINTA MAGN.
00096 IF(TSYS) 10,30,11
00097 11 WRITE(6,100)
00098 100 FORMAT("CICLOS INICIAL Y FINAL A PROCESAR (NO.MAX.INTRV.=5)")
00099 READ(1,*) CIF
00100 C SI SE DA VALOR -1 A CUALQUIERA DE LAS RESPUESTAS A ESTAS
00101 C PREGUNTAS, SE REINICIA EL CICLO DE PREGUNTAS
00102 IF(CIF+1.) 22,10,22
00103 22 WRITE(6,200)
00104 200 FORMAT("CICLOS A ELIMINAR (NO.MAX.CICLOS = 10)")
00105 READ(1,*) CAE
00106 IF(CAE+1.) 66,10,66
00107 66 WRITE(6,700)
00108 700 FORMAT("AR (H,M,S), DEC (G,M,S) (RET.=ANTER.):_")
00109 READ(1,*) ARH,ARM,ARS,DEG,DEM,DES
00110 IF(ARH+1.) 88,10,88
00111 88 WRITE(6,800)
00112 800 FORMAT("VLSR (KM/S) (RET.=ANTER.)= _")
00113 READ(1,*) VLSR
00114 WRITE(6,900)
00115 900 FORMAT("CANAL PARA EL QUE V=VLSR : _")
00116 READ(1,*) CVLSR
00117 C
00118 WRITE(6,9117)
00119 9117 FORMAT("DESEAS CORREGIR LOS DATOS RECIEN ENTRADOS?"
00120 "" (SI=1) _")
```

```
00121 READ(1,*) ICORR
00122 CALL EXEC(2,506B,6924,1)
00123 IF(ICORR) 10,9119,10
00124 C
00125 9119 L=1
00126 NGC=1
00127 C POR SI SOLO SE ANALIZA UN ESPECTRO : N = 1
00128 13 N = 1
00129 IF(-CIF(2*NGC)) 3,2
00130 C NUMERO DE PUNTOS EN ESTE GRUPO DE CICLOS
00131 3 N = CIF(2*NGC) - CIF(2*NGC-1) +1
00132 2 CL = CIF(2*NGC - 1)
00133 C SI NO QUEDAN MAS GRUPOS DE CICLOS POR ANALIZAR VE A 101
00134 IF(CL,EQ.0) GO TO 101
00135 C INICIO DE UN LARGO BUCLE EN QUE SE ANALIZA CADA CICLO
00136 IL=CL
00137 DO 4 I=1,N
00138 25 IF(CL-CAE(L)) 21,41,24
00139 21 CALL LEE62(IL)
00140 ITON = INTR/MPS
00141 C DETERMINACION DEL ESPECTRO PARCIAL
00142 IF(TTDR) 90,91,90
00143 90 TTR = TTDR
00144 91 DO 43 J=1,64
00145 IF(V2(J)-V1(J)) 111,112,111
00146 111 S1(J) = (V1(J)-V3(J))/(V2(J)-V1(J)) * TTR
00147 GO TO 43
00148 112 S1(J) = DUM
00149 43 CONTINUE
00150 DO 449 J=65,128
00151 IF(V4(J)-V3(J)) 113,114,113
00152 113 S2(J) = (V3(J)-V1(J)) / (V4(J)-V3(J)) * TTR
00153 GO TO 449
00154 114 S2(J)=DUM
00155 449 CONTINUE
00156 WRITE(6,110) NOMBR,IL,TTR
00157 110 FORMAT("FUENTE "6A2,2X"CICLO:"I4,2X"TEMP. DEL TDR="F5.1,2X"_)")
00158 GO TO 475
00159 24 L = L + 1
00160 GO TO 25
00161 C GENERACION DE LA CABECERA DEL REGISTRO
00162 475 CL=CL+1
00163 IL=IL+1
00164 IBF(3)=64
00165 IBF(4)=2
00166 IBF(16)=ICLO
00167 C PALABRA DE CONTROL PARA POSTERIOR LECTURA DEL FICHERO
00168 C O EN EL ANALISIS
00169 R(1)=1
00170 R(38)=1
00171 R(45)=ITON*TP100/100
00172 R(46)=R(45)
00173 AR=ARH+(ARM+ARS/60)/60
00174 R(26)=VM(AR/12)
00175 DEC=DEG+(DEM+DES/60)/60
00176 R(27)=VM(DEC/180)
00177 TU=IDAT(3)+(IDAT(4)+IDAT(5)/60)/60
00178 R(18)=VM(TU/12)
00179 ANY=IDAT(1)
00180 DIA=IDAT(2)
```

