

Informe Técnico CAY 1983-2.

PROGRAMAS INTERACTIVOS PARA EL ANALISIS  
DE OBSERVACIONES RADIO EN EL CONTINUO.

Pere Planesas

## I N D I C E

1. INTRODUCCION . . . . .	1
2. PROGRAMAS DE ANALISIS DE OBSERVACIONES EN CONTINUO . . . . .	3
.1 Programa ANA . . . . .	3
1. Creación de ficheros . . . . .	5
2. Generación del fichero INTER . . . . .	6
3. Generación del fichero FINAL . . . . .	8
4. Otros comandos . . . . .	11
5. Puesta en marcha del programa . . . . .	11
6. Errores . . . . .	12
7. Modificaciones en el programa . . . . .	13
.2 Programa VISU . . . . .	15
1. Comandos . . . . .	16
2. Puesta en marcha del programa . . . . .	18
3. Errores . . . . .	18
4. Modificaciones en el programa . . . . .	18
5. Descripción de las variables del COMMON . . . . .	18
3. PROGRAMAS COMPLEMENTARIOS . . . . .	21
.1 Programa SNR21 . . . . .	21
.2 Programa SNR13 . . . . .	23
.3 Programa TRASD . . . . .	25
.4 Programa TRASV . . . . .	27
ANEXOS :	
Anexo A . . . . .	28
Anexo B . . . . .	31
Anexo C (listados de los programas) . . . . .	33

## 1. INTRODUCCION

La adquisición de la unidad de disco magnético para el ordenador HP-2100S del Centro Astronómico de Yebes ha permitido el desarrollo de programas más versátiles y cómodos para el análisis de observaciones que cuando se disponía solamente de la unidad de cinta magnética y las unidades de lectura y escritura en cinta de papel. La reducida capacidad de memoria de dicho ordenador (32 Kpalabras) exigía dividir un proceso de análisis en varios pequeños programas que realizaban unas pocas funciones, utilizando cintas de papel para extraer y guardar los resultados intermedios. Además, cualquier ampliación significativa de uno de estos programas podía hacer que dejara de caber en la memoria del ordenador.

La unidad de disco, en cambio, permite tener los datos grabados en él a priori, con lo que su localización y lectura por parte del programa puede considerarse casi instantánea. Puede utilizarse también para escribir tanto resultados intermedios como los resultados finales, evitando el tiempo que exigía perforar las cintas, la acumulación de éstas y los problemas y averías que venía presentando últimamente la perforadora de cintas de papel.

Además la carga de un programa cualquiera se ha convertido en un proceso muy simple, pudiéndose pasar rápidamente de un programa a otro. Por otra parte, pueden utilizarse programas segmentados, de manera que sólo una parte del programa se encuentre realmente en el ordenador en un determinado instante, lo que evita los problemas de la reducida memoria del HP-2100S. El tener que segmentar los programas se ha aprovechado en el siguiente sentido: un programa de análisis exige una cierta versatilidad, es decir, poder realizar en cada instante la operación que se desee, en un orden arbitrario, y poder añadir con facilidad alguna instrucción que se considere necesaria para un cierto tipo de análisis.

## 2. PROGRAMAS DE ANALISIS DE OBSERVACIONES EN CONTINUO

Como se acaba de decir, el programa de análisis ha sido dividido en dos. El primero de ellos, llamado ANA, lleva a cabo el análisis y manipulación de los datos propiamente dicho, con visualizaciones sencillas (tipo "relieve montañoso") del conjunto de resultados. Se trata de un programa conversacional y por lo tanto muy versátil. Por otra parte, es sencillo introducir en él nuevas instrucciones en caso de que quien va a operar con él las necesite. Más abajo se dan consejos de cómo hacerlo.

El segundo programa, denominado VISU, permite la visualización de los datos en forma de "relieve montañoso" y de "curvas de nivel" (de igual intensidad, "isofotas"), de todos los resultados o una parte de ellos. Permite sustraer fuentes puntuales supuestamente observadas con un haz a definir. Se trata de un programa conversacional, pero mucho más simple que el anterior. Se ha separado de aquel para que pueda ser utilizado fácilmente para otros motivos. Por otra parte, como se verá más abajo, parar uno de los programas y poner en marcha el otro es tarea muy simple.

Aunque estos programas están pensados para analizar medidas en continuo, pueden analizarse o visualizarse también rayas espectrales. Por ejemplo, si se ha realizado un survey de rayas de recombinación en una cierta zona del cielo a intervalos regulares, pueden visualizarse las rayas en forma de isofotas, obteniéndose la llamada "representación  $l-v$ ".

### 2.1 Programa ANA

Permite el tratamiento y ajuste de barridos de hasta 316 puntos. En caso de que los barridos tengan un número mayor de puntos, es necesario promediarlos en grupos de 2 o 3 o el número necesario, a fin de que en total se tengan a lo sumo 316 puntos. Este valor se ha escogido de manera que bastara para analizar cualquiera de las observaciones de continuo realizadas y que diera lugar a unos "buffers" de datos de dimensión

múltiplo de 128 palabras, para que la comunicación con el disco sea lo más rápida posible.

El programa usa tres ficheros creados previamente y que vamos a denominar de ahora en adelante y de modo genérico: DATOS, INTER y FINAL; naturalmente sus nombres no tienen porque ser éstos: el nombre de un fichero puede tener hasta 6 caracteres, el primero de ellos no numérico, pues en caso contrario el ordenador lo tomaría como una unidad l6gica(LU).

El fichero DATOS es creado por los programas complementarios que el usuario debe desarrollar a fin de leer los datos de su cinta magnética o cinta de papel o que introduce desde la consola. Más adelante explicaremos con detalle su contenido. Sólo añadimos ahora que debe tener 50 registros de 1280 palabras (se trata de un fichero tipo 2: registros de longitud dada y acceso aleatorio). Cada registro contiene una cabecera y los datos correspondientes a dos vias (que corresponden a dos polarizaciones distintas, dos canales a frecuencia distinta, a dos bandas distintas o incluso a dos observaciones distintas de la misma zona que hay que promediar entre sí, con lo que serían 100, en ese caso particular las observaciones que se podrían analizar). La lectura de los registros se puede hacer en un orden arbitrario.

El fichero INTER es aquel en el que se colocan los resultados intermedios, por ejemplo los resultados de promediar distintos barridos correspondientes a una misma zona del cielo, habiendo antes eliminado las interferencias que haya en cada uno de los barridos individuales. Se trata de un fichero de acceso aleatorio tanto en escritura como en lectura, de manera que uno puede llenarlo en el orden que le sea más adecuado. En el caso de tener finalmente que visualizar un mapa, los barridos en el fichero INTER es conveniente que estén ordenados consecutivamente y de menor a mayor declinación. De esta manera podrá llevarse a cabo una visualización de los barridos promediados pero sin ajustar.

El fichero INTER sólo tiene capacidad para 25 registros de 1920 palabras. El que los registros sean más largos corresponde a que contienen separadamente el promedio de las vias 1 consideradas válidas, el promedio de las vias 2 consideradas válidas y el promedio de todas las vias, 1 y 2, consideradas válidas, así como una cabecera.

números de bloques (1 bloque = 128 palabras) y los números 1920 y 704 son los respectivos números de palabras de cada registro. El número total de registros vendrá dado por

$$\text{nº de registros} = \frac{\text{nº de bloques} \times 128}{\text{nº de palabras por registro}}$$

Son 25 y 60 respectivamente, como se había dicho antes.

Puede ser conveniente introducir un código de seguridad respecto a borrado, especialmente en el fichero FINAL, que hay que colocar en forma de 2 caracteres tras los dos puntos (:) que siguen al nombre del fichero. Por ejemplo, si el código de seguridad va a ser AA, se escribe

CR,FINAL:AA:12:2:330:704

En caso de no haber sido creados anteriormente, los ficheros INTER y FINAL son creados por el propio programa, tras dar un aviso de error. En este caso es inevitable el dar un código de seguridad al fichero FINAL; si se quiere prescindir de él, contestar 00 (cero, cero).

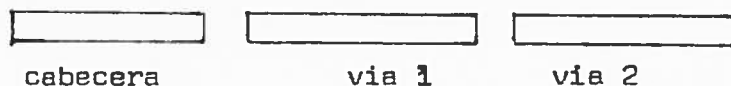
Se puede describir el funcionamiento interactivo de ANA en dos partes, la que va hasta la generación del fichero INTER y la que partiendo de los resultados intermedios que se encuentran en éste genera el fichero final.

### 2.1.2 Generación fichero INTER:

Los comandos más importantes que actúan sobre los datos son los siguientes:

LECFD lectura del fichero de datos (DATOS). Pide el número del registro a leer, hace la gráfica de ambas vías y pregunta cuales son válidas: 0, 1, 2, 12 (0 = ninguna, 12 = ambas).

Estos datos quedan contenidos en un vector de la forma:



SALTD permita corregir saltos bruscos de ganancia y/o interferencias ("spikes").

INTE integra el barrido ya manipulado en un trio de vectores intermedios que corresponden a: promedio de vías 1, promedio de vías 2 y promedio de vías 1 y 2. En la primera llamada estos vectores son nulos. Sólo son integradas (en los vectores correspondientes) las vías consideradas válidas.

Si se integran barridos de distinta longitud, debe empezarse por uno de los más largos, pues el primero va a definir el tamaño. No pueden promediarse barridos de 3 longitudes distintas (si se hace así el resultado estará falseado). Además, los barridos más cortos deben tener siempre el mismo corrimiento respecto de los largos, de forma que el salto que va a aparecer en los vectores intermedios caiga siempre en el mismo lugar.

**ANULI** permite anular los vectores intermedios en caso de querer recomenzar la integración.

**ESCFI** escribe el resultado de la integración en el fichero INTER. Pregunta el registro en el cual se desea escribir (machacando lo que hubiere en él). Recuérdese la recomendación de grabarlos de manera que queden de una forma ordenada (de menor a mayor declinación, para el caso de mapas).

Tras este comando, los vectores intermedios resultan automáticamente anulados, lo que permite iniciar tranquilamente otra integración.

Otros comandos que actúan sobre los vectores de datos son los siguientes:

**SUAVD** suavizado mediante un filtro Hanning. Código = 2.

**COMPD** compresión a la mitad; es decir, promedio de pares de puntos. El número de puntos se reduce a la mitad. Código = 3.

**RESTD** resta el valor mínimo a cada uno de los vectores de datos.

**FACTD** permite multiplicar las vías 1 y 2 por dos factores distintos. Por defecto toma factores unidad.

**LISTD** lista los valores numéricos de los vectores de datos considerados válidos: la vía 1, la vía 2 o ambas.

Los comandos recién citados actúan sólo sobre las vías consideradas válidas (la 1, la 2 o ambas).

**GRAFD** realiza la gráfica de los datos, tras las manipulaciones que hayan tenido lugar. Puede llamarse en cualquier momento.

**DUMPD** permite listar las cabeceras de los registros del fichero DATOS.

El código permite memorizar ciertos tipos de manipulaciones. Si sólo se han corregido interferencias, es CODIGO = 1; si se suaviza, entonces es CODIGO = 2 x CODIGO, y cada vez que se comprime es CODIGO = 3 x CODIGO. Más adelante veremos que cuando se ajusta es CODIGO = 5 x CODIGO. Dado que 2, 3 y 5 son primos, viendo el valor final de CODIGO es fácil determinar qué manipulaciones se han hecho (e incluso cuántas veces) sobre un determinado vector.

### 2.1.3 Generación del fichero FINAL

Se trabaja con 3 vectores intermedios que contienen la siguiente información:

R	S	T
$\frac{\sum \text{vias 1 escogidas}}{N_1}$	$\frac{\sum \text{vias 2 escogidas}}{N_2}$	$\frac{\sum \text{vias 1 esc.} + \sum \text{vias 2 esc.}}{N_1 + N_2}$

Estos vectores son los mismos que se utilizan para la integración de los datos (comando INTE antes citado) a fin de ahorrar memoria. Ello puede provocar algunos errores si no se procede con cautela: véase el comando ANULI más abajo.

Los comandos más importantes son:

- DUMPI lista las cabeceras del fichero INTER
- LECFI lee un registro a escoger del fichero intermedio y realiza la gráfica de la vía deseada (1 = promedio de vías 1; 2 = promedio de vías 2; 12 = promedio de vías 1 y 2; 3 = los tres promedios). Salvo en el último caso, se pide un factor de escala en puntos de pantalla por Kelvin: recuérdese en este punto que la pantalla tiene 780 puntos en sentido vertical. (El programa ya tiene en cuenta por sí mismo el que los datos vienen dados en decimili-Kelvin).
- SALTI corrige saltos que anarecen debido a haber promediado barridos de distinta longitud y también permite corregir "soikes", quizás no corregidos en alguno de los barridos originales, por lo que aparecen todavía en el promedio.



AJUSI permite el ajuste de cada vía por separado mediante un polinomio de tercer grado. En tal ajuste se pueden evitar intervalos (donde se encuentren emisiones por encima del fondo que se trata de sustraer) o puntos aislados que por alguna razón no han sido eliminados anteriormente; si no los hay, se contesta  $\emptyset$  (cero) a la pregunta.

Tras el ajuste es dibujada la línea de base en sobreimpresión.

Si el ajuste no es considerado válido, a la salida se recupera el vector inicial. Si el ajuste es válido, aparece la gráfica del vector residuo, que a su vez es almacenado en el vector de entrada (en la vía correspondiente, sea 1, 2 o 12). Código = 5.

Como en general la escala de la gráfica no es adecuada, por lo menos en la primera ocasión que se va a proceder a un ajuste, es conveniente hacer una llamada a GRAFI antes que a AJUSI.

ESCFA escribe el resultado del ajuste anterior, así como los parámetros del ajuste, en un registro a escoger del fichero FINAL (recordar las recomendaciones hechas anteriormente respecto al orden). Sólo va a grabar la vía que se desee. Si se desean grabar (en distintos registros del fichero FINAL) los ajustes de distintas vías de un mismo barrido hay que hacer repetidas llamadas a ESCFA y escribir cada vía en un registro distinto del fichero, que dispone de 60 registros.

Otros comandos que actúan sobre los tres vectores intermedios son:

SUAVI suavizado Hanning de los vectores intermedios deseados (1, 2, 12, o los 3). Código = 2.

COMPI compresión: promedia pares de valores de cada uno de los vectores intermedios, reduciendo así su longitud a la mitad (y su ruido al 71%). Código = 3.

RESTI resta el valor mínimo correspondiente a cada uno de los tres vectores intermedios.

LISTI lista los valores numéricos de los datos contenidos en los vectores intermedios, sean leídos del fichero INTER o sean resultado

de una integración (recuérdese que estos vectores tienen dos funciones).

GRAFI realiza la gráfica de la vía deseada (1, 2, 12) de los valores contenidos en los vectores intermedios (sean leídos del fichero INTER o resultado de una integración con INTE) con la escala deseada, o bien la gráfica de los tres vectores (3) simultáneamente, en este caso sin escala alguna. Cuando se va a proceder a un tanda de análisis es conveniente antes hacer una llamada a este programa a fin de fijar la escala o cambiarla.

AJUS2 permite el ajuste por separado de dos trozos complementarios de barrido a definir. El resto de comentarios es análogo a AJUS1. El valor de la sigma ( $\sigma$ ) del ajuste en este caso no es exacto, pero en general es una muy buena aproximación (tanto mejor cuanto mayor sea el número de puntos ajustados). Código = 5. Un valor de NSEP distinto de cero indica que se ha utilizado este comando.

ANULI anula los vectores intermedios, como ya se ha dicho antes. Pero es preciso añadir otra función de este comando: si se ha estado manipulando el fichero INTER y vamos a proceder a nuevas lecturas de datos y su integración, es imprescindible usar este comando a fin de borrar el contenido de los vectores intermedios. Naturalmente si lo que se desea hacer es añadir un barrido más al resultado de una integración anterior, que se encuentra en INTER, se lee el registro correspondiente mediante LECFI, se lee el barrido a añadir mediante LECFD y se integran mediante INTE, tras lo cual puede escribirse el resultado en el mismo registro del fichero INTER que antes ocupaba.

Otros comandos que se refieren al fichero INTER son:

DUMPI lista las cabeceras de los registros del fichero INTER.

MONTI permite la visualización de un conjunto de hasta 8 barridos consecutivos almacenados en el fichero INTER.

Se puede escoger el ángulo de inclinación de la visual (por defecto toma  $45^{\circ}$ ) así como el espaciamiento vertical (por defecto toma

un espaciamiento vertical tal que la gráfica quepa bien en la pantalla).

El uso de MONTI altera también el contenido de los vectores intermedios.

#### 2.1.4 Otros comandos

Por último vamos a describir dos comandos que actúan sobre el fichero FINAL, que contiene los datos ya totalmente procesados y listos para su visualización: no pueden ya ser manipulados desde este programa.

DUMPA lista las cabeceras de los registros contenidos en el fichero FINAL.

LECFA permite la lectura de la información contenida en cada uno de los registros del fichero FINAL ( $\sigma$ , parametros del ajuste,....). Realiza la gráfica de los valores y opcionalmente lista los números. No altera el contenido de los vectores intermedios pues utiliza otro vector auxiliar.

Y ya por fin:

FIN este comando detiene el programa, después de haber cerrado los ficheros usados.

#### 2.1.5 Puesta en marcha del programa

Dado que está cargado en absoluto en el cartridge 2, la primera vez que se pone en marcha en una sesión hay que dar la instrucción:

RU,RUANA

El programa RUANA, que se halla en el cartridge 12. (que debe estar montado; si no lo estuviera, habría problemas por cuanto en dicho cartridge deben encontrarse los ficheros DATOS, INTER y FINAL) tiene como función restaurar todos los segmentos que se van a utilizar, operación que realiza en unos 40 segundos.

En los demás casos, mientras no se pare el ordenador basta dar la instrucción:

RU,ANA

para que funcione dicho programa. Para mayor información véase el anexo A.

### 2.1.6 Errores

Si se da un comando desconocido, el programa responde

ERROR: COMANDO 'cccccc' NO VALIDO

y queda listo para la entrada de otro comando.

Si se dan entradas erróneas, en algunos casos se repite la pregunta. Sin embargo no todas las cuestiones tienen control, luego en el caso de dar parámetros erróneos el resultado no es predecible.

Hay una serie de errores de FMP (manipulación de ficheros), ante los que responde de la forma:

NNNNN FMP ERROR CODE: -ccc PAUSE 0000

donde -ccc da el tipo de error y donde NNNNN es un número que permite localizar el programa en que ha tenido lugar el error. A continuación damos una lista de los errores que más posiblemente pueden aparecer:

- 002 Nombre del fichero duplicado. Hay que abortar el programa (AB,ANA), y antes de reiniciarlo cambiar el nombre a tal fichero (RN,nombr,nombrp) o bien crear dentro de ANA un fichero con un nombre distinto.
- 006 Cartridge no montado. Abortar el programa, montar el cartridge en cuestión (MC,nn) y empezar de nuevo (RU,ANA).  
Fichero no encontrado. Abortar el programa, localizar el verdadero nombre del fichero y reiniciar el programa.  
No suficiente espacio en el cartridge para crear el fichero. Empaquetar el cartridge (PK,cc) y reiniciar el programa. Si sigue sin caber, cambiar de cartridge.
- 007 Código de seguridad erróneo. Si ello ocurre en ESCFI, hay que volver a empezar: no está previsto el programa para que el fichero intermedio tenga código de seguridad. Si ocurre en ESCFA, el programa no es interrumpido, sino que solicita dicho código tantas veces como sea necesario.
- 012 Intento de leer o escribir fuera de los límites del fichero. Continuar con \*GO,ANA. La lectura o escritura no habrá tenido lugar.

-Ø14 Directorio del cartridge lleno. Abortar el programa y, o bien purgar ficheros en el cartridge 12, o bien crear el(los) fichero(s) en otro cartridge desde fuera del programa, y luego reiniciar éste.

-Ø15 Nombre ilegal. Abortar el programa y reiniciarlo, dando al fichero un nombre legal.

Estos errores sólo tienen lugar en aquellos programas que actúan de una manera u otra sobre ficheros. A continuación damos una lista de dichos programas (incluidos en ANA) y los números NNNNN que les corresponden:

- ENTRA 1, 2, 3, -2, -3
- LECFD 4, 5
- DUMPD 14, 15
- ESCFI 6, 7
- LECFI 8, 9
- DUMPI 95, 100+I
- MONTI 10
- ESCFA 200, 201, 204
- LECFA 202, 203
- DUMPA 205, 206
- FIN 1001, 1002, 1003

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

\*GO,ANA

En muchos de los casos descritos anteriormente ello no es posible, siendo necesario abortar el programa, mediante:

\*AB,ANA

En aquellos casos en que NNNNN ~~Ø~~ el propio programa corrige el error; en el caso de ENTRA, no pudiendo abrir un fichero con el nombre dado, pues no lo encuentra, crea un fichero con tal nombre en el cartridge 12.

### 2.1.7 Modificaciones en el programa

En caso de desear modificar el programa hay que tomar las siguientes precauciones:

a) Si se añade un comando nuevo, su nombre debe colocarse en el DATA del

programa MANDO y deben modificarse la variable NVERF, aumentándolo en una unidad, y el DIMENSION IVERF, aumentándolo en tres unidades.

b) En caso de tener que modificar el COMMON, en el sentido de añadir nuevas variables, esto se hace o bien añadiendo una instrucción COMMON al final, o bien tras las variables FE o SIG, pero nunca debe modificarse la ristra de valores que van desde INAM hasta T(316), pues forman una unidad en ciertas partes del programa.

### 2.1.8 Descripción de las variables del COMMON

- BF contiene los dos vectores de datos.
- MANDO contiene la llamada al programa MANDO que permite dar los comandos.
- NP número de puntos del barrido. Es leído mediante LECFD. Se divide por 2 cuando se usa COMPD.
- IVIA toma los valores 1, 2 o 12 según se considere válida las vías 1, 2 o ambas.
- INAM, RNAM contienen una letra y un número real que identifican el barrido.
- NPM número de puntos máximo tras el promediado (con INTE) o bien el número de puntos de los vectores intermedios tras la lectura del fichero INTER.
- N1 número de vías 1 promediadas de distintos barridos.
- N2 número de vías 2 promediadas de distintos barridos.
- El número total de vías promediadas es  $N1 + N2$ .
- ICODE código. Si es múltiplo de 2, se ha usado SUAVD o SUAVI.  
Si es múltiplo de 3, se ha usado COMPD o COMPI.  
Si es múltiplo de 5, se ha usado AJUSI o AJUS2.
- R, S, T vectores intermedios. Se usan cuando se integra (INTE) o cuando se lee el fichero INTER para proceder a ajustar y generar el fichero FINAL.
- NREGI número del registro del fichero INTER que estamos analizando. Aparece en numerosas cabeceras.
- FE factor de escala. Viene expresado en puntos de pantalla por decimilikelvin ( $10^{-4}$  K). O sea es el factor que uno introduce dividido por 10000.
- NSEP punto de separación si se utiliza AJUS2. En caso de utilizar AJUSI es  $NSEP = \emptyset$ .

IPIFE puntos inicial y final de los intervalos a eliminar (hasta 5 pares).

IPAE puntos aislados a eliminar (hasta 10 valores).

Están dimensionados al doble de lo necesario aparentemente a fin de poder almacenar el doble de valores cuando se usa AJUS2. Los valores de estos parámetros quedan registrados en el fichero FINAL.

SIG error cuadrático medio del ajuste, en decimiliKelvin.

IDCB, IDCC, IDCD son los buffers auxiliares que utilizan, respectivamente, los ficheros DATOS, INTER y FINAL.

NAME nombre de los ficheros a abrir. En particular, conserva el nombre del fichero FINAL, a fin de poder reabrirlo, en caso necesario, con código de seguridad incluido en el programa ESCFA.

## 2.2 Programa VISU

Trabaja con conjuntos de registros consecutivos de un fichero tipo FINAL, aunque permite sustituir alguno de los barridos por otro situado en cualquier parte del mismo fichero, así como añadir otro barrido cualquiera en un extremo u otro del conjunto de barridos a visualizar.

Esta versión trabaja con 15 barridos de hasta 316 puntos, 16 barridos de hasta 300 puntos o cualquier combinación de tamaño equivalente (con hasta 4800 puntos en total) siempre y cuando no se exceda el valor de 316 puntos (que viene dado por el propio fichero) y no haya más de 20 barridos (de hasta 240 puntos).

Este programa permite también generar y sustraer fuentes puntuales observadas con un radiotelescopio de haz dado (en forma de gaussiana bidimensional).

El fichero que analiza debe estar formado por lo menos por 60 registros de exactamente 704 palabras, estructuradas tal como se muestra en el anexo B.

Si se va a sacar copia de los resultados mediante la HARDCOPY, el programa permite deformar previamente en la pantalla las imágenes de manera que aparezcan bien en la copia. El factor considerado es  $\phi.894$ . Este factor se determinó un cierto día en unas ciertas condiciones: es posible que llevando la HardCopy un tiempo distinto funcionando, habiendo cambiado el rollo de papel sensible y hallándose éste en un estado dis-

tinto de consumición, el factor antedicho sería distinto, pero lo consideramos suficientemente aproximado.

### 2.2.1 Comandos

LECFV lectura de conjuntos de barridos consecutivos del fichero. Tras tal lectura es posible quitar uno (o más, reiterativamente) de cualquier extremo, o añadir uno (o más, reiterativamente) a cualquier extremo, o cambiar alguno de los barridos por otro del mismo fichero, no necesariamente ordenado. Todas estas manipulaciones pueden hacerse en el mismo instante de la lectura de los datos, o bien en una llamada posterior a LECFV.

