

7  
2ej



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

**Un algoritmo para la  
reconstrucción de frentes  
de onda a partir de  
imágenes desenfocadas**

**T E S I S**

Que para obtener el Grado de

**F I S I C O**

P r e s e n t a:

**REMY FERNAND /AVILA FOUCAT**

México, D. F.

1993

**TESIS CON  
FALLA DE ORIGEN**



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# INDICE

<b>Capítulo I. Introducción</b>	1
<b>Capítulo II. Ecuación de Roddier</b>	4
II.1 Ecuación de transporte de irradiancia	4
II.2 Ecuación de Roddier	6
<b>Capítulo III. Algoritmo</b>	9
III.1 Obtención de $\phi$	11
III.2 Obtención de $\xi$	15
III.3 Obtención de $W$	16
<b>Capítulo IV. Pruebas</b>	18
IV.1 Método	18
IV.2 Resultados	20
<b>Capítulo V. Aplicaciones</b>	31
V.1 Función de transferencia de la atmósfera	32
V.2 Probando la óptica de un telescopio	35
<b>Capítulo VI. Conclusión</b>	37
<b>Apéndice A</b>	40
<b>Apéndice B</b>	42
<b>Apéndice C</b>	44
<b>Referencias</b>	62

# CAPITULO I

## Introducción

Siempre es deseable conocer con precisión las características de cualquier instrumento de observación o de medición. En este sentido, los instrumentos ópticos no son la excepción, y una forma de caracterizarlos es midiendo la distribución de fase de la onda electromagnética, o frente de onda, sobre la pupila de salida del instrumento. A partir de esto, se pueden calcular las características de la señal óptica a la salida del instrumento. Por ejemplo, en el caso en que el instrumento sea un telescopio, y la señal una imagen, se puede calcular la resolución angular de la imagen.

En astronomía, cualquier incremento del poder de resolución de un telescopio, es de gran importancia científica. Para este efecto, es necesario conocer las características del sistema óptico, para poder mejorarlas. Es por esto que en el Instituto de Astronomía de la U.N.A.M. (IAUNAM) se ha intentado instrumentar un método para la medición de frentes de onda. Existen básicamente dos técnicas para reconstruir el frente de onda. La primera constituye el método de Shack-Hartmann<sup>1</sup>, que consiste en medir la derivada del frente

de onda, usando un arreglo de muchas lentes pequeñas. La segunda técnica fue propuesta por F. Roddier<sup>2,4</sup>, y consiste en medir la curvatura del frente de onda y su derivada radial en la frontera de la pupila de salida, a partir de imágenes desenfocadas. Se ha reportado<sup>4</sup> que ambos métodos tienen misma razón señal a ruido. Sin embargo, el método de Shack-Hartmann tiene el inconveniente de que su instrumentación es más complicada y más cara. Este método requiere de una gran cantidad de micro-lentes, un detector bidimensional, y una computadora, básicamente. En cambio, el método de Roddier requiere únicamente del detector bidimensional, y de la computadora. Esta es una de las razones por las cuales en el IAUNAM se ha dirigido el interés hacia la técnica propuesta por Roddier.

Este método consiste en medir la distribución de irradianza (a lo que se le llamará "tomar una imagen") a una distancia  $l$  antes del foco del sistema óptico, y tomar otra imagen a la misma distancia  $l$  después del foco. Esta última imagen se rota 180 grados, y se le resta a la primera. El resultado es proporcional al laplaciano del frente de onda en todo el plano de la pupila, salvo en la frontera, donde se le resta la derivada radial. De esta manera se obtiene una ecuación diferencial parcial con condiciones de frontera, donde la función que se busca determinar es el frente de onda. F. Roddier<sup>6</sup> propuso un algoritmo para resolver esta ecuación, basado en transformadas de Fourier iterativas, que pretendía cumplir con la condición del laplaciano, e imponer las condiciones de frontera a la vez. Durante varios meses se intentó implementar este algoritmo. Se le aportó una primera modificación para ajustar el laplaciano. Pero la condición de frontera nunca pudo ser impuesta con el algoritmo propuesto por Roddier. Por medio de comunicaciones personales con F. Roddier, nos enteramos de que ese algoritmo se usaba para encontrar una aproximación de orden cero del frente de onda, para posteriormente emplear métodos de relajación. Entonces, se ideó un algoritmo original para imponer las condiciones de frontera.

Esta tesis se centra en la realización y prueba del algoritmo completo que resuelve la ecuación de Roddier, y esta compuesta de los siguientes capítulos, después de la introducción: El capítulo II es una demostración de la ecuación de Roddier, basada en la

ecuación de transporte de irradiancia. En el capítulo III se explica el algoritmo que resuelve la ecuación de Roddier. En el cuarto capítulo se explican y exponen la realización del programa de cómputo, que aparece en apéndice, y las pruebas a las que se sometió el programa. El capítulo V habla de algunas posibles aplicaciones del algoritmo, y finalmente el último capítulo es una conclusión.

# CAPITULO II

## Ecuación de Roddier

En este capítulo se demuestra la ecuación de Roddier<sup>2,3,4</sup> partiendo de la ecuación de transporte de irradiancia. El esquema de esta demostración aparece en la referencia 3.

Se quiere medir el frente de onda en la pupila de salida  $P$ , del sistema óptico bajo estudio, representado con una lente  $L_0$  de distancia focal  $f$  (ver figura 1). En el foco imagen  $F_0$  de  $L_0$ , se coloca una lente  $L_1$  de distancia focal  $f/2$ . Un haz colimado de rayos paraxiales se propaga en dirección  $z$ . La función de la lente  $L_1$  es formar la imagen de la pupila  $P$ , a una distancia  $f$  de  $F_0$ , y así hacer el sistema completamente simétrico respecto a  $F_0$ . Cabe señalar que la lente  $L_1$  no desvia los rayos ya que estos convergen en  $F_0$  donde se coloca el centro de  $L_1$ .

### II.1 Ecuación de transporte de irradiancia

La amplitud compleja del campo eléctrico esta descrita por:

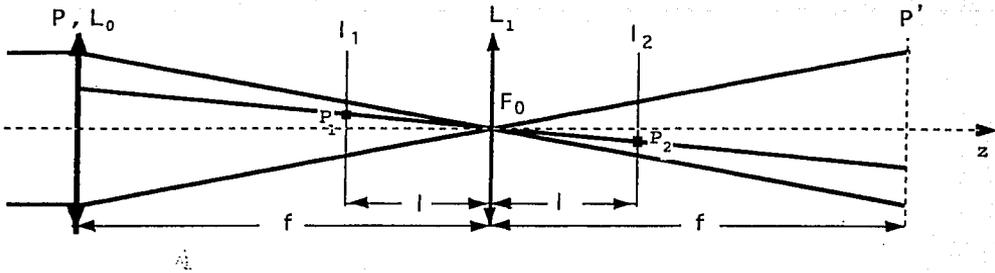


Figura 1: Sistema óptico para la captura de las imágenes intrafocal  $I_1$  y extrafocal  $I_2$ .

$$\Psi(x, y, z) = \sqrt{I(x, y, z)} e^{ikW(x, y, z)}, \quad (1)$$

donde  $I = I(x, y, z)$  es la irradiancia,  $kW = kW(x, y, z)$  es la distribución de fase,  $W$  en unidades de distancia,  $k = \frac{2\pi}{\lambda}$  es el número de onda y  $\lambda$  la longitud de onda.

Suponiendo una propagación de rayos paraxiales, a lo largo del eje  $z$ , la ecuación de transporte de irradiancia<sup>5</sup> es:

$$\frac{\partial I}{\partial z} = -(\nabla I \cdot \nabla W + I \nabla^2 W), \quad (2)$$

$\nabla$  es el operador  $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ ,  $x$  y  $y$  son las coordenadas en un plano transversal, perpendicular al eje  $z$ .

$\nabla W$  y  $\nabla I$  son vectores en el plano transversal, en la dirección de máximo cambio de  $W$  y de  $I$  respectivamente.  $\nabla W(x, y)$  da una medida de la inclinación local del frente de onda. La dirección de propagación de la luz es normal al frente de onda, por lo tanto,  $\nabla W(x, y)$  también indica esta dirección. De hecho  $\nabla W$  es un vector proporcional a la

proyección del vector de propagación sobre el plano transversal. Por todo esto, el término  $\nabla W \cdot \nabla I$ , en la ecuación 2, puede ser considerado como la variación de de irradiancia, en la dirección  $z$ , debida a la inhomogeneidad del haz ( $\nabla I \neq 0$ ) causada a su vez por la inclinación local del frente de onda.

$\nabla^2 W$  representa la curvatura local del frente de onda, y da una medida de la convergencia o divergencia de los rayos. Entonces el término  $I \nabla^2 W$  representa la variación de la irradiancia debida a la convergencia o divergencia de los rayos luminosos.

La ecuación 2 se puede escribir como:

$$\nabla \cdot (I \nabla W) = -\frac{\partial I}{\partial z} \quad (3)$$

De esto se puede ver que la ecuación de transporte de irradiancia (ec. 2) representa una ecuación de conservación de energía por unidad de tiempo en el plano transversal, análoga a una ecuación de conservación de carga, o de masa, en el espacio:  $\text{div}(\rho \vec{v}) = -\frac{\partial \rho}{\partial t}$ , donde  $\rho$  es la densidad de carga, o de masa, y  $\vec{v}$  es la velocidad.

## II.2 Ecuación de Roddier

Se supone que la iluminación en la pupila es uniforme, es decir que:

$$I(x, y, 0) = I_0 P(x, y), \quad \text{con} \quad P(x, y) = \begin{cases} 1, & \text{si } x^2 + y^2 < R; \\ 0 & \text{si no.} \end{cases} \quad (4)$$

Donde  $R$  es el radio de la pupila, y  $P(x, y)$  es la función pupila. Para simplificar los cálculos, en lo que sigue se utiliza una  $R$  unitaria. De esta manera, en la pupila se tiene:

$$\nabla I = -I_0 \delta_c \vec{n}. \quad (5)$$

donde  $\delta_c$  es una distribución impulsional a lo largo de la frontera de la pupila, y  $\vec{n}$  es un vector unitario radial, apuntando hacia fuera de la pupila. Aplicando la ecuación 2 en la pupila, y usando la ecuación 5 se tiene:

$$\frac{\partial I_0}{\partial z} = I_0 \delta_c \bar{n} \cdot \nabla W - P I_0 \nabla^2 W \quad (6)$$

$$\frac{\partial I_0}{\partial z} = I_0 \delta_c \frac{\partial W}{\partial \bar{n}} - P I_0 \nabla^2 W. \quad (7)$$

El principio del método es tomar una imagen  $I_1$  antes del foco, a una distancia  $l$  del mismo, y otra imagen  $I_2$  después del foco, a la misma distancia  $l$  del foco. Estas imágenes se van a restar de tal manera que, a la irradiancia en un punto  $P_1$  de  $I_1$ , se le reste la irradiancia del punto  $P_2$  de  $I_2$ , iluminado por el rayo que pasa por  $P_1$  (ver figura 1). Para esto, se tendrá que invertir la imagen  $I_2(\vec{r})$ , obteniendo  $I_2(-\vec{r})$ , donde  $\vec{r}$  designa la posición en un plano ortogonal al eje  $z$ . Se considera que  $f$  es mucho mayor que el diámetro de la pupila. En ese caso, es válida la aproximación de rayos paraxiales, en óptica geométrica, por lo cual se pueden despreciar las derivadas, de ordenes mayores que el primero, en un desarrollo limitado en series de potencias, de la irradiancia. Se tiene, entonces:

$$I_1(\vec{r}) = I_0(\vec{r}) + \frac{\partial I_0(\vec{r})}{\partial z} (f - l). \quad (8)$$

A través de la lente  $L_1$ , a una distancia  $2f$  de  $P$ , se forma la imagen  $I'_0$  de  $I_0$ . Se tiene entonces  $I'_0(\vec{r}) = I_0(-\vec{r})$ . De esta manera se tiene:

$$I_2(\vec{r}) = I'_0(\vec{r}) - \frac{\partial I'_0(\vec{r})}{\partial z} (f - l). \quad (9)$$

Por lo tanto, al invertir la imagen:

$$I_2(-\vec{r}) = I_0(\vec{r}) - \frac{\partial I_0(\vec{r})}{\partial z} (f - l). \quad (10)$$

La cantidad que se calcula a partir de estas imágenes, que llamaremos la señal del sensor, es:

$$S = \frac{I_1(\vec{r}) - I_2(-\vec{r})}{I_1(\vec{r}) + I_2(-\vec{r})}. \quad (11)$$

Introduciendo las ecuaciones 8 y 10 en 11:

$$S = \frac{\frac{\partial I_e}{\partial z}(f-l)}{I_0}. \quad (12)$$

Sustituyendo la ecuación 7 en la 12 se tiene:

$$S = \left( \delta_c \frac{\partial W}{\partial \vec{n}} - P \nabla^2 W \right) (f-l). \quad (13)$$

Finalmente, se debe considerar que en los planos donde se toman las imágenes, el tamaño del haz se modifica de un factor  $l/f$  respecto a su tamaño en la pupila. Esto produce un aumento de la irradiancia en un factor  $f/l$ . De esta manera se tiene en realidad:

$$P(\vec{r}) = \begin{cases} f/l, & \text{si } r < Rl/f; \\ 0 & \text{si no} \end{cases} \quad (14)$$

y entonces también se introduce un factor  $f/l$  en  $\delta_c$ .

La señal del sensor, dentro de la zona iluminada, de radio unitario, queda finalmente:

$$S = \left( \delta_c \frac{\partial W}{\partial \vec{n}} - \nabla^2 W \right) \frac{f(f-l)}{l}. \quad (15)$$

Esta es la ecuación a la que se quería llegar, y a la que se le llamará *ecuación de Roddier*.

Introduciendo las unidades adecuadas ( $W$  en unidades de  $\lambda$ , y una pupila de radio  $R$ ), la ecuación de Roddier se escribe como

$$S = \left( \delta_c \frac{\partial W}{\partial \vec{n}} - \nabla^2 W \right) \frac{f(f-l)\lambda}{lR^2}. \quad (16)$$

# CAPITULO III

## Algoritmo

El algoritmo que aquí se presenta fue inspirado en un artículo de Roddier<sup>6</sup>, aunque de éste sólo se conservó la idea de la transformada de Fourier iterativa.