```

00181 CALL TSGOH(ANY,DIA,TSG,TSGH,TSGM,TSGS)
00182 TS=TSG+1.00274*IU-0.291
00183 H=15.*(TS-AR)/57.29578
00184 SENH=0.649*SIN(DEC/57.29578)+0.761*COS(DEC/57.29578)*COS(H)
00185 ELEV=ASIN(SENH)*57.29578
00186 R(22)=VM(ELEV/180)
00187 R(19)=VM(TS/12)
00188 SENA=COS(DEC/57.29578)*SENH/COS(ELEV/57.29578)
00189 X=ASIN(SENA)/3.1415927
00190 R(20)=VM(X)
00191 SENQ=0.761*SENH/COS(ELEV/57.29578)
00192 X=ASIN(SINQ)/3.1415927
00193 R(25)=VM(X)
00194 CALL MESDI(IDAT(1),IDAT(2),MES,IDIAM)
00195 R(37)=(IDAT(1)-1900)*10000.+MES*100+IDIAM
00196 R(49)=TTR
00197 R(50)=TTR
00198 R(51)=1
00199 R(52)=1
00200 R(73)=TSYS
00201 R(74)=TSYS
00202 R(29)=VLSR
00203 R(85)=BANDA
00204 R(105)=2272.661
00205 R(93)=BANDA/2272661.*299792.5
00206 R(89)=VLSR-R(93)*(32-CVLSR)
00207 DO 6911 J=1,6
00208 6911 NOMDB(J)=NOMBR(J)
00209 CALL WRITF(IDCC,IE,IBF,768)
00210 CALL ERFMP(15,IE)
00211 WRITE(6,343) IREG
00212 343 FORMAT("REGISTRO"14)
00213 IREG=IREG+1
00214 GO TO 4
00215 41 L = L + 1
00216 CL = CL + 1.
00217 IL = IL + 1
00218 4 CONTINUE
00219 C AQUI TERMINA EL BUCLE
00220 C
00221 IF(1-N) 9442,101
00222 C TODAVIA QUEDA ALGUN GRUPO DE CICLOS POR ANALIZAR?
00223 9442 NGC = NGC + 1
00224 GO TO 13
00225 C SI NO QUEDA NINGUN GRUPO DE CICLOS POR ANALIZAR:
00226 C PERDIDA DE TIEMPO PARA LEER LA LISTA EN PANTALLA
00227 101 DO 106 J=1,800
00228 106 Z=SIN(3.3)
00229 CALL EXEC(2,506B,6924,1)
00230 CALL EXEC(2,506B,7,1)
00231 GO TO 10
00232 C FINAL
00233 30 CALL CLOSE(IDCB,IE)
00234 CALL RWNDF(IDCC,IE)
00235 C LECTURA DE LAS CABECERAS DEL FICHERO DE ESPECTROS
00236 C SE PUEDE ESCOGER EL PRIMER REGISTRO. SE PARA AUTOMATICAMENTE.
00237 K = 1
00238 WRITE(6,2320)
00239 READ(1,*) K
00240 IF(K.NE.1) CALL POSNT(IDCC,IE,K,1)

```