Esta versión no permite intercalar barridos.

Pueden usarse barridos de distinto tamaño. En tal caso, si en el extremo inferior hay precisamente uno de los más cortos, es conveniente empezar por uno de los largos y luego añadir a este extremo inferior los más cortos, con el corrimiento adecuado, pues el primero de los barridos leídos da la longitud que va a considerarse. Como datos hay que entrar además la ascensión recta y la declinación del púlsar o centro del mapa o punto de referencia; tal declinación es usada para contraer el mapa según la ley del coseno. Al introducir estos datos, es forzoso dar 6 números (si falta alguno, contesta: ?).

MONTV visualiza en forma de "relieve montañoso" los registros leídos. Hay que entrarle como parámetros, la inclinación y el factor de escala (en puntos por Kelvin), en caso que desee modificarse. En caso contrario se contesta con el valor cero.

En la primera ocasión en que es llamado tras una lectura, hace una llamada automática a GRAFV.

ISOFV visualiza en forma de "curvas de nivel" los registros leídos. Si no, se ha hecho ninguna llamada a GRAFV, la hace automáticamente. Pide como parámetros el valor de la primera isofota, supuesto el valor máximo normalizado a la unidad, y la distancia entre ellas. Si se le entra un tercer valor en esta misma pregunta, lo tomará como el número de curvas a representar. En caso contrario representará todas las que pueda. Ejemplo: si se contesta 0.2, 0.2 representará las curvas de nivel 0.2, 0.4, 0.6, 0.8, 1.0 (este valor cor-



responde al máximo); si se contesta 0.2,0.2,3 representará únicamente las curvas de nivel (isofotas) de valor 0.2, 0.4 y 0.6 . Representa los lugares donde se encuentran el núcleo y las fuentes puntuales sustraídas mediante una cruz: + .

La escala de la gráfica (ascensión recta-declinación) tiene en cuenta la corrección por el coseno de la declinación.

Aparecen indicados la ascensión recta cada minuto de tiempo y la declinación cada 10'.

ZOOMV permite visualizar una parte del mapa mediante una llamada a ISOFV. El resto se pierde: en caso de querer visualizar de nuevo todo el mapa hay que volver a leer los datos.

Mantiene el mismo tamaño horizontal en puntos de pantalla. Para cambiar dicho valor llamar a GRAFV.

Al intentar representar aquellos signos '+' que se salgan de la gráfica reducida pueden aparecer símbolos raros en la pantalla.

PUNTV Permite sustraer o añadir hasta 5 fuentes puntuales observadas supuestamente con un haz gaussiano a definir. Para sustraer, la amplitud de las fuentes debe ser positiva. Para añadir, la amplitud debe ser negativa.

GRAFV tras una lectura (LECFV), es llamado por el primero de los programas de visualización que se comande (MONTV o ISOFV).

También puede ser llamado en cualquier momento para modificar alguno de los parámetros que exige:

- anchura de la gráfica en puntos de pantalla (sin considerar aún la corrección por el coseno de la declinación). Si se responde 0 dibuja la mayor gráfica posible que quepa en la pantalla.

- distancia entre barridos (en minutos de arco) y entre puntos (en segundos de tiempo).

- coordenadas del punto (1,1) o sea el inferior izquierdo (ascensión recta máxima y declinación mínima). Debe ser coherente con el dato introducido en LECFV.

DUMPV lista las cabeceras del fichero FINAL.

FINV termina el programa, cerrando el fichero.

## 2.2.2 Puesta en marcha del programa

Es análoga a la descrita para el programa ANA (ver 2.1.5). La primera vez que se pone en marcha el programa en una sesión hay que dar la instrucción

RU,RUVISU

En los demás casos, mientras no se pare el ordenador basta dar la instrucción

RU,VISU

## 2.2.3 Errores

Véase el apartado de errores relativo al programa ANA (2.1.6).

Cabe tener en cuenta que los números código NNNNN que corresponden a los distintos programas que componen VISU son:

ENTRV 1  
LECFV 10 + I , 100, 101  
DUMPV 201,202

## 2.2.4 Modificaciones en el programa

En caso de desear modificar el programa hay que tomar las siguientes precauciones:

- a) en caso de desear añadir algún comando, su nombre debe añadirse en el DATA del programa MANDV. Asimismo hay que añadir 3 unidades a la variable NVERF y a la dimension de IVERF.
- b) no hay problema alguno en cuanto a añadir o intercalar variables en el COMMON, siempre y cuando no se trate de vectores de elevada dimensión, pues en tal caso podría haber problemas de memoria, lo que obligaría a reducir el valor NDMAX = 4800, número máximo de valores a analizar, así como la dimensión del vector F(4800).

## 2.2.5 Descripción de las variables del COMMON

MANDO contiene la llamada al programa MANDV que permite dar los comandos.  
IDCB es el buffer auxiliar del fichero de datos FINAL.  
NP número de puntos de cada barrido ( $\leq 316$ ).  
NB número de barridos a representar ( $\leq 20$ ).  
NI número inicial a representar de los barridos. En general es 1.  
F vector que contiene los datos a visualizar.

- IVIA 1, 2 o 12, según cual fuera el vector que se ajustó y registró en el fichero FINAL con el programa ANA.
- ICODE código, ya descrito en 2.1.8
- DEC declinación del púlsar o punto de referencia, en grados.
- RNAM contiene la parte numérica de los nombres que identifican los barridos. Aunque sólo se admiten 20 barridos está dimensionada a 21 por cuestiones del programa.
- RNAMI número que identifica el primer barrido.
- RNAMU número que identifica el último barrido.
- SIG es aproximadamente el valor medio de los errores cuadráticos medios ( $\sigma$ ) de los barridos utilizados para generar el mapa. Se trata de un valor meramente indicativo.
- DX anchura de la gráfica en puntos de pantalla, con corrección por la declinación y el factor FACHC.
- DY altura de la gráfica en puntos de pantalla. Se utiliza tanto en MONTV como en ISOFV.  
Los valores DX y DY son generados en GRAFV. A este programa hay que recurrir para cambiarlos.
- KONTR control. Al iniciar el programa vale -1. Tan pronto se han leído datos toma el valor 0. Al entrar en GRAFV se convierte en 1.  
Si se entra en MONTV y vale 0, sale valiendo 2 y entra en GRAFV, para volver luego automáticamente a MONTV, ya con el valor 1.  
Si se entra en ISOFV y vale 0, sale valiendo 3 y entra en GRAFV, para volver automáticamente a ISOFV, ya con el valor unidad.  
En ZOOMV se hace 1 y entra en ISOFV, con lo que no pasa por GRAFV.
- DIB distancia entre barridos en minutos de arco. Se lee en GRAFV.
- DIP distancia entre puntos en segundos de tiempo, corregida del efecto de la declinación. Se genera en GRAFV.
- AR ascensión recta en minutos de tiempo del punto (1,1).
- DE declinación en minutos de arco del punto (1,1).
- ARP ascensión recta del púlsar o punto de referencia en horas.
- COSD coseno de la declinación del púlsar corregido por FACHC.
- FACHC factor que deforma, si se desea, las imágenes reduciéndolas al 89.4% en sentido horizontal a fin de que en la Hard-Copy dé una imagen bien proporcionada.

El valor 0.894 parece uniforme a lo largo de la pantalla; sin embargo, como ya se ha dicho, no es del todo fiable, pues no sabemos si influyen en él, o hasta qué punto, lo gastado que esté el rollo de papel sensible y el tiempo que pueda llevar funcionando el aparato. Incluso puede ser que varíe ligeramente de una copia a la siguiente, en ciertas condiciones.

ARF      ascensión recta en minutos de tiempo de las hasta 5 fuentes puntuales a deducir.

DEF      declinación en minutos de arco de cada una de las hasta 5 fuentes puntuales a deducir.

Estos valores (ARF, DEF) se utilizan para dibujar una cruz (+) en el lugar donde se encuentran las fuentes deducidas. Como control se da el valor 1.E30 a las componentes de ARF que no corresponden a ninguna fuente puntual.

### 3. PROGRAMAS COMPLEMENTARIOS

La primera utilidad que se les ha dado a estos programas ha sido el análisis de observaciones en el continuo realizadas en el Observatorio Radioastronómico de Nançay (Francia) y en la Estación Espacial de Madrid, de INTA-NASA, sita en Robledo de Chavela.

Los programas complementarios que ha sido necesario desarrollar han sido, en primer lugar, la adaptación de dos programas de lectura de las cintas magnéticas correspondientes a ambos observatorios y que habían sido desarrolladas por Armando del Romero, así como también los programas de análisis en que se han basado los programas ANA y VISU.

Para utilizar los programas de análisis recién citados con datos ya algo analizados, se han desarrollado otros dos programas, uno de los cuales genera un fichero de datos a partir de cintas de papel con datos preanalizados (TRABD) y otro que genera un fichero FINAL a fin de visualizar un mapa (TRASV).

Los nombres SNR21 y SNR13 dados a los programas de lectura de cintas se refieren a: SNR es una abreviatura de "supernova remnant", el objeto buscado; el número expresa la longitud de onda de trabajo en centímetros (en Nançay son 21 cm y en Robledo son 13 cm).

Pasamos a continuación a describir brevemente cada uno de dichos programas.

#### 3.1 Programme SNR21

Crea un fichero de datos en disco capaz para 50 registros de 1280 palabras en el cartucho 12 a partir de cintas magnéticas grabadas en el Observatorio Radioastronómico de Nançay.

Tal como se realizaron las observaciones de continuo, se registraron los datos obtenidos en 8 vías distintas (2 de 15 MHz, 2 de 6 MHz y 4 de 2 MHz; en cada par, una es de polarización horizontal y otra vertical). Dado que sólo interesan las de 15 MHz, o las de 6 MHz, por tener mucho menos ruido, se puede escoger entre unas u otras.

A continuación se va dando el número del fichero que se desea leer en la cinta en cuestión; de esta forma es posible ordenar los barridos en una cierta secuencia en el fichero de datos. Una vez grabados todos los que interesan, se responde "0" a la pregunta "FICHERO A LEER".

Si se tienen datos relativos a una misma fuente en dos cintas distintas es posible colocarlos en un mismo fichero de datos. Para el cambio de cinta, responder también cero (0) a la pregunta anterior.

Acabado el trasvase de datos, es posible obtener un listado de las cabeceras de los registros del fichero.

Vamos a analizar con más detalle las cuestiones que el programa plantea:  
NOMBRE DEL FICHERO DATOS EN DISCO (A CREAR):

- Debe entrarse un nombre de a lo sumo 6 caracteres, el primero de ellos no numérico (si empezara por un número sería interpretado como un periférico). El nombre es conveniente escogerlo siguiendo alguna regla, pues si se acaba teniendo varios ficheros de datos debe poder distinguirse unos de otros con facilidad.

- Debe tenerse cuidado que no haya otro fichero con el mismo nombre en el mismo cartridge (el 12), pues en tal caso aparecerá en pantalla:

0 FMP ERROR CODE: -002 PAUSE 0000

En tales circunstancias debe uno abortar el programa (AB,SNR21) e iniciar de nuevo el programa dando otro nombre al fichero o bien purgando el existente (PU,nombre::12) antes de reiniciarlo.

ANCHO VIAS (15, 6 MHZ):

- Si se responde "6", toma las de 6 MHz; en caso contrario toma las de 15 MHz.

- Si la unidad de cinta magnética no está ON LINE, a continuación responderá

UNIDAD EN LOCAL

PAUSE 0000

Hay que pulsar el botón ON LINE de la unidad de cinta y luego responder

\*GO,SNR21

FICHERO A LEER:

- Se van dando sucesivamente los ficheros de cinta que interesa grabar, en un orden cualquiera. Es cómodo hacerlo de manera que queden bien ordenados en el fichero disco, pero si ello puede llevar mucho tiempo

por exigir, por ejemplo, largos movimientos de cinta o cambios de ella, puede prescindirse de ello: la consulta del disco es prácticamente instantánea independientemente del orden en que deba leer los registros.

- En la lectura, efectuada por la subrutina LEERF, los datos se convierten de formato entero de doble precisión (12 bits por palabra) a coma flotante (2 palabras de 16 bits), se genera una cabecera, que contiene una letra y un número real como nombre de la fuente, tres enteros que dan la fecha (el año es de la forma 82 y no 1982), el número de puntos y el número de fichero que ocupa en la cinta magnética original.

- El número de puntos es disminuido en una unidad (N-1) pues el primer valor es sistemáticamente erróneo y lo despreciamos (009J=2,N).

Al responder  $\emptyset$ , rebobina la cinta y pregunta:

DESEAS CAMBIAR DE CINTA (1) O : ACABAR ( $\emptyset$ ):

- Si se va a responder "1", es necesario cambiar la cinta antes de responder. La numeración de los ficheros es la que le corresponde a esta nueva cinta, sin corrimiento alguno.

- Si se desea acabar, se responde  $\emptyset$  (o cualquier otra cosa distinta de la unidad). En tal caso pregunta:

DESEAS LISTAR EL FICHERO DISCO ? (SI=1) :

- En caso afirmativo se puede conectar el teletipo y desconectar la pantalla antes de responder, con lo que se obtendrá el listado del contenido del fichero, en el que constan nombre de la fuente, fecha y número de puntos.

### 3.2 Programa SNR13

Su función es la misma que la de SNR21, pero es bastante distinto en su forma e incluso en su manejo, consecuencia del muy distinto formato en que están escritos los datos.

La cinta magnética ha sido grabada en la Estación Espacial de Madrid mediante el programa IGN/NAR. Se utiliza la configuración de 100 pps, pero se graba un valor cada segundo en grupos de 60 valores, lo que constituye la unidad llamada "sector" (de duración, por lo tanto, un minuto).

Dado que en Nançay se graba un valor cada 3 segundos y que un valor por segundo es, incluso en Robledo, redundante, así como que las unidades

utilizadas en Nançay son decimikelvin (0.0001 K), para obtener mayor uniformidad multiplicaremos los datos de la EEM por 10000, después de agruparlos de 3 en 3, lo que equivale a tomar un punto cada 3 segundos.

Dado que el número máximo de valores que admite el fichero de datos es 316, podemos admitir barridos de hasta 15 sectores (que equivalen a 300 puntos).

Las vías 1 y 2 equivalen a lo que en la EEM denominan canales 1 y 2.

Vamos a analizar las cuestiones que plantea el programa a su usuario:  
NOMBRE FICHERO DATOS EN DISCO (A CREAR):

Pueden hacerse comentarios análogos a los dichos para SNR21.

BARRIDO A LEER (A.R. + INCREM.DECL.):

- Dado que en la cinta magnética no hay información de nombres de fuentes o posiciones, sino sólo de tiempo, debe dársele el nombre desde el teclado. Aquí está pidiendo el número que va a seguir, sistemáticamente, a la letra P. Se pueden dar 4 cifras que correspondan a la ascensión recta y otras 2 que correspondan a la diferencia de declinación, en minutos de arco, entre este barrido y el pulsar, o punto de referencia.

A continuación escribe:

ULTIMO SECTOR LEIDO nnn

donde 'nnn' es un número que nos indica en qué lugar de la cinta magnética nos hallamos. La siguiente pregunta es:

MOVIMIENTO EN SECTORES (1 SECTOR = 1 MINUTO):

- Responder un número, que puede ser positivo o negativo.

LONGITUD DEL BARRIDO EN SECTORES (<=15):

- Responder un número positivo que debe ser menor o igual a 15.

- En caso de llegar al principio o al final de la cinta, escribirá:

ALCANZADO UN <EOT> O <SOT> PAUSE  $\emptyset\emptyset\emptyset$

- En caso contrario, leerá todos los datos, con un cierto formato.

Lee sector a sector y va rellenando el vector final con los datos agrupados de 3 en 3 y multiplicados por 10000. Al leer el primero de los sectores del grupo lee además el día del año y el tiempo inicial en horas, minutos y segundos, que es necesario transformar de ASCII a binario (CODE) a fin de colocarlo en el lugar de los registros en disco que ocupaba la fecha en el caso de las observaciones de Nançay (aquí interesa la hora,



pues al no ser una antena de tránsito, en un mismo día puede repetirse un mismo barrido en distintos instantes, y hay que poderlos distinguir).

Acabada la lectura escribe

DIA nnnn GMT(INICIAL) nn nn nn SCAN nn.n MIN

donde por \*n\* indicamos cifras. A continuación se genera la cabecera y el buffer de datos y se escribe en el fichero disco.

DESEAS LEER OTRO BARRIDO ? (SI= 1) :

- En caso afirmativo vuelve a la pregunta: BARRIDO A LEER (..)
- En caso de responder  $\emptyset$  (o cualquier cosa distinta de la unidad)

pregunta:

DESEAS LISTAR EL FICHERO ? (SI= 1)

- En caso afirmativo se puede conectar el teletipo y desconectar la pantalla antes de responder. Son listados el nombre de la fuente, la hora y el número de puntos.

### 3.3 Programa TRASD

Permite el traslado de datos de cinta de papel con el formato adecuado para ser analizados con el programa AUX (en Fortran II) a un fichero disco de datos adecuado para ser analizado con el programa ANA.

Son leídas por separado las vías que denominaremos 1 y 2. La cabecera es generada a partir de la cinta correspondiente a la vía 1. Ambas vías deben tener el mismo número de puntos.

Aunque los barridos no deben seguir ningún orden predeterminado, sí deben seguir el mismo orden los barridos de una y otra vía. (no hay ningún tipo de control sobre si coinciden los nombres).

Se puede dar un corrimiento a la vía 2 en aquel caso en que no se correspondan los primeros barridos, corrimiento que sólo puede ser positivo (o nulo, si no hay tal corrimiento). El número de barridos de la vía 2 debe ser igual o menor que el de la vía 1. Aquellos barridos para los que la vía 2 no sea leída contendrán el valor  $\emptyset.\emptyset$ .

Vamos a continuación a comentar las cuestiones que el programa plantea:  
NOMBRE DEL FICHERO DISCO PARA DATOS:

Mismos comentarios que en el caso de SNR21.

HAY QUE CREARLO (1) O ABRIR UNO YA EXISTENTE (Ø)? :

- En caso de crearlo, es creado en el cartridge 12.
- En caso de abrirlo, se pierde lo que contenía.
- Ante otra respuesta, repite la pregunta.

NUMERO DE PUNTOS DE CADA BARRIDO A LEER:

- Todos deben tener el mismo número de puntos. Debe coincidir con el número de puntos que hay perforados en la cinta de papel y debe ser menor o igual que 316.

- Esta pregunta vale para la vía 1 y para la vía 2 simultáneamente.

COLOCAR EN LA LECTORA DE CINTAS LA CINTA CORRESPONDIENTE A LA VIA QUE DENOMINAREMOS VIA1

NUMERO DE BARRIDOS A LEER ( $\leq 50$ ):

- Van a ser leídos a continuación uno detrás de otro, apareciendo en la pantalla el nombre de la fuente, la fecha y la vía.

- Si no todos los barridos se encuentran en la misma cinta, el programa se interrumpe. Hay que colocar la siguiente cinta y responder

\*GO,TRASD

VAS A LEER OTRA CINTA COMO VIA2 (SI=1) :

- En caso negativo, la vía 2 queda con valor Ø.
- En caso afirmativo, escribe

NUMERO DE PUNTOS = nnn

COLOCA LA NUEVA CINTA Y RESPONDE CUALQUIER COSA

Tras responder con cualquier carácter, preguntará

NUMERO DE BARRIDOS A LEER ( $\leq 50$ ):

valor que debe ser menor o igual al número de barridos correspondiente a la vía 1.

CORRIMIENTO ENTRE AMBOS CONJUNTOS DE BARRIDOS:

- La respuesta es Ø salvo en el caso en que para los primeros barridos del conjunto no se tengan los valores correspondientes a la Vía 2. Este número no puede ser negativo (en cuyo caso el programa toma Ø).

- A continuación son leídos el fichero y la cinta, con los datos de ésta se rellena parte del Buffer y se vuelve a escribir en el disco en la misma posición.

DESEAS UN LISTADO DE LAS CABECERAS ? (SI=1) :

- En caso afirmativo lista el nombre, fecha y número de puntos.

### 3.4 Programa TRASV

Permite el traslado del mapa de una fuente leído desde cintas de papel obtenidas a partir de antiguos programas a un fichero de disco apropiado para ser analizado con el programa VISU.

Permite también el promedio de varios mapas. En tal caso se supone que todos los mapas a promediar tienen el mismo número de barridos y un origen común. Se supone también que todos los barridos tienen la misma longitud.

El programa plantea a su usuario las siguientes cuestiones:

NOMBRE DEL FICHERO DISCO DONDE GRABAREMOS LOS DATOS:

- Hasta 6 caracteres, el primero de ellos no numérico. No se trata de un fichero tipo DATOS sino de un fichero FINAL, con resultados ajustados.  
HAY QUE CREARLO (1) O SOLO ABRIR UNO YA EXISTENTE (0) ? :

- mismos comentarios que en TRASD.

NUMERO DE MAPAS A LEER :

- sin límite.

NUMERO DE PUNTOS DE CADA BARRIDO:

- Menor o igual a 316. Para todos los mapas debe ser el mismo.  
- Además, todos los mapas deben empezar por el mismo barrido y ser leídos de declinación más baja a declinación más alta.

A partir de aquí empieza un bucle en el que se pregunta:

NUMERO DE BARRIDOS A LEER (MAPA  $n$ nn) :

- Debe ser menor o igual que 18.

PON EN MARCHA LA LECTORA Y COLOCA LA CINTA DE PAPEL:

- A continuación se responde cualquier carácter.  
- Son leídos los datos, generada la cabecera, invertido el orden de los puntos en los barridos y acumulados los valores.  
- Cuando se trata del primer mapa, se escriben ya las cabeceras en el vector que contiene los datos del buffer.

Acabado el bucle, se escriben los valores en el disco para el menor número de barridos comunes; es decir, si los mapas tenían 13, 10 y 12 barridos, sólo se escribirán 10 en el disco.

Anexo A

Sobre la carga de programas con el RTE II/III del HP-2100S.

a) Dado un programa sencillo de nombre, por ejemplo, SNR contenido, junto con todas las subrutinas que necesita, en el fichero de nombre &SNR, la compilación de programa y subrutinas, ejecutada de la forma

```
RU,FTM4,&SNR,,-
```

coloca el objeto relocatable en el fichero %SNR y el listado en pantalla.

La carga se efectúa en dos pasos:

```
MR,%SNR
```

```
RU,LOADR,99,,,,3
```

donde el número 3 indica que no deseamos ningún listado de los puntos de entrada de las distintas subrutinas del sistema necesarias. El ordenador responde:

```
/LOADR: SNR  READY
```

```
/LOADR: $END
```

Para correr el programa, tantas veces como se desee, mientras no se pare el ordenador, basta dar la instrucción

```
RU,SNR
```

b) Tras la carga descrita, si el programa funciona adecuadamente, puede ser guardado en absoluto en el cartridge 2 mediante la instrucción

```
SP,SNR
```

Una vez guardado en absoluto ya no se pierde al parar el ordenador. Cada vez que vaya a utilizarse dicho programa bastará dar la instrucción

```
RU,SNR
```

sin más preocupación. Para saber qué programas (o segmentos) se encuentran grabados en absoluto en el cartridge 2, basta hacer

```
DL,2
```

que lista el directorio de dicho cartridge, y ver cuáles son de tipo 6.

c) Para la carga de un programa en segmentos y que haga uso de una librería (es el caso más general; tal es el de AM y VISU) el proceso es más largo. En particular, es necesario realizar ciertas operaciones fuera del File Manager (FMGR); en todo lo anterior se suponía que nos hallábamos en él, lo que se conoce pues para pedir comandos escribe en pantalla dos puntos (:).

Sea PRO nuestro programa segmentado y sea &PRO el fichero fuente que contiene dicho programa y todos sus segmentos, aunque no así las subrutinas que, como es probable que cada una de ellas sea utilizada por más de uno de los segmentos, es conveniente que se encuentren en una librería, o sea un fichero fuente del tipo &GRAF (fichero que contiene sólo subrutinas, la mayoría de las cuales sirven para efectuar representaciones gráficas y que se encuentra listado en el apartado correspondiente de este trabajo).

Tanto uno como otro ficheros fuente se compilan tal como se ha descrito antes, dando lugar a ficheros relocatables de nombres, respectivamente %PRO y %GRAF.

Hallándonos dentro del FMGR todavía, puede ser necesario ampliar la memoria del área auxiliar LG necesaria para la carga de un programa. Esta será la primera instrucción:

```
LG,10  
MR,%PRO  
OF,FMGR  
*RU,LOADR,99,,,1,3
```

donde el número 1 indica que se trata de un programa segmentado. Tras un cierto lapso de tiempo el ordenador responde

```
/LOADR: UNDEFINED EXTS  
/LOADR: YYYY
```

siendo YYYY el nombre de la primera subrutina o Function que no encuentra. Este es el momento de proceder a la carga de la librería y reanudar la del programa:

```
*RU,FMGR  
MR,%GRAF  
OF,FMGR  
*GO,LOADR,2,1,1
```

tras un cierto tiempo, que pueden ser varios minutos, la carga finalizará con el mensaje

```
/LOADR: PRO   READY  
/LOADR: $END
```

tras lo cual basta dar

```
*RU,PRO
```

para que el programa PRO funcione.

