

**TRASVASE DEL SOFTWARE DE HOLOGRAFÍA  
DE VMS/VAX A HP/UX**

JOSÉ ANTONIO LÓPEZ PÉREZ

ALBERTO BARCIA CANCIO

Informe Técnico IT-CAY 1998-1

## ÍNDICE DE CONTENIDO

<b><u>INTRODUCCIÓN</u></b>	<b>4</b>
<b><u>CONFIGURACIÓN DE LA CUENTA DE HOLOGRAFÍA DEL CAY</u></b>	<b>5</b>
<b><u>TRASVASE DE FICHEROS FUENTE</u></b>	<b>6</b>
<b><u>CAMBIOS GENERALES</u></b>	<b>7</b>
<b><u>EL PROGRAMA <i>REFORMAT</i></u></b>	<b>8</b>
EL FORMATO RGDATA DE GREG.	8
MODIFICACIONES AL CÓDIGO DE REFORMAT	9
LOCALIZACIÓN DE LOS FICHEROS	9
PROTOCOLO DE COMPILACIÓN Y EJECUCIÓN DE REFORMAT	9
VISUALIZACIÓN DE LOS RESULTADOS DE REFORMAT	10
<b><u>EL PROGRAMA <i>MAPMASK</i></u></b>	<b>11</b>
LOCALIZACIÓN DE LOS FICHEROS	11
PROTOCOLO DE COMPILACIÓN Y EJECUCIÓN DE MAPMASK	11
VISUALIZACIÓN DE LOS RESULTADOS DE MAPMASK	12
<b><u>EL PROGRAMA <i>PICOMISTY</i></u></b>	<b>13</b>
MODIFICACIONES AL CÓDIGO DE PICOMISTY	13
LOCALIZACIÓN DE LOS FICHEROS	13
PROTOCOLO DE COMPILACIÓN Y EJECUCIÓN DE PICOMISTY	14
VISUALIZACIÓN DE LOS RESULTADOS DE PICOMISTY	15
EFFECTOS DEL REDONDEO DE NÚMEROS REALES	17
<b><u>EL PROGRAMA <i>AVMAP</i></u></b>	<b>19</b>
MODIFICACIONES AL CÓDIGO DE AVMAP	19
LOCALIZACIÓN DE LOS FICHEROS	19
PROTOCOLO DE COMPILACIÓN Y EJECUCIÓN DE AVMAP	19
VISUALIZACIÓN DE LOS RESULTADOS DE AVMAP	20
<b><u>EL PROGRAMA <i>MAPERROR</i></u></b>	<b>21</b>
MODIFICACIONES AL CÓDIGO DE MAPERROR	21
LOCALIZACIÓN DE LOS FICHEROS	21
PROTOCOLO DE COMPILACIÓN Y EJECUCIÓN DE MAPERROR	21
VISUALIZACIÓN DE LOS RESULTADOS DE MAPERROR	21

<b><u>PROTOCOLO PARA LA UTILIZACIÓN DEL SOFTWARE DE HOLOGRAFÍA</u></b>	<b><u>23</u></b>
<b><u>REFERENCIAS BIBLIOGRÁFICAS</u></b>	<b><u>24</u></b>
<b><u>ANEXOS</u></b>	<b><u>34</u></b>

#### **INDICE DE FIGURAS**

<b>Figura 1: Configuración actual de la cuenta de holografía.....</b>	<b>5</b>
<b>Figura 2: Ejemplo de utilización del software de holografía.....</b>	<b>23</b>

## INTRODUCCIÓN

Este informe técnico describe el trasvase de los programas FORTRAN de holografía, que implementan la técnica de recuperación de fase mediante el algoritmo de Misell, de la cuenta HOLO del VMS/VAX (Isis) a la cuenta HOLO del HP/UX (Siva).

El motivo de este trasvase no es otro que el de aprovechar las mejores prestaciones que ofrece la máquina HP/UX frente a la VMS/VAX en cuanto a velocidad de computación, capacidad de memoria y presentación gráfica. Además, los sistemas VMS/VAX están cayendo en desuso frente a los sistemas Unix, para los cuales existen además toda una serie de herramientas freeware muy útiles para el análisis, reducción y presentación de datos que no existen para sistemas VMS/VAX.

No se describirá un simple trasvase de ficheros de un sistema a otro porque si fuese así, este informe técnico no tendría sentido. Al tratarse de dos sistemas operativos diferentes (VAX vs. UNIX) y dos implementaciones de lenguaje FORTRAN ligeramente diferentes, han surgido una serie de inconvenientes en la compilación de los ficheros fuente en el sistema HP/UX que hemos tenido que resolver. Además, se ha modificado completamente la presentación de resultados gráficos utilizando para ello las librerías de GREG.

Se describirá, asimismo, la configuración de la cuenta de holografía creada en el HP/UX Siva (username HOLO) y el protocolo de realización de mapas de holografía mediante este software.

Se comentará el trasvase de ficheros fuente propiamente dichos de una máquina a otra. A continuación, el resto de apartados describirán los cambios realizados en cada uno de los ficheros fuente, el procedimiento de compilación de cada uno de ellos en el nuevo sistema operativo, los datos de entrada que necesitan y los resultados de salida que generan así como la creación de ficheros “*script*” para la ejecución automática/desatendida de los mismos.

Finalmente, los anexos recogen los resultados obtenidos al ejecutar los programas en el nuevo sistema tomando como entrada los últimos mapas tomados en Julio de 1995.

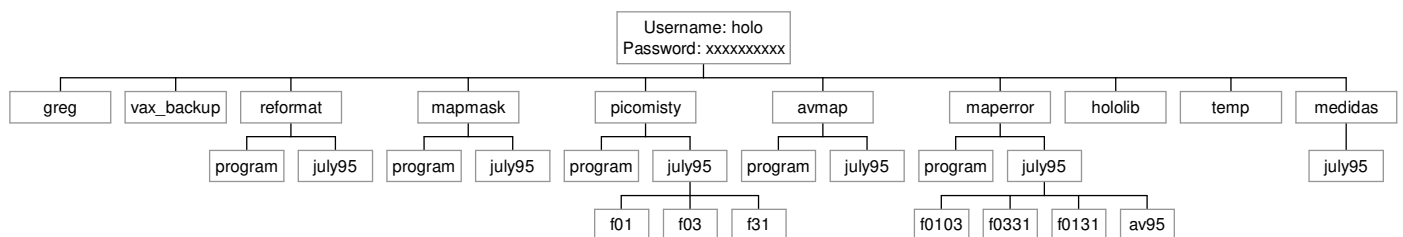
En las referencias [1]-[4] puede encontrarse una descripción más detallada del funcionamiento de los programas en el VMS/VAX y el procedimiento de observación para obtener datos así como resultados interesantes derivados de la actividad holográfica en el CAY.

## CONFIGURACIÓN DE LA CUENTA DE HOLOGRAFÍA DEL CAY

Para la ejecución de los programas de reducción y análisis de datos de holografía se ha creado un cuenta con username *holo* en la máquina HP/UX Siva. Se ha decidido configurar la cuenta del siguiente modo:

- Crear un directorio para cada uno de los programas que se utilizan y cuyo nombre coincida con el del programa que alojará.
- Dentro de este directorio se ha creado un subdirectorio, llamado *program*, que alojará al fichero fuente del programa en cuestión (extensión *.f*), su fichero ejecutable (sin extensión), el fichero *makefile* para posteriores recompilaciones y los ficheros “*script*”-*ejemplo*, si los hubiere, que se consideren oportunos para la ejecución automática de dicho programa (la utilidad y significado de estos ficheros se explicará más adelante).
- También, dentro del directorio cuyo nombre es igual al del programa, se crearán, según se vayan necesitando, subdirectorios cuyo nombre haga alusión a la fecha en que se hacen las observaciones, reducción y análisis de datos de holografía, indicando simplemente el mes y año de observación. Estos subdirectorios contendrán los datos de entrada para el programa que se va a ejecutar y los ficheros “*script*” que se necesiten en su caso para ejecutar automáticamente el programa. El programa se ejecutará dentro de este directorio y nunca se ejecutará en el subdirectorio *program*.
- Un directorio llamado *hololib*, que recoge todas las subrutinas utilizadas por los diferentes programas. Las futuras subrutinas que se generen residirán en este directorio para que puedan ser compartidas por varios programas.
- Un directorio llamado *medidas* donde se recogerán los archivos que contienen las observaciones previa aplicación del programa *javier* en el HP-1000.
- Un directorio temporal, llamado *temp*, para copiar archivos temporales o hacer pruebas.
- Un directorio llamado *vax\_backup* que recogerá copia de los ficheros fuente originales para VMS/VAX.
- Un directorio llamado *greg* que guarda una copia de seguridad de la versión de GREG que se utiliza en este trabajo.

La figura siguiente muestra la configuración actual de la cuenta de holografía del CAY, en la que se aprecia la estructura del árbol de directorios:



**Figura 1: Configuración actual de la cuenta de holografía.**

Esta estructura no será definitiva ya que aún quedan por trasvasar algunos programas que, debido a limitaciones de tiempo, no se han trasvasado aún.

## TRASVASE DE FICHEROS FUENTE

Los ficheros fuente del software de holografía se encuentran en la cuenta HOLO, directorio [hl.retriev.data] del VMS/VAX Isis del CAY. Estos ficheros tienen, por tratarse de fuentes FORTRAN, la extensión .FOR.

El trasvase de una máquina a otra se hace vía *ftp*. Por ejemplo, el trasvase del fichero REFORMAT.FOR se realizaría de la siguiente manera:

- Nos situamos en la cuenta HOLO de Siva, *username holo*, *password* conocido por José Antonio López Pérez, Alberto Barcia Cancio y Pablo de Vicente.
- Abrimos un *xterm*.
- Tecleamos *ftp isis*.
- El VMS/VAX nos pedirá el *username*, en este caso *holo*, y el *password*, conocido por José Antonio López Pérez, Alberto Barcia Cancio y Pablo de Vicente.
- Una vez suministrados estos datos se escribirá: *cd [hl.retriev]*. con lo cual nos situamos en el directorio donde está el fichero que queremos copiar.
- Ahora tecleamos *asc*, para que el modo de transmisión sea ASCII, ya que es un fichero de texto.
- Finalmente escribiremos: *get REFORMAT.FOR*, lo cual generará una copia de este fichero en el directorio en el que nos encontremos de nuestra cuenta de Siva.

Los ficheros fuentes así copiados se guardaron en el directorio *vax\_backup* de la cuenta HOLO en Siva, y servirán de copia de seguridad y estarán protegidos contra escritura; además, si se desean nuevas modificaciones de alguno de estos ficheros se pueden copiar directamente de este directorio a nuestro directorio de trabajo sin necesidad de hacer *ftp*.

Sobre copias de estos programas fuente hemos realizado las modificaciones pertinentes para su adaptación al nuevo sistema operativo.

## CAMBIOS GENERALES

En este apartado se describirán una serie de cambios en los ficheros fuente de los programas de holografía que afectan a todos ellos de forma global.

En primer lugar, las partes de código que generan archivos de salida gráficos han sido sustituidas por subrutinas que realizan funciones análogas pero están basadas en comandos de GREG. Por tanto, es necesario incluir las librerías de GREG en el proceso de compilación de los ficheros fuente. Estas librerías están recogidas en [5], [6] y [7]. Es necesario en este punto destacar que podemos encontrar problemas con la última versión de GREG, sobre todo a la hora de generar ficheros .PS; por este motivo se ha utilizado en este trabajo la versión de Enero de 1995, que no presenta ningún inconveniente.

En segundo lugar, se ha adoptado como norma general que los nombres de los ficheros de entrada y salida de los programas tengan nueve caracteres, ya sean letras o dígitos, sin contar la extensión. Esta norma no es aplicable a los ficheros gráficos que generan los programas, ya que son nombrados por el propio programa con nombres alusivos a su contenido.

Si el nombre que deseamos utilizar tiene menos de nueve caracteres, podemos rellenar con caracteres poco usuales hasta completar nueve caracteres, así, en este trabajo se ha completado, como se verá más adelante, con letras “*q*” hasta tener nueve caracteres.

Por otro lado, es necesario que los ficheros que vayan a servir de entrada a un programa residan en el directorio desde donde se vaya a ejecutar el programa.

Finalmente, se comprobó que varios ficheros fuente hacían usos de subrutinas comunes y que éstas estaban contenidas dentro del cuerpo del fichero; se decidió, para así reducir el número de líneas, eliminar estas subrutinas de los ficheros fuente y crear un directorio “biblioteca”, llamando *hololib* en la cuenta HOLO de Siva, que contuviese todas aquellas subrutinas que los programas de holografía utilizan. Así, si un programa utiliza una subrutina de la biblioteca, tendrá que incluirla en la secuencia de compilación indicando que se encuentra en dicho directorio. Esto se indica en el fichero *makefile*, que realizará la compilación de manera automática.

## EL PROGRAMA *REFORMAT*

El programa *reformat* convierte los datos ASCII, previamente convertidos de tipo ACON a tipo ASCII por el programa *javier* que se ejecutará en el HP-1000 después de la observación, a formato RGDATA de GREG. Además, es capaz de abstraer el offset del mapa y normalizarlo para que la potencia total sea igual a la unidad.

### **El formato RGDATA de GREG.**

El formato del fichero de entrada al programa GREG consiste en una cabecera de cuatro líneas como se describe a continuación:

1ª línea:       NX           XREF           XVAL           XINC  
2ª línea:       texto comentario para el eje X; puede estar en blanco.  
3ª línea:       NY           YREF           YVAL           YINC  
4ª línea:       texto comentario para el eje Y; puede estar en blanco.  
5ª línea y ss.: datos en formato 10z8, que es el que usa GREG por defecto.

donde:

NX: número de datos según el eje X,  
NY: número de datos según el eje Y,  
XREF: índice del dato para el cual su coordenada X vale XVAL,  
YREF: índice del dato para el cual su coordenada Y vale YVAL,  
XVAL: valor de la coordenada de referencia del eje X,  
YVAL: valor de la coordenada de referencia del eje Y,  
XINC: espaciado entre dos datos consecutivos en el eje X,  
YINC: espaciado entre dos datos consecutivos en el eje Y.

Así, la coordenada X del dato i-ésimo según el eje X es obtenida por GREG como:

$$X = (i - XREF) * XINC + XVAL$$

y la coordenada Y del dato j-ésimo según el eje Y es obtenida mediante:

$$Y = (j - YREF) * YINC + YVAL$$

Sin embargo, en lugar del formato por defecto 10z8, nosotros utilizaremos el formato **g15.7e2**, es decir, números reales escritos en un campo de quince caracteres de ancho con siete decimales multiplicado por una potencia cuya base es 10 y cuyo exponente es un número entero de dos cifras con signo. Este formato permite que los datos puedan además ser interpretados por programas como MATLAB, lo cual puede ser interesante para futuras aplicaciones y desarrollos dada la potencia de cálculo y sencillez de programación de MATLAB.



## **Modificaciones al código de REFORMAT**

Además de la modificación del formato de los números que escribe *reformat*, se ha incluido la opción de abstraer el offset presente en los datos crudos. Si se escoge esta opción, el programa restará el valor mínimo del mapa que se procesa, asegurándose así la no negatividad de los datos de potencia, y realizará un escalado del mapa para tener una potencia total unitaria de modo que todos los mapas de entrada a los programas sean compatibles y estén en igualdad de condiciones. Esta labor de substracción y normalización era realizada en la máquina VMS/VAX por el programa *suboffs*. La inclusión del programa *suboffs* dentro de *reformat* como opción ahorra espacio y reduce el número de programas necesarios para la reducción y análisis de datos de holografía.

Tener en cuenta que si se ejecuta *reformat* sin la opción *suboffs* y luego se quiere eliminar al offset a este mapa, habrá que ejecutar de nuevo *reformat* con la opción *suboffs* sobre el fichero *.dat* original para llevar a cabo la operación.

## **Localización de los ficheros**

En el directorio *vax\_backup* de la cuenta HOLO de la máquina HP/UX se encuentran los ficheros fuente *reformat.for* y *suboffs.for*. Por su parte, en el directorio *reformat/program* de la misma cuenta y máquina están el nuevo fichero fuente, *reformat.f*, el fichero *makefile*, para su compilación y el fichero ejecutable *reformat*.

## **Protocolo de compilación y ejecución de REFORMAT**

La compilación del fichero fuente se lleva a cabo mediante el fichero *makefile*, evitando así escribir largas líneas de comandos Unix. Esto sólo será necesario si se modifica el código fuente, ya que el fichero está ya compilado y listo para la ejecución. Para la compilación, desde un terminal y dentro del directorio *reformat/program* se escribirá:

```
$ makefile
```

Si el código fuente tiene errores la compilación no se completará, obviamente. En cuanto a la ejecución de *reformat* es tan simple como escribir:

```
$ reformat
```

Acto seguido el programa irá pidiendo datos y proporcionando, por un lado, opciones al usuario a las que habrá de responder para continuar la ejecución y, por otro lado, resultados de los cálculos que realiza.

Se reproduce un ejemplo de utilización del programa:

```
User:      $ reformat <Enter>
```

```
Siva:     Ascii Data File Name (.dat assumed) ?
```

```
User:     datfilename <Enter>
```

```

Siva:      RGDATA File Name without extension ?
User:      mapfilename <Enter>
Siva:      OPTIONS MENU:
              1 FOR OFFSET SUBTRACTION
              0 FOR NO OFFSET SUBTRACTION
User:      1 <Enter> (por ejemplo)
Siva:      Power = 6756.5000      Offset = 111.0000
              Xsize = 32          Ysize = 32
              STOP End of program reformat with offset substracion.

```

Volvemos a recordar que especificar que **los nombres de los ficheros de entrada** a *reformat* y a cualquier otro programa de holografía **deben tener nueve caracteres**, ya sean letras o dígitos, sin incluir la extensión, la cual nunca se da como entrada porque los programas la asumen. De lo contrario el programa intentará buscar un archivo que no existe y la ejecución se abortará.

### **Visualización de los resultados de REFORMAT**

Los ficheros de salida de *reformat* pueden ser visualizados con GREG mediante el siguiente procedimiento:

- *Llamada a greg: \$ greg <Enter>*
- *Greg> rgdata filename.map /format g15.7e2 <Enter>*
- *Greg> device image white <Enter>*
- *Greg> set plot\_page square <Enter>*
- *Greg> limits /rgdata <Enter>*
- *Greg> box <Enter>*
- *Greg> levels 0.0001 0.001 0.01 0.1 1 10 100 <Enter>*
- *Greg> rgmap /pen /per 1. <Enter>*
- *Greg> label "Arc Seconds" /x <Enter>*
- *Greg> label "Arc Seconds" /y <Enter>*
- *Greg> draw text 0 1 "10 dB contours (0db to -60dB)" 8 /box 8 <Enter>*
- *Greg> hard /plot <Enter>* (para obtener una copia por impresora)
- *Greg> hard filename.ps /device ps fast <Enter>* (para crear ficheros .ps)

Esto presenta contornos cada 10 dB del diagrama de radiación medido. Si a continuación se desea un perspectiva 3-D teclear:

- *Greg> clear <Enter>*
- *Greg> perspective <Enter>*
- *Greg> hard /plot <Enter>* (para obtener una copia por impresora)
- *Greg> hard filename.ps /device ps fast <Enter>* (para crear ficheros .ps)

El anexo I recoge los resultados gráficos de aplicar *reformat* con substracción y normalización (opción 1) a los ficheros que se obtuvieron en la sesión de Julio de 1995.

## EL PROGRAMA MAPMASK

Este programa aplica una máscara circular de radio seleccionable por el usuario (dado en píxeles) para la eliminación de ruido y posibles distorsiones provocadas por el ruido en las medidas de diagramas de radiación enfocados y desenfocados.

El mapa de entrada que se le proporciona es un diagrama de radiación procesado previamente por *reformat*. El mapa de salida puede ser ya uno de los mapas de entrada a *picomisty*.

### Localización de los ficheros

En el directorio *vax\_backup* de la cuenta HOLO de Siva está el fichero fuente original que se utilizaba en el VMS/VAX, y que se llama *mapmask.for*. Por su parte, la nueva versión para Siva, *mapmask.f*, está en el directorio *mapmask/program*. También se encuentran ahí el fichero *makefile*, para su compilación en caso de cambios en el código fuente, y el fichero ejecutable, *mapmask*.

### Protocolo de compilación y ejecución de MAPMASK

Análogamente a como se ha comentado para el programa *reformat*, el programa *mapmask* puede ser recompilado en caso de cambios en su código. Para ello se utilizará el fichero *makefile*.

Por lo que se refiere a la ejecución del programa, a continuación se muestra un ejemplo de utilización.

```
User:      $ mapmask <Enter>
Siva:      Input RGMAP File (.map assumed) ?
User:      mapfilename1 <Enter>
Siva:      Output RGMAP File please (.map assumed) ?
User:      maskedfilename2 <Enter>
Siva:      Input map size = 32 x 32 pixels
Siva:      Radius (pixels) of circular mask please ?
User:      12 <Enter> (por ejemplo)
Siva:      Width of output map please ?
User:      32 <Enter> (por ejemplo)
Siva:      Height of output map please ?
User:      32 <Enter> (por ejemplo)
           STOP End of program mapmask.
```

---

<sup>1</sup> Debe tener nueve caracteres sin la extensión.

<sup>2</sup> Debe tener nueve caracteres sin la extensión.

## Visualización de los resultados de MAPMASK

Los mapas “enmascarados” por *mapmask* pueden visualizarse con GREG siguiendo los siguientes pasos:

- *Llamada a greg: \$ greg <Enter>*
- *Aparecerá el prompt del programa GREG: Greg>*
- *Greg> rgdata filename.map /format g15.7e2 <Enter>*
- *Greg> device image white <Enter>*
- *Greg> set plot\_page square <Enter>*
- *Greg> limits /rgdata <Enter>*
- *Greg> box <Enter>*
- *Greg> levels 0.0001 0.001 0.01 0.1 1 10 100 <Enter>*
- *Greg> rgmap /pen /per 1. <Enter>*
- *Greg> label "Arc Seconds" /x <Enter>*
- *Greg> label "Arc Seconds" /y <Enter>*
- *Greg> draw text 0 1 "10 dB contours (0db to -60dB)" 8 /box 8 <Enter>*
- *Greg> hard /plot <Enter>* (para obtener una copia por impresora)
- *Greg> hard filename.ps /device ps fast <Enter>* (para crear ficheros .ps)

Esto presenta contornos cada 10 dB del diagrama de radiación medido. Si a continuación se desea un perspectiva 3-D teclear:

- *Greg> clear <Enter>*
- *Greg> perspective <Enter>*
- *Greg> hard /plot <Enter>* (para obtener una copia por impresora)
- *Greg> hard filename.ps /device ps fast <Enter>* (para crear ficheros .ps)

El Anexo II recoge los resultados gráficos de los mapas de salida de *mapmask* que servirán de entrada a *picomisty*, programa principal del software de holografía.

## EL PROGRAMA *PICOMISTY*

Puede decirse que este es el principal programa del software de holografía, ya que es el que toma los diagramas de radiación, enfocado y desenfocado, medidos (previa aplicación de *reformat* y *mapmask*) y calcula la distribución de campo en módulo y fase en la apertura de la antena aplicando el algoritmo de Misell, lo cual permite estimar las imperfecciones del reflector.

### **Modificaciones al código de *PICOMISTY***

Las modificaciones realizadas al fichero fuente son las siguientes:

- Cambio del formato de datos leídos por la subrutina que carga los mapas enfocado y desenfocado; recuérdese que debe ser **g15.7e2**,
- Cambio completo de las partes de código que generaban archivos gráficos de salida. Esta versión utiliza las librerías de GREG para generar toda una serie de posibilidades gráficas que permitan una mejor visualización de los resultados, así, pueden obtenerse ficheros .PS con diagramas de contornos, diagramas de gris, diagramas de contornos en escala de gris, diagramas en color, diagramas en perspectiva 3D y ficheros .GDF para su visualización y manipulación con GRAPHIC.

### **Localización de los ficheros**

En la cuenta HOLO de Siva, dentro del directorio *vax\_backup*, se encuentra una copia del fichero fuente, *picomisty.for*, que existe en el VMS/VAX Isis.

Por su parte, también en la cuenta HOLO de Siva, dentro del directorio *picomisty/program*, se encuentran los siguientes ficheros:

- *picomisty.f*: Nuevo fichero fuente,
- *picomisty*: Fichero ejecutable, resultado de compilar *picomisty.f*,
- *makefile*: Fichero “make” para la compilación y linkado automáticos,
- *pico\_input.job*: fichero con los datos de entrada al programa para ejecución desatendida (nota: sólo sirve de ejemplo),
- *picotask*: fichero “script” que invoca a *picomisty* y le da como entrada *pico\_input.job* para la ejecución desatendida (nota: sólo sirve de ejemplo).

En el directorio *picomisty* de la cuenta HOLO se crearán subdirectorios cuyo nombre indique el mes y año de la observación, dentro de los cuales se copiarán los ficheros *pico\_input.job* y *picotask* y se modificarán según nuestras necesidades. Así, para la comprobación del correcto funcionamiento de los nuevos programas se utilizaron los datos de entrada de las medidas de Julio’1995, de ahí que exista un subdirectorio llamado *july95*.

## Protocolo de compilación y ejecución de PICOMISTY

El protocolo de compilación no difiere en nada de los anteriores programas; basta con ejecutar el fichero *makefile* para que se produzca la recompilación del programa debido a cambios que se hayan realizado en su código fuente. Naturalmente, si se hace uso de una nueva subrutina de la biblioteca, ésta habrá de ser incluida en una línea del fichero *makefile* para su correcta compilación y linkado.

Por lo que se refiere a la ejecución de *picomisty*, podemos optar por dos alternativas: la ejecución manual o la ejecución automática/desatendida.

La ejecución manual consiste en invocar al programa desde el terminal y responder a las preguntas y opciones que presenta el programa para su funcionamiento. Por ser un procedimiento sencillo, similar al de casos anteriores, no se describirá debido a que son muchos los datos de entrada y la explicación sería larga (consultar [1] a este respecto).

La ejecución que se da en llamar automática/desatendida consiste en crear un fichero de comandos que invoque al programa principal y le proporcione como entrada, siguiendo las fórmulas de redirección de entrada/salida de Unix, un fichero que contiene las respuestas a cada una de las preguntas que realiza *picomisty*. Así, podemos despreocuparnos de la ejecución y esperar a que ésta haya terminado para analizar los datos resultantes.

El fichero de comandos antes mencionado se llama *picotask*; su contenido es sencillamente el siguiente texto y debe tener permiso de ejecución:

```
/users2/holo/picomisty/program/picomisty < pico_input.job
rm LOG
```

donde la primera línea invoca a *picomisty* desde el directorio en el que nos encontremos y la segunda borra un fichero LOG generado por GREG que no sirve para nada. Como ya se ha dicho anteriormente, existe un ejemplo de este fichero en *picomisty/program*.

Por su parte, el fichero que contiene los datos de entrada, *pico\_input.job*, tiene la siguiente estructura:

```
17/12/97      !COMENTARIO < 80 CARACTERES (fecha, desenfocues, .)
13.72        !REFLECTOR DIAMETER
5.08         !REFLECTOR FOCAL LENGHT
1.085        !SUBREFLECTOR DIAMETER
11.0         !MAGNIFICATION
0.0          !INITIAL DEFOCUS
49.49        !FREQUENCY (GHZ)
0.0          !TRANSMITTER DISTANCE (0=FAR-FIELD)
1            !1 POINTING CORRECTION, 2 POINT.+FOCUS, 3 POINT.+FOCUS+ASTIGM.
0.0         !XTILTI INITIAL POINTING CORRECTION
0.0         !YTILTI INITIAL POINTING CORRECTION
0           !DEFAULT MMAX
0.0         !RTAPER RADIUS (PIXELS)
0           !1 FOR REMOVAL INPUT OFFSETS3
```

---

<sup>3</sup> Esta opción debe estar a cero si se ha utilizado la opción de sustraer el offset con reformat.

```

1          !1 FOR MASKING OUTSIDE APERTURE
0          !1 FOR MASKING OF CENTRAL BLOCKAGE
f01r1295b !OUTPUT MAP NAME
0          !1 FOR AMPLITUDE DATA
f0r1295bq !INFOCUSSED MAP
0.0       !DEFOCUS 1 IN MM
f1r1295aq !DEFOCUSSED MAP NAME
6.35     !DEFOCUS 2 IN MM
1          !FOR FAR-FIELDS CONTOUR PLOT
2          !FOR FAR-FIELDS GREY PLOT
3          !FOR FAR-FIELDS GREY CONTOUR PLOT
4          !FOR FAR-FIELDS COLOR PLOT
5          !FOR FAR-FIELDS 3D PERSPECTIVE PLOT
0          !TO CONTINUE EXECUTION
500       !ITMAX
715      !ISEED
5          !NREPEAT
1          !FOR AMPLITUDE RESIDUAL CONTOUR PLOT
2          !FOR AMPLITUDE RESIDUAL GREY PLOT
3          !FOR AMPLITUDE RESIDUAL GREY CONTOUR PLOT
4          !FOR AMPLITUDE RESIDUAL COLOR PLOT
5          !FOR AMPLITUDE RESIDUAL 3D PERSPECTIVE PLOT
6          !FOR AMPLITUDE RESIDUAL .GDF FILE
0          !TO CONTINUE EXECUTION
6          !NUMBER OF COEFFICIENTS FOR PHASE FIT
1          !FOR AMPLITUDE+FASE SOLUTION CONTOUR PLOT
2          !FOR AMPLITUDE+FASE SOLUTION GREY PLOT
3          !FOR AMPLITUDE+FASE SOLUTION GREY CONTOUR PLOT
4          !FOR AMPLITUDE+FASE SOLUTION COLOR PLOT
5          !FOR AMPLITUDE+FASE SOLUTION 3D PERSPECTIVE PLOT
6          !FOR AMPLITUDE+FASE SOLUTION .GDF FILE
0          !TO CONTINUE EXECUTION

```

Existe un ejemplo de este fichero en *picomisty/program*. Por tanto, si se opta por la ejecución automática, habremos de copiar los ejemplos de *picotask* y *pico\_input.job* que hay en *picomisty/program* a nuestro directorio de trabajo y allí editarlos para modificarlos según nuestras necesidades.

## **Visualización de los resultados de PICOMISTY**

Los resultados de la ejecución de *picomisty* son:

- Fichero ASCII *ftn66*: Contiene interesante información y resultados de los cálculos llevados a cabo por *picomisty*.
- Ficheros *.ps*: Ficheros PostScript que contienen gráficas de los datos de entrada y los resultados. Pueden ser visualizados con GhostView o impresos. El nombre de estos ficheros, alusivo a su contenido, es el siguiente:
  - ◆ *farfields\_color.ps*: mapa en color de los diagramas de radiación utilizados como entrada a *picomisty*,
  - ◆ *farfields\_contgrey.ps*: mapa en contornos de gris de los diagramas de radiación utilizados como entrada a *picomisty*,
  - ◆ *farfields\_contour.ps*: mapa de líneas de contorno de los diagramas de radiación utilizados como entrada a *picomisty*,
  - ◆ *farfields\_grey.ps*: mapa en escala de grises de los diagramas de radiación utilizados como entrada a *picomisty*,

- ◆ farfields\_perspec: mapa en perspectiva 3D de los diagramas de radiación utilizados como entrada a *picomisty*,
- ◆ picomap\_color.ps: mapa en color de la solución (en módulo y fase) obtenida por *picomisty*,
- ◆ picomap\_contgrey.ps: mapa en contornos de gris de la solución (en módulo y fase) obtenida por *picomisty*,
- ◆ picomap\_contour.ps: mapa de líneas de contornos de la solución (en módulo y fase) obtenida por *picomisty*,
- ◆ picomap\_grey.ps: mapa en escala de gris de la solución (en módulo y fase) obtenida por *picomisty*,
- ◆ picomap\_perspec.ps: mapa en perspectiva 3D de la solución (en módulo y fase) obtenida por *picomisty*,
- ◆ resid\_color.ps: mapa en color de los residuos,
- ◆ resid\_contgrey.ps: mapa en contornos de gris de los residuos,
- ◆ resid\_contour.ps: mapa de contornos de líneas de los residuos,
- ◆ resid\_grey.ps: mapa en escala de gris de los residuos,
- ◆ resid\_perspec.ps: mapa en perspectiva 3D de los residuos,
- Ficheros *.gdf*: Para nuevas visualizaciones y manipulaciones de los resultados con GRAPHIC. Sus nombres son los siguientes:
  - ◆ picomap\_amp.gdf: mapa de la amplitud de la solución,
  - ◆ picomap pha.gdf: mapa de la fase de la solución,
  - ◆ residuals.gdf: mapa de los residuos

Los resultados de la ejecución de *picomisty* sobre varias parejas de diagramas de radiación se recogen en el Anexo III de este informe.



## Efectos del redondeo de números reales

La ejecución de *picomisty* en Siva y en Isis, en igualdad de condiciones de datos de entrada, es ligeramente diferente. Ello es debido al distinto redondeo que realizan ambas máquinas, lo que produce resultados distintos cuando se ejecutan los algoritmos de ajuste de la puntería, focalización y astigmatismo en el plano desenfocado.

La tabla siguiente muestra las diferencias en la ejecución para cada una de las parejas de mapas de entrada:

	VMS/VAX ISIS			HP/UX SIVA		
	<i>f0r1295bq</i> <i>f1r1295aq</i>	<i>f0r1295bq</i> <i>f3r1295bq</i>	<i>f3r1295bq</i> <i>f1r1295aq</i>	<i>f0r1295bq</i> <i>f1r1295aq</i>	<i>f0r1295bq</i> <i>f3r1295bq</i>	<i>f3r1295bq</i> <i>f1r1295aq</i>
Ajuste puntería						
XTILT	0.6755049	1.008824	<b>0.9847891</b>	0.8005048	1.13385	<b>0.9847881</b>
YTILT	0.6162951	0.5981594	<b>1.021054</b>	0.7903141	0.7798594	<b>1.02105</b>
SSQERR 1ª iteración	0.9699	1.097	<b>1.262</b>	0.9742	1.106	<b>1.262</b>
SSQERR 2ª iteración	0.9975	1.096	<b>1.2</b>	0.9975	1.106	<b>1.2</b>
SSQERR 3ª iteración	0.9775	1.093	<b>1.173</b>	0.9871	1.098	<b>1.173</b>
SSQERR 4ª iteración	0.9889	1.101	<b>1.172</b>	0.9889	1.093	<b>1.172</b>
SSQERR 5ª iteración	0.9889	1.101	<b>1.195</b>	0.9889	1.108	<b>1.195</b>
RMSDR/MDA <sup>4</sup> (%)	0.8086677	0.3976804	<b>0.9839682</b>	0.7959442	0.4084971	<b>0.9839676</b>
SSR/ITP <sup>5</sup> (%)	1.108445	1.087124	<b>1.641104</b>	1.07384	1.14707	<b>1.6411</b>
Max. Ampl. Residual	0.015125834	0.01278632	<b>1.637393E-2</b>	0.0130908	1.80559E-2	<b>1.6374E-2</b>
Min. Ampl. Residual	-0.0125	-0.0117863	<b>-1.23634E-2</b>	-0.0125805	-1.19858E-2	<b>-1.23633E-2</b>
Max. Solution Amplitude	4.253E-3	3.911E-3	<b>4.212E-3</b>	4.447E-3	3.897E-3	<b>4.212E-3</b>
Min. Solution Amplitude	8.195E-9	2.054E-8	<b>3.264E-8</b>	1.005E-8	1.414E-8	<b>3.263E-8</b>
Phase Fit Coefficients						
XTILT	-1.861E-2	9.159E-3	<b>-2.101E-3</b>	-6.255E-3	2.175E-2	<b>-2.101E-3</b>
YTILT	-4.167E-2	-4.577E-2	<b>4.617E-3</b>	-2.387E-2	-2.565E-2	<b>4.617E-3</b>
PISTON	2.041E-3	4.26E-3	<b>3.113E-2</b>	3.595E-3	6.927E-3	<b>3.113E-2</b>
QUAD	-2.329E-3	-2.496E-3	<b>-2.762E-3</b>	-2.312E-3	-2.386E-3	<b>-2.762E-3</b>
COMA X	-4.035E-5	-3.899E-5	<b>-2.781E-5</b>	-3.938E-5	-3.901E-5	<b>-2.781E-5</b>
COMA Y	1.838E-5	3.119E-5	<b>-3.647E-6</b>	1.584E-5	2.329E-5	<b>-3.647E-6</b>
DELX	-9.873E-4	-9.54E-4	<b>-6.804E-4</b>	-9.636E-4	-9.545E-4	<b>-6.804E-4</b>
DELY	4.496E-4	7.827E-4	<b>-8.924E-5</b>	3.877E-4	5.698E-4	<b>-8.924E-5</b>
DELZ	-1.539E-3	-1.65E-3	<b>-1.826E-3</b>	-1.528E-3	-1.577E-3	<b>-1.826E-3</b>
Max. Sol. Phase	3.013	3.102	<b>3.13</b>	3.063	3.093	<b>3.13</b>
Min. Sol. Phase	-3.028	-3.14	<b>-3.059</b>	-2.961	-3.142	<b>-3.059</b>

Estas diferencias aparecen cuando las parejas de mapas de entrada son:

<sup>4</sup> RMS defoc. Residual/Max. Defoc. Amplitude (%)

<sup>5</sup> Sum Square Residual/Infoc. Total Power (%)

- f0r1295bq.map y f1r1295aq.map
- f0r1295bq.map y f3r1295bq.map

sin embargo no ocurren cuando la pareja de entrada es:

- f3r1295bq.map y f1r1295aq.map (columnas en negrita cursiva)

Si se estudia detenidamente el algoritmo que realiza esta función (ver subrutina *point5.f* en el directorio *hololib* de la cuenta *holo* de Siva) se observa que una simple diferencia en el redondeo puede cambiar el resultado del ajuste, debido a que la subrutina toma caminos diferentes de ejecución provocados por dichas diferencias de redondeo.

Estas diferencias afectan muy poco al resultado final (amplitud y fase del campo en la apertura) porque los resultados de *point5.f* se utilizan sólo para generar un modelo inicial de antena que sirve de arranque al programa *picomisty*, y éste debe llegar a resultados similares aunque parta de modelos iniciales ligeramente diferentes.

Se ha comprobado el comportamiento del programa cuando es ejecutado por Siva en condiciones forzadas, es decir, se ha forzado que los valores de XTILT e YTILT sean en Siva los calculados por Isis; en estas condiciones el resultado forzado de Siva es igual al resultado de Isis, lo que indica la bondad del trabajo realizado.

## EL PROGRAMA AVMAP

Este programa permite promediar un número cualquiera de mapas obtenidos con *picomisty* y restarle un mapa de fase (llamada mapa corrector en el programa) si se desea. La finalidad de este mapa corrector es calcular la nueva distribución de fase en la apertura teniendo en cuenta correcciones anteriores sobre el secundario.

### **Modificaciones al código de AVMAP**

Las modificaciones realizadas a *avmap* afectan básicamente a la presentación de resultados gráficos, para la cual se han utilizado, como ya se ha mencionado en otras ocasiones, las librerías de GREG.

### **Localización de los ficheros**

En el directorio *vax\_backup* de la cuenta de holografía existe una copia del fichero fuente, *avmap2.for*, que se utilizaba en el VMS/VAX.

Por otro lado, en el directorio *avmap/program* se encuentran, de modo análogo a los programas anteriores, el nuevo fichero fuente, *avmap.f*, el ejecutable, *avmap*, el fichero *makefile*, el fichero de comandos *avtask* y el fichero de entrada de datos *av\_input.job*, éstos últimos muy útiles para la ejecución automática/desatendida, si bien sólo sirven de ejemplo.

### **Protocolo de compilación y ejecución de AVMAP**

Si se realizasen nuevos cambios al fichero fuente *avmap.f*, el procedimiento de compilación para por ejecutar el fichero *makefile*, tal y como se ha comentado para casos anteriores.

Por su parte, la ejecución de *avmap* no tiene ninguna dificultad y tan sólo expondremos el significado de cada una de las líneas de los ficheros de comandos y de entrada de datos para la ejecución automática.

El fichero de comandos, *avtask*, contiene la siguiente información:

```
/users2/holo/avmap/program/avmap < av_input.job  
rm LOG
```

y debe tener permiso de ejecución, mientras que el de datos, *av\_input.job*, podría ser el siguiente:

```
49.49          !FREQUENCY (GHZ)
```

```

0          !1 FOR NORMALISATION BEFORE ADDITION
0          !1 TO ROTATE AV. PHASE BEFORE ADDITION
av95r12qq  !OUTPUT FILE NAME
f01r1295b  !FIRST MAP
0          !RADIUS FOR SMOOTHING, ZERO FOR NO SMOOTH.
0          !MMAX DEFAULT
f03r1295b  !SECOND MAP
0          !RADIUS FOR SMOOTHING, ZERO FOR NO SMOOTH.
0          !MMAX DEFAULT
f31r1295b  !THIRD MAP
0          !RADIUS FOR SMOOTHING, ZERO FOR NO SMOOTH.
0          !MMAX DEFAULT
0          !FOR NO MORE INPUT MAPS
0          !FOR NO PHASE CORRECTION MAP
6          !NUMBER OF COEFFICIENTS FOR PHASE FIT
1          !FOR AMPLITUDE+PHASE CONTOUR PLOT
2          !FOR AMPLITUDE+PHASE GREY SCALE PLOT
3          !FOR AMPLITUDE+PHASE GREY CONTOUR PLOT
4          !FOR AMPLITUDE+PHASE COLOR PLOT
5          !FOR AMPLITUDE+PHASE 3D PERSPECTIVE PLOT
6          !FOR AMPLITUDE+PHASE .GDF FILES
0          !TO CONTINUE EXECUTION

```

Por tanto, nuevamente, si se opta por la ejecución automática, habremos de copiar los ejemplos de *avtask* y *av\_input.job* que hay en *avmap/program* a nuestro directorio de trabajo y allí editarlos para modificarlos según nuestras necesidades.