```

00241 917 CALL READF(IDCC,IE,IBF,768,LEN)
00242 CALL ERFMP(19,IE)
00243 IF(LEN.EQ.-1) GO TO 919
00244 IF(R(1).NE.1.) GO TO 919
00245 WRITE(6,918) K,NOMDB
00246 918 FORMAT("REGISTRO "I3,5X"NOMBRE: "6A2)
00247 K = K+1
00248 GO TO 917
00249 919 CALL CLOSE(IDCC,IE)
00250 CALL ERFMP(99,IE)
00251 C
00252 C POSIBLE PURGA DEL FICHERO DE DATOS
00253 K=0
00254 WRITE(6,3011)
00255 3011 FORMAT("DESEAS PURGAR EL FICHERO DE DATOS ? (SI=1) : _")
00256 READ(1,*) K
00257 IF(K.NE.1) STOP
00258 WRITE(6,3022)
00259 3022 FORMAT("REALMENTE DESEAS PURGARLO ? (ESCRIBIR 'SI' O 'NO'): _")
00260 READ(1,3023) K
00261 3023 FORMAT(IA2)
00262 IF(K.NE.2HSI) STOP
00263 CALL PURGE(IDCDB,IE,NAME)
00264 END
00265 C*****
00266 FUNCTION VM(X)
00267 C EN LA REDUCCION A MEDIAS VUELTAS
00268 IF(X.GT.1.) X=X-2
00269 VM=X
00270 RETURN
00271 END
00272 C*****
00273 SUBROUTINE LEE62(IL)
00274 C LECTURA DEL FICHERO GENERADO MEDIANTE "TRAS2"
00275 COMMON IDCDB(291),IDAT1(275),IDAT2(275),IDAT3(275),IDAT4(275)
00276 CALL RWNDF(IDCDB,IE)
00277 CALL ERFMP(12,IE)
00278 K = (IL-1)*4+1
00279 CALL POSNT(IDCDB,IE,K,1)
00280 CALL ERFMP(10,IE)
00281 CALL READF(IDCDB,IE,IDAT1)
00282 CALL ERFMP(11,IE)
00283 WRITE(6,1) IDAT1(1),IDAT1(2)
00284 1 FORMAT("CICLO"17" FASES"14"_)
00285 CALL READF(IDCDB,IE,IDAT2)
00286 WRITE(6,2) IDAT2(2)
00287 2 FORMAT(I4"_)
00288 CALL READF(IDCDB,IE,IDAT3)
00289 WRITE(6,2) IDAT3(2)
00290 CALL READF(IDCDB,IE,IDAT4)
00291 WRITE(6,3) IDAT4(2)
00292 3 FORMAT(I4)
00293 RETURN
00294 END
00295 C*****
00296 SUBROUTINE TSG0H(ANY,DIA,TSG,TSGH,TSGM,TSGS)
00297 C CALCULO DEL TIEMPO SIDEREO EN GREENWICH A 0 HORAS DE TU
00298 A=ANY-1900
00299 TSGS=(9.28E-6*A+2.706580)*A-3.444640*IFIX(0.25*(A-1))-
00300 -(0.444640-5.09E-8*A)*DIA+47.5583+(1260-3*DIA)

```

```
00301 TSGM=IFIX(14-AMOD(A-1,4.)+TSGS/60+4*DIA)
00302 TSGH=IFIX(6.+TSGM/60)
00303 TSGH=AMOD(TSGH,24.)
00304 TSGM=AMOD(TSGM,60.)
00305 TSGS=AMOD(TSGS,60.)
00306 TSG=(TSGS/60+TSGM)/60+TSGH
00307 RETURN
00308 END
00309 C*****
00310 SUBROUTINE MESDI(IANY, IDIAA, MES, IDIAM)
00311 C DADOS EL AÑO Y EL DIA DEL AÑO CALCULA EL MES Y EL DIA DEL MES
00312 I(J,K)=(K-MOD(J,K))/K
00313 M=I(IANY,4)-I(IANY,100)+I(IANY,400)
00314 MES=IDIAA/29+1
00315 IF(MES.EQ.13) MES=12
00316 1 IDIAM=IDIAA-305*MES/10+32-((5-MES/3)/5)*(2-M)-M
00317 IF(MES.EQ.9.OR.MES.EQ.11) IDIAM=IDIAM-1
00318 IF(IDIAM.GT.0) RETURN
00319 MES=MES-1
00320 GO TO 1
00321 END
00322 C*****
00323 SUBROUTINE ERFMP(I,IERR)
00324 C ERROR EN EL PROCESO DE USO DE FICHEROS
00325 IF(IERR.GE.0) RETURN
00326 WRITE(6,1) I,IERR
00327 1 FORMAT(I6,5X"FMP ERROR CODE:"I5)
00328 PAUSE
00329 RETURN
00330 END
00331 C*****
00332 FUNCTION ASIN(X)
00333 C EL RESULTADO SE ENCUENTRA COMPENDIDO ENTRE -PI/2 Y +PI/2.
00334 IF(ABS(X).NE.1.) GO TO 1
00335 ASIN=1.570796*X
00336 RETURN
00337 1 Y=SQRT(1-X*X)
00338 ASIN=ATAN(X/Y)
00339 RETURN
00340 END
00341 $
```