La ecuación 15 puede dividirse en una ecuación de Poisson (EP), y una condición de frontera (CF).

$$\nabla^2 W(x, y) = -\frac{l}{f(f-l)} S(x, y) \quad \text{para } x^2 + y^2 < R^2 \quad (EP)$$

$$y \quad \frac{\partial W}{\partial r}(r = R, \theta) = \frac{\partial W(R, \theta)}{\partial r} = \frac{l}{f(f-l)} S(R, \theta) + \nabla^2 W(R, \theta). \quad (CF)$$

El término  $\nabla^2 W(R, \theta)$  que aparece en la CF es una extrapolación de  $\nabla^2 W(x, y)$  para  $x^2 + y^2 = R^2$ , de manera que el miembro derecho de la CF constituye una cantidad conocida. De esto se habla más adelante. Es de notar que para determinar completamente  $W(x, y)$ , falta otra condición. Sin embargo la condición que falta sólo serviría para determinar una

constante aditiva, lo cual no es de interés. Esta constante se escogerá arbitrariamente, como se verá más adelante. También se debe aclarar que en este algoritmo se considera que la pupila es circular. En particular, en el caso de un telescopio, la sombra del espejo secundario no está tomada en consideración.

La EP y su CF son lineales, por lo tanto se puede proponer una solución de la forma:

$$W(x, y) = \phi(x, y) + \xi(x, y). \quad (17)$$

$\phi$  es una función tal que:

$$\nabla^2 \phi(x, y) = -\frac{l}{f(f-1)} S(x, y) \equiv f(x, y) \quad \text{para } x^2 + y^2 < R^2. \quad (18)$$

Aquí se ha definido la función  $f(x, y)$  por comodidad. Una vez encontrada una  $\phi$  que cumpla con la ecuación 18, se busca la función  $\xi$  tal que:

$$\nabla^2 \xi(x, y) = 0 \quad \text{para todas } x, y. \quad (19)$$

$$y \quad \frac{\partial \xi}{\partial r}(R, \theta) = \frac{l}{f(f-1)} S(R, \theta) + \nabla^2 \phi(R, \theta) - \frac{\partial \phi}{\partial r}(R, \theta). \quad (20)$$

Se debe recalcar que  $\nabla^2 \phi(R, \theta)$  y  $\frac{\partial \phi}{\partial r}(R, \theta)$  son una extrapolación ya que  $f(x, y)$  se conoce para radios estrictamente menores que  $R$  (ver ecuación 18). Sin embargo esta extrapolación no constituye una fuente considerable de error, ya que se espera que la variación de  $\phi$  sea suficientemente pequeña entre un punto de muestreo y sus primeros vecinos.

Por comodidad se define la función  $g(\theta)$  igual al miembro derecho de la ecuación 20. Así se tiene:

$$\frac{\partial \xi}{\partial r}(R, \theta) = g(\theta). \quad (21)$$

Sustituyendo la ecuación 17 en EP y en CF, y considerando las ecuaciones 18, 19, y 20, se ve que la solución propuesta es en efecto una solución del problema.

¿ Y como se obtienen  $\phi$  y  $\xi$  ?

### III.1 Obtención de $\phi$

El algoritmo que se utiliza, similar al de F. Roddier<sup>6</sup>, se basa en un método de tipo Gershberg<sup>7</sup> de super-resolución.

Se tiene un arreglo cuadrado de  $nr$  renglones por  $nc$  columnas, en el cual se encuentra centrada la función  $f(x, y)$  muestreada que se notará :

$$f(x_j; y_i) \equiv f_{i,j} \quad \text{con} \quad x_j = \Delta x(j-1) \quad \text{y} \quad y_i = \Delta y(i-1). \quad (22)$$

$\Delta x$  y  $\Delta y$  son la distancia de muestreo en la dirección  $x$  y  $y$  respectivamente. Los índices  $i$  y  $j$  corren de 1 hasta  $nc$  (número de columnas) y  $nr$  (número de renglones) respectivamente.

Usando la teoría de diferencias finitas<sup>8</sup>, la derivada parcial de  $\phi$  respecto a  $x$ , evaluada en el punto imaginario  $(i, j - 1/2)$ , se escribe (ver figura 2 a):

$$\frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta x} \quad (23)$$

La segunda derivada parcial evaluada en el punto  $(i, j)$  se calcula a partir de las primeras derivadas evaluadas en los puntos imaginarios  $(i, j - 1/2)$  e  $(i, j + 1/2)$ . Es decir:

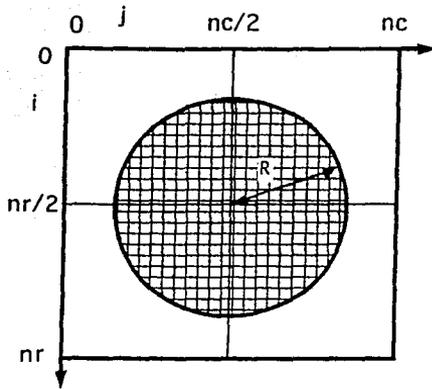
$$\frac{\partial^2 \phi_{i,j}}{\partial x^2} = \frac{\frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta x} - \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta x}}{\Delta x} \quad (24).$$

Simplificando:

$$\frac{\partial^2 \phi_{i,j}}{\partial x^2} = \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{(\Delta x)^2}. \quad (25)$$

De manera análoga :

$$\frac{\partial^2 \phi_{i,j}}{\partial y^2} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{(\Delta y)^2}. \quad (26)$$

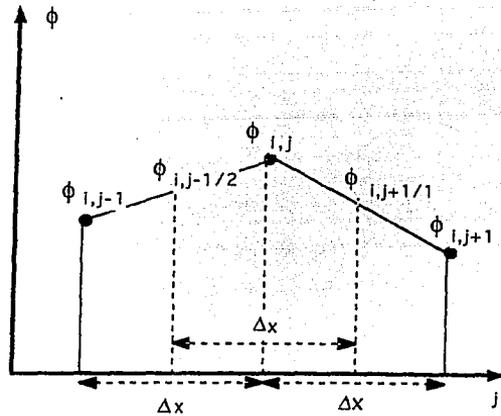


Datos



Cero

a)



b)

Figura 2 : a) Arreglo cuadrado donde se guarda la señal del sensor.

b) Derivadas parciales de una función discreta.

Sumando las ecs 25 y 26, y suponiendo que  $\Delta x = \Delta y \equiv \Delta$ , lo cual es el caso, se obtiene el laplaciano de  $\phi$ , y la EP se escribe :

$$\phi_{i,j+1} + \phi_{i,j-1} + \phi_{i-1,j} + \phi_{i+1,j} - 4\phi_{i,j} = f_{i,j}\Delta^2. \quad (27)$$

En el apéndice A se demuestra que si se calcula la transformada de Fourier discreta en ambos miembros de la ecuación 27, y se despeja la transformada de Fourier de  $\phi$ , se obtiene:

$$\hat{\phi}_{m,n} = \frac{\Delta^2 \hat{f}_{m,n}}{2 \left[ \cos\left(\frac{2\pi}{nc}n\right) + \cos\left(\frac{2\pi}{nr}m\right) - 2 \right]} \quad (28)$$

Los subíndices  $m$  y  $n$  corren desde 0 hasta  $nr - 1$  y  $nc - 1$  respectivamente.  $\hat{f}_{m,n}$

representa la transformada de Fourier de  $f_{i,j}$ . Calculando la transformada de Fourier inversa de la última ecuación, se obtiene en principio  $\phi$ . Sin embargo, la cosa no es tan sencilla. La transformada de Fourier de  $\phi$ , se extiende en todo el espacio de frecuencias, porque  $\phi$  no es analítica, dada la discontinuidad en la frontera de la pupila. Pero sólo se tiene una porción del espacio de frecuencias ya que se trabaja con arreglos finitos, por supuesto. Esto quiere decir que  $\hat{\phi}_{m,n}$  tiene un cierto error respecto a la transformada de Fourier de  $\phi$ . Para disminuir la energía de este error, se hace lo siguiente: Una vez calculada la transformada de Fourier inversa de la ecuación 28, lo cual da una estimación de  $\phi$  que se extiende en todo el arreglo, se calcula su laplaciano y se reemplaza por  $f_{i,j}$  sólo dentro de la pupila. La función así obtenida se usa como entrada al programa y se comienza de nuevo. Este algoritmo es el de Gershberg<sup>7</sup> de superresolución, muy ligeramente modificado.

Se ve que si  $m = 0$  y  $n = 0$ , lo cual corresponde al origen de frecuencias, la ecuación 28 diverge. Por lo tanto para esos valores de  $m$  y  $n$ , no se aplica esa ecuación, y se escoge arbitrariamente :

$$\hat{\phi}_{0,0} = 0. \quad (29)$$

Esto equivale a fijar arbitrariamente el promedio de  $\phi$ , calculado sobre todo el arreglo, ya que :

$$\hat{\phi}_{0,0} = \Delta^2 \sum_{i,j} \phi_{i,j}. \quad (30)$$

Esta arbitrariedad no es importante, ya que es equivalente a introducir una aberración de pistón, la cual no interesa para nada.

Así pues, el algoritmo para el cálculo de  $\phi$  es el siguiente (ver figura 3):

Se calcula la transformada de Fourier discreta de  $f$ , con un algoritmo de *Fast Fourier transform*<sup>8</sup>. Se obtiene una función que se extiende en todo el arreglo cuadrado. Después se efectúa la operación de la ecuación 28 y se fija el valor 0 en el origen de frecuencias. Enseguida se calcula la transformada de Fourier inversa, y se obtiene una primera estimación

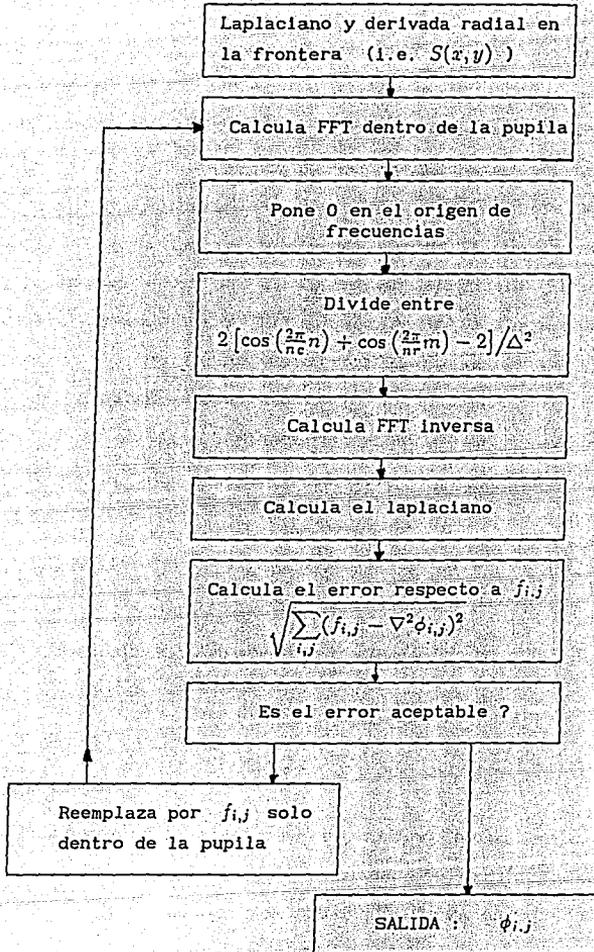


Figura 3: Algoritmo para obtener una solución de una ecuación de Poisson, usando transformadas de Fourier iterativas.

de  $\phi$  que se extiende sobre todo el arreglo, aún fuera de la zona de la pupila. Después se calcula el laplaciano de  $\phi$  en todo el arreglo. Se calcula el error cuadrático promedio de este laplaciano respecto al laplaciano original, es decir  $f$ , pero únicamente dentro de la pupila. Esta cantidad se compara con el promedio cuadrático de  $f$ . Esto equivale a comparar la energía del error con la energía de la señal original. Si el error no es satisfactorio, entonces se reemplaza el laplaciano de  $\phi$  por el original  $f$ , solamente dentro de la pupila, dejando el resto intacto, y se vuelve a comenzar hasta que el error sea satisfactorio.

### III.2 Obtención de $\xi$

La función  $\xi$  debe satisfacer la ecuación 19 con la condición de frontera 21 periódica en  $\theta$ . Esto último inspira a que se proponga la forma siguiente para  $\xi$ :

$$\xi(r, \theta) = \sum_{k,l,p}^{\infty} r^k (a_l \cos(l\theta) + b_p \sin(p\theta)). \quad (31)$$

Donde  $a_l$  y  $b_p$  son coeficientes. En el apéndice B se demuestra que para que el laplaciano de  $\xi$  sea cero,  $k$ ,  $l$ , y  $p$  deben ser iguales. Por lo tanto  $\xi$  es de la forma:

$$\xi(r, \theta) = \sum_p^{\infty} r^p (a_p \cos(p\theta) + b_p \sin(p\theta)). \quad (32)$$

El problema se resume a encontrar los coeficientes  $a_p$  y  $b_p$ . Para esto se sustituye la expresión 32 en la condición de frontera 21, obteniendo:

$$\sum_p^{\infty} pR^{p-1} (a_p \cos(p\theta) + b_p \sin(p\theta)) = g(\theta). \quad (33)$$

Para encontrar  $a_n$ , se multiplica por  $\cos(n\theta)$ , y se integra en  $\theta$ , de  $-\pi$  a  $\pi$ . Gracias a la ortogonalidad de las funciones coseno, y seno, se puede despejar  $a_n$ , obteniendo:

$$a_n = \frac{1}{\pi n R^{n-1}} \int_{-\pi}^{\pi} g(\theta) \cos(n\theta) d\theta. \quad (34)$$

De manera análoga se obtiene  $b_n$ :

$$b_n = \frac{1}{\pi n R^{n-1}} \int_{-\pi}^{\pi} g(\theta) \sin(n\theta) d\theta. \quad (35)$$

En principio se deben calcular una infinidad de coeficientes, ya que la suma, en la expresión de  $\xi$  se extiende hasta el infinito. Pero en la práctica, evidentemente, sólo se calcula un número finito de términos de la suma. Entre más grande sea  $n$ , el término correspondiente en la sumatoria contiene frecuencias espaciales más altas. Todo sistema óptico es impermeable a frecuencias espaciales más altas que un cierto límite llamado frecuencia de corte<sup>11</sup>. Resulta entonces inútil tratar de determinar un término de la suma que tenga una frecuencia mayor que la frecuencia de corte del sistema óptico bajo prueba. Por esto el número de coeficientes  $a_n$  y  $b_n$  que se calculan, es finito, y depende de cada aplicación particular. Teniendo  $a_n$  y  $b_n$ , se usa la expresión 19 para calcular  $\xi$ .