Observación: Las instrucciones que no empiezan por asterisco (\*) se dan dentro del File Manager. Tras dar una de ellas hay que esperar que el FMGR responda con dos puntos (:), indicando que ya ha terminado la instrucción anterior. Téngase en cuenta que la compilación de un conjunto de programas o subrutinas grande puede llevar un minuto y la instrucción MR varios segundos.

Tras la instrucción OF;FMGR no hay que esperar respuesta. Se da acto seguido la siguiente instrucción.

La carga de un programa como ANA puede llevar del orden de 8 minutos en total.

d) Cuando un programa segmentado ya cargado se desea conservar como programa absoluto es necesario un fichero tipo 6 para cada segmento, todos en el cartridge 2. Aún cuando un programa en absoluto SNR corre con sólo dar la instrucción

\*RU,SNR

no ocurre así cuando dicho programa tiene varios segmentos, pues éstos deben ser restaurados, o sea hechos accesibles al sistema. Para ello es necesario aplicar la instrucción RP (restaurar programa) a cada uno de los segmentos y a continuación la instrucción RU al programa principal, que no es necesario restaurar. Este proceso hay que llevarlo a cabo dentro del FMGR.

RP,SEG1

RP,SEG2

⋮

RP,SEGN

RU,PRO

Todo este conjunto de comandos del sistema RTE puede estar contenido en un fichero de nombre, por ejemplo, RUPRO. En tal caso, la instrucción

RU,RUPRO

ejecutará todos los comandos anteriores uno tras otro, restaurando todos los segmentos y poniendo en marcha el programa. Esto es lo que hacen los programas RUANA y RUVISU antes citados.

La siguiente vez que se desee usar el programa bastará dar la instrucción

RU,PRO

mientras no sea parado el ordenador.

Anexo B

Sobre la distribución del contenido de los ficheros.

a) Fichero DATOS:

8 palabras	cabecera
316 reales	vía 1
316 reales	vía 2

Descripción de la cabecera:

1 letra } 1 real }	nombre de la fuente
3 enteros	fecha (d,m,a) u hora (h,m,s)
1 entero	número de puntos del barrido
1 entero	número del fichero en la cinta de Nançay, o número 0 para datos de Robledo de Chavela, o número 0 para datos de cintas de papel.

b) Fichero INTERmedio:

7 palabras	cabecera
316 reales	promedio de vías 1
316 reales	promedio de vías 2
316 reales	promedio de vías 1 y de vías 2
17 palabras	sólo tienen interés las 3 primeras; las demás no tienen tarea adjudicada por el momento.

Descripción de la cabecera:

1 letra } 1 real }	nombre de la fuente
1 entero	número de puntos
1 entero	número de vías 1 promediadas
1 entero	número de vías 2 promediadas
1 entero	código

De las 17 palabras finales, el primer entero indica el número del registro y a continuación un real indica el factor de escala (FE) usado.

c) Fichero FINAL:

50 palabras	cabecera
316 reales	valores ajustados de la via (1, 2, 12) considerada.
22 palabras	no adjudicadas.

Descripción de la cabecera:

1 letra	}	nombre de la fuente
1 real		
1 entero		número de puntos
1 entero		via considerada
1 entero		número de vias promediadas
1 entero		código
1 real		error cuadrático medio del ajuste ( $\sigma$ ).
1 entero		NSEP (punto de separación entre dos ajustes).
20 enteros		intervalos evitados en el(los) ajuste(s)
20 enteros		puntos aislados evitados en el(los) ajuste(s).



Anexo C

Listados de los programas citados.

Entre ellos se incluyen:

a) los que permiten el traslado de datos de cintas magnéticas o de papel a ficheros disco, como son:

&SNR21

&SNR13

&TRASD

&TRASV

b) los programas de análisis propiamente dichos:

&ANA

&VISU

así como la librería que utilizan uno y otro:

&GRF

y los programas que permiten ponerlos en marcha:

RUANA

RUVISU

Hay que hacer notar en este punto que las subrutinas que constituyen la librería &GRF no han sido realizadas ex-profeso para este trabajo, sino que forman parte de la librería que se ha venido desarrollando a lo largo de varios años de trabajos diversos por distintas personas. Se han listado aquí por dar completitud al trabajo y porque algunas de ellas han sido retocadas ligeramente o son específicas del RTE II/III.

```

1 FTN4,L
2 PROGRAM SNR21
3 C 82.04.18
4 DIMENSION IDCB(1296),IBF(1280),ISIZE(2),NAME(3),RDAT(1),BF(1)
5 COMMON MDAT(6144)
6 EQUIVALENCE (MDAT(513),RDAT(1)),(IBF(9),BF(1)),(IBF(2),RNAM)
7 C
8 WRITE(6,8001)
9 8001 FORMAT(10X"COLOCAR LA CINTA DE DATOS DE NANCAY"/)
10 WRITE(6,100)
11 100 FORMAT(" NOMBRE FICHERO DATOS EN DISCO (A CREAR) : _")
12 READ(1,101) NAME
13 101 FORMAT(3A2)
14 C
15 NREG = 0
16 ISIZE(1) = 500
17 ISIZE(2) = 1280
18 CALL CREAT(IDCB,IE,NAME,ISIZE,2,0,12,1280)
19 CALL ERFMP(0,IE)
20 M = 1
21 WRITE(6,200)
22 200 FORMAT(" ANCHO VIAS (15, 6 MHZ): _")
23 READ(1,*) IA
24 IF(IA.EQ.6) M=2
25 3 IF(LOCAL(8)) 1,2
26 1 WRITE(6,111)
27 111 FORMAT("UNIDAD EN LOCAL"/)
28 PAUSE
29 GO TO 3
30 2 WRITE(6,122)
31 122 FORMAT(" FICHERO A LEER: _")
32 READ(1,*) NFICH
33 IF(NFICH.LE.0) GO TO 1000
34 CALL LEERF(NFICH)
35 CALL FMT2(MDAT(3),MDAT(4),RNAM)
36 IBF(1) = MDAT(2)
37 IBF(4) = MDAT(29)
38 IBF(5) = MDAT(30)
39 IBF(6) = MDAT(31)-1900
40 N = MDAT(35)
41 IBF(7) = N-1
42 IBF(8) = NFICH
43 C
44 DO 9 I=M,4,2
45 K = (I-1)*N
46 DO 9 J=2,N
47 9 BF(J+316*((I-1)/2)-1) = RDAT(K+J)
48 NREG = NREG+1
49 CALL WRITF(IDCB,IE,IBF)
50 CALL ERFMP(2,IE)
51 GO TO 2
52 1000 REWIND 8
53 WRITE(6,356)
54 356 FORMAT("DESEAS CAMBIAR DE CINTA (1) O ACABAR (0) ? : _")
55 READ(1,*) ICA
56 IF(ICA.EQ.1) GO TO 3
57 WRITE(6,333)
58 333 FORMAT(" DESEAS LISTAR EL FICHERO DISCO? (SI=1)"/
59 ""(EN CASO AFIRMATIVO, CONECTAR EL TELETIPO Y DESCONECTAR PANTALLA
60 "" ANTES DE RESPONDER)")

```

```

61 READ(1,*) IJK
62 IF(IJK.NE.1) GO TO 1001
63 CALL RWNDF(IDC B,IE)
64 CALL ERFMP(100,IE)
65 WRITE(6,906) IA
66 906 FORMAT(/"ANCHO VIAS : "I3" MHZ"/)
67 DO 444 I=1,NREG
68 IF(((I-1)/10)*10.EQ.(I-1)) WRITE(6,858)
69 CALL READF(IDC B,IE,IBF)
70 CALL ERFMP(I+3,IE)
71 444 WRITE(6,445) I,IBF(1),RNAM,(IBF(J),J=4,7)
72 445 FORMAT(" REGISTRO: "I4,5X,A2,I7,5X" FECHA: "2I3,I5,5X
73 "NPUNT ="I4)
74 858 FORMAT(/)
75 1001 CALL CLOSE(IDC B,IE)
76 CALL ERFMP(999,IE)
77 END
78 SUBROUTINE LEERF (NF)
79 DIMENSION IDAT(1),RDAT(1)
80 COMMON MDAT(6144)
81 EQUIVALENCE (MDAT(513),IDAT(1),RDAT(1))
82 C
83 C ESTA SUBROUTINE HACE LA LECTURA DE LOS DATOS DE UN FICHERO
84 C DE LAS CINTAS MAGNETICAS DE <NANCEY>.
85 C
86 C ES NECESARIO QUE LA PRIMERA VEZ QUE SE UTILICE ESTA SUBROUTINE
87 C LA CINTA MAGNETICA ESTE EN <LOAD>, PARA QUE SE EFECTUE BIEN
88 C LA CUENTA DE FICHEROS.
89 C
90 IF (ISOT(8)) 1,2
91 1 K=1
92 2 MOV=NF-K
93 K=NF
94 IF (MOV-1) 3,4
95 3 MOV=MOV-1
96 4 CALL PTAPE (8,MOV,0)
97 CALL EXEC(1,200,MDAT,2048)
98 N=(MDAT(35)*16+513)/2048
99 IF (N-1) 6,5
100 5 DO 7 I=2,N
101 7 CALL EXEC(1,200,MDAT(2048*(I-1)+1),2048)
102 6 N=MDAT(35)*16
103 DO 9 I=1,N,2
104 J=I/2+1
105 9 CALL FMT2 (IDAT(I),IDAT(I+1),RDAT(J))
106 RETURN
107 END
108 SUBROUTINE FMT2 (IDAT1,IDAT2,RDATO)
109 C
110 C CONVERSION DE FORMATO ENTERO DOBLE PRECISION (12 BITS PALABRA)
111 C A COMA FLOTANTE (2 PALABRAS DE 16 BITS).
112 C
113 IF (2047-IDAT1) 1,2
114 1 RDATO=FLOAT(IDAT2)+FLOAT(IDAT1-4096)*4096.
115 GO TO 3
116 2 RDATO=FLOAT(IDAT2)+FLOAT(IDAT1)*4096.
117 3 RETURN
118 END

```

```
1 FTN,L
2 PROGRAM SNR13
3 C 83.04.18
4 C
5 C "LEE" DATOS DE (ROBLEDO), IGN/NAR, (CONFIGURACION 100 PPS,SECTORES
6 C DE 60 PUNTOS), EN KELVIN
7 C AGRUPA LOS PUNTOS DE 3 EN 3 (EQUIVALENTE A TOMAR 1 PTO.CADA 3 SEG.),
8 C ADMITE BARRIDOS DE HASTA 15 SECTORES(=15 MIN.=300 PTOS.)
9 C
10 C (MULTIPLICA LOS DATOS POR 10000.)
11 C
12 DIMENSION IREG(538),TEMP(256),IGMT(3)
13 DIMENSION IDCB(1296),IBF(1280),ISIZE(2),NAME(3),BF(1)
14 DIMENSION T1(316),T2(316),TM1(316),TM2(316),J0(3)
15 EQUIVALENCE (TEMP(1),IREG(10)),(IREG(3),IGMT(1)),(IBF(2),RNAM)
16 EQUIVALENCE (IBF(9),BF(1))
17 C
18 WRITE(6,100)
19 100 FORMAT(" NOMBRE FICHERO DATOS EN DISCO (A CREAR): _")
20 READ(1,101) NAME
21 101 FORMAT(3A2)
22 ISIZE(1) = 500
23 ISIZE(2) = 1280
24 CALL CREAT(IDCB,IE,NAME,ISIZE,2,0,12,1280)
25 CALL ERFMP(1,IE)
26 WRITE(6,10)
27 10 FORMAT(10X"COLOCAR BANDA MAGNETICA DE DATOS A LEER (ROBLEDO)")
28 REWIND 8
29 NREC=0
30 NREG = 0
31 11 CONTINUE
32 WRITE(6,12)
33 12 FORMAT(/"BARRIDO A LEER(A.R.+INCREMTO DECL.): _")
34 READ(1,*)RNAM
35 L=1
36 WRITE(6,14)NREC
37 14 FORMAT(/"ULTIMO SECTOR LEIDO : "I4)
38 WRITE(6,15)
39 15 FORMAT(/"MOVMIENTO EN SECTORES (1 SECTOR=1 MINUTO): _")
40 READ(1,*)NR
41 NR=NR*30
42 19 WRITE(6,20)
43 20 FORMAT(/"LONGITUD DEL BARR.,EN SECTORES(<=15): _")
44 READ(1,*)NS
45 IF(NS-15)25,25,19
46 25 NREC=NREC+(NR/30)+NS
47 30 CALL PTAPE(8,0,NR)
48 32 IF(IEOT(8))33,34
49 34 IF(ISOT(8))33,35
50 33 WRITE(6,36)
51 36 FORMAT("ALCANZADO UN (EOT) O (SOT) _")
52 PAUSE
53 35 CONTINUE
54 C
55 READ(8,40)IREG(1),IREG(2),IGMT,TSEC1,SIG1,
56 ,(TEMP(LL),LL=1,100),(TEMP(M),M=101,200),(TEMP(N),N=201,256),
57 ,IREG(522),DIO1,BET1,IREG(527),IREG(528),IREG(529),TSEC2,
58 ,SIG2,DIO2,BET2,IREG(538)
59 40 FORMAT(I4,I3,3A2,2F10.3,10(/10F10.3),10(/10F10.3),
60 ,8(/7F10.3),/I4,2F10.3,3I4,4F10.3,I4)
```

```
61 C CAMBIO DE FORMATO DE IGMT: PASO DE ASCII A BINARIO.
62 CALL CODE(6)
63 READ(IGMT,777) IGMT
64 777 FORMAT(3I2)
65 IF(L-1)60,50,60
66 50 I0=IREG(2)
67 DO 55 J=1,3
68 55 J0(J)=IGMT(J)
69 C
70 C AGRUPAMOS LOS DATOS DE CADA CANAL (1 Y 2) EN UN VECTOR
71 C
72 60 DO 70 J=1,60
73 T1(J)=TEMP(2*J-1)
74 T2(J)=TEMP(2*J)
75 70 CONTINUE
76 C
77 C AGRUPAMOS LOS PUNTOS DE CADA VECTOR DE 3 EN 3, PARA TENER UNA
78 C TEMPERATURA CADA 3 SEGUNDOS
79 C
80 DO 90 J=1,60,3
81 K=((J+2)/3)+L-1
82 TM1(K)=(T1(J)+T1(J+1)+T1(J+2))/3.*10000.
83 TM2(K)=(T2(J)+T2(J+1)+T2(J+2))/3.*10000.
84 90 CONTINUE
85 NSEC=K/20
86 C
87 IF(NSEC-NS)120,140,140
88 120 L=K+1
89 NR=0
90 GO TO 30
91 C
92 140 SCAN = K/20.
93 WRITE(6,160) I0,J0,SCAN
94 160 FORMAT("DIA"14" GMT(INICIAL)"3I2" SCAN "F4.1" MIN."/)
95 IBF(1) = 2HP
96 IBF(4) = J0(1)
97 IBF(5) = J0(2)
98 IBF(6) = J0(3)
99 IBF(7) = K
100 IBF(8) = 0
101 DO 9 I=1,K
102 BF(I) = TM1(I)
103 9 BF(I+316) = TM2(I)
104 CALL WRITF(IDCIB,IE,IBF)
105 CALL ERFMP(2,IE)
106 NREG = NREG + 1
107 WRITE(6,200)
108 200 FORMAT("DESEAS LEER OTRO BARRIDO? (SI=1): _")
109 READ(1,*) I
110 IF(I.EQ.1) GO TO 11
111 REWIND 8
112 WRITE(6,333)
113 333 FORMAT("DESEAS LISTAR EL FICHERO? (SI=1)"/
114 ""(EN CASO AFIRMATIVO CONECTAR EL TELETIPO Y DESCONECTAR PANTALLA"
115 " " ANTES DE RESPONDER)")
116 READ(1,*) IJK
117 IF(IJK.NE.1) GO TO 1001
118 CALL RWNDF(IDCIB,IE)
119 CALL ERFMP(3,IE)
120 DO 444 I=1,NREG
```

```
121 IF(((I-1)/10)*10.EQ.(I-1)) WRITE(6,809)
122 CALL READF(IDCIB,IE,IBF)
123 CALL ERFMP(3+I,IE)
124 444 WRITE(6,445) I,IBF(1),RNAM,(IBF(J),J=4,7)
125 445 FORMAT(" REGISTRO : "I4,5X,A2,I7,5X"HORA: "3I3,5X
126 ""NPUNT ="I4)
127 809 FORMAT(/)
128 1001 CALL CLOSE(IDCIB,IE)
129 CALL ERFMP(999,IE)
130 END
```

```

1 FTN,L
2 PROGRAM TRASD
3 C 83.04.18
4 C
5 C TRASLADA DATOS DE CINTA PERFORADA EN EL FORMATO ADECUADO PARA
6 C SER USADAS EN "AUX" A UN FICHERO DISCO ADECUADO PARA SER USADO
7 C CON "ANA".
8 C
9 DIMENSION IDCB(1296),IBF(1280),ISIZE(2),NAME(3),BF(1),A(636)
10 EQUIVALENCE (IBF(2),RNM),(IBF(9),BF(1))
11 DO 1 I=1,1280
12 1 IBF(I) = 0
13 WRITE(6,100)
14 100 FORMAT("NOMBRE DEL FICHERO DISCO PARA DATOS : _")
15 READ(1,101) NAME
16 101 FORMAT(3A2)
17 3 WRITE(6,102)
18 102 FORMAT("HAY QUE CREARLO (1) O ABRIR UNO YA EXISTENTE (0)"
19 " ? : _")
20 READ(1,*) ICR
21 IF(ICR.NE.1.AND.ICR.NE.0) GO TO 3
22 IF(ICR.EQ.0) CALL OPEN(IDCB,IE,NAME)
23 ISIZE(1) = 500
24 ISIZE(2) = 1280
25 IF(ICR.EQ.1) CALL CREAT(IDCB,IE,NAME,ISIZE,2,0,12,1280)
26 CALL ERFMP(1,IE)
27 WRITE(6,200)
28 200 FORMAT("NUMERO DE PUNTOS DE CADA BARRIDO A LEER : _")
29 READ(1,*) NP
30 WRITE(6,400)
31 400 FORMAT("POR QUE FACTOR DEBES MULTIPLICAR PARA QUE LAS"
32 " UNIDADES SEAN MILIKELVIN: _")
33 FAC = 1.
34 READ(1,*) FAC
35 WRITE(6,201)
36 201 FORMAT("COLOCAR EN LA LECTORA DE CINTAS LA CINTA CORRESPONDIENTE"
37 "A LA VIA"/"QUE DENOMINAREMOS (VIA 1)")
38 4 WRITE(6,202)
39 202 FORMAT("/"NUMERO DE BARRIDOS A LEER ((50) : _")
40 READ(1,*) NB
41 IF(NB.LE.0.OR.NB.GT.50) GO TO 4
42 IBF(7) = NP
43 IBF(8) = 0
44 C
45 DO 5 I=1,NB
46 READ(5) NAM,RNAM,IVIA,M29,M30,M31,IBARR,SGM
47 READ(5) (BF(J),J=1,NP)
48 IBF(1) = NAM
49 RNM = RNAM
50 IBF(4) = M29
51 IBF(5) = M30
52 IBF(6) = MOD(M31,1900)
53 WRITE(6,203) NAM,RNAM,IVIA,M29,M30,IBF(6)
54 203 FORMAT("FUENTE: "A1,I7,3X"VIA" I2,3X"FECHA"3I3)
55 DO 35 J=1,NP
56 35 BF(J) = BF(J)*FAC
57 CALL WRITF(IDCB,IE,IBF)
58 5 CALL ERFMP(2,IE)
59 CALL RWNDF(IDCB,IE)
60 CALL ERFMP(3,IE)

```

```

61 C
62 WRITE(6,300)
63 300 FORMAT("VAS A LEER OTRA CINTA COMO (VIA 2) ? (SI=1) : _")
64 READ(1,*) ICR
65 IF(ICR.NE.1) GO TO 99
66 WRITE(6,301) NP
67 301 FORMAT(" NUMERO DE PUNTOS = "I5
68 &/"COLOCA LA NUEVA CINTA Y CONTESTA CUALQUIER COSA")
69 READ(1,101) NAME
70 C
71 6 WRITE(6,202)
72 READ(1,*) NBP
73 IF(NBP.LE.0.OR.NBP.GT.NB) GO TO 6
74 WRITE(6,302)
75 302 FORMAT("CORRIMIENTO ENTRE AMBOS CONJUNTOS DE BARRIDOS : _")
76 READ(1,*) ICR
77 IF(ICR.LE.0) GO TO 9
78 IF(ICR+NBP.GT.NB) NBP=NB-ICR
79 DO 8 I=1,ICR
80 CALL READF(IDC B,IE,IBF)
81 8 WRITE(6,203) IBF(1),RNM,IVIA,IBF(4),IBF(5),IBF(6)
82 9 DO 7 I=1,NBP
83 CALL READF(IDC B,IE,IBF)
84 READ(5) NAM,RNAM,IVIA,M29,M30,M31,IBARR,SGM
85 READ(5) (BF(J),J=317,NP+316)
86 WRITE(6,203) NAM,RNAM,IVIA,M29,M30,M31
87 DO 36 J=317,NP+316
88 36 BF(J) = BF(J)*FAC
89 CALL POSNT(IDC B,IE,-1)
90 CALL WRITF(IDC B,IE,IBF)
91 7 CALL ERFMP(4,IE)
92 99 CALL RWNDF(IDC B)
93 WRITE(6,304)
94 304 FORMAT("DESEAS UN LISTADO DE LAS CABECERAS ? (SI=1) : _")
95 READ(1,*) ICR
96 IF (ICR.NE.1) GO TO 999
97 DO 11 I=1,NB
98 CALL READF(IDC B,IE,IBF)
99 11 WRITE(6,303) I,IBF(1),RNM,(IBF(J),J=4,7)
100 303 FORMAT("REGISTRO "I3,5X,A1,I7,5X"FECHA"3I3,5X"NPUNT="I4)
101 999 CALL CLOSE(IDC B)
102 END

```



```

1  FTN,L
2  PROGRAM TRASV
3  C      83.03.29
4  DIMENSION IDCB(720),IBF(704),BF(316)
5  DIMENSION NAME(3),ISIZE(2)
6  DIMENSION LLL(12672),A(6336)
7  EQUIVALENCE (IBF(2),RNM),(IBF(8),SG),(IBF(51),BF(1))
8  EQUIVALENCE (LLL(1),A(1))
9  C      INICIALIZACIONES
10 C
11 DO 21 I=1,704
12 21  IBF(I) = 0
13 DO 22 I=1,12672
14 22  LLL(I) = 0
15      NB = 32767
16 C
17 C
18 WRITE(6,100)
19 100  FORMAT("NOMBRE DEL FICHERO DISCO DONDE GRABAREMOS LOS"
20      " DATOS : _")
21 READ(1,101) NAME
22 101  FORMAT(3A2)
23 3    WRITE(6,102)
24 102  FORMAT("HAY QUE CREARLO (1) O SOLO ABRIR UNO YA EXISTENTE (0)"
25      " ? : _")
26 READ(1,*) ICR
27 IF(ICR.NE.1.AND.ICR.NE.0) GO TO 3
28 IF(ICR.EQ.0) CALL OPEN(IDCB,IE,NAME)
29 ISIZE(1) = 330
30 ISIZE(2) = 704
31 IF(ICR.EQ.1) CALL CREAT(IDCB,IE,NAME,ISIZE,2)
32 CALL ERFMP(1,IE)
33 WRITE(6,108)
34 108  FORMAT("NUMERO DE MAPAS A LEER ? : _")
35 READ(1,*) NMAP
36 WRITE(6,104)
37 104  FORMAT("NUMERO DE PUNTOS DE CADA BARRIDO : _")
38 READ(1,*) NP
39 C
40 C
41 C
42 DO 49 MAPA = 1,NMAP
43 90  WRITE(6,105) MAPA
44 105  FORMAT("NUMERO DE BARRIDOS A LEER (MAPA"I3") : _")
45 READ(1,*) NBL
46 IF(NBL.GT.18) GO TO 90
47 IF(NB.GT.NBL) NB = NBL
48 WRITE(6,106)
49 106  FORMAT("PON EN MARCHA LA LECTORA Y COLOCA LA CINTA"/
50      "A CONTINUACION RESPONDE CUALQUIER COSA")
51 READ(1,101) NAME
52 C
53 C
54 DO 10 L=1,NBL
55 READ(5) NAM,RNAM,IVIA,M29,M30,M31,IBARR,SGM
56 READ(5) (BF(I),I=1,NP)
57 IBF(1) = NAM
58 RNM = RNAM
59 IBF(4) = NP
60 IBF(5) = 1