```

00001 FTNA
00002 PROGRAM DATEF
00003 C 83.06.21
00004 C LECTURA DE LAS CINTAS DE DATOS EFFELBERG
00005 C
00006 DIMENSION IDCB(784),NAME(3),ISIZE(2)
00007 DIMENSION IDAT(4800),JBUF(1255),DAT1(193),DAT2(63)
00008 DIMENSION KBUF(410),IBUF(768),LBUF(256)
00009 EQUIVALENCE (IBUF(257),DAT1(1)),(IBUF(643),DAT2(1))
00010 EQUIVALENCE (IDAT(1601),JBUF(1)),(IDAT(3201),KBUF(1))
00011 EQUIVALENCE (IBUF(1),LBUF(1))
00012 WRITE(6,20)
00013 20 FORMAT("NOMBRE DEL FICHERO DISCO (MAX 6 CARACT.) : _")
00014 READ(1,21) NAME
00015 21 FORMAT(3A2)
00016 WRITE(6,22)
00017 22 FORMAT("CODIGO DE SEGURIDAD (2 CARACT.) : _")
00018 READ(1,*) ISECU
00019 24 WRITE(6,23)
00020 23 FORMAT("HAY QUE CREARLO ( 1 ) O ABRIRLO ( 0 ) ? : _")
00021 READ(1,*) ICR
00022 IF(ICR.NE.1.AND.ICR.NE.0) GO TO 24
00023 WRITE(6,26)
00024 26 FORMAT("NUMERO DEL CARTRIDGE (DEFECTO=15) : _")
00025 ICAR = 15
00026 READ(1,*) ICAR
00027 IF(ICR.NE.1) GO TO 311
00028 WRITE(6,31)
00029 31 FORMAT("NUMERO DE BLOQUES (-1=RESTO DEL CARTRIDGE): _")
00030 ISIZE(1) = -1
00031 ISIZE(2) = 768
00032 READ(1,*) ISIZE(1)
00033 IF(ICR.EQ.1) CALL CREAT(IDCB,IE,NAME,ISIZE,2,ISECU,ICAR)
00034 311 IF(ICR.EQ.0) CALL OPEN(IDCB,IE,NAME,0,ISECU)
00035 CALL ERFMP(1,IE)
00036 10 WRITE(6,1)
00037 1 FORMAT("NUMERO DE ESCANES A LEER ? (0=FIN,-1=REWIND,"
00038 , "9999= TODOS): _")
00039 READ(1,*) NSCAN
00040 IF(NSCAN.EQ.9999) NSCAN = 32767
00041 IF(NSCAN) 2,3,4
00042 2 REWIND 8
00043 GO TO 10
00044 3 REWIND 8
00045 CALL CLOSE(IDCB)
00046 STOP
00047 CCCCCCCCCCCCCC
00048 4 DO 11 I=1,NSCAN
00049 IF(I.EQ.I/10*10) WRITE(6,9) I
00050 9 FORMAT("ESCAN "I4)
00051 CALL EXEC(1,136,IDAT,1600)
00052 IF(IEOT(8).LE.0) GO TO 29
00053 CALL EXEC(1,136,IDAT(1601),1600)
00054 C **CABECERA**
00055 CALL HEADR(IDAT,LBUF)
00056 IF(LBUF(1).EQ.424) CALL EXEC(1,136,IDAT(3201),1600)
00057 C ** DATOS **
00058 CALL CODE(2509)
00059 READ(JBUF,613) DAT1
00060 613 FORMAT(193E13.7)

```