Los cálculos numéricos se realizan con arreglos cuadrados, y en la teoría se tienen funciones angulares, como  $g(\theta)$ . Esto provocó cierta dificultad particularmente para calcular las integrales en las ecuaciones 34 y 35, para lo cual se ideó un algoritmo especial. Primero se calcula  $g(\theta)$ . Para esto, se calcula el miembro derecho de la ecuación 20 en todo el arreglo, es decir para toda  $r$  y  $\theta$ . Después se van tomando los valores en los elementos del arreglo que se encuentren a una distancia comprendida en el intervalo  $[R, R+1]$ , escogiendo estos elementos de manera que  $\theta$  vaya siempre en aumento. Una vez obtenida  $g(\theta)$ , se calculan las integrales con un método tradicional.

### III.3 Obtención de $W$

La ecuación 17 muestra que  $W$  es simplemente la suma de  $\phi$  y  $\xi$ . En realidad existe otro detalle. Si bien el valor del promedio de  $W$  no tiene importancia, dado que corresponde a una aberración de pistón, sí interesa tener control sobre él, sobre todo para probar el algoritmo. Por esto, se calcula el promedio de  $W$  dentro de la pupila, y el resultado se resta al valor de  $W$  en cada punto del arreglo. De esta manera se obtiene una función cuyo promedio, dentro de la pupila, es cero. Esta función es la solución al problema.

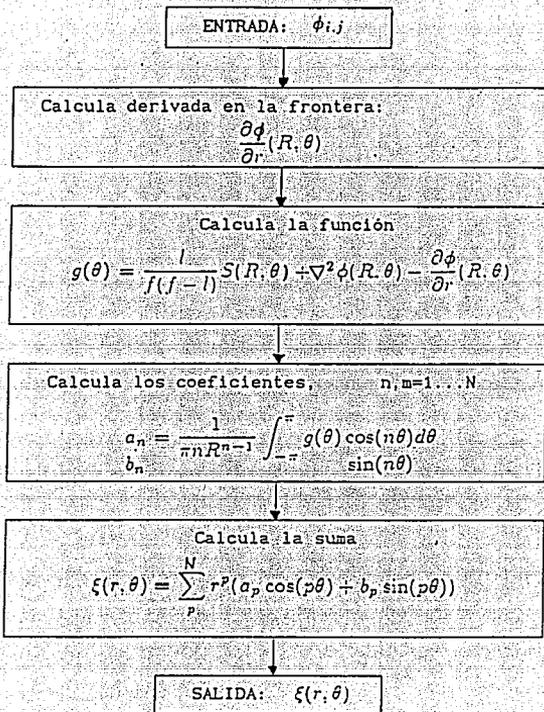


Figura 4: Algoritmo para resolver la ecuación de Laplace 19, con condiciones de frontera dadas por la ecuación 20.

# CAPITULO IV

## Pruebas

### VI.1 Método

El algoritmo presentado aquí requiere de una gran cantidad de cálculos, sobre todo a causa de las transformadas de Fourier, por lo que se escogió usar una computadora SUN. El lenguaje de programación empleado fue el FORTRAN 77, que permite programación estructurada. La rutina de transformación de Fourier en dos dimensiones empleada, proviene de las rutinas publicadas en la referencia 8, y en la graficación se empleó el programa SUPERMONGO. Todo lo demás fue desarrollado de manera independiente.

Se sometió el algoritmo a varias pruebas, siguiendo dos esquemas. El primero consiste en crear un archivo de entrada al programa, *image1.dat*, donde se guarda el laplaciano de un polinomio de Zernike dentro de una pupila circular, en cuya frontera se resta la derivada radial, todo calculado literalmente. Para estas pruebas se tomaron, por separado, aberraciones de esfericidad de cuarto orden<sup>10</sup>, cuyo interés estriba en su independencia

angular, coma de tercer orden<sup>10</sup>, ya que contiene mezclado algo de tilt, y astigmatismo de tercer orden<sup>10</sup>, debido a que su laplaciano es cero, lo cual permite evaluar la parte del algoritmo que se basa en la condición de frontera.

El segundo esquema consiste en crear un frente de onda igual a una combinación de 19 polinomios de Zernike<sup>10</sup>, cuyos coeficientes decrecen conforme aumenta el orden del polinomio. A este frente de onda se le calcula el laplaciano, así como su derivada radial en la frontera de la pupila, ambos con diferencias finitas. Con esto se crea el archivo de entrada al programa.

Por simplicidad, el factor  $(f - l)f/l$  se escogió igual a la unidad, al igual que la distancia de muestreo  $\Delta$ , lo cual corresponde a usar como unidad un píxel. El diámetro de la pupila se tomó igual a 160 píxeles, y se usó un arreglo de  $N$  por  $N$ , con  $N$  igual a 256. De esta manera, los errores por aliasing no son considerables.

El número de iteraciones determina el error en el laplaciano, cuya tolerancia depende de cada aplicación, por lo cual, para esta fase de pruebas, el programa pregunta cuantas iteraciones se quieren. El programa crea una serie de archivos de salida que son graficados una vez terminada su ejecución. (ver figuras 5-12). A continuación se explica el significado de cada gráfica, en el orden en que aparecen en las figuras.

La primera gráfica muestra curvas de nivel del frente de onda teórico, al que se le calcula el laplaciano y la derivada radial para introducir al programa. La segunda gráfica representa curvas de nivel del frente de onda reconstruido por el programa, del cual sólo interesa la zona de la pupila. La siguiente muestra en punteado y en línea continua cortes verticales que pasan por el centro del frente de onda teórico y del reconstruido respectivamente. La primera gráfica del segundo renglón, representa curvas de nivel de la diferencia de las dos primeras gráficas, o sea el residuo, del cual se muestra un corte vertical en la siguiente gráfica. La última del segundo renglón es una gráfica del error rms porcentual del laplaciano de  $W$  reconstruida, respecto al laplaciano de  $W$  teórica, en función del número de iteraciones. El último renglón contiene en primer término, un corte diametral de la distribución de irradiancia que se tendría en el plano imagen, si el frente de onda en la

pupila de salida del instrumento fuera igual al del residuo. Esto corresponde al caso en que se hubieran compensado las deformaciones del frente de onda, en cantidad exactamente igual al frente de onda reconstruido, como en un sistema de óptica activa o adaptiva, cuyo sistema de actuadores fuera perfecto. La segunda gráfica de este renglón es la misma que la anterior, con el eje horizontal ensanchado. Y la última, representa una imagen perfecta, lo que corresponde a un frente de onda plano. Finalmente, en la siguiente figura, se sobreponen las dos últimas gráficas, y se amplifican, de manera a poder compararlas. Las unidades de estas últimas cuatro gráficas, en el eje vertical, están normalizadas a la irradiancia máxima de una imagen perfecta, y en el eje horizontal, una unidad corresponde a  $\lambda/(\Delta N)$  radianes.

## VI.2 Resultados

En lo que sigue se analizan los resultados de las pruebas.

La figura 5 muestra que la aberración de esfericidad se reconstruye muy satisfactoriamente, después de 10 iteraciones. Se ve en la gráfica 4 que el residuo tiene picos de apenas un centésimo del valor máximo del frente de onda teórico. La gráfica 6 muestra que al cabo de la décima iteración, el error rms porcentual es de alrededor de 4%. La fuente de este error en el ajuste del laplaciano, parece provenir principalmente de las orillas del frente de onda, ya que es donde más diferencia existe entre  $W$  teórica y la  $W$  calculada. También se corrió el programa con 40 iteraciones, y el error bajó hasta un 2%, pero esto no provocó cambios sustantivos en la reconstrucción del frente de onda. En la figura 6 se ve que la imagen perfecta se sobrepone a la imagen correspondiente al residuo.

Para el caso de la aberración de coma, figuras 7 y 8, se observa que el frente de onda reconstruido no se ajusta perfectamente al teórico (gráfica 3), principalmente a causa de una aberración de tilt adicional, es decir, una inclinación del frente de onda (gráficas 4 y 5). Sin embargo la gráfica 6 indica que el error en el laplaciano es, al cabo de 10 iteraciones, muy pequeño, lo cual indica que el error proviene de ajustar las condiciones de frontera.

Esto suena coherente ya que el laplaciano de una aberración de tilt (i.e. de un plano inclinado) es cero. En la figura 8 se observa que el centro de la imagen correspondiente al residuo (punteada), está desplazado, lo cual corresponde en el plano de la pupila a la aberración de tilt. Aunque también se observa cierto ensanchamiento de la base. Esto indica que además del tilt, se tienen otras deformaciones del frente de onda, las cuales se aprecian ligeramente en la gráfica 5, sobre todo en las cercanías de la frontera de la pupila. Cabe recordar que la figura 8 muestra cortes de las imágenes, por lo cual uno no se debe alarmar si aparentemente el flujo de la imagen reconstruida es mayor que el de la imagen perfecta.

Las figuras 9 y 10 corresponden a una aberración de astigmatismo. En este caso se observa que el frente de onda reconstruido es prácticamente igual al frente de onda teórico, por lo cual la figura 10 muestra que las imágenes del residuo y del frente de onda plano, coinciden perfectamente. En este caso no aparece la gráfica de convergencia (gráfica 6) ya que el laplaciano del astigmatismo es cero. Por lo tanto no vale la pena hacer iteraciones para ajustarlo. Además, al calcular el error en el laplaciano, se divide 0 entre 0. Lo interesante de esta prueba, es que el frente de onda se reconstruye utilizando solamente el algoritmo que ajusta las condiciones de frontera.

Por último, las figuras 11 y 12, muestran la prueba con la combinación de 19 polinomios de Zernike, descrita anteriormente. Se empezó con 50 iteraciones, y los resultados no fueron muy satisfactorios. Entonces se decidió iterar 1000 veces. Lo primero que se ve, en la gráfica 6 de la figura 11, es que la disminución del error a partir de, aproximadamente, la iteración número 200, se vuelve despreciable. Sin embargo, las gráficas 1 a 4 muestran que el frente de onda reconstruido tiene ciertas diferencias respecto al teórico, aunque sean parecidos. En este caso se aprecia de nuevo una componente importante de tilt. En la figura 12 se observa que la imagen del residuo no sólo está desplazada, sino que además el pico central está entre 0.65 y 0.7 veces el pico de la imagen perfecta. Es decir que la imagen reconstruida tiene una razón de Strehl entre 0.65 y 0.7, lo cual puede ser suficiente, dependiendo de la aplicación. Una imagen con una razón de Strehl de 0.8 es considerada

como excelente en astronomía<sup>12</sup>.

De los resultados de estas pruebas, se pueden concluir inmediatamente dos cosas. La primera es que la manera de estimar el error no es muy eficiente, y la segunda es que el ajuste de las condiciones de frontera debe ser refinado. Estos dos temas se discuten más ampliamente en el último capítulo que corresponde a la conclusión.

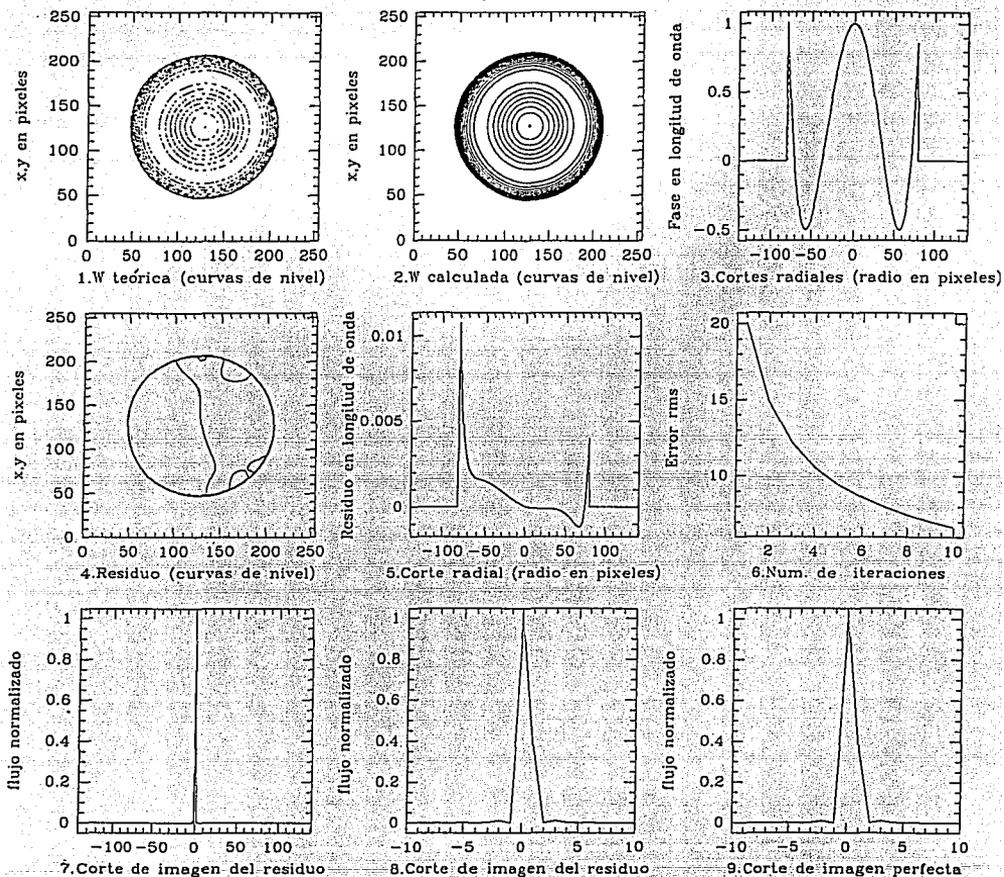


Figura 5: Reconstrucción de aberración de esfericidad. Las gráficas 8 y 9 se superponen en la figura 6