## **Visualización de los resultados de AVMAP**

Los resultados de la ejecución de *avmap* son:

- Fichero ASCII *ftn22*: Contiene interesante información y resultados de los cálculos llevados a cabo por *avmap*.
- Ficheros *.ps*: Ficheros PostScript que contienen gráficas de los datos y resultados. Pueden ser visualizados con GhostView o impresos. Estos ficheros son los siguientes:
  - ◆ *avmap\_color.ps*: mapa en color de la solución (en módulo y fase) obtenida por *avmap*,
  - ◆ *avmap\_contgrey.ps*: mapa en contornos de gris de la solución (en módulo y fase) obtenida por *avmap*,
  - ◆ *avmap\_contour.ps*: mapa de líneas de contornos de la solución (en módulo y fase) obtenida por *avmap*,
  - ◆ *avmap\_grey.ps*: mapa en escala de gris de la solución (en módulo y fase) obtenida por *avmap*,
  - ◆ *avmap\_perspec.ps*: mapa en perspectiva 3D de la solución (en módulo y fase) obtenida por *avmap*,
- Ficheros *.gdf*: Para posteriores manipulaciones de los resultados:
  - ◆ *avmap\_amp.gdf*: mapa de amplitud,
  - ◆ *avmap\_pha.gdf*: mapa de fase.

Los resultados de la ejecución de *avmap* sobre varios mapas de campo en la apertura, obtenidos con *picomisty*, se recogen en el Anexo IV de este informe.

## **EL PROGRAMA MAPERROR**

La función de este programa es la comparación del error de fase en la apertura entre dos mapas para obtener la reproducibilidad. Por otro lado, si se ejecuta con un solo mapa de entrada, se obtendrá una medida del error de fase rms en la apertura del reflector.

### **Modificaciones al código de MAPERROR**

De nuevo, las modificaciones afectan a la presentación de resultados gráficos. Además, se ha incluido el cálculo del error rms (ponderado y sin ponderar) en la apertura en micras a partir del dato de la frecuencia de trabajo que no se pedía anteriormente (sólo se calculaba el error rms en radianes) y se genera una gráfica de la variación del error rms en micras en función del radio normalizado de la apertura (antes sólo se proporcionaba error rms en radianes vs. radio normalizado). Estos cálculos sólo se realizan cuando se ejecuta el programa con un solo mapa de entrada.

### **Localización de los ficheros**

En el directorio *vax\_backup* de Siva existe una copia del fichero fuente original que hay en el VMS/VAX: *maperror2.for*. En el subdirectorio *maperror/program* de Siva se encuentran el nuevo fichero fuente, *maperror.f*, el fichero ejecutable, *maperror*, y el fichero *makefile* para su compilación y linkado en caso de cambios.

### **Protocolo de compilación y ejecución de MAPERROR**

La compilación de *maperror* sigue los mismos pasos que para los programas anteriores: utilizar el fichero *makefile*.

Por otro lado, la ejecución no plantea ningún inconveniente porque las preguntas van guiando al usuario. Para utilizar el programa con un solo mapa de entrada se contestará 0 a la petición del segundo mapa.

### **Visualización de los resultados de MAPERROR**

Los resultados generados por *maperror*, recogidos en el Anexo V, son:

- Fichero ASCII *ftn66*, que contiene información interesante sobre los datos de entrada y los cálculos que realiza,
- Ficheros *.ps* con gráficas de los resultados obtenidos que pueden visualizarse con GhostView o imprimirse a través de las impresoras láser. Los ficheros son los siguientes:
  - ◆ *data\_in\_rms.ps*: mapa que recoge los píxeles utilizados en el cálculo de la desviación rms.
  - ◆ *maperror\_color.ps*: mapa en color del resultado de *maperror*,

- ◆ `maperror_contgrey.ps`: mapa de contornos de gris del resultado de `maperror`,
- ◆ `maperror_contour.ps`: mapa de contornos de líneas del resultado de `maperror`,
- ◆ `maperror_grey.ps`: mapa en escala de gris del resultado de `maperror`,
- ◆ `maperror_perspec.ps`: mapa en perspectiva 3D del resultado de `maperror`,
- ◆ `rsmmic_error.ps`: Desviación rms en micras vs. Radio normalizado, (sólo en el caso de un único mapa de entrada).
- ◆ `rmsrad_error.ps`: Desviación rms en radianes vs. Radio normalizado,
- Ficheros `.gdf` para posteriores visualizaciones y manipulaciones con GRAPHIC:
  - ◆ `maperror.gdf`: Mapa resultado de `maperror`.

## PROTOCOLO PARA LA UTILIZACIÓN DEL SOFTWARE DE HOLOGRAFÍA

El siguiente diagrama muestra un ejemplo de aplicación de los programas de reducción y análisis de datos de holografía, que es el que se ha llevado a cabo en el trabajo de trasvase para comprobar la bondad de los nuevos programas.

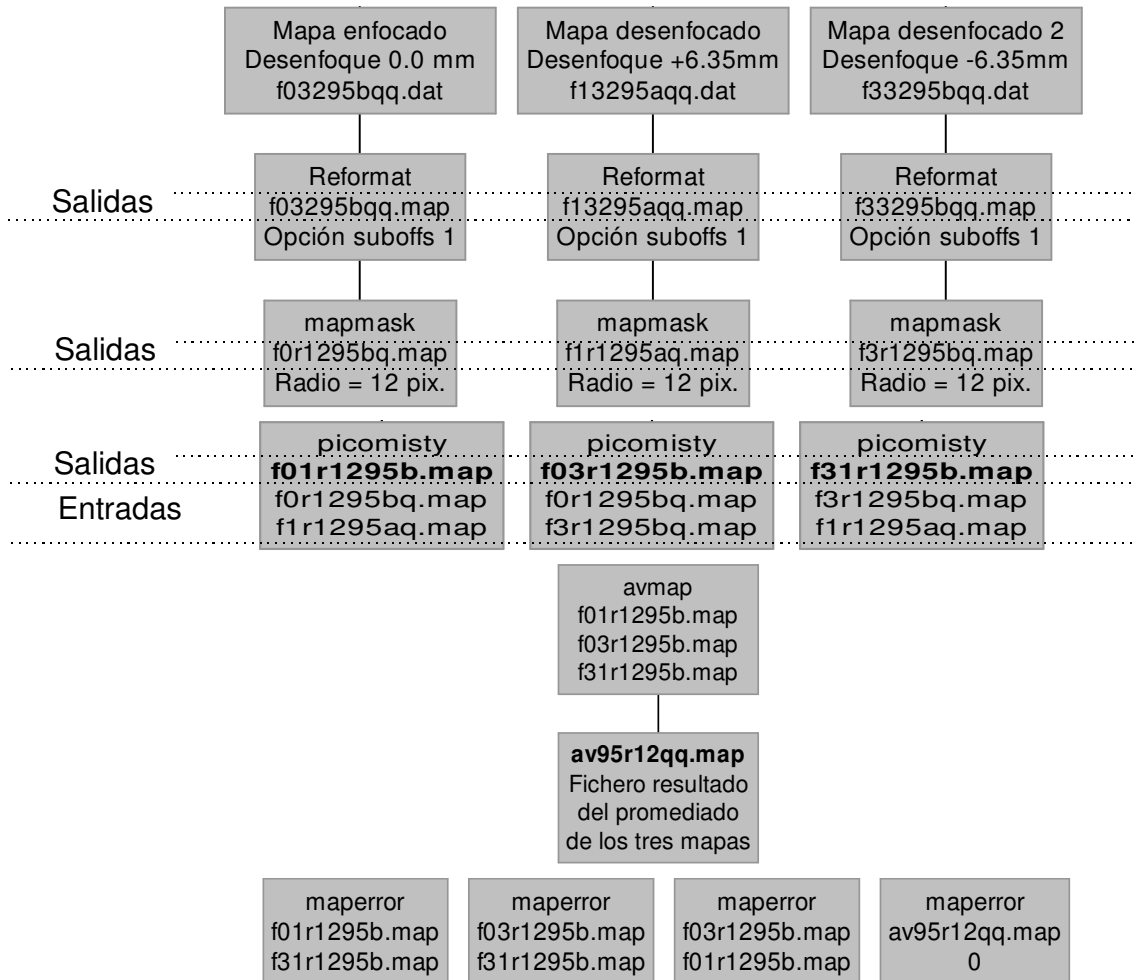


Figura 2: Ejemplo de utilización del software de holografía.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] J. E. Garrido, A. Barcia, P. Planesas, D. Morris: “*Medidas de la superficie del reflector primario del radiotelescopio del CAY mediante técnicas holográficas. I.*”. Informe Técnico IT-CAY 1992-2.
- [2] J. E. Garrido, A. Barcia, D. Morris: “*Medidas de la superficie del reflector primario del radiotelescopio del CAY mediante técnicas holográficas. II.*”. Informe Técnico IT-CAY 1993-5.
- [3] J. E. Garrido, D. Morris, A. Barcia: “*Medidas de la superficie del reflector primario del radiotelescopio del CAY mediante técnicas holográficas. III.*”. Informe Técnico IT-CAY 1994-16.
- [4] J. E. Garrido, A. Barcia, J. M. Páez: “*Medidas de la superficie del reflector primario del radiotelescopio del CAY mediante técnicas holográficas. IV. Correcciones en el secundario.*”. Informe Técnico IT-CAY 1995-9.
- [5] G. Buisson, L. Desbats, G. Duvert, T. Forveille, R. Gras, S. Guilloteau, R. Lucas, P. Valiron: “*GILDAS: Grenoble Image and Line Data Analysis System. Users Manual and Task Reference*”. April 29, 1997. Version 2.1.  
Disponible en <http://iram.fr/GS/gildas.html>
- [6] G. Buisson, L. Desbats, G. Duvert, T. Forveille, R. Gras, S. Guilloteau, R. Lucas, P. Valiron: “*GREG: Grenoble Graphics*”. November 4, 1995.  
Disponible en <http://iram.fr/GS/gildas.html>
- [7] G. Buisson, L. Desbats, G. Duvert, T. Forveille, R. Gras, S. Guilloteau, R. Lucas, P. Valiron: “*SIC: Sympathetic Interpreter of Commands*”. October 24, 1997. Version 7.0.  
Disponible en <http://iram.fr/GS/gildas.html>



# SEPARATA

## INDICE DE CONTENIDO

<b><u>INTRODUCCIÓN</u></b>	<b>26</b>
<b><u>EL PROGRAMA <i>PHASEXTRACT</i></u></b>	<b>27</b>
MODIFICACIONES AL CÓDIGO DE PHASEPROM	27
LOCALIZACIÓN DE LOS FICHEROS	27
PROTOCOLO DE COMPILACIÓN Y EJECUCIÓN DE PHASEPROM	28
VISUALIZACIÓN DE RESULTADOS DE PHASEPROM	29
<b><u>EL PROGRAMA <i>PARCHES</i></u></b>	<b>30</b>
MODIFICACIONES AL CÓDIGO DE PARCHES	30
LOCALIZACIÓN DE LOS FICHEROS	30
PROTOCOLO DE COMPILACIÓN Y EJECUCIÓN DE PARCHES	30
VISUALIZACIÓN DE RESULTADOS DE PARCHES	31
<b><u>EL PROGRAMA <i>PARCHQUANT</i></u></b>	<b>32</b>
MODIFICACIONES AL CÓDIGO DE PARCHQUANT	32
LOCALIZACIÓN DE LOS FICHEROS	32
PROTOCOLO DE COMPILACIÓN Y EJECUCIÓN DE PARCHQUANT	32
VISUALIZACIÓN DE RESULTADOS DE PARCHQUANT	33
<b><u>ANEXOS</u></b>	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>

## INTRODUCCIÓN

Esta separata al informe técnico “*Trasvase del software de holografía de VMS/VAX a HP/UX*” IT-CAY 1998-1, pretende describir la transferencia de nuevos ficheros para la reducción y análisis de datos de holografía, entre ellos: *phasextract*, *parches* y *parchquant*.

Al igual que se hizo en el informe IT-CAY 1998-1, se describirán, para cada uno de los ficheros, los cambios realizados, el procedimiento de compilación, los datos de entrada que necesitan y los resultados de salida que generan.

Por su parte, los anexos recogen los resultados de ejecutar cada uno de los programas conforme se indica en esta separata.

Con estos tres programas se completaría el conjunto de programas necesarios para realizar una conformación de la superficie del subreflector a partir de los datos observacionales de una sesión de holografía.

Por último, aclarar que las referencias que se hagan en este texto están dadas en la página 24 el informe técnico al que pertenece.

## EL PROGRAMA *PHASEXTRACT*

La función del programa *phasextract* es la de generar una tabla con los valores de las coordenadas de cada píxel y la fase correspondiente a dicho píxel.

Existen tres programas diferentes para realizar esta función, a saber:

- *phasextract* [1], lista para todos los píxeles del mapa de entrada el valor de la coordenada X, la coordenada Y y la fase correspondiente (ver en [1] el sentido de la lista y las opciones),
- *phaseint* [4], genera una lista de características similares con la particularidad de que realiza un diezmado reduciendo el número de puntos de la transformada de Fourier de la apertura; así, el número de píxeles a la salida es menor que el número de píxeles del mapa de entrada,
- *phaseprom* [4], análogo al caso anterior con la excepción de que el diezmado se produce promediando píxeles vecinos.

Los ficheros fuente de estos programas (en su versión para VAX) están en el directorio *vax\_backup* de la cuenta *holo* de Siva.

Dado que para la conformación del secundario llevada a cabo en 1995 se utilizó el programa *phaseprom*, se decidió adaptar solamente este programa para su funcionamiento en Siva.

### **Modificaciones al código de *PHASEPROM***

Los cambios han sido mínimos; tan sólo se ha mejorado la presentación del fichero de resultados.

### **Localización de los ficheros**

En el directorio *vax\_backup* de la cuenta *holo* de Siva se encuentra el fichero fuente *phaseprom.for*. Por otro lado, en el directorio *phasextract* de esta misma máquina encontraremos:

- el subdirectorio *program*, que aloja a la nueva versión del código fuente adaptada para su funcionamiento en Siva, el fichero *makefile* para la compilación automática y el fichero ejecutable *phaseprom*.
- el subdirectorio *march94*, que contiene los ficheros de entrada (mapa *av94r12qq.map*, resultado de la sesión de holografía previa a la conformación del secundario), los ficheros de salida, que serán descritos más adelante y el fichero *script* llamado *promtask* para invocar al programa *phaseprom*, residente en el subdirectorio *program*.

## **Protocolo de compilación y ejecución de PHASEPROM**

Para la compilación de phaseprom nos situaremos en el subdirectorio *program* antes descrito y escribiremos:

*\$ make*

La ejecución se llevará a cabo con la ayuda del fichero *script*. Los pasos son los siguientes:

```
User: $ cd /users2/holo/phasextract/march94
User: $ promtask
Siva: FREQUENCY (GHZ) =
User: 49.49
Siva: TELESCOPE DIAMETER (METERS) ?
User: 13.72
Siva: FOCAL LENGTH OF TELESCOPE (METERS) ?
User: 5.08
Siva: INPUT FILE NAME (9 CHARS. MAX.) ?
User: av94r12qq
Siva: MMAX = 25
Siva: OVERWRITE MMAX PLEASE
User: 25
Siva: PICOMIS MAP
Siva: 1 FOR ZERO MASKING OF PHASE OUTSIDE RADIUS MMAX
User: 1
Siva: NO. OF COEFS. FOR PHASE FIT LE.6 ?
User: 6
Siva: mmaX:      12 mcent:      16

Siva: PHASE FIT COEFFS. =
                XTILT=-1.455E-02
                YTILT=-7.775E-02
                PISTON=-5.611E-02
                QUAD= 6.335E-04
                COMA X=-1.077E-05
                COMA Y= 5.191E-05
                DELX=-2.915E-05 MM
                DELY= 1.405E-04 MM
                DELZ= 9.646E-05 MM

Siva: Amp Min= 1.31093E-04
Siva: Amp Max= 4.68625E-02

Siva: Phase Min= -2.87426 at 22 26
Siva: Phase Max= 2.66441 at 16 17
Siva: 1 FOR OUTPUT TABLE WITH COORDS, OTHERWISE JUST PHASE
User: 0
Siva: 1 FOR OUTPUT ON FTN68.DAT OF AMPLITUDE ARRAY
User: 1
```

## Visualización de resultados de PHASEPROM

Los resultados del programa son los siguientes:

- Un fichero *ftn66*, que contiene resultados interesantes de la ejecución del programa:

```
OUTPUT FROM PHASEPROM
```

```
STARTING DATE: 15-Jan-98  
STARTING TIME: 18:48:38
```

```
FREQ. (GHZ) = 49.49  
LAMBDA (um) = 6057.637
```

```
INPUT FILE av94r12qq.map
```

```
MMAX = 25  
NEW MMAX = 25  
PICOMIS MAP
```

```
DATA MASKED TO ZERO OUTSIDE MMAX
```

```
PHASE FIT COEFFS. =  
XTILT=-1.455E-02  
YTILT=-7.775E-02  
PISTON=-5.611E-02  
QUAD= 6.335E-04  
COMA X=-1.077E-05  
COMA Y= 5.191E-05  
DELX=-2.915E-05 MM  
DELY= 1.405E-04 MM  
DELZ= 9.646E-05 MM
```

```
Amp Min= 1.31093E-04  
Amp Max= 4.68625E-02
```

```
phase min= -2.87426 at 22 26  
phase max= 2.66441 at 16 17
```

- Un fichero *ftn67*, que guarda bien una tabla de coordenadas y fase correspondiente si se respondió 1 a la pregunta “1 FOR OUTPUT TABLE WITH COORDS,OTHERWISE JUST PHASE”, o bien sólo la fase si se respondió 0. El fichero *ftn 67* generado con la opción 0 servirá de entrada a los programas *parches* y *parchquant*.
- Un fichero *ftn68* que contiene los datos de amplitud para cada píxel.

## EL PROGRAMA *PARCHES*

Este programa expresa el valor en milímetros del espesor a añadir en la dirección normal a la superficie del subreflector en cada uno de los píxeles en los que se le divide; así se podrá para llevar a cabo la conformación de dicha superficie.

### **Modificaciones al código de *PARCHES***

La mayor modificación consistió en generar junto con el fichero *parches.res* de salida otro llamado *parches.fig*; este fichero es igual que *parches.res* pero sin la cabecera indicativa del significado de cada columna.

Antes, para la presentación de resultados gráficos, había que editar *parches.res*, quitarle la cabecera y guardarlo con el nombre *parches.fig* porque de lo contrario GREG no lo podía interpretar. Como ahora *parches.fig* se genera automáticamente se evita esta tarea de edición y manipulación que sólo nos haría perder tiempo.

### **Localización de los ficheros**

En el directorio *vax\_backup* de la cuenta *holo* de Siva se encuentra la versión para VAX del programa *parches*: *parches.for*.

En el subdirectorio *parches/program* se hallan el nuevo fichero fuente, *parches.f*, el fichero de compilación automática, *makefile*, y el ejecutable, *parches*.

Por otro lado, en el subdirectorio *parches/march94* se encuentra el fichero *ftn67* (resultado de *phaseprom*) que se necesita como entrada y el fichero *script* llamado *parches* que invoca al programa, residente en *parches/program*.

### **Protocolo de compilación y ejecución de *PARCHES***

La compilación se realiza con ayuda del fichero *makefile* como ya se ha descrito otras muchas veces. La ejecución sigue los siguientes pasos:

```
User: $ cd /users2/holo/parches/march94
User: $ parches
Siva: FREQUENCY (GHZ) =
User: 49.49
Siva: Longitud de onda (mm) : 6.05764
Siva: Tamaño pixel ("):
User: 72
Siva: Apertura explorada = 2864.788 lambdas
Siva: Apertura explorada = 17353.85 mm de lado
Siva: Num. de scans (=pixels/scan):
User: 32
```

## Visualización de resultados de PARCHES

Los ficheros de salida del programa *parches* son los siguientes:

- *parches.res*: Fichero de resultados con la tabla que muestra:
  - ◆ L (mm): longitud radial sobre la superficie del hiperboloide,
  - ◆ PHI(grados): ángulo respecto al norte (hacia arriba en el subreflector),
  - ◆ XPL(mm): abscisa,
  - ◆ YPL(mm): ordenada
  - ◆ Corrección (mm): Espesor a añadir.
- *parches.fig*: fichero idéntico a *parches.res* pero sin la cabecera para que pueda ser interpretado por GREG.

Con ayuda de los programas GREG y GRAPHIC se pueden visualizar los resultados del programa *parches*:

- Visualización con GREG: Ejecutaremos un fichero de comandos de GREG, *parches.greg*, que puede ser alterado según nuestras necesidades, para que se represente un diagrama de contornos de la corrección sobre la superficie del secundario. Además, el fichero de comandos genera el fichero *parches.gdf* para la posterior visualización de resultados con GRAPHIC.

```
$ cd /users2/holo/parches/march94
$ greg
Greg> @parches
Greg> hard /plot           !para obtener copia por impresora
Greg> exit                 !para abandonar el programa Greg
```

- Visualización con GRAPHIC: Análogamente se ejecutará un fichero de comandos, *parches.graphic*, lo que permite obtener mapas de color y de grises:

```
$ cd /users2/holo/parches/march94
$ graphic
Graphic> @parches
Graphic> hard /plot       !para obtener copia por impresora
Graphic> exit             !para salir del programa Graphic
```

El Anexo II recoge los resultados gráficos del programa *parches*.

## EL PROGRAMA *PARCHQUANT*

Este programa expresa el valor del espesor a añadir en la dirección normal a la superficie del subreflector en cada uno de los píxeles en los que se le divide. Sin embargo, este programa expresa el espesor como número de capas de un determinado grosor [4].

### **Modificaciones al código de *PARCHQUANT***

La mayor modificación consistió en generar junto con el fichero *parchquant.res* de salida otro llamado *parchquant.fig*; este fichero es igual que *parchquant.res* pero sin la cabecera indicativa del significado de cada columna. Antes, para la presentación de resultados gráficos, había que editar *parchquant.res*, quitarle la cabecera y guardarlo con el nombre *parchquant.fig* porque de lo contrario GREG no lo podía interpretar. Como ahora *parchquant.fig* se genera automáticamente se evita esta tarea de edición y manipulación que sólo nos haría perder tiempo.

### **Localización de los ficheros**

En el directorio *vax\_backup* de la cuenta *holo* de Siva se encuentra la versión para VAX del programa parches: *parchquant.for*. En el subdirectorio *parchquant/program* se hallan el nuevo fichero fuente, *parchquant.f*, el fichero de compilación automática, *makefile*, y el ejecutable, *parchquant*.

Por otro lado, en el subdirectorio *parchquant/march94* se encuentra el fichero *fn67* (resultado de *phaseprom*) que se necesita como entrada y el fichero *script* llamado *parchquant* que invoca al programa, residente en *parchquant/program*.

### **Protocolo de compilación y ejecución de *PARCHQUANT***

La compilación se realiza con ayuda del fichero *makefile* como ya se ha descrito otras muchas veces. La ejecución sigue los siguientes pasos:

```
User: $ cd /users2/holo/parchquant/march94
User: $ parchquant
Siva: FREQUENCY (GHZ) =
User: 49.49
Siva: Longitud de onda (mm) : 6.05764
Siva: Tamaño pixel ("):
User: 72
Siva: Apertura explorada = 2864.788 lambdas
Siva: Apertura explorada = 17353.85 mm de lado
Siva: Num. de scans (=pixels/scan):
User: 32
```



## Visualización de resultados de PARCHQUANT

Los ficheros de salida del programa *parchquant* son los siguientes:

- *parchquant.res*: Fichero de resultados con la tabla que muestra:
  - ◆ L (mm): longitud radial sobre la superficie del hiperboloide,
  - ◆ PHI(grados): ángulo respecto al norte (hacia arriba en el subreflector),
  - ◆ XPL(mm): abscisa,
  - ◆ YPL(mm): ordenada
  - ◆ Número de capas: Espesor a añadir.
- *parchquant.fig*: fichero idéntico a *parchquant.res* pero sin la cabecera para que pueda ser interpretado por GREG.

Con ayuda de los programas GREG y GRAPHIC se pueden visualizar los resultados del programa *parchquant*:

- Visualización con GREG: Ejecutaremos un fichero de comandos de GREG, *parchquant.greg*, que puede ser alterado según nuestras necesidades, para que se represente un diagrama de contornos de la corrección sobre la superficie del secundario. Además, el fichero de comandos genera el fichero *parchquant.gdf* para la posterior visualización de resultados con GRAPHIC.

```
$ cd /users2/holo/parchquant/march94
$ greg
Greg> @parches
Greg> hard /plot           !para obtener copia por impresora
Greg> exit                 !para abandonar el programa Greg
```

- Visualización con GRAPHIC: Análogamente se ejecutará un fichero de comandos, *parchquant.graphic*, lo que permite obtener mapas de color y de grises:

```
$ cd /users2/holo/parchquant/march94
$ graphic
Graphic> @parches
Graphic> hard /plot       !para obtener copia por impresora
Graphic> exit             !para salir del programa Graphic
```

El Anexo III recoge los resultados gráficos del programa *parchquant*.

## ANEXOS

<b><u>ANEXO VI: FICHERO FUENTE REFORMAT.F</u></b>	<b>36</b>
<b><u>ANEXO VII: FICHERO FUENTE MAPMASK.F</u></b>	<b>38</b>
<b><u>ANEXO VIII: FICHERO FUENTE PICOMISTY.F</u></b>	<b>40</b>
<b><u>ANEXO IX: FICHERO FUENTE DE AVMAP.</u></b>	<b>56</b>
<b><u>ANEXO X: FICHERO FUENTE DE MAPERROR.</u></b>	<b>63</b>
<b><u>ANEXO XI: SUBROUTINA ANTMAK4.F</u></b>	<b>69</b>
<b><u>ANEXO XII: SUBROUTINA ASCIICMPLXOUT.F</u></b>	<b>71</b>
<b><u>ANEXO XIII: SUBROUTINA ASCIIOUT.F</u></b>	<b>72</b>
<b><u>ANEXO XIV: SUBROUTINA CARGASCILF</u></b>	<b>74</b>
<b><u>ANEXO XV: SUBROUTINA CLIP.F</u></b>	<b>75</b>
<b><u>ANEXO XVI: SUBROUTINA CPHAS.F</u></b>	<b>76</b>
<b><u>ANEXO XVII: SUBROUTINA DATAOUT.F</u></b>	<b>76</b>
<b><u>ANEXO XVIII: SUBROUTINA DEFOCUS01.F</u></b>	<b>77</b>
<b><u>ANEXO XIX: SUBROUTINA DEFOCUS4.F</u></b>	<b>78</b>
<b><u>ANEXO XX: SUBROUTINA FFT1.F</u></b>	<b>79</b>
<b><u>ANEXO XXI: SUBROUTINA FRESNEL2.F</u></b>	<b>82</b>
<b><u>ANEXO XXII: SUBROUTINA GETMAP3.F</u></b>	<b>82</b>
<b><u>ANEXO XXIII: SUBROUTINA GREG_COLOR.F</u></b>	<b>84</b>
<b><u>ANEXO XXIV: SUBROUTINA GREG_CONTGREY.F</u></b>	<b>90</b>

<b><u>ANEXO XXV: SUBRUTINA GREG_CONTOUR.F</u></b>	<b>96</b>
<b><u>ANEXO XXVI: SUBRUTINA GREG_CURVE.F</u></b>	<b>102</b>
<b><u>ANEXO XXVII: SUBRUTINA GREG_GREY.F</u></b>	<b>104</b>
<b><u>ANEXO XXVIII: SUBRUTINA GREG_PERS.F</u></b>	<b>111</b>
<b><u>ANEXO XXIX: SUBRUTINA LEGBLOCK.F</u></b>	<b>115</b>
<b><u>ANEXO XXX: SUBRUTINA LESSFIT6.F</u></b>	<b>117</b>
<b><u>ANEXO XXXI: SUBRUTINA MAKEGDF.F</u></b>	<b>117</b>
<b><u>ANEXO XXXII: SUBRUTINA MAPIN3.F</u></b>	<b>121</b>
<b><u>ANEXO XXXIII: SUBRUTINA MKFACTOR1.F</u></b>	<b>121</b>
<b><u>ANEXO XXXIV: SUBRUTINA OMISEL5.F</u></b>	<b>123</b>
<b><u>ANEXO XXXV: SUBRUTINA ORGSHFT.F</u></b>	<b>125</b>
<b><u>ANEXO XXXVI: SUBRUTINA PHASFIT6.F</u></b>	<b>126</b>
<b><u>ANEXO XXXVII: SUBRUTINA PHASMXMN.F</u></b>	<b>128</b>
<b><u>ANEXO XXXVIII: SUBRUTINA POINT5.F</u></b>	<b>130</b>
<b><u>ANEXO XXXIX: SUBRUTINA POINTERR5.F</u></b>	<b>132</b>
<b><u>ANEXO XL: SUBRUTINA PRNTOUT.F</u></b>	<b>133</b>
<b><u>ANEXO XLI: SUBRUTINA PSIREAD.F</u></b>	<b>134</b>
<b><u>ANEXO XLII: SUBRUTINA RMASK.F</u></b>	<b>135</b>
<b><u>ANEXO XLIII: SUBRUTINA RMAXMIN.F</u></b>	<b>135</b>
<b><u>ANEXO XLIV: SUBRUTINA SMOOTH.F</u></b>	<b>136</b>
<b><u>ANEXO XLV: SUBRUTINA SMOOTHC.F</u></b>	<b>137</b>
<b><u>ANEXO XLVI: SUBRUTINA TILT.F</u></b>	<b>137</b>

## Anexo VI: Fichero fuente reformat.f

```
PROGRAM REFORMAT
C   version adaptada por Jose Antonio Lopez Perez
C   para su funcionamiento en siva.
C
C   This program reads an ascii file and writes
C   a RGDATA file readable with GREG.
C
C   Input map=filein.DAT
C   Output map=fileout.MAP
C
C   filein and fileout variables cannot have
C   uppercase characters; only lowercase and numbers
C   are allowed.
C
C   Es opcional restar el valor minimo del mapa, lo
C   que implica no necesitar el programa SUBOFFS.
C
C   character lin*80,filein*9,fileout*9
C   character indat*13,outmap*13
C   real rgdata(100,100)
C
1  type *, 'Ascii Data File Name (.dat assumed) ?'
   read(*, '(a9)') filein
   indat=filein//'.dat'
   open(20,file=indat,status='old',readonly,err=1)
2  type *, 'RGDATA File Name without extension ?'
   read(*, '(a9)') fileout
   if(fileout(:1).eq.' ') fileout=filein
   outmap=fileout//'.map'
   open(30,file=outmap,status='new',err=2)
C
   read(20, '(a)', end=99) lin
   read(lin, *) nx
   write(30, '(a)') lin
   read(20, '(a)', end=99) lin
   write(30, '(a)') lin
   read(20, '(a)', end=99) lin
   read(lin, *) ny
   write(30, '(a)') lin
   read(20, '(a)', end=99) lin
   write(30, '(a)') lin
C
   read(20, *, end=200, err=20) ((rgdata(i, j), i=1, nx), j=1, ny)
C
200 write(6, *) '
   write(6, *) 'OPTIONS MENU: '
   write(6, *) '      1 FOR OFFSET SUBSTRUCTION'
   write(6, *) '      0 FOR NO OFFSET SUBSTRUCTION'
   read(5, *) opcion
   if (opcion.eq.0) then
2000 write(30, 2000) ((rgdata(i, j), i=1, nx), j=1, ny)
   format(g15.7e2)
   close(20)
   close(30, status='keep')
   stop 'End of program reformat with no offset substruction'
   else if (opcion.eq.1) then
*****
C Subtract offset and normalise map values to have total power unity
   power=0.0
   minim=17000.0 ! maximum value for the CAY A/D converter is 16383
   do j=1, ny
     do i=1, nx
       power=power+rgdata(i, j)
       if (rgdata(i, j).lt.minim) minim=rgdata(i, j)
```

```

        end do
    end do
    power=power-(minim*(nx*ny))
    write (*,30) power, minim, nx, ny
30   format(' Power = ',f15.7,/, 'Offset = ',f15.7,/, 'X size = ',i3,
1    /, 'Y size = ',i3,/)
c
    do j=1,ny
        do i=1,nx
            rgdata(i,j)=(rgdata(i,j)-minim)/power+1.0e-37
        end do
    end do
c
    write(30,2000) ((rgdata(i,j),i=1,nx),j=1,ny)
    close(20)
    close(30,status='keep')
    stop 'End of program reformat with offset subtraction'
*****
    else
        write(6,*) 'Bad option, try again!'
        goto 200
    end if
C
20   type *,'Error Reading Data from the Input File'
    type *,'We continue processing, anyway.'
    write(30,2000) ((rgdata(i,j),i=1,nx),j=1,ny)
C
99   close(20)
    close(30,status='delete')
    stop 'Error Reading Input File Header'
end

```

## Anexo VII: Fichero fuente mapmask.f

```
Program mapmask
c  Version adaptada por Jose Antonio Lopez Perez
c  para su funcionamiento en Siva.
c  To mask the outer parts of a GREG RGMAP with a circle and output
c  a "smaller map" if needed (not bigger map, be carefull).
c
c  parameter(mdim=128)
c
c  character*9 filein, fileout
c  character*13 infile, outfile
c  character*80 comment1,comment2
c
c  dimension data(mdim,mdim)
c
c  write(6,*) 'Input RGMAP file (.map assumed) ?'
c  read(5,10) filein
c  infile = filein//'.map'
c
c  write(6,*) 'Output RGMAP file please (.map assumed) ?'
c  read(5,10) fileout
10  format(a9)
c  outfile=fileout//'.map'
c
c  open(unit=1,file=infile ,status='old',readonly)
c  open(unit=2,file=outfile,status='new')
c
c  read(1,*) nx,rx,vx,delx
c  read(1,20) comment1
c  read(1,*) ny,ry,vy,dely
c  read(1,20) comment2
20  format(a80)
c
c  write(6,*) 'Input map size = ',nx,' x',ny,' pixels'
c
c  read(1,30) ((data(ix,iy),ix=1,nx),iy=1,ny)
30  format(g15.7e2)
c
c  Apply circular mask
c  write(6,*) 'Radius (pixels) of circular mask please ?'
c  read(5,*) imask
c  if(imask.ne.0) then
c      mdim2=nx/2+1
c      do ix=1,nx
c      do iy=1,ny
c          r2=(ix-mdim2)**2 + (iy-mdim2)**2
c          ir=sqrt(r2)
c          if(ir.gt.imask) data(ix,iy)=0.
c      end do
c      end do
c  end if
c
c  write(6,*) 'Width of output map please ?'
c  read(5,*) noutx
c  write(6,*) 'Height of output map please ?'
c  read(5,*) nouty
c
c  r0=0. !dummy
c  write(2,*) noutx,rx,vx,delx
c  write(2,20) comment1
c  write(2,*) nouty,ry,vy,dely
c  write(2,20) comment2
c
c  istartx=nx/2+1-noutx/2
c  istopx =istartx+noutx-1
```

```
    istory=ny/2+1-nouty/2
    istory =istory+nouty-1
    write(2,30) ((data(ix,iy),ix=istartx,istopx),iy=istory,istory)
c
    close(unit=1)
    close(unit=2,status='keep')
c
    stop ' End of program mapmask.'
end
```

## Anexo VIII: Fichero fuente picomisty.f

```
PROGRAM PICOMISTY
C*****
***
C          COMENTARIOS PREVIOS
C*****
***
C  VERSION DE PICOMISTY ADAPTADA POR JOSE ANTONIO LOPEZ PEREZ
C  PARA SU FUNCIONAMIENTO EN SIVA.
C  VER EL FICHERO MAKEFILE PARA CONOCER LAS SUBROUTINAS QUE UTILIZA
C  Y QUE SE ENCUENTRAN EN /USERS2/HOLO/HOLOBIB/
C
C  PRIMERA VERSION, PARA VAX, DEBIDA A D.MORRIS (IRAM SEPT. 1994)
C
C*****
C          CHANGE FOR DIFFERENT MAP SIZES          MDIM  *
C          FOR DIFFERENT TELESCOPES                LEG45  *
C*****
C
C  FOCUSED AND DEFOCUSSED FAR FIELD PATTERNS ON LUCAS FORMAT
C  GREG '.map' FILES
C
C  EACH CAN BE DEFOCUSSED FROM NOMINAL BEST FOCUS (BY DEFOCUS1,DEFOCUS2)
C
C  ZERO LEVEL OF MAPS CAN BE RESET TO REMOVE DETECTOR OFFSETS
C
C  INPUT DATA REVERSED TO AGREE WITH PICOH2 FORMAT
C  SIGN OF PSI REVERSED
C
C  NO ARRAY OVERFLOWS WITH USE OF OFFSETS
C
C  VERSION TO AVERAGE OVER NREPEAT VALUES OF ISEED
C
C  TRANSMITTER OPTION WITH NEAR-FIELD CORRECTIONS
C  FIRST AND SECOND ORDER TERMS
C
C  POSITION OFFSETS MOFFSET,NOFFSET IN AZM AND ELV
C  POSITIVE VALUES MOVE OUTPUT IMAGE UP AND TO RIGHT
C
C  AUTOMATIC CORRECTION
C  OF ASTIGMATISM,FOCUS AND POINTING ERRORS (BY TILT) IN FIRST GUESS
C
C  INPUT DATA CAN BE SMOOTHED BY GAUSSIAN FUNCTION - RTAPER
C
C  FIRST GUESS CAN HAVE THIN LEGS AT 45 DEGREES, LEG45.EQ.1
C  IF LEG45.EQ.30 THEN MORE DETAILED BLOCKING MODEL FOR 30M TELESCOPE
C
C  PLOTS MAPS OF RESIDUALS IN FINAL AVERAGE
C
C  USE DEFOCUS4 FOR OPTIONAL MASKING OUTSIDE APERTURE FOR VARIABLE (IMASK)
C  NUMBER OF ITERATIONS (IMASK=1 FOR ALL ITERATIONS,=0 FOR NONE)
C
C  OUTPUT LISTING ON FTN66
C
C  LINKED WITH GREG PLOT LIBRARIES TO OBTAIN GRAPHIC RESULTS.
C
C  SOLUTION OUTPUT PSI(M,N) UNFORMATTED FILE ,WITH DIMENSION
C  AND HEADERS
C
C  REMOVES NEGATIVE INPUT VALUES BY TAKING ABS VALUES
C
C  OPTION OF USING AMPLITUDE DATA AS INPUT (IAMP.EQ.1)
C
C  WITH POSSIBILITY TO WRITE A LINE OF COMMENT IN THE HEADER
C
```



```

C      OPTION TO MASK CENTRAL BLOCKING DURING ITERATIONS
C
C*****
C      DEFINICION DE VARIABLES
C*****
C
C      PARAMETER (MDIM=64,MHEADER=29,MHEADER2=32)
C
C      DIMENSION PSI (MDIM,MDIM) ,BLURR (MDIM,MDIM) ,FFELD (MDIM,MDIM)
C      DIMENSION PSIW (MDIM,MDIM)
C      DIMENSION XX (MDIM) ,IBIT (MDIM)
C      DIMENSION FACTOR (MDIM,MDIM)
C      DIMENSION IHEADER (MHEADER) ,IHEADER2 (MHEADER2)
C      DIMENSION PHCOEF (6) ,A (5) ,DELTA (5)
C
C
C      INTEGER*4 ISEED          !for random number generator
C
C      CHARACTER*8 TSTART,TSTARTIT,TSTOPIT
C      CHARACTER*9 DSTART
C      CHARACTER TITLE*64
C      CHARACTER INFOCN*9,DEFOCN*9
C      CHARACTER OMAPNAME*9
C      CHARACTER*116 COMMENT
C
C      COMPLEX PSI,XX
C      COMPLEX CZERO
C      COMPLEX PSIW
C      COMPLEX FACTOR
C
C      LOGICAL L1
C
C      EQUIVALENCE (COMMENT,IHEADER(1))
C
C
C      PARAMETERS FOR ANALYSIS PROGRAM
C
C      MDIM          DIAMETER (PIXELS) OF DATA ARRAYS
C      PSIW          COMPLEX WORKING ARRAY FOR APERTURE DISTRIBUTION
C      PSI           COMPLEX ARRAY FOR APERTURE DISTRIBUTION
C      FFELD        IN FOCUS FAR FIELD AMPLITUDE PATTERN
C      BLURR        DEFOCUSSED FAR FIELD AMPLITUDE PATTERN
C      XX           COMPLEX 1 DIMENSIONAL WORKING ARRAY FOR FFT
C      IBIT         REAL 1 DIMENSIONAL WORKING ARRAY FOR FFT
C      FACTOR       COMPLEX WORKING ARRAY FOR FOCUSING
C      PLTARY       REAL ARRAY FOR DATA TO BE PLOTTED-PART OF PSII.
C
C
C      CONTROL PARAMETERS FOR ANALYSIS
C
C      ITMAX        NUMBER OF ITERATIONS
C      RTAPER       HALF POWER RADIUS(PIXELS) FOR INPUT SMOOTHING OF DATA
C      NPRINT       NO. ITERATIONS BETWEEN PRINT OUT OF CURRENT ERROR ETC.
C      ACC          LIMITING ACCURACY- STOPS IF LESS THAN
C      CNVERG      CONVERGENCE RATE LIMIT- STOPS IF LESS THAN
C      ISEED        SEED OF RANDOM NOS. FOR INITIAL PHASE GUESS
C      NREPEAT      NO OF REPEATS OVER ISEED VALUES WHICH ARE AVERAGED
C      IS          INTERACTIVE STEERING FLAG-SELECTS PLOT OPTION
C      IMASK        IF 1 AMPLITUDE MASKED TO ZERO OUTSIDE APERTURE
C                  IF .GT.1 MASKING IS DONE FOR IMASK ITERATIONS
C      IMASKC      1 TO MASK CENTRAL BLOCKAGE
C      IAMP         1 FOR AMPLITUDE INPUT DATA
C      IRECENT      1 FOR RECENTERING INPUT DATA ACCORDING TO REFERENCE PIXEL
C      IPOINT       1 TO OPTIMIZE POINTING,2 POINTING AND FOCUS,
C                  3 TO OPTIMIZE POINTING+FOCUS+ASTIGMATISM
C
C      PARAMETERS FOR DATA
C