```

00061 CALL CODE(819)
00062 READ(KBUF,614) DAT2
00063 614 FORMAT(63E13.7)
00064 IF(LBUF(1).EQ.424) GOTO 700
00065 DO 103 J=129,256
00066 103 DAT1(J)=0
00067 700 CALL WRITF(1DCR,1E,1BUF,768)
00068 CALL ERFMP(2,1E)
00069 5 CONTINUE
00070 CCCCCCCCCCCCCC
00071 11 CONTINUE
00072 6 CALL MOVE(1)
00073 GO TO 10
00074 29 WRITE(6,20)
00075 30 FORMAT("ALCANZADO EL FIN DE LA CINTA")
00076 GO TO 3
00077 END
00078 SUBROUTINE MOVE (KEY)
00079 C
00080 C KEY=0 PARA ESCRIBIR
00081 C KEY#0 PARA LEER
00082 C
00083 IF (KEY) 20,3,20
00084 20 KEY=1
00085 3 IF (LOCAL(8)) 1,2
00086 1 WRITE (6,100)
00087 100 FORMAT ("UNIDAD EN LOCAL. _")
00088 PAUSE
00089 GO TO 3
00090 2 CALL EXEC (13,8,ISTAT)
00091 IF (IAND(ISTAT,4)-4*KEY) 5,11,4
00092 4 WRITE (6,103)
00093 103 FORMAT ("PONER ANILLA. _")
00094 GO TO 6
00095 5 WRITE (6,104)
00096 104 FORMAT ("QUITAR ANILLA. _")
00097 6 PAUSE
00098 GO TO 3
00099 11 NR=0
00100 WRITE (6,101)
00101 101 FORMAT ("MOVIMIENTO EN FICHEROS Y EN REGISTROS: _")
00102 READ (1,*) NF,NR
00103 CALL PTAPE (8,NF,NR)
00104 IF (IOR(IEOT(8),ISOT(8))) 14,8
00105 14 WRITE (6,102)
00106 102 FORMAT ("ALCANZADO UN 'EOT' O 'SOT'. _")
00107 PAUSE
00108 8 RETURN
00109 END
00110 SUBROUTINE ERFMP(I,IERR)
00111 IF(IERR) 1,2
00112 1 WRITE(6,100) I,IERR
00113 100 FORMAT(/15,5X"FMP ERROR CODE:"I4,5X"_")
00114 PAUSE
00115 2 RETURN
00116 END
00117 SUBROUTINE HEADR(IDAT,LBUF)
00118 DOUBLE PRECISION D(28)
00119 DIMENSION IDAT(1011),LBUF(256),MBUF(256)
00120 DIMENSION RI(8),NAM(20),R(96),A(3),NM(6)

```

```
00121 DIMENSION IN(5),JN(2),DP(16),KN(3)
00122 C
00123 EQUIVALENCE (MBUF(1),IN(1)),(MBUF(6),DUMMY),
00124 ,(MBUF(8),JN),(MBUF(10),NM),(MBUF(16),KN),
00125 ,(MBUF(19),R),(MBUF(211),DP)
00126 C
00127 C EL FORMATO DE SALIDA ES EL SIGUIENTE:
00128 C 5 ENTEROS, 1 REAL, 2 ENTEROS, 12 CARACTERES,
00129 C 3 ENTEROS, 112 REALES.
00130 CALL CODE(2021)
00131 READ(IDAT,1) RI,NAM,A,R,D
00132 1 FORMAT(6E13.7,2F10.0,20A2,3F10.0,96E13.7,28D22.16)
00133 DO 2 I=1,5
00134 2 IN(I)=RI(I)
00135 DUMMY = RI(6)
00136 JN(1) = RI(7)
00137 JN(2) = RI(8)
00138 DO 3 I=1,16
00139 3 DP(I)=D(I)
00140 DO 4 I=1,6
00141 4 NM(I) = NAM(I)
00142 DO 5 I=1,3
00143 5 KN(I) = A(I)
00144 DO 6 I=1,256
00145 6 LBUF(I) = MBUF(I)
00146 RETURN
00147 END
00148 $
```