```

```
61      IBF(6) = IBARR
62      IBF(7) = 15
63      SG = SGM
64      DO 199 I=1,NP
65 199    BF(I) = BF(I)/NMAP
66      DO 204 I=1,NP/2
67      TEMP = BF(I)
68      BF(I) = BF(NP-I+1)
69 204    BF(NP-I+1) = TEMP
70      DO 14 J=1,NP
71      K = (L-1)*352+J+25
72 14    A(K) = A(K) + BF(J)
73      IF(MAPA.NE.1) GO TO 10
74      DO 325 J=1,50
75      K = (L-1)*704+J
76 325    LLL(K) = IBF(J)
77 10    CONTINUE
78 49    CONTINUE
79      C
80      C
81      DO 378 I=1,NB
82      DO 377 J=1,704
83      K = (I-1)*704+J
84 377    IBF(J) = LLL(K)
85      CALL WRITF(IDCB,IE,IBF)
86      CALL ERFMP(4,IE)
87 378    CONTINUE
88      CALL CLOSE(IDCB,IE)
89      CALL ERFMP(5,IE)
90      END
```

```

1 FTN,L
2 C
3 C*****
4 C
5 C   PROGRAMA DE ANALISIS PARA OBSERVACIONES EN EL CONTINUO.
6 C
7 C   PERMITE EL TRATAMIENTO Y AJUSTE DE BARRIDOS DE HASTA 316
8 C   PUNTOS.
9 C
10 C*****
11 C
12 C   PROGRAM ANA,3
13 C
14 C           83.04.08
15 C
16 C   ESTE PROGRAMA USA TRES FICHEROS DISCO CREADOS
17 C   ANTERIORMENTE DE LA SIGUIENTE MANERA:
18 C
19 C           CR,DATXX::12:2:500:1280 (EN PROGRAMAS 'SNR')
20 C           CR,INTXX::12:2:375:1920 (25 REGISTROS)
21 C           CR,FINXX::12:2:330:704 (60 REGISTROS)
22 C
23 C   DIMENSION IPRG(3)
24 C EL COMMON QUE SIGUE DEBE SER MODIFICADO CON PRECAUCION:
25 C LOS VALORES QUE VAN DESDE 'INAM,...,T(316)' FORMAN UNA UNIDAD
26 C QUE NO DEBE SER ROTA. SI ACASO ES NECESARIO ANYADIR VARIABLES
27 C HACERLO TRAS 'FE' O TRAS 'SIG'.
28 C   COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
29 C   COMMON R(316),S(316),T(316),NREGI,FE
30 C   COMMON NSEP,IPIFE(20),IPAE(20),SIG
31 C   COMMON IDCB(1296),IDCC(1936),IDCD(720),NAME(3)
32 C
33 C   MANDO(1) = 2HMA
34 C   MANDO(2) = 2HND
35 C   MANDO(3) = 2HO
36 C   IPRG(1) = 2HEN
37 C   IPRG(2) = 2HTR
38 C   IPRG(3) = 2HA
39 C   CALL EXEC(8,IPRG)
40 C   END
41 C
42 C*****
43 C
44 C   PROGRAM ENTRA,5
45 C ESTE PROGRAMA NO ES ACCESIBLE A PARTIR DE UN COMANDO.
46 C INICIALIZA VARIABLES Y ABRE FICHEROS (LOS TRES ANTES CITADOS).
47 C SI ALGUNO DE LOS FICHEROS (SALVO EL DE DATOS) NO EXISTE
48 C ES CREADO AUTOMATICAMENTE, EN EL CARTRIDGE 12.
49 C   DIMENSION ISIZE(2)
50 C   COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
51 C   COMMON R(316),S(316),T(316),NREGI,FE
52 C   COMMON NSEP,IPIFE(20),IPAE(20),SIG
53 C   COMMON IDCB(1296),IDCC(1936),IDCD(720),NAME(3)
54 C   WRITE(6,6)
55 6   FORMAT(/10X"PROGRAM ANA"10X"** 83.04.19 **"//
56 C   "NOMBRE FICHERO DATOS INICIALES ?      ")
57 C   READ(1,101) NAME
58 101  FORMAT(3A2)
59 C   CALL OPEN(IDCB,IE,NAME)
60 C   CALL ERFMP(1,IE)

```

```

61      WRITE(6,102)
62 102  FORMAT("NOMBRE FICHERO RESULTADOS INTERMEDIOS? _")
63      READ(1,101) NAME
64      CALL OPEN(IDCC,IE,NAME)
65      CALL ERFMP(-2,IE)
66      IF(IE.GE.0) GO TO 106
67      ISIZE(1) = 375
68      ISIZE(2) = 1920
69      CALL CREAT(IDCC,IE,NAME,ISIZE,2,0,12)
70      CALL ERFMP(2,IE)
71      IF(IE.GE.0) WRITE(6,77)
72 77   FORMAT("FICHERO CREADO")
73 106  WRITE(6,103)
74 103  FORMAT("NOMBRE FICHERO RESULTADOS FINALES? _")
75      READ(1,101) NAME
76      CALL OPEN(IDCDC,IE,NAME)
77      CALL ERFMP(-3,IE)
78      IF(IE.GE.0) GO TO 107
79      ISIZE(1) = 330
80      ISIZE(2) = 704
81      WRITE(6,666)
82 666  FORMAT("CODIGO DE SEGURIDAD (2 LETRAS) : _")
83      READ(1,667) KODE
84      IF(KODE.EQ.30060B) KODE = 0
85 667  FORMAT(A2)
86      CALL CREAT(IDCDC,IE,NAME,ISIZE,2,KODE,12)
87      CALL ERFMP(3,IE)
88      IF(IE.GE.0) WRITE(6,77)
89 107  N1 = 0
90      N2 = 0
91      NREGI = 0
92      FE = 0.025
93      NSEP = 0
94      DO 119 I=1,316
95          R(I) = 0
96          S(I) = 0
97 119  T(I) = 0
98      DO 139 I=1,20
99          IPIFE(I) = 0
100 139  IPAE(I) = 0
101      CALL EXEC(8,MANDO)
102      END
103  C
104  C*****
105  C
106      PROGRAM DUMPD,5
107  C ESTE PROGRAMA LISTA EL CONTENIDO DEL FICHERO DE DATOS
108  C (COMPLETO O SOLO UNA PARTE)
109      DIMENSION IBFP(1280)
110      COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
111      COMMON R(316),S(316),T(316),NREGI,FE
112      COMMON NSEP,IPIFE(20),IPAE(20),SIG
113      COMMON IDCB(1296),IDCC(1936),IDCD(720)
114      EQUIVALENCE (IBFP(2),RNM)
115      WRITE(6,200)
116 200  FORMAT("REGISTROS INICIAL Y FINAL A LEER ? (99=TODOS) : _")
117      NF = 50
118      READ(1,*) NI,NF
119      IF(NI.EQ.99) NI=1
120      IF(NF.LT.NI) NF=NI

```

```

21 DO 210 I=NI,NF
22 CALL APOSN(IDC B,IE,I)
123 CALL ERFMP(44,IE)
124 CALL READF(IDC B,IE,IBFP)
25 CALL ERFMP(45,IE)
126 210 WRITE(6,220) I,IBFP(1),RNM,(IBFP(J),J=4,7)
127 220 FORMAT(" REGISTRO:"I4,4X"FUENTE: "A2,I7,4X"FECHA/HORA:"3I3,4X
28 "NPUNT="I3)
129 CALL EXEC(8,MANDO)
130 END
31 C
32 C*****
133 C
134 PROGRAM LECFD,5
35 C PROGRAMA DE LECTURA DE DATOS EN UN FICHERO GENERADO CON UN
136 C PROGRAMA DE TIPO 'SNR', QUE LOS NORMALIZA DE UNA DETERMINADA
137 C MANERA.
38 DIMENSION IBFP(1280),IBF(1264)
39 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
140 COMMON R(316),S(316),T(316),NREGI,FE
41 COMMON NSEP,IPIFE(20),IPAE(20),SIG
42 COMMON IDC B(1296),IDCC(1936),IDCD(720)
143 EQUIVALENCE (IBFP(2),RNM),(IBF(1),BF(1))
144 WRITE(6,201)
45 201 FORMAT("NUMERO DEL REGISTRO 'DATOS' A LEER? _")
146 READ(1,*) NR
147 CALL APOSN(IDC B,IE,NR)
48 CALL ERFMP(4,IE)
149 CALL READF(IDC B,IE,IBFP)
150 CALL ERFMP(5,IE)
51 DO 209 I=1,1264
52 209 IBF(I) = IBFP(I+8)
153 NP = IBFP(7)
154 INAM = IBFP(1)
55 RNAM = RNM
156 ICODE = 1
157 C INVERTIMOS LOS VECTORES DE DATOS PARA OBTENER UNA
158 C REPRESENTACION DE LOS BARRIDOS TAL COMO ESTAN EN EL CIELO.
159 DO 204 I=1,NP/2
160 TEMP = BF(I)
61 BF(I) = BF(NP-I+1)
62 204 BF(NP-I+1) = TEMP
163 DO 205 I=317,316+NP/2
64 TEMP = BF(I)
65 BF(I) = BF(633+NP-I)
166 205 BF(633+NP-I) = TEMP
167 CALL PAGE(2)
68 WRITE(6,202) NR,INAM,RNAM,(IBFP(J),J=4,7)
69 202 FORMAT("REGISTRO:"I4,5X,A2,I7,5X"FECHA/HORA:"3I3,5X"NPUNT="I4)
170 CALL GRAF(BF(1),NP,0.,500.,1000.,250.,10,A)
71 CALL GRAF(BF(317),NP,0.,230.,1000.,250.,10,A)
72 CALL PLOT(0,0,210,0,0)
173 WRITE(6,203)
174 203 FORMAT("QUE VIAS ACEPTAS? (0,1,2,12): _")
75 READ(1,*) IVIA
176 CALL EXEC(8,MANDO)
177 END
78 C
79 C*****
180 C

```

```

81 PROGRAM GRAFD,5
82 C REALIZA UNA GRAFICA SIN ESCALA DE AMBAS VIAS DEL ULTIMO
83 C BARRIDO LEIDO
84 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
85 CALL PAGE(2)
86 WRITE(6,328) INAM,RNAM,NP
87 328 FORMAT("FUENTE ",A2,I7,5X"NPUNT ="I4)
88 CALL GRAF(BF,NP,0.,500.,1000.,250.,10,A)
89 CALL GRAF(BF(317),NP,0.,230.,1000.,250.,10,A)
90 CALL PLOT(0,0,210,0,0)
91 CALL EXEC(8,MANDO)
92 END
93 C
94 C*****
95 C
96 PROGRAM SUAVI,5
97 C HACE UN SUAVIZADO (O FILTRADO) 'HANNING' DE LA VIA DESEADA
98 C DE LOS DATOS CONTENIDOS EN LOS VECTORES INTERMEDIOS (CODIGO = 2)
99 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
200 COMMON R(316),S(316),T(316),NREGI,FE
201 WRITE(6,6)
202 6 FORMAT("QUE VIA DESEAS SUAVIZAR ? (1,2,12, LAS 3) : _")
203 READ(1,*) IVI
204 IF(IVI.EQ.1.OR.IVI.EQ.3) CALL HANN(R,1,NPM)
205 IF(IVI.EQ.2.OR.IVI.EQ.3) CALL HANN(S,1,NPM)
206 IF(IVI.EQ.12.OR.IVI.EQ.3) CALL HANN(T,1,NPM)
207 ICODE = ICODE*2
208 CALL EXEC(8,MANDO)
209 END
210 C
211 C*****
212 C
213 PROGRAM SUAVD,5
214 C REALIZA UN SUAVIZADO (O FILTRADO) 'HANNING' DE LA VIA DESEADA
215 C DE LOS DATOS LEIDOS.
216 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
217 WRITE(6,6)
218 6 FORMAT("QUE VIA DESEAS SUAVIZAR ? (1,2,12) : _")
219 READ(1,*) IVI
220 IF(IVI.EQ.1.OR.IVI.EQ.12) CALL HANN(BF,1,NP)
221 IF(IVI.EQ.2.OR.IVI.EQ.12) CALL HANN(BF(317),1,NP)
222 ICODE = ICODE*2
223 CALL EXEC(8,MANDO)
224 END
225 C
226 C*****
227 C
228 PROGRAM INTE,5
229 C REALIZA LA INTEGRACION EN LOS VECTORES INTERMEDIOS DE LOS
230 C DATOS ANTES LEIDOS Y TRATADOS.
231 C SE PUEDEN INTEGRAR BARRIDOS DE DISTINTA LONGITUD Y CON
232 C CORRIMIENTOS RELATIVOS (EN CUALQUIER SENTIDO), PERO
233 C TOMA COMO REFERENCIA SIEMPRE EL PRIMER BARRIDO LEIDO
234 C PERDIENDOSE LOS DATOS QUE SE SALGAN FUERA DEL ESPACIO
235 C CUBIERTO POR AQUEL.
236 C SOLO SE PUEDEN INTEGRAR DOS TIPOS DISTINTOS DE BARRIDOS:
237 C SI HAY MAS DE DOS, EL RESULTADO ES ERRONEO.
238 C SI SE VA A INICIAR UNA NUEVA INTEGRACION SIN HABER
239 C GRABADO EL RESULTADO DE LA ANTERIOR EN EL FICHERO INTERMEDIO
240 C O HABIENDO LEIDO DATOS DEL FICHERO INTERMEDIO, DADO QUE LOS

```

```

241 C VECTORES USADOS COMO INTERMEDIOS SON SIEMPRE LOS MISMOS
242 C ES IMPRESCINDIBLE BORRARLOS, PARA LO QUE BASTA LLAMAR AL
243 C PROGRAMA 'ANULI' ANTES DE LLAMAR A ESTE.
244     COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
245     COMMON R(316),S(316),T(316),NREGI,FE
246     IF (IVIA.EQ.0) GO TO 707
247 711  WRITE(6,700)
248 700  FORMAT("CORRIMIENTO HACIA LA DERCHA RESPECTO DEL PRIMER "
249        ""BARRIDO DE LA INTEGRACION"/"(ESTE CORRIMIENTO HAY QUE "
250        ""EXPRESARLO EN NUMERO DE PUNTOS) ; _")
251     ISHIF = 0
252     READ(1,*) ISHIF
253     N12 = N1+N2
254     IF(N12.EQ.0) NPM = NP
255     NM = MAX0(ISHIF+1,1)
256     NL = MIN0(ISHIF+NP,NPM)
257     IF(IVIA.EQ.2) GO TO 717
258     N1 = N1+1
259     DO 703 I=NM,NL
260 703  R(I) = (R(I)*(N1-1)+BF(I-ISHIF))/N1
261 717  IF(IVIA.EQ.1) GO TO 727
262     N2 = N2+1
263     DO 713 I=NM,NL
264 713  S(I) = (S(I)*(N2-1)+BF(I-ISHIF+316))/N2
265 727  IF(IVIA.NE.1.AND.IVIA.NE.12) GO TO 725
266     N12 = N12+1
267     DO 724 I=NM,NL
268 724  T(I) = (T(I)*(N12-1)+BF(I-ISHIF))/N12
269 725  IF(IVIA.NE.2.AND.IVIA.NE.12) GO TO 707
270     N12 = N12+1
271     DO 726 I=NM,NL
272 726  T(I) = (T(I)*(N12-1)+BF(I-ISHIF+316))/N12
273 707  CALL EXEC(8,MANDO)
274     END
275 C
276 C*****
277 C
278     PROGRAM RESTD,5
279 C RESTA EL VALOR MINIMO A LA(S) VIA(S) CONSIDERADA(S) BUENA(S)
280 C EN LOS DATOS RECIEN LEIDOS.
281     COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
282     IF(IVIA.EQ.2) GO TO 72
283     CALL AMIN(BF(1),NP,YMIN,M)
284     DO 71 I=1,NP
285 71  BF(I) = BF(I)-YMIN
286 72  IF(IVIA.EQ.1) GO TO 74
287     CALL AMIN(BF(317),NP,YMIN,M)
288     DO 73 I=1,NP
289     K = I+316
290 73  BF(K) = BF(K)-YMIN
291 74  CALL EXEC(8,MANDO)
292     END
293 C
294 C*****
295 C
296     PROGRAM RESTI,5
297 C RESTA EL VALOR MINIMO A CADA UNO DE LOS VECTORES INTERMEDIOS
298 C (CORRESPONDIENTES A LAS VIAS 1, VIAS 2 Y VIAS 1+2)
299     COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
300     COMMON R(316),S(316),T(316),NREGI,FE

```

```

301 CALL AMIN(R,NPM,RMIN,M)
302 CALL AMIN(S,NPM,SMIN,M)
303 CALL AMIN(T,NPM,TMIN,M)
304 DO 17 I=1,NPM
305 R(I) = R(I)-RMIN
306 S(I) = S(I)-SMIN
307 17 T(I) = T(I)-TMIN
308 CALL EXEC(8,MANDO)
309 END
310 C
311 C*****
312 C#
313 PROGRAM MANDO,5
314 C ESTE PROGRAMA LEE Y VERIFICA LOS COMANDOS QUE SE LE DAN
315 C AL PROGRAMA.
316 DIMENSION MANDO(3),IVERF(81)
317 DATA IVERF/2HLE,2HCF,2HD ,2HIN,2HTE,2H ,2HSU,2HAV,2HI ,
318 ,2HFI,2HN ,2H ,2HFA,2HCT,2HD ,2HGR,2HAF,2HI ,2HRE,2HST,2HD ,
319 ,2HGR,2HAF,2HD ,2HSU,2HAV,2HD ,2HLI,2HST,2HD ,2HLI,2HST,2HI ,
320 ,2HCO,2HMP,2HD ,2HES,2HCF,2HI ,2HLE,2HCF,2HI ,2HCU,2HMP,2HI ,
321 ,2HAN,2HUL,2HI ,2HCO,2HMP,2HI ,2HAJ,2HUS,2HI ,2HRE,2HST,2HI ,
322 ,2HSA,2HLT,2HD ,2HSA,2HLT,2HI ,2HAJ,2HUS,2H2 ,2HMO,2HNT,2HI ,
323 ,2HES,2HCF,2HA ,2HLE,2HCF,2HA ,2HCU,2HMP,2HA ,2HCU,2HMP,2HD /
324 DATA NVERF/27/
325 8999 CALL PAGE(3)
326 WRITE(6,9000)
327 9000 FORMAT("_ ")
328 READ(1,9001) MANDO
329 9001 FORMAT(3A2)
330 DO 9010 I=1,NVERF
331 K = (I-1)*3
332 IF(MANDO(1).EQ.IVERF(K+1).AND.MANDO(2).EQ.IVERF(K+2).AND.
333 .MANDO(3).EQ.IVERF(K+3)) GO TO 9012
334 9010 CONTINUE
335 WRITE(6,9009) MANDO
336 9009 FORMAT("ERROR:"5X"COMANDO '"3A2"' NO VALIDO"/)
337 GO TO 8999
338 9012 CALL EXEC(8,MANDO)
339 END
340 C
341 C*****
342 C
343 PROGRAM LISTD,5
344 C LISTA LOS VALORES NUMERICOS DE LA(S) VIA(S) CONSIDERADA(S)
345 C VALIDA(S) DE LOS DATOS LEIDOS.
346 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
347 CALL PAGE(2)
348 WRITE(6,329) INAM,RNAM
349 329 FORMAT("FUENTE "A2,I7" ")
350 IF(IVIA.EQ.2) GO TO 323
351 WRITE(6,321)
352 321 FORMAT("VIA 1"/)
353 WRITE(6,320) (BF(J),J=1,NP)
354 320 FORMAT(/(10I7))
355 323 IF(IVIA.EQ.1) GO TO 326
356 WRITE(6,324)
357 324 FORMAT(/16X"VIA 2"/)
358 WRITE(6,320) (BF(J),J=317,NP+316)
359 326 CALL EXEC(8,MANDO)
360 END

```



```

361 C
362 C*****
363 C
364     PROGRAM LISTI,5
365 C LISTA LOS VALORES NUMERICOS DE LOS DATOS CONTENIDOS EN LOS
366 C VECTORES INTERMEDIOS, SEAN RESULTADO DE UNA INTEGRACION O
367 C SEAN LEIDOS DEL FICHERO INTERMEDIO.
368     COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
369     COMMON R(316),S(316),T(316),NREGI,FE
370     CALL PAGE(2)
371     WRITE(6,331) INAM,RNAM
372 331  FORMAT("FUENTE "A2,I7)
373     WRITE(6,330) (R(I),I=1,NPM)
374     WRITE(6,330) (S(I),I=1,NPM)
375     WRITE(6,330) (T(I),I=1,NPM)
376 330  FORMAT(/(10I7))
377     CALL EXEC(8,MANDO)
378     END
379 C
380 C*****
381 C
382     PROGRAM FACTD,5
383 C PERMITE MULTIPLICAR CADA UNA DE LAS VIAS DE LOS DATOS LEIDOS
384 C POR UN FACTOR.
385 C POR DEFECTO TOMA FACTOR = 1.
386 C REALMENTE SOLO ACTUA SOBRE LA(S) VIA(S) CONSIDERADA(S)
387 C VALIDA(S).
388     COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
389     FAC1 = 1
390     FAC2 = 1
391     WRITE(6,61)
392 61  FORMAT("POR QUE FACTORES DESEAS MULTIPLICAR LAS VIAS 1 Y 2 ? _")
393     READ(1,*) FAC1,FAC2
394     IF(IVIA.EQ.2) GO TO 63
395     DO 62 I=1,NP
396 62  BF(I) = BF(I)*FAC1
397 63  IF(IVIA.EQ.1) GO TO 65
398     DO 64 I=1,NP
399     K = I+316
400 64  BF(K) = BF(K)*FAC2
401 65  CALL EXEC(8,MANDO)
402     END
403 C
404 C*****
405 C
406     PROGRAM COMPD,5
407 C COMPRIME LOS DATOS (DE LAS VIAS ..... ) PROMEDIANDO PARES DE
408 C PUNTO (CODIGO = 3). EL NUMERO DE PUNTOS SE REDUCE A LA MITAD.
409     COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
410     IF(IVIA.EQ.2) GO TO 410
411     DO 411 I=1,NP,2
412     J=(I+1)/2
413 411  BF(J) = 0.5*(BF(I)+BF(I+1))
414 410  IF(IVIA.EQ.1) GO TO 412
415     DO 413 I=1,NP,2
416     J=(I+1)/2+316
417 413  BF(J) = 0.5*(BF(316+I)+BF(317+I))
418 412  NP = NP/2
419     ICODE = ICODE*3
420     CALL EXEC(8,MANDO)

```

```

21      END
22      C
23      C*****
24      C
25      PROGRAM COMPI,5
26      C PROMEDIA PARES DE VALORES DE CADA UNO DE LOS VECTORES
27      C INTERMEDIOS, REDUCIENDO ASI SU LONGITUD A LA MITAD (CODIGO = 3)
28      COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
29      COMMON R(316),S(316),T(316),NREGI,FE
30      DO 55 I=1,NPM,2
31      I1 = I+1
32      J = I1/2
33      R(J) = 0.5*(R(I)+R(I1))
34      S(J) = 0.5*(S(I)+S(I1))
35      55 T(J) = 0.5*(T(I)+T(I1))
36      NPM = NPM/2
37      ICODE = ICODE*3
38      CALL EXEC(8,MANDO)
39      END
40      C
41      C*****
42      C
43      PROGRAM ESCFI,5
44      C ESCRIBE LOS VALORES ALMACENADOS EN LOS VECTORES INTERMEDIOS
45      C (Y YA TRATADOS CONVENIENTEMENTE) EN UN REGISTRO A ESCOGER
46      C DEL FICHERO INTERMEDIO.
47      C SE RECOMIENDA GRABARLOS DE MANERA ORDENADA (DECLINACIONES
48      C CRECIENTES SEGUN AUMENTA EL NUMERO DEL REGISTRO)
49      C TRAS ESTA ESCRITURA LOS VECTORES INTERMEDIOS QUEDAN
50      C ANULADOS A EFECTOS DE POSTERIORES INTEGRACIONES, A
51      C NO SER QUE MEDIE ENTRE ESTA 'ESCFI' Y LA 'INTE'
52      C UNA LECTURA DE DATOS EN EL FICHERO INTERMEDIO.
53      DIMENSION IBF(1920)
54      COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
55      COMMON R(316),S(316),T(316),NREGI,FE
56      COMMON NSEP,IPIFE(20),IPAE(20),SIG
57      COMMON IDCB(1296),IDCC(1936),IDCD(720)
58      EQUIVALENCE (IBF(1),INAM)
59      92 WRITE(6,93)
60      93 FORMAT("NUMERO DE REGISTRO EN FICHERO INTERMEDIO: _")
61      READ(1,*) NREGI
62      IF(NREGI.GT.25.OR.NREGI.LE.0) GO TO 92
63      CALL APOSN(IDCC,IE,NREGI)
64      CALL ERFMP(6,IE)
65      CALL WRITF(IDCC,IE,IBF)
66      CALL ERFMP(7,IE)
67      CALL RWNDF(IDCC)
68      N1 = 0
69      N2 = 0
70      CALL EXEC(8,MANDO)
71      END
72      C
73      C*****
74      C
75      PROGRAM LECFI,5
76      C LEE LOS DATOS CONTENIDOS EN UN REGISTRO DEL FICHERO INTER-
77      C MEDIO INTRODUCIENDOLOS EN LOS VECTORES INTERMEDIOS, QUE
78      C PERDERAN LA INFORMACION QUE ANTES POSEIAN.
79      C APARECE LA GRAFICA DE LA VIA DESEADA (1,2, 1+2) CON UNA
80      C ESCALA A ESCOGER, O BIEN LAS TRES GRAFICAS PERO SIN