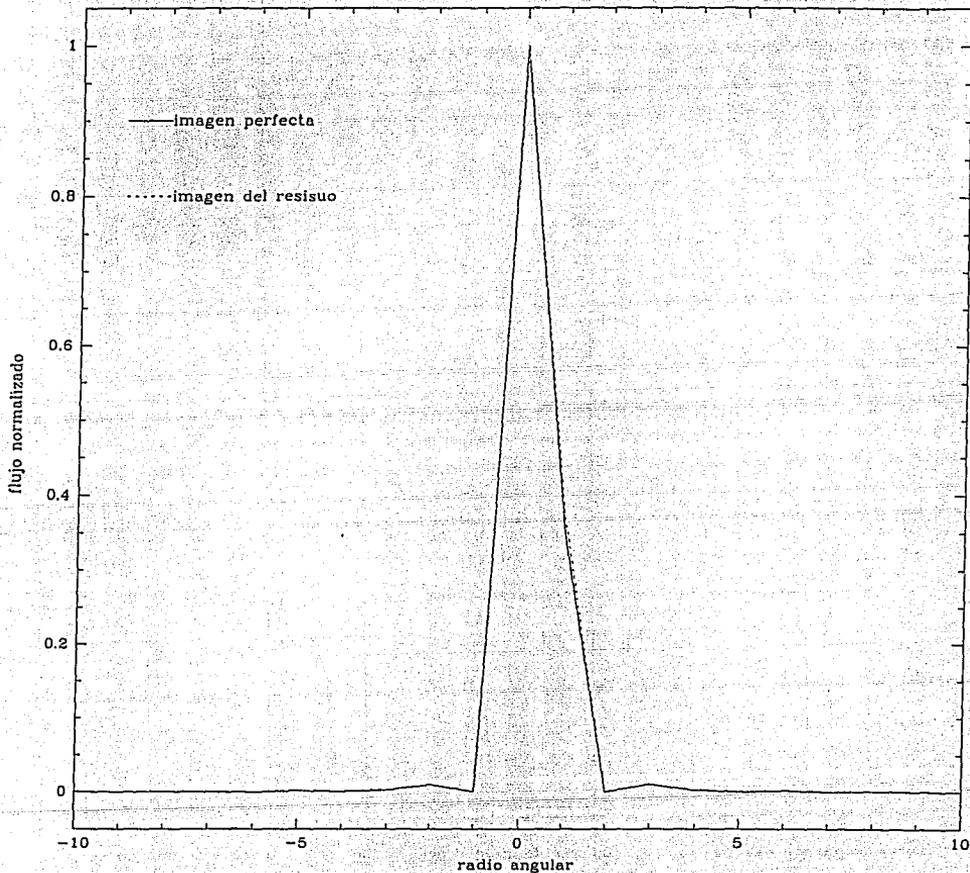


Figura 6: Imágenes perfecta (línea continua), y del residuo (línea punteada) en el caso de aberración de esfericidad. En el eje horizontal, una unidad corresponde a  $\lambda/(\Delta N)$ .

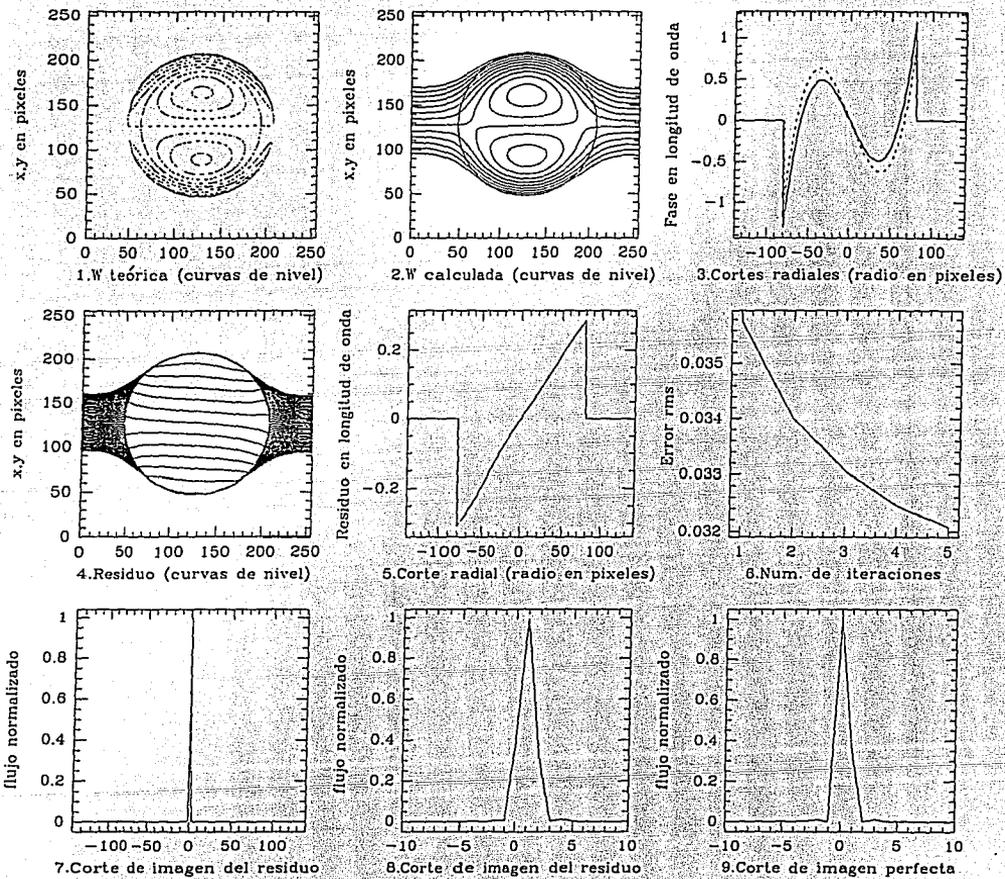


Figura 7: Reconstrucción de aberración de coma. Las gráficas 8 y 9 se sobreponen en la figura 8

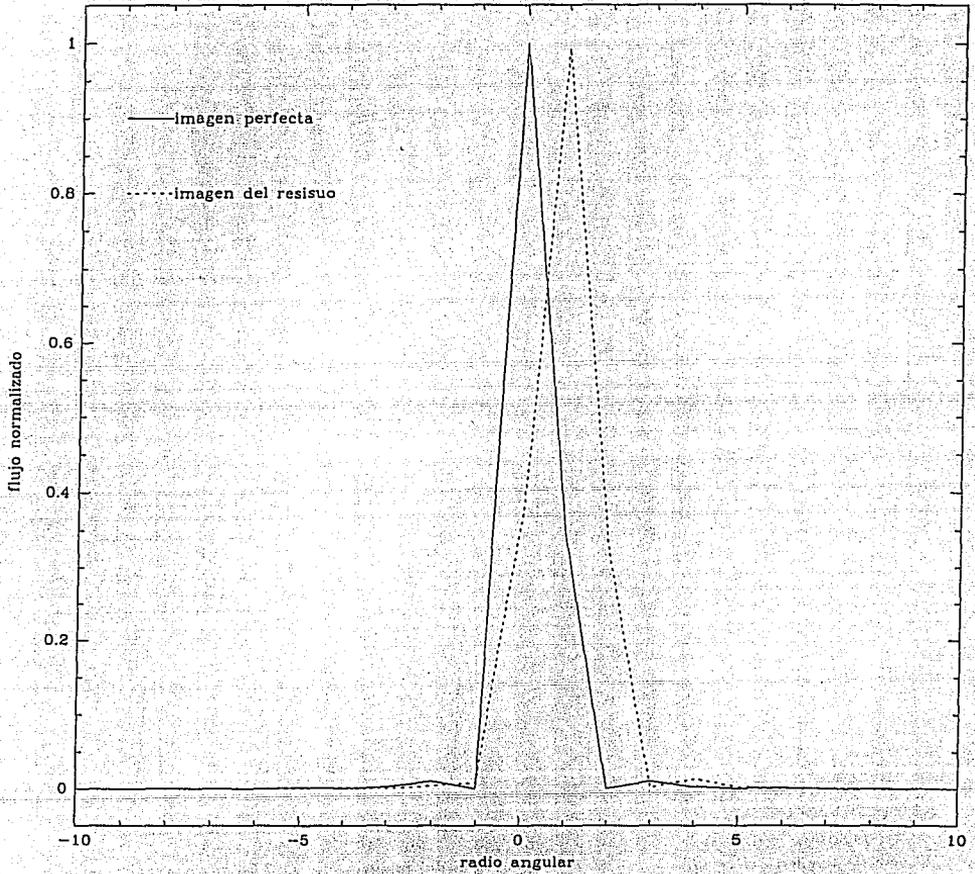


Figura 8: Imágenes perfecta (línea continua), y del resisuo (línea punteada) en el caso de aberración de coma. En el eje horizontal, una unidad corresponde a  $\lambda/(\Delta N)$ .

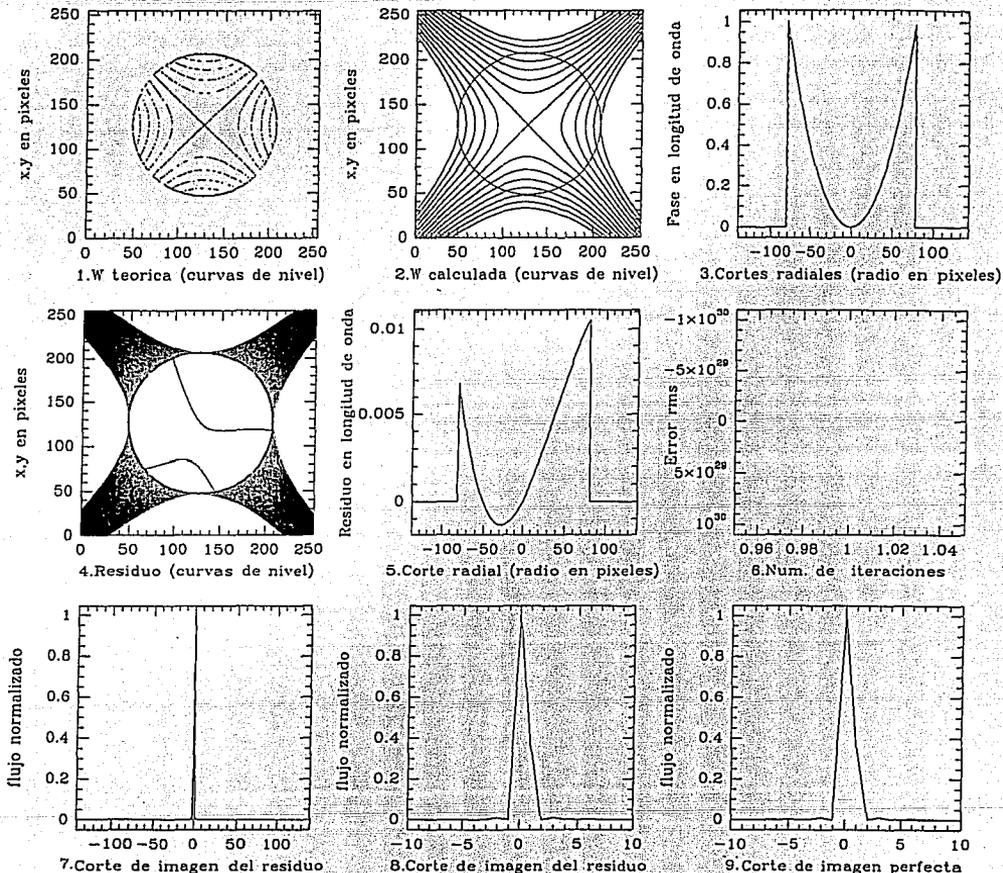


Figura 9: Reconstrucción de aberración de astigmatismo. Las gráficas 8 y 9 se sobreponen en la figura 10

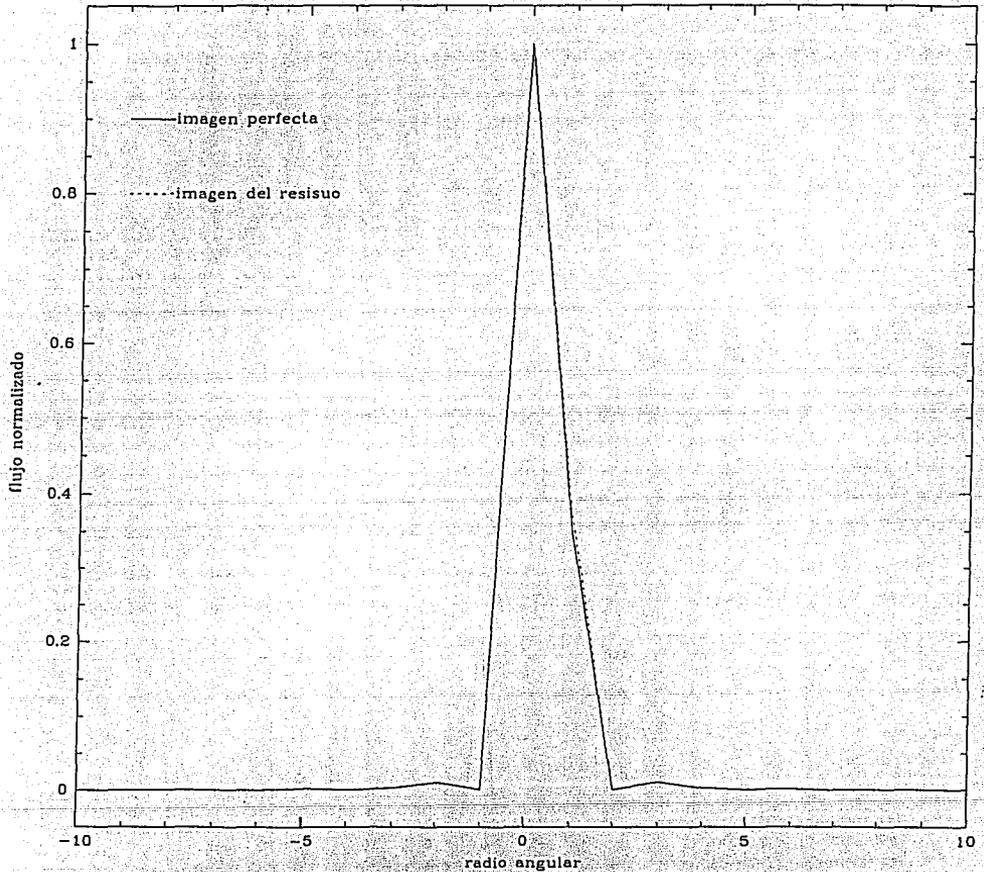


Figura 10: Imágenes perfecta (línea continua), y del residuo (línea punteada) en el caso de aberración de astigmatismo. En el eje horizontal, una unidad corresponde a  $\lambda/(\Delta N)$ .