```
00001 FTN
00002 PROGRAM AREA ( ),83.10.25
00003 C PERMITE EL CALCULO DEL AREA DE UNA RAYA.
00004 C SE LEEN FICHEROS DEL MISMO TIPO QUE LOS
00005 C ANALIZADOS O GENERADOS POR 'RRL'.
00006 C NO TIENE EN CUENTA LOS DUMMIES.
00007 REAL I1,I2
00008 DIMENSION IDCB(784),NAME(3),IBF(768),V(128),W(128)
00009 EQUIVALENCE (IBF(257),V),(IBF(513),W)
00010 WRITE(6,1)
00011 1 FORMAT("NOMBRE DEL FICHERO: _")
00012 READ(1,2) NAME
00013 2 FORMAT(3A2)
00014 CALL OPEN(IDCB,IE,NAME)
00015 IF(IE.LT.0) GO TO 88
00016 IRG=0
00017 3 I2=0
00018 C REGISTRO A LEER SI ES NEGATIVO,LO PERFORA (VIA 1)
00019 C POR DEFECTO,TOMA EL ANTERIOR (>0)
00020 WRITE(6,4)
00021 4 FORMAT("REGISTRO A LEER (0=FIN)((0=PERF): _")
00022 READ(1,*) IREG
00023 IS=IREG/IABS(IREG)
00024 IREG=IABS(IREG)
00025 IF(IREG.EQ.0) GO TO 99
00026 IF(IREG.EQ.IRG) GO TO 55
00027 IRG=IREG
00028 CALL POSNT(IDCB,IE,IREG,1)
00029 IF(IE.LT.0) GO TO 88
00030 CALL READF(IDCB,IE,IBF)
00031 IF(IE.LT.0) GO TO 88
00032 IF(IS.EQ.-1) GO TO 9
00033 55 WRITE(6,5)
00034 5 FORMAT("CANALES ENTRE LOS QUE QUIERES CALCULAR EL AREA"
00035 & "(0=LIST): ")
00036 READ(1,*) INI,IFI
00037 IF(INI.LE.0) GO TO 7
00038 CALL ITRAP(V,INI,IFI,I1)
00039 IF(IBF(4).EQ.2) CALL ITRAP(W,INI,IFI,I2)
00040 C PASO A KM/S
00041 I1=I1*6.6
00042 I2=I2*6.6
00043 WRITE(6,6) I1,I2
00044 6 FORMAT("VIA1: INTEG="E12.5/"VIA2: INTEG="E12.5//)
00045 GO TO 3
00046 C LISTADO EN PANTALLA
00047 7 N=IBF(3)
00048 WRITE(6,8) (V(I),I=1,N)
00049 8 FORMAT(10F7.3)
00050 IF(IBF(4).EQ.2) WRITE(6,8) (W(I),I=1,N)
00051 GO TO 3
00052 C PERFORACION EN CINTA DE PAPEL
00053 9 WRITE(4,10) V
00054 10 FORMAT(10F7.3)
00055 GO TO 3
00056 C MENSAJE DE ERROR
00057 88 WRITE(6,89) IE
00058 89 FORMAT("ERROR EN FICHERO: "I6)
00059 99 STOP
00060 END
```

```
00061 Cxxxxxxxxxxxxxxxxx
00062     SUBROUTINE ITRAP(V,INI,IFI,INTEG)
00063 C         CALCULO DE UNA INTEGRAL POR EL
00064 C         METODO DEL TRAPECIO.
00065     REAL INTEG
00066     DIMENSION V(128)
00067     INI1=INI+1
00068     IFI1=IFI-1
00069     INTEG=0
00070     DO 2 I=INI1,IFI1
00071 2     INTEG=INTEG+V(I)
00072     INTEG=INTEG+(V(INI)+V(IFI))/2
00073     RETURN
00074     END
```