```

```

481 C ESCALA ALGUNA.
482 DIMENSION IBF(1920)
483 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
484 COMMON R(316),S(316),T(316),NREGI,FE
485 COMMON NSEP,IPIFE(20),IPAE(20),SIG
486 COMMON IDCB(1296),IDCC(1936),IDCD(720)
487 EQUIVALENCE (IBF(1),INAM)
488 96 WRITE(6,97)
489 97 FORMAT("REGISTRO A LEER EN FICHERO INTERMEDIO? _")
490 READ(1,*) NREGI
491 IF(NREGI.LE.0.OR.NREGI.GT.25) GO TO 96
492 CALL APOSN(IDCC,IE,NREGI)
493 CALL ERFMP(8,IE)
494 CALL READF(IDCC,IE,IBF)
495 CALL ERFMP(9,IE)
496 701 WRITE(6,702)
497 702 FORMAT("QUE VIA DESEAS REPRESENTAR? (1,2,12,LAS 3) ;_")
498 READ(1,*) IVI
499 IF(IVI.EQ.3) GO TO 710
500 IF(IVI.NE.1.AND.IVI.NE.2.AND.IVI.NE.12.AND.IVI.NE.0) GO TO 701
501 IF(IVI.EQ.0) GO TO 709
502 CALL PAGE(2)
503 WRITE(6,704)
504 704 FORMAT("FACTOR DE ESCALA? (PTOS/K) = _")
505 FES = 0
506 READ(1,*) FES
507 IF(FES.NE.0.0) FE=FES/10000
508 WRITE(6,705) NREGI,INAM,RNAM,ICODE
509 705 FORMAT("REGISTRO "I2,5X"FUENTE "A2,I7,5X"CODIGO "I2)
510 K=(MOD(IVI,9)-1)*316+1
511 CALL GRAF2(R(K),NPM,0.,410.,1000.,300.,10,FE)
512 CALL NUMBR(0.,388.,1000.,NPM,10)
513 CALL PLOT(0,0,360,0,0)
514 709 CALL EXEC(8,MANDO)
515 710 CALL PAGE(2)
516 WRITE(6,705) NREGI,INAM,RNAM,ICODE
517 CALL GRAF(R,NPM,0.,550.,1000.,200.,10,A)
518 CALL GRAF(S,NPM,0.,340.,1000.,200.,10,A)
519 CALL GRAF(T,NPM,0.,130.,1000.,200.,10,A)
520 CALL PLOT(0,0,110,0,0)
521 GO TO 709
522 END
523 C
524 C*****
525 C
526 PROGRAM GRAFI,5
527 C REALIZA LA GRAFICA DE LA VIA DESEADA (1, 2, 1+2) DE LOS
528 C VALORES CONTENIDOS EN LOS VECTORES INTERMEDIOS (COMO RESULTADO
529 C DE INTEGRACION O DE LECTURA DEL FICHERO INTERMEDIO) CON LA
530 C ESCALA DESEADA, O BIEN LA GRAFICA DE LAS TRES (3) VIAS SIN
531 C ESCALA.
532 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
533 COMMON R(316),S(316),T(316),NREGI,FE
534 701 WRITE(6,702)
535 702 FORMAT("QUE VIA DESEAS REPRESENTAR? (1,2,12,LAS 3) _")
536 READ(1,*) IVI
537 IF(IVI.EQ.3) GO TO 710
538 IF(IVI.NE.1.AND.IVI.NE.2.AND.IVI.NE.12.AND.IVI.NE.0) GO TO 701
539 IF(IVI.EQ.0) GO TO 709
540 CALL PAGE(2)

```

```

41 FES = 1
42 WRITE(6,704)
543 704 FORMAT("FACTOR DE ESCALA? (PTOS/K) : _")
44 READ(1,*) FES
45 IF(FES.NE.1) FE = FES/10000
546 WRITE(6,705) NREGI,INAM,RNAM,ICODE
547 705 FORMAT("REGISTRO "I2,5X"FUENTE "A2,I7,5X"CODIGO "I2)
48 K = (MOD(IVI,9)-1)*316+1
549 CALL GRAF2(R(K),NPM,0.,410.,1000.,300.,10,FE)
550 CALL NUMBR(0.,388.,1000.,NPM,10)
551 CALL PLOT(0,0,360,0,0)
552 709 CALL EXEC(8,MANDO)
553 710 CALL PAGE(2)
554 WRITE(6,705) NREGI,INAM,RNAM,ICODE
555 CALL GRAF(R,NPM,0.,550.,1000.,200.,10,A)
556 CALL GRAF(S,NPM,0.,340.,1000.,200.,10,A)
557 CALL GRAF(T,NPM,0.,130.,1000.,200.,10,A)
558 CALL PLOT(0,0,110,0,0)
559 GO TO 709
560 END

```

61 C  
62 C\*\*\*\*\*  
63 C

```

64 PROGRAM DUMPI,5
65 C LISTA LAS CABECERAS DE LOS 25 REGISTROS DEL FICHERO INTERMEDIO
566 DIMENSION IBF(1920)
567 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
68 COMMON R(316),S(316),T(316),NREGI,FE
69 COMMON NSEP,IPIFE(20),IPAE(20),SIG
570 COMMON IDCB(1296),IDCC(1936),IDCD(720)
71 EQUIVALENCE (IBF(2),RNM)
72 CALL RWNDF(IDCC,IE)
573 CALL ERFMP(95,IE)
574 WRITE(6,33)
75 33 FORMAT("QUE REGISTROS ? (99= TODOS) : _")
576 NF = 25
577 READ(1,*) NI,NF
78 IF(NI.EQ.99) NI = 1
79 DO 97 I=NI,NF
580 CALL READF(IDCC,IE,IBF,1920,LEN)
81 CALL ERFMP(100+I,IE)
82 IF(LEN.EQ.-1) GO TO 99
583 97 WRITE(6,98) I, IBF(1),RNM,IBF(4)
84 98 FORMAT("REGISTRO "I2" FUENTE "A2,I7" NPUNT ="I4)
85 99 CALL EXEC(8,MANDO)
86 END

```

87 C  
88 C\*\*\*\*\*  
89 C

```

90 PROGRAM ANULI,5
91 C ANULA EL CONTENIDO DE LOS TRES VECTORES INTERMEDIOS.
92 C EN CIERTAS CONDICIONES ES IMPRESCINDIBLE HACER TAL COSA
93 C ANTES DE EMPEZAR UNA INTEGRACION,
94 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
95 COMMON R(316),S(316),T(316),NREGI,FE
96 COMMON NSEP,IPIFE(20),IPAE(20),SIG
97 COMMON IDCB(1296),IDCC(1936),IDCD(720)
98 NREGI = 0
99 N1 = 0
600 N2 = 0

```

```

01      DO 45 I=1,948
02 45   R(I) = 0
603    CALL EXEC(8,MANDO)
04     END
05     C
606 C*****
607 C
08     PROGRAM AJUSI,5
09 C   PERMITE EL AJUSTE DE CUALQUIERA DE LOS. VECTORES INTERMEDIOS.
110 C SE PUEDEN EVITAR EN TAL AJUSTE INTERVALOS (DONDE SE ENCUENTREN
11 C EMISIONES POR ENCIMA DEL FONDO) O PUNTOS AISLADOS QUE POR
12 C ALGUNA RAZON NO HAYAN SIDO ELIMINADOS ANTERIORMENTE.
613 C TRAS EL AJUSTE ES DIBUJADA LA LINEA DE BASE EN SOBREIMPRESION.
14 C SI EL AJUSTE NO ES CONSIDERADO VALIDO, A LA SALIDA SE RECUPERA
15 C EL VECTOR INICIAL.
616 C SI EL AJUSTE ES VALIDO, APARECE LA GRAFICA DEL VECTOR RESIDUO
617 C QUE A SU VEZ ES ALMACENADO EN EL VECTOR DE ENTRADA.
18 C (EL POLINOMIO AJUSTADO POR MINIMOS CUADRADOS ES DE TERCER GRADO)
19 C (CODIGO = 5)
620     DIMENSION A(316),B(316)
621     COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
622     COMMON R(316),S(316),T(316),NREGI,FE
623     COMMON NSEP,IPIFE(20),IPAE(20),SIG
624     NSEP = 0
625     WRITE(6,31)
626 31   FORMAT("VIA A AJUSTAR? (1,2,12) : _")
627     READ(1,*) IVI
628     IF(IVI.NE.1.AND.IVI.NE.2.AND.IVI.NE.12) GO TO 39
629     CALL PAGE(2)
630     K = (MOD(IVI,9)-1)*316
631     DO 32 I=1,NPM
632     J = I+K
633 32   A(I) = R(J)
634     CALL AMIN(A,NPM,AMN,M)
635     DO 3312 I=1,NPM
636 3312 A(I) = A(I)-AMN
637     CALL SAP3(A,NPM,SIG,FE,B,NPA,IPIFE,IPAE)
638     CALL PLOT(0,0,757,0,0)
639     WRITE(6,40) NREGI,INAM,RNAM,IVI,ICODE,SIG
640 40   FORMAT("REGISTRO"14" FUENTE "A2,I7" VIA"I3" CODIGO "I2,
641     " SIGMA ="F7.0)
642     CALL GRAFE(B,NPM,0.,310.,1000.,400.,400,FE,100.)
643     CALL PLOT(0,0,50,0,0)
644     WRITE(6,33)
645 33   FORMAT("CONSIDERAS DEFINITIVO ESTE AJUSTE ? (SI=1) _")
646     READ(1,*) L
647     IF(L.NE.1) GO TO 39
648     ICODE = ICODE*5
649     DO 34 I=1,NPM
650     J = I+K
651 34   R(J) = A(I)-B(I)
652     CALL GRAFE(R(K+1),NPM,0.,260.,1000.,400.,100,FE,50.)
653     CALL PLOT(0,0,22,0,0)
654 39   CALL EXEC(8,MANDO)
655     END
656     C
657 C*****
658 C
59     PROGRAM AJUS2,5
660 C SE PUEDEN HACER LOS MISMOS COMENTARIOS QUE PARA EL PROGRAMA

```

```

61 C 'AJUSI' Y ANYADIR QUE:
62 C ESTE PROGRAMA PERMITE EL AJUSTE POR SEPARADO DE DOS
663 C TROZOS COMPLEMENTARIOS DE BARRIDO.
664     DIMENSION A(316),B(316),C(10)
665     COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
666     COMMON R(316),S(316),T(316),NREGI,FE
667     COMMON NSEP,IPIFE(20),IPAE(20),SIG
668     WRITE(6,31)
669 31  FORMAT("VIA A AJUSTAR ? (1,2,12) : _")
670     READ(1,*) IVI
671     IF(IVI.NE.1.AND.IVI.NE.2.AND.IVI.NE.12) GO TO 39
672     K = (MOD(IVI,9)-1)*316
673     DO 32 I=1,NPM
674     J = K+I
675 32  A(I) = R(J)
676     CALL AMIN(A,NPM,AMN,M)
677     DO 3312 I=1,NPM
678 3312 A(I) = A(I)-AMN
679     CALL PAGE (2)
680     CALL GRAF2(A,NPM,0.,400.,1000.,300.,10,FE)
681     CALL NUMBR(0.,378.,1000.,NPM,10)
682     CALL PLOT(0,0,350,0,0)
683     WRITE(6,66)
684 66  FORMAT("PUNTO DE SEPARACION ENTRE LOS DOS AJUSTES ? : _")
685     READ(1,*) NSEP
686     CALL PAGE(2)
687     CALL SAP3(A,NSEP+5,SIG1,FE,B,NPA1,IPIFE,IPAE)
688     CALL PLOT (0,0,757,0,0)
689     WRITE(6,40) NREGI,INAM,RNAM,IVI,ICODE,SIG1
690 40  FORMAT("REGISTRO"14" FUENTE "A2,I7" VIA"13" CODIGO "12,
691     " SIGMA ="F7.0)
692     CALL GRAFE(B,NSEP+5,0.,310.,1000.,400.,400,FE,100.)
693     CALL PAGE(1)
694     DO 175 I=NSEP-4,NSEP+5
695 175 C(I-NSEP+5) = B(I)
696     CALL SAP3(A(NSEP-4),NPM-NSEP+5,SIG2,FE,B(NSEP-4),NPA2,IPIFE,IPAE)
697     CALL PLOT(0,0,757,0,0)
698     WRITE(6,40) NREGI,INAM,RNAM,IVI,ICODE,SIG2
699     CALL GRAFE(B(NSEP-4),NPM-NSEP,0.,310.,1000.,400.,400,FE,100.)
700     CALL PAGE(1)
701     SIG = (SIG1**2*(NPA1-4)+SIG2**2*(NPA2-4))/(NPA1+NPA2-8)
702     SIG = SQRT(SIG)
703     WRITE(6,40) NREGI,INAM,RNAM,IVI,ICODE,SIG
704     CALL GRAFE(A,NPM,0.,310.,1000.,400.,10,FE,100.)
705     CALL NUMBR(0.,713.,1000.,NPM,10)
706     DO 543 I=NSEP-4,NSEP
707     J = I-NSEP+5
708     B(I) = (B(I)*J+C(J)*(10-J))*0.1
709 543 B(NSEP+6-J) = (B(NSEP+6-J)*(10-J) + C(11-J)*J)*0.1
710     CALL GRAFE(B,NPM,0.,310.,1000.,400.,400,FE,100.)
711     CALL PLOT(0,0,50,0,0)
712     WRITE(6,33)
713 33  FORMAT("CONSIDERAS DEFINITIVO ESTE AJUSTE ? (SI=1) : _")
714     READ(1,*) L
715     IF(L.NE.1) GO TO 39
716     ICODE = ICODE * 5
717     DO 34 I=1,NPM
718     J = I+K
719 34  R(J) = A(I)-B(I)
720     CALL GRAFE(R(K+1),NPM,0.,260.,1000.,400.,100,FE,50.)

```

```

721 CALL PLOT(0,0,22,0,0)
722 39 CALL EXEC(8,MANDO)
723 END
724 C
725 C*****
726 C
727 PROGRAM SALTD,5
728 C PERMITE LA CORRECCION DE SALTOS DE GANANCIA E INTERFERENCIAS
729 C QUE AFECTEN A UNOS POCOS PUNTOS CONSECUTIVOS ("SPIKES").
730 C PARA CORREGIR SALTOS DEBIDOS A LA INTEGRACION, VER 'SALTI'.
731 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
732 COMMON R(316),S(316),T(316),NREGI,FE
733 IF(IVIA.EQ.2) GO TO 1
734 WRITE(6,10)
735 10 FORMAT("VIA 1 ? (SI=1) _")
736 READ(1,*) IVI
737 IF(IVI.NE.1) GO TO 1
738 CALL NIVEL(BF,NP,FE)
739 1 IF(IVIA.EQ.1) GO TO 2
740 WRITE(6,11)
741 11 FORMAT("VIA 2 ? (SI=1) _")
742 READ(1,*) IVI
743 IF(IVI.NE.1) GO TO 2
744 CALL NIVEL(BF(317),NP,FE)
745 2 CALL EXEC(8,MANDO)
746 END
747 C
748 C*****
749 C
750 PROGRAM SALTI,5
751 C PERMITE CORREGIR SALTOS EN LOS BARRIDOS CONSECUENCIA DE LA
752 C INTEGRACION DE BARRIDOS DE DISTINTA LONGITUD.
753 C ASIMISMO PERMITE LA CORRECCION DE ALGUN "SPIKE" REMANENTE.
754 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
755 COMMON R(316),S(316),T(316),NREGI,FE
756 WRITE(6,6)
757 6 FORMAT("EN QUE VIA DESEAS CORREGIR LOS SALTOS ? (1,2,12) = _")
758 READ(1,*) IVI
759 K = (MOD(IVI,9)-1)*316+1
760 CALL NIVEL (R(K),NPM,FE)
761 CALL EXEC(8,MANDO)
762 END
763 C
764 C*****
765 C
766 PROGRAM MONTI,5
767 C PERMITE LA VISUALIZACION DE UN CONJUNTO DE HASTA 8 BARRIDOS
768 C CONSECUTIVOS ALMACENADOS (SIN AJUSTAR) EN EL FICHERO
769 C INTERMEDIO.
770 C SE PUEDE ESCOGER EL ANGULO DE INCLINACION DE LA VISUAL ASI
771 C COMO EL VERTICAL (POR DEFECTO TOMA ALFA = 45 Y UN
772 C ESPACIAMIENTO VERTICAL TAL QUE LA GRAFICA QUEPA EN LA
773 C PANTALLA, SALVO CASOS PARTICULARES).
774 C PERMITE GRAFICAR UN SOLO BARRIDO SI ASI SE DESEA.
775 DIMENSION IBF(1920),A(8),F(316,8)
776 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
777 COMMON R(316),S(316),T(316),NREGI,FE
778 COMMON NSEP,IPIFE(20),IPAE(20),SIG
779 COMMON IDCB(1296),IDCC(1936),IDCD(720)
780 EQUIVALENCE (IBF(1),INAM)

```

```

81 WRITE(6,6)
82 6 FORMAT("REGISTROS INICIAL Y FINAL ? : _")
783 NF = 0
84 READ(1,*) NI,NF
85 IF(NI.GE.1.AND.NF.LE.25.AND.(NF-NI).LT.8) GO TO 99
786 WRITE(6,89)
787 89 FORMAT(" NI<1 , NF>25 , NF-NI>=8 "/" )
88 GO TO 77
789 99 IF(NF.LT.NI) NF = NI
790 WRITE(6,65)
91 65 FORMAT("QUE VIA VAS A REPRESENTAR ? (1,2,12) : _")
92 READ(1,*) K
793 K = (MOD(K,9)-1)*316+1
94 DO 7 I=NI,NF
95 CALL APOSN(IDCC,IE,I)
796 CALL ERFMP(10,IE)
797 CALL READF(IDCC,IE,IBF)
98 CALL ERFMP(11,IE)
799 WRITE(6,8) I,INAM,RNAM,NPM,ICODE
800 8 FORMAT("REGISTRO "I2" FUENTE "A2,I7" NPUNT ="I4" CODIGO "
01 &I2)
02 CALL AMIN(R(K),NPM,RMIN,M)
803 L=I-NI+1
04 DO 9 J=1,NPM
05 9 F(J,L) = R(J+K-1)-RMIN
806 7 CALL AMAX(F(1,L),NPM,A(L),M)
807 CALL AMAX(A,NF-NI+1,AMX,M)
08 FE = 750./(1.+0.15*(NF-NI))
809 DO 10 I=NI,NF
810 L = I-NI+1
11 DO 10 J=1,NPM
12 10 F(J,L) = F(J,L)/AMX*FE
813 WRITE(6,75)
14 75 FORMAT("/"ANGULO DE INCLINACION , INTERVALO EN Y : _")
15 DY = 0.3*FE*(NF-NI)
816 ALFA = 45
817 READ(1,*) ALFA,DY
18 CALL PAGE(2)
819 CALL MOUNT(NPM,NF-NI+1,1000.,DY,ALFA,10,10,316,F)
820 IF(NF.EQ.NI) DY = 0.3*FE
21 LY = DY*SIN(ALFA)
22 IF(NF.EQ.NI) NF=NI+1
823 CALL AXIS(10,10,NPM/10,1000,NF-NI,LY,5,10)
24 CALL PAGE(1)
25 77 CALL EXEC(8,MANDO)
826 END
827 C
28 C*****
29 C
830 PROGRAM ESCFA,5
31 C ESCRIBE EN EL FICHERO DE DATOS AJUSTADOS EL RESULTADO DEL
32 C AJUSTE HECHO (SOLO UNA VIA).
833 C EL REGISTRO SE PUEDE ESCOGER A VOLUNTAD, PERO ES CONVENIENTE
34 C SEGUIR UN METODO: LAS DISTINTAS VIAS EN DISTINTOS GRUPOS,
35 C DENTRO DE CADA GRUPO, A MAYOR DECLINACION MAYOR NUMERO DE
836 C REGISTRO; TODO ELLO FACILITA EL USO POSTERIOR DEL
837 C PROGRAMA DE VISUALIZACION FINAL, 'VISU' .
38 C QUEDAN GRABADOS TAMBIEN TODOS LOS PARAMETROS RELATIVOS AL
39 C AJUSTE REALIZADO, ASI COMO SUAVIZADO Y COMPRESION.
840 DIMENSION IBF(704),A(316)

```



```

41 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
842 COMMON R(316),S(316),T(316),NREGI,FE
843 COMMON NSEP,IPIFE(20),IPAE(20),SIG
44 COMMON IDCB(1296),IDCC(1936),IDCD(720),NAME(3)
845 EQUIVALENCE (IBF(51),A(1)),(IBF(2),RNM),(IBF(8),SG)
846 WRITE(6,2)
47 2 FORMAT("EN QUE REGISTRO DEL FICHERO FINAL VAS A GRABAR ? (<60)"
48 " : _")
849 READ(1,*) IR
50 WRITE(6,1)
51 1 FORMAT("QUE VIA DESEAS GRABAR ? (1,2,12) : _")
852 READ(1,*) IVI
853 IF(IVI.EQ.1) NA = N1
54 IF(IVI.EQ.2) NA = N2
855 IF(IVI.EQ.12) NA = N1+N2
856 K = (MOD(IVI,9)-1)*316
57 DO 3 I=1,NPM
58 3 A(I) = R(I+K)
859 IBF(1) = INAM
60 RNM = RNAM
61 IBF(4) = NPM
862 IBF(5) = IVI
63 IBF(6) = NA
64 IBF(7) = ICODE
65 SG = SIG
866 IBF(10) = NSEP
67 DO 4 I=1,20
68 IBF(10+I) = IPIFE(I)
869 4 IBF(30+I) = IPAE(I)
70 CALL APOSN(IDCD,IE,IR)
71 CALL ERFMP(200,IE)
872 76 CALL WRITF(IDCD,IE,IBF)
873 IF(IE.EQ.-7) GO TO 77
74 CALL ERFMP(201,IE)
875 CALL EXEC(8,MANDO)
876 77 WRITE(6,78)
877 78 FORMAT("FMP ERROR -007: BAD SECURITY CODE"/
878 "CODIGO DE SEGURIDAD ? (2 LETRAS): _")
879 READ(1,79) KODE
880 79 FORMAT(A2)
881 IF(KODE.EQ.30060B) KODE=0
882 CALL OPEN(IDCD,IE,NAME,0,KODE)
883 CALL ERFMP(204,IE)
884 GO TO 76
885 END
886 C
887 C*****
888 C
889 PROGRAM LECFA,5
890 C PERMITE LA LECTURA DE TODA LA INFORMACION CONTENIDA EN
891 C CADA UNO DE LOS REGITROS DEL FICHERO FINAL.
892 C REALIZA LA GRAFICA DE LOS VALORES GRABADOS Y, OPCIONALMENTE
893 C LISTA SUS VALORES NUMERICOS.
894 DIMENSION IBF(704),A(316)
895 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
896 COMMON R(316),S(316),T(316),NREGI,FE
897 COMMON NSEP,IPIFE(20),IPAE(20),SIG
898 COMMON IDCB(1296),IDCC(1936),IDCD(720)
899 EQUIVALENCE (IBF(51),A(1)),(IBF(2),RNM),(IBF(8),SG)
900 WRITE(6,1)

```

```

901 1   FORMAT("QUE REGISTRO DESEAS LEER EN EL FICHERO FINAL ? (<=60) "
902     "": _")
903     READ(1,*) IR
904     CALL APOSN(IDCD,IE,IR)
905     CALL ERFMP(202,IE)
906     CALL READF(IDCD,IE,IBF)
907     CALL ERFMP(203,IE)
908     CALL PAGE(2)
909     WRITE(6,6) IBF(1),RNM,IBF(5)
910 6   FORMAT("FUENTE : "A2,I7"          VIA : "I2)
911     CALL GRAF(A,IBF(4),0.,400.,1000.,300.,10,Z)
912     CALL PLOT(0,0,350,0,0)
913     WRITE(6,7) IBF(4),IBF(6),IBF(7),SG
914 7   FORMAT("NPUNT = "I4"  VIAS PROM. : "I2"  CODIGO : "I2"  SIGMA = "
915     "F7.0)
916     NSEP = IBF(10)
917     IF(NSEP.NE.0) GO TO 77
918     WRITE(6,9) (IPIFE(I),I=1,10),(IPAE(I),I=1,10)
919 9   FORMAT("INTERVALOS EVITADOS EN EL AJUSTE : "/
920     &5(I3,"I3";")/"PUNTOS AISLADOS EVITADOS EN EL AJUSTE : "/
921     &10(I3,""))
922     GO TO 88
923 77  WRITE(6,10) IPIFE,IPAE (IBF(I), I = 11, 50.)
924 10  FORMAT("INTERVALOS EVITADOS EN EL AJUSTE : "/
925     &5(I3,"I3";")/5(I3,"I3" ")/
926     &"PUNTOS AISLADOS EVITADOS EN EL AJUSTE : "/
927     &10(I3,"")/10(I3,""))
928     IF(NSEP.NE.0) WRITE(6,1038) NSEP
929 1038 FORMAT("PUNTO DE SEPARACION DE LOS AJUSTES:"I4)
930 88  WRITE(6,99)
931 99  FORMAT("DESEAS LISTAR LOS VALORES NUMERICOS ? (SI=1) : _")
932     READ(1,*) IN
933     IF(IN.NE.1) GO TO 100
934     WRITE(6,98) (A(I),I=1,IBF(4))
935 98  FORMAT(/(10I7))
936 100 CALL EXEC(8,MANDO)
937     END
938 C
939 C*****
940 C
941     PROGRAM DUMPA,5
942 C   LISTA LAS CABECERAS DE LOS 60 REGISTROS CONTENIDOS EN EL
943 C   FICHERO FINAL (DE DATOS ANALIZADOS O AJUSTADOS).
944     DIMENSION IBF(704)
945     COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
946     COMMON R(316),S(316),T(316),NREGI,FE
947     COMMON NSEP,IPIFE(20),IPAE(20),SIG
948     COMMON IDCB(1296),IDCC(1936),IDCD(720)
949     EQUIVALENCE (IBF(2),RNM),(IBF(8),SG)
950     CALL RWNDF(IDCD,IE)
951     CALL ERFMP(205,IE)
952     WRITE(6,100)
953 100 FORMAT(" QUE REGISTROS ? (99= TODOS) : _")
954     NF = 60
955     READ(1,*) NI,NF
956     IF(NI.EQ.99) NI = 1
957     DO 1 I=NI,NF
958     CALL READF(IDCD,IE,IBF,704,LEN)
959     CALL ERFMP(206,IE)
960     IF(LEN.EQ.-1) GO TO 3