```

```

C      SAMPI      FAR FIELD SAMPLING INTERVAL-UNITS WAVELENGTH/DIAMETER
C      FD         F/D OF PARABOLOID
C      RMAG      MAGNIFICATION OF CASS.(SET INFINITE FOR PRIME FOCUS)
C      MMAX      RADIUS(PIXELS) OF APERTURE-USED FOR MASKING OUTPUT
C              AND APERTURE RADIUS IN FIRST GUESS
C      DELTA1     DEFOCUS IN WAVELENGTHS (POSITIVE AWAY FROM VERTEX)
C      DELTA2
C      DELTA21    RELATIVE DEFOCUS
C
C
C      PARAMETERS FOR FIRST GUESS AT APERTURE DISTRIBUTION
C
C      MOFSTI     OFFSET OF ILLUMINATION PATTERN-NORMALLY ZERO
C      NOFSTI     OFFSET OF ILLUMINATION PATTERN-NORMALLY ZERO
C      NCASSI     RADIUS(PIXELS) OF CENTRAL BLOCKING(-1 FOR NONE)
C      DCASS      DIAMETER OF CASSEGRAIN MIRROR (M)
C      NLEGI      DIAMETER OF LEG(PIXELS) 4 AZ. ELV. (-1 FOR NONE)
C      ASTIGX     ASTIGMATISM (RADIANS) NORMALLY ZERO
C      ASTIGY
C      DEFOCI     DEFOCUS (WAVELENGTHS) NORMALLY ZERO (CAN BE OPTIMIZED)
C      RANPHII    R.M.S. RANDOM PHASE DISTRIBUTION
C      TAPERI     AMPLITUDE TAPER IN DB. E.G. 7.0 FOR 14 DB IN POWER
C      LEG45      IF 1 ALLOWS THIN LEGS AT 45 DEGREES
C      XTILT      X TILT OF FIRST GUESS (FAR FIELD PIXELS)
C      YTILT      Y TILT OF FIRST GUESS (FAR FIELD PIXELS)
C
C
C      LOGICAL UNITS
C      LTEK       INTERACTIVE OUTPUT
C      LREAD      INTERACTIVE INPUT COMMANDS
C      LPRINT     OUTPUT REPORT FILE
C*****
C***
C      DECLARACION DE CONSTANTES
C*****
C***
C
C      PY=3.1415926535
C
C      SET LOGICAL UNITS
C      LTEK=6
C      LREAD=5
C      LPRINT=66
C
C      WRITE(LTEK,*) 'PROGRAM PICOMISTY'
C      WRITE(LTEK,*) ' '
C      WRITE(LTEK,*) 'INTRODUZCA ALGUN COMENTARIO SI LO DESEA'
C      READ(LREAD,25) COMMENT
25  FORMAT(A80)
C      WRITE(LTEK,*) ' '
C
C
C      CALL TIME(TSTART) !TIME OF START
C      CALL DATE(DSTART) !DATE OF START
C
C
C      WRITE(LPRINT,*) 'OUTPUT FROM PICOMISTY'
C
C      WRITE(LPRINT,*) 'STARTING DATE: ',DSTART
C      WRITE(LPRINT,*) 'STARTING TIME: ',TSTART
C      WRITE(LPRINT,*) ' '
C
C      NDIM=MDIM
C      MDIM21=MDIM/2+1
C      MDIM2=MDIM/2
C
C      IRECENT=0          !DON'T RECENTER INPUT DATA
C

```

```

C      INPUT FOR MODEL
      WRITE (LTEK,*) 'TELESCOPE DIAMETER (METERS) ?'
      READ (LREAD,*) DIAM
C
      WRITE (LTEK,*) 'FOCAL LENGTH OF TELESCOPE (METERS) ?'
      READ (LREAD,*) FOC
C
      WRITE (LTEK,*) 'DIAMETER OF CASSEGRAIN SUBREFLECTOR (METERS) ?'
      READ (LREAD,*) DCASS
C
      WRITE (LTEK,*) 'MAGNIFICATION ? (ZERO PRIME FOCUS) '
      READ (LREAD,*) RMAG
      IF (RMAG.EQ.0.) RMAG=2.5E+9
C
      WRITE (LTEK,*) 'INITIAL DEFOCUS (WAVELENGTHS) ?'
      READ (LREAD,*) DEFOCI
      DEFOCI=0.0
C
C
      WRITE (LTEK,*) 'FREQUENCY (GHZ) = '
      READ (LREAD,*) FG
C
      F=FG*1.0E+9
      VELC=2.997925E+8
      RLAMDA=VELC/F
C
      WRITE (LTEK,*) 'TRANSMITTER DISTANCE ? (M) (ZERO FOR FAR FIELD) '
      READ (LREAD,*) ZTX
C
C*****
C      PARAMETROS GEOMETRICOS DE VARIOS TELESCOPIOS
C*****
C      DIAM=13.72    ! (METERS)      !14M TELESCOPE
C      FOC=5.08      !YEBES
C      DCASS=1.085
C      RMAG=11.0
C
C      DIAM=15.0    ! (METERS)      !15M TELESCOPE
C      FOC=4.875    !PLATEAU DE BURE?
C      DCASS=1.55
C      RMAG=
C
C      DIAM=30.0    ! (METERS)      !30M TELESCOPE
C      FOC= 10.5    !PICO VELETA
C      DCASS=2.0
C      RMAG=27.8
C
C      DIAM=19.81   ! (METERS)      !20M TELESCOPE
C      FOC=8.98     !ONSALA
C      DCASS=1.83
C      RMAG=
C*****
      FD=FOC/DIAM
      WRITE (LPRINT,*) '
      WRITE (LPRINT,*) 'DIAM (M) = ',DIAM
      WRITE (LPRINT,*) 'FD RATIO = ',FD
      WRITE (LPRINT,*) 'DCASS (M) = ',DCASS
      WRITE (LPRINT,*) '
C
C
      WRITE (LPRINT,*)
      WRITE (LPRINT,*) 'FREQ. (GHZ) = ',FG
      WRITE (LPRINT,*) 'LAMBDA (mm) = ',RLAMDA*(1E3)
      WRITE (LPRINT,*) 'MAGNIFICATION = ',RMAG
C
C
      IF (ZTX.NE.0.0) THEN
          WRITE (LPRINT,*) 'TRANSMITTER DISTANCE ',ZTX,' Meters'

```

```

ELSE
    WRITE(LPRINT,*) 'TRANSMITTER IN FAR FIELD'
ENDIF
WRITE(LPRINT,*) '          '
C
MOFFSET=0      !TX. OFFSETS PUT TO ZERO
NOFFSET=0
C
IF (MOFFSET.NE.0.OR.NOFFSET.NE.0) THEN
    WRITE(LPRINT,*) 'TX. OFFSET IN AZM ',MOFFSET,' PIXELS'
    WRITE(LPRINT,*) 'TX. OFFSET IN ELV ',NOFFSET,' PIXELS'
    WRITE(LPRINT,*) '          '
END IF
C
C
AUTOMATIC POINTING CORRECTION ?
600 WRITE(LTEK,*) '          '
    WRITE(LTEK,*) 'AUTOMATIC POINTING CORRECTION'
    WRITE(LTEK,*) '  0 FOR NO CORRECTION'
    WRITE(LTEK,*) '  1 FOR POINTING CORRECTION'
    WRITE(LTEK,*) '  2 FOR POINTING+FOCUS CORRECTION'
    WRITE(LTEK,*) '  3 FOR POINTING+FOCUS+ASTIGM. CORRECTION'
    READ(LREAD,*) IPOINT
C
    IF (IPOINT.EQ.1) THEN
        WRITE(LPRINT,*) 'POINTING CORRECTION ON'
    ELSE IF (IPOINT.EQ.2) THEN
        WRITE(LPRINT,*) 'POINTING+FOCUS CORRECTION ON'
    ELSE IF (IPOINT.EQ.3) THEN
        WRITE(LPRINT,*) 'POINTING+FOCUS+ASTIGM. CORRECTION ON'
    ELSE IF (IPOINT.EQ.0) THEN
        WRITE(LPRINT,*) 'POINTING CORRECTION OFF'
    ELSE
        WRITE(LTEK,*) 'BAD OPTION! TRY AGAIN!'
        GOTO 600
    ENDIF
    WRITE(LPRINT,*) '          '
C
C
READ IN TILTS TO APPLY, IF NOT DONE AUTOMATICALLY
C
OR FIRST GUESSES IF DONE AUTOMATICALLY
WRITE(LTEK,*) 'AZIMUTH TILT OF FIRST GUESS (PIXELS)'
READ(LREAD,*) XTILTI
C
WRITE(LTEK,*) 'ELEVATION TILT OF FIRST GUESS (PIXELS)'
READ(LREAD,*) YTILTI
C
IF ((XTILTI.EQ.0.).AND.(YTILTI.EQ.0.)) THEN
    WRITE(LPRINT,*) 'AZ. & EL. TILT FIRST GUESS = 0. (pixels)'
ELSE
    WRITE(LPRINT,*) '  AZIMUTH TILT FIRST GUESS = ',XTILTI,'pixels'
    WRITE(LPRINT,*) '  ELEVATION TILT FIRST GUESS = ',YTILTI,'pixels'
ENDIF
WRITE(LPRINT,*) '          '
C
C
ANTENNA BEAM IN ARC SECS
BEAM=180.*3600.*RLAMDA/DIAM/PY
WRITE(LPRINT,*) 'ANTENNA BEAM (arc secs) = ',BEAM
WRITE(LPRINT,*) '          '
C
C
READ IN MMAX FOR MASKING AND RTAPER TO TAPER INPUT DATA
WRITE(LTEK,870)
870 FORMAT(2X,'MMAX FOR MASKING ? (ZERO FOR DEFAULT)')
READ(LREAD,*) MMAX1
IF (MMAX1.NE.0) MMAX=MMAX1
C
WRITE(LTEK,871)
871 FORMAT(2X,'HALF POWER RADIUS RTAPER (PIXELS) FOR INPUT TAPER',/,
1 2X,' 0. FOR NO SMOOTHING')
READ(LREAD,*) RTAPER

```

```

C
WRITE (LTEK,*) '1 FOR REMOVAL OF OFFSETS IN INPUT MAPS '
READ (LREAD,*) IZERO
C
WRITE (LTEK,*) '1 FOR MASKING OUTSIDE APERTURE'
READ (LREAD,*) IMASK
C
WRITE (LTEK,*) '1 FOR MASKING OF CENTRAL BLOCKAGE'
READ (LREAD,*) IMASKC
C
IF (MMAX1.NE.0) THEN
WRITE (LPRINT,872) MMAX,RTAPER
872 FORMAT (2X,'MMAX=',2X,I6,' FORCED',6X,'RTAPER=',F6.1,/)
ELSE
WRITE (LPRINT,*) 'MMAX DEFAULT,          RTAPER = ',RTAPER
END IF
C
WRITE (LPRINT,*) '          '
WRITE (LPRINT,25) COMMENT
WRITE (LPRINT,*) '          '
C
WRITE (LTEK,*) 'OUTPUT FILE NAME (9 CHARS., .map ASSUMED) ?'
READ (LREAD,150) OMAPNAME
150 FORMAT (A9)
C
WRITE (LPRINT,152) OMAPNAME
152 FORMAT (2X,/,1X,'OUTPUT MAP NAME ',A9,'.map',/)
C
INPUT PARAMETERS FOR SOLUTION
C
ACC=.0001      !LIMITING ACCURACY. STOPS IF LESS THAN
NPRINT=10     !No. ITERAT. BETWEEN PRINT OUT OF CURRENT RESULT
CNVERG=-10.0  !CONVERGENCE RATE LIMIT. STOPS IF LESS THAN
DEFAULT
ITMAX=100     !No OF ITERATIONS
C
601 WRITE (LTEK,*) '1 FOR AMPLITUDE DATA, 0 FOR POWER DATA'
READ (LREAD,*) IAMP
C
IF (IAMP.EQ.1) THEN
WRITE (LPRINT,*) 'INPUT DATA: AMPLITUDES'
ELSE IF (IAMP.EQ.0) THEN
WRITE (LPRINT,*) 'INPUT DATA: POWER'
ELSE
WRITE (LTEK,*) 'BAD OPTION! TRY AGAIN!'
GOTO 601
END IF
WRITE (LPRINT,*) '          '
C
READ IN THE INFOCUS FAR FIELD POWER PATTERN
C
IN "NORMAL" ORDER I.E. ZERO AT CENTRE
C
WRITE (LTEK,*) ' INFOCUS MAP (9 CHARS., .map ASSUMED)?'
READ (LREAD,133) INFOCN
133 FORMAT (A9)
WRITE (LPRINT,134) INFOCN
134 FORMAT (2X,'INFOCUS MAP: ',A9,'.map')
C
CALL MAPIN3 (FFELD,MDIM,INFOCN,NXF,RXF,VXF,DELXF,NYF,RYF,VYF,DELYF,
1 MOFFSET,NOFFSET,IRecent)
IF (DELXF.NE.DELYF) WRITE (LTEK,*) 'X,Y SAMPLING UNEQUAL ',DELXF,DELYF
C
WRITE (LTEK,*) 'DEFOCUS 1 (+ or -) FROM NOMINAL IN MM ?'
READ (LREAD,*) DELTAMM1
DELTA1=DELTAMM1/1000./RLAMDA
WRITE (LPRINT,*) '          '
WRITE (LPRINT,*) ' DEFOCUS 1 = ',DELTAMM1,' millimeters'

```

```

WRITE(LPRINT,*) ' DEFOCUS 1 = ',DELTA1,' wavelengths'
WRITE(LPRINT,*) '          '
C
C   RESET MAP ZERO TO MINIMUM VALUE TO REMOVE DETECTOR OFFSET
C   TAKE ABS VALUE TO AVOID NEGATIVE VALUES OF AMP.
IF (IZERO.EQ.1) THEN
    CALL RMAXMIN (FFELD,MDIM,MDIM,RMAX11,RMIN11)
ENDIF
DO M=1,MDIM
    DO N=1,NDIM
        IF (IZERO.EQ.1) FFELD (M,N)=FFELD (M,N)-RMIN11
        FFELD (M,N)=ABS (FFELD (M,N) )+1.0E-37
    END DO
END DO

C
C
C   CALC SAMPLING INTERVAL FROM MAP DATA HEADER DELXF
SAMPI=ABS (DELXF/BEAM)
WRITE (LTEK,*) ' SAMPI= ',SAMPI
WRITE (LPRINT,*) ' FAR FIELD SAMPLING INTERVAL = ',SAMPI
WRITE (LPRINT,*) ' (in lambda/diameter units)'
WRITE (LTEK,*) '          '
WRITE (LPRINT,*) '          '

C
C   CALCULATE DEFAULT VALUE OF MMAX
IF (MMAX1.EQ.0) THEN
    MMAX=MDIM*SAMPI/2.
ENDIF

C
NMAX=MMAX

C
C
C   READ IN DEFOCUSSED DATA
WRITE (LTEK,*) 'DEFOCUSSED MAP (9 CHARS., .map ASSUMED)?'
READ (LREAD,133) DEFOCN
WRITE (LPRINT,135) DEFOCN
135  FORMAT (2X,'DEFOCUSSED MAP: ',A9,'.map')

C
CALL MAPIN3 (BLURR,MDIM,DEFOCN,NXD,RXD,VXD,DELXD,NYD,RYD,VYD,DELYD,
1 MOFFSET,NOFFSET,IRECENT)
C   RESET ZERO TO REMOVE DETECTOR OFFSET
IF (IZERO.EQ.1) THEN
    CALL RMAXMIN (BLURR,MDIM,MDIM,RMAX12,RMIN12)
ENDIF

C
C   TAKE ABS VALUE TO AVOID NEGATIVE AMPLITUDES
DO M=1,MDIM
    DO N=1,NDIM
        IF (IZERO.EQ.1) BLURR (M,N)=BLURR (M,N)-RMIN12
        BLURR (M,N)=ABS (BLURR (M,N) )+1.0E-37
    END DO
END DO

C
IF (DELXF.NE.DELXD) WRITE (LTEK,*) 'TWO MAPS HAVE UNEQUAL SAMPLING'
1 ,DELXF,DELXD

C
WRITE (LTEK,*) 'DEFOCUS 2 (+ or -) FROM NOMINAL IN MM ?'
READ (LREAD,*) DELTAMM2
DELTA2=DELTAMM2/1000./RLAMDA
WRITE (LPRINT,*) '          '
WRITE (LPRINT,*) ' DEFOCUS 2 = ',DELTAMM2,' millimeters'
WRITE (LPRINT,*) ' DEFOCUS 2 = ',DELTA2,' wavelengths'
WRITE (LPRINT,*) '          '

C
DELTA21=DELTA2-DELTA1      !RELATIVE DEFOCUS (WAVELENGTHS)

C
IF (IMASK.EQ.1) THEN
    WRITE (LPRINT,*) 'IMASK =',IMASK,' MASKED OUTSIDE APERTURE'

```

```

        WRITE (LTEK , *) 'IMASK =', IMASK, ' MASKED OUTSIDE APERTURE '
ELSE
    WRITE (LPRINT, *) 'IMASK =', IMASK, ' NO MASKED OUTSIDE APERTURE '
    WRITE (LTEK , *) 'IMASK =', IMASK, ' NO MASKED OUTSIDE APERTURE '
END IF
WRITE (LPRINT, *) '          '
WRITE (LTEK, *) '          '
C
IF (IMASKC.EQ.1) THEN
    WRITE (LPRINT, *) 'IMASKC =', IMASKC, ' CENTRAL BLOCKAGE MASKED '
    WRITE (LTEK , *) 'IMASKC =', IMASKC, ' CENTRAL BLOCKAGE MASKED '
ELSE
    WRITE (LPRINT, *) 'IMASKC =', IMASKC, ' NO CENTRAL BLOCKAGE MASKED '
    WRITE (LTEK , *) 'IMASKC =', IMASKC, ' NO CENTRAL BLOCKAGE MASKED '
END IF
WRITE (LPRINT, *) '          '
WRITE (LTEK, *) '          '
C
C*****
C    PLOT INPUT DATA: FOCUSSED AND DEFOCUSSED MAPS
C*****
C    PLOT INFOCUS FAR FIELD POWER
    TITLE='INFOCUS FAR FIELD POWER PATTERN '//INFOCN
    CALL RMAXMIN (FFELD, MDIM, NDIM, RMAX3, RMIN3)
    WRITE (LPRINT, *) '          '
    WRITE (LPRINT, 2) TITLE, RMAX3, RMIN3
    WRITE (LTEK, 2) TITLE, RMAX3, RMIN3
    2 FORMAT (1X, A64, /, 9X, 'MAX=', 2X, 1PE10.3, 2X, 'MIN=', 2X, 1PE10.3, /)
C
C    PLOT DEFOCUSSED FAR FIELD POWER PATTERN
    TITLE='DEFOCUSSED FAR FIELD POWER PATTERN'//DEFOCN
    CALL RMAXMIN (BLURR, MDIM, NDIM, RMAX8, RMIN8)
    WRITE (LPRINT, 2) TITLE, RMAX8, RMIN8
    WRITE (LTEK, 2) TITLE, RMAX8, RMIN8
C
C
    CALL LOAD_GREG ('LIBRARY')
602 WRITE (LTEK, *) '          '
    WRITE (LTEK, *) 'PLOT MENU FOR FAR FIELD PATTERN INPUT DATA:'
    WRITE (LTEK, *) '  1 FOR CONTOURS'
    WRITE (LTEK, *) '  2 FOR GREY'
    WRITE (LTEK, *) '  3 FOR GREY-CONTOURS'
    WRITE (LTEK, *) '  4 FOR COLOR'
    WRITE (LTEK, *) '  5 FOR PERSPECTIVE'
    WRITE (LTEK, *) '  ZERO TO CONTINUE EXECUTION'
    READ (LREAD, *) IS
    IF (IS.EQ.1) THEN
        CALL GREG_CONTOUR (FFELD, BLURR, MDIM, 4)
    ELSE IF (IS.EQ.2) THEN
        CALL GREG_GREY (FFELD, BLURR, MDIM, 4)
    ELSE IF (IS.EQ.3) THEN
        CALL GREG_CONTGREY (FFELD, BLURR, MDIM, 4)
    ELSE IF (IS.EQ.4) THEN
        CALL GREG_COLOR (FFELD, BLURR, MDIM, 4)
    ELSE IF (IS.EQ.5) THEN
        CALL GREG_PERS (FFELD, BLURR, MDIM, 4)
    ELSE IF (IS.EQ.0) THEN
        GOTO 603
    ELSE
        WRITE (LTEK, *) 'BAD OPTION. TRY AGAIN!'
    ENDIF
    GOTO 602
603 CONTINUE
C*****
C    END FAR FIELD PLOTS
C*****
    WRITE (LTEK, 78)
    78 FORMAT (2X, 'ITMAX = ?')

```

```

      READ(LREAD,*) ITMAX
C
      WRITE(LTEK,83)
83  FORMAT(2X,'ISEED = ?')
      READ(LREAD,*) ISEED
C
C      NO OF REPEATS OVER RANDOM ISEED VALUES
      WRITE(LTEK,84)
84  FORMAT(2X,'NO OF REPEATS OVER ISEED VALUES ?')
      READ(LREAD,*) NREPEAT
      WRITE(LPRINT,*) 'NO OF REPEATS OVER ISEED VALUES = ',NREPEAT
C
      9  FORMAT(2X,'ACC=',F8.5,2X,'CNVERG=',F8.4,2X,'NPRINT=',I4,2X,'
1  ITMAX=',I4,/,/)
C
      WRITE(LPRINT,9) ACC,CNVERG,NPRINT,ITMAX
      WRITE(LPRINT,85) ISEED
85  FORMAT(2X,'ISEED = ',I6)
      WRITE(LTEK,9) ACC,CNVERG,NPRINT,ITMAX
C
C      CONVERT POWER TO AMPLITUDE IF NEEDED
      IF(IAMP.NE.1) THEN
          DO 55 M=1,MDIM
          DO 55 N=1,NDIM
              FFELD(M,N)=SQRT(FFELD(M,N))
55      BLURR(M,N)=SQRT(BLURR(M,N))
      END IF
C
C      SMOOTH DATA WITH GAUSSIAN,USING PSI AS WORKING ARRAY
      IF(RTAPER.NE.0.) THEN
          CALL SMOOTH(FFELD,PSI,MDIM,RTAPER,XX,IBIT)
          CALL SMOOTH(BLURR,PSI,MDIM,RTAPER,XX,IBIT)
      ENDIF
C
C      REORDER FAR-FIELD MEASUREMENTS IN "NATURAL" ORDER
C      I.E. WITH ZERO AT CORNERS,FOR FFT ALGORITHM
      CALL ORGSHFT(FFELD,MDIM/2,NDIM)
      CALL ORGSHFT(BLURR,MDIM/2,NDIM)
C
C      CLEAR PSI ARRAY
      CZERO=CMPLX(0.,0.)
      DO 115 M=1,MDIM
      DO 115 N=1,NDIM
115  PSI(M,N)=CZERO
C
C*****
C      MAKE INITIAL GUESS AT APERTURE DISTRIBUTION
C*****
      MOFSTI=0          !OFFSET OF ILUMINATION PATTERN
      NOFSTI=0
      RCASSI=DCASS*MDIM*SAMPI/DIAM/2.
      NCASSI=RCASSI          !ACTUAL SECONDARY BLOCKING
      IF(RCASSI.EQ.0.) NCASSI=-1 !NO CASS BLOCKING
      NLEGI=-1             !NO LEG BLOCKING AT 0,90 DEG.
      TAPERI=8.75          !AMPLITUDE TAPER
      ASTIGX=0.0
      ASTIGY=0.0
      RANPHII=0.10
      LEG45=0              !30m LEG BLOCKING AT 45 DEGREES
C
C                          !IF 1 ALLOWS THIN LEGS AT 45 DEG.
C                          !IF 30, THEN MORE DETAILED BLOCKING
C                          !          FOR 30m PICO VELETA
C
C      OPTIMIZE POINTING AND FOCUS AUTOMATICALLY
      IF(IPOINT.NE.0) THEN
          A(1)=XTILTI          !START WITH FIRST GUESS TO TILTS
          A(2)=YTILTI
          A(3)=DEFOCI          !FIRST GUESS TO FOCUS OFFSET

```



```

A(4)=ASTIGX      !FIRST GUESS AT "X" COMPONENT OF ASTIGMATISM
A(5)=ASTIGY      !FIRST GUESS AT "Y" COMPONENT OF ASTIGMATISM
DELTA(1)=0.25    !INITIAL INCREMENT 0.25 PIXELS
DELTA(2)=0.25
DELTA(3)=0.1     !INITIAL INCREMENT OF FOCUS (WAVELENGTHS)
DELTA(4)=0.05   !INITIAL INCREMENT OF "X" ASTIGMATISM (rad)
DELTA(5)=0.05   !INITIAL INCREMENT OF "Y" ASTIGMATISM (rad)
WRITE(LTEK,*)
WRITE(LPRINT,*)
WRITE(LTEK,*) ' POINTING+FOCUS OPTIM. IN DEFOC. PLANE'
WRITE(LPRINT,*) ' POINTING+FOCUS OPTIM. IN DEFOC. PLANE'
WRITE(LTEK,*)
WRITE(LPRINT,*)
RANPHIP=0.       !PHASE DISTRIBUTION FOR POINTING OPTIMIZATION
C OPTIMISE IN 2 STAGES
  DO IOPT=1,2
    CALL POINT5(A,DELTA,PSI,MDIM,NDIM,BLURR,FFELD,XX,
1      IBIT,FD,RMAG,SAMPI,DELTA1,DELTA2,MOFSETI,NOFSETI,MMAX,NMAX,
1      NCASSI,NLEGI,TAPERI,ASTIGX,ASTIGY,DEFOCI,RANPHIP,ISEED,
1      LEG45,ZTX,DELXF,RLAMDA,LTEK,LPRINT,IMASK,DIAM,IPOINT,
1      IMASKC)
    END DO
    XTILT=A(1)
    YTILT=A(2)
    DEFOCI=A(3)
    ASTIGX=A(4)
    ASTIGY=A(5)
  ELSE
    XTILT=XTILTI !IF NO OPTIMISATION OF POINTING
    YTILT=YTILTI
  END IF
  WRITE(LTEK,*)
  WRITE(LPRINT,*)
  WRITE(LTEK,*) ' XTILT= ',XTILT
  WRITE(LTEK,*) ' YTILT= ',YTILT
  WRITE(LTEK,*) ' DEFOCI= ',DEFOCI
  WRITE(LTEK,*) ' ASTIGX= ',ASTIGX
  WRITE(LTEK,*) ' ASTIGY= ',ASTIGY
C
  WRITE(LPRINT,*) ' XTILT= ',XTILT
  WRITE(LPRINT,*) ' YTILT= ',YTILT
  WRITE(LPRINT,*) ' DEFOCI= ',DEFOCI
  WRITE(LPRINT,*) ' ASTIGX= ',ASTIGX
  WRITE(LPRINT,*) ' ASTIGY= ',ASTIGY
C
C WRITE OUT THE INPUT PARAMETERS
LUNIT=LTEK
11 WRITE(LUNIT,23)
13 FORMAT(1X,/,/)
WRITE(LUNIT,12)
12 FORMAT(2X,'INITIAL GUESS FOR SOLUTION',/)
WRITE(LUNIT,13) MOFSTI
13 FORMAT(2X,'MOFST',10X,I8)
WRITE(LUNIT,14) NOFSTI
14 FORMAT(2X,'NOFST',10X,I8)
WRITE(LUNIT,71) XTILT
71 FORMAT(2X,'XTILT',8X,1PE10.3)
WRITE(LUNIT,72) YTILT
72 FORMAT(2X,'YTILT',8X,1PE10.3)
WRITE(LUNIT,15) NCASSI
15 FORMAT(2X,'NCASS',10X,I8)
WRITE(LUNIT,16) NLEGI
16 FORMAT(2X,'NLEG',11X,I8)
WRITE(LUNIT,21) 2.*TAPERI
21 FORMAT(2X,'TAPER',10X,F8.3,2X,'dB')
WRITE(LUNIT,17) ASTIGX
17 FORMAT(2X,'ASTIGX',10X,F8.3)
WRITE(LUNIT,36) ASTIGY

```

```

36 FORMAT (2X, 'ASTIGY', 10X, F8.3)
WRITE (LUNIT, 18) DEFOCI
18 FORMAT (2X, 'DEFOCI', 9X, F8.3)
WRITE (LUNIT, 19) RANPHII
19 FORMAT (2X, 'RANPHI', 9X, F8.3)
WRITE (LUNIT, 32) FD
32 FORMAT (2X, 'F/D', 12X, F8.3)
WRITE (LUNIT, 33) RMAG
33 FORMAT (2X, 'MAG.', 9X, F8.3)
WRITE (LUNIT, 34) MMAX
34 FORMAT (2X, 'MMAX', 13X, I6)
WRITE (LUNIT, 35) LEG45
35 FORMAT (2X, 'LEG45', 12X, I6, /, /)
IF (IZERO.EQ.1) WRITE (LUNIT, *) 'MAP ZEROES RESET'
IF (IZERO.EQ.1) WRITE (LUNIT, *)
IF (LUNIT.EQ.LPRINT) GOTO 20
LUNIT=LPRINT
GOTO 11
20 CONTINUE
C
C
C LEG BLOCKING FOR 30M TELESCOPE (values after increasing diam 1987)
ALPHA=py/4. !ORIENTATION
DLEG1=0.294 !LEG DIAMETER (MAIN UPPER PART)
RLEG1=10. !RADIUS OF INTERSECTION WITH MIRROR
DSHAD=1.218 !SHADOW DIAMETER AT EDGE OF MIRROR
DLEG2=0.654 !DIAMETER OF LOWER PART OF LEG
RLEG2=7.5 !RADIUS AT BEGINNING OF THICKER PART
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C LOOP OVER SOLUTIONS WITH DIFFERENT ISEED VALUES
C
DO 1000 NNR=1, NREPEAT
ISEED=ISEED-1+NNR
C
C
C MAKE INITIAL GUESS WITH RANDOM PHASES
CALL ANTMAK4 (PSIW, MDIM, NDIM, MOFSTI, NOFSTI, MMAX, NMAX, NCASSI, NLEGI,
1 TAPERI, ASTIGX, ASTIGY, DEFOCI, RANPHII, ISEED, LEG45, RMAG, FD)
C MORE REALISTIC LEG BLOCKING FOR 30M TELESCOPE
IF (LEG45.EQ.30) THEN
CALL LEGBLOCK (PSIW, MDIM, SAMPI, DIAM, DLEG1, RLEG1, DSHAD, DLEG2,
1 RLEG2, DCASS, ALPHA)
END IF
C
C TILT APERTURE PLANE PHASE TO ALLOW FOR POINTING ERRORS
CALL TILT (PSIW, MDIM, XTILT, YTILT, 1)
C
C DEFOCUS TO ALLOW FOR OFFSET FROM NOMINAL
II=+1
CALL DEFOCUS01 (PSIW, MDIM, MDIM, FD, RMAG, SAMPI, DELTA1, II, IMASK
1, NCASSI, IMASKC)
C
C DEFOCUS FOR NEAR-FIELD CORRECTION
IF (ZTX.NE.0.) CALL FRESNEL2 (PSIW, MDIM, RLAMDA, -ZTX, DIAM, SAMPI)
C
C TRANSFORM TO FAR FIELD
L1=.TRUE.
CALL FFT1 (PSIW, MDIM, L1, XX, IBIT)
C
WRITE (LTEK, *) 'ISEED = ', ISEED
WRITE (LPRINT, *) 'ISEED = ', ISEED
C*****
C MAKE SOLUTION
C
WRITE (LPRINT, 9) ACC, CNVERG, NPRINT, ITMAX

```

```

WRITE (LTEK, 9) ACC, CNVERG, NPRINT, ITMAX
CALL TIME (TSTARTIT) !FIND TIME OF START OF ITERATIONS
CALL OMISEL5 (PSIW, FACTOR, MDIM, NDIM, BLURR, FFELD, NPRINT, ITMAX
1 , SSQERR, ACC, XX, IBIT, CNVERG, FD, RMAG, SAMPI, DELTA21, LPRINT, LTEK
2 , IMASK, NCASSI, IMASKC)
CALL TIME (TSTOPIT) !FIND TIME OF STOP OF ITERATIONS
WRITE (LPRINT, 336) SSQERR
WRITE (6, 336) SSQERR
336 FORMAT (2X, 'SSQERR=', 1PE10.3, ' PER CENT', /)
C
C*****
C OUTPUT CALCULATION TIME
WRITE (LTEK, *)
WRITE (LPRINT, *)
WRITE (LTEK, 44) TSTARTIT, TSTOPIT
WRITE (LPRINT, 44) TSTARTIT, TSTOPIT
44 FORMAT (1X, 'ITERATION START, STOP TIME ', A8, 4X, A8)
WRITE (LTEK, *)
WRITE (LPRINT, *)
C
C TRANSFORM FAR FIELD SOLUTION TO APERTURE PLANE
L1=.FALSE.
CALL FFT1 (PSIW, MDIM, L1, XX, IBIT)
C
C REFOCUS FOR NEAR-FIELD CORRECTION
IF (ZTX.NE.0.) CALL FRESNEL2 (PSIW, MDIM, RLAMDA, +ZTX, DIAM, SAMPI)
C
C REFOCUS TO NOMINAL FOCUS TO ALLOW FOR OFFSET
II=-1
CALL DEFOCUS01 (PSIW, MDIM, MDIM, FD, RMAG, SAMPI, DELTA1, II, IMASK
1 , NCASSI, IMASKC)
C
C REMOVE INITIAL APERTURE PLANE TILT FROM SOLUTION
CALL TILT (PSIW, MDIM, XTILT, YTILT, -1)
C
C ADD CURRENT SOLUTION INTO PSI ARRAY
C FIRST MAKE SURE AVERAGE PHASE IS ZERO
C FIT COEFFS AND THEN REMOVE 3RD
CALL PHASFIT6 (PSIW, MDIM, MDIM2, MMAX, 1, PHCOEF)
DO 117 N=1, 6
IF (N.EQ.3) GOTO 117
PHCOEF (N)=0.
117 CONTINUE
CALL LESSFIT6 (PSIW, MDIM, MDIM2, PHCOEF)
C NOW ADD INTO PSI ARRAY
DO 110 M=1, MDIM
DO 110 N=1, MDIM
110 PSI (M, N)=PSI (M, N)+PSIW (M, N)
C
C
1000 CONTINUE
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C AVERAGE PSI VALUES AND REVERSE SIGN TO AGREE WITH PICOH2 FORMAT
DO 120 M=1, MDIM
DO 120 N=1, MDIM
120 PSI (M, N)=CONJG (PSI (M, N)) /NREPEAT
C
C OUTPUT RAW MAP ON BINARY 'MAP' FILE WITH HEADERS
IHEADER2 (1)=MMAX
IHEADER2 (2)=1000.*SAMPI
IHEADER2 (3)=MDIM
C LABEL MAP AS COMING FROM PICOMIS--FOR CENTERING OUTLINES
IHEADER2 (4)=1
CALL DATAOUT (PSI, MDIM, NDIM, OMAPNAME, IHEADER, MHEADER
1 , IHEADER2, MHEADER2)
C

```

```

C      CALCULATE RESIDUALS OF FINAL AVERAGE AND PLOT
C      IN BLURR ARRAY
C
C      TRANSFER SOLN. TO WORKING ARRAY PSIW WITH ORIGINAL SIGN
      DO 250 M=1,MDIM
      DO 250 N=1,NDIM
250    PSIW(M,N)=CONJG(PHI(M,N))
C
C      REINSERT APERTURE PLANE TILT FOR CALCULATION OF RESIDUALS
      CALL TILT(PSIW,MDIM,XTILT,YTILT,1)
C
C      DEFOCUS TO DEFOCUS 2 ('DEFOCUSSED PLANE')
      II=1
      CALL DEFOCUS01(PSIW,MDIM,MDIM,FD,RMAG,SAMPI,DELTA2,II,IMASK
1      ,NCASSI,IMASKC)
C
C      NEAR FIELD CORRECTION
      IF(ZTX.NE.0.) CALL FRESNEL2(PSIW,MDIM,RLAMDA,-ZTX,DIAM,SAMPI)
C
C      TRANSFORM TO FAR FIELD
      L1=.TRUE.
      CALL FFT1(PSIW,MDIM,L1,XX,IBIT)
C
C      CALCULATE RESIDUALS,SUM SQUARE ETC.
      RESID2=0.
      TOTPOW=0.
      BLMAX=0.
      DO 251 M=1,MDIM
      DO 251 N=1,NDIM
      TOTPOW=TOTPOW+FFELD(M,N)**2           !TOT POWER IN INFOCUS FAR FIELD
      IF(BLURR(M,N).GT.BLMAX) BLMAX=BLURR(M,N)!MAX.DEFOCUSSED AMPLITUDE
      BLURR(M,N)=BLURR(M,N)-CABS(PSIW(M,N)) !RESIDUAL (AMP)
251    RESID2=RESID2+BLURR(M,N)**2           !RESIDUAL SQUARED
      WRITE(LTEK,*) 'TOTAL POWER INFOCUS FAR FIELD = ',TOTPOW
      WRITE(LPRINT,*) 'TOTAL POWER INFOCUS FAR FIELD = ',TOTPOW
      WRITE(LTEK,*)
      WRITE(LPRINT,*)
C
C      NORMALISE RESIDUAL BY MAX. DEFOCUSSED AMPLITUDE %
      RISID=SQRT(RESID2)*100./BLMAX/MDIM
      WRITE(LTEK,*) 'RMS DEFOC. RESIDUAL/MAX. DEFOC. AMP (%) = ',RISID
      WRITE(LPRINT,*) 'RMS DEFOC. RESIDUAL/MAX. DEFOC. AMP (%) = ',RISID
      WRITE(LTEK,*)
      WRITE(LPRINT,*)
C
C      NORMALISE BY TOTAL POWER IN INFOCUS FAR FIELD
      RESID2=RESID2/TOTPOW
      TOTAMP=SQRT(TOTPOW)
      DO 252 M=1,MDIM
      DO 252 N=1,NDIM
252    BLURR(M,N)=BLURR(M,N)/TOTAMP
C
C      PUT ZERO IN CENTER FOR EASE OF DISPLAY
      CALL ORGSHFT(BLURR,MDIM/2,NDIM)
C
      RESID2PC=RESID2*100.
      WRITE(LTEK,*) 'SUM SQUARE RESIDUAL/INFOCUS TOT.POW (%) = ',RESID2PC
      WRITE(LPRINT,*) 'SUM SQUARE RESIDUAL/INFOCUS TOT.POW (%) = ',RESID2PC
      WRITE(LTEK,*)
      WRITE(LPRINT,*)
C
C      PLOT AMPLITUDE RESIDUALS
      CALL RMAXMIN(BLURR,MDIM,NDIM,RMAX9,RMIN9)
      WRITE(LTEK,*) 'AMP. RESIDUALS MAX= ',RMAX9,' MIN= ',RMIN9
      WRITE(LPRINT,*) 'AMP. RESIDUALS MAX= ',RMAX9,' MIN= ',RMIN9
      WRITE(LTEK,*) 'NORMALISED TO ROOT OF INFOCUS TOTAL POWER'
      WRITE(LPRINT,*) 'NORMALISED TO ROOT OF INFOCUS TOTAL POWER'
      WRITE(LPRINT,*)