```

00001 FTN
00002 PROGRAM LINEA ( ),83.10.27
00003 C
00004 C ESTE PROGRAMA PERMITE EL AJUSTE DE UNA GAUSIANA
00005 C MEDIANTE LA SUBROUTINA 'AJGAU' A UN CONJUNTO DE PUNTOS
00006 C MENOR O IGUAL QUE 100 Y MAYOR O IGUAL QUE 10 QUE SE
00007 C INTRODUCEN DESDE EL TECLADO.
00008 C
00009 C LOS DATOS INTRODUCIDOS SON LISTADOS CON FORMATO 'I6'
00010 C A FIN DE PODER SER CORREGIDOS.
00011 C
00012 DIMENSION X1(100),Y1(100),ITULO(32),AX(20),AY(20),
00013 ,X(100),Y(100),B(100)
00014 X1(1) = X1(2)
00015 2 WRITE(6,1)
00016 1 FORMAT(/"AJUSTE DE UNA GAUSSIANA A UN CONJUNTO DE PUNTOS"
00017 "" EQUIDISTANTES (<= 100)"/"NUMERO DE PUNTOS ? _")
00018 READ(1,*) N1
00019 IF(N1.LT.10.OR.N1.GT.100) GO TO 2
00020 WRITE(6,31)
00021 31 FORMAT("NUMERO DE CARACTERES DEL MEMBRETE ? _")
00022 READ(1,*) NT
00023 WRITE(6,3)
00024 3 FORMAT("MEMBRETE :")
00025 READ(1,4) ITULO
00026 4 FORMAT(32A2)
00027 43 WRITE(6,44)
00028 44 FORMAT("LECTURA DE PANTALLA (1) O DE CINTA DE PAPEL (5)? _")
00029 LEC=1
00030 READ(1,*) LEC
00031 IF(LEC.NE.1.AND.LEC.NE.5) GO TO 43
00032 IF(LEC.EQ.1) GO TO 59
00033 READ(5,48) Y1
00034 48 FORMAT(10F7.3)
00035 GO TO 61
00036 59 DO 5 I=1,N1
00037 WRITE(6,6) I
00038 6 FORMAT("Y("I3" ) = _")
00039 5 READ(1,*) Y1(I)
00040 61 CALL EXEC(2,506B,7,1)
00041 DO 36 I=1,N1
00042 36 Y(I) = ABS(Y1(I))
00043 CALL AMAX(Y,N1,YMAX,I)
00044 77 CALL EXEC(2,506B,6924,1)
00045 K = ALOGT(YMAX)
00046 FAC = 10.**(-K)
00047 DO 89 I=1,N1
00048 89 Y1(I) = Y1(I)*FAC
00049 WRITE(6,7) (Y1(I),I=1,N1)
00050 DO 90 I=1,N1
00051 90 Y1(I) = Y1(I)/FAC
00052 7 FORMAT(10I7)
00053 88 WRITE(6,55)
00054 55 FORMAT("QUE VALOR DESEAS CORREGIR ? (NINGUNO = 0) : I = _")
00055 READ(1,*) L
00056 IF(L.LE.0.OR.L.GT.100) GO TO 93
00057 WRITE(6,6) L
00058 READ(1,*) Y1(L)
00059 GO TO 88
00060 93 CALL AJGAU(N1,X1,Y1,NT,ITULO,N,X,Y,B,AX,AY)

```



```
00061      HIH=455.
00062      CALL GRAF(Y1,N1,10.,300.,1000.,HIH,N1/10,0)
00063      A1=AX(1)
00064      A2=AX(2)
00065      A3=AX(3)
00066      A4=AX(4)
00067      A5=AX(5)
00068      A6=AX(6)
00069      A7=AX(7)
00070      DO 73 I=1,N1
00071      G=I
00072      H=A1*EXP(-A3*(G-A2)**2)+A4+G*(A5+G*(A6+G*A7))
00073 73    B(I)=H
00074      SHIF=0
00075      CALL AMIN(Y1,N1,YMIN,J)
00076      IF(YMIN.GT.0.) GO TO 78
00077      CALL AMAX(Y1,N1,YMAX,J)
00078      FE=(YMAX-YMIN)/455.
00079      SHIF=-YMIN/FE
00080      CALL AMIN(B,N1,BMIN,J)
00081      CALL AMAX(B,N1,BMAX,J)
00082      HIH=(BMAX-BMIN)/(YMAX-YMIN)*455.
00083 78    CALL GRAF(B,N1,10.,300.+SHIF,1000.,HIH,N1/10,0)
00084      CALL PLOT(0,0,250)
00085      WRITE(6,8)
00086 8     FORMAT("ESTAS SEGURO QUE NO DESEAS ANALIZAR NUEVAMENTE ESTOS"
00087      " MISMOS VALORES ?"/"(RESPONDER 'SI' PARA PARAR EL PROGRAMA) _")
00088      READ(1,9) K
00089 9     FORMAT(A2)
00090      IF(K.NE.2HSI) GO TO 77
00091      END
00092      END$
```