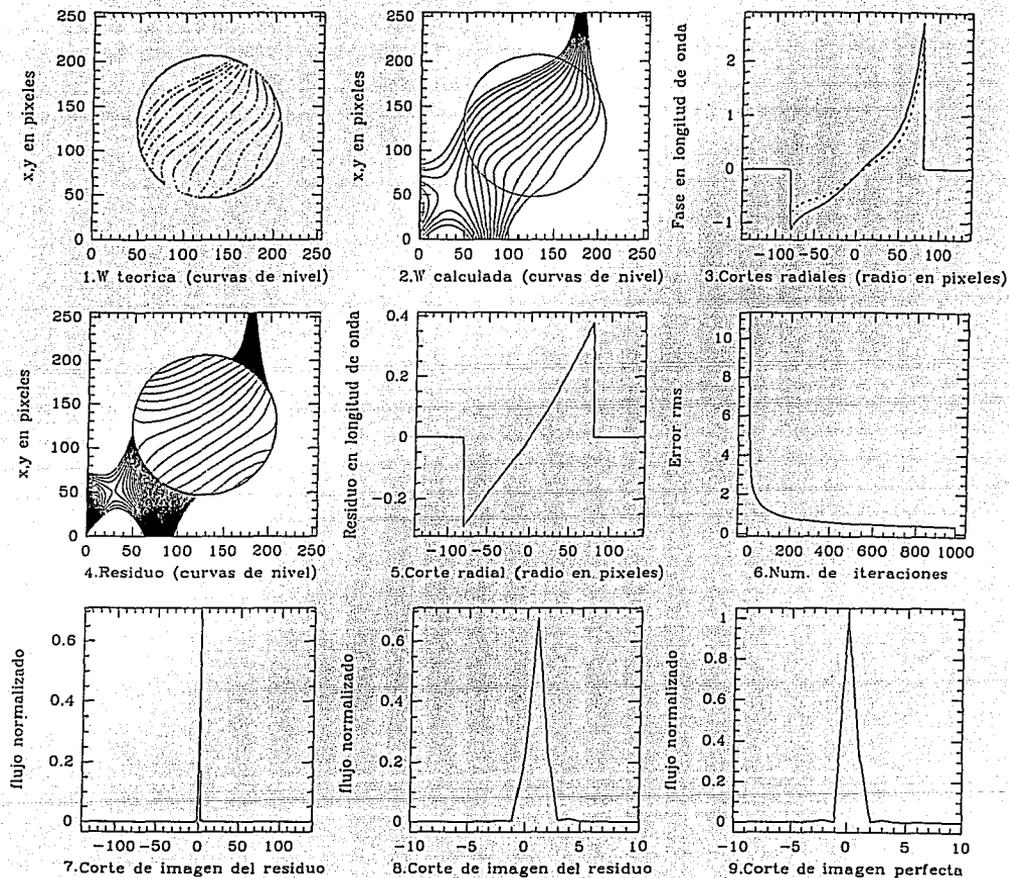


Figura 11: Reconstrucción de una combinación lineal de los 19 primeros polinomios de Zernike, cuyos coeficientes decrecen conforme aumenta el orden del polinomio. Las gráficas 8 y 9 se sobrepone en la figura 12

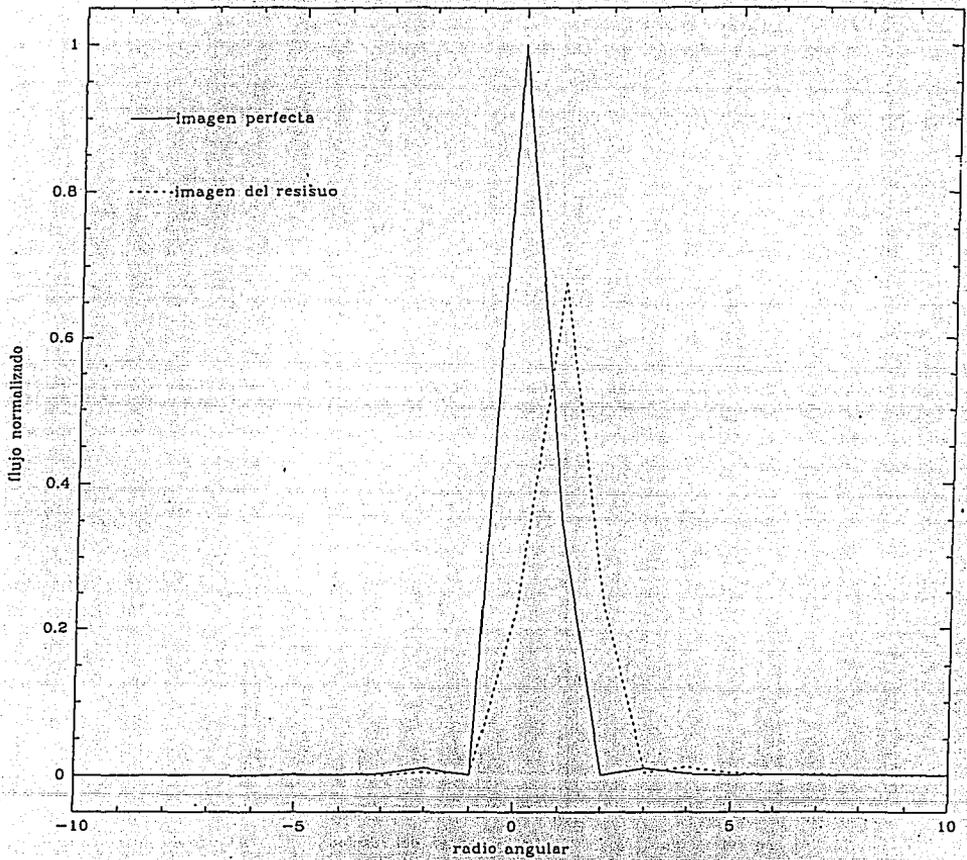


Figura 12: Imágenes perfecta (línea continua), y del residuo (línea punteada) en el caso de la combinación de 19 polinomios de Zernike. En el eje horizontal, una unidad corresponde a  $\lambda/(\Delta N)$ .

# CAPITULO V

## Aplicaciones

En principio, el método que se ha venido describiendo sirve para medir el frente de onda en la pupila de salida de cualquier instrumento óptico enfocador. En particular, interesa medir las aberraciones del telescopio de 2.1 metros de San Pedro Mártir, en un futuro cercano. Para este proposito el algoritmo debe ser modificado ligeramente de manera a considerar la sombra del espejo secundario, y sobre todo, debe ser afinado para que reconstruya perfectamente el frente de onda. En este caso, también podría ser usado como parte del sensor de frente de onda en un sistema de óptica activa. En estas dos posibles aplicaciones, la atmósfera interviene como parte del sistema óptico. Por esto, en este capítulo se presenta de manera esquemática el cálculo de la función de transferencia del sistema atmósfera-telescopio (Roddier<sup>9</sup>), y se discute brevemente la intervención de la turbulencia atmosférica en la medición de las aberraciones de un telescopio.

### V.1 Función de transferencia de la atmósfera

Un flujo es turbulento, si el número de Reynolds, dado por  $Re = vL/\nu$ , excede un valor crítico. En la atmósfera, el número de Reynolds del flujo de aire es de alrededor de  $10^6$ . Esto corresponde a una turbulencia llamada plenamente desarrollada.

Para nuestros fines, es interesante describir las fluctuaciones de índice de refracción. Partiendo del espectro de fluctuaciones de temperatura, dado por la ley de Kolmogorov, se calcula la función de estructura de estas fluctuaciones. A partir de esto, con la ayuda de la relación de Gladstone se calcula la función de estructura  $D_N$  de las fluctuaciones del índice de refracción  $N$ , definida como:

$$D_N(\vec{\rho}) = \langle |N(\vec{r} + \vec{\rho}) - N(\vec{r})|^2 \rangle \quad (36)$$

donde los signos " $\langle, \rangle$ " indican un promedio en un ensemble de realizaciones. Se obtiene el siguiente resultado:

$$D_N(\rho) = C_N^2 \rho^2/3 \quad (37)$$

$C_N^2$  se define como la constante de estructura de las fluctuaciones de índice de refracción.

El sistema óptico empleado en la observación astronómica comprende básicamente la atmósfera terrestre, y el telescopio. La iluminación se realiza con luz incoherente ( luz de las estrellas ), por lo cual el sistema es lineal respecto a la irradiancia. Por esto se estudia la función de transferencia de irradiancia del sistema atmósfera-telescopio  $\hat{S}(f)$ .

Suponiendo que se observa en el zenit, una fuente monocromática de longitud de onda  $\lambda$ , para largos tiempos de exposición, se obtiene:

$$\langle \hat{S}(\vec{f}) \rangle = \frac{1}{\rho} \int \langle \bar{\Psi}(\vec{u})\Psi(\vec{u} - \vec{f}) \rangle \bar{P}(\vec{u})P(\vec{u} - \vec{f})d\vec{u} \quad (38)$$

En esta expresión,  $\vec{f}$  es una frecuencia espacial,  $\rho$  es el área de la pupila de entrada del telescopio, en unidades de longitud de onda,  $P(\vec{u})$  es la función pupila, y  $\Psi$  es la

amplitud compleja de la onda que llega a la pupila de entrada, y que viene deformada por la atmósfera.

La cantidad

$$B_{\Psi}(\vec{f}) = \langle \bar{\Psi}(\vec{u})\Psi(\vec{u} - \vec{f}) \rangle . \quad (39)$$

sólo depende de  $\vec{f}$  por lo que puede salir de la integral. Definiendo:

$$T(\vec{f}) = \int \bar{P}(\vec{u})P(\vec{u} - \vec{f})d\vec{u} \quad (34)$$

se llega a la relación:

$$\langle \hat{S}(\vec{f}) \rangle = B_{\Psi}(\vec{f})T(\vec{f}). \quad (41)$$

$T(\vec{f})$  representa la función de transferencia del telescopio.  $B_{\Psi}$  es el momento de segundo orden de  $\Psi$ , y representa la función de transferencia de la atmósfera, para largos tiempos de exposición.

Considerando sólo una capa turbulenta delgada, de ancho  $\Delta h$  situada a una altura  $h$ , la función de transferencia de la atmósfera resulta ser :

$$B_{\Psi_h}(\vec{\epsilon}) = e^{-\frac{1}{2}D_{\phi}(\vec{\epsilon})}, \quad (42)$$

Donde  $D_{\phi}(\vec{\epsilon})$  representa la función de estructura de las variaciones de fase  $\phi$ , provocadas por las variaciones de índice de refracción, que se define como :

$$D_{\phi}(\vec{\epsilon}) = \langle |\phi(\vec{x} - \vec{\epsilon}) - \phi(\vec{x})|^2 \rangle . \quad (43)$$

El cálculo de esta cantidad se basa en la expresión de la función de estructura de las fluctuaciones de índice de refracción (ec. 37). Se llega a:

$$D_{\phi}(\vec{\epsilon}) = 2.91k^2 C_N^2 \Delta h \epsilon^{5/3}, \quad (44)$$

donde  $k$  es el número de onda.

Se obtiene finalmente el momento de segundo orden de las fluctuaciones de la amplitud compleja a la salida de una capa turbulenta delgada:

$$B_{\Psi_h}(\bar{x}) = e^{-1.45k^2 C_N^2 \Delta h \epsilon^{5/3}} \quad (45)$$

Y a la salida de capas continuas, observando a un ángulo  $\theta$  respecto al zenit, se obtiene:

$$B_{\Psi}(\bar{\epsilon}) = \exp\left(-1.49k^2 (\cos(\theta))^{-1} \epsilon^{5/3} \int_{atm} C_N^2(h_p) dh_p\right) \quad (46)$$

Una expresión más sencilla se obtiene en función del parámetro de Fried  $r_0$ , definido como el diámetro de un telescopio que en ausencia de atmósfera daría la misma resolución que la de un telescopio perfecto, de diámetro infinito, pero con atmósfera. Se obtiene:

$$B_{\Psi}(\bar{f}) = e^{-3.44 \left(\frac{f}{f_0}\right)^{5/3}} \quad (47)$$

Donde  $f_0 = r_0/\lambda$ . Comparando esta ecuación con la ecuación 42, se ve que la función de estructura de las variaciones de fase se escribe:

$$D_{\phi}(f) = 6.88 \left(\frac{f}{f_0}\right)^{5/3} \quad (48)$$

o bien:

$$D_{\phi}(r) = 6.88 \left(\frac{r}{r_0}\right)^{5/3} \quad (49)$$

Siguiendo el tratamiento de Fried<sup>13</sup>, calculando la integral, sobre todo el espacio de frecuencias, de la función de transferencia del sistema atmósfera-telescopio, se obtiene la resolución límite, según el criterio de Strehl. El inverso de esta cantidad representa el diámetro angular de las imágenes, de objetos puntuales, deformadas por la atmósfera. Se obtiene:

$$\alpha_{limite} = \begin{cases} 1.27 \frac{\lambda}{r_0} & \text{si } r_0 < D; \\ 1.27 \frac{\lambda}{D} & \text{si } r_0 > D. \end{cases} \quad (50)$$

$D$  es el diámetro del telescopio, supuesto exento de aberraciones.

Comparando las ecuaciones 46 y 47, se calcula la expresión teórica del parámetro de Fried:

$$r_0 = \left[ 0.423 k^2 (\cos(\theta))^{-1} \int C_N^2(h) dh \right]^{-3/5} \quad (51)$$

Y se ve la relación de proporcionalidad :

$$r_0 \propto \lambda^{6/5} \quad (52)$$

## V.2 Probando la óptica de un telescopio

El algoritmo presentado aquí, una vez refinado, puede ser usado para caracterizar la óptica de un telescopio. En realidad se deben aportar ciertas modificaciones para considerar la sombra del espejo secundario, que aparecería en las imágenes desenfocadas, en un telescopio reflector. Pero el principio básico es aplicable.

Estas imágenes provendrían de una estrella brillante. Por lo tanto, las deformaciones del frente de onda provocadas por las aberraciones del telescopio, que se quieren medir, se mezclarían con las deformaciones provocadas por la atmósfera. Este problema se soluciona tomando imágenes de largo tiempo de exposición, comparado con el tiempo característico de fluctuaciones de fase<sup>12</sup>. En ese caso, las deformaciones del frente de onda se promedian, pero un valle y un pico de igual magnitud, producidos por la atmósfera, tienen la misma probabilidad de ocurrir, es decir que la densidad de probabilidad es par, por lo tanto el promedio de las deformaciones producidas por la atmósfera es cero<sup>9</sup>. Entonces se medirían sólo las deformaciones del frente de onda producidas por las aberraciones del telescopio.

Conociendo estas aberraciones, un sistema de actuadores sobre el espejo primario y/o secundario, podría compensarlas. Este es el principio de la óptica activa. Suponiendo que la corrección fuera perfecta, y si  $D$  es mayor que  $r_0$ , entonces la resolución del sistema óptico estaría limitada por  $r_0$ , de acuerdo a la ecuación 50. Si la corrección no fuera

perfecta, pero el tamaño de la imagen de un objeto puntual, en ausencia de atmósfera, fuera todavía menor que el que impone la atmósfera ( $1.27 \frac{\lambda}{r_0}$ ), entonces la corrección de la óptica del telescopio sería perfectamente aceptable. En realidad no se distinguiría de una corrección perfecta.

Se puede ver un ejemplo de esto considerando que se quieren medir y compensar las aberraciones del telescopio de 2.1 metros de San Pedro Mártir. Se supone que se toman las imágenes de largo tiempo de exposición, a una longitud de onda igual a  $0.7 \mu\text{m}$ . Basandose en los valores de  $r_0$  publicados por Roddier<sup>9</sup>, para distintos grados de turbulencia, se puede considerar que, a  $0.5 \mu\text{m}$  de longitud de onda, el parámetro de Fried  $r_0$  es igual a 10 cm. Usando la relación 52, se calcula que a una longitud de onda de  $0.7 \mu\text{m}$ ,  $r_0$  es igual a 15 cm. La compensación de las deformaciones del frente de onda producidas por las aberraciones del telescopio, debe ser tal que la imagen de una fuente puntual, a través del telescopio corregido, y sin atmósfera, tenga un diámetro angular  $\alpha_c$  acotado por:

$$\alpha_c < 1.27 \frac{\lambda}{r_0} = 5.9 \cdot 10^{-6} \text{ rad} \quad (53)$$

$$\text{i.e. } \alpha_c < 1.2 \text{ segundos de arco.} \quad (54)$$

El tamaño de una imagen limitada por difracción en este telescopio es:

$$\alpha_{lim} = 1.27 \frac{\lambda}{D} = 0.08 \text{ segundos de arco.} \quad (55)$$

Se observa que

$$\alpha_{lim} \ll \alpha_c, \quad (56)$$

por lo que uno puede esperar que la corrección de las aberraciones del telescopio requerida para ser considerada como perfecta, sea realizable.

# CAPITULO VI

## Conclusión

Las pruebas del algoritmo propuesto en este trabajo, revelaron que la parte del algoritmo que ajusta las condiciones de frontera debe ser refinada, y que la manera de estimar el error debe ser revisada. A continuación se discuten estos puntos. En el caso de la aberración de coma, se ve que el frente de onda reconstruido, contiene una componente de tilt adicional a la del frente de onda teórico. Esto sólo puede provenir del ajuste de las condiciones de frontera, ya que el laplaciano del tilt es cero, por lo cual es transparente ante el ajuste del laplaciano. Este tilt no está presente en las aberraciones de esfericidad ni de astigmatismo, pero reaparece en el caso de la combinación de los primeros 19 polinomios de Zernike. Uno de estos polinomios corresponde a la aberración de coma. Uno podría pensar que el tilt adicional proviene, entonces, del ajuste de la aberración de coma, lo cual no es descabellado porque la expresión analítica de la coma de tercer orden, tiene un término que corresponde a un tilt. Sin embargo, tilt no es la única aberración residual, sobre todo en el caso de la combinación de los polinomios de Zernike. Por lo que se podría pensar, que

ciertas geometrías de las deformaciones del frente de onda, no se ajustan correctamente. La prueba realizada con la aberración de astigmatismo, muestra que en ciertos casos el ajuste de las condiciones de frontera es perfecto, ya que para esta aberración solamente se cuenta con los datos de la derivada radial en la frontera, porque el laplaciano es cero.

Por otro lado, se observa que el error rms porcentual en el caso de la aberración de esfericidad, es de 4% al cabo de la décima iteración, aunque no se aprecien diferencias significativas entre los frentes de onda calculado, y el teórico. Esto es una evidencia de que este criterio de error no es muy eficaz. Se podría pensar que este valor del error se debe únicamente a la contribución de los puntos que se encuentran muy cerca de la frontera, ya que es donde la diferencia de los frentes de onda teórico y calculado, es mayor. El problema de calcular un error rms, es que se pierde la información de la localización de los errores. Se pensó en calcular el error respecto al laplaciano porque, en el caso de sólo tener datos provenientes de la diferencia de las imágenes desenfocadas (ver ecuaciones 6 a 9), el laplaciano es el único dato que se tiene en todos los puntos de la pupila. Para las pruebas del algoritmo, el mejor criterio de error es la diferencia entre el frente de onda calculado y el teórico.

Se pretende medir las aberraciones del telescopio de 2.1 m de San Pedro Mártir. Para esta tarea, este algoritmo no reconstruye suficientemente bien el frente de onda. Una posible manera de refinarlo es usar el resultado que arroja, como entrada a otro programa basado en métodos de relajación. Además se deberá pensar en como cambiar la geometría de la pupila en el algoritmo, ya que en el caso de un telescopio, la pupila es una "dona", a causa de la sombra del espejo secundario. También se tendrá el problema de que las dimensiones de la "dona" para cada imagen desenfocada pueden no ser iguales, en cuyo caso se deben aplicar ciertas transformaciones, para que las imágenes coincidan, que posiblemente alteren las mediciones de las aberraciones. Entonces, se deberá conocer la relación entre las transformaciones de las imágenes, y la consecuente transformación de las aberraciones.

Una vez habiendo dominado esto, este algoritmo puede ser usado en un sistema de

óptica activa, como parte del sensor de frente de onda. Para esto sería necesario optimizar el programa, de manera a que su ejecución sea más rápida; lo cual no debe ser muy difícil, ya que el programa se desarrolló sin pensar en su rapidez de ejecución.

El método de medir frentes de onda a partir de imágenes desenfocadas, resulta muy atractivo en particular porque no requiere de una instrumentación importante. Basta un detector bidimensional, y una computadora. Esta es una de las razones por las cuales ha suscitado tanto interés no sólo en México, sino en el mundo entero. De ser refinado e instrumentado para telescopios, el algoritmo que se presentó aquí, podría publicarse, y ser aplicado en algunos telescopios en el extranjero.

# APENDICE A

En este apéndice, se demuestra la ecuación 28, siguiendo el tratamiento que aparece en la referencia 8.

La transformada de Fourier inversa bidimensional  $\phi_{ij}$ , de una función discreta  $\hat{\phi}_{mn}$  se escribe:

$$\phi_{ij} = \frac{1}{nr.nc} \sum_{m=0}^{nr-1} \sum_{n=0}^{nc-1} \hat{\phi}_{mn} e^{2\pi ilm/nr} e^{2\pi ijn/nc}. \quad (A.1)$$

Si se sustituye cada miembro de la ecuación 27 por su equivalente dado por la ecuación A.1, se obtiene:

$$\begin{aligned} & \frac{1}{nr.nc} \sum_{m=0}^{nr-1} \sum_{n=0}^{nc-1} \hat{\phi}_{mn} (e^{2\pi ilm/nr} e^{2\pi i(j+1)n/nc} + e^{2\pi ilm/nr} e^{2\pi i(j-1)n/nc} \\ & + e^{2\pi i(l-1)m/nr} e^{2\pi ijn/nc} + e^{2\pi i(l+1)m/nr} e^{2\pi ijn/nc} - 4e^{2\pi ilm/nr} e^{2\pi ijn/nc}) \\ & = \frac{\Delta^2}{nr.nc} \sum_{m=0}^{nr-1} \sum_{n=0}^{nc-1} f_{mn} e^{2\pi ilm/nr} e^{2\pi ijn/nc}. \end{aligned} \quad (A.2)$$

Por la ortogonalidad de las funciones trigonométricas del tipo  $e^{2\pi ix}$ , la igualdad en la ecuación anterior se cumple para cada término  $mn$  de las sumas. Por esto, se pueden eliminar las sumatorias, y si además se divide la ecuación A.2 entre  $e^{2\pi ilm/nr} e^{2\pi ijn/nc}$ , se llega a:

$$\hat{\phi}_{mn} (e^{2\pi in/nc} + e^{-2\pi in/nc} + e^{-2\pi im/nr} + e^{2\pi im/nr} - 4) = \Delta^2 f_{mn}, \quad (A.3)$$

o bien,

$$2\hat{\phi}_{mn} \left[ \cos\left(\frac{2\pi}{nc}n\right) + \cos\left(\frac{2\pi}{nr}m\right) - 2 \right] = \Delta^2 f_{mn}. \quad (A.4)$$

Despejando, se obtiene finalmente, la ecuación 28:

$$\hat{\phi}_{mn} = \frac{\Delta^2 f_{mn}}{2 \left[ \cos\left(\frac{2\pi}{nc}n\right) + \cos\left(\frac{2\pi}{nr}m\right) - 2 \right]}$$

## APENDICE B

El laplaciano en coordenadas polares, de la función  $\xi(r, \theta)$ , se escribe:

$$\nabla^2 \xi(r, \theta) = \frac{\partial^2 \xi}{\partial r^2} + \frac{1}{r} \frac{\partial \xi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \xi}{\partial \theta^2}. \quad (B.1)$$

La función  $\xi$  de la forma de la ecuación 31:

$$\xi(r, \theta) = \sum_{k,l,p}^{\infty} r^k (a_l \cos(l\theta) + b_p \sin(p\theta)). \quad (31)$$

Se tiene:

$$\begin{aligned} \frac{\partial \xi}{\partial r} &= \sum_{k,l,p}^{\infty} k r^{k-1} (a_l \cos(l\theta) + b_p \sin(p\theta)), \\ \frac{\partial^2 \xi}{\partial r^2} &= \sum_{k,l,p}^{\infty} k(k-1) r^{k-2} (a_l \cos(l\theta) + b_p \sin(p\theta)), \\ \frac{\partial^2 \xi}{\partial \theta^2} &= \sum_{k,l,p}^{\infty} r^k (-a_l l^2 \cos(l\theta) - b_p p^2 \sin(p\theta)). \end{aligned} \quad (B.2)$$

Sustituyendo las ecuaciones B.2 en la ecuación B.1, y factorizando el seno y el coseno, se obtiene:

$$\begin{aligned} \nabla^2 \xi &= \sum_{k,l,p}^{\infty} r^{k-2} [(k(k-1) + k - l^2) a_l \cos(l\theta) + ((k(k-1) + k - p^2) b_p \sin(p\theta))] \\ &= \sum_{k,l,p}^{\infty} r^{k-2} [(k^2 - l^2) a_l \cos(l\theta) + (k^2 - p^2) b_p \sin(p\theta)]. \end{aligned} \quad (B.3)$$

Una de las condiciones de  $\xi$  es que su laplaciano sea cero, para toda  $r$  y  $\theta$ . De la ecuación B.3 se observa que esto se cumple si y solo si:

$$k^2 - l^2 = 0 \quad \text{y} \quad k^2 - p^2 = 0$$

Es decir que  $k = l = p$ , por lo tanto se llega al resultado siguiente:

Si  $\xi$  es de la forma de la ecuación 31 y su laplaciano es cero, entonces  $\xi$  es de la forma:

$$\xi(r, \theta) = \sum_p^{\infty} r^p (a_p \cos(p\theta) + b_p \sin(p\theta)).$$

Lo cual corresponde a la ecuación 32.