```

```

61 1 WRITE(6,2) I,IBF(1),RNM,IBF(5),IBF(7),SG
62 2 FORMAT("REGISTRO:"I2" FUENTE:"A2,I7" VIA:"I3" CODIGO:"I3
963 "" SIGMA ="F7.0)
64 3 CALL EXEC(8,MANDO)
65 END
966 C
967 C*****
68 C
769 PROGRAM FIN,5
970 C CIERRA LOS FICHEROS UTILIZADOS, SIN BORRARLOS, Y TERMINA
71 C LA EJECUCION DE ESTE PROGRAMA.
72 COMMON BF(632),MANDO(3),NP,IVIA,INAM,RNAM,NPM,N1,N2,ICODE
973 COMMON R(316),S(316),T(316),NREGI,FE
74 COMMON NSEP,IPIFE(20),IPAE(20),SIG
75 COMMON IDCB(1296),IDCC(1936),IDCD(720)
976 CALL CLOSE(IDCB,IE)
977 CALL ERFMP(1001,IE)
78 CALL CLOSE(IDCC,IE)
779 CALL ERFMP(1002,IE)
980 CALL CLOSE(IDCD,IE)
81 CALL ERFMP(1003,IE)
82 WRITE(6,1004)
983 1004 FORMAT(/"END PROGRAM ANA"/)
84 CALL EXEC(6)
85 END
986 $

```

```

1 FTN,L
2 C
3 C
4 C*****
5 C
6 C
7 C          PROGRAMA DE VISUALIZACION DE ZONAS DEL CIELO
8 C          YA PROMEDIADAS, AJUSTADAS Y CORREGIDAS.
9 C          PERMITE GENERAR Y SUSTRAER FUENTES PUNTUALES
10 C
11 C          ESTA VERSION PERMITE TRABAJAR CON 15 BARRIDOS
12 C          DE 316 PUNTOS, 16 BARRIDOS DE 300 PUNTOS O
13 C          CUALQUIER COMBINACION DE TAMANYO EQUIVALENTE
14 C          (4800 PUNTOS).
15 C          EL NUMERO MAXIMO DE BARRIDOS QUE ADMITE ES 20.
16 C*****
17 C
18 C          PROGRAM VISU,3
19 C          DIMENSION IPRG(3)
20 C          COMMON MANDO(3),IDCB(720),NP,NB,NI,F(4800)
21 C          COMMON IVIA,ICODE,DEC,RNAM(21),RNAMI,RNAMU,SIG,DX,DY,KONTR
22 C          COMMON DIB,DIP,AR,DE,ARP,COSD,FACHC,ARF(5),DEF(5)
23 C          PROGRAMA PRINCIPAL
24 C          KONTR = -1
25 C          MANDO(1) = 2HMA
26 C          MANDO(2) = 2HND
27 C          MANDO(3) = 2HV
28 C          IPRG(1) = 2HEN
29 C          IPRG(2) = 2HTR
30 C          IPRG(3) = 2HV
31 C          CALL EXEC(8,IPRG)
32 C          END
33 C
34 C*****
35 C
36 C          PROGRAM ENTRV,5
37 C          APERTURA DEL FICHERO A ANALIZAR
38 C          DEBE ESTAR FORMADO POR LO MENOS POR 60 REGISTROS DE
39 C          EXACTAMENTE 704 PALABRAS.
40 C          DIMENSION NAME(3)
41 C          COMMON MANDO(3),IDCB(720),NP,NB,NI,F(4800)
42 C          COMMON IVIA,ICODE,DEC,RNAM(21),RNAMI,RNAMU,SIG,DX,DY,KONTR
43 C          COMMON DIB,DIP,AR,DE,ARP,COSD,FACHC,ARF(5),DEF(5)
44 C          WRITE(6,100)
45 C          100 FORMAT(/10X"PROGRAM VISU"10X"** 83.04.08 **"///
46 C          ""NOMBRE DEL FICHERO ? :_")
47 C          READ(1,101) NAME
48 C          101 FORMAT(3A2)
49 C          CALL OPEN(IDCB,IE,NAME)
50 C          CALL ERFMP(1,IE)
51 C          WRITE(6,10)
52 C          10 FORMAT(/"LA HARD-COPY DEFORMA LA IMAGEN EN UN SENTIDO."/
53 C          ""?DESEAS DEFORMAR EN LA PANTALLA LAS IMAGENES DE MANERA "
54 C          ""QUE QUEDEN BIEN EN LA HARD-COPY ? (SI = 1) :_")
55 C          READ(1,*) IFACHC
56 C          FACHC = .894
57 C          IF(IFACHC.NE.1 ) FACHC = 1
58 C          CALL EXEC(8,MANDO)
59 C          END
60 C

```

```

61 C*****
62 C
63 PROGRAM LECFV,5
64 C LECTURA DE CONJUNTOS DE BARRIDOS CONSECUTIVOS DEL FICHERO.
65 C TRAS TAL LECTURA ES POSIBLE QUITAR UNO (O MAS) DE CUALQUIER
66 C EXTREMO, O ANYADIR UNO (O MAS) A CUALQUIER EXTREMO O CAMBIAR
67 C ALGUNO DE LOS BARRIDOS POR OTRO DEL MISMO FICHERO, NO
68 C NECESARIAMENTE ORDENADO. ESTA VERSION NO PERMITE INTERCALAR
69 C BARRIDOS.
70 C PUEDEN USARSE BARRIDOS DE DISTINTO TAMANYO. EN TAL CASO,
71 C SI EN EL EXTREMO INFERIOR HAY PRECISAMENTE UNO DE LOS MAS
72 C CORTOS, ES CONVENIENTE EMPEZAR POR ALGUNO DE LOS LARGOS
73 C Y LUEGO ANYADIR A ESTE EXTREMO INFERIOR LOS MAS CORTOS.
74 C
75 DIMENSION IBF(704),A(316),Z(6)
76 COMMON MANDO(3),IDCB(720),NP,NB,NI,F(4800)
77 COMMON IVIA,ICODE,DEC,RNAM(21),RNAMI,RNAMU,SIG,DX,DY,KONTR
78 COMMON DIB,DIP,AR,DE,ARP,COSD,FACHC,ARF(5),DEF(5)
79 EQUIVALENCE (IBF(2),RNM),(IBF(8),SG),(IBF(51),A(1))
80 NDMAX = 4800
81 NBMAX = 20
82 C
83 IF(KONTR.EQ.-1) GO TO 93
84 333 WRITE(6,957)
85 957 FORMAT("DESEAS LEER NUEVOS BARRIDOS ( 1 ) O SOLO MODIFICAR"
86 "" ALGUNO (0) ? : _")
87 READ(1,*) IC
88 IF(IC.EQ.0) GO TO 25
89 93 CALL PAGE(2)
90 NF = 316
91 KONTR = 0
92 WRITE(6,3372)
93 3372 FORMAT("ASCENSION RECTA (H,M,S) Y DECLINACION (G,', "1H"")"
94 "" PULSAR : _")
95 DO 13 I=1,6
96 13 Z(I) = 1.E30
97 READ(1,*) Z
98 17 DO 14 I=1,6
99 IF(Z(I).EQ.1.E30) GO TO 15
100 14 CONTINUE
101 GO TO 18
102 15 WRITE(6,16)
103 16 FORMAT("? _")
104 READ(1,*) (Z(J),J=I,6)
105 GO TO 17
106 18 ARP = (Z(3)/60+Z(2))/60+Z(1)
107 DEC = (Z(6)/60+Z(5))/60+Z(4)
108 COSD = FACHC*COS(DEC/57.29578)
109 DO 325 I=1,5
110 325 ARF(I) = 1.E30
111 WRITE(6,200)
112 200 FORMAT("PUNTOS INICIAL Y FINAL A REPRESENTAR DE CADA UNO"
113 "" DE LOS BARRIDOS ? _")
114 READ(1,*) NI,NF
115 NP = NF-NI+1
116 999 WRITE(6,300)
117 300 FORMAT("REGISTROS INICIAL Y FINAL ? : _")
118 READ(1,*) IBI,IBU
119 IF( (IBI.LE.0.OR.IBU.GT.60.OR.IBI.GT.IBU) GO TO 999
120 NB = IBU-IBI+1

```

```

121 IF(NP*NB.GT.NDMAX) GO TO 9999
122 IF(NB.GT.20) GO TO 9998
123 WRITE(6,100)
124 100 FORMAT("REGISTRO FUENTE"5X"VIA PUNTOS CODIGO SIGMA")
125 NANT = NP
126 SIG = 0
127 DO 10 I=IBI,IBU
128 CALL APOSN(IDCB,IE,I)
129 CALL ERFMP(10+I,IE)
130 CALL READF(IDCB,IE,IBF)
131 CALL ERFMP(10+I,IE)
132 RNAM(I-IBI+1) = RNM
133 SIG = SIG + SG
134 N = IBF(4)
135 WRITE(6,303) I,IBF(1),RNM,IBF(5),N,IBF(7),SG
136 303 FORMAT(I5,5X,A2,I7,I6,I8,I10,F9.0)
137 KORR = 0
138 IF(N.GE.NP.AND.N.GE.NANT) GO TO 129
139 WRITE(6,130)
140 130 FORMAT("ESTE BARRIDO ES MAS CORTO QUE LOS DEMAS."/
141 "CORRIMIENTO RESPECTO DE LOS DE REFERENCIA (EN NUMERO DE PUNTOS)/
142 "EXPRESANDO LA DIFERENCIA DE ASCENSIONES RECTAS FINALES) :_")
143 READ(1,*) KORR
144 129 J = I-IBI
145 M = MIN0(N,NP-KORR)
146 DO 1 K=1,M
147 L = J*NP+K+KORR
148 1 F(L) = A(K+NI-1)
149 IF(M.GE.NP) GO TO 10
150 M1 = M+1+KORR
151 IF(M1.GT.NP) GO TO 291
152 DO 2 K=M1,NP
153 L = J*NP+K
154 2 F(L) = 0
155 291 IF(KORR.LE.0) GO TO 10
156 DO 292 K=1,KORR
157 L=J*NP+K
158 292 F(L) = 0
159 10 NANT = N
160 SIG = SIG/NB
161 WRITE(6,1221) SIG
162 1221 FORMAT("/"SIGMA MEDIA = "F7.0/)
163 IVIA = IBF(5)
164 ICODE = IBF(7)
165 25 RNAMI = RNAM(1)
166 RNAMU = RNAM(NB)
167 NDIM = NP*NB
168 CALL AMAX(F,NDIM,AMX,LL)
169 CALL AMIN(F,NDIM,AMN,LL)
170 ALT = AMX-AMN
171 FE = 500./ALT/(1.+0.15*(NB-1))
172 DV = 0.3*FE*ALT*(NB-1)
173 ALFA = 45
174 DO 20 I=1,NDIM
175 20 F(I) = F(I)*FE
176 CALL PAGE(3)
177 CALL PAGE(1)
178 WRITE(6,1173) RNAMI,RNAMU
179 1173 FORMAT("DESDE "I7" HASTA "I7)
180 CALL MOUNT(NP,NB,1000.,DV,ALFA,10,220,NP,F)

```

```

181 IF(NB.EQ.1) DV = 0.3*FE
182 LY = DV*SIN(ALFA)
83 CALL AXIS(10,220,NP/10,1000,NB-1,LY,5,10)
84 FEI = 1./FE
185 DO 29 I=1,NDIM
86 29 F(I) = F(I)*FEI
87 CALL PLOT(0,0,133,0,0)
188 WRITE(6,400)
189 400 FORMAT("DESEAS CAMBIAR ( 1 ), ANYADIR ( 2 ) O QUITAR ( 3 )"
90 "ALGUN BARRIDO ? (NO=0) : _")
191 READ(1,*) I123
192 IF(I123-1) 99,30,40
93 30 WRITE(6,401)
94 401 FORMAT("QUE BARRIDO DESEAS CAMBIAR ? : _")
195 READ(1,*) IB
96 IF(IB.EQ.0) GO TO 99
97 IF(IB.LT.1.OR.IB.GT.NB) GO TO 30
198 WRITE(6,402)
199 402 FORMAT("QUE REGISTRO DESEAS LEER ? : _")
200 READ(1,*) IG
201 IF(IG.EQ.0) GO TO 99
202 CALL APOSN(IDC B,IE,IG)
203 CALL ERFMP(100,IE)
204 CALL READF(IDC B,IE,IBF)
205 CALL ERFMP(100,IE)
206 N = IBF(4)
207 WRITE(6,100)
208 WRITE(6,303) I,IBF(1),RNM,IBF(5),N,IBF(7),SG
209 RNAM(IB) = RNM
210 IF(IB.EQ.1) RNAMI = RNM
211 IF(IB.EQ.NB) RNAMU = RNM
212 KORR = 0
213 IF(N.GE.NP.AND.N.GE.NANT) GO TO 153
214 WRITE(6,130)
215 READ(1,*) KORR
216 153 M = MIN0(N,NP-KORR)
217 DO 31 K=1,M
218 L = (IB-1)*NP+K+KORR
219 31 F(L) = A(K+NI-1)
220 NANT = N
221 IF(M.GE.NP) GO TO 25
222 M1 = M+1+KORR
223 IF(M1.GT.NP) GO TO 294
224 DO 32 K=M1,NP
225 L = (IB-1)*NP+K
226 32 F(L) = 0
227 294 IF(KORR.LE.0) GO TO 25
228 DO 295 K=1,KORR
229 L = (IB-1)*NP+K
230 295 F(L) = 0
231 GO TO 25
232 40 IF(I123.EQ.3) GO TO 111
233 IF(NDIM+NP.GT.NDMAX) GO TO 9999
234 WRITE(6,500)
235 500 FORMAT("DESEAS ANYADIRLO AL PRINCIPIO ( -1 ) O AL FINAL ( 1 )"
236 " ? _")
237 READ(1,*) IC
238 IF(IC) 41,99,406
239 406 DO 48 I=NDIM+1,NDIM+NP
240 48 F(I) = 0

```

```

241      GO TO 42
242 41    DO 43 I=NDIM,1,-1
243 43    F(NP+I) = F(I)
244      DO 45 I=1,NP
245 45    F(I) = 0
246      K = 0
247 42    IF(IC.GT.0) K = NDIM
248      WRITE(6,501)
249 501   FORMAT("QUE REGISTRO DEL FICHERO DESEAS LEER ? I_")
250      READ(1,*) IG
251      IF(IG.LE.0) GO TO 99
252      IF(IG.GT.60) GO TO 42
253      CALL APOSN(IDCIB,IE,IG)
254      CALL ERFMP(101,IE)
255      CALL READF(IDCIB,IE,IBF)
256      CALL ERFMP(101,IE)
257      N = IBF(4)
258      KORR = 0
259      WRITE(6,100)
260      WRITE(6,303) I,IBF(1),RNM,IBF(5),N,IBF(7),SG
261      IF(N.GE.NP.AND.N.GE.NANT) GO TO 155
262      WRITE(6,130)
263      READ(1,*) KORR
264 155   M = MIN0(N,NP-KORR)
265      DO 51 J=1,M
266      L = J+K+KORR
267 51    F(L) = A(J+NI-1)
268      NB = NB+1
269      IF(NB.GT.NBMAX) GO TO 9998
270      NANT = N
271      RNAM(NB) = RNM
272      IF(IC.GT.0) GO TO 25
273      DO 1171 I=NB,2,-1
274 1171  RNAM(I) = RNAM(I-1)
275      RNAM(1) = RNM
276      GO TO 25
277 111   WRITE(6,112)
278 112   FORMAT("DESEAS QUITARLO DEL PRINCIPIO ( -1 ) O DEL FINAL"
279      " ( 1 ) ? : ")
280      READ(1,*) IC
281      IF(IC.EQ.0) GO TO 99
282      NB = NB-1
283      IF(IC.GT.0) GO TO 25
284      DO 114 I=1,NDIM-NP
285 114   F(I) = F(I+NP)
286      DO 1172 I=1,NB
287 1172  RNAM(I) = RNAM(I+1)
288      GO TO 25
289 99    CALL EXEC(8,MANDO)
290 9999  WRITE(6,900) NDIM,NDMAX
291 900   FORMAT("/" CAPACIDAD INSUFICIENTE: DISMINUIR NUMERO DE BARRIDOS"
292      " O DE PUNTOS"/"DIM ="I7"   DIM MAX = "I7/)
293      GO TO 93
294 9998  WRITE(6,980) NB
295 980   FORMAT("NUMERO DE BARRIDOS MAYOR QUE "I3)
296      GO TO 333
297      END
298 C
299 C*****
300 C

```



```

301 PROGRAM GRAFV,5
302 DIMENSION Z(6),IGRAF(6)
303 COMMON MANDO(3),IDCB(720),NP,NB,NI,F(4800)
304 COMMON IVIA,ICODE,DEC,RNAM(21),RNAMI,RNAMU,SIG,DX,DY,KONTR
305 COMMON DIB,DIP,AR,DE,ARP,COSD,FACHC,ARF(5),DEF(5)
306 DATA IGRAF/2HMO,2HNT,2HV,2HIS,2HOF,2HV /
307 WRITE(6,6) NB,NP,NI
308 6 FORMAT("SE VA A REPRESENTAR "I2" BARRIDOS DE "I3
309 "" PUNTOS."/ "EL PRIMER PUNTO A SER REPRESENTADO ES EL"
310 "I3" DEL BARRIDO REAL.")
311 WRITE(6,130)
312 130 FORMAT("ANCHURA DE LA GRAFICA EN PUNTOS DE PANTALLA (SIN COSENO)"
313 "" ? (<1023):_")
314 READ(1,*) RIX
315 DX = RIX*COSD
316 WRITE(6,416)
317 416 FORMAT("DISTANCIA ENTRE BARRIDOS (MIN.ARC.) Y ENTRE PUNTOS"
318 "" (SEG.TIEMPO):_")
319 READ(1,*) DIB,DIPP
320 DIP = DIPP/60*COSD
321 DZ = DIB*(NB-1)/(15*DIP*(NP-1))
322 IF(RIX.GT.0.0) GO TO 33
323 IF(DZ.LE.0.74) DX=1023
324 IF(DZ.GT.0.74) DX=757./DZ
325 33 DY = DX*DZ
326 WRITE(6,329)
327 329 FORMAT("COORDENADAS DEL PUNTO (1,1) :"/
328 ""ASCENSION RECTA (H,M,S) Y DECLINACION (G,', "1H"") :_")
329 DO 13 I=1,6
330 13 Z(I) = 1E30
331 READ(1,*) Z
332 17 DO 14 I=1,6
333 IF(Z(I).EQ.1.E30) GO TO 15
334 14 CONTINUE
335 GO TO 18
336 15 WRITE(6,16)
337 16 FORMAT("?_")
338 READ(1,*) (Z(J),J=I,6)
339 GO TO 17
340 18 AR = ((Z(3)/60+Z(2))/60+Z(1))*60
341 DE = ((Z(6)/60+Z(5))/60+Z(4))*60
342 IF(KONTR-2) 21,22,23
343 21 KONTR = 1
344 CALL EXEC(8,MANDO)
345 22 KONTR = 1
346 CALL EXEC(8,IGRAF)
347 23 KONTR = 1
348 CALL EXEC(8,IGRAF(4))
349 END
350 C
351 C*****
352 C
353 PROGRAM MONTV,5
354 DIMENSION IGRAF(3)
355 COMMON MANDO(3),IDCB(720),NP,NB,NI,F(4800)
356 COMMON IVIA,ICODE,DEC,RNAM(21),RNAMI,RNAMU,SIG,DX,DY,KONTR
357 COMMON DIB,DIP,AR,DE,ARP,COSD
358 DATA IGRAF/2HGR,2HAF,2HV /
359 WRITE(6,5) RNAMI,RNAMU,IVIA,ICODE,DEC
360 5 FORMAT("DESDE "I7" HASTA "I7/"VIA "I2" CODIGO"I3" DECLI"

```

```

361      "NACION "F5.1)
362      WRITE(6,6) NB,NP
363 6      FORMAT(I3" BARRIDOS DE "I3" PUNTOS")
364      IF(KONTR.EQ.0) GO TO 300
365      NDIM = NP*NB
366      CALL AMAX(F,NDIM,FMAX,LL)
367      DO 10 I=1,NDIM
368 10     F(I) = F(I)*0.0001
369      CALL AMIN(F,NP,FMIN,LL)
370      FEZP = 1
371      WRITE(6,160)
372 160    FORMAT("INCLINACION ? (0<ALFA<90) : _")
373      READ(1,*) ALFA
374 150    WRITE(6,161) FMAX
375 161    FORMAT("VALOR MAXIMO:"I9/"FACTOR DE ESCALA (PTOS/K) ? (NO = 0)"
376      " : _")
377      READ(1,*) FEZ
378      IF(FEZ.EQ.0) GO TO 200
379      FF = FEZ/FEZP
380      DO 165 I=1,NDIM
381 165    F(I) = F(I)*FF
382      FMIN = FMIN*FF
383      IMIN = ABS(FMIN)
384      FEZP = FEZ
385      CALL PAGE(2)
386      CALL MOUNT(NP,NB,DX,DY,ALFA,0,IMIN,NP,F)
387      LY = DY*SIN(ALFA)
388      ALFA = ALFA*57.29578
389      IX = DX
390      CALL AXIS(0,IMIN,NP/10,IX,NB-1,LY,5,5)
391      IF(FEZP+LY.GT.750.) GO TO 33
392      CALL PLOT(0,0,754,0,0)
393      WRITE(6,66) RNAMI,RNAMU,IVIA,ICODE,SIG
394 66     FORMAT(I7" A "I7" VIA"I3" CODIGO "I3" SIG.APROX.="F5.0)
395 33     CALL PAGE(1)
396      GO TO 150
397 200    FEZI = 10000./FEZP
398      DO 225 I=1,NDIM
399 225    F(I) = F(I)*FEZI
400      CALL EXEC(8,MANDO)
401 300    KONTR = 2
402      CALL EXEC(8,IGRAF)
403      END
404  C
405  C*****
406  C
407      PROGRAM ISOFV,5
408      DIMENSION IGRAF(3)
409      COMMON MANDO(3),IDCB(720),NP,NB,NI,F(4800)
410      COMMON IVIA,ICODE,DEC,RNAM(21),RNAMI,RNAMU,SIG,DX,DY,KONTR
411      COMMON DIB,DIP,AR,DE,ARP,COSD,FACHC,ARF(5),DEF(5)
412      DATA IGRAF/2HGR,2HAF,2HV /
413      IF(NB.LE.1) GO TO 999
414      IF(KONTR.EQ.0) GO TO 300
415      NDIM = NP*NB
416      CALL AMAX(F,NDIM,FMAX,LL)
417      WRITE(6,2) FMAX
418 2      FORMAT("VALOR MAXIMO "I12)
419      FI = 1./FMAX
420      DO 10 I=1,NDIM

```

```

421 10 F(I) = F(I)*FI
422 WRITE(6,251)
423 251 FORMAT("VALOR DE LA PRIMERA ISOFOTA, DISTANCIA ENTRE ELLAS "
424 1": -")
425 LIN = 0
426 READ(1,*) VMF, DV, LIN
427 IF(LIN.EQ.0) LIN = (1.-VMF)/DV+1.
428 NXI = INT(AR)
429 NXF = INT(AR-(NP-1)*DIP/COSD)+1
430 IF(DE.EQ.0.0) DE = 0.00001
431 NYI = INT(DE/10)+(1+ABS(DE)/DE)/2.
432 NYF = INT((DE+(NB-1)*DIB)/10)
433 IDX = DX
434 IDY = DY
435 CALL PAGE(2)
436 IF(IDY.GT.757) GO TO 67
437 WRITE(6,66) RNAMI, RNAMU, IVIA, ICODE, SIG
438 66 FORMAT(I7" A "I7" VIA" I3" CODIGO " I3" SIG.APROX.="F5.0)
439 67 CALL PLOT(1,0,1,IDX,1)
440 CALL PLOT(1,IDX,1,IDX,IDY)
441 CALL PLOT(1,IDX,IDY,0,IDY)
442 CALL PLOT(1,0,IDY,0,1)
443 FAR = DX/(NP-1)/DIP*COSD
444 DO 3 I=NXF, NXI
445 J = (AR-I)*FAR
446 CALL PLOT(1,J,1,J,10)
447 3 CALL PLOT(1,J, IDY, J, IDY-10)
448 FDE = DY/(NB-1)/DIB
449 DO 4 I=NYI, NYF
450 J = (10*I-DE)*FDE+1
451 CALL PLOT(1,0,J,10,J)
452 4 CALL PLOT(1,IDX,J, IDX-10, J)
453 C
454 NX = (AR-ARP*60)*FAR-4
455 NY = (DEC*60.-DE)*FDE-5
456 IF(NX.GT.1023.OR.NX.LT.0.OR.NY.GT.780.OR.NY.LT.0) GO TO 745
457 CALL PLOT(0,NX,NY,0,0)
458 WRITE(6,6)
459 6 FORMAT("+ ")
460 745 DO 332 I=1,5
461 IF(ARF(I).EQ.1.E30) GO TO 334
462 NX = (AR-ARF(I))*FAR-4
463 NY = (DEF(I)-DE)*FDE-5
464 IF(NX.GT.1023.OR.NX.LT.0.OR.NY.GT.780.OR.NY.LT.0) GO TO 332
465 CALL PLOT(0,NX,NY,0,0)
466 WRITE(6,6)
467 332 CONTINUE
468 C
469 334 CALL ISOF(NP, NB, DX, DY, VMF, DV, LIN, 0, 1, NP, F)
470 DO 110 I=1, NDIM
471 110 F(I) = F(I)*FMAX
472 CALL PAGE(3)
473 CALL PAGE(1)
474 299 CALL EXEC(8, MANDO)
475 300 KONTR = 3
476 CALL EXEC(8, IGRAF)
477 999 WRITE(6,133)
478 133 FORMAT(" NB = 1 : REALIZACION DEL MAPA IMPOSIBLE ")
479 GO TO 299
480 END