```

```

        WRITE (LTEK,*)
C*****
C      AMPL. RES. STORED IN BLURR
606  WRITE (LTEK,*) '
      WRITE (LTEK,*) 'PLOT MENU FOR RESIDUALS:'
      WRITE (LTEK,*) ' 1 FOR CONTOURS'
      WRITE (LTEK,*) ' 2 FOR GREY'
      WRITE (LTEK,*) ' 3 FOR GREY-CONTOURS'
      WRITE (LTEK,*) ' 4 FOR COLOR'
      WRITE (LTEK,*) ' 5 FOR PERSPECTIVE'
      WRITE (LTEK,*) ' 6 FOR GDF'
      WRITE (LTEK,*) ' ZERO TO CONTINUE EXECUTION'
      READ (LREAD,*) IS
      IF (IS.EQ.1) THEN
          CALL GREG_CONTOUR (BLURR, BLURR, MDIM, 3)
      ELSE IF (IS.EQ.2) THEN
          CALL GREG_GREY (BLURR, BLURR, MDIM, 3)
      ELSE IF (IS.EQ.3) THEN
          CALL GREG_CONTGREY (BLURR, BLURR, MDIM, 3)
      ELSE IF (IS.EQ.4) THEN
          CALL GREG_COLOR (BLURR, BLURR, MDIM, 3)
      ELSE IF (IS.EQ.5) THEN
          CALL GREG_PERS (BLURR, BLURR, MDIM, 3)
      ELSE IF (IS.EQ.6) THEN
          CALL MAKEGDF (BLURR, MDIM, NDIM, 5)
      ELSE IF (IS.EQ.0) THEN
          GOTO 607
      ELSE
          WRITE (LTEK,*) 'BAD OPTION. TRY AGAIN!'
      ENDIF
      GOTO 606
607  CONTINUE
C*****
C      PLOT APERTURE PLANE SOLUTION
C
C      FIRST POWER DISTRIBUTION
      DO 5 M=1, MDIM
      DO 5 N=1, NDIM
5     BLURR (M, N) = CABS (PSI (M, N) ) **2
      NNOT=-1
      CALL RMASK (BLURR, MDIM, NDIM, MMAX, NMAX, NNOT, NNOT, 1.0E-7, 0)
      CALL RMAXMIN (BLURR, MDIM, NDIM, RMAX4, RMIN4)
      TITLE='OUTPUT POWER  '//OMAPNAME//' .map'
C
      WRITE (LPRINT, 2) TITLE, RMAX4, RMIN4
      WRITE (LTEK, 2)  TITLE, RMAX4, RMIN4
      CALL RMASK (BLURR, MDIM, NDIM, MMAX, NMAX, NNOT, NNOT, RMIN4, 0)
C*****
C      AMPLITUD STORED IN BLURR
C*****
C      NOW PHASE DISTRIBUTION
C      NO COEFS FOR PHASE FIT
      WRITE (LTEK, 735)
735  FORMAT (2X, 'No. OF COEFF. FOR PHASE FIT (LE.6) ?')
      READ (LREAD,*) ICOEF
C
C      FIT PISTON, TILTS, QUARATIC AND COMA TERMS
      CALL PHASFIT6 (PSI, MDIM, MDIM2, MMAX, 1, PHCOEF)
C      SET TO ZERO COEFFS NOT NEEDED
      IF (ICOEF.GE.6) GOTO 331
      DO 330 IN=ICOEF+1, 6
330  PHCOEF (IN)=0.
331  CONTINUE
      CALL LESSFIT6 (PSI, MDIM, MDIM2, PHCOEF)
C
C      CALC FOCUS OFFSETS
      DELX=PHCOEF (5) *2.*RLAMDA*(FOC*2.*MMAX/DIAM)**3/PY
      DELY=PHCOEF (6) *2.*RLAMDA*(FOC*2.*MMAX/DIAM)**3/PY

```

```

        DELZ=PHCOEF (4) *4. *RLAMDA*( FOC*MMAX/DIAM) **2/PY
C
C   WRITE PHASE FIT RESULTS ON PRINTER
        WRITE (LPRINT, 532) PHCOEF (1) , PHCOEF (2) , PHCOEF (3) , PHCOEF (4) ,
1 PHCOEF (5) , PHCOEF (6) , DELX, DELY, DELZ
532  FORMAT (33X, 'XTILT=' , 1PE10.3, /, 33X, 'YTILT=' , 1PE10.3, /, 33X,
1 'PISTON=' , 1PE10.3, /, 33X, 'QUAD=' , 1PE10.3, /, 33X, 'COMA X=' ,
2 1PE10.3, /, 33X, 'COMA Y=' , 1PE10.3, /, 33X, 'DELX=' , 1PE10.3,
3 /, 33X, 'DELY=' , 1PE10.3, /, 33X, 'DELZ=' , 1PE10.3)
C
        WRITE (LTEK, 532) PHCOEF (1) , PHCOEF (2) , PHCOEF (3) , PHCOEF (4) ,
1 PHCOEF (5) , PHCOEF (6) , DELX, DELY, DELZ
C
C   PLOT OUTPUT PHASE
        DO 6 M=1, MDIM
        DO 6 N=1, NDIM
6 FFELD (M, N) =CPHAS (PSI (M, N) )
C
        CALL RMASK (FFELD, MDIM, NDIM, MMAX, NMAX, NNOT, NNOT, 0., 0)
        CALL RMAXMIN (FFELD, MDIM, NDIM, RMAX5, RMIN5)
C
        TITLE='OUTPUT PHASE  '//OMAPNAME//'.MAP'
        WRITE (LPRINT, 2) TITLE, RMAX5, RMIN5
        WRITE (LTEK, 2)  TITLE, RMAX5, RMIN5
C*****
C   PHASE STORED IN FFELD
C*****
C
604  WRITE (LTEK, *) '
        WRITE (LTEK, *) 'PLOT MENU FOR SOLUTION:'
        WRITE (LTEK, *) ' 1 FOR CONTOURS'
        WRITE (LTEK, *) ' 2 FOR GREY'
        WRITE (LTEK, *) ' 3 FOR GREY-CONTOURS'
        WRITE (LTEK, *) ' 4 FOR COLOR'
        WRITE (LTEK, *) ' 5 FOR PERSPECTIVE'
        WRITE (LTEK, *) ' 6 FOR GDF'
        WRITE (LTEK, *) ' ZERO TO CONTINUE EXECUTION'
        READ (LREAD, *) IS
        IF (IS.EQ.1) THEN
            CALL GREG_CONTOUR (BLURR, FFELD, MDIM, 2)
        ELSE IF (IS.EQ.2) THEN
            CALL GREG_GREY (BLURR, FFELD, MDIM, 2)
        ELSE IF (IS.EQ.3) THEN
            CALL GREG_CONTGREY (BLURR, FFELD, MDIM, 2)
        ELSE IF (IS.EQ.4) THEN
            CALL GREG_COLOR (BLURR, FFELD, MDIM, 2)
        ELSE IF (IS.EQ.5) THEN
            CALL GREG_PERS (BLURR, FFELD, MDIM, 2)
        ELSE IF (IS.EQ.6) THEN
            CALL MAKEGDF (BLURR, MDIM, NDIM, 3)
            CALL MAKEGDF (FFELD, MDIM, NDIM, 4)
        ELSE IF (IS.EQ.0) THEN
            GOTO 605
        ELSE
            WRITE (LTEK, *) 'BAD OPTION. TRY AGAIN!'
        ENDIF
        GOTO 604
605  CONTINUE

        CALL RMASK (FFELD, MDIM, NDIM, MMAX, NMAX, NNOT, NNOT, RMIN5, 0)
C
C   CLIP, MASK FOR PRINTOUT OF PHASE
        C99=99.*PY/180.
        CALL CLIP (FFELD, MDIM, NDIM, -C99, +C99)
        CALL RMASK (FFELD, MDIM, NDIM, MMAX, NMAX, NCASSI, NLEGI, 0., 0)
        CALL PRNTOUT (FFELD, MDIM, NDIM, LPRINT)
C

```

```
WRITE(LTEK,*) 'END OF PROGRAM PICOMISTY.'  
STOP  
END  
C*****  
C                               END OF PROGRAM PICOMISTY  
C*****
```

## Anexo IX: Fichero fuente de avmap.

```
PROGRAM AVMAP
C
C*****
**
C   VERSION DE AVMAP2.FOR ADAPTADA POR JOSE ANTONIO LOPEZ PEREZ
C   PARA SU FUNCIONAMIENTO EN SIVA. SE AGRADECE A JAVIER ALCOLEA
C   SUS EXPLICACIONES SOBRE GREG.
C
C   PRIMERA VERSION PARA VAX DEBIDA A D.MORRIS (IRAM JANUARY 1992)
C
C   TO AVERAGE A SERIES OF "PICOH1.MAP" SOLUTIONS FOR PSI
C   INPUT DATA ON UNFORMATTED BINARY FILES WITH SIZES AND HEADERS
C   PRECEDING. FIRST HEADER COMES DIRECTLY FROM RAW DATA, SECOND
C   FROM PICOH1.
C
C   IHEADER2(4) IS USED TO SENSE MAPS FROM PICOMIS WITH OFFSET
C   CENTERS
C
C   CAN ROTATE AVERAGE PHASE TO ZERO BEFORE ADDITION
C
C   CAN SUBTRACT PHASE CORRECTIONS READ IN FROM A CORRECTION MAP
C
C   CAN SMOOTH OUTPUT MAP WITH GAUSSIAN
C
C   RECYCLES FOR SEVERAL PLOTS
C
C   OUTPUT LISTING ON FTN22
C
C   LINKED WITH GREG PLOT LIBRARIES TO OBTAIN GRAPHIC RESULTS.
C   OPTIONS:      - AVMAP_CONTOUR.PS  (AMPLITUDE AND PHASE)
C                 - AVMAP_GREY.PS    (AMPLITUDE AND PHASE)
C                 - AVMAP_CONTGREY.PS (AMPLITUDE AND PHASE)
C                 - AVMAP_COLOR.PS   (AMPLITUDE AND PHASE)
C                 - AVMAP_PERSPEC.PS  (AMPLITUDE AND PHASE)
C   .PS FILES ARE PHASE CORRECTED BY A PHASE FIT.
C
C   WE ALSO GET THE FOLLOWING FILES:
C                 - AVMAP_AMP.GDF      (AMPLITUDE)
C                 - AVMAP_PHA.GDF      (FASE)
C   .GDF FILES ARE NOT PHASE CORRECTED.
C*****
C   DEFINICION DE VARIABLES
C*****
C
C   PARAMETER (MDIM=64, MHEAD=29, MHEAD2=32)
C
C   DIMENSION PSI (MDIM, MDIM) , BLURR (MDIM, MDIM) , FFELD (MDIM, MDIM)
C   DIMENSION PSIIN (MDIM, MDIM)
C   DIMENSION MODFIELD (MDIM, MDIM) , PHASEFIELD (MDIM, MDIM)
C   DIMENSION IHEADER (MHEAD) , IHEADER2 (MHEAD2)
C   DIMENSION IHEADERC (MHEAD) , IHEADER2C (MHEAD2)
C   DIMENSION PHCOEF (6)
C   DIMENSION XX (MDIM) , IBIT (MDIM)
C
C   CHARACTER TITLE*64
C   CHARACTER COUTFILE*9, CFILE*9
C   CHARACTER ZERO, SPACES4*4
C   CHARACTER*9 DSTART
C   CHARACTER*8 TSTART
C
C   COMPLEX PSI, XX, PSIIN
C
C   REAL MODFIELD, PHASEFIELD
C
```



```

C*****
C
C          DECLARACION DE CONSTANTES
C*****
C
C      PY=3.1415926535
C      ZERO='0'
C      SPACES4=' '
C
C      MDIM21=MDIM/2+1
C
C      LOGICAL UNITS
C      LTEK=6          ! PANTALLA
C      LREAD=5         ! TECLADO
C      LPRINT=22       ! FTN22
C
C      ANTENNA DIMENSIONS (METERS)
C      FOC=5.08        ! DISTANCIA FOCAL EN METROS
C      DIAM=13.72      ! DIAMETRO DEL PRIMARIO EN METROS
C
C      VELC=2.997925E+8 ! SPEED OF LIGHT IN VACUUM
C
C      NDIM=MDIM       ! IGUAL N0. DE PIXELS EN AMBOS EJES
C
C      NCASSI=-1       ! BLOQUEO CENTRAL NULO
C      NLEGI=-1        ! BLOQUEO DE PATAS NULO
C
C*****
C          COMIENZA EL CUERPO DEL PROGRAMA
C*****
C
C      CALL TIME(TSTART) ! HORA DE COMIENZO
C      CALL DATE(DSTART) ! FECHA DE COMIENZO
C
C
C      WRITE(LPRINT,*) 'RESULTS FROM PROGRAM AVMAP64'
C      WRITE(LPRINT,*) 'STARTING DATE: ',DSTART
C      WRITE(LPRINT,*) 'STARTING TIME: ',TSTART
C      WRITE(LPRINT,*) ' '
C
C
C      WRITE(LTEK,*) 'FREQUENCY (GHZ) = '
C      READ(LREAD,*) F
C      WRITE(LPRINT,*) 'FREQ. (GHZ) = ',F
C      RLAMDA=VELC/(F*1E9) !LONGITUD DE ONDA
C
C
C      WRITE(LPRINT,*) 'LAMBDA (mm) = ', (RLAMDA*1E3)
C      WRITE(LPRINT,*) ' '
C
C
C      SHOULD MAPS BE NORMALISED BEFORE ADDING?
672 WRITE(LTEK,222)
222 FORMAT(2X,'1 FOR NORMALISATION BEFORE ADDITION')
READ(LREAD,*) INORM
C
C      IF (INORM.EQ.0) THEN
C          WRITE(LPRINT,*) 'NO NORMALISATION BEFORE ADDITION'
C      ELSE IF (INORM.EQ.1) THEN
C          WRITE(LPRINT,*) 'NORMALISATION BEFORE ADDITION'
C      ELSE
C          WRITE(LTEK,*) 'BAD OPTION! TRY AGAIN!'
C          GOTO 672
C      ENDIF
C
C      SHOULD AVERAGE PHASE BE ROTATED TO ZERO BEFORE AVERAGING?
673 WRITE(LTEK,*) ' 1 TO ROTATE AV. PHASE BEFORE ADDITION'
READ(LREAD,*) IROT
C
C      IF (IROT.EQ.0) THEN
C          WRITE(LPRINT,*) 'NO AVERAGE PHASE SUBSTRACTED'
C      ELSE IF (IROT.EQ.1) THEN
C          WRITE(LPRINT,*) 'AVERAGE PHASE SUBSTRACTED'

```

```

ELSE
    WRITE(LTEK,*) 'BAD OPTION! TRY AGAIN!'
    GOTO 673
ENDIF
C
WRITE(LPRINT,*) '
WRITE(LPRINT,*) 'FILES AVERAGED ARE:'
C
ASK OUTPUT FILE NAME
WRITE(LTEK,*) 'OUTPUT FILE NAME (9 CHARS.)?'
READ(LREAD,667) COUTFILE
667 FORMAT(A9)
C
C
NREADIN=0      ! INICIALIZACION A CERO DEL CONTADOR
                ! DEL NUMERO DE MAPAS A PROMEDIAR
C
C
INICIALIZACION DEL MAPA
DO M=1,MDIM
    DO N=1,NDIM
        PSI(M,N)=CMLPX(0.,0.)
    END DO
END DO
C
C
READ IN REQUIRED FILE NAMES TO BE AVERAGED
100 CONTINUE
WRITE(LTEK,666)
666 FORMAT(2X,'INPUT FILE NAME (9 CHARS.), ZERO FOR END')
READ(LREAD,667) CFILE
IF(CFILE.NE.ZERO) WRITE(LPRINT,665) CFILE
665 FORMAT(20X,A9,'.map')
IF(CFILE.EQ.ZERO) GOTO 103 ! NO HAY MAS FICHEROS DE ENTRADA
C
C
READ IN AND AVERAGE DATA
CALL PSIREAD(PSIIN,MDIM,CFILE,IHEADER,MHEAD,
1 IHEADER2,MHEAD2,NREAD)
IF(NREAD.EQ.0) THEN
    WRITE(LTEK,*) 'TRY ANOTHER FILE PLEASE'
    WRITE(LPRINT,*) 'ERROR: NON VALID FILE'
    GOTO 100
ENDIF
C
NREADIN=NREADIN+1      !INCREMENTA EL CONTADOR PORQUE HA LEIDO UNO MAS
C
C
SHOULD OUTPUT MAP BE SMOOTHED?
WRITE(LTEK,*) 'RADIUS (PIXELS) FOR HALF POWER GAUSSIAN SMOOTHING'
WRITE(LTEK,*) '
                ZERO FOR NO SMOOTHING'
READ(LREAD,*) NTAPER
IF(NTAPER.NE.0) WRITE(LPRINT,*) 'SMOOTHING RADIUS (PIXELS) = ', NTAPER
C
C
MASKING RADIUS
MMAX=IHEADER2(1)
WRITE(LTEK,*) 'MMAX= ',MMAX
WRITE(LTEK,*) 'CHANGE MMAX FOR MASKING ?'
WRITE(LTEK,*) '
                ZERO FOR NO CHANGE'
READ(LREAD,*) MMAX1
IF(MMAX1.NE.0) MMAX=MMAX1
WRITE(LPRINT,*) 'MMAX= ',MMAX
NMAX=MMAX
C
C
ADJUST CENTER FOR MAPS FROM PICOMIS
ICENT=1
IF(IHEADER2(4).NE.0) ICENT=0
IF(IHEADER2(4).NE.0) WRITE(LTEK,*) 'PICOMIS MAP'
IF(IHEADER2(4).NE.0) WRITE(LPRINT,*) 'PICOMIS MAP'
WRITE(LPRINT,*) '
                '
MCENT=MDIM/2+ICENT
C
C
NORMALISE TO UNITY IF NEEDED

```

```

IF (INORM.EQ.1) THEN
  DO M=1,MDIM
    DO N=1,MDIM
      BLURR(M,N)=CABS(PSIIN(M,N))
    END DO
  END DO
  CALL RMAXMIN(BLURR,MDIM,MDIM,RMAX,RMIN)
C
NORMALISE
  DO M=1,MDIM
    DO N=1,MDIM
      PSIIN(M,N)=PSIIN(M,N)/RMAX
    END DO
  END DO
ENDIF

C
C
REMOVE CONSTANT PHASE OFFSET BEFORE ADDITION
IF (IROT.EQ.1) THEN
  CALL PHASFIT6(PSIIN,MDIM,MCENT,MMAX,1,PHCOEF)
C
SET HIGHER TERMS TO ZERO. NOS QUEDAMOS SOLO CON EL TERMINO INDEPENDIENTE
  DO I=1,2
    PHCOEF(I)=0.
  END DO
  DO I=4,6
    PHCOEF(I)=0.
  END DO
  CALL LESSFIT6(PSIIN,MDIM,MCENT,PHCOEF)
ENDIF

C
C
ADD INTO STORAGE ARRAY
DO M=1,MDIM
  DO N=1,NDIM
    PSI(M,N)=PSIIN(M,N)+PSI(M,N)
  END DO
END DO

C
C
CYCLE FOR MORE DATA IF NEEDED
GOTO 100

C
C
NORMALIZE IF ALL DATA HAS BEEN READ AND ADDED
103 DO M=1,MDIM
  DO N=1,NDIM
    PSI(M,N)=PSI(M,N)/(NREADIN)
  END DO
END DO

C
WRITE(LTEK,785) NREADIN
WRITE(LPRINT,*) '          '
WRITE(LPRINT,785) NREADIN
785 FORMAT(2X,I6,' FILES AVERAGED')

C
WRITE(LPRINT,668) COUTFILE
668 FORMAT(1X,'OUTPUT FILE ',A9,'.map')

C
NOW APPLY PHASE CORRECTION MAP IF WANTED
C
READ IN PHASE CORRECTION MAP
674 WRITE(LTEK,*) 'PHASE CORRECTION MAP:'
WRITE(LTEK,*) '          CORRECTION MAP FILE NAME (9 CHARS.)'
WRITE(LTEK,*) '          ZERO FOR NO CORRECTION MAP'
READ(LREAD,667) CFILE

C
IF(CFILE.NE.ZERO) WRITE(LPRINT,801) CFILE
801 FORMAT(2X,'PHASE CORRECTION FILE ',2X,A9,'.map')

C
IF(CFILE.NE.ZERO) THEN
  CALL PSIREAD(PSIIN,MDIM,CFILE,IHEADERC,MHEAD,IHEADER2C,
1 MHEAD2,NREADC)
  IF(NREADC.EQ.0) THEN
    WRITE(LTEK,*) 'TRY ANOTHER FILE PLEASE'

```

```

        WRITE(LPRINT,*) 'ERROR: NON VALID FILE'
        GOTO 674
    ENDIF
C   MAKE PHASE CORRECTIONS
    DO M=1,MDIM
        DO N=1,NDIM
            COR=-CPHAS (PSIIN(M,N) )
            PSI (M,N)=PSI (M,N) *CMLPX (COS (COR) , SIN (COR) )
        END DO
    END DO
ELSE
        WRITE(LPRINT,*) 'NO PHASE CORRECTION MAP APPLIED'
        WRITE(LPRINT,*) ' '
        WRITE(LTEK,*) 'NO PHASE CORRECTION MAP APPLIED'
    ENDIF
C
C   SMOOTH OUTPUT MAP WITH GAUSSIAN IF NECESSARY
    IF(NTAPER.NE.0) CALL SMOOTHC(PSI,MDIM,NTAPER,XX,IBIT)
C
C   NOW OUTPUT AVERAGE PSI ON BINARY FILE .map
C   AND ASCII FILES .txa (AMPLITUD) AND .txp (PHASE)
C
    DO M=1,MDIM
        DO N=1,NDIM
            MODFIELD (M,N)=CABS (PSI (M,N) )
            PHASEFIELD (M,N)=CPHAS (PSI (M,N) )
        END DO
    END DO
C
    CALL DATAOUT (PSI,MDIM,NDIM,COUTFILE,IHEADER,MHEAD,IHEADER2,MHEAD2)
C*****
*   aqui hacemos el mapa de numeros complejos
C*****
    CALL ASCIICMLPXOUT (PSI,MDIM,NDIM,COUTFILE,
1 IHEADER,MHEAD,IHEADER2,MHEAD2)
C
C   PLOT OUTPUT POWER
C   FIRST POWER DISTRIBUTION
    DO M=1,MDIM
        DO N=1,NDIM
            BLURR (M,N)=CABS (PSI (M,N) ) **2 + 1.0E-37
        END DO
    END DO
C
    CALL RMASK (BLURR,MDIM,NDIM,MMAX,NMAX,NCASSI,NLEGI,1.0E-6,ICENT)
    CALL RMAXMIN (BLURR,MDIM,NDIM,RMAX4,RMIN4)
C
    TITLE='OUTPUT POWER'//SPACES4//COUTFILE//' .map'
    WRITE(LPRINT,2) TITLE,RMAX4,RMIN4
    WRITE(LTEK,2) TITLE,RMAX4,RMIN4
2 FORMAT (1X,A64,/,2X,'MAX=',2X,1PE10.3,/,2X,'MIN=',2X,1PE10.3)
C
    CALL RMASK (BLURR,MDIM,NDIM,MMAX,NMAX,NCASSI,NLEGI,RMIN4,ICENT)
C
C   CONVERT TO DB
    PMAX4=10.*ALOG10 (RMAX4)
    DO M=1,MDIM
        DO N=1,NDIM
            BLURR (M,N)=10.*ALOG10 (BLURR (M,N) ) -PMAX4
        END DO
    END DO
C
C*****
C   YA TENEMOS EL MODULO DEL CAMPO EN LA APERTURA EN LA VARIABLE
C   MODFIELD Y LA FASE EN PHASEFIELD.
C   A CONTINUACION CORREGIREMOS LA FASE DE ABERRACIONES Y LA
C   ALMACENAREMOS EN FFELD.
C*****

```

```

C
C   FIT TILT, DEFOCUS AND PRIMARY COMA TERMS TO PHASE DISTRIBUTION
C   WRITE (LTEK, 35)
35  FORMAT (12X, 'No. OF COEFF. FOR PHASE FIT (LE.6) ?')
    READ (LREAD, *) ICOEF
    CALL PHASFIT6 (PSI, MDIM, MCENT, MMAX, 1, PHCOEF)
C   SET COEFFS NOT WANTED TO ZERO
    IF (ICOEF.GE.6) GOTO 31
    DO IN=ICOEF+1, 6
        PHCOEF (IN)=0.
    END DO
31  CONTINUE
    CALL LESSFIT6 (PSI, MDIM, MCENT, PHCOEF)
C
C   PLOT OUTPUT PHASE
C   DO M=1, MDIM
        DO N=1, NDIM
            FFELD (M, N)=CPHAS (PSI (M, N))
        END DO
    END DO
C
C   CALL RMASK (FFELD, MDIM, NDIM, MMAX, NMAX, NCASSI, NLEGI, 0., ICENT)
C   CALL RMAXMIN (FFELD, MDIM, NDIM, RMAX5, RMIN5)
C
C   TITLE='OUTPUT PHASE'//SPACES4//COUTFILE//'.map'
C   WRITE (LPRINT, 2) TITLE, RMAX5, RMIN5
C   WRITE (LTEK, 2) TITLE, RMAX5, RMIN5
C
C   CALCULATE FOCUS OFFSETS
C   DELX=PHCOEF (5) *2.*RLAMDA* (FOC*2.*MMAX/DIAM) **3/PY
C   DELY=PHCOEF (6) *2.*RLAMDA* (FOC*2.*MMAX/DIAM) **3/PY
C   DELZ=PHCOEF (4) *4.*RLAMDA* (FOC*MMAX/DIAM) **2/PY
C
C   WRITE (LPRINT, 32) PHCOEF (1), PHCOEF (2), PHCOEF (3), PHCOEF (4), PHCOEF (5)
C   1, PHCOEF (6), DELX, DELY, DELZ
32  FORMAT (33X, 'XTILT=', 1PE10.3, 2X, '/', 33X, 'YTILT=', 1PE10.3, '/', 33X,
1   'PISTON=', 1PE10.3, '/', 33X, 'QUAD.=', 1PE10.3, '/', 33X, 'COMA X=', 1PE10.3,
2   '/', 33X, 'COMA Y=', 1PE10.3, '/', 33X, 'DELX=', 1PE10.3, '/', 33X, 'DELY=',
3   1PE10.3, '/', 33X, 'DELZ=', 1PE10.3)
C
C*****
C   AHORA ESCRIBIMOS LOS VALORES ASCII, LLAMAMOS A GREG
C   Y HACEMOS LAS GRAFICAS MEDIANTE UN BUCLE PARA QUE EL USUARIO
C   INDIQUE LAS GRAFICAS QUE DESEA.
C*****
C
C   CALL ASCIIOUT (MODFIELD, PHASEFIELD, MDIM, NDIM, COUTFILE, IHEADER,
1   MHEAD, IHEADER2, MHEAD2)
C
C   CALL LOAD_GREG ('LIBRARY')
670 WRITE (LTEK, *) '
    WRITE (LTEK, *) 'PLOT MENU:'
    WRITE (LTEK, *) ' 1 FOR CONTOUR'
    WRITE (LTEK, *) ' 2 FOR GREY'
    WRITE (LTEK, *) ' 3 FOR GREY CONTOURS'
    WRITE (LTEK, *) ' 4 FOR COLOR'
    WRITE (LTEK, *) ' 5 FOR PERSPECTIVE'
    WRITE (LTEK, *) ' 6 FOR GDF'
    WRITE (LTEK, *) ' ZERO TO QUIT'
    READ (LREAD, *) IS
    IF (IS.EQ.1) THEN
        CALL GREG_CONTOUR (MODFIELD, FFELD, MDIM, 1)
    ELSE IF (IS.EQ.2) THEN
        CALL GREG_GREY (MODFIELD, FFELD, MDIM, 1)
    ELSE IF (IS.EQ.3) THEN
        CALL GREG_CONTGREY (MODFIELD, FFELD, MDIM, 1)
    ELSE IF (IS.EQ.4) THEN
        CALL GREG_COLOR (MODFIELD, FFELD, MDIM, 1)

```

```

ELSE IF (IS.EQ.5) THEN
    CALL GREG_PERS (MODFIELD, FFELD, MDIM, 1)
ELSE IF (IS.EQ.6) THEN
    CALL MAKEGDF (MODFIELD, MDIM, NDIM, 1)
    CALL MAKEGDF (FFELD, MDIM, NDIM, 2)
ELSE IF (IS.EQ.0) THEN
    GOTO 671
ELSE
    WRITE (LTEK, *) 'BAD OPTION. TRY AGAIN!'
ENDIF
GOTO 670
671 CONTINUE
C
C CLIP AND MASK BEFORE PRINTOUT
C99=(99.*PY)/180.
CALL CLIP (FFELD, MDIM, NDIM, -C99, +C99)
CALL RMASK (FFELD, MDIM, NDIM, MMAX, NMAX, NCASSI, NLEGI, 0., ICENT)
CALL PRNTOUT (FFELD, MDIM, NDIM, LPRINT)
WRITE (6, *) 'END OF PROGRAM AVMAP64.'
STOP
END
C*****
C                               END OF PROGRAM AVMAP
C*****

```

## Anexo X: Fichero fuente de maperror.

```
PROGRAM MAPERROR
C
C CALCULATE PHASE ERRORS FROM TWO MAPS ON CFILE1 AND CFILE2
C "MAP" FORMAT FROM PICOH1
C
C THIS VERSION READS IHEAD12(4) TO SET MAP ORIGIN FOR MAPS
C MADE BY PICOMIS
C
C PHASE DIFFERENCE MAP IS PLOTTED
C PHASE DIFFERENCES GREATER THEN PHASLIM ARE IGNORED
C
C GIVES GREY DISPLAY OF POINTS INCLUDED IN RMS CALCULATION
C
C GREY SCALE ON LASER PRINTER
C
C GIVES RMS AND WEIGHTED RMS ERROR
C RMS AS FUNCTION OF RADIUS
C
C GIVES EFFICIENCY REDUCTION
C
C LINKED WITH GREG LIBRARIES FOR GRAPHIC OUTPUT
C
C
C PARAMETER (MMDIM=64,MHEAD=29,MHEAD2=32)
C
C DIMENSION PSI1 (MMDIM,MMDIM),PSI2 (MMDIM,MMDIM),AMP (MMDIM,MMDIM)
C DIMENSION PHASE (MMDIM,MMDIM)
C DIMENSION IHEAD1 (MHEAD),IHEAD12 (MHEAD2)
C DIMENSION IHEAD2 (MHEAD),IHEAD22 (MHEAD2)
C DIMENSION PHCOEF1 (6),PHCOEF2 (6)
C DIMENSION NRR (10),RMSR (10),RMEAN (10),RMSMIC (10)
C DIMENSION RMONITOR (MMDIM,MMDIM)
C COMPLEX PSI1,PSI2,FFAMP
C CHARACTER CFILE1*9,CFILE2*9
C CHARACTER ZERO
C CHARACTER*9 DSTART
C CHARACTER*8 TSTART
C
C ZERO='0'
C LTEK=6
C LREAD=5
C LPRINT=66
C VELC=2.997925E+8
C PY=3.1415926535
C
C WRITE (LPRINT,*) 'OUTPUT FROM PROGRAM MAPERROR'
C WRITE (LPRINT,*) '
C CALL TIME (TSTART) ! HORA DE COMIENZO
C CALL DATE (DSTART) ! FECHA DE COMIENZO
C WRITE (LPRINT,*) 'STARTING DATE: ',DSTART
C WRITE (LPRINT,*) 'STARTING TIME: ',TSTART
C WRITE (LPRINT,*) '
C
C READ IN FILE NAMES TO BE READ
C WRITE (LTEK,666)
666 FORMAT (2X,'INPUT FILE NO.1 NAME (9 CHARS., .map assumed)')
C READ (LREAD,667) CFILE1
667 FORMAT (A9)
C WRITE (LPRINT,*) 'INPUT FILES: '
C WRITE (LPRINT,670) CFILE1
670 FORMAT (2X,A9,'.map')
C WRITE (LTEK,668)
668 FORMAT (2X,'INPUT FILE NO.2 NAME (9 CHARS., .map assumed)')
```

```

READ(LREAD,667) CFILE2
WRITE(LPRINT,670) CFILE2
C
C   READ IN DATA
CALL PSIREAD (PSI1,MMDIM,CFILE1,IHEAD1,MHEAD,IHEAD12,
1 MHEAD2,MDIM)
IF (CFILE2.NE.ZERO) THEN
CALL PSIREAD (PSI2,MMDIM,CFILE2,IHEAD2,MHEAD,IHEAD22,
1 MHEAD2,MDIM)
ELSE
WRITE(LTEK,*) 'CALCULATIONS WITH ONE ONLY MAP'
WRITE(LPRINT,*) 'CALCULATIONS WITH ONE ONLY MAP'
WRITE(LPRINT,*) ' '
WRITE(LTEK,*) 'FREQUENCY (GHZ) = '
READ(LREAD,*) F
WRITE(LPRINT,*) 'FREQ. (GHZ) = ',F
RLAMDA=VELC/(F*1E9) !LONGITUD DE ONDA
WRITE(LPRINT,*) 'LAMBDA (um) = ', (RLAMDA*1E6)
WRITE(LPRINT,*) ' '
RAD2MIC=RLAMDA*1E6/(4*PY)
END IF
C
MMAX=IHEAD12(1)
SAMPINT=IHEAD12(2)/1000.
WRITE(LTEK,*)
WRITE(LPRINT,*)
WRITE(LTEK,*) 'MMAX AS READ IN ',MMAX
WRITE(LTEK,*) 'OVERWRITE MMAX PLEASE'
READ(LREAD,*) MMAX
WRITE(LPRINT,*) 'MMAX = ',MMAX
WRITE(LTEK,*) 'SAMPINT = ',SAMPINT
WRITE(LPRINT,*) 'SAMPINT = ',SAMPINT
C
WRITE(LTEK,*) 'PHASE LIMIT ABOVE WHICH VALUES ARE IGNORED?'
READ(LREAD,*) PHASLIM
WRITE(LPRINT,*) 'LIMITING PHASE DIFFERENCE= ',PHASLIM
WRITE(LPRINT,*)
WRITE(LTEK,*)
C
C   ADJUST CENTER FOR MAPS FROM PICOMIS
ICENT=1
IF (IHEAD12(4).NE.0) ICENT=0
IF (ICENT.EQ.0) WRITE(LTEK,*) 'PICOMIS MAP'
IF (ICENT.EQ.0) WRITE(LPRINT,*) 'PICOMIS MAP'
C
M2=MDIM/2+ICENT
N2=M2
C
C   REMOVE ABERRATIONS BEFORE COMPARISONS IF WANTED
WRITE(LTEK,*) '1 FOR SUBTRACTION OF ABERRATIONS (UP TO COMA)'
READ(LREAD,*) IS
IF (IS.EQ.1) THEN
WRITE(LTEK,*) 'No. OF COEFF. FOR PHASE FIT (LE.6) ?'
READ(LREAD,*) ICOEF
WRITE(LPRINT,*) 'NO. OF COEFF FOR LEAST SQUARES: ',ICOEF
C   LEAST SQUARES FIT
IWT=1
CALL PHASFIT6 (PSI1,MDIM,M2,MMAX,IWT,PHCOEF1)
IF (CFILE2.NE.ZERO) CALL PHASFIT6 (PSI2,MDIM,M2,MMAX,1,PHCOEF2)
C   SET COEFFS. NOT WANTED TO ZERO
IF (ICOEF.GE.6) GO TO 31
DO 30 IN=ICOEF+1,6
PHCOEF1(IN)=0.
30 PHCOEF2(IN)=0.
31 CONTINUE
C   SUBTRACT ABERRATIONS
C   WRITE(LPRINT,*) 'COEF'S SUBTRACTED: '
WRITE(LPRINT,32) PHCOEF1(1),PHCOEF1(2),PHCOEF1(3),

```



```

1      PHCOEF1 (4) , PHCOEF1 (5) , PHCOEF1 (6)
      WRITE (LTEK, 32) PHCOEF1 (1) , PHCOEF1 (2) , PHCOEF1 (3) ,
1      PHCOEF1 (4) , PHCOEF1 (5) , PHCOEF1 (6)
      IF (CFILE2.NE.ZERO) THEN
          WRITE (LPRINT, 32) PHCOEF2 (1) , PHCOEF2 (2) ,
1          PHCOEF2 (3) , PHCOEF2 (4) , PHCOEF2 (5) , PHCOEF2 (6)
          WRITE (6, 32) PHCOEF2 (1) , PHCOEF2 (2) ,
1          PHCOEF2 (3) , PHCOEF2 (4) , PHCOEF2 (5) , PHCOEF2 (6)
      END IF
      CALL LESSFIT6 (PSI1, MDIM, M2, PHCOEF1)
      IF (CFILE2.NE.ZERO) CALL LESSFIT6 (PSI2, MDIM, M2, PHCOEF2)
      END IF
C
32  FORMAT (19X, 'XTILT=' , 1PE10.3, 2X, / , 19X, 'YTILT=' , 1PE10.3, / , 19X,
1  'PISTON=' , 1PE10.3, / , 19X, 'QUAD.=' , 1PE10.3, / , 19X, 'COMA X=' , 1PE10.3,
2  / , 19X, 'COMA Y=' , 1PE10.3)
C
C      CAL. MEAN AMP. AND PHASE DIFFERENCE MAPS
C      AND AVERAGE PHASE DIFF.
      AVPHASE=0.
      NPTS=0
      NTOT=0
      DO 3 M=1, MDIM
      DO 3 N=1, MDIM
      NR2= (N-N2) **2
      MR2= (M-M2) **2
      RN2=FLOAT (NR2+MR2)
      RNN=SQRT (RN2)
      NR=SQRT (RN2)
      IF (RNN.GT.FLOAT (MMAX)) GO TO 3
C      IF (NR.GT.MMAX) GO TO 3
      NTOT=NTOT+1
      AMP (M, N) = (CABS (PSI1 (M, N)) + CABS (PSI2 (M, N))) / 2.
      PHASE (M, N) = CPHAS (PSI1 (M, N)) - CPHAS (PSI2 (M, N))
      IF (ABS (PHASE (M, N)) .GT. PHASLIM) GO TO 3
      NPTS=NPTS+1
      AVPHASE=AVPHASE+PHASE (M, N)
3  CONTINUE
      AVPHASE=AVPHASE/NPTS
      WRITE (LTEK, 13) AVPHASE
      WRITE (LPRINT, 13) AVPHASE
13  FORMAT (2X, 'AVPHASE = ' , 1PE10.3)
      WRITE (LPRINT, *)
      WRITE (LTEK, *)
C
C      REMOVE AVERAGE PHASE DIFF.
      DO 4 M=1, MDIM
      DO 4 N=1, MDIM
4  PHASE (M, N) = PHASE (M, N) - AVPHASE
C
C      CALC. RMS AND AMP. WEIGHTED RMS PHASE ERRORS OF AVERAGE MAP
C      CALC. RMS AS FUNCTION OF NORMALISED RADIUS
C      USING POINTS WITHIN RADIUS MMAX ONLY
C      AND THOSE WITH PHASE LESS THAN PHASLIM
      DO 26 I=1, 10
      NRR (I) = 0
      RMEAN (I) = 0.
26  RMSR (I) = 0.
      RMS=0.
      WRMS=0.
      SAMP=0.
      NPTS1=0
      FFAMP=CMPLX (0., 0.)
C      LOOP TO CALCULATE ERRORS
      DO 5 M=1, MDIM
      DO 5 N=1, MDIM
      NR2= (N-N2) **2
      MR2= (M-M2) **2

```

```

RN2=FLOAT(NR2+MR2)
RNN=SQRT(RN2)
NR=SQRT(RN2)
IF(RNN.GT.FLOAT(MMAX)) GO TO 5
IF(ABS(PHASE(M,N)).GT.PHASLIM) GO TO 5
IR=1.+10.*(RNN/FLOAT(MMAX))**2 !INDEX FOR BOXES
IF(IR.GT.10) IR=10
RMONITOR(M,N)=1. !POINT INCLUDED (FOR GREY PLOT)
NPTS1=NPTS1+1
RMS=RMS+PHASE(M,N)**2
WRMS=WRMS+AMP(M,N)*PHASE(M,N)**2
SAMP=SAMP+AMP(M,N)
FFAMP=FFAMP+AMP(M,N)*CPLX(COS(PHASE(M,N)),SIN(PHASE(M,N))) !FAR FIELD
NRR(IR)=NRR(IR)+1
RMEAN(IR)=RMEAN(IR)+(FLOAT(NR)/FLOAT(MMAX))**2
RMSR(IR)=RMSR(IR)+PHASE(M,N)**2
5 CONTINUE

C
EFF=(CABS(FFAMP)/SAMP)**2 !EFFICIENCY REDUCTION

C
C NORMALISE AND SQRT
C ERROR CALCULATED FOR ONE INPUT FILE
WRITE(LPRINT,*)
WRITE(LPRINT,*) 'TOTAL ERROR POWER DIVIDED BY 2'
WRITE(LPRINT,*) 'REFERS TO ONE MAP'
RMS=SQRT(RMS/NPTS1/2.)
WRMS=SQRT(WRMS/SAMP/2.)
DO 20 I=1,10
IF(NRR(I).NE.0) RMEAN(I)=SQRT(RMEAN(I)/NRR(I))
IF(NRR(I).EQ.0) WRITE(LTEK,*) 'NRR',I,' = ZERO'
20 IF(NRR(I).NE.0) RMSR(I)=SQRT(RMSR(I)/NRR(I)/2.)

C
WRITE(LTEK,14) RMS
WRITE(LPRINT,14) RMS
14 FORMAT(19X,'RMS ERROR = ',1PE10.3,' radians')
IF(CFILE2.EQ.ZERO) THEN
WRITE(LTEK,33) RMS*SQRT(2.)*RAD2MIC
WRITE(LPRINT,33) RMS*SQRT(2.)*RAD2MIC
WRITE(LTEK,*)
WRITE(LPRINT,*)
ENDIF
33 FORMAT(19X,'RMS ERROR = ',1PE10.3,' micrometres')

C
WRITE(LTEK,15) WRMS
WRITE(LPRINT,15) WRMS
15 FORMAT(19X,'WEIGHTED = ',1PE10.3,' radians')
IF(CFILE2.EQ.ZERO) THEN
WRITE(LTEK,34) WRMS*SQRT(2.)*RAD2MIC
WRITE(LPRINT,34) WRMS*SQRT(2.)*RAD2MIC
WRITE(LTEK,*)
WRITE(LPRINT,*)
ENDIF
34 FORMAT(19X,'WEIGHTED = ',1PE10.3,' micrometres')

C
WRITE(LTEK,18) EFF
WRITE(LPRINT,18) EFF
18 FORMAT(19X,'EFF.REDUCTION = ',1PE10.3)
WRITE(LTEK,16) NTOT
16 FORMAT(/,19X,'PTS. WITHIN MMAX ',I5)
WRITE(LPRINT,16) NTOT
WRITE(LTEK,17) NPTS1
WRITE(LPRINT,17) NPTS1
17 FORMAT(19X,'No. USED IN RMS ',I5)

C
WRITE(LTEK,*)
WRITE(LPRINT,*)
WRITE(LPRINT,*)
WRITE(LTEK,*) 'RMS AS FUNCTION OF RADIUS'

```

```

WRITE(LPRINT,*) 'RMS AS FUNCTION OF RADIUS'
WRITE(LTEK,23)
WRITE(LPRINT,23)
23  FORMAT(2X,/,7X,'POINTS',4X,'MEAN R',3X,'RMS (rad)')
    DO 21 I=1,10
    WRITE(LPRINT,22) I,NRR(I),RMEAN(I),RMSR(I)
21  WRITE(LTEK,22) I,NRR(I),RMEAN(I),RMSR(I)
22  FORMAT(2X,I2,2X,I5,2X,1PE10.3,2X,1PE10.3)
C
C
CALL LOAD_GREG('LIBRARY')
C
PLOT RMS AS FUNCTION OF RADIUS
WRITE(LTEK,*)
WRITE(LTEK,*) '1 FOR RMS(rad.) vs. RADIUS PLOT'
READ(LREAD,*) IPLOT
IF (IPLOT.EQ.1) CALL GREG_CURVE(RMEAN,RMSR,10,1)
C
IF (CFILE2.EQ.ZERO) THEN
    WRITE(LTEK,*)
    WRITE(LTEK,*) '1 FOR RMS(um) vs. RADIUS PLOT'
    READ(LREAD,*) IPLOT
    IF (IPLOT.EQ.1) THEN
        DO I=1,10
            RMSMIC(I)=RMSR(I)*SQRT(2.)*RAD2MIC
        END DO
    ENDIF
    CALL GREG_CURVE(RMEAN,RMSMIC,10,2)
ENDIF
C
C
PLOT PHASE DIFFERENCE MAP: PHASE(M,N)
606 WRITE(LTEK,*) '
WRITE(LTEK,*) 'PLOT MENU FOR FASE DIFFERENCE MAP:'
WRITE(LTEK,*) ' 1 FOR CONTOURS'
WRITE(LTEK,*) ' 2 FOR GREY'
WRITE(LTEK,*) ' 3 FOR GREY-CONTOURS'
WRITE(LTEK,*) ' 4 FOR COLOR'
WRITE(LTEK,*) ' 5 FOR PERSPECTIVE'
WRITE(LTEK,*) ' 6 FOR GDF'
WRITE(LTEK,*) ' ZERO TO CONTINUE EXECUTION'
READ(LREAD,*) IS
IF (IS.EQ.1) THEN
    CALL GREG_CONTOUR(PHASE,PHASE,MDIM,5)
ELSE IF (IS.EQ.2) THEN
    CALL GREG_GREY(PHASE,PHASE,MDIM,5)
ELSE IF (IS.EQ.3) THEN
    CALL GREG_CONTGREY(PHASE,PHASE,MDIM,5)
ELSE IF (IS.EQ.4) THEN
    CALL GREG_COLOR(PHASE,PHASE,MDIM,5)
ELSE IF (IS.EQ.5) THEN
    CALL GREG_PERS(PHASE,PHASE,MDIM,5)
ELSE IF (IS.EQ.6) THEN
    CALL MAKEGDF(PHASE,MDIM,NDIM,6)
ELSE IF (IS.EQ.0) THEN
    GOTO 607
ELSE
    WRITE(LTEK,*) 'BAD OPTION. TRY AGAIN!'
ENDIF
GOTO 606
607 CONTINUE
C
C
GREY DISPLAY OF POINTS INCLUDED IN ERROR CALCULATION
WRITE(LTEK,*)
WRITE(LTEK,*) '1 FOR GREY DISPLAY OF POINTS INCLUDED'
WRITE(LTEK,*) ' IN ERROR CALCULATION'
READ(LREAD,*) IGREY
IF (IGREY.EQ.1) CALL GREG_GREY(RMONITOR,RMONITOR,MDIM,6)
C
STOP 'END OF PROGRAM MAPERROR.'
```

C            END

## Anexo XI: Subrutina antmak4.f

```
      SUBROUTINE ANTMAK4 (FOUR,NDIM,MDIM,NOFST,MOFST,NMAX,MMAX,NCASS
1 ,NLEG,TAPER,ASTIGX,ASTIGY,DEFOC,RANPHI,IS,LEG45,RMAG,FD)
C
C   TO MAKE MODEL ANTENNA APERTURE
C   OPTION FOR LEGS AT 45 DEGREES,IF LEG45.EQ.1
C
C   EXACT FORM FOR DEFOCUS,DEFOC IN WAVELENGTHS
C
C   ASTIGMATISM EXPRESSED IN RADIANS AS TWO COMPONENTS IN X AND Y
C
      DIMENSION FOUR(NDIM*2,MDIM)
      DATA ISEED/0/
C
C   CALL WITH IS=0 FOR RANDOM PHASE ERRORS,WITH IS NOT 0 FOR RANDOM
C   ERRORS WHICH REPEAT
C
C   INITIALIZE SEED FOR RANDOM NUMBERS
      ISEED=ISEED+1
      IF (ISEED.GT.1) GO TO 345
      CALL SECOND (SECS)
C   WRITE (6,344) SECS
C   WRITE (66,344) SECS
344  FORMAT (2X,'SECS',1PE10.3)
      ISEED1=1000.*SECS+1.
345  CONTINUE
C
C   CONSTANT SEED OPTION FOR SURFACE ERRORS FOR EXAMPLE
      IF (IS.EQ.0) GO TO 450
      ISTORE=ISEED1
      ISEED1=IS
450  CONTINUE
C
      IDIM=2*NDIM
      NDIM2=NDIM/2
      MDIM2=MDIM/2
      RMAG2=RMAG**2
C
C   for NDIM*MDIM complex array
C   antenna radius nmax,mmax <=NDIM/2,MDIM/2
C   blocking radius ncass
C   leg radius nleg,AMPLITUDE taper at nmax=taper db
C
      ELOG=0.4342944819
      SIG2=5.*(NMAX**2)*ELOG/TAPER
      PY=3.1415926535
C
      DO 10 N=1, IDIM-1,2
      DO 10 M=1,MDIM
      X=(N+1)/2-NDIM2-nofst
      Y=M-MDIM2-MOFST
      NR2=( (N+1)/2-NDIM2)**2+(M-MDIM2)**2
      RN2=FLOAT(NR2)
      NR=SQRT(RN2)
C   PHASE CALCULATION
C   ASTIGMATISM HAS MAX ASTIGX AND ASTIGY RADIANS AT RIM
      M1=M-MDIM2
      N1=(N+1)/2-NDIM2
      IF (M1.EQ.0.AND.N1.EQ.0) THEN
      PHIA=0.
      ELSE
      PHIA=(ASTIGX*(M1**2-N1**2)+ASTIGY*2.*M1*N1)/FLOAT(NMAX)
      PHIA=PHIA/FLOAT(NMAX)
      END IF
C   DEFOCUS HAS DEFOC WAVELENGTHS
```

```

RN3=RN2/FLOAT(NMAX)/FLOAT(NMAX)
RN3=RN3/16./(FD**2)
PHIDF=4.*PY*DEFOC*((RN3/(1.+RN3))+(RN3/RMAG2)/(1.+RN3/RMAG2))
C PHIDF=DEFOC*RN2/FLOAT(NMAX)/FLOAT(NMAX)
PHI=PHIA+PHIDF
C UNIFORM RANDOM PHASE 0-RANPHI RADIANS
IF(RANPHI.NE.0.) PHI=PHI+RANPHI*RAN(ISEED1)
C GAUSSIAN TAPER OF -TAPER DB
four(N,M)=100.*EXP(-(X**2+Y**2)/2./SIG2)*COS(PHI)
four(N+1,M)=100.*EXP(-(X**2+Y**2)/2./SIG2)*SIN(PHI)
C LEG BLOCKING
IF(LEG45.EQ.1) THEN !LEGS AT 45 DEGREES
    IF(ABS(M-MDIM2).EQ.ABS((N+1)/2-NDIM2)) THEN
        FOUR(N,M)=0. !MINIMUM THICKNESS
        FOUR(N+1,M)=0.
    END IF
ELSE
    IF((iabs(M-MDIM2)).LE.NLEG) four(N,M)=0.
    IF((IABS(M-MDIM2)).LE.NLEG) FOUR(N+1,M)=0.
    IF(iabs((N+1)/2-NDIM2).LE.NLEG) four(N,M)=0.
    IF(IABS((N+1)/2-NDIM2).LE.NLEG) FOUR(N+1,M)=0.
END IF
C CASS MIRROR BLOCKING
    IF(NR.LE.NCASS) four(N,M)=0.
    IF(NR.LE.NCASS) FOUR(N+1,M)=0.
C EDGE OF DISH
    IF(NR.GT.NMAX) four(N,M)=0.
    IF(NR.GT.NMAX) FOUR(N+1,M)=0.
10 CONTINUE
C RESTORE SEED FOR RANOME NOS.
IF(IS.NE.0) ISEED1=ISTORE
RETURN
END
*****
SUBROUTINE SECOND(SECS)
C DUMMY
RETURN
END
*****

```

## Anexo XII: Subrutina asciicmplxout.f

```
      SUBROUTINE ASCIICMPLXOUT (MAPA, MDIM, NDIM, OUTFILE, IHEADER,
1  MHEADER, IHEADER2, MHEADER2)
C
      COMPLEX MAPA (MDIM, NDIM)
      DIMENSION IHEADER (MHEADER), IHEADER2 (MHEADER2)
      CHARACTER FMAP*13, OUTFILE*9
C
C     OUTPUT MAP FILE NAME
      FMAP=OUTFILE//'.txc'
C     OPEN OUTPUT FILES
      OPEN (UNIT=98, STATUS='NEW', FILE=FMAP, FORM='FORMATTED',
1  ERR=900, IOSTAT=IOS)
C
      NRECORD=1
C
      WRITE (98, *) 'SIZES: MHEADER, MHEADER2, NRECORD, MDIM'
C
      WRITE (98, *) MHEADER
C
      WRITE (98, *) MHEADER2
C
      WRITE (98, *) NRECORD
C
      WRITE (98, *) MDIM
C
      WRITE (98, *) 'HEADER2: IHEADER2 (i) '
C
      DO I=1, MHEADER2
          WRITE (98, *) IHEADER2 (I)
      END DO
C
      WRITE (98, *) 'DATA: '
C
      DO I=1, MDIM
          DO J=1, NDIM
              WRITE (98, FMT=' (G15.7E2) ') MAPA (I, J)
              WRITE (98, FMT=*) '-----'
          END DO
      END DO
C
      CLOSE FILES
      CLOSE (UNIT=98, STATUS='KEEP')
      RETURN
C
C     ERROR HANDLING
900 WRITE (6, *) 'ERROR OPENING OUTPUT .txc FILE', IOS
      RETURN
      END
C
C     Esta subrutina escribe en un fichero ascii los valores
C     RE e IM de un mapa complejo que se le pasa.
```

## Anexo XIII: Subrutina asciout.f

```
      SUBROUTINE ASCIIOUT (MAPAA, MAPAP, MDIM, NDIM, OUTFILE, IHEADER,
1  MHEADER, IHEADER2, MHEADER2)
C
      REAL MAPAA (MDIM, NDIM), MAPAP (MDIM, NDIM)
      DIMENSION IHEADER (MHEADER), IHEADER2 (MHEADER2)
      CHARACTER FMAPA*13, FMAPP*13, OUTFILE*9
C
C     OUTPUT MAP FILE NAMES
      FMAPA=OUTFILE//'.txa'
      FMAPP=OUTFILE//'.txp'
C
      OPEN OUTPUT FILES
      OPEN (UNIT=98, STATUS='NEW', FILE=FMAPA, FORM='FORMATTED',
1  ERR=900, IOSTAT=IOS)
      OPEN (UNIT=97, STATUS='NEW', FILE=FMAPP, FORM='FORMATTED',
1  ERR=901, IOSTAT=IOS)
C
      NRECORD=1
C
      WRITE (98, *) 'SIZES: MHEADER, MHEADER2, NRECORD, MDIM'
      WRITE (97, *) 'SIZES: MHEADER, MHEADER2, NRECORD, MDIM'
C
      WRITE (98, *) MHEADER
      WRITE (97, *) MHEADER
C
      WRITE (98, *) MHEADER2
      WRITE (97, *) MHEADER2
C
      WRITE (98, *) NRECORD
      WRITE (97, *) NRECORD
C
      WRITE (98, *) MDIM
      WRITE (97, *) MDIM
C
      WRITE (98, *) 'HEADER2: IHEADER2 (i) '
      WRITE (97, *) 'HEADER2: IHEADER2 (i) '
C
      DO I=1, MHEADER2
          WRITE (98, *) IHEADER2 (I)
          WRITE (97, *) IHEADER2 (I)
      END DO
C
      WRITE (98, *) 'DATA: '
      WRITE (97, *) 'DATA: '
C
      DO I=1, MDIM
          DO J=1, NDIM
              WRITE (98, FMT=' (G15.7E2) ') MAPAA (I, J)
              WRITE (97, FMT=' (G15.7E2) ') MAPAP (I, J)
          END DO
      END DO
C
      CLOSE FILES
      CLOSE (UNIT=98, STATUS='KEEP')
      CLOSE (UNIT=97, STATUS='KEEP')
      RETURN
C
      ERROR HANDLING
900  WRITE (6, *) 'ERROR OPENING OUTPUT .txa FILE', IOS
      RETURN
901  WRITE (6, *) 'ERROR OPENING OUTPUT .txp FILE', IOS
      RETURN
      END
C
      Esta subrutina escribe en dos ficheros ascii los valores
```



C de los mapas que se le pasan.

## Anexo XIV: Subrutina cargascii.f

```
      SUBROUTINE CARGASCII (PSIIN, MDIM, NDIM, INFILE, IHEADER, MHEADER,  
1  IHEADER2, MHEADER2, LEIDO)  
C  
C      LEEREMOS LOS FICHEROS .TXA Y .TXP Y GENERAMOS EL MAPA COMPLEJO  
C      PSIIN  
C  
      DIMENSION IHEADER (MHEADER)  
      DIMENSION IHEADER2 (MHEADER2)  
      DIMENSION IHEADER2A (MHEADER2)  
      DIMENSION IHEADER2P (MHEADER2)  
      COMPLEX PSIIN (MDIM, MDIM)  
      REAL*4 AMPLI, PHASE  
      CHARACTER INFILE*9  
      CHARACTER AFILE*13, PFILE*13  
      CHARACTER*37 SIZETITLE  
      CHARACTER*20 HEADER2TITLE  
      CHARACTER*6 DATATITLE  
      LOGICAL LEIDO  
C  
      AFILE=INFILE//'.TXA'  
      PFILE=INFILE//'.TXP'  
C  
      OPEN (UNIT=30, STATUS='OLD', FILE=AFILE, FORM='FORMATTED',  
1  ERR=900, IOSTAT=IOS, READONLY, ACCESS='SEQUENTIAL')  
C  
      OPEN (UNIT=31, STATUS='OLD', FILE=PFILE, FORM='FORMATTED',  
1  ERR=901, IOSTAT=IOS, READONLY, ACCESS='SEQUENTIAL')  
C  
      READ (30, FMT=' (A37) ') SIZETITLE  
      READ (31, FMT=' (A37) ') SIZETITLE  
  
      READ (30, FMT=' (I12) ') MHEADERA  
      READ (30, FMT=' (I12) ') MHEADER2A  
      READ (30, FMT=' (I12) ') NRECORDA  
      READ (30, FMT=' (I12) ') MDIMA  
C  
      READ (31, FMT=' (I12) ') MHEADERP  
      READ (31, FMT=' (I12) ') MHEADER2P  
      READ (31, FMT=' (I12) ') NRECORDP  
      READ (31, FMT=' (I12) ') MDIMP  
C  
      IF (MHEADERA.NE.MHEADERP) THEN  
          WRITE (6, *) 'ERROR: MHEADER .TXA # MHEADER .TXP '  
          CLOSE (30, STATUS='KEEP')  
          CLOSE (31, STATUS='KEEP')  
          RETURN  
      ELSE IF (MHEADER2A.NE.MHEADER2P) THEN  
          WRITE (6, *) 'ERROR: MHEADER2 .TXA # MHEADER2 .TXP '  
          CLOSE (30, STATUS='KEEP')  
          CLOSE (31, STATUS='KEEP')  
          RETURN  
      ELSE IF (NRECORDA.NE.NRECORDP) THEN  
          WRITE (6, *) 'ERROR: NRECORD .TXA # NRECORD .TXP '  
          CLOSE (30, STATUS='KEEP')  
          CLOSE (31, STATUS='KEEP')  
          RETURN  
      ELSE IF (MDIMA.NE.MDIMP) THEN  
          WRITE (6, *) 'ERROR: MDIM .TXA # MDIM .TXP '  
          CLOSE (30, STATUS='KEEP')  
          CLOSE (31, STATUS='KEEP')  
          RETURN  
      END IF  
C  
      MHEADER=MHEADERA
```

```

MHEADER2=MHEADER2A
NRECORD=NRECORDA
MDIM=MDIMA
C
DO I=1,MHEADER
    IHEADER(I)=0
END DO
C
READ(30,FMT='(A20)') HEADER2TITLE
READ(31,FMT='(A20)') HEADER2TITLE
C
DO I=1,MHEADER2
    READ(30,FMT='(I12)') IHEADER2A(I)
    READ(31,FMT='(I12)') IHEADER2P(I)
    IF (IHEADER2A(I).NE.IHEADER2P(I)) THEN
        WRITE(6,*) 'ERROR: IHEADER2 .TXA # IHEADER2 .TXP EN I=',I
        CLOSE(30,STATUS='KEEP')
        CLOSE(31,STATUS='KEEP')
        RETURN
    ELSE
        IHEADER2(I)=IHEADER2A(I)
    ENDIF
END DO
READ(30,FMT='(A6)') DATATITLE
READ(31,FMT='(A6)') DATATITLE
DO I=1,MDIM
    DO J=1,NDIM
        READ(30,FMT='(G15.7E2)',ERR=902,IOSTAT=IOS) AMPLI
        READ(31,FMT='(G15.7E2)',ERR=903,IOSTAT=IOS) PHASE
        PSIIIN(I,J)=CMPLX(AMPLI*COS(PHASE),AMPLI*SIN(PHASE))
    END DO
END DO
C
CLOSE(30,STATUS='KEEP')
CLOSE(31,STATUS='KEEP')
LEIDO=.TRUE.
RETURN
C
900 WRITE(6,*) 'CARGA_ASCII ERROR OPENING .TXA FILE',IOS
GOTO 906
901 WRITE(6,*) 'CARGA_ASCII ERROR OPENING .TXP FILE',IOS
GOTO 906
902 WRITE(6,*) 'CARGA_ASCII ERROR READING .TXA FILE',IOS
GOTO 906
903 WRITE(6,*) 'CARGA_ASCII ERROR READING .TXP FILE',IOS
GOTO 906
906 CLOSE(30,STATUS='KEEP')
CLOSE(31,STATUS='KEEP')
LEIDO=.FALSE.
RETURN
END

```

## Anexo XV: Subrutina clip.f

```

SUBROUTINE CLIP(A,MDIM,NDIM,DWNLIMIT,UPLIMIT)
DIMENSION A(MDIM,NDIM)
C
TO CLIP ARRAY TO MIN DWNLIMIT,MAX UPLIMIT
DO 10 M=1,MDIM
DO 10 N=1,NDIM
IF(A(M,N).GT.UPLIMIT) A(M,N)=UPLIMIT
10 IF(A(M,N).LT.DWNLIMIT) A(M,N)=DWNLIMIT
RETURN
END

```

## Anexo XVI: Subrutina cphas.f

```
FUNCTION CPHAS (A)
C   CALCULATES THE PHASE OF COMPLEX NUMBER A IN RADIANS
   DIMENSION A (2)
   IF ((A (1) .EQ.0.) .AND. (A (2) .EQ.0.)) THEN
       CPHAS=0.
   ELSE
       CPHAS=ATAN2 (A (2) , A (1) )
   ENDIF
   RETURN
END
```

## Anexo XVII: Subrutina dataout.f

```
      SUBROUTINE DATAOUT (PSI, MDIM, NDIM, OUTFILE, IHEADER, MHEADER
1  , IHEADER2, MHEADER2)
C
   DIMENSION PSI (MDIM, NDIM)
   DIMENSION IHEADER (MHEADER) , IHEADER2 (MHEADER2)
   COMPLEX PSI
   CHARACTER FMAP*13, OUTFILE*9
C
C   OUTPUT MAP FILE NAME
   FMAP=OUTFILE//'.map'
C   OPEN OUTPUT FILE
   OPEN (UNIT=99, STATUS='NEW', FILE=FMAP, FORM='UNFORMATTED',
1  ERR=900, IOSTAT=IOS)
C
   NRECORD=1
C
C   FIRST WRITE SIZES
   WRITE (99) MHEADER
   WRITE (99) MHEADER2
   WRITE (99) NRECORD
   WRITE (99) MDIM
C   NOW HEADERS
   DO 11 I=1, MHEADER
11  WRITE (99) IHEADER (I)
   DO 12 I=1, MHEADER2
12  WRITE (99) IHEADER2 (I)
C
C   FINALLY WRITE DATA
   DO 10 M=1, MDIM
   DO 10 N=1, NDIM
10  WRITE (99) PSI (M, N)
C   CLOSE FILE
   CLOSE (UNIT=99, STATUS='KEEP')
   RETURN
C   ERROR HANDLING
900 WRITE (6, *) 'ERROR OPENING OUTPUT .map FILE', IOS
   RETURN
END
```

## Anexo XVIII: Subrutina defocus01.f

```
      SUBROUTINE DEFOCUS01 (PSI, MDIM, NDIM, FD, RMAG, SAMPI, DELTA, II, IMASK
1 , NCASSI, IMASKC)
C
C      TO ADD II=1 OR SUBTRACT II=-1 PHASE ERROR OF FORM (1-COS(THETA))
C      OR DELTA*R**2/(1.+R**2)
C
C      D.MORRIS IRAM SEPT 1994
C
C      DELTA      -AXIAL DEFOCUS IN WAVELENGTHS
C      FD         -FOCAL LENGTH/DIAM. OF PARABOLOID
C      RMAG       -MAGNIFICATION OF CASS., SET INFINITE FOR PRIME FOCUS
C      SAMPI      -SAMPLING INTERVAL IN FAR FIELD AS FRACTION OF CRITICAL
C                 VALUE, (WAVELENGTH/DIAM.)
C
C
C      OPTIONAL MASKING OUTSIDE APERTURE IF (IMASK.EQ.1) THEN
C      IF OUTSIDE ANTENNA RADIUS SET PSI=(0.0,0.0)
C      OPTIONAL MASKING OF CENTRAL BLOCKAGE
C
C      DIMENSION PSI (MDIM, NDIM)
C
C      COMPLEX PSI, FACTOR
C      PY=3.1415926535
C
C      NDIM2=NDIM/2
C      MDIM2=MDIM/2
C      RMAX=MDIM*SAMPI/2.
C      MMAX=RMAX
C
C      CALCULATE FACTOR
C      DO 100 M=1, MDIM
C      DO 100 N=1, NDIM
C      NR2=(N-NDIM2)**2+(M-MDIM2)**2
C      RN2=FLOAT(NR2)/RMAX/RMAX
C      NR=SQRT(FLOAT(NR2))
C      R2=RN2/16./(FD**2)
C      PHIDF=4.*PY*DELTA*((R2/(1.+R2))+(R2/RMAG/RMAG)/(1.+R2/RMAG/RMAG))
C      FACTOR=CMPLX(COS(PHIDF), SIN(PHIDF))
C      IF (IMASK.EQ.1) THEN
C          IF (NR.GT.MMAX) FACTOR=(0.0,0.0)
C      END IF
C      IF (IMASKC.EQ.1) THEN
C          IF (NR.LE.NCASSI) FACTOR=(0.0,0.0)
C      END IF
C
C      APPLY FACTOR TO ARRAY
C      IF (II.LT.0) THEN
C          IF (CABS(FACTOR).NE.0) THEN
C              PSI(M,N)=PSI(M,N)/FACTOR
C          END IF
C      ELSE
C          PSI(M,N)=PSI(M,N)*FACTOR
C      END IF
C
C      100 CONTINUE
C
C      RETURN
C      END
```

## Anexo XIX: Subrutina defocus4.f

```
      SUBROUTINE DEFOCUS4 (PSI, FACTOR, MDIM, NDIM, FD, RMAG,
1  SAMPI, DELTA, II, IMASK)
C
C      TO ADD II=1 OR SUBTRACT II=-1 PHASE ERROR OF FORM (1-COS(THETA))
C      OR DELTA*R**2/(1.+R**2)
C
C      D.MORRIS IRAM 1992
C
C      DELTA      -AXIAL DEFOCUS IN WAVELENGTHS
C      FD         -FOCAL LENGTH/DIAM. OF PARABOLOID
C      RMAG       -MAGNIFICATION OF CASS., SET INFINITE FOR PRIME FOCUS
C      SAMPI      -SAMPLING INTERVAL IN FAR FIELD AS FRACTION OF CRITICAL
C                 VALUE, (WAVELENGTH/DIAM.)
C
C      COMPLEX PSI (MDIM, NDIM)
C      COMPLEX FACTOR (MDIM, NDIM)
C
C      DATA I/O/
C      PY=3.1415926535
C
10  IF (II.LT.0) GOTO 20
      DO M=1, MDIM
          DO N=1, NDIM
              PSI (M, N) = PSI (M, N) * FACTOR (M, N)
          END DO
      END DO
      RETURN
C
20  DO 300 M=1, MDIM
          DO 300 N=1, NDIM
              IF (CABS (FACTOR (M, N)) .EQ. 0.) GOTO 350
              PSI (M, N) = PSI (M, N) / FACTOR (M, N)
              GOTO 300
350          PSI (M, N) = (0.0, 0.0)
300  CONTINUE
C
      RETURN
      END
```

## Anexo XX: Subrutina fft1.f

```
      SUBROUTINE FFT1 (A,N,INVERS,X,IBIT)
C
C      TWO-DIMENSIONAL,RADIX 2,DECIMATION IN FREQ. FFT ALGORITHM
C
C      BERNARD ARAMBEPOLA,ENGINEERING DEPT.,TRUMPINGTON ST.
C      CAMBRIDGE, CB2 1PZ ,ENGLAND
C
C      A          -TWO DIMENSIONAL COMPLEX INPUT SEQUENCE
C      N          -NO. OF POINTS ALONG EACH DIMENSION
C      INVERS     -.FALSE. FOR FORWARD TRANSFORM, .TRUE. FOR INVERSE
C      X          -ONE DIMENSIONAL COMPLEX ARRAY OF N ELEMENTS
C      IBIT      -ONE DIMENSIONAL INTEGER ARRAY OF N ELEMENTS
C
C
C      ON OUTPUT...
C      THE 2-D COMPLEX ARRAY 'A' CONTAINS THE 2-D DFT(OR INVERSE)
C      IN NATURAL ORDER
C
C      DIMENSION A(N,N),X(N),IBIT(N)
C      COMPLEX A,X,AA,BB
C      LOGICAL L1,INVERS
C
C      STEP 1 : FORMING THE COMPLEX EXPONENTIALS
C
C      N1=N/2
C      PI=3.1415926536
C      QI=PI/N1
C      N2=N1/2
C      N3=N1+N2+1
C      N5=N1+2
C      N6=N
C      N7=N1
C      X(1)=(1.0,0.0)
C      X(N2+1)=(0.0,-1.0)
C      IF(INVERS) X(N2+1)=(0.0,1.0)
C      X(N1+1)=(-1.0,0.0)
C      X(N3)=(0.0,1.0)
C      IF(INVERS) X(N3)=(0.0,-1.0)
C      DO 9 I=2,N2
C      QQI=QI*(I-1)
C      XR=COS(QQI)
C      XI=-SIN(QQI)
C      IF(INVERS) XI=-XI
C      X(I)=CMPLX(XR,XI)
C      X(N7)=CMPLX(-XR,XI)
C      X(N5)=CMPLX(-XR,-XI)
C      X(N6)=CMPLX(XR,-XI)
C      N7=N7-1
C      N5=N5+1
C      9 N6=N6-1
C
C      STEP 2 : 2-D,RADIX 2,D.I.F.,F.F.T. STAGES
C
C      ITW=N1
C      ITW1=N
C      ITV=1
C      30 IC=1
C      ID=ITW
C      DO 20 I=1,ITV
C      IE=1
C      IG=ITW
C      DO 23 II=1,ITV
```

```

      K1=1
      DO 21 I1=IC, ID
      K2=1
      DO 22 J1=IE, IG
      I2=I1+ITW
      J2=J1+ITW
C
C      2-D BUTTERFLY ADDITIONS AND SUBTRACTIONS
C
      AA=A(I1, J1)+A(I1, J2)
      A(I1, J2)=A(I1, J1)-A(I1, J2)
      BB=A(I2, J1)+A(I2, J2)
      A(I2, J2)=A(I2, J1)-A(I2, J2)
      A(I1, J1)=AA+BB
      A(I2, J1)=AA-BB
      AA=A(I1, J2)+A(I2, J2)
      A(I2, J2)=A(I1, J2)-A(I2, J2)
      A(I1, J2)=AA
C
C      TWIDDLE FACTOR MULTIPLICATIONS
C
      K3=K1+K2-1
      IF(K3.EQ.1) GOTO 22
      A(I1, J2)=A(I1, J2)*X(K2)
      A(I2, J1)=A(I2, J1)*X(K1)
      A(I2, J2)=A(I2, J2)*X(K3)
22  K2=K2+ITV
21  K1=K1+ITV
      IE=IE+ITW1
23  IG=IG+ITW1
      IC=IC+ITW1
20  ID=ID+ITW1
      IF(ITW.EQ.1) GOTO 100
      ITV=ITV+ITV
      ITW1=ITW
      ITW=ITW/2
      GOTO 30
C
C      STEP 3 : 2-D BIT REVERSAL
C
C      FORMING THR BIT REVERSED ARRAY 'IBIT'
C
100  J=1
      IBIT(N)=N
      N3=N-1
      DO 10 I=1, N3
      IBIT(I)=J
      N2=N1
11  IF(N2.GE.J) GOTO 10
      J=J-N2
      N2=N2/2
      GOTO 11
10  J=J+N2
C
C      BIT-REVERSING THE 2-D ARRAY
C
      DO 12 I=1, N
      IF(IBIT(I).LT.I) GOTO 12
      I1=IBIT(I)
      L1=.FALSE.
      IF(I1.EQ.I) L1=.TRUE.
      DO 12 J=1, N
      IF(L1.AND.IBIT(J).LE.J) GOTO 12
      J1=IBIT(J)
      AA=A(I, J)
      A(I, J)=A(I1, J1)
      A(I1, J1)=AA
12  CONTINUE

```



```
C
C   STEP 4 : SCALING OPERATIONS
DO 50 I=1,N
DO 50 J=1,N
50 A(I,J)=A(I,J)/N
RETURN
END
```

## Anexo XXI: Subrutina fresnel2.f

```
      SUBROUTINE FRESNEL2 (PSI, MDIM, RLAMDA, ZTX, DIAM, SAMPI)
C
C   FIRST AND SECOND ORDER FRESNEL PHASE CORRECTION FOR NEAR FIELD
C   PATH OF FORM  $(X^{**2}+Y^{**2})/2.*ZTX - (X^{**2}+Y^{**2})^{**2}/ZTX^{**3}/8.$ 
C
C   CENTER OF APERTURE MDIM/2
C   DIMENSION PSI (MDIM,MDIM)
C   COMPLEX PSI, PHASEC
C
C   PY=3.1415926535
C
C   MDIM21=MDIM/2
C
C
C   DO 10 M=1, MDIM
C   DO 10 N=1, MDIM
C   SAMPLING INTERVAL IN APERTURE
C   DELR=DIAM/SAMPI / (MDIM)
C   R2= ((M-MDIM21)**2+(N-MDIM21)**2)*DELR**2
C   FIRST ORDER CORRECTION
C   PHASE=PY*R2/RLAMDA/ZTX
C   SECOND ORDER CORRECTION
C   PHASE=PHASE-PY*R2**2/RLAMDA/ZTX**3/8.
C
C   PHASEC=CMPLX (COS (PHASE) , SIN (PHASE) )
10  PSI (M, N) =PSI (M, N) *PHASEC
C
C   RETURN
C   END
```

## Anexo XXII: Subrutina getmap3.f

```
      SUBROUTINE GETMAP3 (DATA, MDIM, NUNIT, NX, RX, VX, DELX, NY, RY, VY, DELY,
1 MOFFSET, NOFFSET, IRECENT)
C
C   TO READ DATA FROM GREG FORMAT MAP WRITTEN BY LUCAS' "MAP" PROGRAM
C   VERSION TO REVERSE DATA ORDER TO AGREE WITH PICOH2 FORMAT
C   VERSION TO AVOID ARRAY OVERFLOW WITH OFFSETS
C
C   NX, NY NO. OF PIXELS IN X AND Y
C   RX, RY REFERENCE PIXELS E.G. 1
C   VX, VY VALUE OF REFERENCE E.G -57.5 ARC SECS.
C   DELX, DELY SEPARATION OF PIXELS IN ARC SECS.
C
C   DATA ARE WRITTEN INTO ARRAY WITH CENTER MDIM/2 +1
C   WITH OFFSETS MOFFSET, NOFFSET IN AZ ,ELV
C
C   WITH (INTEGRAL) RECENTERING IF :-
C           IRECENT=1
C           REFERENCE PIXEL NOT ZERO
C           REFERENCE PIXEL IS 1 AND ITS VALUE IS LESS THAN 1
C
C   FOR PHASE RETRIEVAL
C
C
C   DIMENSION DATA (MDIM, MDIM) , BUFFER (20)
C
C   CHARACTER COMMENT*80
C
C   LOGICAL LEIDO
C
```

```

C      READ HEADER DATA
      READ (NUNIT, *) NX, RX, VX, DELX
      READ (NUNIT, 3) COMMENT
3     FORMAT (A80)
      WRITE (66, *)
      WRITE (66, *) NX, RX, VX, DELX
      WRITE (66, 3) COMMENT
      WRITE (6, *) NX, RX, VX, DELX
      WRITE (6, 3) COMMENT
      IF (NX.GT.MDIM) WRITE (6, *) 'TOO MUCH DATA NX.GT.MDIM'
      READ (NUNIT, *) NY, RY, VY, DELY
      READ (NUNIT, 3) COMMENT
      IF (NY.GT.MDIM) WRITE (6, *) 'TOO MUCH DATA NY.GT.MDIM'
      WRITE (66, *) NY, RY, VY, DELY
      WRITE (66, 3) COMMENT
      WRITE (66, *)
      WRITE (6, *) NY, RY, VY, DELY
      WRITE (6, 3) COMMENT

C
      MDIM21=MDIM/2+1

C
C      IF REFERENCE PIXEL IS NOT ZERO OR IF IT IS 1 AND THE REFERENCE VALUE IS
C      LESS THEN 1 THEN RECENTER BY AN INTEGER VALUE
      IVY=ABS (VY/DELY)
      IF ( RY.EQ.0. .or. (RY.EQ.1. .AND. IVY.LT.1) ) THEN
          MCENTY=0
          MCENTX=0
      ELSE
          IF (IRECENT.EQ.1) THEN
              MCENTY= ((RY-1.-NY/2)*DELY-VY)/DELY
              MCENTX= ((RX-1.-NX/2)*DELX-VX)/DELX
          ELSE
              MCENTX=0
              MCENTY=0
          END IF
      END IF

C
      MM=MDIM21+NX/2+MOFFSET+MCENTX
      NN=MDIM21+NY/2+NOFFSET+MCENTY

C
C      READ DATA
      NREADS=0
      LEIDO=.FALSE.
1     READ (NUNIT, 1000, END=70) (BUFFER(I), I=1, 20)
1000  FORMAT (G15.7E2)
71    NREADS=NREADS+1
      DO 10 I=1, 20
          NDAT=(NREADS-1)*20 +I
          IF (NDAT.GT.NX*NY) GOTO 99
          N=1+(NDAT-1)/NX
          M=NDAT-(N-1)*NX
          M1=MM-M
          N1=NN-N
          IF (M1.GT.MDIM) GOTO 10
          IF (N1.GT.MDIM) GOTO 10
          IF (M1.LT.1) GOTO 10
          IF (N1.LT.1) GOTO 10
          DATA (M1, N1)=BUFFER(I)
10    CONTINUE
C      LOOP
      IF (LEIDO) GOTO 99
      GOTO 1

C
70    DO J=I+1, 20
          BUFFER(J)=0.
      END DO
      LEIDO=.TRUE.
      GOTO 71

```

```

C
99 CONTINUE
WRITE(6,*) ' NO. DATA POINTS READ = ',NDAT-1
WRITE(66,*) ' NO. DATA POINTS READ = ',NDAT-1
C
C
FILL WITH ZEROES
do u=1,mdim
  do v=1,mdim
    if ((u.le.16).or.(u.ge.49)) then
      data(u,v)=0.
    else if ((v.le.16).or.(v.ge.49)) then
      data(u,v)=0.
    end if
  end do
end do
C
RETURN
END

```

### Anexo XXIII: Subrutina greg\_color.f

```

SUBROUTINE GREG_COLOR(MAPAA,MAPAP,MDIM, IDEN)
C
DIBUJA DOS MAPAS EN COLOR
C
AMBOS MAPAS DEBEN SER DE 64x64
C
C
EL ENTERO IDEN IDENTIFICA EL TIPO DE MAPA A REALIZAR
C
1 PARA GRAFICAS DE AVMAP
C
2 PARA GRAFICAS DE AMPL+PHASE CON PICOMISTY
C
3 PARA AMPLITUD RESIDUALS DE PICOMISTY
C
4 FOC. FAR FIELD AND DEFOC. FAR FIELD DE PICOMISTY
C
5 PARA GRAFICAS DE MAPERROR
C
6 LA QUE TU QUIERAS ANIADIR POSTERIORMENTE.
C
REAL MAPAA(MDIM,MDIM),MAPAP(MDIM,MDIM)
REAL*8 CONVA(6),CONVP(6),A,PXX,PYY
INTEGER I,J, IDEN
C
DIBUJO: SELECT CASE (IDEN)
C*****
C
CASE 1
C*****
CASE (1) DIBUJO
CALL GR_EXEC('CLEAR')
C
PONEMOS PAGINA NO APAISADA PARA SACAR
C
EN LA MISMA PAGINA MODULO Y FASE
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C
PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')
C
C
CONVA(1)=1.0 ! REFERENCE PIXEL IN X
CONVA(2)=0.0 ! X USER COORD. VALUE AT CONV(1)
CONVA(3)=1.0 ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0 ! REFERENCE PIXEL IN Y
CONVA(5)=0.0 ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0 ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN 5 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN 0.05*RGMAX RGMAX')
* CALL GR_EXEC('LEVELS 5 TO 95 BY 10')

```

```

*      CALL GR_EXEC('RGM /PER 1.')
      CALL GR_EXEC('SET ORIENTATION 90')
      CALL GR_EXEC('DRAW TEXT 2.5 -0 "AMPLITUDE (%)" 2 /BOX 6')
      CALL GR_EXEC('SET ORIENTATION 0')
      CALL GR_EXEC('TICK 1 2 1 2')
      CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
      CALL GR_EXEC('ELLIPSE 13.7|2 /USER 0 0')
      CALL GR_EXEC('BOX')
      CALL GR_EXEC('LABEL "Metres" /Y')
      CALL GR_EXEC('SET CHAR 0.65')
      CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
      CALL GR_EXEC('SET CHAR .5')
C
C      AHORA LA FASE
      CONV(1)=1.0      !REFERENCE PIXEL IN X
      CONV(2)=0.0      !X USER COORDINATE VALUE AT CONV(1)
      CONV(3)=1.0      !X USER COORDINATE INCREMENT PER PIXEL
      CONV(4)=1.0      !REFERENCE PIXEL IN Y
      CONV(5)=0.0      !Y USER COORDINATE VALUE AT CONV(4)
      CONV(6)=1.0      !Y USER COORDINATE INCREMENT PER PIXEL
C
      CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
      CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
      CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
      CALL GR_EXEC('PLOT /SCAL LIN -2.5 2.5')
      CALL GR_EXEC('WEDGE')
*      CALL GR_EXEC('LEVELS -95 TO 95 BY 10')
*      CALL GR_EXEC('RGM /PER 1.')
      CALL GR_EXEC('SET ORIENTATION 90')
      CALL GR_EXEC('DRAW TEXT 2.5 -0 "PHASE (rad)" 2 /BOX 6')
      CALL GR_EXEC('SET ORIENTATION 0')
      CALL GR_EXEC('TICK 1 2 1 2')
      CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
      CALL GR_EXEC('BOX')
      CALL GR_EXEC('LABEL "Metres" /X')
      CALL GR_EXEC('LABEL "Metres" /Y')
      CALL GR_EXEC('ELLIPSE 1.12|2/USER 0 0')
      CALL GR_EXEC('ELLIPSE 7.72|2/USER 0 0')
      CALL GR_EXEC('ELLIPSE 13.7|2/USER 0 0')
C
      CALL GR_EXEC('PENCIL 0 /COLOUR 0 /DASHED 1 /WEIGHT 1')
      DO I=0,23
        A=I*3.1416/12.0
        PXX=(1.12/2.)*SIN(A)
        PYY=(1.12/2.)*COS(A)
        CALL GR8_GIVE('X',1,PXX)
        CALL GR8_GIVE('Y',1,PYY)
        CALL GR_EXEC('DRAW REL X Y /USER')
        PXX=(7.72/2.)*SIN(A)
        PYY=(7.72/2.)*COS(A)
        CALL GR8_GIVE('X',1,PXX)
        CALL GR8_GIVE('Y',1,PYY)
        CALL GR_EXEC('DRAW LIN X Y /USER')
      END DO
C
      DO J=0,47
        A=J*3.1416/24.0
        PXX=(7.72/2.)*SIN(A)
        PYY=(7.72/2.)*COS(A)
        CALL GR8_GIVE('X',1,PXX)
        CALL GR8_GIVE('Y',1,PYY)
        CALL GR_EXEC('DRAW REL X Y /USER')
        PXX=(13.7/2.)*SIN(A)
        PYY=(13.7/2.)*COS(A)
        CALL GR8_GIVE('X',1,PXX)
        CALL GR8_GIVE('Y',1,PYY)
        CALL GR_EXEC('DRAW LIN X Y /USER')
      END DO

```

```

C      CALL GR_EXEC('HARDCOPY AVMAP_COLOR.PS /DEV PS COLOR')
      CALL GR_EXEC('CLEAR')
      RETURN
C*****
C          CASE 2
C*****
      CASE (2) DIBUJO
      CALL GR_EXEC('CLEAR')
C      PONEMOS PAGINA NO APAISADA PARA SACAR
C      EN LA MISMA PAGINA MODULO Y FASE
      CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C      PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
      CALL GR_EXEC('SET FONT DUPLEX')
      CALL GR_EXEC('SET CHARACTER 0.5')
C
      CONVA(1)=1.0      ! REFERENCE PIXEL IN X
      CONVA(2)=0.0      ! X USER COORD. VALUE AT CONV(1)
      CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
      CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
      CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
      CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL
C
      CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
      CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
      CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
      CALL GR_EXEC('PLOT /SCAL LIN 5 100')
      CALL GR_EXEC('WEDGE')
      CALL GR_EXEC('PLOT /SCAL LIN 0.05*RGMAX RGMAX')
*      CALL GR_EXEC('LEVELS 5 TO 95 BY 10')
*      CALL GR_EXEC('RGM /PER 1.')
      CALL GR_EXEC('SET ORIENTATION 90')
      CALL GR_EXEC('DRAW TEXT 2.5 -0 "AMPLITUDE (%)" 2 /BOX 6')
      CALL GR_EXEC('SET ORIENTATION 0')
      CALL GR_EXEC('TICK 1 2 1 2')
      CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
      CALL GR_EXEC('ELLIPSE 13.7|2 /USER 0 0')
      CALL GR_EXEC('BOX')
      CALL GR_EXEC('LABEL "Metres" /Y')
      CALL GR_EXEC('SET CHAR 0.65')
      CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
      CALL GR_EXEC('SET CHAR .5')
C
      AHORA LA FASE

      CONVP(1)=1.0      !REFERENCE PIXEL IN X
      CONVP(2)=0.0      !X USER COORDINATE VALUE AT CONV(1)
      CONVP(3)=1.0      !X USER COORDINATE INCREMENT PER PIXEL
      CONVP(4)=1.0      !REFERENCE PIXEL IN Y
      CONVP(5)=0.0      !Y USER COORDINATE VALUE AT CONV(4)
      CONVP(6)=1.0      !Y USER COORDINATE INCREMENT PER PIXEL
C
      CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
      CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
      CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
      CALL GR_EXEC('PLOT /SCAL LIN -2.5 2.5')
      CALL GR_EXEC('WEDGE')
*      CALL GR_EXEC('LEVELS -95 TO 95 BY 10')
*      CALL GR_EXEC('RGM /PER 1.')
      CALL GR_EXEC('SET ORIENTATION 90')
      CALL GR_EXEC('DRAW TEXT 2.5 -0 "PHASE (rad)" 2 /BOX 6')
      CALL GR_EXEC('SET ORIENTATION 0')
      CALL GR_EXEC('TICK 1 2 1 2')
      CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
      CALL GR_EXEC('BOX')
      CALL GR_EXEC('LABEL "Metres" /X')
      CALL GR_EXEC('LABEL "Metres" /Y')

```

```

CALL GR_EXEC('ELLIPSE 1.12|2/USER 0 0')
CALL GR_EXEC('ELLIPSE 7.72|2/USER 0 0')
CALL GR_EXEC('ELLIPSE 13.7|2/USER 0 0')
C
CALL GR_EXEC('PENCIL 0 /COLOUR 0 /DASHED 1 /WEIGHT 1')
DO I=0,23
  A=I*3.1416/12.0
  PXX=(1.12/2.)*SIN(A)
  PYY=(1.12/2.)*COS(A)
  CALL GR8_GIVE('X',1,PXX)
  CALL GR8_GIVE('Y',1,PYY)
  CALL GR_EXEC('DRAW REL X Y /USER')
  PXX=(7.72/2.)*SIN(A)
  PYY=(7.72/2.)*COS(A)
  CALL GR8_GIVE('X',1,PXX)
  CALL GR8_GIVE('Y',1,PYY)
  CALL GR_EXEC('DRAW LIN X Y /USER')
END DO
C
DO J=0,47
  A=J*3.1416/24.0
  PXX=(7.72/2.)*SIN(A)
  PYY=(7.72/2.)*COS(A)
  CALL GR8_GIVE('X',1,PXX)
  CALL GR8_GIVE('Y',1,PYY)
  CALL GR_EXEC('DRAW REL X Y /USER')
  PXX=(13.7/2.)*SIN(A)
  PYY=(13.7/2.)*COS(A)
  CALL GR8_GIVE('X',1,PXX)
  CALL GR8_GIVE('Y',1,PYY)
  CALL GR_EXEC('DRAW LIN X Y /USER')
END DO
C
CALL GR_EXEC('HARDCOPY PICOMAP_COLOR.PS /DEV PS COLOR')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C                      CASE 3
C*****
CASE (3) DIBUJO
CALL GR_EXEC('CLEAR')
C
PONEMOS PAGINA NO APAISADA PARA SACAR
C
EN LA MISMA PAGINA MODULO Y FASE
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C
PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')
C
CONVA(1)=1.0          ! REFERENCE PIXEL IN X
CONVA(2)=0.0          ! X USER COORD. VALUE AT CONV(1)
CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN -100 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN -1.0*RGMAX RGMAX')
*
CALL GR_EXEC('LEVELS -95 TO 95 BY 10')
*
CALL GR_EXEC('RGM /PER 1.')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 2.5 0 "RESIDUAL AMPLITUDE (%)" 2 /BOX 6')
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 5 10 5 10')

```

```

CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Arcmin" /X')
CALL GR_EXEC('LABEL "Arcmin" /Y')
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
CALL GR_EXEC('SET CHAR .5')
CALL GR_EXEC('HARDCOPY RESID_COLOR.PS /DEV PS COLOR')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C                               CASE 4
C*****
CASE (4) DIBUJO
CALL GR_EXEC('CLEAR')
C PONEMOS PAGINA NO APAISADA PARA SACAR
C EN LA MISMA PAGINA MODULO Y FASE
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')
C
CONVA(1)=1.0      ! REFERENCE PIXEL IN X
CONVA(2)=0.0      ! X USER COORD. VALUE AT CONV(1)
CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN 0 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN 0 RGMAX')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 2.5 0 "INFOCUSSED POWER" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3.5 0 " PATTERN (%)" 2 /BOX 6')
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 5 10 5 10')
CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Arcmin" /Y')
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
CALL GR_EXEC('SET CHAR .5')
C
C AHORA LA FASE
CONVP(1)=1.0      !REFERENCE PIXEL IN X
CONVP(2)=0.0      !X USER COORDINATE VALUE AT CONV(1)
CONVP(3)=1.0      !X USER COORDINATE INCREMENT PER PIXEL
CONVP(4)=1.0      !REFERENCE PIXEL IN Y
CONVP(5)=0.0      !Y USER COORDINATE VALUE AT CONV(4)
CONVP(6)=1.0      !Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN 0 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN 0 RGMAX')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 2.5 0 "DEFOCUSSED POWER" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3.5 0 " PATTERN (%)" 2 /BOX 6')
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 5 10 5 10')
CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')

```



```

CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Arcmin" /X')
CALL GR_EXEC('LABEL "Arcmin" /Y')
C
CALL GR_EXEC('HARDCOPY FARFIELDS_COLOR.PS /DEV PS COLOR')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C
CASE 5
C*****
CASE (5) DIBUJO
CALL GR_EXEC('CLEAR')
C
PONEMOS PAGINA NO APAISADA PARA SACAR
C
EN LA MISMA PAGINA MODULO Y FASE
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C
PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')
C
CONVA(1)=1.0      ! REFERENCE PIXEL IN X
CONVA(2)=0.0      ! X USER COORD. VALUE AT CONV(1)
CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN -100 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN -1.0*RGMAX RGMAX')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 2.5 0 "PHASE DIFFERENCE (%)" 2 /BOX 6')
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 1 2 1 2')
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /X')
CALL GR_EXEC('LABEL "Metres" /Y')
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
CALL GR_EXEC('SET CHAR .5')
CALL GR_EXEC('HARDCOPY MAPERROR_COLOR.PS /DEV PS COLOR')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C
CASE DEFAULT
C*****
CASE DEFAULT DIBUJO
WRITE(6,*) 'IDENTIFICADOR DE DIBUJO INCORRECTO'
WRITE(6,*) 'ERROR EN SUBROUTINA GREG_COLOR.F'
RETURN
C*****
C
FIN DEL CASE
C*****
END SELECT DIBUJO
END

```

## Anexo XXIV: Subrutina greg\_contgrey.f

```

SUBROUTINE GREG_CONTGREY (MAPAA, MAPAP, MDIM, IDEN)
C
C DIBUJA LO CONTORNOS DE GRIS DE DOS MAPAS
C
C AMBOS MAPAS DEBEN SER DE 64x64
C
C
C EL ENTERO IDEN IDENTIFICA EL TIPO DE MAPA A REALIZAR
C
C 1 PARA GRAFICAS DE AVMAP
C
C 2 PARA GRAFICAS DE AMPL+PHASE CON PICOMISTY
C
C 3 PARA AMPLITUD RESIDUALS DE PICOMISTY
C
C 4 FOC. FAR FIELD AND DEFOC. FAR FIELD DE PICOMISTY
C
C 5 PARA GRAFICAS DE MAPERROR
C
C 6 LA QUE TU QUIERAS ANIADIR POSTERIORMENTE.
C
C
C REAL MAPAA (MDIM, MDIM) , MAPAP (MDIM, MDIM)
C REAL*8 CONVA (6) , CONVP (6)
C INTEGER IDEN
C
C
C DIBUJO:          SELECT CASE (IDEN)
C*****
C
C          CASO 1
C*****
C
C CASE (1) DIBUJO
C CALL GR_EXEC('CLEAR')
C
C PONEMOS PAGINA NO APAISADA PARA SACAR
C EN LA MISMA PAGINA MODULO Y FASE
C CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C
C PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
C CALL GR_EXEC('SET FONT DUPLEX')
C CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C
C DEFINIMOS LA MATRIZ DE CONVERSION DE AMPLITUD (VER HELP)
C CONVA(1)=1.0          ! REFERENCE PIXEL IN X
C CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
C CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
C CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
C CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
C CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
C
C PASAMOS EL PRIMER MAPA
C CALL GR4_RGIVE (MDIM, MDIM, CONVA, MAPAA)
C
C CENTRAMOS LA CAJA EN LA PARTE SUPERIOR DE LA PAGINA
C CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
C
C PONEMOS LIMITES EN PIXELES, DE MOMENTO
C CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C
C ESTABLECEMOS LOS NIVELES QUE QUEREMOS DIBUJAR
C CALL GR_EXEC('LEVELS 5 TO 95 BY 10')
C
C LOS DIBUJAMOS
C CALL GR_EXEC('RGM /PER 1. /GREY')
C
C DIBUJAMOS ETIQUETA DE AMPLITUD
C CALL GR_EXEC('SET ORIENTATION 90')
C CALL GR_EXEC('DRAW TEXT 1 0 "AMPLITUDE (%)" 2 /BOX 6')
C CALL GR_EXEC('DRAW TEXT 2 0 " 10% contours" 2 /BOX 6')
C CALL GR_EXEC('DRAW TEXT 3 0 "   5% to 95% " 2 /BOX 6')
C
C DESHACEMOS LA ORIENTACION VERTICAL
C CALL GR_EXEC('SET ORIENTATION 0')
C
C QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
C CALL GR_EXEC('TICK 1 2 1 2')
C
C PONEMOS LIMITES REALES 8.55225=63/2/2*0.543
C RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
C CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
C
C DIBUJAMOS EL CONTORNO DEL PRIMARIO
C CALL GR_EXEC('ELLIPSE 13.7|2 /USER 0 0')
C
C ENMARCAMOS EL DIBUJO
C CALL GR_EXEC('BOX')

```

```

CALL GR_EXEC('LABEL "Metres" /Y')
C
C CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C
LO VOLVEMOS A PONER A SU VALOR USUAL
CALL GR_EXEC('SET CHAR .5')

C
AHORA LA FASE
C DEFINIMOS LA MATRIZ DE CONVERSION DE FASE (VER HELP)
CONVP(1)=1.0      !REFERENCE PIXEL IN X
CONVP(2)=0.0      !X USER COORDINATE VALUE AT CONV(1)
CONVP(3)=1.0      !X USER COORDINATE INCREMENT PER PIXEL
CONVP(4)=1.0      !REFERENCE PIXEL IN Y
CONVP(5)=0.0      !Y USER COORDINATE VALUE AT CONV(4)
CONVP(6)=1.0      !Y USER COORDINATE INCREMENT PER PIXEL

C
PASAMOS EL SEGUNDO MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
C CENTRAMOS LA CAJA EN LA PARTE INFERIOR DE LA PAGINA
CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
C PONEMOS LIMITES EN PIXELES, DE MOMENTO
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C ESTABLECEMOS NIVELES A DIBUJAR
CALL GR_EXEC('LEVELS -95 TO 95 BY 10')
C LOS DIBUJAMOS
CALL GR_EXEC('RGM /PER 1. /GREY')
C DIBUJAMOS ETIQUETA DE FASE
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "PHASE (radians)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 " 10% contours " 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 " -95% to +95% " 2 /BOX 6')
C DESHACEMOS LA ORIENTACION VERTICAL
CALL GR_EXEC('SET ORIENTATION 0')
C QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
CALL GR_EXEC('TICK 1 2 1 2')
C PONEMOS LIMITES REALES 8.55225=63/2/2*0.543
C RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
C EL MARCO
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /X')
CALL GR_EXEC('LABEL "Metres" /Y')
C EL CONTORNO DEL PRIMARIO
CALL GR_EXEC('ELLIPSE 13.7|2/USER 0 0')

C
C Y POR ULTIMO MANDAMOS QUE SE IMPRIMA EN UN FICHERO .PS
CALL GR_EXEC('HARDCOPY AVMAP_CONTGREY.PS /DEV PS FAST')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C          CASE 2
C*****
CASE (2) DIBUJO
CALL GR_EXEC('CLEAR')
C PONEMOS PAGINA NO APAISADA PARA SACAR
C EN LA MISMA PAGINA MODULO Y FASE
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')

C
C PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')      !TAMANIO LETRAS 0.5CM

C
C DEFINIMOS LA MATRIZ DE CONVERSION DE AMPLITUD (VER HELP)
CONVA(1)=1.0      ! REFERENCE PIXEL IN X
CONVA(2)=0.0      ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0      ! REFERENCE PIXEL IN Y

```

```

CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL
C
C PASAMOS EL PRIMER MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
C CENTRAMOS LA CAJA EN LA PARTE SUPERIOR DE LA PAGINA
CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
C PONEMOS LIMITES EN PIXELES, DE MOMENTO
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C ESTABLECEMOS LOS NIVELES QUE QUEREMOS DIBUJAR
CALL GR_EXEC('LEVELS 5 TO 95 BY 10')
C LOS DIBUJAMOS
CALL GR_EXEC('RGM /PER 1. /GREY')
C DIBUJAMOS ETIQUETA DE AMPLITUD
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "AMPLITUDE (%)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 " 10% contours" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 "   5% to 95%" 2 /BOX 6')
C DESHACEMOS LA ORIENTACION VERTICAL
CALL GR_EXEC('SET ORIENTATION 0')
C QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
CALL GR_EXEC('TICK 1 2 1 2')
C PONEMOS LIMITES REALES  $8.55225=63/2/2*0.543$ 
C RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
C DIBUJAMOS EL CONTORNO DEL PRIMARIO
CALL GR_EXEC('ELLIPSE 13.7|2 /USER 0 0')
C ENMARCAMOS EL DIBUJO
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /Y')
C
C CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C LO VOLVEMOS A PONER A SU VALOR USUAL
CALL GR_EXEC('SET CHAR .5')
C
C AHORA LA FASE
C DEFINIMOS LA MATRIZ DE CONVERSION DE FASE (VER HELP)
CONVP(1)=1.0      !REFERENCE PIXEL IN X
CONVP(2)=0.0      !X USER COORDINATE VALUE AT CONV(1)
CONVP(3)=1.0      !X USER COORDINATE INCREMENT PER PIXEL
CONVP(4)=1.0      !REFERENCE PIXEL IN Y
CONVP(5)=0.0      !Y USER COORDINATE VALUE AT CONV(4)
CONVP(6)=1.0      !Y USER COORDINATE INCREMENT PER PIXEL
C
C PASAMOS EL SEGUNDO MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
C CENTRAMOS LA CAJA EN LA PARTE INFERIOR DE LA PAGINA
CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
C PONEMOS LIMITES EN PIXELES, DE MOMENTO
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C ESTABLECEMOS NIVELES A DIBUJAR
CALL GR_EXEC('LEVELS -95 TO 95 BY 10')
C LOS DIBUJAMOS
CALL GR_EXEC('RGM /PER 1. /GREY')
C DIBUJAMOS ETIQUETA DE FASE
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "PHASE (radians)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 " 10% contours " 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 " -95% to +95%" 2 /BOX 6')
C DESHACEMOS LA ORIENTACION VERTICAL
CALL GR_EXEC('SET ORIENTATION 0')
C QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
CALL GR_EXEC('TICK 1 2 1 2')
C PONEMOS LIMITES REALES  $8.55225=63/2/2*0.543$ 
C RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')

```

```

C      EL MARCO
      CALL GR_EXEC('BOX')
      CALL GR_EXEC('LABEL "Metres" /X')
      CALL GR_EXEC('LABEL "Metres" /Y')
C      EL CONTORNO DEL PRIMARIO
      CALL GR_EXEC('ELLIPSE 13.7|2/USER 0 0')
C
C      Y POR ULTIMO MANDAMOS QUE SE IMPRIMA EN UN FICHERO .PS
      CALL GR_EXEC('HARDCOPY PICOMAP_CONTGREY.PS /DEV PS FAST')
      CALL GR_EXEC('CLEAR')
      RETURN
C*****
C                      CASE 3
C*****
      CASE (3) DIBUJO
      CALL GR_EXEC('CLEAR')
      CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
      CALL GR_EXEC('SET FONT DUPLEX')
      CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C      DEFINIMOS LA MATRIZ DE CONVERSION (VER HELP)
      CONVA(1)=1.0          ! REFERENCE PIXEL IN X
      CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
      CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
      CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
      CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
      CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
      CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
      CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
      CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
      CALL GR_EXEC('LEVELS -95 TO 95 BY 10')
      CALL GR_EXEC('RGM /PER 1. /GREY')
      CALL GR_EXEC('SET ORIENTATION 90')
      CALL GR_EXEC('DRAW TEXT 1 0 "RESIDUAL AMPLITUDE (%)" 2 /BOX 6')
      CALL GR_EXEC('DRAW TEXT 2 0 "      10% contours      " 2 /BOX 6')
      CALL GR_EXEC('DRAW TEXT 3 0 "      -95% to +95%      " 2 /BOX 6')
      CALL GR_EXEC('SET ORIENTATION 0')
      CALL GR_EXEC('TICK 5 10 5 10')
      CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')
      CALL GR_EXEC('BOX')
      CALL GR_EXEC('LABEL "Arcmin" /X')
      CALL GR_EXEC('LABEL "Arcmin" /Y')
C
C      CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
      CALL GR_EXEC('SET CHAR 0.65')
      CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C      LO VOLVEMOS A PONER A SU VALOR USUAL
      CALL GR_EXEC('SET CHAR .5')
      CALL GR_EXEC('HARDCOPY RESID_CONTGREY.PS /DEV PS FAST')
      CALL GR_EXEC('CLEAR')
      RETURN
C*****
C                      CASE 4
C*****
      CASE (4) DIBUJO
      CALL GR_EXEC('CLEAR')
C      PONEMOS PAGINA NO APAISADA PARA SACAR
C      EN LA MISMA PAGINA MODULO Y FASE
      CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C      PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
      CALL GR_EXEC('SET FONT DUPLEX')
      CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C      DEFINIMOS LA MATRIZ DE CONVERSION DE AMPLITUD (VER HELP)
      CONVA(1)=1.0          ! REFERENCE PIXEL IN X

```

```

CONVA(2)=0.0      ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL

C
C PASAMOS EL PRIMER MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
C CENTRAMOS LA CAJA EN LA PARTE SUPERIOR DE LA PAGINA
CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
C PONEMOS LIMITES EN PÍXELES, DE MOMENTO
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C ESTABLECEMOS LOS NIVELES QUE QUEREMOS DIBUJAR
CALL GR_EXEC('LEVELS 0.0001 0.001 0.01 0.1 1 10 100')
C LOS DIBUJAMOS
CALL GR_EXEC('RGM /PER 1. /GREY')
C DIBUJAMOS ETIQUETA DE AMPLITUD
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "INFOC. POWER PATTERN (dB)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 "      10dB contours      " 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 "      -60dB to 0dB      " 2 /BOX 6')
C DESHACEMOS LA ORIENTACION VERTICAL
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 5 10 5 10')
CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Arcmin" /Y')

C
C CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C LO VOLVEMOS A PONER A SU VALOR USUAL
CALL GR_EXEC('SET CHAR .5')

C
C DEFINIMOS LA MATRIZ DE CONVERSION DE FASE (VER HELP)
CONVP(1)=1.0      !REFERENCE PIXEL IN X
CONVP(2)=0.0      !X USER COORDINATE VALUE AT CONV(1)
CONVP(3)=1.0      !X USER COORDINATE INCREMENT PER PIXEL
CONVP(4)=1.0      !REFERENCE PIXEL IN Y
CONVP(5)=0.0      !Y USER COORDINATE VALUE AT CONV(4)
CONVP(6)=1.0      !Y USER COORDINATE INCREMENT PER PIXEL

C
C PASAMOS EL SEGUNDO MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
C CENTRAMOS LA CAJA EN LA PARTE INFERIOR DE LA PAGINA
CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
C PONEMOS LIMITES EN PÍXELES, DE MOMENTO
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C ESTABLECEMOS NIVELES A DIBUJAR
CALL GR_EXEC('LEVELS 0.0001 0.001 0.01 0.1 1 10 100')
C LOS DIBUJAMOS
CALL GR_EXEC('RGM /PER 1. /GREY')
C DIBUJAMOS ETIQUETA DE FASE
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "DEFOC. POWER PATTERN (dB)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 "      10dB contours      " 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 "      -60dB to 0dB      " 2 /BOX 6')
C DESHACEMOS LA ORIENTACION VERTICAL
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 5 10 5 10')
CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')
C EL MARCO
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Arcmin" /X')
CALL GR_EXEC('LABEL "Arcmin" /Y')

C
C Y POR ULTIMO MANDAMOS QUE SE IMPRIMA EN UN FICHERO .PS
CALL GR_EXEC('HARDCOPY FARFIELDS_CONTGREY.PS /DEV PS FAST')

```

```

CALL GR_EXEC('CLEAR')
RETURN
C*****
C                                CASE 5
C*****
CASE (5) DIBUJO
CALL GR_EXEC('CLEAR')
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C   DEFINIMOS LA MATRIZ DE CONVERSION (VER HELP)
CONVA(1)=1.0          ! REFERENCE PIXEL IN X
CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('LEVELS -95 TO 95 BY 10')
CALL GR_EXEC('RGM /PER 1. /GREY')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "PHASE DIFFERENCE (%)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 " 10% contours " 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 " -95% to +95% " 2 /BOX 6')
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 1 2 1 2')
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /X')
CALL GR_EXEC('LABEL "Metres" /Y')
C
C   CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C   LO VOLVEMOS A PONER A SU VALOR USUAL
CALL GR_EXEC('SET CHAR .5')
CALL GR_EXEC('HARDCOPY MAPERROR_CONTGREY.PS /DEV PS FAST')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C                                CASE DEFAULT
C*****
CASE DEFAULT DIBUJO
WRITE(6,*) 'IDENTIFICADOR DE DIBUJO INCORRECTO'
WRITE(6,*) 'ERROR EN SUBROUTINA GREG_CONTGREY.F'
RETURN
C*****
C                                FIN DEL CASE
C*****
END SELECT DIBUJO
END

```

## Anexo XXV: Subrutina greg\_contour.f

```

      SUBROUTINE GREG_CONTOUR (MAPAA,MAPAP,MDIM, IDEN)
C     DIBUJA LAS CURVAS DE NIVEL DE DOS MAPAS
C     AMBOS MAPAS DEBEN SER DE 64x64
C
C     EL ENTERO IDEN IDENTIFICA EL TIPO DE MAPA A REALIZAR
C     1 PARA GRAFICAS DE AVMAP
C     2 PARA GRAFICAS DE AMPL+PHASE CON PICOMISTY
C     3 PARA AMPLITUD RESIDUALS DE PICOMISTY
C     4 FOC. FAR FIELD AND DEFOC. FAR FIELD DE PICOMISTY
C     5 PARA GRAFICAS DE MAPERROR
C     6 LA QUE TU QUIERAS ANIADIR POSTERIORMENTE.
C
      REAL MAPAA (MDIM,MDIM) ,MAPAP (MDIM,MDIM)
      REAL*8 CONVA (6) ,CONVP (6)
      INTEGER IDEN
C
      DIBUJO:      SELECT CASE (IDEN)
C*****
C                   CASO 1
C*****
      CASE (1) DIBUJO
      CALL GR_EXEC('CLEAR')
      CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
      CALL GR_EXEC('SET FONT DUPLEX')
      CALL GR_EXEC('SET CHARACTER 0.5')
C
C     DEFINIMOS LA MATRIZ DE CONVERSION DE AMPLITUD (VER HELP)
      CONVA(1)=1.0      ! REFERENCE PIXEL IN X
      CONVA(2)=0.0      ! X USER COORDINATE VALUE AT CONV(1)
      CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
      CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
      CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
      CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL
C
C     PASAMOS EL PRIMER MAPA
      CALL GR4_RGIVE (MDIM,MDIM,CONVA,MAPAA)
C     CENTRAMOS LA CAJA EN LA PARTE SUPERIOR DE LA PAGINA
      CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
C     PONEMOS LIMITES EN PIXELES, DE MOMENTO
      CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C     ESTABLECEMOS LOS NIVELES QUE QUEREMOS DIBUJAR
      CALL GR_EXEC('LEVELS 5 TO 95 BY 10')
C     LOS DIBUJAMOS
      CALL GR_EXEC('RGM /PER 1.')
C     DIBUJAMOS ETIQUETA DE AMPLITUD
      CALL GR_EXEC('SET ORIENTATION 90')
      CALL GR_EXEC('DRAW TEXT 1 0 "AMPLITUDE (%)" 2 /BOX 6')
      CALL GR_EXEC('DRAW TEXT 2 0 " 10% contours" 2 /BOX 6')
      CALL GR_EXEC('DRAW TEXT 3 0 "   5% to 95%" 2 /BOX 6')
C     DESHACEMOS LA ORIENTACION VERTICAL
      CALL GR_EXEC('SET ORIENTATION 0')
C     QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
      CALL GR_EXEC('TICK 1 2 1 2')
C     PONEMOS LIMITES REALES 8.55225=63/2/2*0.543
C     RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
      CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
C     DIBUJAMOS EL CONTORNO DEL PRIMARIO
      CALL GR_EXEC('ELLIPSE 13.7|2 /USER 0 0')
C     ENMARCAMOS EL DIBUJO
      CALL GR_EXEC('BOX')
      CALL GR_EXEC('LABEL "Metres" /Y')
C
C     CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
```



```

CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C
LO VOLVEMOS A PONER A SU VALOR USUAL
CALL GR_EXEC('SET CHAR .5')

C
AHORA LA FASE
C
DEFINIMOS LA MATRIZ DE CONVERSION DE FASE (VER HELP)
CONVP(1)=1.0      !REFERENCE PIXEL IN X
CONVP(2)=0.0      !X USER COORDINATE VALUE AT CONV(1)
CONVP(3)=1.0      !X USER COORDINATE INCREMENT PER PIXEL
CONVP(4)=1.0      !REFERENCE PIXEL IN Y
CONVP(5)=0.0      !Y USER COORDINATE VALUE AT CONV(4)
CONVP(6)=1.0      !Y USER COORDINATE INCREMENT PER PIXEL

C
PASAMOS EL SEGUNDO MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
C
CENTRAMOS LA CAJA EN LA PARTE INFERIOR DE LA PAGINA
CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
C
PONEMOS LIMITES EN PIXELES, DE MOMENTO
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C
ESTABLECEMOS NIVELES A DIBUJAR
CALL GR_EXEC('LEVELS -95 TO 95 BY 10')
C
LOS DIBUJAMOS
CALL GR_EXEC('RGM /PER 1.')
C
DIBUJAMOS ETIQUETA DE FASE
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "PHASE (radians)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 " 10% contours " 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 " -95% to +95% " 2 /BOX 6')
C
DESHACEMOS LA ORIENTACION VERTICAL
CALL GR_EXEC('SET ORIENTATION 0')
C
QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
CALL GR_EXEC('TICK 1 2 1 2')
C
PONEMOS LIMITES REALES  $8.55225=63/2/2*0.543$ 
C
RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
C
EL MARCO
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /X')
CALL GR_EXEC('LABEL "Metres" /Y')
C
EL CONTORNO DEL PRIMARIO
CALL GR_EXEC('ELLIPSE 13.7|2/USER 0 0')

C
C
Y POR ULTIMO MANDAMOS QUE SE IMPRIMA EN UN FICHERO .PS
CALL GR_EXEC('HARDCOPY AVMAP_CONTOUR.PS /DEV PS FAST')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C
CASE 2
C*****
CASE (2) DIBUJO
CALL GR_EXEC('CLEAR')
C
PONEMOS PAGINA NO APAISADA PARA SACAR
C
EN LA MISMA PAGINA MODULO Y FASE
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')

C
PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')      !TAMANIO LETRAS 0.5CM

C
DEFINIMOS LA MATRIZ DE CONVERSION DE AMPLITUD (VER HELP)
CONVA(1)=1.0      ! REFERENCE PIXEL IN X
CONVA(2)=0.0      ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL

C

```

```

C PASAMOS EL PRIMER MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
C CENTRAMOS LA CAJA EN LA PARTE SUPERIOR DE LA PAGINA
CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
C PONEMOS LIMITES EN PIXELES, DE MOMENTO
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C ESTABLECEMOS LOS NIVELES QUE QUEREMOS DIBUJAR
CALL GR_EXEC('LEVELS 5 TO 95 BY 10')
C LOS DIBUJAMOS
CALL GR_EXEC('RGM /PER 1.')
C DIBUJAMOS ETIQUETA DE AMPLITUD
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "AMPLITUDE (%)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 " 10% contours" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 " 5% to 95%" 2 /BOX 6')
C DESHACEMOS LA ORIENTACION VERTICAL
CALL GR_EXEC('SET ORIENTATION 0')
C QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
CALL GR_EXEC('TICK 1 2 1 2')
C PONEMOS LIMITES REALES  $8.55225=63/2/2*0.543$ 
C RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
C DIBUJAMOS EL CONTORNO DEL PRIMARIO
CALL GR_EXEC('ELLIPSE 13.7|2 /USER 0 0')
C ENMARCAMOS EL DIBUJO
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /Y')
C
C CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C LO VOLVEMOS A PONER A SU VALOR USUAL
CALL GR_EXEC('SET CHAR .5')
C
C AHORA LA FASE
C DEFINIMOS LA MATRIZ DE CONVERSION DE FASE (VER HELP)
CONVP(1)=1.0 !REFERENCE PIXEL IN X
CONVP(2)=0.0 !X USER COORDINATE VALUE AT CONV(1)
CONVP(3)=1.0 !X USER COORDINATE INCREMENT PER PIXEL
CONVP(4)=1.0 !REFERENCE PIXEL IN Y
CONVP(5)=0.0 !Y USER COORDINATE VALUE AT CONV(4)
CONVP(6)=1.0 !Y USER COORDINATE INCREMENT PER PIXEL
C
C PASAMOS EL SEGUNDO MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
C CENTRAMOS LA CAJA EN LA PARTE INFERIOR DE LA PAGINA
CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
C PONEMOS LIMITES EN PIXELES, DE MOMENTO
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C ESTABLECEMOS NIVELES A DIBUJAR
CALL GR_EXEC('LEVELS -95 TO 95 BY 10')
C LOS DIBUJAMOS
CALL GR_EXEC('RGM /PER 1.')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "PHASE (radians)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 " 10% contours" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 " -95% to +95%" 2 /BOX 6')
C DESHACEMOS LA ORIENTACION VERTICAL
CALL GR_EXEC('SET ORIENTATION 0')
C QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
CALL GR_EXEC('TICK 1 2 1 2')
C PONEMOS LIMITES REALES  $8.55225=63/2/2*0.543$ 
C RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
C EL MARCO
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /X')
CALL GR_EXEC('LABEL "Metres" /Y')

```

```

C      EL CONTORNO DEL PRIMARIO
      CALL GR_EXEC('ELLIPSE 13.7|2/USER 0 0')
C
C      Y POR ULTIMO MANDAMOS QUE SE IMPRIMA EN UN FICHERO .PS
      CALL GR_EXEC('HARDCOPY PICOMAP_CONTOUR.PS /DEV PS FAST')
      CALL GR_EXEC('CLEAR')
      RETURN
C*****
C                      CASE 3
C*****
      CASE (3) DIBUJO
      CALL GR_EXEC('CLEAR')
      CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
      CALL GR_EXEC('SET FONT DUPLEX')
      CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C      DEFINIMOS LA MATRIZ DE CONVERSION (VER HELP)
      CONVA(1)=1.0          ! REFERENCE PIXEL IN X
      CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
      CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
      CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
      CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
      CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
      CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
      CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
      CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
      CALL GR_EXEC('LEVELS -95 TO 95 BY 10')
      CALL GR_EXEC('RGM /PER 1.')
      CALL GR_EXEC('SET ORIENTATION 90')
      CALL GR_EXEC('DRAW TEXT 1 0 "RESIDUAL AMPLITUDE (%)" 2 /BOX 6')
      CALL GR_EXEC('DRAW TEXT 2 0 "      10% contours      " 2 /BOX 6')
      CALL GR_EXEC('DRAW TEXT 3 0 "      -95% to +95%      " 2 /BOX 6')
      CALL GR_EXEC('SET ORIENTATION 0')
      CALL GR_EXEC('TICK 5 10 5 10')
      CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')
      CALL GR_EXEC('BOX')
      CALL GR_EXEC('LABEL "Arcmin" /X')
      CALL GR_EXEC('LABEL "Arcmin" /Y')
C
      CALL GR_EXEC('SET CHAR 0.65')
      CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
      CALL GR_EXEC('SET CHAR .5')
      CALL GR_EXEC('HARDCOPY RESID_CONTOUR.PS /DEV PS FAST')
      CALL GR_EXEC('CLEAR')
      RETURN
C*****
C                      CASE 4
C*****
      CASE (4) DIBUJO
      CALL GR_EXEC('CLEAR')
      CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
      CALL GR_EXEC('SET FONT DUPLEX')
      CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C      DEFINIMOS LA MATRIZ DE CONVERSION (VER HELP)
      CONVA(1)=1.0          ! REFERENCE PIXEL IN X
      CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
      CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
      CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
      CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
      CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
C      PASAMOS EL PRIMER MAPA
      CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
      CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
      CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')

```

```

CALL GR_EXEC('LEVELS 0.0001 0.001 0.01 0.1 1 10 100')
CALL GR_EXEC('RGM /PER 1.')
CALL GR_EXEC('LEVELS 50')
CALL GR_EXEC('PENCIL 0 /WEIGHT 3')
CALL GR_EXEC('RGM /PEN /PER 1.')
CALL GR_EXEC('PENCIL 0 /WEIGHT 1')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "INFOC. POWER PATTERN (dB)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 "      10dB contours      " 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 "      -60dB to 0dB      " 2 /BOX 6')
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 5 10 5 10')
CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Arcmin" /Y')
C CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C LO VOLVEMOS A PONER A SU VALOR USUAL
CALL GR_EXEC('SET CHAR .5')

C DEFINIMOS LA MATRIZ DE CONVERSION (VER HELP)
CONVP(1)=1.0      !REFERENCE PIXEL IN X
CONVP(2)=0.0      !X USER COORDINATE VALUE AT CONV(1)
CONVP(3)=1.0      !X USER COORDINATE INCREMENT PER PIXEL
CONVP(4)=1.0      !REFERENCE PIXEL IN Y
CONVP(5)=0.0      !Y USER COORDINATE VALUE AT CONV(4)
CONVP(6)=1.0      !Y USER COORDINATE INCREMENT PER PIXEL

C
C PASAMOS EL SEGUNDO MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('LEVELS 0.0001 0.001 0.01 0.1 1 10 100')
CALL GR_EXEC('RGM /PER 1.')
CALL GR_EXEC('LEVELS 50')
CALL GR_EXEC('PENCIL 0 /WEIGHT 3')
CALL GR_EXEC('RGM /PEN /PER 1.')
CALL GR_EXEC('PENCIL 0 /WEIGHT 1')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "DEFOC. POWER PATTERN (dB)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 "      10dB contours      " 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 "      -60dB to 0dB      " 2 /BOX 6')
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 5 10 5 10')
CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Arcmin" /X')
CALL GR_EXEC('LABEL "Arcmin" /Y')

C
C Y POR ULTIMO MANDAMOS QUE SE IMPRIMA EN UN FICHERO .PS
CALL GR_EXEC('HARDCOPY FARFIELDS_CONTOUR.PS /DEV PS FAST')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C                      CASE 5
C*****
CASE (5) DIBUJO
CALL GR_EXEC('CLEAR')
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM

C
C DEFINIMOS LA MATRIZ DE CONVERSION (VER HELP)
CONVA(1)=1.0      ! REFERENCE PIXEL IN X
CONVA(2)=0.0      ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0      ! REFERENCE PIXEL IN Y

```

```

CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('LEVELS -95 TO 95 BY 10')
CALL GR_EXEC('RGM /PER 1.')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "PHASE DIFFERENCE (%)" 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 2 0 " 10% contours      " 2 /BOX 6')
CALL GR_EXEC('DRAW TEXT 3 0 " -95% to +95%      " 2 /BOX 6')
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 1 2 1 2')
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /X')
CALL GR_EXEC('LABEL "Metres" /Y')
C
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
CALL GR_EXEC('SET CHAR .5')
CALL GR_EXEC('HARDCOPY MAPERROR_CONTOUR.PS /DEV PS FAST')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C                                CASE DEFAULT
C*****
CASE DEFAULT DIBUJO
WRITE(6,*) 'IDENTIFICADOR DE DIBUJO INCORRECTO'
WRITE(6,*) 'ERROR EN SUBROUTINA GREG_CONTOUR.F'
RETURN
C*****
C                                FIN DEL CASE
C*****
END SELECT DIBUJO
END

```

## Anexo XXVI: Subrutina greg\_curve.f

```

SUBROUTINE GREG_CURVE (AX,AY,NDATA, IDEN)
C   DIBUJA LAS CURVAS AY VS. AX
C   N ES LA DIMENSION DE LOS ARRAYS
C
C   EL ENTERO IDEN IDENTIFICA EL TIPO DE MAPA A REALIZAR
C   1 PARA GRAFICAS DE MAPERROR
C   2 LA QUE TU QUIERAS ANIADIR POSTERIORMENTE.
C
C   INTEGER NDATA
C   REAL AX (NDATA) ,AY (NDATA)
C   INTEGER IDEN
C
C   DIBUJO:      SELECT CASE (IDEN)
C*****
C               CASO 1
C*****
C               CASE (1) DIBUJO
C               CALL GR_EXEC('CLEAR')
C               CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C               CALL GR_EXEC('SET FONT DUPLEX')
C               CALL GR_EXEC('SET CHARACTER 0.5')
C
C               CALL GR4_GIVE('X',NDATA,AX)
C               CALL GR4_GIVE('Y',NDATA,AY)
C               CENTRAMOS LA CAJA EN LA PARTE SUPERIOR DE LA PAGINA
C               CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
C               CALL GR_EXEC('LIMITS')
C               CALL GR_EXEC('BOX')
C               CALL GR_EXEC('CONNECT')
C               CALL GR_EXEC('LABEL "Normalised Radius" /X')
C               CALL GR_EXEC('LABEL "RMS Error (radians)" /Y')
C
C               CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
C               CALL GR_EXEC('SET CHAR 0.65')
C               CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C               LO VOLVEMOS A PONER A SU VALOR USUAL
C               CALL GR_EXEC('SET CHAR .5')
C               CALL GR_EXEC('HARDCOPY RMSRAD_ERROR.PS /DEV PS FAST')
C               CALL GR_EXEC('CLEAR')
C               RETURN
C*****
C               CASO 2
C*****
C               CASE (2) DIBUJO
C               CALL GR_EXEC('CLEAR')
C               CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C               CALL GR_EXEC('SET FONT DUPLEX')
C               CALL GR_EXEC('SET CHARACTER 0.5')
C
C               CALL GR4_GIVE('X',NDATA,AX)
C               CALL GR4_GIVE('Y',NDATA,AY)
C               CENTRAMOS LA CAJA EN LA PARTE SUPERIOR DE LA PAGINA
C               CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
C               CALL GR_EXEC('LIMITS')
C               CALL GR_EXEC('BOX')
C               CALL GR_EXEC('CONNECT')
C               CALL GR_EXEC('LABEL "Normalised Radius" /X')
C               CALL GR_EXEC('LABEL "RMS Error (um)" /Y')
C
C               CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
C               CALL GR_EXEC('SET CHAR 0.65')
C               CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')

```

```

C      LO VOLVEMOS A PONER A SU VALOR USUAL
      CALL GR_EXEC('SET CHAR .5')
      CALL GR_EXEC('HARDCOPY RMSMIC_ERROR.PS /DEV PS FAST')
      CALL GR_EXEC('CLEAR')
      RETURN
C*****
C                                 CASE DEFAULT
C*****
      CASE DEFAULT DIBUJO
      WRITE(6,*) 'IDENTIFICADOR DE DIBUJO INCORRECTO'
      WRITE(6,*) 'ERROR EN SUBROUTINA GREG_CURVE.F'
      RETURN
C*****
C                                 FIN DEL CASE
C*****
      END SELECT DIBUJO
      END

```

## Anexo XXVII: Subrutina greg\_grey.f

```

SUBROUTINE GREG_GREY (MAPAA, MAPAP, MDIM, IDEN)
C   DIBUJA DOS MAPAS EN NIVELES DE GRIS
C   AMBOS MAPAS DEBEN SER DE 64x64
C
C   EL ENTERO IDEN IDENTIFICA EL TIPO DE MAPA A REALIZAR
C   1 PARA GRAFICAS DE AVMAP
C   2 PARA GRAFICAS DE AMPL+PHASE CON PICOMISTY
C   3 PARA AMPLITUD RESIDUALS DE PICOMISTY
C   4 FOC. FAR FIELD AND DEFOC. FAR FIELD DE PICOMISTY
C   5 PARA MAPERROR
C   6 PARA MAPERROR-DATA USED IN RMS CALCULATIONS
C   5 LA QUE TU QUIERAS ANIADIR POSTERIORMENTE.
C
REAL MAPAA (MDIM, MDIM) , MAPAP (MDIM, MDIM)
REAL*8 CONVA (6) , CONV (6)
INTEGER IDEN
C
C   DIBUJO:          SELECT CASE (IDEN)
C*****
C                   CASO 1
C*****
C   CASE (1) DIBUJO
C   CALL GR_EXEC ('CLEAR')
C   PONEMOS PAGINA NO APAISADA PARA SACAR
C   EN LA MISMA PAGINA MODULO Y FASE
C   CALL GR_EXEC ('SET PLOT_PAGE PORTRAIT')
C
C   PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
C   CALL GR_EXEC ('SET FONT DUPLEX')
C   CALL GR_EXEC ('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C   DEFINIMOS LA MATRIZ DE CONVERSION DE AMPLITUD (VER HELP)
C   CONVA (1)=1.0          ! REFERENCE PIXEL IN X
C   CONVA (2)=0.0          ! X USER COORDINATE VALUE AT CONV (1)
C   CONVA (3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
C   CONVA (4)=1.0          ! REFERENCE PIXEL IN Y
C   CONVA (5)=0.0          ! Y USER COORDINATE VALUE AT CONV (4)
C   CONVA (6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
C   PASAMOS EL PRIMER MAPA
C   CALL GR4_RGIVE (MDIM, MDIM, CONVA, MAPAA)
C   CENTRAMOS LA CAJA EN LA PARTE SUPERIOR DE LA PAGINA
C   CALL GR_EXEC ('SET BOX_LOCATION 4 16 15.5 27.5')
C   PONEMOS LIMITES EN PIXELES, DE MOMENTO
C   CALL GR_EXEC ('LIMITS -0.5 62.5 -0.5 62.5')
C   DIBUJAMOS LA ESCALA EN %
C   EL PRIMER PLOT NO LO UTILIZAMOS PARA ENGANIAR A WEDGE
C   CALL GR_EXEC ('PLOT /SCAL LIN 5 100')
C   CALL GR_EXEC ('WEDGE')
C   DIBUJAMOS DE 0.05% EN ADELANTE
C   RGMAX ES UNA VARIABLE DE GREG QUE CONTIENE EL VALOR MAXIMO
C   DEL MAPA
C   CALL GR_EXEC ('PLOT /SCAL LIN 0.05*RGMAX RGMAX')
C   DIBUJAMOS ETIQUETA DE AMPLITUD
C   CALL GR_EXEC ('SET ORIENTATION 90')
C   CALL GR_EXEC ('DRAW TEXT 2.5 -0 "AMPLITUDE (%)" 2 /BOX 6')
C   DESHACEMOS LA ORIENTACION VERTICAL
C   CALL GR_EXEC ('SET ORIENTATION 0')
C   QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
C   CALL GR_EXEC ('TICK 1 2 1 2')
C   PONEMOS LIMITES REALES  $8.55225=63/2/2*0.543$ 
C   RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
C   CALL GR_EXEC ('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
C   DIBUJAMOS EL CONTORNO DEL PRIMARIO

```



```

CALL GR_EXEC('ELLIPSE 13.7|2 /USER 0 0')
C
ENMARCAMOS EL DIBUJO
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /Y')
C
CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C
LO VOLVEMOS A PONER A SU VALOR USUAL
CALL GR_EXEC('SET CHAR .5')

C
AHORA LA FASE
C
DEFINIMOS LA MATRIZ DE CONVERSION DE FASE (VER HELP)
CONVP(1)=1.0      !REFERENCE PIXEL IN X
CONVP(2)=0.0      !X USER COORDINATE VALUE AT CONV(1)
CONVP(3)=1.0      !X USER COORDINATE INCREMENT PER PIXEL
CONVP(4)=1.0      !REFERENCE PIXEL IN Y
CONVP(5)=0.0      !Y USER COORDINATE VALUE AT CONV(4)
CONVP(6)=1.0      !Y USER COORDINATE INCREMENT PER PIXEL

C
PASAMOS EL SEGUNDO MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
C
CENTRAMOS LA CAJA EN LA PARTE INFERIOR DE LA PAGINA
CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
C
PONEMOS LIMITES EN PIXELES, DE MOMENTO
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C
DIBUJAMOS FASES ENTRE -2.5RAD Y +2.5RAD
CALL GR_EXEC('PLOT /SCAL LIN -2.5 2.5')
CALL GR_EXEC('WEDGE')
C
DIBUJAMOS ETIQUETA DE FASE
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 2.5 -0 "PHASE (rad)" 2 /BOX 6')
C
DESHACEMOS LA ORIENTACION VERTICAL
CALL GR_EXEC('SET ORIENTATION 0')
C
QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
CALL GR_EXEC('TICK 1 2 1 2')
C
PONEMOS LIMITES REALES 8.55225=63/2/2*0.543
C
RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
C
EL MARCO
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /X')
CALL GR_EXEC('LABEL "Metres" /Y')
C
EL CONTORNO DEL PRIMARIO
CALL GR_EXEC('ELLIPSE 13.7|2/USER 0 0')

C
Y POR ULTIMO MANDAMOS QUE SE IMPRIMA EN UN FICHERO .PS
CALL GR_EXEC('HARDCOPY AVMAP_GREY.PS /DEV PS GREY')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C
CASE 2
C*****
CASE (2) DIBUJO
CALL GR_EXEC('CLEAR')
C
PONEMOS PAGINA NO APAISADA PARA SACAR
EN LA MISMA PAGINA MODULO Y FASE
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')

C
PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')      !TAMANIO LETRAS 0.5CM

C
DEFINIMOS LA MATRIZ DE CONVERSION DE AMPLITUD (VER HELP)
CONVA(1)=1.0      ! REFERENCE PIXEL IN X
CONVA(2)=0.0      ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)

```

```

CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
C PASAMOS EL PRIMER MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
C CENTRAMOS LA CAJA EN LA PARTE SUPERIOR DE LA PAGINA
CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
C PONEMOS LIMITES EN PIXELES, DE MOMENTO
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C DIBUJAMOS LA ESCALA EN %
C EL PRIMER PLOT NO LO UTILIZAMOS PARA ENGANIAR A WEDGE
CALL GR_EXEC('PLOT /SCAL LIN 5 100')
CALL GR_EXEC('WEDGE')
C DIBUJAMOS DE 0.05% EN ADELANTE
C RGMX ES UNA VARIABLE DE GREG QUE CONTIENE EL VALOR MAXIMO
C DEL MAPA
CALL GR_EXEC('PLOT /SCAL LIN 0.05*RGMAX RGMAX')
C DIBUJAMOS ETIQUETA DE AMPLITUD
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 2.5 -0 "AMPLITUDE (%)" 2 /BOX 6')
C DESHACEMOS LA ORIENTACION VERTICAL
CALL GR_EXEC('SET ORIENTATION 0')
C QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
CALL GR_EXEC('TICK 1 2 1 2')
C PONEMOS LIMITES REALES 8.55225=63/2/2*0.543
C RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
C DIBUJAMOS EL CONTORNO DEL PRIMARIO
CALL GR_EXEC('ELLIPSE 13.7|2 /USER 0 0')
C ENMARCAMOS EL DIBUJO
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /Y')
C CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C LO VOLVEMOS A PONER A SU VALOR USUAL
CALL GR_EXEC('SET CHAR .5')

C
C AHORA LA FASE
C DEFINIMOS LA MATRIZ DE CONVERSION DE FASE (VER HELP)
CONVP(1)=1.0          !REFERENCE PIXEL IN X
CONVP(2)=0.0          !X USER COORDINATE VALUE AT CONV(1)
CONVP(3)=1.0          !X USER COORDINATE INCREMENT PER PIXEL
CONVP(4)=1.0          !REFERENCE PIXEL IN Y
CONVP(5)=0.0          !Y USER COORDINATE VALUE AT CONV(4)
CONVP(6)=1.0          !Y USER COORDINATE INCREMENT PER PIXEL

C
C PASAMOS EL SEGUNDO MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
C CENTRAMOS LA CAJA EN LA PARTE INFERIOR DE LA PAGINA
CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
C PONEMOS LIMITES EN PIXELES, DE MOMENTO
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C DIBUJAMOS FASES ENTRE -2.5RAD Y +2.5RAD
CALL GR_EXEC('PLOT /SCAL LIN -2.5 2.5')
CALL GR_EXEC('WEDGE')
C DIBUJAMOS ETIQUETA DE FASE
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 2.5 -0 "PHASE (rad)" 2 /BOX 6')
C DESHACEMOS LA ORIENTACION VERTICAL
CALL GR_EXEC('SET ORIENTATION 0')
C QUEREMOS EJES SEPARADOS UNA UNIDAD PERO INDICANDO EL No. CADA 2
CALL GR_EXEC('TICK 1 2 1 2')
C PONEMOS LIMITES REALES 8.55225=63/2/2*0.543
C RECORDAR QUE 0.543 ES LA RESOLUCION ESPACIAL EN METROS
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
C EL MARCO
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /X')

```

```

CALL GR_EXEC('LABEL "Metres" /Y')
C
EL CONTORNO DEL PRIMARIO
CALL GR_EXEC('ELLIPSE 13.7|2/USER 0 0')
C
C
Y POR ULTIMO MANDAMOS QUE SE IMPRIMA EN UN FICHERO .PS
CALL GR_EXEC('HARDCOPY PICOMAP_GREY.PS /DEV PS GREY')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C
CASE 3
C*****
CASE (3) DIBUJO
CALL GR_EXEC('CLEAR')
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C
DEFINIMOS LA MATRIZ DE CONVERSION (VER HELP)
CONVA(1)=1.0          ! REFERENCE PIXEL IN X
CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN -100 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN -1.0*RGMAX RGMAX')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 2.5 0 "RESIDUAL AMPLITUDE (%)" 2 /BOX 6')
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 5 10 5 10')
CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Arcmin" /X')
CALL GR_EXEC('LABEL "Arcmin" /Y')
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
CALL GR_EXEC('SET CHAR .5')
CALL GR_EXEC('HARDCOPY RESID_GREY.PS /DEV PS GREY')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C
CASE 4
C*****
CASE (4) DIBUJO
CALL GR_EXEC('CLEAR')
C
PONEMOS PAGINA NO APAISADA PARA SACAR
C
EN LA MISMA PAGINA MODULO Y FASE
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C
PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C
DEFINIMOS LA MATRIZ DE CONVERSION DE AMPLITUD (VER HELP)
CONVA(1)=1.0          ! REFERENCE PIXEL IN X
CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
C
PASAMOS EL PRIMER MAPA
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)

```

```

C      CENTRAMOS LA CAJA EN LA PARTE SUPERIOR DE LA PAGINA
      CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
C      PONEMOS LIMITES EN PIXELES, DE MOMENTO
      CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C      DIBUJAMOS LA ESCALA EN %
C      EL PRIMER PLOT NO LO UTILIZAMOS PARA ENGANIAR A WEDGE
      CALL GR_EXEC('PLOT /SCAL LIN 0 100')
      CALL GR_EXEC('WEDGE')
C      RGMAX ES UNA VARIABLE DE GREG QUE CONTIENE EL VALOR MAXIMO
C      DEL MAPA
      CALL GR_EXEC('PLOT /SCAL LIN 0 RGMAX')
C      DIBUJAMOS ETIQUETA DE AMPLITUD
      CALL GR_EXEC('SET ORIENTATION 90')
      CALL GR_EXEC('DRAW TEXT 2.5 0 "INFOCUSSED POWER" 2 /BOX 6')
      CALL GR_EXEC('DRAW TEXT 3.5 0 " PATTERN (%)" 2 /BOX 6')
C      DESHACEMOS LA ORIENTACION VERTICAL
      CALL GR_EXEC('SET ORIENTATION 0')
      CALL GR_EXEC('TICK 5 10 5 10')
      CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')
      CALL GR_EXEC('BOX')
      CALL GR_EXEC('LABEL "Arcmin" /Y')
C      CAMBIAMOS TAMANIO DE TEXTO PARA TITULO DEL DIBUJO
      CALL GR_EXEC('SET CHAR 0.65')
      CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C      LO VOLVEMOS A PONER A SU VALOR USUAL
      CALL GR_EXEC('SET CHAR .5')
C
C      DEFINIMOS LA MATRIZ DE CONVERSION DE FASE (VER HELP)
      CONVP(1)=1.0      !REFERENCE PIXEL IN X
      CONVP(2)=0.0      !X USER COORDINATE VALUE AT CONV(1)
      CONVP(3)=1.0      !X USER COORDINATE INCREMENT PER PIXEL
      CONVP(4)=1.0      !REFERENCE PIXEL IN Y
      CONVP(5)=0.0      !Y USER COORDINATE VALUE AT CONV(4)
      CONVP(6)=1.0      !Y USER COORDINATE INCREMENT PER PIXEL
C
C      PASAMOS EL SEGUNDO MAPA
      CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
C      CENTRAMOS LA CAJA EN LA PARTE INFERIOR DE LA PAGINA
      CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
C      PONEMOS LIMITES EN PIXELES, DE MOMENTO
      CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
      CALL GR_EXEC('PLOT /SCAL LIN 0 100')
      CALL GR_EXEC('WEDGE')
      CALL GR_EXEC('PLOT /SCAL LIN 0 RGMAX')
C      DIBUJAMOS ETIQUETA DE FASE
      CALL GR_EXEC('SET ORIENTATION 90')
      CALL GR_EXEC('DRAW TEXT 2.5 0 "DEFOCUSSED POWER" 2 /BOX 6')
      CALL GR_EXEC('DRAW TEXT 3.5 0 " PATTERN (%)" 2 /BOX 6')
C      DESHACEMOS LA ORIENTACION VERTICAL
      CALL GR_EXEC('SET ORIENTATION 0')
      CALL GR_EXEC('TICK 5 10 5 10')
      CALL GR_EXEC('LIMITS -38.4 +38.4 -38.4 +38.4')
C      EL MARCO
      CALL GR_EXEC('BOX')
      CALL GR_EXEC('LABEL "Arcmin" /X')
      CALL GR_EXEC('LABEL "Arcmin" /Y')
C
      CALL GR_EXEC('HARDCOPY FARFIELDS_GREY.PS /DEV PS GREY')
      CALL GR_EXEC('CLEAR')
      RETURN
C*****
C      CASE 5
C*****
      CASE (5) DIBUJO
      CALL GR_EXEC('CLEAR')
      CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
      CALL GR_EXEC('SET FONT DUPLEX')
      CALL GR_EXEC('SET CHARACTER 0.5')      !TAMANIO LETRAS 0.5CM

```

```

C
C   DEFINIMOS LA MATRIZ DE CONVERSION (VER HELP)
CONVA(1)=1.0      ! REFERENCE PIXEL IN X
CONVA(2)=0.0      ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN -100 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN -1.0*RGMAX RGMAX')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 2.5 0 "PHASE DIFFERENCE (%)" 2 /BOX 6')
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 1 2 1 2')
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /X')
CALL GR_EXEC('LABEL "Metres" /Y')
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
CALL GR_EXEC('SET CHAR .5')
CALL GR_EXEC('HARDCOPY MAPERROR_GREY.PS /DEV PS GREY')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C                               CASE 6
C*****
CASE (6) DIBUJO
CALL GR_EXEC('CLEAR')
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')           !TAMANIO LETRAS 0.5CM
C
C   DEFINIMOS LA MATRIZ DE CONVERSION (VER HELP)
CONVA(1)=1.0      ! REFERENCE PIXEL IN X
CONVA(2)=0.0      ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN -100 100')
CALL GR_EXEC('PLOT /SCAL LIN -1.0*RGMAX RGMAX')
CALL GR_EXEC('SET ORIENTATION 90')
CALL GR_EXEC('DRAW TEXT 1 0 "DATA USED IN RMS CALC." 2 /BOX 6')
CALL GR_EXEC('SET ORIENTATION 0')
CALL GR_EXEC('TICK 1 2 1 2')
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
CALL GR_EXEC('BOX')
CALL GR_EXEC('LABEL "Metres" /X')
CALL GR_EXEC('LABEL "Metres" /Y')
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
CALL GR_EXEC('SET CHAR .5')
CALL GR_EXEC('HARDCOPY DATA_IN_RMS.PS /DEV PS GREY')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C                               CASE DEFAULT
C*****

```

```
      CASE DEFAULT DIBUJO
      WRITE(6,*) 'IDENTIFICADOR DE DIBUJO INCORRECTO'
      WRITE(6,*) 'ERROR EN SUBROUTINA GREG_GREY.F'
      RETURN
C*****
C                               FIN DEL CASE
C*****
      END SELECT DIBUJO
      END
```

## Anexo XXVIII: Subrutina greg\_pers.f

```

SUBROUTINE GREG_PERS (MAPAA, MAPAP, MDIM, IDEN)
C   DIBUJA DOS MAPAS EN PERSPECTIVA
C   AMBOS MAPAS DEBEN SER DE 64x64
C
C   EL ENTERO IDEN IDENTIFICA EL TIPO DE MAPA A REALIZAR
C   1 PARA GRAFICAS DE AVMAP
C   2 PARA GRAFICAS DE AMPL+PHASE CON PICOMISTY
C   3 PARA AMPLITUD RESIDUALS DE PICOMISTY
C   4 FOC. FAR FIELD AND DEFOC. FAR FIELD DE PICOMISTY
C   5 PARA MAPERROR
C   6 LA QUE TU QUIERAS ANIADIR POSTERIORMENTE.
C
REAL MAPAA (MDIM, MDIM), MAPAP (MDIM, MDIM)
REAL*8 CONVA (6), CONVP (6)
INTEGER IDEN
C
C   DIBUJO:          SELECT CASE (IDEN)
C*****
C                   CASO 1
C*****
C   CASE (1) DIBUJO
C   CALL GR_EXEC('CLEAR')
C   PONEMOS PAGINA NO APAISADA PARA SACAR
C   EN LA MISMA PAGINA MODULO Y FASE
C   CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C   PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
C   CALL GR_EXEC('SET FONT DUPLEX')
C   CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C   CONVA(1)=1.0          ! REFERENCE PIXEL IN X
C   CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
C   CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
C   CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
C   CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
C   CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
C   CALL GR4_RGIVE (MDIM, MDIM, CONVA, MAPAA)
C   CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
C   CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C   CALL GR_EXEC('PERSPECTIVE / CENTER')
C   CALL GR_EXEC('DRAW TEXT 0 0 "AMPLITUDE" 2 /BOX 2')
C   CALL GR_EXEC('SET CHAR 0.65')
C   CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C   CALL GR_EXEC('SET CHAR .5')
C
C   AHORA LA FASE
C
C   CONVP(1)=1.0          !REFERENCE PIXEL IN X
C   CONVP(2)=0.0          !X USER COORDINATE VALUE AT CONV(1)
C   CONVP(3)=1.0          !X USER COORDINATE INCREMENT PER PIXEL
C   CONVP(4)=1.0          !REFERENCE PIXEL IN Y
C   CONVP(5)=0.0          !Y USER COORDINATE VALUE AT CONV(4)
C   CONVP(6)=1.0          !Y USER COORDINATE INCREMENT PER PIXEL
C
C   CALL GR4_RGIVE (MDIM, MDIM, CONVP, MAPAP)
C   CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
C   CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C   CALL GR_EXEC('PERSPECTIVE /CENTER')
C   CALL GR_EXEC('DRAW TEXT 0 +1.5 "PHASE" 2 /BOX 2')
C
C   CALL GR_EXEC('HARDCOPY AVMAP_PERSPEC.PS /DEV PS FAST')
C   CALL GR_EXEC('CLEAR')
RETURN

```

```

C*****
C
C          CASE 2
C*****
C          CASE (2) DIBUJO
C          CALL GR_EXEC('CLEAR')
C          PONEMOS PAGINA NO APAISADA PARA SACAR
C          EN LA MISMA PAGINA MODULO Y FASE
C          CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C          PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
C          CALL GR_EXEC('SET FONT DUPLEX')
C          CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C          CONVA(1)=1.0          ! REFERENCE PIXEL IN X
C          CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
C          CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
C          CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
C          CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
C          CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
C          CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
C          CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
C          CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C          CALL GR_EXEC('PERSPECTIVE / CENTER')
C          CALL GR_EXEC('DRAW TEXT 0 0 "AMPLITUDE" 2 /BOX 2')
C          CALL GR_EXEC('SET CHAR 0.65')
C          CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
C          CALL GR_EXEC('SET CHAR .5')
C
C          AHORA LA FASE
C
C          CONVP(1)=1.0          !REFERENCE PIXEL IN X
C          CONVP(2)=0.0          !X USER COORDINATE VALUE AT CONV(1)
C          CONVP(3)=1.0          !X USER COORDINATE INCREMENT PER PIXEL
C          CONVP(4)=1.0          !REFERENCE PIXEL IN Y
C          CONVP(5)=0.0          !Y USER COORDINATE VALUE AT CONV(4)
C          CONVP(6)=1.0          !Y USER COORDINATE INCREMENT PER PIXEL
C
C          CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
C          CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
C          CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
C          CALL GR_EXEC('PERSPECTIVE /CENTER')
C          CALL GR_EXEC('DRAW TEXT 0 0.5 "PHASE" 2 /BOX 2')
C
C          CALL GR_EXEC('HARDCOPY PICOMAP_PERSPEC.PS /DEV PS FAST')
C          CALL GR_EXEC('CLEAR')
C          RETURN
C*****
C          CASE 3
C*****
C          CASE (3) DIBUJO
C          CALL GR_EXEC('CLEAR')
C          PONEMOS PAGINA NO APAISADA PARA SACAR
C          EN LA MISMA PAGINA MODULO Y FASE
C          CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C          PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
C          CALL GR_EXEC('SET FONT DUPLEX')
C          CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
C          CONVA(1)=1.0          ! REFERENCE PIXEL IN X
C          CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
C          CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
C          CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
C          CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
C          CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
C          CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)

```



```

CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PERSPECTIVE / CENTER')
CALL GR_EXEC('DRAW TEXT 0 1 "RESIDUAL AMPLITUDE" 2 /BOX 2')
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
CALL GR_EXEC('SET CHAR .5')
CALL GR_EXEC('HARDCOPY RESID_PERSPEC.PS /DEV PS COLOR')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C                               CASE 4
C*****
CASE (4) DIBUJO
CALL GR_EXEC('CLEAR')
C   PONEMOS PAGINA NO APAISADA PARA SACAR
C   EN LA MISMA PAGINA MODULO Y FASE
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C
C   PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
CALL GR_EXEC('SET FONT DUPLEX')
CALL GR_EXEC('SET CHARACTER 0.5')           !TAMANIO LETRAS 0.5CM
C
CONVA(1)=1.0           ! REFERENCE PIXEL IN X
CONVA(2)=0.0           ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0           ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0           ! REFERENCE PIXEL IN Y
CONVA(5)=0.0           ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0           ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('SET BOX_LOCATION 4 16 15.5 27.5')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PERSPECTIVE / CENTER')
CALL GR_EXEC('DRAW TEXT 0 0.5 "INFOCUSSED POWER" 2 /BOX 2')
CALL GR_EXEC('DRAW TEXT 0 -0.5 " PATTERN (%) " 2 /BOX 2')
CALL GR_EXEC('SET CHAR 0.65')
CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
CALL GR_EXEC('SET CHAR .5')
C
AHORA LA FASE
CONVP(1)=1.0           !REFERENCE PIXEL IN X
CONVP(2)=0.0           !X USER COORDINATE VALUE AT CONV(1)
CONVP(3)=1.0           !X USER COORDINATE INCREMENT PER PIXEL
CONVP(4)=1.0           !REFERENCE PIXEL IN Y
CONVP(5)=0.0           !Y USER COORDINATE VALUE AT CONV(4)
CONVP(6)=1.0           !Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR4_RGIVE(MDIM,MDIM,CONVP,MAPAP)
CALL GR_EXEC('SET BOX_LOCATION 4 16 2 14')
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PERSPECTIVE /CENTER')
CALL GR_EXEC('DRAW TEXT 0 0.5 "DEFOCUSSED POWER" 2 /BOX 2')
CALL GR_EXEC('DRAW TEXT 0 -0.5 " PATTERN (%) " 2 /BOX 2')
C
CALL GR_EXEC('HARDCOPY FARFIELDS_PERSPEC.PS /DEV PS FAST')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C                               CASE 5
C*****
CASE (5) DIBUJO
CALL GR_EXEC('CLEAR')
C   PONEMOS PAGINA NO APAISADA PARA SACAR
C   EN LA MISMA PAGINA MODULO Y FASE
CALL GR_EXEC('SET PLOT_PAGE PORTRAIT')
C

```

```

C     PONEMOS LETRAS DE CALIDAD SUPERIOR, QUE MENOS!
      CALL GR_EXEC('SET FONT DUPLEX')
      CALL GR_EXEC('SET CHARACTER 0.5')          !TAMANIO LETRAS 0.5CM
C
      CONVA(1)=1.0          ! REFERENCE PIXEL IN X
      CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
      CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
      CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
      CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
      CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
      CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
      CALL GR_EXEC('SET BOX_LOCATION 2.5 16.5 6.8 20.8')
      CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
      CALL GR_EXEC('PERSPECTIVE / CENTER')
      CALL GR_EXEC('DRAW TEXT 0 1 "PHASE DIFFERENCE MAP" 2 /BOX 2')
      CALL GR_EXEC('SET CHAR 0.65')
      CALL GR_EXEC('DRAW TEXT 0 0.3 "14m CAY RADIOTELESCOPE" 8 /BOX 8')
      CALL GR_EXEC('SET CHAR .5')
      CALL GR_EXEC('HARDCOPY MAPERROR_PERSPEC.PS /DEV PS COLOR')
      CALL GR_EXEC('CLEAR')
      RETURN
C*****
C                                 CASE DEFAULT
C*****
      CASE DEFAULT DIBUJO
      WRITE(6,*) 'IDENTIFICADOR DE DIBUJO INCORRECTO'
      WRITE(6,*) 'ERROR EN SUBROUTINA GREG_PERS.F'
      RETURN
C*****
C                                 FIN DEL CASE
C*****
      END SELECT DIBUJO
      END

```

## Anexo XXIX: Subrutina legblock.f

```
      SUBROUTINE LEGBLOCK (PSI, MDIM, SAMPI, DIAM, DLEG1,
1 RLEG1, DSHAD, DLEG2, RLEG2, DCASS, ALPHA)
C
C   Blocking by quadrupode legs and Cassegrain mirror
C   Plane wave and spherical wave shadows for complex aperture plane
C   With 4 legs each with 2 sections (for 30m telescope case)
C   Spherical wave shadows taken as triangular
C   psi    Complex array of aperture amplitudes
C   mdim   Array size
C   sampi  Sampling interval as fraction of far field Nyquist rate
C   diam   Telescope diameter (m)
C   dleg1  Smallest leg diameter for whole length
C   dleg2  Larger leg diameter valid from rleg2 outwards in aperture
C   rleg1  Starting radius of thinnest spherical wave shadow (zero)
C   rleg2  Starting radius of larger section plane wave shadow
C   dshad  Diameter of spherical wave shadow at periphery
C   alpha  Orientation to az-el axis (0 or 45 degrees)
C
      COMPLEX PSI (MDIM, MDIM)
      LOGICAL L1, L2, L3, L4
C
      PY=3.1415926535
      PY4=PY/4.
      MENOSPY4=-1.0*PY4
C
      center of aperture
      MCENT=MDIM/2
C
      Pixel separation
      DX=DIAM/SAMPI/ (MDIM)
C
      Semi angle of shadow
      TANTHETA=(DSHAD-DLEG1)/(DIAM-2.*RLEG1)
C
      Actual x coordinate of apex of triangular shadow
      IF (TANTHETA.NE.0.) DELX=0.5*DLEG1/TANTHETA
C
      X0=RLEG1-DELX
C
      Loop over 4 quadrants
      DO 100 I=1,4
      ALPHA1=ALPHA+(I-1)*PY/2.
C
      Main loop for 1 quadrant
      DO 100 M=1,MDIM
      DO 100 N=1,MDIM
C
      convert to meters
      X1=(M-MCENT)*DX
      Y1=(N-MCENT)*DX
C
      Rotate coordinate system
      X=X1*COS (ALPHA1)-Y1*SIN (ALPHA1)
      Y=X1*SIN (ALPHA1)+Y1*COS (ALPHA1)
      R2=X**2+Y**2
      R=SQRT (R2)
C
C   Cassegrain blockage
      IF (R.LE. (DCASS/2.)) THEN
          PSI (M, N)=CMPLX (0., 0.)
      ENDIF
C
C   now legs
      IF (X.NE.0.) THEN
          TANTHETA1=Y/X
      ENDIF
C
```

```

        IF((X.EQ.0.).AND.(Y.EQ.0.)) THEN
            THETA1=0.
        ELSE
            THETA1=ATAN2(Y,X)
        ENDIF
C
C   Select outer part first quadrant,inside mirror diameter
        IF(THETA1.LT.PY4) THEN
            L1=.TRUE.
        ELSE
            L1=.FALSE.
        ENDIF
C
        IF(THETA1.GT.MENOSPY4) THEN
            L2=.TRUE.
        ELSE
            L2=.FALSE.
        ENDIF
C
        IF(X.GE.X0) THEN
            L3=.TRUE.
        ELSE
            L3=.FALSE.
        ENDIF
C
        IF(R.LT.(DIAM/2.)) THEN
            L4=.TRUE.
        ELSE
            L4=.FALSE.
        ENDIF
C
        IF((L1.AND.L2).AND.(L3.AND.L4)) THEN
            TANTHET2=Y/(X-X0)
C   Spherical wave shadow
            IF(ABS(TANTHET2) .LE. ABS(TANTHETA)) THEN
                PSI(M,N)=CMPLX(0.,0.)
                NPTS=NPTS+1
            ENDIF
        ENDIF
C   Plane wave shadows
        IF(THETA1.LT.PY4.AND.THETA1.GT.MENOSPY4.AND.R.LE.DIAM/2.) THEN
            IF(ABS(Y) .LE. DLEG1/2.) PSI(M,N)=CMPLX(0.,0.) !Thinner
            IF(X.GT.RLEG2.AND.ABS(Y) .LE. DLEG2/2.) PSI(M,N)=CMPLX(0.,0.)
        ENDIF
100 CONTINUE
C
C
        RETURN
        END

```

## Anexo XXX: Subrutina lessfit6.f

```
      SUBROUTINE LESSFIT6 (PSI,MDIM,MCENT,PHCOEF)
C     TO SUBTRACT PHASE FUNCTION DERIVED BY PHASFIT FROM
C     COMPLEX APERTURE DISTRIBUTION PSI (M,N)
C
      DIMENSION PSI (MDIM,MDIM)
      DIMENSION PHCOEF (6)
      COMPLEX PSI
C
      A=PHCOEF (1)
      B=PHCOEF (2)
      C=PHCOEF (3)
      D=PHCOEF (4)
      E=PHCOEF (5)
      F=PHCOEF (6)
      DO M=1,MDIM
          DO N=1,MDIM
              X=M-MCENT
              Y=N-MCENT
              R2=X**2 + Y**2
C             COMA TERMS
              S=X**3+X*Y**2
              T=Y**3+Y*X**2
              PHI=- ( A*X + B*Y + C + D*R2 + E*S + F*T )
              PSI (M,N)=PSI (M,N) *CMPLX (COS (PHI) , SIN (PHI) )
          END DO
      END DO
      RETURN
      END
```

## Anexo XXXI: Subrutina makegdf.f

```
      SUBROUTINE MAKEGDF (MAPAA,MDIM,NDIM, IDEN)
C     GENERA FICHEROS .GDF UTILES PARA TRABAJAR
C     CON GRAPHIC.
C     AMBOS MAPAS DEBEN SER DE 64x64
C
C     IDEN ES EL IDENTIFICADOR DE TAREA A REALIZAR POR
C     LA SUBROUTINA, Y SIGNIFICA:
C         1: PARA AVMAP_AMPLITUD.GDF
C         2: PARA AVMAP_FASE.GDF
C         3: PARA PICOMAP_AMPLITUD.GDF
C         4: PARA PICOMAP_FASE.GDF
C         5: PARA RESIDUALS.GDF
C         6: PARA MAPERROR.GDF
C
      REAL MAPAA (MDIM,NDIM)
      REAL*8 CONVA (6)
      INTEGER IDEN
C
      DIBUJO: SELECT CASE (IDEN)
C*****
C     CASE 1
C*****
      CASE (1) DIBUJO
C
      CONVA (1)=1.0          ! REFERENCE PIXEL IN X
      CONVA (2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
      CONVA (3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
      CONVA (4)=1.0          ! REFERENCE PIXEL IN Y
      CONVA (5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
```

```

CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR_EXEC('CLEAR')
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN 5 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN 0.05*RGMAX RGMAX')
CALL GR_EXEC('TICK 1 2 1 2')
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
CALL GR_EXEC('BOX')
CALL GR_EXEC('WRITE IMAGE AVMAP_AMP.GDF')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C          CASE 2
C*****
CASE (2) DIBUJO
C
CONVA(1)=1.0          ! REFERENCE PIXEL IN X
CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR_EXEC('CLEAR')
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN 5 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN 0.05*RGMAX RGMAX')
CALL GR_EXEC('TICK 1 2 1 2')
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
CALL GR_EXEC('BOX')
CALL GR_EXEC('WRITE IMAGE AVMAP_PHA.GDF')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C          CASE 3
C*****
CASE (3) DIBUJO
CONVA(1)=1.0          ! REFERENCE PIXEL IN X
CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0          ! REFERENCE PIXEL IN Y
CONVA(5)=0.0          ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0          ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR_EXEC('CLEAR')
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN 5 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN 0.05*RGMAX RGMAX')
CALL GR_EXEC('TICK 1 2 1 2')
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
CALL GR_EXEC('BOX')
CALL GR_EXEC('WRITE IMAGE PICOMAP_AMP.GDF')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C          CASE 4
C*****
CASE (4) DIBUJO
CONVA(1)=1.0          ! REFERENCE PIXEL IN X
CONVA(2)=0.0          ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0          ! X USER COORDINATE INCREMENT PER PIXEL

```

```

CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR_EXEC('CLEAR')
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN 5 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN 0.05*RGMAX RGMAX')
CALL GR_EXEC('TICK 1 2 1 2')
CALL GR_EXEC('LIMITS -8.55225 +8.55225 -8.55225 +8.55225')
CALL GR_EXEC('BOX')
CALL GR_EXEC('WRITE IMAGE PICOMAP_PHA.GDF')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C          CASE 5
C*****
CASE (5) DIBUJO
CONVA(1)=1.0      ! REFERENCE PIXEL IN X
CONVA(2)=0.0      ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR_EXEC('CLEAR')
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN -100 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN -1.0*RGMAX RGMAX')
CALL GR_EXEC('BOX')
CALL GR_EXEC('WRITE IMAGE RESIDUALS.GDF')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C          CASE 6
C*****
CASE (6) DIBUJO
CONVA(1)=1.0      ! REFERENCE PIXEL IN X
CONVA(2)=0.0      ! X USER COORDINATE VALUE AT CONV(1)
CONVA(3)=1.0      ! X USER COORDINATE INCREMENT PER PIXEL
CONVA(4)=1.0      ! REFERENCE PIXEL IN Y
CONVA(5)=0.0      ! Y USER COORDINATE VALUE AT CONV(4)
CONVA(6)=1.0      ! Y USER COORDINATE INCREMENT PER PIXEL
C
CALL GR_EXEC('CLEAR')
CALL GR4_RGIVE(MDIM,MDIM,CONVA,MAPAA)
CALL GR_EXEC('LIMITS -0.5 62.5 -0.5 62.5')
CALL GR_EXEC('PLOT /SCAL LIN -100 100')
CALL GR_EXEC('WEDGE')
CALL GR_EXEC('PLOT /SCAL LIN -1.0*RGMAX RGMAX')
CALL GR_EXEC('BOX')
CALL GR_EXEC('WRITE IMAGE MAPERROR.GDF')
CALL GR_EXEC('CLEAR')
RETURN
C*****
C          CASE DEFAULT
C*****
CASE DEFAULT DIBUJO
WRITE(6,*) 'IDENTIFICADOR DE DIBUJO INCORRECTO'
WRITE(6,*) 'ERROR EN SUBROUTINA MAKEGDF.F'
RETURN
C*****
C          FIN DEL CASE
C*****

```

END SELECT DIBUJO  
END



## Anexo XXXII: Subrutina mapin3.f

```
      SUBROUTINE MAPIN3 (DATA,MDIM,NSCAN,NX,RX,VX,DELX,NY,RY,VY,DELY,
1 MOFFSET,NOFFSET,IRecent)
C
C   TO READ IN A MAP IN LUCAS' "GREG" FORMAT
C   WITH (irecent=1) OR WITHOUT RECENTERING
C
      DIMENSION DATA (MDIM,MDIM)
      CHARACTER NSCAN*9,INFILE*13
      INFILE=NSCAN//'.map'
C
      OPEN (UNIT=1,STATUS='OLD',FILE=INFILE,FORM=
1 'FORMATTED',IOSTAT=IOS1)
      IF (IOS1.NE.0) WRITE (6,*) 'INPUT FILE STATUS (MAPIN)',IOS1
C
      CALL GETMAP3 (DATA,MDIM,1,NX,RX,VX,DELX,NY,RY,VY,DELY,MOFFSET,
1 NOFFSET,IRecent)
C
      CLOSE (UNIT=1)
C
      RETURN
      END
```

## Anexo XXXIII: Subrutina mkfactor1.f

```
      SUBROUTINE MKFACTOR1 (PSI,FACTOR,MDIM,NDIM,
1 FD,RMAG,SAMPI,DELTA,II,IMASK,NCASSI,IMASKC)
C
C   TO FILL DEFOUSSING ARRAY FACTOR
C   TO ADD II=1 OR SUBTRACT II=-1 PHASE ERROR OF FORM (1-COS(THETA))
C   OR DELTA*R**2/(1.+R**2)
C
C   OPTIONAL MASKING OUTSIDE APERTURE IF (IMASK.NE.0) THEN
C   IF OUTSIDE ANTENNA RADIUS SET PSI=(0.0,0.0)
C   OPTIONAL MASKING IN CENTRAL BLOCKAGE
C
C   D.MORRIS IRAM SEPT 1994
C
C   DELTA    -AXIAL DEFOCUS IN WAVELENGTHS
C   FD       -FOCAL LENGTH/DIAM. OF PARABOLOID
C   RMAG     -MAGNIFICATION OF CASS.,SET INFINITE FOR PRIME FOCUS
C   SAMPI    -SAMPLING INTERVAL IN FAR FIELD AS FRACTION OF CRITICAL
C             VALUE, (WAVELENGTH/DIAM.)
C   NCASSI   -RADIUS OF CENTRAL BLOCKAGE
C   IMASKC   - 1 FOR MASKING OF CENTRAL BLOCKAGE
C
C
C   DIMENSION PSI (MDIM,NDIM),FACTOR (MDIM,NDIM)
C
C   COMPLEX PSI,FACTOR
C   PY=3.1415926535
C   RMAG2=RMAG**2
C
C   MAKE AND LOAD FACTOR ARRAY
C   NDIM2=NDIM/2
C   MDIM2=MDIM/2
C   RMAX=MDIM*SAMPI/2.
C   MMAX=RMAX
C   LOAD FACTOR
C   DO 100 M=1,MDIM
C   DO 100 N=1,NDIM
```

```

NR2=(N-NDIM2)**2+(M-MDIM2)**2
RN2=FLOAT(NR2)/RMAX/RMAX
NR=SQRT(FLOAT(NR2))
R2=RN2/16./(FD**2)
PHIDF=4.*PY*DELTA*(R2/(1.+R2)+(R2/RMAG2)/(1.+R2/RMAG2))
FACTOR(M,N)=CMPLX(COS(PHIDF),SIN(PHIDF))
IF(IMASK.NE.0) THEN
    IF(NR.GT.MMAX) FACTOR(M,N)=(0.0,0.0)
END IF
IF(IMASKC.EQ.1) THEN
    IF(NR.LE.NCASSI) FACTOR(M,N)=(0.0,0.0)
END IF
100 CONTINUE
C
RETURN
END

```

## Anexo XXXIV: Subrutina omisel5.f

```
      SUBROUTINE OMISEL5 (PSI, FACTOR, M, N, BLURR, FFELD, NPRINT, ITMAX
1 , SSQERR, ACC, XX, IBIT, CNVERG, FD, RMAG, SAMPI, DELTA, LPRINT, LTEK, IMASK
2 , NCASSI, IMASKC)
C
C      ITERATIVE CAL.OF COMPLEX FAR FIELD WAVEFUNCTION COMPATABLE
C      WITH INFOCUS AND DEFOCUSSED FAR FIELD AMPLITUDE PATTERNS
C      USES EXACT PHASE ERROR FUNCTION OF RADIUS-RUZE 1979
c      OF FORM DELTA*(1.-COS(THETA)) I.E. R**2/(1.+R**2)
C
C      D.MORRIS IRAM SEPT 1994
C
C      OPTION TO MASK OUTSIDE APERTURE FOR IMASK ITERATIONS
C
C      OPTION TO MASK CENTRAL BLOCKAGE
C
C      BASED ON GSALG
C      SEE "COMPUTER TECHNIQUES FOR IMAGE PROCESSING IN ELECTRON
C      MICROSCOPY" BY W.O.SAXTON,ACADEMIC PRESS 1978.
C
C      VERSION TO USE FFT1 (2-D FFT ROUTINE OF ARAMBEPOLA)
C
C      VERSION USING DEFOCUS4 FOR OPTIONAL MASKING OF APERTURE
C
C      PSI(M,N)   -COMPLEX WORKING ARRAY
C                  (M AND N ARE POWERS OF 2)
C      BLURR      -DEFOCUSSED FAR FIELD AMPLITUDE DISTRIBUTION
C      FFELD      -IN FOCUS FAR FIELD AMPLITUDE DISTRIBUTION
C      NPRINT     -NUMBER OF CYCLES BETWEEN ERROR PRINTOUT
C      ITMAX      -MAXIMUM NUMBER OF ITERATIONS TO BE PERFORMED
C      SSQERR     -SUM SQUARE ERROR IN FAR FIELD PLANE ESTIMATE
C      ACC        -REQUIRED ACCURACY
C      CNVERG     -CONVERGENCE RATE LIMIT,PERCENT PER NPRINT
C      XX         -COMPLEX ARRAY OF M ELEMENTS
C      IBIT       -INTEGER ARRAY OF M ELEMENTS
C      DELTA      -AXIAL DEFOCUS IN WAVELENGTHS
C      FD         -FOCAL LENGTH/DIAMETER FOR PARABOLOID
C      RMAG       -MAGNIFICATION OF CASS. SYSTEM,INFINITE FOR
C                  PRIME FOCUS CASE.
C      SAMPI      -SAMPLE INTERVAL IN FAR-FIELD,AS FRACTION OF
C                  CRITICAL VALUE (DIAM./WAVELENGTH)
C      IMASK      -IF NE.0 THEN APERTURE MASKED TO ZERO AT EACH ITERATION
C                  FOR IMASK ITERATIONS,IF 1 FOR ALL ITERATIONS
C      IMASKC     - 1 FOR MASKING OF CENTRAL BLOCKAGE DURINT ITERATIONS
C
C      THE ITERATION QUILTS IF:-
C      1) SSQERR < ACC
C      2) ITMAX CYCLES HAVE BEEN PERFORMED
C      3) THE CONVERGENCE GETS VERY SLOW
C
C      COMPLEX PSI
C      COMPLEX FACTOR, XX
C      DIMENSION PSI (M*N) , BLURR (M*N) , FFELD (M*N) , NN (2) , XX (M) , IBIT (M)
C      DIMENSION FACTOR (M*N)
C      LOGICAL L1
C      MN=M*N
C      NN(1)=M
C      NN(2)=N
C      M2=ALOG (FLOAT (M) ) /ALOG (2. )
C      N2=ALOG (FLOAT (N) ) /ALOG (2. )
C      IDIM=M*2
C      IORIG=M/2
C      JORIG=N/2
C
C      IF (IMASK.EQ.1) IMASK=ITMAX+1      !If IMASK.EQ.1 mask for all iterations
```

```

C
C CHECK NORMALIZATION OF MODULUS DATA AND
C FORCE MODULUS OF SUPPLIED PSI IN 'INFOCUS' PLANE
E1=0.
E2=0.
DO 1 I=1,MN
E1=E1+BLURR(I)*BLURR(I)
1 E2=E2+FFELD(I)*FFELD(I)
T0=SQRT(E1/E2)
DO 2 I=1,MN
S=CABS(PSI(I))
IF(S.EQ.0.) GO TO 2
PSI(I)=PSI(I)*FFELD(I)/S
2 BLURR(I)=BLURR(I)/T0
C
C INITIALIZE ITERATION
RMN=MN
RMN=1./SQRT(RMN)
TR=1E30
TC=CNVERG*NPRINT
NC=1
NIT=NPRINT
WRITE(LTEK,14)
WRITE(LPRINT,14)
C
C INITIALISE DEFOCUSING AND MASKING ARRAY FACTOR
CALL MKFACTOR1(PSI,FACTOR,M,N,FD,RMAG,SAMPI,DELTA,1,IMASK,NCASSI,
1 IMASKC)
C
C START ITERATIONS HERE
C COMPUTE ERROR AND CORRECT MODULUS IN 'DEFOCUSSED' PLANE
5 L1=.FALSE.
CALL FFT1(PSI,M,L1,XX,IBIT)
IF(NC.EQ.IMASK) THEN !STOP MASKING AT THIS ITERATION
CALL MKFACTOR1(PSI,FACTOR,M,N,FD,RMAG,SAMPI,DELTA,1,0,
1 NCASSI,0)
END IF
CALL DEFOCUS4(PSI,FACTOR,M,N,FD,RMAG,SAMPI,DELTA,1,IMASK)
L1=.TRUE.
CALL FFT1(PSI,M,L1,XX,IBIT)
C CALCULATE ERROR IN DEFOCUSSED PLANE
T1=0.
DO 3 I=1,MN
S=CABS(PSI(I))
P=S-BLURR(I)
T1=T1+P*P
C MAKE CORRECTION IN DEFOCUSSED PLANE
3 IF(S.NE.0.) PSI(I)=PSI(I)*BLURR(I)/S
T1=T1*T0**2*100./E1
C
C REPEAT CORRECTION IN 'INFOCUS' PLANE
L1=.FALSE.
CALL FFT1(PSI,M,L1,XX,IBIT)
CALL DEFOCUS4(PSI,FACTOR,M,N,FD,RMAG,SAMPI,DELTA,-1,IMASK)
L1=.TRUE.
CALL FFT1(PSI,M,L1,XX,IBIT)
C CALCULATE ERROR IN 'INFOCUS' PLANE
T2=0.
DO 33 I=1,MN
S=CABS(PSI(I))
P=S-FFELD(I)
T2=T2+P*P
C MAKE CORRECTION IN 'INFOCUS' PLANE
IF(S.NE.0) PSI(I)=PSI(I)*FFELD(I)/S
33 CONTINUE
T2=T2*100./E2
C
C CHECK EXIT CRITERIA AND LOOP

```

```

        IF (NIT.EQ.NPRINT) GO TO 6
        NIT=NIT+1
        GO TO 7
    6   TTT=TR-T2
        WRITE (LTEK,10) NC,T1,T2,TTT
        WRITE (LPRINT,10) NC,T1,T2,TTT
        NIT=1
    7   IF (T2.GT.ACC) GO TO 9
        WRITE (LTEK,11)
        WRITE (LPRINT,11)
        SSQERR=T2
        RETURN
    9   IF (NC.LT.ITMAX) GO TO 20
        WRITE (LTEK,12) ITMAX
        WRITE (LPRINT,12) ITMAX
        SSQERR=T2
        RETURN
    20  IF (TR-T2.GT.TR*TC) GO TO 8
        WRITE (LTEK,13)
        WRITE (LPRINT,13)
        SSQERR=T2
        RETURN
C
C   RECYCLE
    8   TR=T2
        NC=NC+1
        GO TO 5
    10  FORMAT (I8,5X,F15.5,'% ',3X,F15.5,'% ',7X,1PE10.3)
    11  FORMAT (' REQUESTED ACCURACY ACHIEVED'//)
    12  FORMAT (/I6,' ITERATIONS NOW PERFORMED'//)
    13  FORMAT (' CONVERGENCE SLOW - ITERATION TERMINATED'//)
    14  FORMAT ('NUMBER OF',9X,'SSQ ERROR IN',7X,'SSQ ERROR IN',
    1   6X,'CONVERGENCE',/, 'ITERATIONS',7X,' DEFOCUSSED PLANE',3X,
    2   'FOCUSSED PLANE',8X,'RATE'//)
        END

```

## Anexo XXXV: Subrutina orgshft.f

```

        SUBROUTINE ORGSHFT (A,MMAX,NMAX)
C   TO MOVE ORIGIN FROM CORNERS TO CENTRE
C   A (M,N) HAS (2*MMAX)*NMAX REAL COMPONENTS I.E. MMAX*NMAX COMPLEX
        DIMENSION A (MMAX*2,NMAX)
C
C   FIRST SHIFT IN X DIRECTION (M)
    DO 1 N=1,NMAX
    DO 1 M=1,MMAX
        STORE=A (M,N)
        M1=M+MMAX
        A (M,N)=A (M1,N)
    1   A (M1,N)=STORE
C
C   SECOND SHIFT IN Y DIRECTION (N)
        MMAX2=MMAX*2
        NMAX2=NMAX/2
    DO 2 M=1,MMAX2
    DO 2 N=1,NMAX2
        STORE=A (M,N)
        N1=N+NMAX2
        A (M,N)=A (M,N1)
    2   A (M,N1)=STORE
        RETURN
        END

```

## Anexo XXXVI: Subrutina phasfit6.f

```

SUBROUTINE PHASFIT6 (PSI, MDIM, MCENT, MMAX, IWT, PHCOEF)
C
C   LEAST SQUARE FIT OF PISTON, SLOPE, AND QUADRATIC PHASE
C   AND 3'RD ORDER COMA TERMS-PRIMARY COS(THETA)*R**3
C   FUNCTION USING CONSTSNT WEIGHT ( IWT=0 ) OR AMPLITUDE
C   WEIGHTING ( IWT=1 ).
C   PSI IS 2D COMPLEX APERTURE DISTRIBUTION.
C
C   FOR LARGE ARRAYS (MDIM>32) WEIGHTS DIVIDED BY 10000. TO AVOID OVERFLOWS
C
C
C   PHI= aX +bY + c + d(X**2 + Y**2)
C         +E(X**3+X*Y**2) +F(Y**3+Y*X**2)
C
C   DIMENSION PSI (MDIM, MDIM)
C   DIMENSION PHCOEF (6)
C   DIMENSION SUM (7, 6), rMAT (6, 6), aMAT (6, 6), BMAT (6, 6)
C   DIMENSION CMAT (6, 6), DMAT (6, 6)
C   DIMENSION EMAT (6, 6), FMAT (6, 6)
C   COMPLEX PSI
C
C   FIND MAX AMPLITUDE FOR NORMALIZATION OF WEIGHTS
C   AMAX=0.
C   DO M=1, MDIM
C       DO N=1, MDIM
C           AMP=CABS (PSI (M, N))
C           IF (AMP.GT.AMAX) AMAX=AMP
C       END DO
C   END DO
C
C   DIVID WEIGHTS IF LARGE ARRAYS
C   IF (MDIM.GT.32) AMAX=AMAX*10000.
C
C   CLEAR SUM ARRAY
C   DO I=1, 7
C       DO J=1, 6
C           SUM (I, J)=0.
C       END DO
C   END DO
C
C   FILL SUM ARRAY
C   DO 2 M=1, MDIM
C       DO 2 N=1, MDIM
C           PHASE=CPHAS (PSI (M, N))
C           X=M-MCENT
C           Y=N-MCENT
C           R2=X**2 + Y**2
C           COMA TERMS
C           S=X**3+X*Y**2
C           T=Y**3+Y*X**2
C           WT=CABS (PSI (M, N)) /AMAX
C           IF (IWT.EQ.0) WT=1.0
C           RMAX2=MMAX**2
C           IF (R2.GT.RMAX2) GOTO 2
C
C           SUM (1, 1)=SUM (1, 1) + WT*X**2
C           SUM (2, 1)=SUM (2, 1) + WT*X*Y
C           SUM (3, 1)=SUM (3, 1) + WT*X
C           SUM (4, 1)=SUM (4, 1) + WT*X*R2
C           SUM (5, 1)=SUM (5, 1)+WT*S*X
C           SUM (6, 1)=SUM (6, 1)+WT*T*X
C           SUM (7, 1)=SUM (7, 1) + WT*X*PHASE
C           SUM (2, 2)=SUM (2, 2) + WT*Y**2
C           SUM (3, 2)=SUM (3, 2) + WT*Y

```

```

SUM(4,2)=SUM(4,2) + WT*Y*R2
SUM(5,2)=SUM(5,2)+WT*S*Y
SUM(6,2)=SUM(6,2)+WT*T*Y
SUM(7,2)=SUM(7,2) + WT*Y*PHASE
SUM(3,3)=SUM(3,3) + WT
SUM(4,3)=SUM(4,3) + WT*R2
SUM(5,3)=SUM(5,3)+WT*S
SUM(6,3)=SUM(6,3)+WT*T
SUM(7,3)=SUM(7,3) + WT*PHASE
SUM(4,4)=SUM(4,4) + WT*R2**2
SUM(5,4)=SUM(5,4)+WT*R2*S
SUM(6,4)=SUM(6,4)+WT*R2*T
SUM(7,4)=SUM(7,4) + WT*R2*PHASE
SUM(5,5)=SUM(5,5)+WT*S*S
SUM(6,5)=SUM(6,5)+WT*S*T
SUM(7,5)=SUM(7,5)+WT*S*PHASE
SUM(6,6)=SUM(6,6)+WT*T*T
SUM(7,6)=SUM(7,6)+WT*T*PHASE
2 CONTINUE
C
SUM(1,3)=SUM(3,1)
SUM(2,3)=SUM(3,2)
SUM(1,2)=SUM(2,1)
SUM(3,4)=SUM(4,3)
SUM(1,4)=SUM(4,1)
SUM(2,4)=SUM(4,2)
SUM(1,5)=SUM(5,1)
SUM(2,5)=SUM(5,2)
SUM(3,5)=SUM(5,3)
SUM(4,5)=SUM(5,4)
SUM(1,6)=SUM(6,1)
SUM(2,6)=SUM(6,2)
SUM(3,6)=SUM(6,3)
SUM(4,6)=SUM(6,4)
SUM(5,6)=SUM(6,5)
C
C FORM MATRICES
NEQ=6
CALL MATFIL(SUM,NEQ,aMAT,1)
CALL MATFIL(SUM,NEQ,bMAT,2)
CALL MATFIL(SUM,NEQ,cMAT,3)
CALL MATFIL(SUM,NEQ,dMAT,4)
CALL MATFIL(SUM,NEQ,eMAT,5)
CALL MATFIL(SUM,NEQ,fMAT,6)
CALL MATFIL(SUM,NEQ,rMAT,0)
C
C SOLUTION
DET=DETERM(RMAT,6)
A=DETERM(AMAT,6)/DET
B=DETERM(BMAT,6)/DET
C=DETERM(CMAT,6)/DET
D=DETERM(DMAT,6)/DET
E=DETERM(EMAT,6)/DET
F=DETERM(FMAT,6)/DET
PHCOEF(1)=A
PHCOEF(2)=B
PHCOEF(3)=C
PHCOEF(4)=D
PHCOEF(5)=E
PHCOEF(6)=F
RETURN
END
C*****
FUNCTION DETERM(ARRAY,NORDER)
C CALCULATE DETERMINANT OF MATRIX ARRAY MAX DIMENSIONS NORDER
C FROM BEVINGTON "DATA REDUCTION AND ERROR ANALYSIS FOR THE PHYSICAL
C SCIENCES" P 294.
DIMENSION ARRAY(NORDER,NORDER)

```

```

C      MAXIMUM ORDER SET TO 10
10  DETERM=1.
11  DO 50 K=1,NORDER
C
C      INTERCHANGE COLS IF DIAGONAL ELEMENT IS ZERO
C
      IF (ARRAY (K,K) ) 41,21,41
21  DO 23 J=K,NORDER
      IF (ARRAY (K,J) ) 31,23,31
23  CONTINUE
      DETERM=0.
      GOTO 60
31  DO 34 I=K,NORDER
      SAVE=ARRAY (I,J)
      ARRAY (I,J)=ARRAY (I,K)
34  ARRAY (I,K)=SAVE
      DETERM=-DETERM
C
C      SUBTRACT ROW K FROM LOWER ROWS TO GET DIAGONAL MATRIX
C
41  DETERM=DETERM*ARRAY (K,K)
      IF (K-NORDER) 43,50,50
43  K1=K+1
      DO 46 I=K1,NORDER
      DO 46 J=K1,NORDER
46  ARRAY (I,J)=ARRAY (I,J)-ARRAY (I,K) * (ARRAY (K,J) /ARRAY (K,K) )
C      NOTE EXTRA BRACKETS TO REDUCE CHANCE OF FLOATING PT. OVERFLOW
50  CONTINUE
60  RETURN
      END
C*****
      SUBROUTINE MATFIL (SUM,NEQ,RMAT,ICOL)
C      MAKE SQUARE MATRIX BY COL. REPLACEMENT FROM RECTANGULAR ARRAY
C      SUM.FOR SOLN OF SIMULTANEOUS EQUATIONS WITH DETERMINANTS
C
      DIMENSION SUM (NEQ+1,NEQ) ,RMAT (NEQ,NEQ)
              DO I=1,NEQ
              II=I
              IF (I.EQ.ICOL) II=NEQ+1
              DO J=1,NEQ
              RMAT (I,J)=SUM (II,J)
              END DO
      END DO
      RETURN
      END
C*****

```

## Anexo XXXVII: Subrutina phasmxmn.f

```

      SUBROUTINE PHASMXMN (PSI,MDIM,PHMAX,PHMIN)
C      FINDS PHASE LIMITS OF COMPLEX APERTURE DISTRIBUTION PSI (M,N)
      DIMENSION PSI (MDIM,MDIM)
      COMPLEX PSI
C
      PHMAX=-1.0E+37
      PHMIN=+1.0E+37
      DO 10 M=1,MDIM
      DO 10 N=1,MDIM
      IF (CABS (PSI (M,N) ) .EQ.0.) GOTO 10
      PHI=CPHAS (PSI (M,N) )
      IF (PHI.LT.PHMIN) PHMIN=PHI
      IF (PHI.GT.PHMAX) PHMAX=PHI
10  CONTINUE
C

```



RETURN  
END

## Anexo XXXVIII: Subrutina point5.f

```

SUBROUTINE POINT5(A, DELTAA, PSI, MDIM, NDIM, BLURR, FFELD, XX,
1 IBIT, FD, RMAG, SAMPI, DELTA1, DELTA2, MOFSETI, NOFSETI, MMAX, NMAX,
1 NCASSI, NLEGI, TAPERI, ASTIGX, ASTIGY, DEFOCI, RANPHII, ISEED, LEG45,
1 ZTX, DELXF, RLAMDA, LTEK, LPRINT, IMASK, DIAM, IPOINT, IMASKC)
C
C   TO FIND MINIMUM OF POINTERR  OF FOUR VARIABLES X=A(1),Y=A(2),Z=A(3)...
C   STEP IN X,Y IS DELTAA(J), J=1,2,3,4 ,THIS IS CHANGED IN PROGRAM
C   FOR A SUBSEQUENT CALL FOR NEXT APPROXIMATION
C   PROCESS SEARCHES BEGINNING AT VALUES A(J) IN CALL
C
C   WITH OPTION FOR OPTIMIZATION OF TAPER AND FOCUS (SEPT 1992)
C
C   BASED ON GRIDLS OF BENNINGTON
C
C   DIMENSION A(5), DELTAA(5)
C   DIMENSION PSI(MDIM,MDIM), BLURR(MDIM,MDIM)
C   DIMENSION FFELD(MDIM,MDIM), XX(MDIM), IBIT(MDIM)
C
C   COMPLEX PSI,XX
C
C   WRITE(LTEK,*)
C   WRITE(LPRINT,*)
C
200  FORMAT(3X,'SSQERR',9X,'XTILT',8X,'YTILT',8X,'DEFOCI',
1 7X,'ASTIGX',7X,'ASTIGY',/,18X,'pixels',7X,'pixels',
2 7X,'lambda',7X,'(rad.)',7X,'(rad.)')
C
C   WRITE(LTEK,200)
C   WRITE(LPRINT,200)
C
C   WRITE(LTEK,*)
C   WRITE(LPRINT,*)
C
C   IF(IPOINT.EQ.1) IDIMENS=2 !OPTIMISE POINTING
C   IF(IPOINT.EQ.2) IDIMENS=3 !OPTIMIZE POINTING, FOCUS
C   IF(IPOINT.GE.3) IDIMENS=5 !OPTIMISE POINTING, FOCUS AND ASTIGMATISM
20  DO 90 J=1, IDIMENS          !LOOP OVER THE MAX 5 DIMENSIONS
C
C   CALL POINTERR5(A, DELTAA, PSI, MDIM, NDIM, BLURR, FFELD, T1, XX, IBIT,
1 FD, RMAG, SAMPI, DELTA1, DELTA2, MOFSETI, NOFSETI, MMAX, NMAX, NCASSI,
1 NLEGI, TAPERI, ASTIGX, ASTIGY, DEFOCI, RANPHII, ISEED, LEG45, ZTX, DELXF,
1 RLAMDA, IMASK, DIAM, IMASKC)
C   WRITE(LTEK,91) T1,A(1),A(2),A(3),A(4),A(5)
C   WRITE(LPRINT,91) T1,A(1),A(2),A(3),A(4),A(5)
91  FORMAT(1X,6(F10.6,3X))
C   Z1=T1          !STARTING POINT AT A VALUES IN CALL
C   DELTA=DELTAA(J)
C   IF(DELTA.EQ.0.) GO TO 90 !DON'T MINIMIZE FOR THIS VARIABLE
C   FN=0.
41  A(J)=A(J)+DELTA          !INCREMENT X OR Y
C   CALL POINTERR5(A, DELTAA, PSI, MDIM, NDIM, BLURR, FFELD, T1, XX, IBIT,
1 FD, RMAG, SAMPI, DELTA1, DELTA2, MOFSETI, NOFSETI, MMAX, NMAX, NCASSI,
1 NLEGI, TAPERI, ASTIGX, ASTIGY, DEFOCI, RANPHII, ISEED, LEG45, ZTX, DELXF,
1 RLAMDA, IMASK, DIAM, IMASKC)
C   WRITE(LTEK,91) T1,A(1),A(2),A(3),A(4),A(5)
C   WRITE(LPRINT,91) T1,A(1),A(2),A(3),A(4),A(5)
C   Z2=T1          !SECOND TRY
45  IF(Z1-Z2) 51,41,61 !REVERSE SIGN IF Z INCREASING
C
51  DELTA=-DELTA
C   A(J)=A(J) + DELTA
C
C   SAVE=Z1          !NOW SWOP Z1 , Z2
C   Z1=Z2

```

```

57  Z2=SAVE
C
61  FN=FN+1.           !USED IF SECOND CALL FOR HIGHER ACCURACY
C
    A(J)=A(J) + DELTA
    CALL POINTERR5(A, DELTAA, PSI, MDIM, NDIM, BLURR, FFELD, T1, XX, IBIT,
1  FD, RMAG, SAMPI, DELTA1, DELTA2, MOFSETI, NOFSETI, MMAX, NMAX, NCASSI,
1  NLEGI, TAPERI, ASTIGX, ASTIGY, DEFOCI, RANPHII, ISEED, LEG45, ZTX, DELXF,
1  RLAMDA, IMASK, DIAM, IMASKC)
    WRITE(LTEK, 91)    T1, A(1), A(2), A(3), A(4), A(5)
    WRITE(LPRINT, 91) T1, A(1), A(2), A(3), A(4), A(5)
    Z3=T1             !THIRD ATTEMPT
C
66  IF(Z3-Z2) 71, 81, 81
71  Z1=Z2
    Z2=Z3
C
    GO TO 61
C
C  FIND MINIMUM DEFINED BY PARABOLA THROUGH LAST 3 POINTS
81  IF((Z3-Z2).NE.0.) THEN
    DELTA=DELTA*(1./(1.+(Z1-Z2)/(Z3-Z2))+0.5)
    ELSE
    DELTA=1.5*DELTA
    END IF
    A(J)=A(J) - DELTA      !OUTPUT VALUE
C  DELTAA(J)=DELTA(J)*FN/3. !SMALLER STEP FOR NEXT CALL
    DELTAA(J)=DELTA(J)/2.  !SMALLER STEP FOR NEXT CALL
C
90  CONTINUE
C
C  CLEAR PSI ARRAY AT END
DO 100 M=1, MDIM
DO 100 N=1, MDIM
100 PSI(M, N)=CMPLX(0., 0.)

    RETURN
    END

```

## Anexo XXXIX: Subrutina pointerr5.f

```
      SUBROUTINE POINTERR5(A, DELTAA, PSI, M, N, BLURR, FFELD, T1, XX, IBIT,
1  FD, RMAG, SAMPI, DELTA1, DELTA2, MOFSETI, NOFSETI, MMAX, NMAX, NCASSI,
1  NLEGI, TAPERI, ASTIGX, ASTIGY, DEFOCI, RANPHII, ISEED, LEG45, ZTX, DELXF,
1  RLAMDA, IMASK, DIAM, IMASKC)
C
C      CALCULATE SUM SQUARE ERROR IN DEFOCUSSED MAP FOR USE
C      FOCUS AND POINTING ERROR CALCULATION
C
C      PSI(M,N)  -COMPLEX WORKING ARRAY
C                (M AND N ARE POWERS OF 2)
C      BLURR    -DEFOCUSSED FAR FIELD AMPLITUDE DISTRIBUTION
C      FFELD    -IN FOCUS FAR FIELD AMPLITUDE DISTRIBUTION
C      T1       -SUM SQUARE ERROR IN FAR FIELD PLANE ESTIMATE
C      XX       -COMPLEX ARRAY OF M ELEMENTS
C      IBIT     -INTEGER ARRAY OF M ELEMENTS
C      DELTA    -AXIAL DEFOCUS IN WAVELENGTHS
C      FD       -FOCAL LENGTH/DIAMETER FOR PARABOLOID
C      RMAG     -MAGNIFICATION OF CASS. SYSTEM, INFINITE FOR
C                PRIME FOCUS CASE.
C      SAMPI    -SAMPLE INTERVAL IN FAR-FIELD, AS FRACTION OF
C                CRITICAL VALUE (DIAM./WAVELENGTH)
C      IMASK    -FOR MASKING OUTSIDE APERTURE LIMITS
C      IMASKC   -1 FOR MASKING WITHIN CENTRAL BLOCKAGE
C
C      DIMENSION PSI(M*N), BLURR(M*N), FFELD(M*N), XX(M), IBIT(M)
C      DIMENSION A(5), DELTAA(5)
C      COMPLEX PSI
C      COMPLEX XX
C      LOGICAL L1
C      MN=M*N
C      IDIM=M*2
C      PY=3.1415926535
C
C      LEG BLOCKING FOR 30M TELESCOPE (values after increasing diam 1987)
C      ALPHA=PY/4.      !ORIENTATION
C      DLEG1=0.294      !LEG DIAMETER (MAIN UPPER PART)
C      RLEG1=10.        !RADIUS OF INTERSECTION WITH MIRROR
C      DSHAD=1.218      !SHADOW DIAMETER AT EDGE OF MIRROR
C      DLEG2=0.654      !DIAMETER OF LOWER PART OF LEG
C      RLEG2=7.5        !RADIUS AT BEGINNING OF THICKER PART
C
C
C      MAKE "TILTED" APERTURE PLANE DISTRIBUTION AND TRANSFORM TO FAR FIELD
C      CALL ANTMAK4(PSI, M, N, MOFSETI, NOFSETI, MMAX, NMAX, NCASSI,
1  NLEGI, TAPERI, A(4), A(5), A(3), RANPHII, ISEED, LEG45, RMAG, FD)
C      MORE REALISTIC LEG BLOCKING FOR 30M TELESCOPE
C      IF(LEG45.EQ.30) THEN
C          DCASS=2.0
C          CALL LEGBLOCK(PSI, M, SAMPI, DIAM, DLEG1, RLEG1, DSHAD, DLEG2,
1  RLEG2, DCASS, ALPHA)
C      END IF
C
C      DEFOCUS TO "INFOCUS" POSITION
C      II=+1
C      CALL DEFOCUS01(PSI, M, N, FD, RMAG, SAMPI, DELTA1,
1  II, IMASK, NCASSI, IMASKC)
C      CALL TILT(PSI, M, A(1), A(2), 1)
C      IF(ZTX.NE.0.) CALL FRESNEL2(PSI, M, RLAMDA, -ZTX, DIAM, SAMPI)
C      L1=.TRUE.
C      CALL FFT1(PSI, M, L1, XX, IBIT)
C
C      CHECK NORMALIZATION OF MODULUS DATA AND
```

```

C      FORCE MODULUS OF SUPPLIED PSI IN 'INFOCUS' PLANE
      E1=0.
      E2=0.
      DO 1 I=1,MN
      E1=E1+BLURR(I)*BLURR(I)
1     E2=E2+FFELD(I)*FFELD(I)
      T0=SQRT(E1/E2)
      DO 2 I=1,MN
      S=CABS(PHI(I))
      IF(S.EQ.0.) GO TO 2
      PSI(I)=PSI(I)*FFELD(I)/S
2     BLURR(I)=BLURR(I)/T0
C
C      COMPUTE ERROR IN 'DEFOCUSSED' PLANE
5     L1=.FALSE.
      CALL FFT1(PHI,M,L1,XX,IBIT)
      DELTA21=DELTA2-DELTA1
      II=+1
      CALL DEFOCUS01(PHI,M,N,FD,RMAG,SAMPI,DELTA21,
1     II,IMASK,NCASSI,IMASKC)
      L1=.TRUE.
      CALL FFT1(PHI,M,L1,XX,IBIT)
C      CALCULATE ERROR IN DEFOCUSSED PLANE
      T1=0.
      DO 3 I=1,MN
      S=CABS(PHI(I))
      P=S-BLURR(I)
3     T1=T1+P*P
      T1=T1*T0**2*100./E1
C
      RETURN
      END

```

## Anexo XL: Subrutina prntout.f

```

      SUBROUTINE PRNTOUT(DATA,MDIM,NDIM,LPRINT)
C      TO PRINT OUT SIMPLE +-99 DATA
C      INPUT IN RADIANS
C      OUTPUT IN DEGREES ASSUMED LE 99
C      IF DIMENSIONS GT.32 THEN PRINT ALTERNATE VALUES
      DIMENSION DATA(MDIM,NDIM)
      PY=3.1415926535
C
      INC=1.+(MDIM-1)/32.
      WRITE(LPRINT,200) INC
200  FORMAT(2X,/,2X,'INCREMENT=',I4,/)
      DO 101 N=1,NDIM,INC
      WRITE(LPRINT,100) (IFIX(DATA(N,M)*180./PY),M=1,MDIM,INC)
100  FORMAT(2X,32I3.0,/)
101  CONTINUE
      RETURN
      END

```

## Anexo XLI: Subrutina psiread.f

```
      SUBROUTINE PSIREAD (PSI, MDIM, CFILE, IHEADER, MHEADER,  
1 IHEADER2, MHEADER2, NREAD)  
C  
C      READ IN COMPLEX FILE IN "PICOH1 'MAP'" OUTPUT FORMAT  
C  
      DIMENSION PSI (MDIM, MDIM)  
      DIMENSION IHEADER (MHEADER), IHEADER2 (MHEADER2)  
      COMPLEX PSI  
C  
      CHARACTER*13 FILEN  
      CHARACTER*9 CFILE  
C  
C      OPEN FILE  
      NREAD=0  
      FILEN=CFILE//'.map'  
      OPEN (UNIT=98, NAME=FILEN, STATUS='OLD', READONLY, FORM='UNFORMATTED',  
1 ERR=90, IOSTAT=IOS)  
C  
C      FIRST READ SIZES  
      READ (98, ERR=91, IOSTAT=IOS) NHEADER  
      READ (98, ERR=92, IOSTAT=IOS) NHEADER2  
      READ (98, ERR=93, IOSTAT=IOS) NRECORD  
      READ (98, ERR=94, IOSTAT=IOS) NREAD  
C  
      NEXT READ HEADERS  
      DO 11 I=1, NHEADER  
11  READ (98, ERR=95, IOSTAT=IOS) IHEADER (I)  
      DO 12 I=1, NHEADER2  
12  READ (98, ERR=96, IOSTAT=IOS) IHEADER2 (I)  
C  
      READ DATA  
      DO 25 M=1, NREAD  
      DO 25 N=1, NREAD  
25  READ (98, ERR=97, IOSTAT=IOS) PSI (M, N)  
      CLOSE (UNIT=98, STATUS='KEEP', ERR=80, IOSTAT=IOS)  
      RETURN  
C  
C      ERROR HANDLING AND CLOSING FILES  
90  WRITE (6, *) 'PSIREAD ERROR OPENING FILE ', IOS  
      GOTO 85  
91  WRITE (6, *) 'PSIREAD ERROR ON READ 1', IOS  
      GOTO 85  
92  WRITE (6, *) 'PSIREAD ERROR ON READ 2', IOS  
      GOTO 85  
93  WRITE (6, *) 'PSIREAD ERROR ON READ 3', IOS  
      GOTO 85  
94  WRITE (6, *) 'PSIREAD ERROR ON READ 4', IOS  
      GOTO 85  
95  WRITE (6, *) 'PSIREAD ERROR ON READ 5', IOS  
      GOTO 85  
96  WRITE (6, *) 'PSIREAD ERROR ON READ 6', IOS  
      GOTO 85  
97  WRITE (6, *) 'PSIREAD ERROR ON READ 7', IOS  
      GOTO 85  
80  WRITE (6, *) 'ERROR CLOSING FILE IN PSIREAD', IOS  
85  CLOSE (UNIT=98, STATUS='KEEP')  
      RETURN  
      END
```

## Anexo XLII: Subrutina rmask.f

```
      SUBROUTINE RMASK (A, MDIM, NDIM, MMAX, NMAX, NCASS, NLEG, AMASK, ICENT)
C     TO MASK APERTURE PLANE BEYOND EDGE AND WHERE BLOCKED
      DIMENSION A (MDIM, NDIM)
      M2=MDIM/2+ICENT
      N2=NDIM/2+ICENT
      DO 100 M=1, MDIM
      DO 100 N=1, NDIM
      NR2= (N-N2) **2
      MR2= (M-M2) **2
      RN2=FLOAT (NR2+MR2)
      NR=SQRT (RN2)
      IF (NR.GT.MMAX) A (M, N) =AMASK
      IF (NR.LE.NCASS) A (M, N) =AMASK
      IF (IABS (N-N2) .LE.NLEG ) A (M, N) =AMASK
      IF (IABS (M-M2) .LE.NLEG) A (M, N) =AMASK
100  CONTINUE
      RETURN
      END
```

## Anexo XLIII: Subrutina rmaxmin.f

```
      SUBROUTINE RMAXMIN (A, MDIM, NDIM, RMAX, RMIN)
C     FINDS MAX, MIN VALUES OF REAL ARRAY OF DIMENSION MDIM X NDIM
      DIMENSION A (MDIM, NDIM)
      RMAX=-1.E37
      RMIN=+1.E37
      DO 10 M=1, MDIM
      DO 10 N=1, NDIM
      IF (RMAX.LE.A (M, N) ) RMAX=A (M, N)
      IF (RMIN.GE.A (M, N) ) RMIN=A (M, N)
10  CONTINUE
      RETURN
      END
```

## Anexo XLIV: Subrutina smooth.f

```
      SUBROUTINE SMOOTH(AMP,PSI,MDIM,RTAPER,XX,IBIT)
C
C   TO SMOOTH REAL ARRAY WITH GAUSSIAN BY MULTIPLICATION IN TRANSFORM
C   SPACE
C
C   PSI IS COMPLEX WORKING ARRAY--SO IS XX
C
      DIMENSION AMP(MDIM,MDIM),PSI(MDIM,MDIM),XX(MDIM),IBIT(MDIM)
      COMPLEX PSI,XX
C
      LOGICAL L1
C
C   FILL PSI WITH AMP TO BE SMOOTHED
      DO 20 M=1,MDIM
      DO 20 N=1,MDIM
20    PSI(M,N)=CMPLX(AMP(M,N),0.)
C   TRANSFORM AND CHANGE ORIGIN TO CENTER
      L1=.TRUE.
      CALL FFT1(PSI,MDIM,L1,XX,IBIT)
      CALL ORGSHFT(PSI,MDIM,MDIM)
C   MULTIPLY BY GAUSSIAN
      MDIM21=MDIM/2 + 1
      SIG22=RTAPER**2/0.6931471
      DO 10 M=1,MDIM
      DO 10 N=1,MDIM
      R22=(M-MDIM21)**2 + (N-MDIM21)**2
      TAPE=EXP(-R22/SIG22)
10    PSI(M,N)=PSI(M,N)*TAPE
C   RESET ORIGIN TO CORNERS
      CALL ORGSHFT(PSI,MDIM,MDIM)
C   TRANSFORM BACK TO ORIGINAL SPACE
      L1=.FALSE.
      CALL FFT1(PSI,MDIM,L1,XX,IBIT)
C   PLACE ABSOLUTE VALUE BACK IN AMP ARRAY
      DO 30 M=1,MDIM
      DO 30 N=1,MDIM
30    AMP(M,N)=CABS(PSI(M,N))
C
      RETURN
      END
```



## Anexo XLV: Subrutina smoothc.f

```
      SUBROUTINE SMOOTHC (PSI,MDIM,NTAPER,XX,IBIT)
C
C   TO SMOOTH COMPLEX MAP PSI WITH GAUSSIAN
C   BY MULTIPLICATION FT BY GAUSSIAN
C
C   DIMENSION PSI (MDIM,MDIM) , XX (MDIM) , IBIT (MDIM)
C
C   COMPLEX PSI,XX
C
C   LOGICAL L1
C
C   TRANSFORM AND PUT NEW ORIGIN AT CENTER
L1=.TRUE.
CALL FFT1 (PSI,MDIM,L1,XX,IBIT)
CALL ORGSHFT (PSI,MDIM,MDIM)
C   MULTIPLY BY GAUSSIAN
MDIM21=MDIM/2+1           !ASSUMED CENTER
SIG22=NTAPER**2/0.6931471
DO 10 M=1,MDIM
DO 10 N=1,MDIM
R22=(M-MDIM21)**2 + (N-MDIM21)**2
TAPE=EXP (-R22/SIG22)
10 PSI (M,N)=PSI (M,N) *TAPE
C
C   RESET ORIGIN TO CORNERS
CALL ORGSHFT (PSI,MDIM,MDIM)
C
C   TRANSFORM BACK TO ORIGINAL MAP SPACE
L1=.FALSE.
CALL FFT1 (PSI,MDIM,L1,XX,IBIT)
C
C   RETURN
END
C*****
C llama a las subrutinas FFT1.F y ORGSHFT.F
C*****
```

## Anexo XLVI: Subrutina tilt.f

```
      SUBROUTINE TILT (PSI,MDIM,XTILT,YTILT,ISIGN)
C
C   INTRODUCE LINEAR PHASE GRADIENT ACROSS APERTURE PLANE
C
C   DIMENSION PSI (MDIM,MDIM)
C
C   COMPLEX PSI,CORR
C
C   PY=3.1415926535
C
C   MDIM2=MDIM/2
C
C   DO 10 M=1,MDIM
C   DO 10 N=1,MDIM
PHIX=2.*PY*XTILT*(M-MDIM2)/(MDIM-1)
PHIY=2.*PY*YTILT*(N-MDIM2)/(MDIM-1)
PHI=ISIGN*(PHIX+PHIY)
CORR=CMPLX (COS (PHI) , SIN (PHI) )
10 PSI (M,N)=PSI (M,N) *CORR
C
C   RETURN
END
```