## PROGRAM concha

```

C-----
C  Calcula el frente de onda a partir de su laplaciano, y de su derivada
C  radial evaluada en la frontera de la pupila. La pupila es circular.
C-----
C  nr es el numero de renglones del arreglo, y nc el numero de columnas

PARAMETER(nr=256,nc=256,dnr=2*nr)

C Define las variables

CHARACTER IMAGE1*12 , IMAGE2*12
REAL dmuestr,dmuestrc,f,D,l
REAL datos11(nr,nc), datosi2(nr,nc)
REAL datosm12(nr,nc), ssign(nr,nc), cssign(dnr,nc)
REAL tfssign(dnr,nc), u(dnr), v(nr), tfphi(dnr,nc)
REAL phi(dnr,nc), cphidx(dnr,nc), cphidy(dnr,nc), dphidr
REAL lepphi(dnr,nc), lepphi1(dnr,nc), pl,cc,dmuest,lepdib(nc)
REAL den(dnr,nc), coc(dnr,nc), phidib(nc), coedib(nc)
REAL suma,erms,sumacs,x1(dnr,nc),g(dnr,nc),X(dnr,nc)
REAL r,dif,cfssign(nr,nc),ccfssign(dnr,nc),erms(1000)
INTEGER NN(2),RADIO,x,y,numit
pi=acos(-1.)

C pregunta el numero de iteraciones

print*, 'cuantas iteraciones?'
read(*,*) ni

C llama la subrutina que define parametros

call param(image1,image2,dmuestr,dmuestrc,f,D,l,radio)

C Lee los datos S que estan en un archivo IMAGE1.DAT y se guardan en
C la variable datos11.

call leedatos(IMAGE1,datos11)

C En la variable ssign se guardan los datos del laplaciano (dentro de la pupila),
C y en la variable cfssign se guardan los datos de la derivada radial (en la frontera
C de la pupila)

do j=1,nc
do i=1,nr
x=j-nc/2
y=-(i-nr/2)
r=sqrt(float(x**2+y**2))
dif=(r-radio)
if (dif .lt. 0.0) then
ssign(i,j)=datos11(i,j)
else
ssign(i,j)=0.0
cfssign(i,j)=datosi1(i,j)
end if
enddo
enddo
enddo

```

C Crea una matriz CSSIGN, compleja con parte imaginaria=0 y parte real=ssign  
 C CSSIGN es un arreglo de dnr=2\*nr renglones y nc columnas.

call matcom(ssign,cssign)

C Calcula la T.F. de cssign

```
10 numit=numit+1
    NN(1)=nr
    NN(2)=nc
    NDIM=2
    ISIGN=-1
```

call fourn(cssign,NN,NDIM,ISIGN)

C Guarda la T.F. en tfssign

```
do j=1,nc
  do i=1,dnr
    tfssign(i,j)=cssign(i,j)
  enddo
enddo
```

C Pone 0 en el origen de coordenadas de frecuencia

```
tfssign(1,1)=0
tfssign(2,1)=0
```

C Calcula las coordenadas del espacio de frecuencias,  
 C correspondientes a los índices de la matriz tfssign.

C Ordenadas (i.e renglones)--> u(i), hay dnr,

C pero solo nr distintas.

C Abscisas (i.e columnas)--> v(j), hay nc

C

C U-->REGLONES V-->COLUMNAS

call coordfrec(u,v,dmuest,dmuestc)

C Divide tfssign entre "los cosenos" y lo guarda en tfphi

C Para este calculo suponemos dmuest=dmuestc=dmuest

```
dmuest=dmuestc
cc=2.*pi*dmuest
do j=1,nc
  do i=1,dnr
    if (j .EQ. 1 .AND. ((i .EQ. 1) .OR. (i .eq. 2))) then
      tfphi(i,j)=tfssign(i,j)
    else
      den(i,j)=2.*(cos(cc*u(i))+cos(cc*v(j)))-2.
      tfphi(i,j)=dmuest**2.*tfssign(i,j)/den(i,j)
    end if
  enddo
enddo
```

C calcula la T.F. inversa de tfphi, y la guarda en phi

```
NN(1)=nr
NN(2)=nc
NDIM=2
ISIGN=1
```

call fourn(tfphi,NN,NDIM,ISIGN)

```
do j=1,nc
  do i=1,dnr
    phi(i,j)=(1./(nr*nc))*tfphi(i,j)
```

```

        enddo
        enddo
C   Calcula el laplaciano de phi, lo guarda en lapphi
        call lap(phi, lapphi, dmuestr, dmuestr)
C   Crea de nuevo una matriz compleja con parte imaginaria=0, y parte real
C   igual a los datos originales
        call matcom(ssign, cssign)
C   Calcula el error cuadrático promedio del laplaciano de phi (lapphi),
C   respecto a los datos originales (cssign), y lo guarda en el vector
C   errms, que es función del número de iteración en proceso numit.
        nu=0
        do j=1,nc
            do i=1,dnr-1,2
                r2=((i+1)/2-dnr/2)**2+(j-nc/2)**2
                if (r2 .lt. radio**2) then
                    nu=nu+1
                    sumacs=sumacs+cssign(i, j)**2
                    coc(i, j)=lapphi(i, j)-cssign(i, j)
                    suma=suma+coc(i, j)**2
                end if
            enddo
        enddo
        errms(numit)=sqrt(suma/float(nu))/sqrt(sumacs/float(nu))*100
C   Cuando el número de iteraciones es igual a ni, crea una matriz compleja
C   a partir de los datos originales en la frontera, y calcula el gradiente
C   de phi
        if (numit .eq. ni) then
            call matcom(cfssign, ccfssign)
            call ngrad(phi, dphidx, dphidy, dmuestr, dmuestr)
C   Construye la función g
        do j=1,nc
            do i=1,dnr-1,2
                x=(j-nc/2)
                y=(i+1)/2-dnr/2
                r=sqrt(float(x**2+y**2))
                dif=(r-radio)
                if ((dif .ge. 0.0) .and. (dif .lt. 1.0)) then
                    dphidr=dphidx(i, j)*(float(x))/r+dphidy(i, j)*(float(y))/r
                    g(i, j)=ccfssign(i, j)-lapphi(i, j)-dphidr
                end if
            enddo
        enddo
C   A partir de g, la subrutina INTEGRA calcula xi, calculando
C   los coeficientes en y bn a partir de integrales del tipo
C   int(g*cos(n*teta)), para an y analogamente para bn con sen
        call integra(g, radio, xi)
C   Calcula la solución final phi=phi+xi
        do j=1,nc
            do i=1,dnr-1,2
                W(i, j)=phi(i, j)+xi(i, j)
            enddo
        enddo

```

```
c La subrutina arregla le resta a W su promedio, y crea todos los
c archivos de salida
```

```
    call arregla(W, radio, errms, numit)
    goto 20
end if

call remplaza(csign, lapphi, lapphif, radio)
goto 10
20  END
```

---

C

```
C Subrutina de lectura de los archivos de datos
C Lee el archivo cuyo nombre se guarda en la variable
C que es argumento de la subrutina.
C Los datos se guardan en la variable DATOS(nr,nc)
```

```

SUBROUTINE LEEDATOS( NOMBRE, VARIABLE )
PARAMETER(nr=256,nc=256)
CHARACTER nombre*12
REAL variable(nr,nc)

```

```

    open(UNIT=1,FILE=NOMBRE,ERR=10)
    read(1,*,ERR=20) variable
    return
    goto 50
10  write(*,*)'Error abriendo ',NOMBRE
20  write(*,*)'Error leyendo ',NOMBRE
50  end

```

C-----

C-----

C Subrutina que invierte la imagen 2 y la guarda en datosi2-

```

SUBROUTINE INVIERTE(entrada, salida)
PARAMETER(nr=256,nc=256)
REAL entrada(nr,nc),salida(nr,nc)

```

```

do j=1,nc
do i=1,nr
    salida(i,j)=entrada(nr-i+1,nc-j+1)
enddo
enddo
return
end

```

C-----

C-----

C Subrutina que crea una matriz SALIDA compleja con parte Im=0  
C y Re=ENTRADA

```

SUBROUTINE MATCOM( entrada, salida)
PARAMETER(nr=256,nc=256,dnr=2*nr)
REAL entrada(nr,nc),salida(dnr,nc)

```

```

do j=1,nc
do i=1,nr
    salida(2*i-1,j)=entrada(i,j)
    salida(2*i,j)=0
enddo
enddo
return
end

```

C-----

C-----

C Subrutina que calcula las coordenadas del espacio de frecuencias  
C U--->REGLONES (ordenadas) V--->COLUMNAS (abscisas)

```

SUBROUTINE COORDFREC( uren, vcol, espy, espX)
PARAMETER(nr=256,nc=256,dnr=2*nr)

```

REAL uren(dnr),vcol(nc), esp,espy

C Ordenadas de frecuencias positivas o sea subíndice<dnr/2

```
do i=1,dnr-1,2
  uren(i)=(i-1)/2/(nr*espy)
  uren(i+1)=uren(i)
enddo
```

C Recordemos que los datos de índice i=dnr/2 y/o j=nc/2 son aliasing points, que no nos interesan, pero para hacer los C calculos continuos, vamos a escoger que estos puntos tengan C como abscisa 1/2dmuestr, y de ordenada 1/2dmuestr.

C Ordenadas de frec. negativas o sea de subíndice>dnr/2

```
do i=dnr/2+3,dnr-1,2
  uren(i)=-uren(dnr+2-i)
  uren(i+1)=uren(i)
enddo
```

C Abscisas de frecuencias positivas

```
do i=1,nc
  vcol(i)=(i-1)/(nc*espx)
enddo
```

C Abscisas de frecuencias negativas

```
do i=nc/2+2,nc
  vcol(i)=-vcol(nc+2-i)
enddo
return
end
```

C-----

C-----

C Subrutina que le anexa a CSSIGN los datos de LAPPHI,  
C fuera de la pupila.

```
SUBROUTINE REMPLAZA(CSSIGN, LAPPHI, LAPPHIF, RADIO)
PARAMETER(nr=256,nc=256,dnr=2*nr)
REAL CSSIGN(DNR,NC), LAPPHI(DNR,NC), LAPPHIF(DNR,NC)
REAL x,y,r,dif
INTEGER RADIO
```

```
do j=1,nc
do i=1,dnr-1,2
  x=(j-nc/2.)
  y=-((i+1)/2.-nr/2.)
  r=sqrt(x**2+y**2)
  dif=(r-radio)
  if (dif .ge. 0) then
    cssign(i,j)=lapphi(i,j)
    cssign(i+1,j)=lapphi(i+1,j)
  endif
enddo
enddo
```

```

return
end

```

---

C

c Subrutina que sustrae a phi su promedio calculado dentro de la pupila, y crea todos los archivos de salida.

```

SUBROUTINE ARREGLA(PHI, RADIO, ERRMS, numit)
PARAMETER(nr=256, nc=256, dnr=2*nr)
REAL PHI (DNR, NC), RPHI (NR, NC), suma, ERRMS (1000)
REAL datos2(nr, nc), residuo(nr, nc)
INTEGER RADIO, numit

do j=1, nc
do i=1, dnr-1, 2
r2=((i+1)/2-nr/2)**2+(j-nc/2)**2
if (r2 .le. radio**2) then
ni=i+1
suma=suma+phi(i, j)
end if
enddo

open(UNIT=4, FILE='phior.dat')
read(4, *) datos2
close(4)
do j=1, nc
do i=1, dnr-1, 2
r2=((i+1)/2-nr/2)**2+(j-nc/2)**2
rphi((i+1)/2, j)=phi(i, j)-suma/float(ni)
residuo((i+1)/2, j)=rphi((i+1)/2, j)-datos2((i+1)/2, j)
enddo
enddo
open(UNIT=3, FILE='smphi.dat', FORM='unformatted', ERR=10)
open(UNIT=1, FILE='smplo.dat', ERR=20)
open(UNIT=2, FILE='smerror.dat', ERR=25)
open(UNIT=4, FILE='smresplot.dat', ERR=30)
open(UNIT=8, FILE='smresiduo.dat', FORM='unformatted', ERR=40)

write(3, ERR=10) rphi
write(8, ERR=40) residuo
do j=1, nc
if ((j .ge. 48) .and. (j .le. 207)) then
write(1, *, ERR=20) j, rphi(nr/2, j)
write(4, *, ERR=30) j, (rphi(nr/2, j)-datos2(nr/2, j))
else
write(1, *, ERR=20) j, 0.0
write(4, *, ERR=30) j, 0.0
end if
enddo
do i=1, numit
write(2, *, ERR=25) i, errms(i)
enddo
close(1)
close(2)
close(3)
goto 50

10 print*, 'error en smphi.dat'
20 print*, 'error en smplo.dat'
25 print*, 'error en smerror.dat'
30 print*, 'error en smresplot.dat'

```

```
40     print*, 'error en smresiduo.dat'  
50     end
```

C Subrutina que calcula la funcion xi, calculando los coeficientes  
 C an y bn, a partir de la funcion g.

```

SUBROUTINE INTEGRA(G,RADIO,PHI2)
PARAMETER(NR=256,NC=256,DNR=2*NR,NCIRC=1000,ORDMAX=5)
REAL G(DNR,NC),A(ORDMAX),B(ORDMAX),rphi2(nr,nc)
REAL r,dif,teta,circ(NCIRC),pi,dteta,phi2(dnr,nc)
INTEGER radio,np,x,y,ndatos

```

```
pi=acos(-1.0)
```

C Crea una funcion CIRC que contiene los elementos de la  
 C frontera (es decir que RADIO<=r<RADIO+1), ordenados en  
 C orden creciente del angulo TETA, en coordenadas polares

```
call circulo(g,circ,radio,ndatos)
dteta=2*pi/ndatos
```

C Calcula, a partir de CIRC, los coeficientes an y bn, por  
 C medio de un integracion de de funciones del tipo  
 C circ(teta)\*cos(n\*teta)

```
call coefs(circ,a,b,dteta,radio,ndatos)
```

C Calcula finalmente la funcion xi (aqui se llama phidos)

```
call phidos(a,b,phi2)
return
end
```

---

```

SUBROUTINE COEFS(FUN,A,B,DTETA,RADIO,ii)
PARAMETER(NCIRC=1000,ORDMAX=5)
REAL FUN(NCIRC),A(ORDMAX),B(ORDMAX),DTETA,sumacos,sumasin
REAL concos1,concos2,consin1,consin2,pi
INTEGER ORDEN,RADIO,ii
pi=acos(-1.0)

```

```
do orden=1,ordmax
do l=1,ii-1
concos1=fun(l)*cos(orden*(l-1)*dteta)
concos2=fun(l+1)*cos(orden*(l)*dteta)
sumacos=sumacos+dteta*(concos1+concos2)/2.0
consin1=fun(l)*sin(orden*(l-1)*dteta)
consin2=fun(l+1)*sin(orden*(l)*dteta)
sumasin=sumasin+dteta*(consin1+consin2)/2.0

```

```
enddo
a(orden)=sumacos*(1.0/(pi*orden*RADIO**(orden-1)))
b(orden)=sumasin*(1.0/(pi*orden*RADIO**(orden-1)))
sumacos=0.0
sumasin=0.0
enddo
return
end
```

---

```

SUBROUTINE CIRCULO(G,CIRC,RADIO,ii)
PARAMETER(NR=256,NC=256,DNR=2*NR)
REAL G(DNR,NC),CIRC(1000)
REAL dif,r,teta,pi
INTEGER RADIO,x,y,cont,ii

```

```

pi=acos(-1.0)
x=radio
y=0
ii=1
dif=0.0

do while (x .gt. 0)
do while ((dif .ge. 0.0) .and. (dif .lt. 1.0))
teta=atan(float(y)/float(x))
i=2*(-y+nr/2)-1
j=x+nc/2
circ(ii)=g(i,j)*cos(0.0)
ii=ii+1
y=y+1
r=sqrt(float(x**2+y**2))
dif=(r-radio)
cont=0
enddo
cont=cont+1
if (cont .ge. 2) then
y=-1
goto 50
end if
x=x-1
50 r=sqrt(float(x**2+y**2))
dif=(r-radio)
enddo

x=0
y=-radio+1
circ(ii)=g(2*(-y+nr/2)-1,x+nc/2)*cos(0.0)
ii=ii+1
x=-1
y=radio
dif=0.0

do while (y .gt. 0.0)
do while ((dif .ge. 0.0) .and. (dif .lt. 1.0))
teta=atan(float(y)/float(x)) +pi
i=2*(-y+nr/2)-1
j=x+nc/2
circ(ii)=g(i,j)*cos(0.0)
ii=ii+1
x=x-1
r=sqrt(float(x**2+y**2))
dif=(r-radio)
cont=0
enddo
cont=cont+1
if (cont .ge. 2) then
x=x+1
goto 60
end if
y=y-1
60 r=sqrt(float(x**2+y**2))
dif=(r-radio)
enddo

x=-radio-1
y=0
dif=0.0

```

```

do while (x .lt. 0.0)
do while ((dif .ge. 0.0) .and. (dif .lt. 1.0))
teta=atan(float(y)/float(x)) *pi
i=2*(-y+nr/2)-1
j=x+nc/2
circ(ii)=g(i,j)*cos(0.0)
ii=ii+1
y=y-1
r=sqrt(float(x**2+y**2))
dif=(r-radio)
cont=0
enddo
cont=cont+1
if (cont .ge. 2) then
y=y+1
goto 70
end if

70  x=x+1
r=sqrt(float(x**2+y**2))
dif=(r-radio)

enddo

x=0
y=-radio-1
circ(ii)=g(2*(-y+nr/2)-1,x+nc/2)*cos(0.0)
ii=ii+1
x=1
y=-radio-1
dif=0.0

do while (y .lt. 0.0)
do while ((dif .ge. 0.0) .and. (dif .lt. 1.0))
teta=2*pi-atan(float(y)/float(x))
i=2*(-y+nr/2)-1
j=x+nc/2
circ(ii)=g(i,j)*cos(0.0)
ii=ii+1
x=x+1
r=sqrt(float(x**2+y**2))
dif=(r-radio)
cont=0
enddo
cont=cont+1
if (cont .ge. 2) then
x=x-1
goto 80
end if
y=y+1
80  r=sqrt(float(x**2+y**2))
dif=(r-radio)
enddo

return
end

```

```

SUBROUTINE PHIDOS(A,B,PHI2)
PARAMETER (NR=256,NC=256,DNR=2*NR,ORDMAX=5)
REAL A(ORDMAX),B(ORDMAX),PHI2(DNR,NC)
REAL r,dif,teta,phideux

```

```

INTEGER x,y,radio
pi=acos(-1.0)
do j=1,nc
do i=1,dnr-1,2
x=(j-nc/2)
y=-((i+1)/2-nr/2)
r=sqrt(float(x**2+y**2))
dif=(r-radio)
if ((x .eq. 0) .and. (y .gt.0)) then
teta=pi/2.
end if
if ((x .eq. 0) .and. (y .lt. 0)) then
teta=3.*pi/2
end if
if (x .lt. 0) then
teta=atan(float(y)/float(x))+pi
end if
if ((x .gt. 0) .and. (y .lt. 0)) then
teta=atan(float(y)/float(x))+2.*pi
end if
if ((x .eq. 0) .and. (y .eq. 0)) then
teta=0.0
end if
if ((x .gt. 0) .and. (y .ge. 0)) then
teta=atan(float(y)/float(x))
end if

do n=1,ordmax
phideux=phideux+(r**n) * (a(n)*cos(n*teta)+b(n)*sin(n*teta))
enddo
phi2(i,j)=phideux
phideux=0.0
enddo
enddo
return
end

```

C Subrutina que calcula el laplaciano de una matriz de 2 variables  
C x ( columnas ) y y ( renglones).

```
SUBROUTINE LAP (FUN, LAFUN, ESPX, ESPY)  
PARAMETER (NR=256, NC=256, DNR=2*NR)  
REAL FUN (DNR, NC), LAFUN (DNR, NC), AUX1 (DNR, NC), AUX2 (DNR, NC)  
REAL AUX3 (DNR, NC), ESPX, ESPY
```

```
do i=3, dnr-2  
do j=2, nc-1  
aux1 (i, j) = (fun (i, j+1) + fun (i, j-1)) / (espx**2.)  
aux2 (i, j) = (fun (i+2, j) + fun (i-2, j)) / (espy**2.)  
aux3 (i, j) = (-2. / (espx**2.) + 2. / (espy**2.)) * fun (i, j)  
lafun (i, j) = aux1 (i, j) + aux2 (i, j) + aux3 (i, j)  
enddo  
return  
end
```

C Subrutina que calcula derivadas parciales en x ( columnas )  
C y en y (renglones)

```
SUBROUTINE NGRAD (FUN, DFUNDX, DFUNDY, ESPX, ESPY)  
PARAMETER (NR=256, NC=256, DNR=2*NR)  
REAL FUN (DNR, NC), DFUNDX (DNR, NC), DFUNDY (DNR, NC)  
REAL ESPX, ESPY
```

```
do i=1, nr  
do j=1, nc/2  
dfundx (i, j) = (fun (i, j+1) - fun (i, j)) / esp  
dfundy (i, j) = (fun (i+2, j) - fun (i, j)) / espy  
enddo  
do j=nc/2+1, nc  
dfundx (i, j) = (fun (i, j) - fun (i, j-1)) / esp  
dfundy (i, j) = (fun (i+2, j) - fun (i, j)) / espy  
enddo  
enddo  
do i=nr+1, dnr  
do j=1, nc/2  
dfundx (i, j) = (fun (i, j+1) - fun (i, j)) / esp  
dfundy (i, j) = (fun (i, j) - fun (i-2, j)) / espy  
enddo  
do j=nc/2+1, nc  
dfundx (i, j) = (fun (i, j) - fun (i, j-1)) / esp  
dfundy (i, j) = (fun (i, j) - fun (i-2, j)) / espy  
enddo  
enddo  
return  
end
```

C Archivo que guarda parametros

```

SUBROUTINE PARAM(pimage1,pimage2,pdmuestr
,pdmuestrc,pf,pD,pl,pradio)

```

```

CHARACTER pIMAGE1*12, pIMAGE2*12
REAL pdmuestr,pdmuestrc
REAL pf,pD,pl
INTEGER pradio

```

C Nombres de los archivos donde se guardan las imagenes

```

pIMAGE1='image1.dat'
pIMAGE2='image2.dat'

```

C Distancia (m) de muestreo de las imagenes

C Para las pruebas, esta distancia se escoge de 1 pixel

```

pdmuestr=1.
pdmuestrc=1.

```

C Parametros del instrumento (todo en m)

C Esto no interviene en las pruebas

```

pf=1.
pD=.5

```

C Distancia a la que se toman las imagenes, a partir del foco (m)

C Esto tampoco interviene en las pruebas.

```

pl=.1

```

C Radio, en num. de pixeles, de las imagenes

```

pradio=80

```

```

return
end

```

## PROGRAM COMA

C-----

C Programa que crea un archivo que guarda el laplaciano de un  
 C frente de onda con aberración de coma, dentro de una pupila  
 C circular de diámetro igual a 80 pixeles, y su derivada radial  
 C sobre la frontera de la pupila. Se muestran, en comentario, los  
 C cambios que se deben hacer para considerar otras aberraciones.

C-----

```

parameter (nr=256,nc=256)
REAL DATOS(nr,nc),DATOS1(nr,nc),dpi,r,costeta
REAL phiorplot(nc),dif
INTEGER radio
radio=80
dpi=2.*acos(-1.)
DO J=1,nc
  DO I=1,nr
    r=sqrt(float((i-nr/2)**2+(j-nc/2)**2))
    dif=(r-radio)
    costeta=float((j-nc/2))/r
    if (dif.lt.1.0) then
      DATOS(I,J)=(24./radio**3)*float((j-nc/2))
C Para esfericidad: DATOS(I,J)=(56.0/radio**4)*r**2-24.0/radio**2
C Para astigmatismo: DATOS(I,J)=0.0
      end if
      if ((dif.lt.1.0).and.(dif.ge.0.0))
        & then
          DATOS(I,J)=DATOS(I,J)+(7.0/float(radio))*costeta
C Para esfericidad: datos(i,j)=datos(i,j)+12.0/radio
C Para astigmatismo:
C datos(i,j)=datos(i,j)+(2.0/float(radio))*(2.*costeta**2-1.)
      end if
    ENDDO
  ENDDO
  OPEN(UNIT=1,FILE='imgel.dat',ERR=3)
  WRITE(1,*,ERR=2) DATOS
  close(1)

GO TO 5

2 WRITE(*,*)'Error escribiendo DATOS '
3 WRITE(*,*)'Error abriendo DATOS '
5 END

```

ESTA TESIS NO DEBE  
 SALIR DE LA BIBLIOTECA

```
PROGRAM comaor
```

```
-----
C Programa que crea un archivo donde se guarda un frente de onda con
C aberracion de coma, cuyo promedio es cero, dentro de una pupila
C circular de diametro 80. Se muestran, en comentario, las modificaciones
C necesarias para crear otras aberraciones.
-----
```

```
parameter(nr=256,nc=256)
REAL DATOS(nr,nc),datos2(nr,nc),dpi,r,suma
REAL phidibo(nc),n,phidib(nc),phico(nc),cos
INTEGER radio
radio=80
dpi=2.*accs(-1.)
DO I=1,nr
  DO J=1,nc
    r=sqrt((i-nr/2.)**2+(j-nc/2.)**2)
    cos=(j-nc/2.)/r
    if (r .le. radio) then
      n=n+1
      DATOS(I,J)=(3.0*r**2/float(radio)**3-2.0/radio)
      * (j-nc/2.)
C Para esfericidad: DATOS(I,J)=(6.0/radio**4)*r**4-(6.0/radio**2)*r**2+1
C Para astigmatismo: DATOS(I,J)=(r**2/radio**2)*(2.0*((j-nc/2.)/r)**2 -1.0)
      suma=suma+DATOS(I,J)
    end if
  ENDDO
ENDDO
```

```
C Resta el promedio
```

```
do i=1,nr
  do j=1,nc
    r2=(i-nr/2)**2+(j-nc/2)**2
    if (r2 .le. radio**2) then
      datos2(i,j)=DATOS(i,j)-(suma/n)
    end if
  enddo
enddo

do j=1,nc
  phidibo(j)=datos2(nr/2-1,j)
enddo

open(UNIT=2,FILE='smphiorplot.dat')
do i=1,nc
  write(2,*) i,phidibo(i)
enddo
close(2)

OPEN(UNIT=1,FILE='smphior.dat',FORM='UNFORMATTED',ERR=3)
WRITE(1,ERR=2) DATOS2
close(1)

OPEN(UNIT=3,FILE='phior.dat')
WRITE(3,*) DATOS2
close(3)
```

```
GO TO 5
2  WRITE(*,*)'Error escribiendo DATOS2'
3  WRITE(*,*)'Error abriendo smphior.dat'
5  END
```

# Referencias

1. Shack, R.V., and Platt, B.C., *J. Opt. Soc. Am.* **61**, p 1200 (1974).
2. F. Roddier, "Curvature sensing and compensation: a new concept in adaptive optics", *Appl. Opt.* **27**, 1223-1225 (1988).
3. F. Roddier, "Wavefront sensing and the irradiance transport equation", *Appl. Opt.* **29**, 1402-1403 (1990).
4. C. Roddier, F. Roddier, "Wavefront reconstruction from defocused images and the testing of ground-based optical telescopes", preprint submitted to *J. Opt. Soc. Am. A* (1992).
5. K. Ichikawa, A. Lohmann, M. Takeda, "Phase retrieval based on the irradiance transport equation and the Fourier transform method: experiments", *Appl. Opt.* **27**, 3433-3436 (1988).
6. F. Roddier, C. Roddier, "Wavefront reconstruction using iterative Fourier transforms", *Appl. Opt.* **30**, 1325-1327 (1991).
7. R.W. Gershberg, "Super-resolution through error energy reduction", *Opt. Acta* **21**, 709-720 (1974).
8. W.H. Press, B.P. Flannery, S.A. Tevkolsky, W.T. Vetterling, "Numerical Recipes the Art of Scientific Computing (FORTRAN Version)", Cambridge University Press, reprinted (1990).
9. F. Roddier, "The effects of atmospheric turbulence in optical astronomy", *Progress in Optics* **19**, 283-376 (1981).
10. R.J. Noll, "Zernike polynomials and atmospheric turbulence", *J. Opt. Soc. Am.* **66**, 207-211 (1976).

11. J.-M. Mariotti, "Introduction to Fourier optics and coherence", in proc. NATO ASI *Diffraction limited imaging telescopes*, Cargese, 3-32, September 1988, D.M. Alloin and J.-M. Mariotti (eds.) Kluwer Academic Pub.
12. F. Roddier, "Status of Astronomical Adaptive Optics Developments", in proc. of *High Resolution by Interferometry II*, ESO-Garching RFA, 11-14 October 1991.
13. Fried, D. L., *J. Opt. Soc. Am.* **56**, 1372 (1966).