```

```

481 C
482 C*****
483 C
484     PROGRAM ZOOMV,5
485 C
486 C PERMITE VISUALIZAR UNA PARTE DEL MAPA MEDIANTE UNA
487 C LLAMADA A 'ISOFV'.
488 C EL RESTO SE PIERDE: EN CASO DE QUERER REPRESENTAR
489 C DE NUEVO TODO EL MAPA HAY QUE VOLVER A LEER LOS DATOS.
490 C MANTIENE EL MISMO TAMANYO HORIZONTAL EN PUNTOS DE
491 C PANTALLA. PARA CAMBIAR ESTE VALOR, LLAMAR 'GRAFV'.
492 C
493     DIMENSION IPROG(3)
494     COMMON MANDO(3),IDCB(720),NP,NB,NI,F(4800)
495     COMMON IVIA,ICODE,DEC,RNAM(21),RNAMI,RNAMU,SIG,DX,DY,KONTR
496     COMMON DIB,DIP,AR,DE,ARP,COSD
497     DATA IPROG/2HIS,2HOF,2HV /
498 C
499     WRITE(6,10)
500 10  FORMAT("PUNTO INFERIOR IZQUIERDO  X,Y =_")
501     READ(1,*) IX,IY
502     WRITE(6,11)
503 11  FORMAT("PUNTO SUPERIOR DERECHO  X,Y =_")
504     READ(1,*) JX,JY
505     IF(IX.LE.0.OR.IX.GT.JX.OR.JX.GT.NP
506     ..OR.IY.LE.0.OR.IY.GT.JY.OR.JY.GT.NB) GO TO 99
507     K = 0
508     DO 1 I=IY,JY
509     DO 1 J=IX,JX
510     K = K+1
511     L = (I-1)*NP+J
512 1   F(K) = F(L)
513     DO 2 I=IY,JY
514     K = I-IY+1
515 2   RNAM(K) = RNAM(I)
516     AR = AR-DIP*(IX-1)/COSD
517     DE = DE+DIB*(IY-1)
518     NI = NI+IX-1
519     NB = JY-IY+1
520     NP = JX-IX+1
521     DZ = DIB*(NB-1)/(15.*DIP*NP)
522     IF(DZ.LE.0.74) DX = 1023
523     IF(DZ.GT.0.74) DX = 757/DZ
524     DY = DX*DZ
525     RNAMI = RNAM(1)
526     RNAMU = RNAM(NB)
527     KONTR = 1
528     CALL EXEC(8,IPROG)
529 99  WRITE(6,999)
530 999 FORMAT("PARAMETROS NO VALIDOS")
531     CALL EXEC(8,MANDO)
532     END
533 C
534 C*****
535 C
536     PROGRAM PUNTV,5
537     DIMENSION Z(6),AMP(5)
538     COMMON MANDO(3),IDCB(720),NP,NB,NI,F(4800)
539     COMMON IVIA,ICODE,DEC,RNAM(21),RNAMI,RNAMU,SIG,DX,DY,KONTR
540     COMMON DIB,DIP,AR,DE,ARP,COSD,FACHC,ARF(5),DEF(5)

```

```

541 C
542 C PERMITE SUSTRAR O ANYADIR FUENTES PUNTUALES OBSERVADAS
543 C MEDIANTE UN HAZ GAUSIANO A DEFINIR.
544 C PARA SUSTRAR, LA AMPLITUD DEBE SER POSITIVA
545 C PARA ANYADIR, LA AMPLITUD DEBE SER NEGATIVA.
546 C
547 73 WRITE(6,9)
548 9 FORMAT("NUMERO DE FUENTES PUNTUALES ? (0( <5) : _")
549 READ(1,*) NFP
550 IF(NFP.LT.0.OR.NFP.GT.5) GO TO 73
551 IF(NFP.EQ.0) GO TO 79
552 DO 1 I=1,NFP
553 WRITE(6,10) I
554 10 FORMAT("FUENTE"13": AR(H,M,S), DEC(G,', "1H"") = _")
555 DO 13 K=1,6
556 13 Z(K) = 1E30
557 READ(1,*) Z
558 17 DO 14 K=1,6
559 IF(Z(K).EQ.1.E30) GO TO 15
560 14 CONTINUE
561 GO TO 18
562 15 WRITE(6,16)
563 16 FORMAT("?_")
564 READ(1,*) (Z(J),J=K,6)
565 GO TO 17
566 18 ARF(I) = ((Z(3)/60+Z(2))/60+Z(1))*60
567 DEF(I) = ((Z(6)/60+Z(5))/60+Z(4))*60
568 WRITE(6,21)
569 21 FORMAT("AMPLITUD (K) = _")
570 1 READ(1,*) AMP(I)
571 WRITE(6,20)
572 20 FORMAT("/"PARAMETROS DEL HAZ:"/"HPBWA ('), HPBWB (' ) = _")
573 READ(1,*) HPBWA,HPBWB
574 SGX = HPBWA/2.354/15.
575 SGY = HPBWB/2.354
576 C
577 DO 100 K=1,NFP
578 AMP(K) = AMP(K)*10000.
579 DO 90 I=1,NB
580 ZY = ABS(DEF(K)-DE-FLOAT(I-1)*DIB)/SGY
581 IF(ZY-3.) 30,30,90
582 30 DO 80 J=1,NP
583 X = AR-FLOAT(J-1)*DIP/COSD
584 ZX = ABS((X-ARF(K))*COSD)/SGX
585 IF(ZX-3.) 40,40,80
586 40 W = (ZX*ZX+ZY*ZY)/2.
587 W = AMP(K)*EXP(-W)
588 L = (I-1)*NP+J
589 F(L) = F(L)-W
590 80 CONTINUE
591 90 CONTINUE
592 100 CONTINUE
593 CALL EXEC(8,MANDO)
594 79 DO 81 I=1,5
595 81 ARF(I) = 1.E30
596 CALL EXEC(8,MANDO)
597 END
598 C
599 C*****
600 C$

```

```

601      PROGRAM MANDV,5
602 C    LEE LOS COMANDOS Y COMPRUEBA SI SON ADECUADOS.
603      DIMENSION MAN(3),IVERF(24)
604      DATA IVERF/2HLE,2HCF,2HV ,2HDU,2HMP,2HV ,2HFI,2HNV,2H ,
605      ,2HMO,2HNT,2HV ,2HIS,2HOF,2HV ,2HGR,2HAF,2HV ,2HZO,2HOM,2HV ,
606      ,2HPU,2HNT,2HV /
607      NVERF = 24
608 3    CALL PAGE(3)
609      WRITE(6,6)
610 6    FORMAT("V)_")
611      READ(1,13) MAN
612 13   FORMAT(3A2)
613      DO 1 I=1,NVERF,3
614      IF(MAN(1).EQ.IVERF(I).AND.MAN(2).EQ.IVERF(I+1).AND.
615      .MAN(3).EQ.IVERF(I+2)) GO TO 2
616 1    CONTINUE
617      WRITE(6,10) MAN
618 10   FORMAT("COMANDO '3A2' NO VALIDO"/)
619      GO TO 3
620 2    CALL EXEC(8,MAN)
621      END
622 C
623 C*****
624 C
625 C
626      PROGRAM DUMPV,5
627 C    PERMITE EL LISTADO PARCIAL O TOTAL DEL FICHERO QUE SE
628 C    ESTA VISUALIZANDO.
629      DIMENSION IBF(704)
630      COMMON MANDO(3),IDCB(720),NP,NB,NI,F(4800)
631      EQUIVALENCE (IBF(2),RNAM),(IBF(8),SIG)
632      CALL RWNDF(IDCB,IE)
633      CALL ERFMP(201,IE)
634      WRITE(6,6)
635 6    FORMAT("QUE REGISTROS ? (99= TODOS) : _")
636      NF = 0
637      READ(1,*) NI,NF
638      IF(NI.EQ.99) NF=60
639      IF(NI.EQ.99) NI = 1
640      IF(NF.LT.NI) NF = NI
641      DO 1 I=NI,NF
642      CALL READF(IDCB,IE,IBF,704,LEN)
643      CALL ERFMP(202,IE)
644      IF(LEN.EQ.-1) GO TO 7
645 1    WRITE(6,2) I,IBF(1),RNAM,IBF(5),IBF(7),SIG
646 2    FORMAT("REGISTRO:"I2" FUENTE:"A2,I7" VIA:"I3" CODIGO:"I3
647      " SIGMA ="F7.0)
648 7    CALL EXEC(8,MANDO)
649      END
650 C*****
651 C
652      PROGRAM FINV,5
653 C    CIERRA EL FICHERO Y TERMINA EL PROGRAMA.
654      COMMON MANDV(3),IDCB(720)
655      CALL CLOSE(IDCB)
656      WRITE(6,6)
657 6    FORMAT("/" END PROGRAM VISU "/)
658      CALL EXEC(6)
659      END
660 $

```

1 FTN,L

2 C

3 C

4 C

5 C

6 C

7 C

8 C

9 C

10 C

11 C

12 C

13 C\*\*\*\*\*

14 C

15 SUBROUTINE AMAX (A,N,VMAX,NPOS)

16 C CALCULO DEL VALOR MAXIMO VMAX DE UN VECTOR

17 C A DE DIMENSION N ASI COMO SU POSICION NPOS :

18 C A(NPOS) = VMAX

19 DIMENSION A(1)

20 VMAX=A(1)

21 NPOS=1

22 DO 1 I=2,N

23 IF (VMAX-A(I)) 2,1

24 2 VMAX=A(I)

25 NPOS=I

26 1 CONTINUE

27 RETURN

28 END

29 C

30 C\*\*\*\*\*

31 C

32 SUBROUTINE AMIN (A,N,VMIN,NPOS)

33 C CALCULO DEL VALOR MINIMO VMIN DE UN VECTOR

34 C A DE DIMENSION N ASI COMO SU POSICION NPOS :

35 C A(NPOS) = VMIN

36 DIMENSION A(1)

37 VMIN=A(1)

38 NPOS=1

39 DO 1 I=2,N

40 IF (A(I)-VMIN) 2,1

41 2 VMIN=A(I)

42 NPOS=I

43 1 CONTINUE

44 RETURN

45 END

46 C

47 C\*\*\*\*\*

48 C

49 SUBROUTINE GRAF (Y,NP,X0,Y0,DX,DY,IBR,AMPL)

50 DIMENSION Y(1)

51 C

52 C ESTA SUBROUTINE SIRVE PARA DIBUJAR EN LA PANTALLA (TEKTRONIX)

53 C LA GRAFICA CORRESPONDIENTE A LOS VALORES DE LA VARIABLE DIMEN-

54 C SIONADA (Y).

55 C AJUSTA LA ESCALA AL TAMAGNO DE LA GRAFICA, DIBUJA BARRAS Y LA

56 C LINEA CERO SI CAE DENTRO.

57 C EL SIGNIFICADO DE LOS PARAMETROS ES EL SIGUIENTE:

58 C Y=NOMBRE DE LA VARIABLE DIMENSIONADA A REPRESENTAR.

59 C NP=NUMERO DE PUNTOS DE LA VARIABLE.

60 C X0, Y0=COORDENADAS DEL PUNTO INFERIOR IZQUIERDA DE LA GRAFI-

```

61 C CA EN UNIDADES DE PANTALLA.
62 C DX, DY=ANCHO Y ALTO DE LA GRAFICA EN UNIDADES DE PANTALLA.
63 C IBR=DIVISOR COMUN DE LAS BARRAS A DIBUJAR.
64 C AMPL=YMAX-YMIN. ESTE VALOR ES DEVUELTO POR LA SUBROUTINE Y
65 C ES LA AMPLITUD MAXIMA DE LA GRAFICA EN LAS MISMAS UNIDADES
66 C QUE <Y>.
67 C PUEDEN DIBUJARSE VARIAS GRAFICAS SUPERPUESTAS.
68 C ATENCION: EL CURSOR QUEDA SITUADO EN EL ULTIMO TRAZO DIBUJADO.
69 C

```

```

70 CALL AMAX (Y, NP, YMAX, I)
71 CALL AMIN (Y, NP, YMIN, I)
72 AMPL=YMAX-YMIN
73 FE=DY/AMPL
74 N=NP-1
75 AX=DX/FLOAT(N)
76 DO 20 I=1, N
77 IXI=FLOAT(I-1)*AX+X0+.5
78 IXF=FLOAT(I)*AX+X0+.5
79 IYI=(Y(I)-YMIN)*FE+Y0+.5
80 IYF=(Y(I+1)-YMIN)*FE+Y0+.5
81 CALL PLOT (1, IXI, IYI, IXF, IYF)
82 IF (MOD(I+1, IBR)) 20, 21, 20
83 21 IY0=Y0
84 IYF0=DY+Y0
85 CALL PLOT (1, IXF, IY0, IXF, IYF0)
86 20 CONTINUE
87 IF (YMAX) 50, 30, 30
88 30 IF (YMIN) 31, 31, 50
89 31 IXI=X0
90 IXF=X0+DX
91 IYI=-YMIN*FE+Y0+.5
92 CALL PLOT (1, IXI, IYI, IXF, IYI)
93 50 RETURN
94 END

```

```

95 C
96 C*****
97 C

```

```

98 SUBROUTINE GRAFE (Y, NP, X0, Y0, DX, DY, IBR, FE, OFFPAN) ,25-10-82
99 C
100 DIMENSION Y(1)

```

```

101 C
102 C ESTA SUBROUTINA DIBUJA EN LA PANTALLA LA GRAFICA CORRESPONDIENTE
103 C A LOS VALORES DE LA VARIABLE DIMENSIONADA <Y>. SE DIFERENCIA DE
104 C <GRAF> EN QUE EL FACTOR DE ESCALA <FE> ES PARAMETRO DE ENTRADA
105 C Y DAMOS TAMBIEN UN OFFSET DE PANTALLA <OFFPAN> PARA CONSEGUIR LA
106 C COLOCACION DESEADA. NO DIBUJA LAS LINEAS QUE SE SALGAN DEL RE-
107 C CUADRO LIMITADO POR <X0, Y0, DX, DY>.
108 C

```

```

109 N=NP-1
110 AX=DX/N
111 IY0=Y0
112 IYF0=Y0+DY
113 IYOF=Y0+OFFPAN
114 DO 100 I=1, N
115 IXI=(I-1)*AX+X0
116 IXF=I*AX+X0
117 IYI=Y(I)*FE+IYOF
118 IYF=Y(I+1)*FE+IYOF
119 AY=IYF-IYI
120 IF (IYI-IY0) 13, 10

```



```

121 10 IF (IYF0-IYI) 12,11
122 11 IF (IYF-IY0) 7,14
123 14 IF (IYF0-IYF) 4,1
124 12 IF (IYF-IY0) 8,15
125 15 IF (IYF0-IYF) 30,5
126 13 IF (IYF-IY0) 30,16
127 16 IF (IYF0-IYF) 9,6
128 4 NXF=AX*(IYF0-IYI)/AY+IXI
129 CALL PLOT (1,IXI,IYI,NXF,IYF0)
130 GO TO 30
131 7 NXF=AX*(IY0-IYI)/AY+IXI
132 CALL PLOT (1,IXI,IYI,NXF,IY0)
133 GO TO 30
134 1 CALL PLOT (1,IXI,IYI,IXF,IYF)
135 GO TO 30
136 5 NXI=AX*(IYF0-IYI)/AY+IXI
137 CALL PLOT (1,NXI,IYF0,IXF,IYF)
138 GO TO 30
139 8 NXI=AX*(IYF0-IYI)/AY+IXI
140 NXF=AX*(IY0-IYI)/AY+IXI
141 CALL PLOT (1,NXI,IYF0,NXF,IY0)
142 GO TO 30
143 6 NXI=AX*(IY0-IYI)/AY+IXI
144 CALL PLOT (1,NXI,IY0,IXF,IYF)
145 GO TO 30
146 9 NXI=AX*(IY0-IYI)/AY+IXI
147 NXF=AX*(IYF0-IYI)/AY+IXI
148 CALL PLOT (1,NXI,IY0,NXF,IYF0)
149 30 IF (MOD(I+1,IBR)) 100,22,100
150 22 CALL PLOT (1,IXF,IY0,IXF,IYF0)
151 100 CONTINUE
152 IF (IYOF.GT.IYF0.OR.IYOF.LT.IY0) RETURN
153 IXI=X0
154 CALL PLOT (1,IXI,IYOF,IXF,IYOF)
155 RETURN
156 END

```

```

157 C
158 C*****
159 C

```

```

160 SUBROUTINE NUMBR (X0,Y0,DX,NP,IBR) ,16-8-82

```

```

161 C
162 C NUMERACION DE LAS BARRAS DIBUJADAS
163 C CON GRAF. PARA CONSEGUIR LA POSICION
164 C DESEADA PODEMOS JUGAR MODIFICANDO X0
165 C E Y0. EL SIGNIFICADO DE LOS PARAMETROS
166 C ES EL MISMO QUE EN GRAF.
167 C

```

```

168 IXF=0
169 F=DX/(NP-1)
170 IY=Y0
171 I=IBR
172 20 IF (I.GT.NP) RETURN
173 LOG=ALOGT(FLOAT(I)+.5)+1
174 IX=X0+(I-1)*F-LOG*14+9
175 IF (IX.LE.IXF) GO TO 10
176 IXF=IX+LOG*14+4
177 CALL PLOT (0,IX,IY)
178 GO TO (1,2,3,4),LOG
179 1 WRITE (6,101) I
180 101 FORMAT (I1)

```

```

81      GO TO 10
82  2    WRITE (6,102) I
183 102  FORMAT (I2)
84      GO TO 10
85  3    WRITE (6,103) I
186 103  FORMAT (I3)
87      GO TO 10
88  4    WRITE (6,104) I
189 104  FORMAT (I4)
190 10    I=I+IBR
91      GO TO 20
92      END
193  C
94  C*****
95  C
196      SUBROUTINE PLOT (K,IXI,IYI,IXF,IYF)          ,25-8-82
197  C
98      DIMENSION ICAR(5)
199  C
200  C SI K#0 DIBUJA LA LINEA DADA POR IXI, IYI, IXF, IYF.
01  C
02  C SI K=0 POSICIONA EL CURSOR EN EL PUNTO IXI,IYI.
203  C
04      ICAR(1)=IOR(7424,32+IYI/32)
05      ICAR(2)=IOR((96+IYI-32*(IYI/32))*256,32+IXI/32)
206      ICAR(3)=(64+IXI-32*(IXI/32))*256
07      IF (K.EQ.0) GO TO 1
08      ICAR(3)=IOR(ICAR(3),32+IYF/32)
209      ICAR(4)=IOR((96+IYF-32*(IYF/32))*256,32+IXF/32)
210      ICAR(5)=IOR((64+IXF-32*(IXF/32))*256,31)
11      CALL EXEC (2,506B,ICAR,5)
12      RETURN
213  1    ICAR(3)=IOR(ICAR(3),31)
14      CALL EXEC (2,506B,ICAR,3)
15      RETURN
216      END
217  C
18  C*****
219  C
220      SUBROUTINE PAGE (I)          ,23-8-82
221  C
222  C          I=1    PIDE PAGINA
223  C          I=2    PASA PAGINA
224  C          I=3    SUENA PITIDO
225  C
226      IF (I-2) 1,2,3
227  1    CALL PLOT (0,0,0)
228      IBUF=10
229 10    CALL EXEC (2,506B,IBUF,1)
230      RETURN
231  2    IBUF=6924
232      GO TO 10
233  3    IBUF=7
234      GO TO 10
235      END
236  C
237  C*****
238  C
239      SUBROUTINE ERFMP (I,IERR), 23-12-82
240  C

```

```

241 C ANALIZA (IERR) DESPUES DE USAR ALGUNA (FMP CALLS).
242 C (I) ES UNA ETIQUETA QUE PONEMOS PARA ENCONTRAR LA
243 C POSICION DEL ERROR EN EL PROGRAMA.
244 C
245     IF (IERR) 1,2
246 1   WRITE (6,100) I,IERR
247 100 FORMAT (/I5,5X"FMP ERROR CODE:"I4,5X"_"")
248     PAUSE
249 2   RETURN
250     END
251 C
252 C*****
253 C
254     SUBROUTINE HANN(V,NI,NF) , 01 DE SEPT DE 1982
255 C
256 C ** FILTRADO HANNING **
257 C
258 C V VECTOR A SUAVIZAR
259 C NI COMPONENTE DE V EN QUE DEBE INICIARSE EL SUAVIZADO
260 C NF COMPONENTE DE V EN QUE DEBE FINALIZAR EL SUAVIZADO
261 C
262 C COMO CONSECUENCIA DEL SUAVIZADO, LA DESVIACION TIPICA
263 C EN LOS DATOS SE REDUCE EN UN FACTOR SQRT(3/8) = 0.612
264 C
265 C LAS COMPONENTES VALIDAS DEL VECTOR V SERAN, TRAS EL
266 C SUAVIZADO, DESDE V(NI+1) HASTA V(NF-1) .
267 C SIN EMBARGO, CUANDO SE FILTRA UN ESPECTRO DE POTENCIA
268 C CALCULADO MEDIANTE UNA TRASFORMADA RAPIDA DE FOURIER
269 C SON NI = 1 Y NF = N/2+1 (N = NUMERO DE PUNTOS
270 C TRANSFORMADOS) Y EN ESTE CASO SON VALIDOS, TRAS
271 C EL FILTRADO, TODOS LOS VALORES DE V , DESDE V(1)
272 C HASTA V(N/2+1), PUES SE TIENE EN CUENTA EL FENOMENO
273 C DEL ALIASING (BENDAT & PIERSOL, 1966, P. 293).
274 C
275     DIMENSION V(1)
276     NP = NF-NI+1
277     IF(NI.LT.1.OR.NP.LT.3) GO TO 99
278     NI1 = NI+1
279     NF1 = NF-1
280     A = V(NI)
281     V(NI) = 0.5*(V(NI)+V(NI1))
282     C = 0.5*(V(NF)+V(NF1))
283     DO 1 I=NI1,NF1
284     B = V(I)
285     V(I) = 0.25*(A+2.*V(I)+V(I+1))
286 1   A = B
287     V(NF) = C
288     RETURN
289 99  WRITE(6,100) NI,NF
290 100 FORMAT(/" ERROR EN HANN : NI = "I5"      NF = "I5/")
291     RETURN
292     END
293 C
294 C*****
295 C
296     SUBROUTINE SAP3(A,NPT,SIG,FE,XP,NPA,IPIFE,IPAE)
297 C
298 C ESTA SUBROUTINA AJUSTA UN POLINOMIO DE GRADO 3 A LOS VALORES 'A'
299 C EVITANDO ALGUNOS PUNTOS DADOS POR IPAE Y EN UNOS INTERVALOS
300 C QUE SE VAN A LEER A CONTINUACION.

```

```

301 C EL NUMERO MAXIMO DE PUNTOS A AJUSTAR ES DE 316.
302 C
303 DIMENSION IPIFE(10),IPAE(10),YP(316),COE(4),A(1),XP(1)
304 DO 2 I=1,10
305 IPAE(I) = 0
306 2 IPIFE(I) = 0
307 C
308 5 IF(NPT.GT.0.AND.NPT.LE.316) GO TO 20
309 WRITE(6,200) NPT
310 200 FORMAT("N ="I6">316 , <1")
311 RETURN
312 300 FORMAT(" PUNTOS INICIAL Y FINAL A ELIMINAR (NO.MAX.INTERV.=5)")
313 20 CALL GRAFE(A(1),NPT,0.,310.,1000.,400.,10,FE,100.)
314 CALL NUMBR(0.,713.,1000.,NPT,10)
315 CALL PLOT(0,0,235,0,0)
316 WRITE(6,300)
317 READ(1,*)IPIFE
318 WRITE(6,400)
319 400 FORMAT(" PUNTOS AISLADOS A ELIMINAR (NO.MAX.PUNTOS = 10)")
320 READ(1,*) IPAE
321 C
322 C INTERVALO DE PUNTOS A ESTUDIAR
323 C
324 IJ = 0
325 NPI = 1
326 NPF = NPT
327 IF(-IPIFE(1)) 29,36
328 29 IF(IPIFE(1)-2) 31,32
329 31 NPI = IPIFE(2)
330 32 DO 33 I=1,5
331 J = IPIFE(2*I)
332 IF(-J) 34,33
333 34 IJ = 2*I
334 33 CONTINUE
335 C
336 IF(IPIFE(IJ)-NPT) 36,35
337 35 IF(-IJ) 37,36
338 37 NPF = IPIFE(IJ-1)
339 36 CONTINUE
340 C
341 C PUNTOS A ESTUDIAR
342 C
343 IJK = 1
344 IF(1-NPI) 49,48
345 49 IJK = 3
346 48 JK = 1
347 J = NPI
348 IJ = 1
349 47 IF(NPF-J) 472,471
350 471 IF(J-IPAE(JK)) 52,51,52
351 52 IF(J-IPIFE(IJK)) 53,54,53
352 53 XP(IJ) = FLOAT(J)
353 IJ = IJ + 1
354 J = J + 1
355 GO TO 47
356 51 JK = JK + 1
357 J = J + 1
358 GO TO 47
359 54 IJK = IJK + 2
360 J = IPIFE(IJK-1) + 1

```

```

361      GO TO 47
362      C
363      472      NPA = IJ - 1
364      C
365      C  AJUSTE DEL ESPECTRO PARCIAL
366      C
367          DO 59 J=1,NPA
368          X = XP(J)
369          K = IFIX(X)
370      59      YP(J) = A(K)
371      C
372          CALL APOL3(NPA,XP,YP,COE)
373      C
374      C  CALCULO DE LOS RESIDUOS
375      C
376          SUM = 0.0
377          DO 61 J=1,NPA
378          X = XP(J)
379          RESID = YP(J) - (((COE(4)*X+COE(3))*X+COE(2))*X+COE(1))
380      61      SUM = SUM + RESID**2
381      C  SIGMA
382          SIG = SQRT(SUM/FLOAT(NPA-4))
383      C
384          DO 60 J=1,NPT
385          X = FLOAT(J)
386      60      XP(J) = ((COE(4)*X+COE(3))*X+COE(2))*X+COE(1)
387      C
388      C  ESCRIBE VECTORES IPIFE(I) E IPAE(I) EN CABECERA GRAFICA AJUSTE
389      C
390          IF(IPIFE(1))500,550,500
391      500      DO 510 J=1,10
392          IF(IPIFE(J))510,520,510
393      510      CONTINUE
394      520      K=J-1
395          WRITE(6,530)(IPIFE(J),J=1,K)
396      530      FORMAT("INTERVALOS ELIMINADOS : "5(I3,"I3");")
397      550      CONTINUE
398          IF(IPAE(1))560,600,560
399      560      DO 570 J=1,10
400          IF(IPAE(J))570,580,570
401      570      CONTINUE
402      580      K=J-1
403          WRITE(6,590)(IPAE(J),J=1,K)
404      590      FORMAT("PUNTOS ELIMINADOS: "10(I3,""))
405      600      CONTINUE
406          RETURN
407          END
408      C
409      C*****
410      C
411          SUBROUTINE APOL3(NPA,XP,YP,COE)
412      C  ESTA SUBROUTINA AJUSTA NPA PUNTOS DE LA FORMA (XP,YP)
413      C  A UN POLINOMIO DE TERCER GRADO POR EL METODO DE LOS MINIMOS CUADRADOS
414      C
415          DOUBLE PRECISION S(6),SY(4),A(4,5),DCOE(4),SAVE,R,A2,A3,A4,CXP
416          DIMENSION COE(4),XP(1),YP(1)
417          CXP = 0.
418          DO 17 I=1,NPA
419      17      CXP = CXP + XP(I)
420          CXP = CXP/FLOAT(NPA)

```

```

421 DO 55 NK=1,10
422 55 S(NK) = 0.0
423 DO 65 NP=1,NPA
424 XP(NP)= XP(NP)- CXP
425 DO 70 NK=1,6
426 70 S(NK) = S(NK) + XP(NP)**NK
427 SY(1) = SY(1) + YP(NP)
428 DO 65 NK=2,4
429 INK = NK-1
430 65 SY(NK) = SY(NK)+YP(NP)*XP(NP)**INK
431 C
432 C A CONTINUACION HAY LA ADAPTACION DE UNA SUBROUTINA DEL LIBRO
433 C APPLIED NUMERICAL ANALYSIS DE C. F. GERALD, PAGINA 163.
434 C
435 N = 4
436 NP1 = N+1
437 A(1,1) = NPA
438 DO 75 I=2,N
439 A(I,1) = S(I-1)
440 75 A(I,5) = SY(I)
441 A(1,5) = SY(1)
442 DO 80 J=2,N
443 K = J-2
444 DO 80 I=1,N
445 KI = K+I
446 80 A(I,J) = S(KI)
447 DO 20 I=2,N
448 DO 20 J=I,N
449 IF(A(I-1,I-1)) 1,2,1
450 2 IM1 = I-1
451 DO 21 M = I,N
452 IF(A(M,IM1)) 3,21,3
453 3 DO 22 MM=IM1,NP1
454 SAVE = A(M,MM)
455 A(M,MM) = A(IM1,MM)
456 22 A(IM1,MM) = SAVE
457 GO TO 1
458 21 CONTINUE
459 WRITE(6,199)
460 199 FORMAT(" MATRIZ SINGULAR")
461 RETURN
462 1 R = A(J,I-1)/A(I-1,I-1)
463 DO 20 K=I,NP1
464 20 A(J,K) = A(J,K) - R*A(I-1,K)
465 DO 30 I=2,N
466 K = N-I+2
467 R = A(K,NP1)/A(K,K)
468 DO 30 J=I,N
469 L = N-J+1
470 30 A(L,NP1) = A(L,NP1) - R*A(L,K)
471 DO 40 I=1,N
472 40 DCOE(I) = A(I,NP1)/A(I,I)
473 C
474 A4 = DCOE(4)
475 A3 = DCOE(3) - 3.*A4*CXP
476 A2 = DCOE(2) - 2.*DCOE(3)*CXP + 3.*A4*CXP*CXP
477 COE(1) = DCOE(1) - DCOE(2)*CXP + DCOE(3)*CXP*CXP - A4*CXP**3
478 COE(2) = A2
479 COE(3) = A3
480 COE(4) = A4

```

```

81      DO 755 NP=1,NPA
482 755  XP(NP) = XP(NP) + CXP
483      RETURN
84      END
85  C
486 C*****
87  C
88      SUBROUTINE NIVEL(V,NP,FE)
489      DIMENSION V(1)
490 1      WRITE(6,2)
91 2      FORMAT("FACTOR DE ESCALA? (PTOS/K) (YA ESTA BIEN=0) : _")
492      READ(1,*) FES
493      IF(FES.EQ.0) GO TO 30
94      FE = FES/10000.
95 10     CALL PAGE(2)
496      CALL GRAF2(V,NP,0.,476.,1000.,300.,10,FE)
497      CALL NUMBR(0.,454.,1000.,NP,10)
498      CALL PLOT(0,0,432,0,0)
499      GO TO 1
500 30     WRITE(6,31)
501 31     FORMAT("POSICION (DERECHA) DEL SALTO A CORREGIR (NO=0) : _")
502      READ(1,*) I0
503      IF (I0.LE.0) GO TO 70
504      I1 = I0-1
505      IF(I1.EQ.0) I1=1
506      DIF = (V(I0)-V(I1))/10000.
507      WRITE(6,35) DIF
508 35     FORMAT("SALTO ="F7.3" K"/
509      "CONSTANTE A RESTAR A LA DERECHA DEL SALTO ? = _")
510      READ(1,*) DIF
511      DO 50 I=I0,NP
512 50     V(I) = V(I)-DIF*10000.
513      CALL GRAF2(V,NP,0.,66.,1000.,300.,10,FE)
514      CALL NUMBR(0.,44.,1000.,NP,10)
515      CALL PLOT(0,0,22,0,0)
516 70     WRITE(6,80)
517 80     FORMAT("CORRIGES ALGUN 'SPIKE' ? (SI=1, NO=0, SALTO=-1) : _")
518      READ(1,*) KK
519      IF(KK) 10,120,85
520 85     CALL PAGE(2)
521      CALL GRAF2(V,NP,0.,458.,1000.,300.,10,FE)
522      CALL NUMBR(0.,436.,1000.,NP,10)
523      CALL PLOT(0,0,414,0,0)
524      WRITE(6,90)
525 90     FORMAT("PUNTOS LATERALES DEL 'SPIKE'? J1,J2 : _")
526      READ(1,*) J1,J2
527      VA = (V(J1)+V(J1-1))/2
528      VB = (V(J2)+V(J2+1))/2
529      JA = J1+1
530      JB = J2-1
531      IF(JA-JB) 100,95,100
532 95     V(JA) = (VA+VB)/2
533      GO TO 110
534 100    DO 105 JJ=JA,JB
535      A = (VB-VA)/(J2-J1)
536 105    V(JJ) = VA + A*(JJ-J1)
537 110    CALL GRAF2(V,NP,0.,88.,1000.,300.,10,FE)
538      CALL NUMBR(0.,66.,1000.,NP,10)
539      CALL PAGE(1)
540      GO TO 70

```

```

541 120 RETURN
542 END
543 C
544 C*****
545 C
546 SUBROUTINE GRAF2(Y,NP,X0,Y0,DX,DY,IBR,FE)
547 DIMENSION Y(1)
548 C
549 C ESTA SUBROUTINE SIRVE PARA DIBUJAR EN LA PANTALLA (TEKTRONIX)
550 C LA GRAFICA CORRESPONDIENTE A LOS VALORES DE LA VARIABLE DIMEN-
551 C SIONADA (Y).
552 C AJUSTA LA ESCALA AL TAMAGNO DE LA GRAFICA, DIBUJA BARRAS Y LA
553 C LINEA CERO SI CAE DENTRO.
554 C EL SIGNIFICADO DE LOS PARAMETROS ES EL SIGUIENTE:
555 C Y=NOMBRE DE LA VARIABLE DIMENSIONADA A REPRESENTAR.
556 C NP=NUMERO DE PUNTOS DE LA VARIABLE.
557 C X0, Y0=COORDENADAS DEL PUNTO INFERIOR IZQUIERDA DE LA GRAFI-
558 C CA EN UNIDADES DE PANTALLA.
559 C DX, DY=ANCHO Y ALTO DE LA GRAFICA EN UNIDADES DE PANTALLA.
560 C IBR=DIVISOR COMUN DE LAS BARRAS A DIBUJAR.
561 C
562 CALL AMAX (Y,NP,YMAX,I)
563 CALL AMIN (Y,NP,YMIN,I)
564 N=NP-1
565 AX=DX/FLOAT(N)
566 DO 20 I=1,N
567 IXI=FLOAT(I-1)*AX+X0+.5
568 IXF=FLOAT(I)*AX+X0+.5
569 IYI=(Y(I)-YMIN)*FE+Y0+.5
570 IYF=(Y(I+1)-YMIN)*FE+Y0+.5
571 CALL PLOT (1,IXI,IYI,IXF,IYF)
572 IF (MOD(I+1,IBR)) 20,21,20
573 21 IY0=Y0
574 IYF0=DY+Y0
575 CALL PLOT (1,IXF,IY0,IXF,IYF0)
576 20 CONTINUE
577 IF (YMAX) 50,30,30
578 30 IF (YMIN) 31,31,50
579 31 IXI=X0
580 IXF=X0+DX
581 IYI=-YMIN*FE+Y0+.3
582 CALL PLOT (1,IXI,IYI,IXF,IYI)
583 50 RETURN
584 END
585 C
586 C*****
587 C
588 SUBROUTINE AXIS (IX,IY,NX,LX,NY,LY,LTX,LTY) ,26-8-82
589 C
590 C DIBUJO DE EJES CARTESIANOS EN LA PANTALLA
591 C
592 C IX,IY: ORIGEN DE LOS EJES
593 C NX,NY: NUM. DE DIVISIONES EN LOS EJES X E Y
594 C LX,LY: LONGITUD DE LOS EJES
595 C LTX,LTY:LONGITUD DE LOS TRAZOS EN LOS EJES X E Y
596 C
597 CALL PLOT (1,IX,IY,IX+LX,IY)
598 CALL PLOT (1,IX,IY,IX,IY+LY)
599 DX=FLOAT(LX)/NX
600 DY=FLOAT(LY)/NY

```



```

01      IYF=IY+LTX
02      DO 10 I=1,NX
03      IXI=IX+DX*I+.5
04  10   CALL PLOT (1,IXI,IY,IXI,IYF)
05      IXF=IX+LTY
06      DO 20 I=1,NY
07      IYI=IY+DY*I+.5
08  20   CALL PLOT (1,IX,IYI,IXF,IYI)
09      RETURN
10     END
11     C
12     C*****
13     C
14     SUBROUTINE MOUNT (NP,NF,DX,DY,ALFA,MX,MY,IDIM1,F), 9-12-82
15     C
16     C  ESTA SUBRUTINA TRAZA UNA PERSPECTIVA DE UNA FUNCION
17     C  BIDIMENSIONAL DE VALORES POSITIVOS.
18     C
19     C  NP=NUMERO DE PUNTOS POR FILA.
20     C  NF=NUMERO DE FILAS.
21     C  DX,DY=DIMENSIONES EN LA PANTALLA DE LA PROYECCION
22     C  ORTOGONAL DEL CONJUNTO.
23     C  ALFA=ANGULO QUE FORMA LA LINEA DE MIRA CON LA HORIZONTAL.
24     C  MX,MY=POSICION QUE DAMOS EN LA PANTALLA AL ANGULO INFERIOR
25     C  IZQUIERDO DE LA GRAFICA.
26     C  IDIM1=PRIMERA DE LAS DOS DIMENSIONES CON QUE FUE DEFINIDO
27     C  EL ARRAY F.
28     C  F=ARRAY DE LOS DATOS.
29     C
30     DIMENSION F(1),NO(100),N1(100)
31     C
32     ALFA=ALFA/57.29578
33     NP1=NP-1
34     ADX=DX/NP1
35     ADY=DY*SIN(ALFA)/(NF-1)
36     CALFA=COS(ALFA)
37     DO 11 I=1,NP
38     DO 11 J=1,NF
39     IJ=I+(J-1)*IDIM1
40  11   F(IJ)=F(IJ)*CALFA
41     DO 20 I=1,NP1
42     IXI=MX+ADX*(I-1)
43     IXF=IXI+ADX
44     NNOS=IXF-IXI+1
45     DO 10 J=1,NNOS
46     N1(J)=1000
47  10   NO(J)=0
48     DO 20 J=1,NF
49     IJ=I+(J-1)*IDIM1
50     ADYJ1=ADY*(J-1)+MY
51     IYI=ADYJ1+F(IJ)
52     IYF=ADYJ1+F(IJ+1)
53     GR=FLOAT(IYF-IYI)/(NNOS-1)
54     IF (IYI.GE.NO(1).AND.IYF.GE.NO(NNOS)) GO TO 1
55     IF (IYI.GE.NO(1).AND.IYF.LT.NO(NNOS)) GO TO 2
56     IF (IYI.LT.NO(1).AND.IYF.GE.NO(NNOS)) GO TO 3
57     IF (IYI.LE.NO(1).AND.IYF.LE.NO(NNOS)) GO TO 4
58     C
59     1   CALL PLOT (1,IXI,IYI,IXF,IYF)
60     DO 30 K=1,NNOS

```

```

661 30  N0(K)=IYI+(K-1)*GR
662      GO TO 200
663  C
664  2    DO 31 K=1,NNOS
665      NYF=IYI+(K-1)*GR
666      IF (N0(K).GT.NYF) GO TO 32
667 31  N0(K)=NYF
668 32  NXF=IXI+K-2
669      NYF=IYI+(K-2)*GR
670      CALL PLOT (1,IXI,IYI,NXF,NYF)
671      GO TO 200
672  C
673  3    DO 33 K=NNOS,1,-1
674      NYI=IYI+(K-1)*GR
675      IF (N0(K).GT.NYI) GO TO 34
676 33  N0(K)=NYI
677 34  NXI=IXI+K
678      NYI=IYI+K*GR
679      CALL PLOT (1,NXI,NYI,IXF,IYF)
680      GO TO 200
681  C
682  4    DO 35 K=1,NNOS
683      NYI=IYI+(K-1)*GR
684      IF (N0(K).LE.NYI) GO TO 36
685 35  CONTINUE
686      GO TO 200
687 36  NOI=K
688      NXI=IXI+K-1
689      DO 37 K=NNOS,1,-1
690      NYF=IYI+(K-1)*GR
691      IF (N0(K).LE.NYF) GO TO 38
692 37  CONTINUE
693      GO TO 200
694 38  NOF=K
695      NXF=IXI+K-1
696      CALL PLOT (1,NXI,NYI,NXF,NYF)
697      DO 40 K=NOI,NOF
698 40  N0(K)=IYI+(K-1)*GR
699 200  IF (IYI.GT.N1(1).AND.IYF.GT.N1(NNOS)) GO TO 8
700      IF (IYI.LE.N1(1).AND.IYF.LE.N1(NNOS)) GO TO 5
701      IF (IYI.LE.N1(1).AND.IYF.GT.N1(NNOS)) GO TO 6
702      IF (IYI.GT.N1(1).AND.IYF.LE.N1(NNOS)) GO TO 7
703  C
704  5    CALL PLOT (1,IXI,IYI,IXF,IYF)
705      DO 50 K=1,NNOS
706 50  N1(K)=IYI+(K-1)*GR
707      GO TO 20
708  C
709  6    DO 61 K=1,NNOS
710      NYF=IYI+(K-1)*GR
711      IF (N1(K).LT.NYF) GO TO 62
712 61  N1(K)=NYF
713 62  NXF=IXI+K-2
714      NYF=IYI+(K-2)*GR
715      CALL PLOT (1,IXI,IYI,NXF,NYF)
716      GO TO 20
717  C
718  7    DO 71 K=NNOS,1,-1
719      NYI=IYI+(K-1)*GR
720      IF (N1(K).LT.NYI) GO TO 72

```

```

721 71 N1(K)=NYI
722 72 NXI=IXI+K
723 NYI=IYI+K*GR
724 CALL PLOT (1,NXI,NYI,IXF,IYF)
725 GO TO 20
726 C
727 8 DO 81 K=1,NNOS
728 NYI=IYI+(K-1)*GR
729 IF (N1(K).GE.NYI) GO TO 82
730 81 CONTINUE
731 GO TO 20
732 82 NOI=K
733 NXI=IXI+K-1
734 DO 83 K=NNOS,1,-1
735 NYF=IYI+(K-1)*GR
736 IF (N1(K).GE.NYF) GO TO 84
737 83 CONTINUE
738 GO TO 20
739 84 NOF=K
740 NXF=IXI+K-1
741 CALL PLOT (1,NXI,NYI,NXF,NYF)
742 DO 85 K=NOI,NOF
743 85 N1(K)=IYI+(K-1)*GR
744 20 CONTINUE
745 C
746 DO 111 I=1,NP
747 DO 111 J=1,NF
748 IJ=I+(J-1)*IDIM1
749 111 F(IJ)=F(IJ)/CALFA
750 RETURN
751 END
752 C
753 C*****
754 C
755 SUBROUTINE ISOF(NX,NY,DX,DY,VMF,DV,LIN,MX,MY,IDIM1,F),26-8-82
756 C
757 C NX,NY: NUM. DE PUNTOS EN X E Y DE LA MATRIZ A DIBUJAR.
758 C DX,DY: TAMAGNO DE LA GRAFICA EN X E Y.
759 C VMF: VALOR DE LA PRIMERA ISOFOTA A TRAZAR.
760 C DV: EQUIDISTANCIA ENTRE ISOFOTAS.
761 C LIN: NUMERO DE ISOFOTAS A TRAZAR.
762 C MX,MY: COORDENADAS DEL PUNTO INFERIOR-IZQUIERDA,
763 C CORRESPONDIENTE AL PUNTO F(1).
764 C IDIM1: PRIMERA DE LAS DOS DIMENSIONES CON QUE FUE
765 C DEFINIDO EL ARRAY F.
766 C F: ARRAY DE LOS DATOS.
767 C
768 DIMENSION FP(4),IPX(4),IPY(4),F(1)
769 C
770 NX1=NX-1
771 NY1=NY-1
772 AX=DX/NX1
773 AY=DY/NY1
774 DO 19 NC=1,LIN
775 DO 1 J=1,NX1
776 DO 1 K=1,NY1
777 JK=J+(K-1)*IDIM1
778 FP(1)=F(JK)
779 FP(2)=F(JK+1)
780 FP(3)=F(JK+IDIM1)

```

```
781 FP(4)=F(JK+IDIM1+1)
782 VM=(FP(1)+FP(2))/2.
783 DIF=ABS(FP(1)-FP(2))/2.
784 VM=ABS(VM-VMF)
785 IF(VM-DIF)2,2,3
786 2 IF(FP(1)-FP(2))47,48,47
787 47 DIF=ABS(FP(1)-VMF)*AX/ABS(FP(1)-FP(2))
788 IX=IFIX(DIF)
789 GO TO 49
790 48 IX=AX/2+.5
791 49 IPX(1)=MX+(J-1)*AX+.5+IX
792 IPY(1)=MY+(K-1)*AY+.5
793 GO TO 100
794 3 IPX(1)=-100
795 IPY(1)=-100
796 100 VM=(FP(3)+FP(4))/2.
797 DIF=ABS(FP(3)-FP(4))/2.
798 VM=ABS(VM-VMF)
799 IF(VM-DIF)4,4,5
800 4 IF(FP(3)-FP(4))50,51,50
801 50 DIF=ABS(FP(3)-VMF)*AX/ABS(FP(3)-FP(4))
802 IX=IFIX(DIF)
803 GO TO 52
804 51 IX=AX/2+.5
805 52 IPX(2)=MX+(J-1)*AX+.5+IX
806 IPY(2)=MY+K*AY+.5
807 GO TO 110
808 5 IPX(2)=-100
809 IPY(2)=-100
810 110 VM=(FP(1)+FP(3))/2.
811 DIF=ABS(FP(1)-FP(3))/2.
812 VM=ABS(VM-VMF)
813 IF(VM-DIF)6,6,7
814 6 IF(FP(1)-FP(3))53,54,53
815 53 DIF=ABS(FP(1)-VMF)*AY/ABS(FP(1)-FP(3))
816 IY=IFIX(DIF)
817 GO TO 55
818 54 IY=AY/2+.5
819 55 IPX(3)=MX+(J-1)*AX+.5
820 IPY(3)=MY+(K-1)*AY+.5+IY
821 GO TO 120
822 7 IPX(3)=-100
823 IPY(3)=-100
824 120 VM=(FP(2)+FP(4))/2.
825 DIF=ABS(FP(2)-FP(4))/2.
826 VM=ABS(VM-VMF)
827 IF(VM-DIF)8,8,9
828 8 IF(FP(2)-FP(4))56,57,56
829 56 DIF=ABS(FP(2)-VMF)*AY/ABS(FP(2)-FP(4))
830 IY=IFIX(DIF)
831 GO TO 58
832 57 IY=AY/2+.5
833 58 IPX(4)=MX+J*AX+.5
834 IPY(4)=MY+(K-1)*AY+.5+IY
835 GO TO 130
836 9 IPX(4)=-100
837 IPY(4)=-100
838 130 CONTINUE
839 NN=0
840 DO 11 L=1,4
```

```
841      IF(IPX(L))10,11,11
842  10    NN=NN+1
843  11    CONTINUE
844      IF(NN-4)12,1,1
845  12    IF(NN)13,13,14
846  13    CALL PLOT(1,IPX(1),IPY(1),IPX(2),IPY(2))
847      CALL PLOT(1,IPX(3),IPY(3),IPX(4),IPY(4))
848      GO TO 1
849  14    IF(NN-3)15,1,1
850  15    LK=0
851      DO 16 L=1,4
852      IF(IPX(L))16,17,17
853  17    LK=LK+1
854      IPX(LK)=IPX(L)
855      IPY(LK)=IPY(L)
856  16    CONTINUE
857      DO 18 L=2,LK
858      L1=L-1
859  18    CALL PLOT(1,IPX(L1),IPY(L1),IPX(L),IPY(L))
860  1     CONTINUE
861  19    VMF=VMF+DV
862      RETURN
863      END
864  $
```

LI, RUAJA

RUAJA T=00004 IS 01 CR00012 USING 00003 BLKS R=0033

0001 : RP, AJUS1  
0002 : RP, AJUS2  
0003 : RP, AJUL1  
0004 : RP, COMPD  
0005 : RP, COMPI  
0006 : RP, DUMPA  
0007 : RP, DUMPD  
0008 : RP, DUMPI  
0009 : RP, EJTRA  
0010 : RP, ESCFA  
0011 : RP, ESCFI  
0012 : RP, FACTD  
0013 : RP, FIJ  
0014 : RP, GRAFD  
0015 : RP, GRAFI  
0016 : RP, IJTE  
0017 : RP, LECFA  
0018 : RP, LECFD  
0019 : RP, LECFI  
0020 : RP, LISTD  
0021 : RP, LISTI  
0022 : RP, MAJDO  
0023 : RP, MOJTI  
0024 : RP, RESTD  
0025 : RP, RESTI  
0026 : RP, SALTU  
0027 : RP, SALT I  
0028 : RP, SJAVD  
0029 : RP, SJAVI  
0030 : RU, AJA

:

LI, RUVISU

RUVISU T=00004 IS 01 CR00012 USING 00002 BLKS R=0011

0001 : RP, DUMPV  
0002 : RP, EJTRU  
0003 : RP, FINV  
0004 : RP, GRAFV  
0005 : RP, ISOFV  
0006 : RP, LECFV  
0007 : RP, MAJDV  
0008 : RP, MOJTV  
0009 : RP, PUJTV  
0010 : RP, Z001V  
0011 : RU, VISU

: