

70
24



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE CIENCIAS

**PROGRAMA PARA EL DISEÑO DE DOBLETES
ACROMATICOS**

T E S I S
QUE PARA OBTENER EL TITULO DE
F I S I C O
P R E S E N T A :
CITLALI LOPEZ ORTIZ

México, D. F.

1991

**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

	Página
Introducción General	3
Definiciones	5
Capítulo 1. Aberraciones Ópticas y Dobletes	7
1.1 Introducción	7
1.2 Aberraciones Ópticas	9
1.3 Por qué un Doblete	16
1.4 Óptica Geométrica para el Diseño de Dobletes	17
1.5 Comentarios finales	26
Capítulo 2. Algoritmo de Diseño y el Programa Doblete	27
2.1 Introducción	27
2.2 Algoritmo de Diseño	29
2.3 Modificaciones al Algoritmo de Diseño	33
2.4 Implementación del programa	34
2.5 Guía para el uso del Programa Doblete	37
2.6 Comentarios finales	39
Capítulo 3. Diseño con el Programa Doblete	40
3.1 Introducción	40
3.2 Ejemplos de Diseño de dobles	41
3.3 Conclusiones	54
Conclusiones Generales	55
Bibliografía	57
Apéndice	58

INTRODUCCION GENERAL

En la actualidad, el trabajo de diseñador de lentes consta de dos etapas:

- 1) La elección y prediseño del sistema de interés.
- 2) La optimización del prediseño en una computadora usando programas de diseño óptico. Muchos de estos son programas son poderosos paquetes comerciales, profesionales de diseño .

La primera parte del trabajo es muy importante porque en ella el diseñador desglosa el problema y trata de analizar el funcionamiento del sistema en un prediseño que se acerque tanto como sea posible a la solución del problema.

En la segunda parte, las modificaciones que sufre el sistema están únicamente bajo control de la computadora y en caso de que el resultado de la optimización del sistema sea de muy difícil o imposible construcción en el Taller Óptico, el diseñador debe cambiar los parámetros de optimización o bien, encontrar un nuevo prediseño que lleve a la computadora a optimizarlo en otra región del espacio solución, esperando obtener un diseño de construcción factible.

Dada la importancia de la obtención de "buenos" prediseños, en este trabajo se planteó el problema de programar en lenguaje Turbo Pascal V 5.0 un algoritmo de diseño de dobletes acromáticos cementados propuesto por R.

Kingslake¹. Se espera obtener un buen diseño, que prácticamente no requiera posterior arreglo, o alguno mínimo si acaso, utilizando ya un paquete comercial de alta calidad para el Diseño Óptico. Se muestran en particular, algunos ejemplos de dobletes optimizados con este método que presentan un comportamiento satisfactorio.

En el Capítulo 1 se presenta un tratamiento de la Óptica Geométrica para aberraciones ópticas de tercer orden y se establece el marco teórico necesario para el diseño de Dobletes acromáticos.

En el Capítulo 2 se expone el algoritmo de diseño de dobletes acromáticos cementados y la implementación del mismo en un lenguaje de programación. Este capítulo constituye en sí un manual para el manejo del programa implementado.

El Capítulo 3 contiene algunos diseños obtenidos en el programa Doblete y se hace un análisis cualitativo de los mismos atendiendo a variaciones e los datos de entrada.

Finalmente, se enuncian las principales conclusiones derivadas de este trabajo y se exponen algunos puntos que, como resultado del trabajo desarrollado, sería conveniente explorar posteriormente.

En el Apéndice se presentan los listados del programa Doblete y de las utilerías programadas, comentados renglón por renglón.

¹Ref 171

Para los ángulos, U designa la inclinación de un rayo meridional como positiva si se rota en sentido de las manecillas del reloj al ir del eje óptico al rayo. El ángulo de incidencia I es positivo si se efectúa una rotación contra las manecillas del reloj para ir de la normal a la superficie del rayo.

Por último, todos los datos que relacionan la porción del rayo que está en el espacio izquierdo de una superficie, usualmente el espacio objeto, son representados con símbolos no primados, mientras que si nos referimos a datos en la parte de un rayo que está en el espacio a la derecha de una superficie, se denotan por símbolos primados.

CAPITULO 1

ABERRACIONES OPTICAS Y DOBLETES

1.1 INTRODUCCION

El problema del Diseño Óptico consiste en encontrar un conjunto de parámetros de una lente que minimicen los defectos en la imagen producida por ella. La dificultad radica en que para el caso ideal, todos los rayos para todas las longitudes de onda que se originan en un punto objeto dado, deben hacerse pasar a través de la lente de tal forma que la imagen de ese punto sea puntual también y la imagen del plano objeto sea también un plano, esto es, sin aparición de desviaciones en las imágenes de líneas rectas. Estos errores en la imagen son llamados aberraciones ópticas. Las hay de diferentes tipos y en general, aparecen todas mezcladas en la imagen .

Los parámetros o grados de libertad que el diseñador tiene a su disposición para tratar de encontrar una solución al problema de minimizar las aberraciones ópticas son en general, los radios de curvatura de las superficies, los grosores, los espaciamientos de aire, los tipos de vidrios con sus respectivos índices de refracción y las aberturas o diafragmas. Al variar los grados de libertad para minimizarlas, debe siempre mantenerse fija la distancia focal de la lente o sistema, porque en caso contrario se obtiene un sistema diferente al que se quiere construir.

En la sección 1.2 se presenta una breve descripción de las aberraciones ópticas más importantes

La explicación de por qué escoger un doblete como sistema a diseñar se expone en la sección 1.3

La sección 1.4 contiene los desarrollos de la óptica geométrica necesarios para el diseño de dobletes en este trabajo.

Los comentarios finales a este capítulo están contenidos en la sección 1.5.

1.2 ABERRACIONES ÓPTICAS

Existen dos clasificaciones principales de las aberraciones ópticas:

- I) Aberraciones monocromáticas
- II) Aberraciones cromáticas

Las primeras, consideradas únicamente en aproximación a tercer orden, pueden subdividirse en dos tipos:

i) Las que deterioran la nitidez de la imagen haciéndola "confusa": aberración esférica, coma y astigmatismo.

(i) Las que "deforman" la imagen: curvatura de campo y distorsión.³

Las segundas se deben a que el índice de refracción n depende de la longitud de onda.

En seguida se presenta una breve descripción de ellas.

1.2.1 Aberraciones monocromáticas

1.2.1.1 Aberración Esférica.

Se entiende como *aberración esférica* la variación de la distancia focal con la altura de incidencia del rayo en la lente.

Para hacer un cálculo directo de esta aberración, se traza un rayo meridional del objeto a la imagen y se encuentra la distancia imagen L' .

³ Ref (7)

Esta distancia se compara directamente con la distancia imagen paraxial l' correspondiente al mismo punto objeto. Por lo tanto⁴,

$$\text{aberración esférica longitudinal} = LA' = l' - l' \quad (1)$$

Si consideramos un plano perpendicular al eje óptico que pase por el foco paraxial y calculamos la distancia al eje de la intersección de los rayos con este plano obtenemos la *aberración esférica transversal*:

$$\text{aberración esférica transversal} = LA' = LA' \tan \theta \quad (2)$$

donde θ es el ángulo que se muestra en la figura 2.

La aberración esférica puede ser expresada en términos de la altura del rayo incidente en forma de serie de potencias. Este desarrollo es bastante útil en el análisis de sistemas ópticos y es de la forma

$$LA' = aY^2 + bY^4 + cY^6 + \dots \quad (3)$$

En esta ecuación el primer término es llamado aberración esférica primaria y más adelante nos ocuparemos de él.

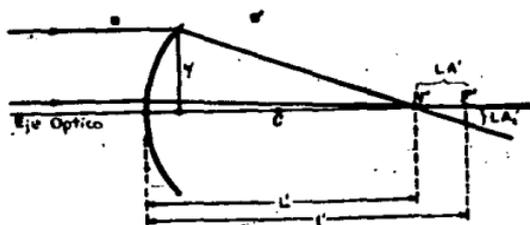


Figura 2

Aberración esférica longitudinal y transversal

⁴ Ref [7]

1.2.1.2 Coma.

La aberración de coma se debe a que la amplificación en la imagen es diferente para rayos que cruzan la lente por distintas zonas. Si la amplificación para los rayos exteriores en una lente es mayor que para los rayos centrales, la coma es positiva, mientras que si sucede al revés como en el diagrama de la figura 3, la coma es negativa. Usando la teoría de aberraciones de tercer orden es posible encontrar que la coma sagital para una lente simple puede escribirse como

$$C_s = \frac{j h^2 (G p + W q)}{f^3} \quad (4)$$

donde $j, h, y f$ son las distancias que se indican en el diagrama. G, W, p y q corresponden a las siguientes fórmulas

$$G = \frac{3(2n + 1)}{4n} \quad W = \frac{3(n + 1)}{4n(n - 1)} \quad p = \frac{l' - l}{l' + l} \quad q = \frac{r_2 + r_1}{r_2 - r_1} \quad (5)$$

aquí n es el índice de refracción de la lente, l y l' las distancias objeto e imagen respectivamente y r_1, r_2 , los radios de curvatura de la primera y segunda superficies de la lente.

La coma tangencial puede expresarse en términos de la coma sagital de la manera siguiente:

$$C_T = 3 C_s \quad (6)$$

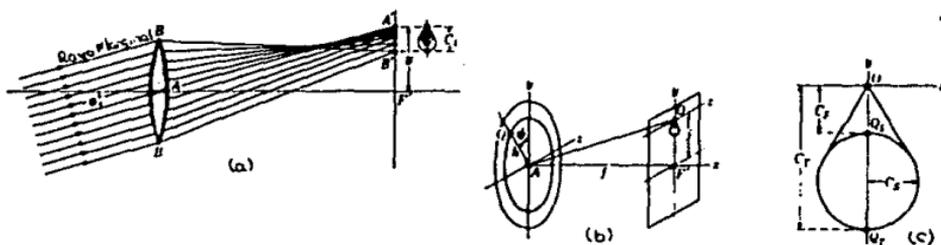


Figura 3

- (a) Coma del haz de rayos tangenciales (b) Geometría para calcular coma
(c) Relación entre C_T y C_s

1.2.1.3 Astigmatismo.

Para explicar la aberración de astigmatismo consideremos rayos emergiendo de un punto objeto Q fuera de eje como en la figura 4. Si nos fijamos en el haz de rayos que además están contenidos en el plano tangencial, observamos que se cruzarán en el espacio imagen en el punto T. Considerando ahora los que están contenidos en el plano sagital, vemos que todos pasarán por S en el espacio imagen. Determinando las posiciones de los puntos imagen T y S para un conjunto de puntos objeto encontraremos que están contenidos en superficies parabólicas como se muestra en la figura. De acuerdo con esto, la cantidad de *astigmatismo* o *diferencia astigmática* para un haz de rayos está dada por la distancia entre estas dos superficies medida a lo largo del rayo principal. El rayo principal de un haz oblicuo de rayos es el rayo que se encuentra a la misma distancia de los rayos superior e inferior que limitan el haz y pasan a través del sistema⁵.

Las ecuaciones de las distancias astigmáticas para una sola superficie refractora son :

$$\frac{n \cos^2 \phi}{l} + \frac{n' \cos^2 \phi'}{l_T'} = \frac{n' \cos \phi' - n \cos \phi}{r} \quad \text{y} \quad \frac{n}{l} + \frac{n'}{l_S'} = \frac{n' \cos \phi' - n \cos \phi}{r} \quad (7)$$

donde ϕ y ϕ' son los ángulos de incidencia y refracción del rayo principal, r el radio de curvatura, l la distancia objeto y l_T' y l_S' las distancias imagen T y S medidas a lo largo del rayo principal.

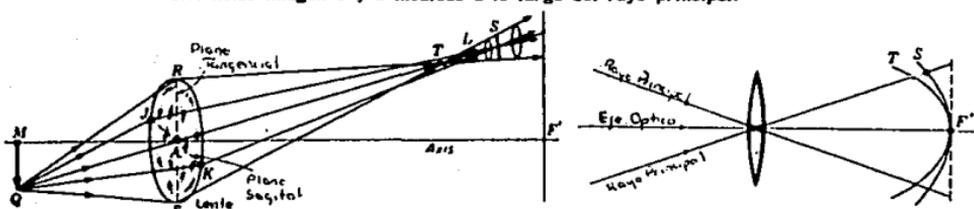


Figura 4

(a) Vista en perspectiva (b) Corte meridional

⁵ Ref [7]. Algunos autores definen de otra manera.

1.2.1.4 Curvatura de Campo.

Aún cuando un sistema esté corregido de astigmatismo, la superficie focal, en general, no es plana. Esta aberración se conoce como *curvatura de campo*. Una ecuación, a tercer orden, para el radio de la superficie focal en un sistema de k superficies es:

$$\frac{1}{\rho} = -nk' \sum_{j=1}^k \left(\frac{n_j' - n_j}{n_j n_j' r_j} \right) \quad (8)$$

donde r_j es el radio de la j -ésima superficie, n_j y n_j' son los índices de refracción de la interfase de radio r_j .

Para un sistema de lentes delgadas se cumple la ecuación

$$\frac{1}{r} = \frac{n-1}{n} \left(\frac{1}{r_1} - \frac{1}{r_2} \right) = -\frac{1}{nf} \quad (9)$$

donde f es la distancia focal del sistema, por lo tanto, en este caso, la ecuación (8) puede simplificarse como

$$\frac{1}{\rho} = -\sum_{j=1}^k \left(\frac{1}{n_j f_j} \right) \quad (10)$$



Figura 5

T = curva de astigmatismo tangencial

S = curva de astigmatismo sagital

P = superficie focal libre de astigmatismo de radio ρ

1.2.1.5 Distorsión

La aberración por distorsión se manifiesta como una amplificación no constante para diferentes alturas de la imagen . El cálculo de la distorsión en un punto imagen se obtiene de la resta siguiente :

$$D = y_{\text{imagen}} - y_{\text{paraxial}} \quad (11)$$

De esta forma, la distorsión es positiva si la amplificación aumenta con la altura y negativa si disminuye (ver figura 6).

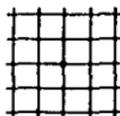


Imagen de una red sin distorsión

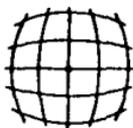


Imagen de la misma red con distorsión positiva

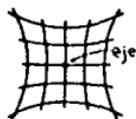


Imagen de la misma red con distorsión negativa

Figura 6

1.2.2 Aberraciones cromáticas

Debido a que el índice de refracción de un medio óptico varía con la longitud de onda de la luz, se sigue que cada propiedad que dependa del índice de refracción de una lente, también variará con la longitud de onda. Esto incluye la distancia focal, la aberración esférica, la curvatura de campo y todas las demás aberraciones. Este efecto es conocido como *aberración cromática*.

La *aberración cromática longitudinal* se define como

$$\text{aberración cromática longitudinal} = Lc' = Lr' - Lc' \quad (11)$$

donde las cantidades Lr' y Lc' son las distancias imagen en las líneas F (486.1327 nm) y C (656.2725 nm) del espectro electromagnético, respectivamente.

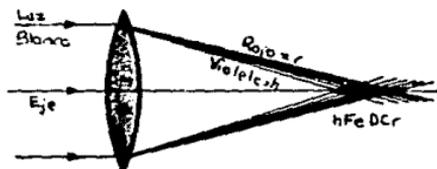


Figura 7
Aberración Cromática longitudinal

Un recurso usual es expresar la aberración cromática como serie de potencias en función de la altura del rayo incidente, obteniendo un desarrollo en potencias pares de la altura :

$$Lc' = a + bY^2 + cY^4 + \dots \quad (12)$$

El término constante a es llamado *paraxial* o *aberración cromática primaria*. Más adelante lo calcularemos en forma explícita.

1.3 POR QUE UN DOBLETE

En el diseño y fabricación de sistemas ópticos, el diseño de dobletes cuenta con particular importancia. Esto se debe, entre otras razones, a las siguientes:

(i) Después de una lente simple corregida de aberraciones, el siguiente sistema en complejidad es un doblete.

(ii) La idea de construir dobletes se debe a tratar de compensar las aberraciones de una lente simple. Se piensa en un doblete como en un sistema de dos lentes simples, una positiva y otra negativa, que cancelen mutuamente las aberraciones introducidas por cada una de ellas. Así se tiene la ventaja de que crece el número de grados de libertad del sistema y por lo tanto existe una mayor probabilidad de encontrar una buena solución al problema planteado.

(iii) Este tipo de componentes es de gran utilidad cuando se quiere obtener la imagen de un objeto cercano al eje y de tamaño angular pequeño. Por consiguiente la aplicación más relevante es su uso en objetivos de telescopio.

Por lo anterior, en un doblete es posible corregir las aberraciones esférica y cromática de manera satisfactoria y el campo utilizado es tan pequeño que las demás aberraciones no introducen ningún problema.

Podemos clasificar los dobletes en dos grupos importantes: los que tienen espaciamiento de aire entre las dos componentes y los que son cementados. Esto último significa que la curvatura interna de las lentes que conforman el doblete coinciden perfectamente y se ponen en contacto. En este trabajo se considerarán únicamente dobletes del segundo tipo.

Hagamos un estudio de aberraciones esférica y cromática para dobletes cementados.

1.4 ÓPTICA GEOMÉTICA PARA EL DISEÑO DE DOBLETES

1.4.1 Aberración esférica.

La relación simple para encontrar la aberración esférica de la ecuación (1) resulta frecuentemente inadecuada para el análisis de un sistema óptico porque no nos da indicio de dónde aparece la aberración. Entonces es mucho más útil calcular la aberración como la suma de contribuciones en cada superficie.

Para ello consideremos el siguiente diagrama

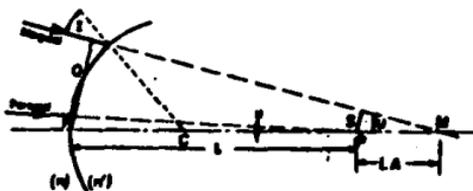


Figura 8
Aberración Esférica Longitudinal

Aquí se muestran un rayo marginal y uno paraxial incidiendo en una superficie esférica. La longitud S es la perpendicular trazada del punto objeto paraxial P al rayo marginal. El rayo marginal está definido por Q y U, el paraxial por y y u.

Considerando la ecuación de trazo

$$S = Q - l \operatorname{sen} U \quad (13)$$

y multiplicando por u obtenemos

$$Su = Qu - y \operatorname{sen} U \quad (14)$$

Si reemplazamos u en la derecha por $yc - l$ y $\operatorname{sen} U$ por $Qc - \operatorname{sen} l$ y multiplicamos toda la ecuación por n obtenemos

$$Snu = ync - y \operatorname{sen} l - Qni \quad (15)$$

Haciendo lo mismo para el rayo refractado y sustrayendo la ecuación sin primas de la ecuación primada llegamos a

$$S'n'u' - Snu = (Q - Q')ni \quad (16)$$

Escribiendo esto para k superficies y sumando, después de múltiples cancelaciones debido a que $(S'n'i')_i = (Snu)_i$, resulta

$$(S'n'u')_k - (Snu)_i + \sum (Q - Q')ni \quad (17)$$

Fijándonos en la figura 8 obtenemos

$$LA = S/\operatorname{sen} U \text{ y } LA' = S'/\operatorname{sen} U' \quad (18)$$

entonces podemos reescribir (17) y despejar LA'

$$LA' = LA \left(\frac{n_i u_i \operatorname{sen} U_i}{n_k' u_k' \operatorname{sen} U_k'} \right) + \sum \frac{(Q_i - Q_i') n_i l_i}{n_k' u_k' \operatorname{sen} U_k'} \quad (19)$$

Esta es ya una ecuación bastante más útil para el diseñador que la ecuación (1). En esta expresión, la cantidad bajo la sumatoria es la contribución de cada superficie a la aberración esférica para el rayo considerado y el primer término es la transferencia de la aberración del objeto a través de la lente hasta el espacio imagen.

Manipulando esta ecuación y tomando la aproximación paraxial es posible llegar a la siguiente expresión para la aberración esférica primaria

$$LA'_p = LA_p \left(\frac{n_i u_i^2}{n_k' u_k'^2} \right) + \sum_i \frac{y n_i i_i (u-u') (i-i')}{2 n_k' u_k'^2} \quad (20)$$

Con esta expresión para las contribuciones superficiales, que denotaremos por SC, para un conjunto de lentes delgadas cercanas, la contribución de la aberración esférica primaria a la imagen final es

$$SC = - \frac{y^4}{n_o' u_o'^2} \sum (G_{1c}^2 - G_{2c}^2 c_i + G_{3c}^2 v_i + G_{4c} c_i^2 - G_{5c} v_i + G_{6c} v_i^2) \quad (21)$$

donde los términos con sufijo 0 se refieren a la imagen final y los otros términos se aplican a cada elemento simple. Aquí c y c_i tienen sus significados usuales, $c_i = 1/r_i$ y $c = (1/f)(n-1)$. El símbolo v_i es el recíproco de la distancia objeto del elemento en cuestión y las G_i son funciones del índice de refracción, a saber,

$$\begin{aligned} G_1 &= 0.5 n^2 (n-1) & G_2 &= 0.5 (2n+1)(n-1) \\ G_3 &= 0.5 (3n+1)(n-1) & G_4 &= 0.5 (n+2)(n-1)/n \\ G_5 &= 2 (n^2-1)/n & G_6 &= 0.5 (3n+2)(n-1)/n \end{aligned} \quad (22)$$

De esta forma obtenemos

$$LA'_p = LA_p \left(\frac{i'}{i} \right)^2 + y^2 i'^2 \sum (\text{suma de las } G\text{'s}) \quad (23)$$

Para el caso de un doblete cementado con $LA_p = 0$ esta ecuación toma la forma⁶

$$LA'_p = y^2 i'^2 (\text{suma de } G\text{'s}_a + \text{suma de } G\text{'s}_b) \quad (24)$$

⁶ Ref [1]

En este caso, de la ecuación de Gauss para lentes delgadas y considerando un doblete cementado se cumplen las ecuaciones

$$(v_1)_b = (v_1)_a + c_a (n_a - 1) \quad (25)$$

y

$$c_{1b} = c_{2a} \quad (26)$$

a su vez,

$$c_{1a} = c_a + c_{2a} \quad (27)$$

y

$$c_{1b} = c_b + c_{2b} \quad (28)$$

Haciendo estas consideraciones y desarrollando completamente la ecuación (24) se obtiene una ecuación de la forma

$$L_{Ap}' = a c_1^2 + b c_1 + c \quad (29)$$

o bien

$$L_{Ap}' = A c_2^2 + B c_2 + C \quad (30)$$

De aquí, si se tiene una propuesta para los valores de c_a y c_b del doblete, resolviendo la ecuación $L_{Ap}' = 0$, pueden encontrarse una o dos soluciones reales para las curvaturas de cada superficie y obtener una propuesta para las curvaturas de los elementos del doblete con corrección de aberración esférica a primer orden.

En caso de no existir solución real a la ecuación $L_{Ap}' = 0$ puede calcularse el mínimo del valor absoluto de la función $L_{Ap}'(c_1)$ o $L_{Ap}'(c_2)$ y tomar ese número como solución al problema.

1.4.2 Aberración Cromática

1.4.2.1 Ecuaciones para lentes delgadas.

Para corregir la aberración cromática en un doblete cementado consideremos el siguiente desarrollo.

La relación clásica de las distancias objeto a imagen para lentes delgadas es

$$\frac{1}{l} = \frac{1}{l'} + (n - 1)c \quad (31)$$

Escribiendo esta ecuación para las líneas F y C del espectro electromagnético y restando obtenemos

$$\frac{1c' - 1r'}{l'^2} - \frac{1c - 1r}{l^2} = (n_F - n_C)c = -\frac{1}{fV} \quad (32)$$

donde V es el número de Abbe definido por

$$V = (n_D - 1)/(n_F - n_C) \quad (33)$$

Usando la definición de L_{ch} (ecuación (11)) tenemos

$$\frac{L_{ch}'}{l'^2} - \frac{L_{ch}}{l^2} = -\frac{1}{fV} \quad (34)$$

Para un sistema de lentes delgadas y cercanas, podemos sumar las contribuciones de cada elemento y llegamos a

$$\frac{L_{ch}'}{l'^2} - \frac{L_{ch}}{l^2} = \Sigma - \frac{1}{fV} \quad (35)$$

La cantidad en el miembro izquierdo de la ecuación se llama el

residuo cromático R. Entonces, usando la ecuación para la potencia de una lente,

$$R = \sum - \frac{1}{fV} = - \sum (c \Delta n) \quad (36)$$

En el caso de un doblete, R toma la forma

$$R = -c_a \Delta n_a - c_b \Delta n_b \quad (37)$$

Ahora consideremos la potencia de un doblete delgado cementado

$$\phi = V_a (c \Delta n)_a + V_b (c \Delta n)_b \quad (38)$$

desarrollando la curvatura de cada elemento obtenemos

$$\phi = V_a ((c_1-c_2)_a \Delta n_a) + V_b ((c_1-c_2)_b \Delta n_b) \quad (39)$$

Sabiendo que $\phi = 1/F'$ y usando (37) y (39) para despejar c_a y c_b , encontramos

$$c_a = \frac{1}{F' (V_a - V_b) \Delta n_a} + \frac{R V_b}{F' (V_a - V_b) \Delta n_a} \quad (40)$$

$$c_b = \frac{1}{F' (V_b - V_a) \Delta n_b} + \frac{R V_b}{F' (V_b - V_a) \Delta n_b} \quad (41)$$

Estas ecuaciones se llaman las ecuaciones (c_a, c_b) y son usadas para iniciar el diseño de un doblete acromático delgado.

Después de usar las ecuaciones (c_a, c_b) como una primera aproximación a la solución buscada es posible obtener un mayor grado de corrección de la aberración cromática usando el método $D - d$ de Conrady que se explica a continuación.

1.4.2.2 Método D - d⁷

Este método surge de la idea de considerar la aberración cromática como una diferencia de caminos ópticos de los frentes de onda para diferentes longitudes de onda.

En la figura 9, O designa cualquier sistema de lentes que refracta la luz de un punto objeto B a un punto imagen B'. Del punto B emergerán ondas esféricas hacia el sistema de lentes y éste alterará la curvatura de los frentes de onda y los dirigirá hacia B'.

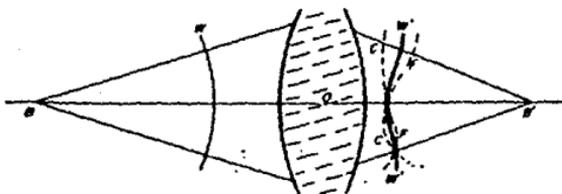


Figura 9

Aberración cromática como una diferencia de caminos ópticos

Desde este punto de vista la aberración cromática consiste en que las ondas refractadas correspondientes a diferentes colores tendrán mayor o menor curvatura que W' cuando sean tangentes entre sí en el eje óptico. El caso de acromaticidad perfecta existiría si fuera posible hacer coincidir completamente todos estos frentes de onda de diferentes colores con W' cuando pasan por el mismo punto del eje óptico. Esta condición física de acromatismo implica una forma geométrica apropiada para el cálculo si introducimos caminos ópticos entre los frentes de onda W y W'.

Por definición de frentes de onda, los caminos ópticos de puntos

⁷ Ref [2]

correspondientes de W y W' son exactamente iguales. Introduzcamos el símbolo d' para las distancias recorridas por el frente de onda a lo largo del eje óptico y D en el caso de distancias a cierta altura respecto al eje óptico. Si consideramos esta nomenclatura para cada índice de refracción atravesado por el frente de onda, trayectoria total entre W y la superficie de la primera lente y entre la última superficie y W' , tendremos la trayectoria a lo largo del eje como $\Sigma d' \cdot n'$ y aquella a lo largo del rayo marginal igual a $\Sigma D \cdot n'$. Entonces la condición de igualdad de los caminos ópticos para las ondas conjugadas W y W' es

$$\Sigma(D'-d') \cdot n' = 0 \quad (42)$$

donde la suma se extiende a todos los espacios de lentes entre W y W' ya que para propósitos de diseño óptico se desprecian las contribuciones los espaciamentos de aire entre lentes.

La condición cromática demanda que además

$$\Sigma(D'-d') \cdot (n' + \Delta n') = 0 \quad (43)$$

donde $\Delta n'$ es un pequeño incremento en el índice refractivo correspondiente a pequeñas diferencias en longitud de onda. La diferencia de las dos ecuaciones anteriores es la que expresa la condición cromática que deseamos. Esta es⁸

$$\Sigma(D'-d') \cdot \Delta n' = 0 \quad (44)$$

Si deseamos corrección cromática de las líneas F y C del espectro entonces simplemente tomamos

⁸ Esta resta no se realiza de manera rigurosa ya que en la ecuación (42) $D-d$ debe calcularse para el índice n y en la ecuación (43) debe calcularse para $n + \Delta n$ entonces $(D-d)11 \neq (D-d)12$. Sin embargo esta diferencia se desprecia. A pesar de ello, este método siempre ha funcionado eficientemente para los diseñadores ópticos como un método de corrección fina de aberración cromática.

$$\Delta n' = nr' - nc' \quad (45)$$

y podemos reescribir la ecuación (44) como⁹

$$\Sigma(D'-d') \cdot (nr' - nc') = 0 \quad (46)$$

La longitud D se encuentra usando la relación¹⁰

$$D = (d + X_2 - X_1)/\cos U' \quad (47)$$

donde $X = r [1 - \cos(U+1)]$ y los subíndices 1 y 2 se refieren a la primera y segunda superficies de cada elemento.

Ahora bien, en la práctica, para acromatizar una lente y hacer que $\Sigma(D'-d') \cdot \Delta n'$ sea igual a cero, generalmente se ajusta la curvatura de la última superficie. Supongamos que la suma $D-d$ para todas las lentes anteriores a la última es Σ_0 , entonces, para la última debe cumplirse

$$\Sigma(D'-d') \cdot \Delta n' = -\Sigma_0 \quad (48)$$

De aquí, dada d' , se puede despejar el valor de D' para obtener acromaticidad. Posteriormente se calculan X_1 y Y_1 del último elemento con las ecuaciones

$$X_2 = D' \cos(U_1') + X_1 - d \quad \text{y} \quad Y_2 = Y_1 - D \text{ sen}(U_1') \quad (49)$$

A continuación se obtiene el valor del radio de curvatura de la última superficie usando la relación

$$r = (X_2^2 + Y_2^2)/2X_2 \quad (50)$$

y el problema está resuelto.

⁹ En este punto surge la pregunta Para que índice se calcula $D-d$? Para nF o para nC ? En este trabajo se ha decidido hacer el calculo para nD ya que $nF > nD > nC$ y se obtiene una aproximacion intermedia del valor $D-d$.

¹⁰ Ref [7]

1.5 COMENTARIOS FINALES

En este Capítulo se han presentado las aberraciones monocromáticas a y la aberración cromática a tercer orden. Se ha discutido porqué la importancia de considerar un doblete. Se ha desarrollado la teoría necesaria para corrección de aberración esférica en aproximación paraxial y corrección de aberración cromática en aproximación de lentes delgadas y con el método $D - d$ de Conrady.

Esta es la base de la cual se partirá para desarrollar un algoritmo de diseño de dobletes cementados.

CAPITULO 2

ALGORITMO DE DISEÑO Y EL PROGRAMA DOBLETE

2.1 INTRODUCCION

Basándose en los métodos descritos en el capítulo anterior es posible obtener un algoritmo que optimice el diseño del doblete deseado sin necesidad de utilizar las técnicas usuales de optimización programadas en paquetes comerciales de Diseño Optico¹¹. Este proceso alternativo de diseño fue reportado por primera vez por Conrady¹². Constituye uno de los métodos de algebraicos de diseño de dobletes utilizados por los diseñadores opticos principalmente en el siglo pasado y principios del presente, antes de la introducción del cálculo de sistemas ópticos en computadoras y supercomputadoras. Es hasta 1978 que este algoritmo es presentado por Kingslake¹³ en forma de diagrama de flujo para ser programado en una computadora. No se conocen paquetes o programas de Diseño Optico que lo

¹¹ Estos paquetes utilizan técnicas sofisticadas de análisis numérico para encontrar mínimos de una función de mérito que en general es de la forma $\Phi = a_1 R_1^2 + a_2 R_2^2 + \dots + a_n R_n^2$ donde las a_i son coeficientes de peso y las R_i son los cuadrados de las aberraciones como función de los parámetros libres del sistema. Consultar referencias [3] y [4].

¹² Ref [1]

¹³ Ref [7]

utilicen y puede deberse a que en éstos, los radios de curvatura del sistema son siempre datos de entrada y en este algoritmo aparecen como datos de salida. Este hecho le da relevancia a su implementación en un programa computacional.

En la sección 2.2 se presenta y explica el algoritmo de diseño propuesto por R. Kingslake.

La sección 2.3 contiene una lista de las principales modificaciones al algoritmo de diseño que resultaron necesarias en la programación.

La sección 2.4 consta de dos partes: las características principales de la implementación del programa doblete y una lista de las unidades programadas.

La guía para el uso del programa constituye la sección 2.5.

Finalmente, en la sección 2.6 están los comentarios finales del Capítulo 2.

2.2 ALGORITMO DE DISEÑO

Si se considera el problema de encontrar los radios de curvatura adecuados para obtener un doblete corregido de esfericidad y cromatismo, podemos utilizar las ecuaciones desarrolladas en la sección 1.4 con este fin.

Supóngase que se conocen ciertos datos del doblete a diseñar como son: la distancia focal, la distancia objeto, los tipos de vidrio y los grosores de cada elemento. Para dar un primer paso a la solución del problema pueden considerarse las ecuaciones (ca,cb) y así obtener información preliminar acerca de las curvaturas con corrección cromática. Ahora sería deseable conocer la curvatura de cada elemento con corrección de esfericidad. Para ello podemos utilizar la ecuación $L_{Ap}' = 0$ y encontrar dos posibles valores para c_2 . Eligiendo uno de ellos, mediante las ecuaciones (27) y (28), podemos obtener una primera solución al problema. Ahora bien, es sabido que el método D-d da una solución a la corrección de cromatismo que las ecuaciones (ca, cb) entonces se ignora la solución encontrada para c_2 y se resuelve el método D-d para la superficie J. Con esto, se tiene una mejor solución que la anterior, ya que se ha hecho una corrección fina de cromatismo para una distancia focal dada. Ahora se debe saber cómo corregir este diseño de manera iterativa para encontrar un diseño optimizado, que cumpla con los requisitos de tolerancias que el diseñador impone para la distancia focal -corrigiendo cromatismo- y la esfericidad.

El algoritmo iterativo de diseño es el que se muestra en el diagrama figura 10. En este caso los datos de entrada son los índices de refracción de cada vidrio en las líneas C, D y F, el grosor en la orilla para el elemento positivo, el grosor en el centro para el elemento negativo, la distancia focal deseada, su tolerancia, la aberración esférica permitida para el rayo marginal, su tolerancia y la distancia objeto. Los datos de salida son las curvaturas del doblete, el grosor central del elemento

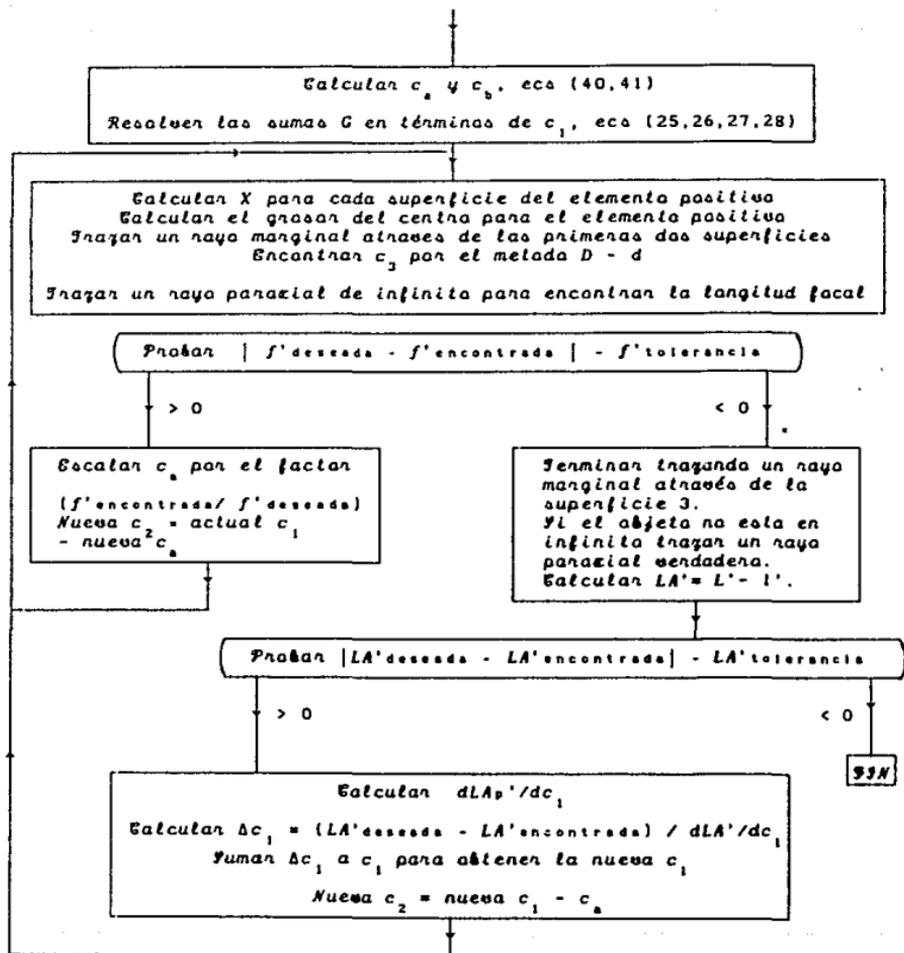


Figura 10
 Algoritmo de diseño

positivo, la distancia focal obtenida y la aberración esférica para el rayo marginal.

En este algoritmo, los parámetros libres para la corrección del diseño son las curvaturas de las superficies. Debido a que tenemos tres curvaturas libres, podemos corregir tres cantidades en el diseño. Primero, la aberración cromática, después la distancia focal y finalmente la aberración esférica. Esto se logra de la siguiente manera:

Como paso inicial, se calculan los valores para c_2 y c_3 utilizando las ecuaciones (40) y (41) con residuo cromático igual a cero. Esto asegura la corrección de aberración cromática a primer orden. Posteriormente se encuentra el valor de c_1 resolviendo la ecuación (29). Con esto hemos encontrado una primera solución al problema planteado ya que las ecuaciones (40), (41) minimizan aberración cromática y la ecuación (29) minimiza aberración esférica para una distancia focal dada y con estos valores se determinan las tres curvaturas. Pero esta primera solución puede no ser tan buena como se desea por lo tanto es necesario evaluar el diseño y hacer correcciones.

Suponiendo que la distancia focal obtenida no cae dentro de la tolerancia permitida, se realiza un escalamiento del diseño como se muestra en el diagrama de la figura 10 y esto se repite iterativamente hasta obtener un resultado que cumpla con los requisitos planteados. En este escalamiento, la aberración cromática para el primer elemento permanece corregida según la ecuación (40) y se altera según el factor de escalamiento del diseño. Pero la curvatura de c_3 se encuentra nuevamente por el método D-d y así se obtiene una solución nuevamente corregida de cromaticidad.

Una vez corregida esta parte, se procede a evaluar la aberración esférica. Si ésta no está dentro de la tolerancia permitida, se toma una aproximación lineal de la función $LA_p'(c_1)$ tratando de acercarse a un cero de la curva de aberración esférica en función de las curvaturas.

Habiendo corregido aberración esférica es necesario volver a iniciar todo el proceso a partir de la corrección de la distancia focal para

asegurar que hemos encontrado una solución al problema planteado.

Las ecuaciones involucradas en los procesos de diseño son ecuaciones para lentes delgadas. Esto implica que los diseños obtenidos están corregidos únicamente en esta aproximación, sin embargo, la evaluación final para el cálculo de tolerancias está hecha con trazo exacto de rayos y ecuaciones para lentes gruesas.

Cabe resaltar que este algoritmo no utiliza variables ortogonales como parámetros libres en el diseño y entonces no podemos hacer un análisis riguroso para asegurar su convergencia en todos los casos.

2.4 IMPLEMENTACION DEL PROGRAMA

2.4.1 Características principales del programa Doblete

El programa DOBLETE es un programa para microcomputadoras PC compatibles con IBM y monitor VGA o EGA¹⁴ que sirve como auxiliar en el diseño de dobletes cementados corregidos a primer orden de aberraciones por esfericidad y por cromatismo.

Este programa ofrece al usuario las siguientes opciones

- 1) Creación de archivos de datos de entrada.
- 2) Creación de archivos de datos de salida.
- 3) Obtención de diseños corregidos de esfericidad y cromatismo a primer orden.
- 4) Elección del número máximo de iteraciones deseadas en el procedimiento de diseño.
- 5) Graficación de aberración cromática longitudinal con escalamiento automático.
- 6) Graficación de aberración esférica longitudinal o transversal con escalamiento automático y posibilidad de defoco.
- 7) Despliegue de tablas de trazo exacto para rayos marginales en cada superficie para tres posibles longitudes de onda.
- 8) Despliegue de las constantes paraxiales del sistema.
- 9) Dibujo del doblete con escalamiento automático.

El programa contempla posibilidad de acceder todas estas opciones sin necesidad de salir del mismo para mandar llamar archivos o editores o cambiar de modo gráfico a modo texto como en algunos paquetes profesionales de diseño óptico.

Las características mencionadas fueron obtenidas programando en

¹⁴ Es posible usar el programa Doblete en cualquier tipo de pantalla modificando algunos parámetros en la unidad encargada de graficación (Unidad Dgráfica, ver apéndice).

Las características mencionadas fueron obtenidas programando en lenguaje Turbo Pascal V5.0. Este lenguaje de programación ofrece la ventaja de programación estructurada teniendo como límite único de utilización de memoria el tamaño físico de la memoria de la máquina. Esto se logra implementando una serie de programas independientes, compilados en disco, que son llamados a ejecución por el programa principal. Estos programas se conocen con el nombre de unidades.

Contando con esta facilidad en la programación se pensó en desarrollar un programa principal consistente en un menú de opciones donde cada una implicara una llamada a alguna unidad donde estarían las órdenes a ejecutar según la opción deseada. Esto tiene la ventaja de que con relativa facilidad se pueden quitar o añadir opciones sin alterar la estructura básica del programa.

Además las unidades pueden llamarse a ejecución entre sí. Como ejemplo, en el programa Doblete, las rutinas de diseño, graficación, y despliegue de tablas hacen uso del trazo exacto de rayos, entonces, basta la creación de una sola unidad de trazo exacto que es llamada cuando se necesita por cada una de estas rutinas. La versatilidad de este tipo de programación se pone de manifiesto con mayor evidencia en el hecho de que cada una de estas unidades funciona como un módulo independiente en la programación y puede ser llamada a ejecución por cualquier otro programa que la requiera sin ningún problema. Si deseamos trazar rayos a través de x número de superficies, simplemente se escribe un pequeño programa que haga x llamadas a la unidad de trazo exacto que utiliza el programa Doblete, sin tener que programar nuevamente el trazo de rayos o modificar el programa Doblete.

2.4.2 Unidades Programadas

La lista de unidades programadas puede dividirse en tres grupos atendiendo al nivel el que se realiza al llamada a ejecución en la corrida del programa :

- I. crt /Unidad de TPS.O de manejo de modo texto
Declara /Unidad de declaración de variables globales

- Dpresent/Unidad para la presentación de programa
 Diseño /Unidad del algoritmo de diseño del doblete
 Dleearch/Unidad de lectura e impresión de archivos
 Dgesferi/Unidad de graficación de aberración esférica
 Dgcromat/Unidad de graficación de aberración cromática
 Dgbujo /Unidad para dibujar el doblete
 Dtabmar /Unidad de despliegue de trazo marginal
 Dtabpar /Unidad de despliegue de constantes paraxiales
- II. dos /Unidad de TPS.0 de llamada al sistema operativo
 Datos /Unidad de lectura de datos de entrada
 DsumaGc2/Unidad de cálculo de la suma de G's
 Dmargina/Unidad de trazo marginal
 Dfocal /Unidad de cálculo y corrección de la
 distancia focal obtenida
 Dipparax/Unidad de cálculo de l'
 Dtolera /Unidad de evaluación de tolerancias
 Desferic/Unidad de cálculo de la aberración esférica
 Dguarda /Unidad para archivar los datos
 Dfunción/Unidad de declaración de funciones
 Dcal_x /Unidad de cálculo de sagitas
 graph /Unidad de graficación de TPS.0
 Dgrafica/Unidad de inicialización de modo gráfico
 Dgesfer /Unidad de cálculo de la ab. esférica para
 graficación
 Dgcrom /Unidad de cálculo de la ab. cromática para
 graficación
 Dtabla /Unidad de creación de la tabla de datos
 distancia focal obtenida
- III. Dtrazmer/Unidad de trazo meridional
 Dmenosd /Unidad del método D menos d
 Dtrazpar/Unidad de trazo paraxial

En el Apéndice se anexa el listado del programa y de las unidades implementadas. Allí se presenta una explicación renglón por renglón de las órdenes ejecutadas por el programa.

2.5 GUIA PARA EL USO DEL PROGRAMA

Notación. Aquí se ha conservado la notación utilizada hasta el momento con la salvedad de que el símbolo ' ha sido reemplazado por la letra p debido a restricciones del lenguaje de programación.

Para entrar en el programa, desde el sistema operativo de la computadora, se tecldea la palabra DOBLETE. Aparece la pantalla de presentación del programa de la siguiente forma:

```
Programa para el Diseño de
Dobletes Acromáticos

Por Citlali López Ortiz

Presione cualquier tecla..
```

Al presionar cualquier tecla, aparece la siguiente pantalla:

```
Indique la opción que desea

(C)rear un nuevo archivo de datos

(L)eer un archivo de datos ya creado

(S)alir del programa DOBLETE
```

Si se escoge la última opción, se regresa al sistema operativo de la computadora y si se escoge cualquiera de la otras dos opciones, el programa va indicando paso a paso los datos de entrada que necesita.

Después de acceder este menú, se despliega un nuevo menú de opciones

<p style="text-align: center;">Indique la opción que desea</p> <p style="text-align: center;">(L)istar los datos del doblete Graficar aberración (E)sférica Graficar aberración (C)romática (P)intar el doblete (T)abla de trazo marginal (V)alores paraxiales (S)alir del programa</p>

Si se escoge la opción (L) se despliegan los datos del doblete y existe la posibilidad de mandarlos a impresión o no.

Si se escoge la opción (E), el programa pregunta si se desea graficar la aberración esférica longitudinal o la transversal y con qué defoco hacerlo.

Si se oprime la letra (C), se grafica la curva de aberración cromática longitudinal con el defoco deseado.

Si se opta por la opción (P) se hace un dibujo escalado del doblete.

Tecleando la letra (T) el programa pregunta en qué línea se desea el trazo marginal (roja, amarilla o azul) y se despliega una tabla de trazo superficie por superficie.

Cuando se elige la letra (V) aparece una tabla con la distancia focal

efectiva del doblete, su número f y los puntos nodales primario y secundario.

Finalmente, al teclear la letra (S), se regresa a la primera pantalla de opciones.

2.6 COMENTARIOS FINALES

Como comentario final a este capítulo es importante recalcar la estructura modular del programa doblete. Esta característica da gran flexibilidad al usuario que desee modificar o complementar el programa y da la posibilidad del uso de las mismas rutinas con diversos fines. De esta manera puede irse conformando una librería básica de rutinas de diseño óptico que tienen la ventaja de ser conocidas por el usuario y no conforman una "caja negra" como en la mayoría de los paquetes de diseño óptico

CAPITULO 3

DISEÑO CON EL PROGRAMA DOBLETE

3.1 INTRODUCCION

Con miras a obtener experiencia con el método expuesto anteriormente, se decidió encontrar un diseño como referencia para hacer cambios en algunos de los parámetros de entrada y observar las modificaciones en los diseños resultantes.

Se eligió un sistema $f/10$ (con $f' = 10$ cm) y objeto puntual en infinito del tipo objetivo para telescopio ya que esta es la aplicación principal de los dobletes acromáticos cementados.

3.2 EJEMPLOS DE DISEÑO DE DOBLETES

3.2.2 Doblete Acromático f/10

Supongamos que se desea diseñar un doblete acromático f/10 con objeto en infinito, distancia focal de 10 cm y un error del 0.5%¹⁵, esto es, $f' = (10 \pm 0.05)$ cm. Para elegir la tolerancia en la aberración esférica longitudinal, se hicieron pruebas en el programa y se vio que se razonable pedir que ésta tenga un valor en el intervalo (± 0.009) cm.

Seleccionando vidrios BK7 y F4 con 0.1 cm de grosor central para el elemento positivo y 0.2 cm como grosor central para el elemento negativo, se tienen todos los datos que necesita el programa Doblete para encontrar una solución.

Después de introducir los datos, se obtienen los resultados de la tabla 1.

Distancia focal	Aberración esférica
$f' = 10.003066$	$LA' = 0.008774$
Radios de curvatura	
$r1 = 4.569817$	
$r2 = -4.279410$	
$r3 = -60.703233$	

Tabla 1
Doblete BK7F4

¹⁵ Los fabricantes de dobletes acromáticos comerciales generalmente reportan una tolerancia en la distancia focal de sus sistemas del 2.0%

El dibujo del doblete reportado por el programa es el siguiente:

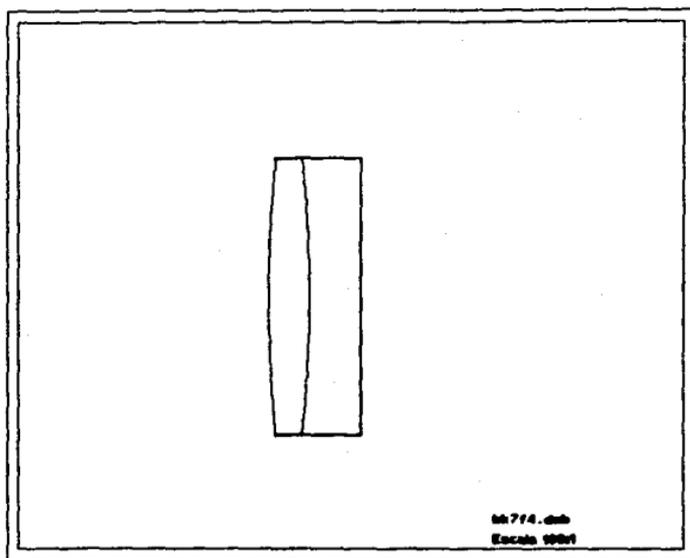


Figura II
Dibujo escalado del doblete.

Nota: La escala indicada se aplica únicamente para el despliegue del dibujo en monitor de la computadora

Para conocer el comportamiento de este diseño conviene graficar las aberraciones cromática y esférica. La gráfica de aberración cromática longitudinal es la que se muestra en al figura 12.

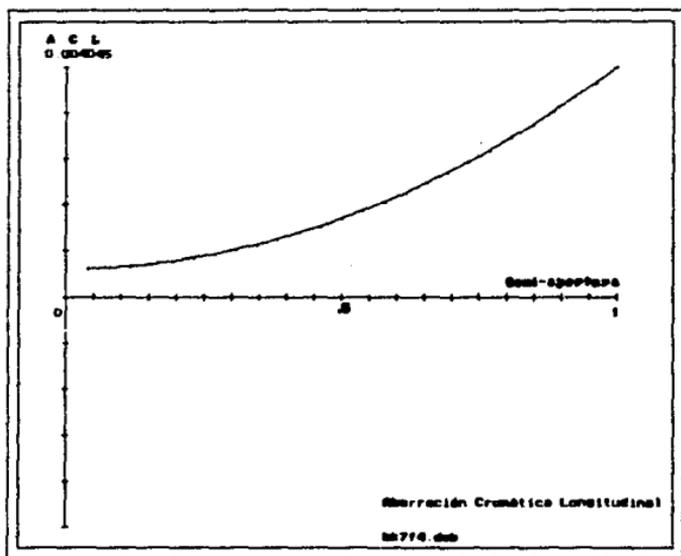


Figura 12
Aberración cromática longitudinal

En esta gráfica se observa que el valor máximo de la aberración cromática cae por debajo del valor permitido para la aberración esférica longitudinal, por lo tanto puede decirse que está satisfactoriamente corregida dentro de nuestros parámetros

Las gráficas de aberración esférica longitudinal y transversal se presentan en las figuras 13 y 14 respectivamente. Aunque en el despliegue de estas gráficas en la pantalla de la computadora cada curva aparece con el color de la longitud de onda correspondiente a cada una, en estas gráficas en blanco y negro podemos distinguir a qué longitud de onda corresponden si se consideran las siguientes reglas:

1) La longitud de onda de referencia en el diseño es $\lambda = 587.5618 \times 10^{-7}$ cm, que corresponde a la longitud de onda de la línea espectral amarilla del helio, entonces, la curva más cercana al origen, será la de color amarillo.

2) La curva que sufra más desviaciones, siempre será la que corresponde a $\lambda = 486.1427 \times 10^{-7}$ cm, longitud de onda de la línea espectral azul del hidrógeno.

3) Por eliminación, la restante será la correspondiente a $\lambda = 626.2725 \times 10^{-7}$ cm, que es la longitud de onda de la línea espectral roja del hidrógeno.

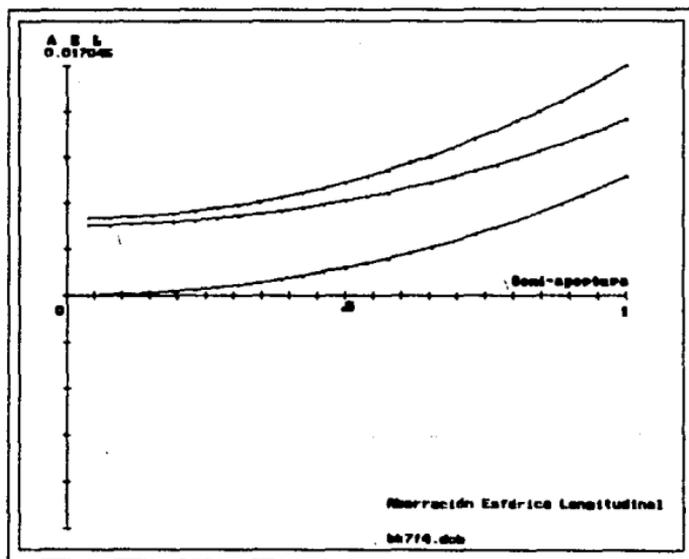


Figura 13
Aberración esférica longitudinal

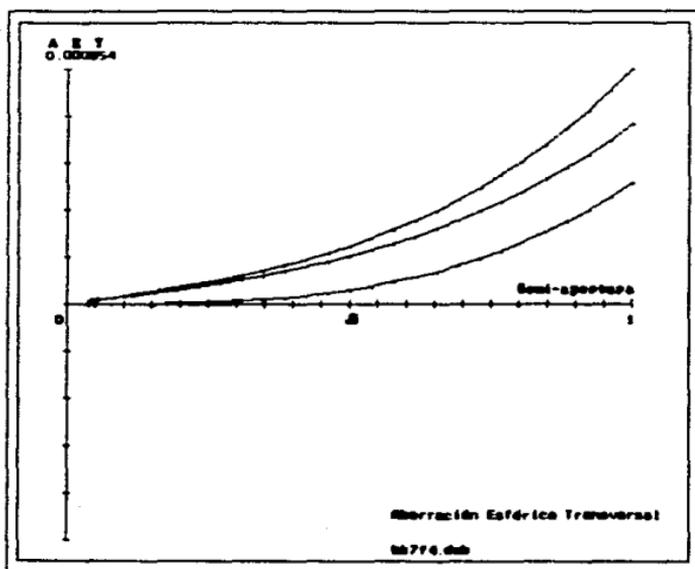


Figura 14
 Aberración esférica transversal

En la gráfica de aberración esférica longitudinal se observa que sería conveniente introducir un defoco positivo del orden de la mitad del máximo de la aberración obtenida para interceptar los rayos más cerca del eje óptico y así mejorar la calidad en la imagen. Con esta modificación, obtenemos las gráficas de las figuras 15 y 16.

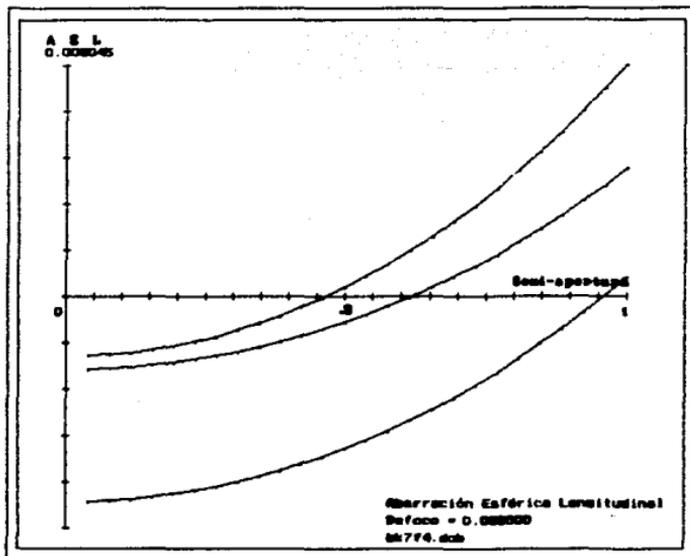


Figura 15
Aberración esférica longitudinal

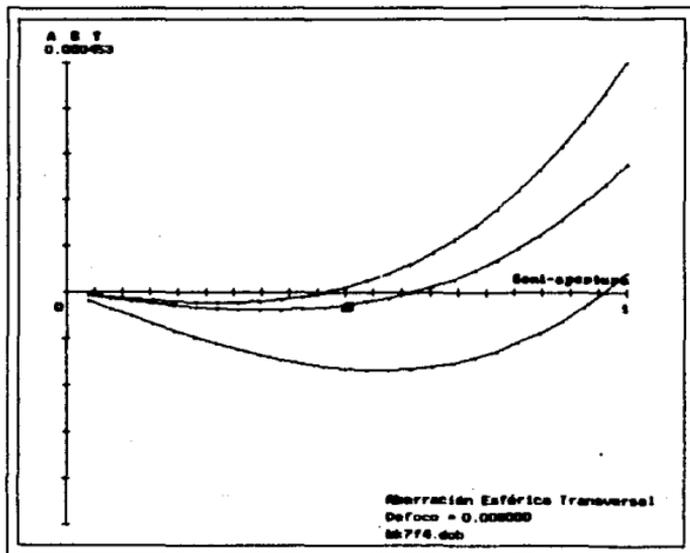


Figura 16
Aberración esférica transversal

La escala en las gráficas de aberración se ha reducido considerablemente. Analicemos estos resultados.

Es sabido que el límite de resolución de un sistema óptico está dado por el radio del disco de Airy. Tomando esta cantidad como parámetro de comparación en este diseño resulta el siguiente desarrollo.

El radio angular del disco de Airy esta dado por

$$\theta_A = 1.220 \lambda/D \quad (51)$$

Para la curva de mayor aberración, $\lambda = 486.1 \times 10^{-7}$ cm y en este caso $D = 1$ cm, por lo tanto,

$$\theta_A = 5.930 \times 10^{-6} \text{ rad} \quad (52)$$

y el radio lineal del disco dado por $r_A = \theta_A f$ es entonces

$$r_A = 5.930 \times 10^{-4} \text{ cm} \quad (53)$$

Comparando este valor con el máximo de la gráfica, obtenemos

$$\frac{r_{\max}}{r_A} = \frac{4.53 \times 10^{-4} \text{ cm}}{5.930 \times 10^{-4} \text{ cm}} < 1 \quad (54)$$

por lo que es sistema esta limitado por difracción para esta longitud de onda para objetos puntuales en eje. Las demás longitudes de onda son mayores que λ_f entonces sus correspondientes discos de Airy son también mayores, pero sus curvas de aberración en la gráfica están por debajo de la curva para λ_f por lo tanto el sistema está limitado por difracción en todas las longitudes de onda graficadas.

Se puede comprobar que los resultados de evaluación de Doblete son razonablemente comparables a los reportados por paquetes comerciales. Las diferencias se deben a que éstos utilizan coprocesador matemático, lo cual implica mayor precisión en los cálculos, y a que cada programa utiliza diferentes algoritmos para el cálculo de aberraciones. En la tabla

2 se presenta una evaluación de este diseño en el paquete SUPER-OSLO.

El significado de las siglas en el listado es el siguiente:

SRF es el número de la superficie

EFL es la distancia focal efectiva

FNB es el número f del sistema

SA3 es la aberración esférica a tercer orden

CM3 es la coma a tercer orden

AS3 es el astigmatismo

PZ3 es la curvatura de campo

DS3 es la distorsión

PAC es la aberración cromática paraxial

SM es la suma de las contribuciones de cada superficie convertidas a desplazamientos transversales

SRF	RADIUS	THICKNESS	APERTURE	GLASS	
1	--	--	0.500000 A	AIR	
2	4.569817	0.157000	0.500000 S	BK7	
3	-4.279410	0.200000	0.495182 S	F4	
4	-60.703233	10.000000	0.490852 S	AIR	
5	--	--	0.113554 S	AIR	
		EFL	FNB		
		10.003088	10.003088		
SRF	SA3	CM3	AS3	PZ3	DS3
1	--	--	--	--	--
2	-0.000147	-0.000013	-1.22887E-06	-1.86395E-06	-2.82673E-07
3	0.000195	-8.10619E-06	3.37035E-07	2.37751E-07	-2.38981E-08
4	-0.000054	9.25397E-06	-1.58879E-06	-1.57082E-07	2.99743E-07
SM	-0.000060	-0.000123	-0.000025	-0.000018	-6.83030E-08
		SRF	PAC		
		1	--		
		2	-0.000290		
		3	0.000584		
		4	-0.000295		
		SM	-0.000029		

Tabla 2

Evaluación del diseño BK7F4 con el paquete SUPER-OSLO

Si para alguna aplicación específica fuera necesario corregir estas aberraciones aún más, ya se tiene un buen prediseño como punto de partida para ser optimizado en un programa más poderoso que Doblete.

Para tratar de hacer un análisis cualitativo de la eficiencia del algoritmo de diseño se cambiaron uno a uno los datos de entrada, respetando las tolerancias, y se obtuvieron los resultados de la siguiente sección.

3.2.2 Análisis cualitativo de la eficiencia del programa Doblete

3.2.2.1 Primer caso: Cambio de vidrios

En el ejemplo anterior, la dispersión (o número de Abbe) de los vidrios elegidos es

$$V_b = 64.17 \text{ (BK7)}$$

$$V_b = 36.63 \text{ (F4)}$$

y resultó ser una selección razonablemente buena. Cambiemos primero el segundo vidrio, escogiendo uno de mayor dispersión y otro de menor:

$$\text{LaFN32 con } V_b = 49.71$$

$$\text{SF11 con } V_b = 25.76$$

Los archivos de salida reportan los datos de la tabla 3.

Distancia focal	Aberración esférica
$f' = 10.049873$	$LA' = -0.497931$
Radios de curvatura	
$r1 = 1.412861$	
$r2 = -2.797817$	
$r3 = 2.999419$	

Tabla 3A
Doblete BK7LaFN32

Distancia focal **Aberración esférica**
 $f' = 9.999579$ $LA' = 0.003659$

Radios de curvatura

$r1 = 6.054548$

$r2 = -6.475389$

$r3 = -13.917498$

Tabla 3B
Doblete BK7SF11

Como era de esperarse, para el caso de un vidrio con mayor dispersión se obtiene un diseño con menor corrección en la aberración esférica que incluso se sale de la tolerancia permitida y Doblete no encuentra una mejor solución. En el caso del vidrio SF11 con menor dispersión que F4, los resultados mejoran tanto en precisión de la distancia focal como en la aberración esférica longitudinal para el rayo marginal. Entonces convendría cambiar la elección de vidrios por BK7-SF11 en lugar de BK7-F4.

Cambiando ahora solamente el primer vidrio escogamos en sustitución

FK54 con $V = 90.7$

LLF2 con $V = 47.17$

Aquí, según los listados de las tablas 4A y 4B, la elección de un vidrio de dispersión elevada para el primer elemento, es la mejor y por lo tanto convendría condiderar los vidrios FK54-F4.

Distancia focal **Aberración esférica**
 $f' = 9.999412$ $LA' = 0.002229$

Radios de curvatura

$r1 = 5.559007$

$r2 = -4.983729$

$r3 = -10.646672$

Tabla 4A
Doblete FK54F4

Distancia focal **Aberración esférica**
 $f' = 10.046440$ $LA' = -0.234246$

Radios de curvatura

$r1 = 1.812868$
 $r2 = -2.398802$
 $r3 = 3.378030$

Tabla 4B
Doblete LLF2F4

Si ahora se combinan los dos vidrios que arrojaron los mejores resultados obtiene el mejor diseño entre todos los anteriores. Estos vidrios son FK54 para el primer elemento y LAFN32 para el segundo. Los resultados están en la tabla 5

Distancia focal **Aberración esférica**
 $f' = 10.049873$ $LA' = -0.497931$

Radios de curvatura

$r1 = 5.442628$
 $r2 = -7.614361$
 $r3 = -11.931903$

Tabla 5
Doblete FK54SF11

De estos resultados es posible concluir que el algoritmo de diseño programado obtiene mejores resultados mientras mayor sea el valor absoluto de la diferencia de las dispersiones de los vidrios involucrados en el diseño.

3.2.2.2 Segundo caso: Cambio en la distancia focal

Empecemos por aumentar la distancia focal en un 15 cm y 20 cm y después hagamos reducciones a 5 cm y 3 cm. En los listados de los diseños podemos observar que al ir incrementando la distancia focal, se obtienen progresivamente mejores resultados (ver tabla 6). En cambio, si pedimos

dobletes con distancias focales pequeñas, (ver tabla7) el algoritmo encuentra soluciones con desviación razonable en la distancia focal pero aberración esférica considerable. En el caso de distancias focales -números f en general- cada vez más cercanas a 1 o menores, este proceso de diseño falla completamente. En general, si deseamos encontrar dobletes cementados con número f cercano a 1 debemos optar por algún otro método de diseño y generalmente por sistemas ópticos con más grados de libertad que un doblete cementado. Esto mismo se aplica al valor absoluto de distancias focales negativas.

Distancia focal	Aberración esférica
$f' = 20.001533$	$LA' = 0.004636$
Distancia focal	Aberración esférica
$f' = 15.002045$	$LA' = 0.006061$

Tabla 6
Dobletes con $\#f > 10$

Distancia focal	Aberración esférica
$f' = 5.006086$	$LA' = 0.016323$
Distancia focal	Aberración esférica
$f' = 3.042437$	$LA' = 0.032589$

Tabla 7
Dobletes con $\#f < 10$

3.2.2.3 Tercer caso: Cambio en los grosores

Como puede verse en los datos de la tabla 8, en este caso el algoritmo no es tan sensible a variaciones en el grosor del segundo elemento, sin embargo, los cambios en el grosor del primer elemento sí afectan de manera considerable el valor para el radio de la tercera superficie.

Distancia focal	Aberración esférica
$f' = 10.004775$	$LA' = 0.008446$
Radios de curvatura	Grosos
$r1 = 4.527753$	$da\ orilla = 0.100$
$r2 = -4.279410$	$da\ centro = 0.157$
$r3 = -68.417461$	$db\ centro = 0.100$
Distancia focal	Aberración esférica
$f' = 10.020904$	$LA' = 0.006243$
Radios de curvatura	Grosos
$r1 = 4.388731$	$da\ orilla = 0.200$
$r2 = -4.279410$	$da\ centro = 0.258$
$r3 = -122.161043$	$db\ centro = 0.200$
Distancia focal	Aberración esférica
$f' = 10.030446$	$LA' = 0.009400$
Radios de curvatura	Grosos
$r1 = 4.642377$	$da\ orilla = 0.100$
$r2 = -4.279410$	$da\ centro = 0.156$
$r3 = -52.463978$	$db\ centro = 0.300$
Distancia focal	Aberración esférica
$f' = 10.006550$	$LA' = 0.009000$
Radios de curvatura	Grosos
$r1 = 4.167917$	$da\ orilla = 0.300$
$r2 = -4.279410$	$da\ centro = 0.359$
$r3 = 414.051471$	$db\ centro = 0.200$

Tabla 8

Cambio en el grosor de cada elemento del doblete

3.2.2.4 Cuarto caso. Cambio en la distancia objeto

El programa doblete también funciona para diseñar dobletes con distancia objeto finita. En general, puede decirse que el programa ha sido capaz de encontrar soluciones en todas las pruebas realizadas.

3.3 CONCLUSIONES

Para este algoritmo los resultados mejoran al seleccionar vidrios de alta dispersión para el primer elemento y baja dispersión para el segundo. Una correcta elección de los vidrios puede ser un factor determinante para encontrar una buena solución o no. La elección del grosor de cada elemento no es de gran relevancia en el algoritmo, pero deben elegirse grosores que faciliten la construcción del doblete en el Taller Óptico. En términos generales puede decirse que el parámetro crucial para el diseño de dobletes usando este algoritmo es el número f . Para dobletes con valor absoluto del número f mayor que 7 se obtienen resultados que caen dentro de la tolerancia para la distancia focal. Para el caso de corrección de la aberración esférica, el método de diseño resulta un poco limitado ya que los diseños de salida no siempre cumplen con las tolerancias deseadas. La corrección de la aberración cromática queda siempre determinada por la eficiencia de corrección del método D-d. En general basta una iteración global para encontrar el mejor resultado de este algoritmo.

Para dobletes rápidos con valor absoluto del número f menor que 7 el algoritmo va perdiendo precisión hasta fallar completamente al acercarse a 1.

CONCLUSIONES GENERALES

El algoritmo de diseño de dobletes acromáticos presentado por R. Kingslake¹⁶, basado en los métodos algebraicos tradicionales de Diseño Óptico fue programado en el Programa Doblete. Se utilizó lenguaje el lenguaje de programación Turbo Pascal V 5.0. Este programa además de contener el algoritmo de diseño con algunas modificaciones para mejorar su eficiencia, tiene opciones de graficación, trazo de rayos y manejo de archivos de datos. La estructura modular del programa Doblete facilita, al usuario interesado, la modificación o complementación del programa y la utilización de las unidades programadas para otros fines de diseño óptico, que en sí conforman una librería básica para el Diseño Óptico asistido por computadora.

El algoritmo de diseño programado muestra la capacidad de encontrar buenas soluciones. Sin embargo, la habilidad y experiencia del usuario para su uso es importante. En el caso de que se necesite algún diseño con mejor corrección de aberraciones que las que realiza el programa Doblete, puede utilizarse un programa más poderoso, con la gran ventaja de que ya se tiene un prediseño inicial, como es necesario para poder acceder este tipo de programas.

¹⁶ Ref [7]

Para incrementar la versatilidad del programa en el diseño de dobletes, sería deseable desarrollar los siguientes puntos:

i) Estudiar el algoritmo de corrección de la aberración esférica. Es conveniente encontrar un criterio para modificar las curvaturas en incrementos pequeños que sirvan para la corrección de esta aberración y el diámetro efectivo del sistema no se vea afectado de manera considerable.

ii) Para una mejor corrección de la aberración cromática se propone la corrección por el método D-d a diferentes alturas y no sólo en el borde de la apertura del sistema.

iii) Estudiar la posible traducción de estas ecuaciones de diseño de lentes delgadas en ecuaciones para lentes gruesas y utilizar el mismo algoritmo de diseño. Es posible que así se obtengan mejores resultados durante la optimización.

iv) Estudiar la generalización de estos algoritmos para el caso de dobletes con espaciamento de aire ya que en este caso se tienen más grados de libertad para la corrección del diseño.

v) Analizar cuál sería la forma de estas correcciones para el caso de objetos fuera de eje y consideración de las demás aberraciones de tercer orden.

Como comentario final, cabe señalar que este tipo de trabajo de análisis y programación de ecuaciones algebraicas tradicionales ahorra gran cantidad de trabajo de escritorio al diseñador óptico que lo utiliza y se adquiere gran experiencia en el diseño más eficientemente que utilizando directamente con grandes paquetes profesionales comerciales de Diseño Óptico.

BIBLIOGRAFIA

1. Conrady, A.D., Applied Optics and Optical Design, Volumen I, Dover Publications, EUA, 1960, pp 204-245.
2. Conrady, A.D., Applied Optics and Optical Design, Volumen II, Dover Publications, EUA, 1957, pp 640-661.
3. Feder, D.P., Automatic Optical Design, Appl.Opt 2 1209-1226 (1963)
4. Feder, D.P., Optical Calculations with Automatic Computing Machinery, J.Opt.Soc.Am. 41 630-635 (1951)
5. Hopkins, R.E., Korones, H.D., Some Effects of Glass Choice in Telescope Doublets J.Opt.Soc.Am. 49 869-871 (1959)
6. Jenkins y White, Fundamentals of Optics, cuarta edición, Mc Graw Hill, Nueva York, 1981, pp 149-182.
7. Kingslake R. Lens Design Fundamentals Academic Press, Nueva York, 1978.
8. Malcara, D., Óptica Básica primera edición, Fondo de Cultura Económica, México, 1989, pp 115-154.
9. Borland International, Turbo Pascal Reference Guide Version 5.0, EUA, 1988.
10. Imperial College, Notas del curso de Diseño Óptico, Londres Inglaterra, 1989.

APENDICE

A continuación se presentan el listado de programa Doblete y los listados de las unidades programadas que utiliza Doblete. Para facilitar su uso se anexa el siguiente índice:

Programa o Unidad	Página
Doblete	A1
Unidad de diseño de dobletes acromáticos	A4
Unidad de declaración de variables globales	A6
Unidad de presentación del programa	A8
Unidad para lectura de archivos de datos	A9
Unidad para la graficación de aberración esférica	A12
Unidad para la graficación de la aberración cromática longitudinal	A14
Unidad para el dibujo del doblete	A16
Unidad de cálculo y despliegue de constantes paraxiales	A18
Unidad para la tabla de datos marginales	A19
Unidad para lectura de datos de entrada	A20
Unidad de cálculo de c_2 usando la suma de las G_i	A23
Unidad de trazado marginal y cálculo de c_3	A26
Unidad de cálculo y corrección de la distancia focal	A28
Unidad de cálculo de la distancia imagen paraxial	A29
Unidad de evaluación de tolerancias	A30
Unidad de evaluación y corrección de la aberración esférica	A31
Unidad para guardar los datos en un archivo	A32
Unidad de declaración de funciones	A34
Unidad de cálculo de la intersección de la superficie con el eje óptico	A35
Unidad de utilerías para graficación	A36
Unidad de graficación de la aberración esférica	A40

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

Unidad de trazo exacto para la graficación de aberración cromática	A42
Unidad de trazo exacto para la tabla de rayos marginales	A44
Unidad de trazo meridional	A48
Unidad del método D - d	A50
Unidad de trazo paraxial	A51

Programa para el Diseño de Dobletes Acromaticos

Program Doblete;

uses

```

crt,
declara,           (Unidad de declaracion de variables globales)
dpresent,         (Unidad de presentacion del programa )
diseno,           (Unidad de diseño de dobles )
dleearch,         (Unidad de lectura de archivos )
dgesferi,         (Unidad para la graficacion de ab esferica )
dgcromat,         (Unidad para la graficacion de ab cromatica )
dgbujo,           (Unidad para el dibujo del doblete )
dtabmar,         (Unidad para la tabla de trazo marginal )
dtabpar;         (Unidad para la tabla de valores paraxiales )

```

const

```

C = 'C';           (constantes para identificar opciones )
L = 'L';
S = 'S';
E = 'E';
P = 'P';
T = 'T';
V = 'V';

```

var

```
opcion : char;
```

procedure menu;

(procedimiento para despliegue de los menues)

begin

(**procedura** menu)

```
textbackground(1);
```

```
textcolor(15);
```

repeat

```
clrscr;
```

```
writeln;
```

```
writeln;
```

```
writeln('
```

Indique la opcion que desea');

```
);
```

```
writeln;
```

```
writeln;
```

```
writeln('
```

(C)rear un nuevo archivo de datos');

```
);
```

```
writeln;
```

```
writeln;
```

```
writeln('
```

(L)eer un archivo de datos ya creado');

```
);
```

```
writeln;
```

```

writeln;
writeln;
writeln(' (S)alir del programa Doblete');
readln(opcion);
case opcion of
  'c',C : disena(fpdeseada,fpconstrada,fpptolerancia,LPdeseada,
                LPconstrada,LPptolerancia,naC,naD,naF,nbC,nbD,
                nbF,da,db,dacentro,D,via,lp.ca,cb,c1,c2,c3,a,b,
                h12,h21,fa,fb,lppara,iteracion_a,
                iteracion_b,maxiteracion_a,imaginaria,repite1,
                repite2,nombre);
  'l',L : leearchivo(fpdeseada,fpconstrada,fpptolerancia,
                    LPdeseada,LPconstrada,LPptolerancia,c1,c2,
                    c3,naC,naD,naF,nbC,nbD,nbF,da,dacentro,db,D,
                    via,iteracion_a,imaginaria,nombre,1);
  's',S : begin
            textbackground(0);
            clrscr;
            exit
          end;
end;
until ( upcase(opcion) in [C,L,S] ) ;
repeat
  textbackground(3);
  clrscr;
  textcolor(15);
  writeln;
  writeln;
  writeln('          Indique la opcion que desea');
  writeln;
  writeln;
  writeln(' (L)istar los datos del doblete');
  writeln;
  writeln;
  writeln(' Graficar Aberracion (E)sferica');
  writeln;
  writeln;
  writeln(' Graficar Aberracion (C)romatica');
  writeln;
  writeln;
  writeln(' (P)intar el doblete');
  writeln;
  writeln;
  writeln(' (T)abla de trazo marginal');
  writeln;
  writeln;
  writeln(' (V)alores paraxiales');
  writeln;
  writeln;
  writeln(' (S)alir');
  writeln;
  readln(opcion);
case opcion of

```

```

'l',L : leearchivo(fpdeseada,fpencontrada,fptolerancia,
                 LAPdeseada,LApconcontrada,LApptolerancia,c1,c2,
                 c3,naC,naD,naF,nbC,nbD,nbF,da,dacentro,db,D,
                 via,iteracion_a,imaginaria,nombre,2);
'e',E : esferica(naC,nbC,naD,nbD,naF,nbF,da,db,via,D,c1,c2,c3,
                 Lp,dacentro,imaginaria,nombre);
'c',C : cromatica(naC,nbC,naD,nbD,naF,nbF,da,db,via,D,c1,c2,c3,
                 Lp,dacentro,imaginaria,nombre);
'p',P : dibuja(c1,c2,c3,escala,D,dacentro,db,nombre);
't',T : tablamarginal(naC,nbC,naD,nbD,naF,nbF,da,db,via,D,c1,c2,
                    c3,Lp,dacentro,imaginaria);
'v',V : tablaparaxial(1,naD,naD,nbD,nbD,1,dacentro,db,c1,c2,c3,
                    D,fpencontrada);
's',S : menu;
end;
until (upcase(opcion) in [S]);
end;
begin
presentacion;
menu;
end.

```

Unidad de Diseño de Dobletes Acromaticos
--

```

}
Unit Diseño;

```

```

interface

```

```

uses

```

```

  Declara,
  Datos,
  DsumaGc2,
  Dmargina,
  Dfocal,
  Dlparax,
  Dtolera,
  Desferic,
  Dguarda;
  {Unidad de lectura de datos de entrada }
  {U de calculo de la condicion de suma de G's }
  {Unidad de trazo del rayo marginal }
  {U de calculo de la distancia focal obtenida }
  {U de calculo de l' paraxial }
  {U de calculo de tolerancias en el diseno }
  {Unidad de calculo de la aberracion esferica }
  {Unidad para archivar datos }

```

```

procedure disena(var fpdeseada,fpconcontrada,ftpolerancia,LAPdeseada,
  LAPconcontrada,LAPtolerancia,naC,naD,naF,nbC,nbD,nbF,da,
  db,dacentro,D,via,Lp,ca,cb,c1,c2,c3,a,b,h12,h21,fa,fb,
  lparax : real; var iteracion_a,iteracion_b,
  maxiteracion_a : integer; imaginaria,repitel,repite2 :
  boolean; var nombre : string);
  {algoritmo de diseño de R. Kingslake }

```

```

implementation

```

```

procedure disena:

```

```

  begin
    {procedure disena }
    leedatos(fpdeseada,ftpolerancia,LAPdeseada,LAPtolerancia,naC,naD,
      naF,nbC,nbD,nbF,da,db,dacentro,D,via,maxiteracion_a,nombre,
      imaginaria);
    imaginaria := false;
    calcula_c1_c2(naC,naD,naF,nbC,nbD,nbF,via,fpdeseada,ca,cb,c1,c2,a,b,
      imaginaria);
    repitel := true;
    repite2 := true;
    iteracion_a := 0;
    while ((repitel=true) and (imaginaria=false) and
      (iteracion_a<maxiteracion_a)) do
      begin
        iteracion_a := iteracion_a + 1;
        iteracion_b := 0;
        while (repite2 = true) and (imaginaria = false) and
          (iteracion_b<10) do

```

```

begin
  iteracion_b := iteracion_b + 1;
  rayomarginal(naC,naD,naF,nbC,nbD,nbF,da,db,via,D,
               c1,c2,c3,Lp,dacentro,imaginaria);
  If imaginaria = false then
    begin
      fpencontrada := efl(1,naD,naD,nbD,nbD,1,dacentro,
                          db,c1,c2,c3,a11,a12,a22);
      tolerancia1(fpdeseada,fpencontrada,ftolerancia,
                  repite2);
      If repite2 = true then
        nueva_c1(c2,fpencontrada,fpdeseada,ca,c1);
      end
    end;
  If imaginaria = false then
    begin
      lpparax := lppar(dacentro,db,D,c1,c2,c3,naD,nbD,via,0);
      LAPencontrada := LAP(Lp,lpparax);
      tolerancia2(LApdeseada,LAPencontrada,LApTolerancia,
                  repite1);
      If repite1 = true then
        begin
          nueva_c2(ca,D,a,b,LApdeseada,LAPencontrada,
                  lpparax,c1,c2);
          repite2 := true;
        end;
      end;
    end;
  end;
  guarda(fpdeseada,fpencontrada,ftolerancia,LApdeseada,LAPencontrada,
          LAptolerancia,c1,c2,c3,naC,naD,naF,nbC,nbD,nbF,da,dacentro,
          db,D,via,iteracion_a,nombre);
end;
      (procedure disena
      )
end.
      (unit diseno
      )

```

Unidad de declaracion de variables globales

```
unit declara;
```

```
interface
```

```
const
  midX      = 259; (coordenada central de la pantalla en X   )
  midY      = 239; (coordenada central de la pantalla en Y   )
  maxX      = 639; (coordenada maxima de la pantalla en X    )
  maxY      = 479; (coordenada maxima de la pantalla en Y    )
  num       = 26;  (numero de puntos a graficar en cada curva )
  escala    = 100; (escala del dibujo del doblete            )
```

```
type
```

```
  rayo = array [1..13] of real;
           (datos del trazo de un rayo en una interfase      )
  aberracion = array [1..num] of real;
           (valor de la aberracion para cada rayo           )
```

```
var
```

```
  iteracion_a,      (contador del numero de iteraciones en el )
                    (ciclo a del algoritmo de diseno          )
  iteracion_b,      (contador del numero de iteraciones en el )
                    (ciclo b del algoritmo de diseno          )
  maxiteracion_a    (numero maximo deseado de iteraciones en el )
                    (ciclo a                                  )
                    : integer;
  fa,                (longitud focal del elemento a del doblete )
  fb,                (longitud focal del elemento b del doblete )
  fpdeseada,         (longitud focal deseada para el doblete    )
  fpencontrada,      (longitud focal obtenida                    )
  fptolerancia,      (tolerancia en la discrepancia entre las dos )
                    (anteriores                                  )
  h12,               (segundo punto nodal del elemento a del doblete)
  h21,               (primer punto nodal del elemento b del doblete )
  LApdeseada,        (aberracion esferica deseada para el doblete )
  LAPencontrada,     (aberracion esferica obtenida                    )
  LAptolerancia,     (tolerancia en la discrepancia entre las dos )
                    (anteriores                                  )
  naC, naD, naF,     (indices de refraccion de la lente a en las )
                    (lineas C, D y F respectivamente          )
  nbC, nbD, nbF,     (indices de refraccion de la lente b en las )
                    (lineas C, D y F respectivamente          )
  ca, cb,            (curvaturas de las lentes a y b respectivamente)
  da,                (grosor de la orilla para el elemento positivo )
  db,                (grosor del centro para el elemento negativo )
```

```

dacentro,          (grosor del centro para el elemento positivo )
c1, c2, c3,       (curvaturas de las superficies 1, 2 y 3 respec)
a,b,              (coeficientes a y b de la expresion cuadratica)
                  (para la aberracion esferica )
l,               (distancia objeto )
via,             (inverso de la distancia objeto )
D,              (diametro de la lente )
Lp,             (L prima del rayo marginal )
lpparax,        (l prima paraxial )
a11,a12,a22      : real;
imaginaria,      (bandera que indica si el argumento de la raiz)
                  (cuadrada es negativo )
repite1,         (bandera que indica si debe repetirse el ciclo)
                  (a del programa)
repite2          (bandera que indica si debe repetirse el ciclo)
                  (b del programa)
                  : boolean;
nombre           (nom del archivo de datos de entrada y salida )
                  : string;
implementation
end.             (unit declara )

```

Unidad de presentacion del programa

```

unit dpresenta:

interface
uses crt, dos;

procedure espera(x,y : integer);           (detiene la corrida hasta que )
                                           (se pulsa una tecla )
procedure presentacion;                   (caratula de presentacion del )
                                           (programa )

implementation

procedure espera:
var
  ch : char;
begin
  gotoxy(x,y);                             (procedure espera )
  write('Presione cualquier tecla...');
  writeln;
  ch := readkey;                             (procedure espera )
end;

procedure presentacion:
begin
  (procedure presentacion )
  textbackground(1);
  clrscr;
  textcolor(15);
  gotoxy(15,10);
  write('Programa de Diseño de Dobletes Acromaticos');
  gotoxy(25,15);
  write('Por Citlali Lopez-Ortiz');
  espera(35,20);
end;
end.
(unit presenta )

```

Unidad para lectura de los archivos de datos.

```

}
unit dleearch;

interface
uses printer;

type
  datos = real;
  archivo = file of datos;

var
  impresion : char;
  i : integer;
  data : array [1..21] of datos;
  f : archivo;

procedure leearchivo(var fpdeseada,fpconcontrada,fpptolerancia,LAPdeseada,
  LAPencontrada,LAPTolerancia,c1,c2,c3,naC,naD,naF,
  nbC,nbD,nbF,da,dacentro,db,D,via : real; var
  iteracion_a : integer; var imaginaria : boolean;
  var nombre : string; indicador : word);
  (lee los datos del diseño del )
  (archivo deseado )

implementation

procedure leearchivo;

begin
  (procedure leearchivo )
  if indicador = 1 then
    begin
      writeln(' Escriba el nombre del archivo que desea usar');
      readln(nombre)
    end;
    assign(f,nombre);
    reset(f);
    for i := 1 to 21 do
      read(f,data[i]);
    close(f);
    writeln(' ',nombre);
    writeln(' fpdeseada = ',data[1]:1:6);
    writeln(' fpconcontrada = ',data[2]:1:6);
    writeln(' fpptolerancia = ',data[3]:1:3);
    writeln(' LAPdeseada = ',data[4]:1:6);
    writeln(' LAPencontrada = ',data[5]:1:6);
    writeln(' LAPTolerancia = ',data[6]:1:3);
    writeln(' r1 = ',data[7]:1:6);
  end;
end;

```

```

writeln('      r2 = ',data[6]:1:6);
writeln('      r3 = ',data[9]:1:6);
writeln('      naC = ',data[10]:1:6);
writeln('      naD = ',data[11]:1:6);
writeln('      naF = ',data[12]:1:6);
writeln('      nbC = ',data[13]:1:6);
writeln('      nbD = ',data[14]:1:6);
writeln('      nbF = ',data[15]:1:6);
writeln('      da = ',data[16]:1:3);
writeln('      dacentro = ',data[17]:1:3);
writeln('      db = ',data[18]:1:3);
writeln('      D = ',data[19]:1:2);
writeln('      via = ',data[20]:1:3);
writeln('      iteracion_e = ',data[21]:1:0);
fpdeseada := data[1];
fpconcontrada := data[2];
fptolerancia := data[3];
LApdeseada := data[4];
LApconcontrada := data[5];
Laptolerancia := data[6];
c1 := 1/data[7];
c2 := 1/data[8];
c3 := 1/data[9];
naC := data[10];
naD := data[11];
naF := data[12];
nbC := data[13];
nbD := data[14];
nbF := data[15];
da := data[16];
dacentro := data[17];
db := data[18];
D := data[19];
via := data[20];
iteracion_a := round(data[21]);
imaginaria := false;
readln;
writeln('Desea esta informacion impresa? (S/N)');
readln(impression);
if upcase(impression) = 'S' then
begin
    (if
        writeln(lst,'      ',nombre);
        writeln(lst);
        writeln(lst,'      fpdeseada = ',data[1]:1:6);
        writeln(lst,'      fpconcontrada = ',data[2]:1:6);
        writeln(lst,'      fptolerancia = ',data[3]:1:3);
        writeln(lst,'      LAPdeseada = ',data[4]:1:6);
        writeln(lst,'      LAPconcontrada = ',data[5]:1:6);
        writeln(lst,'      Laptolerancia = ',data[6]:1:3);
        writeln(lst,'      r1 = ',data[7]:1:6);
        writeln(lst,'      r2 = ',data[8]:1:6);
        writeln(lst,'      r3 = ',data[9]:1:6);
        writeln(lst,'      naC = ',data[10]:1:6);
    )
end;

```

```

        writeln(1st,'
        naD = ',data[11]::1:6);
        naF = ',data[12]::1:6);
        nbC = ',data[13]::1:6);
        nbD = ',data[14]::1:6);
        nbF = ',data[15]::1:6);
        da = ',data[16]::1:6);
        dacentro = ',data[17]::1:3);
        db = ',data[18]::1:3);
        D = ',data[19]::1:2);
        via = ',data[20]::1:3);
        iteracion_a = ',data[21]::1:0);
    end;
end.
        (if
        (procedure leearchivo
        )
        (unit dleearch
        )

```

Unidad para la graficacion de la aberracion esferica
--

```
unit dgesferi;
```

```
interface
```

```
uses declara,dfuncion,dcal_x,dlpparak,graph,dgrafica,dgesferi;
```

```
var
```

```

X1,           (sagita de la primera superficie )
X2,           (sagita de la segunda superficie )
lpparak,      (l' )
defoco,       (defoco para la grafica )
aux,          (variable auxiliar )
max,          (valor maximo de aberraciones )
maxzC,        (valor maximo ab transversal en C )
maxzD,        (valor maximo ab transversal en D )
maxzF,        (valor maximo ab transversal en F )
maxC,         (valor maximo ab longitudinal en C)
maxD,         (valor maximo ab longitudinal en D)
maxF,         (valor maximo ab longitudinal en F)

              : real;

x,            (altura de inc en la primera sup )
yC,           (aberracion longitudinal en C )
yD,           (aberracion longitudinal en D )
yF,           (aberracion longitudinal en F )
zC,           (aberracion transversal en C )
zD,           (aberracion transversal en D )
zF,           (aberracion transversal en F )

              : aberracion;
opcion
              : char;
```

```

procedure esferica(naC,nbC,naD,nbD,naF,nbF,da,db,vla,D,c1,c2,c3 : real;
var Lp,dacentro : real; var imaginaria : boolean;
nombre : string);
```

```

procedure hazgrafica(maxC,maxD,maxF,defoco : real; x,yC,yD,yF :
aberracion; leyenda : string);
```

```
implementation
```

```
procedure esferica;
```

```
begin
```

```

repeat
  writeln(' Desea Aberracion-Esferica (T)ransversal o');
  writeln(' (L)ongitudinal?');

```

```
(procedure esferica
```

```
)
```

```

    readln(opcion);
until upcase(opcion) in ['T','L'];
calculaX(c1,D,X1);           {calcula sagita de ls sup 1 }
calculaX(c2,D,X2);           {calcula sagita de ls sup 2 }
dacentro := grosorcentro(da,X1,X2); {calcula grosor en el centro de a }
writeln(' Valor para el defoco ? ');
readln(defoco);              {pide defoco para la grafica }
lpparax := lppar(dacentro,db,D,c1,c2,c3,naD,nbD,via,defoco);(l'c/defoc)
modografico(0);              {inicializa modo grafico }
pintaorilla(7);              {pinta el marco }
pintaejes;                   {pinta los ejes }
grayoexacto(naD,nbD,dacentro,db,D,c1,c2,c3,lpparax,X1,Lp,maxD,maxzD,
    imaginaria,x,yD,zD);      {traza rayos y cal ab esfer en D }
grayoexacto(naC,nbC,dacentro,db,D,c1,c2,c3,lpparax,X1,Lp,maxC,maxzC,
    imaginaria,x,yC,zC);      {traza rayos y cal ab esfer en C }
grayoexacto(nF,nbF,dacentro,db,D,c1,c2,c3,lpparax,X1,Lp,maxF,maxzF,
    imaginaria,x,yF,zF);      {traza rayos y cal ab esfer en F }
if upcase(opcion) in ['L'] then {grafica la ab deseada }
    hazgrafica(maxC,maxD,maxF,defoco,x,yC,yD,yF,
        'Aberracion Esferica Longitudinal')
else
    hazgrafica(maxzC,maxzD,maxzF,defoco,x,zC,zD,zF,
        'Aberracion Esferica Transversal');
end;                           {procedure esferica }

procedure hazgrafica;

begin                           {procedure hazgrafica }
if maxF > maxD then              {encuentra el val max entre las }
    aux := maxF                  {aberraciones para C,D y F }
else
    aux := maxD;
if aux > maxC then
    max := aux
else
    max := maxC;
for i := 1 to num do            {grafica punto a punto la ab esf }
    grafica(x[i],yC[i],max,red,i);
for i := 1 to num do            {grafica punto a punto la ab esf }
    grafica(x[i],yD[i],max,yellow,i);
for i := 1 to num do            {grafica punto a punto la ab esf }
    grafica(x[i],yF[i],max,blue,i);
titulos(max,defoco,leyenda,nombre); {escribe los titulos en la grafica}
readln;
closegraph;                     {termina modo grafico }
restorecrtmode;                  {restablece modo texto }
end;                             {procedure hazgrafica }

end.                             {unit dgesferi }

```

Unidad para la graficacion de la aberracion cromatica longitudinal

```
unit dgcromat;
```

```
interfaca
```

```
uses declara,dfuncion,dcal_x,graph,dgrafica,dgcrom;
```

```
var
```

```

X1,                (sagita de la sup 1      )
X2,                (sagita de la sup 2      )
defoco             (variable auxiliar de defoco )
    : real;

x,                (altura del rayo en la sup 1 )
y,                (ab cromatica para cada x[i] )
yC,               (L' en C para cada x[i]    )
yF,               (L' en F para cada x[i]    )
    : aberracion;
```

```

procedure cromatica(naC,nbC,naD,nbD,naF,nbF,da,db,vla,D,c1,c2,c3 : real;
    var Lp,dacentro : real; var imaginaria : boolean;
    nombre : string);
```

```
    (grafica aberracion cromatica )
```

```
implementation
```

```
procedure cromatica;
```

```

begin
    modografico(0);           (procedure cromatica      )
    pintaorilla(7);         (inicializa modo grafico )
    pintaEjes;              (pinta el marco          )
    defoco := 0;            (pinta los ejes coordenados )
    calculaX(c1,D,X1);      (cal la sagita de la sup 1 )
    calculaX(c2,D,X2);      (cal la sagita de la sup 2 )
    dacentro := grosorcentro(da,X1,X2); (cal grosoren el centro de a )
    gcrayoexacto(naC,nbC,dacentro,db,D,c1,c2,c3,X1,Lp,imaginaria,x,yC);
    gcrayoexacto(naF,nbF,dacentro,db,D,c1,c2,c3,X1,Lp,imaginaria,x,yF);
    for i := 1 to num do   (calcula ab cromatica    )
        y[i] := yF[i] - yC[i];
    max := abs(y[num]);
    for i := 1 to num do   (encuentra el valor maximo de )
        if abs(y[i]) > max then (aberracion cromatica     )
            max := abs(y[i]);
    for i := 1 to num do   (manda raficar cada punto  )
        grafica(x[i],y[i],max,red,i);
    titulos(max,defoco,'Aberración Cromatica Longitudinal',nombre);
    readln;
```


Unidad para el dibujo del doblete

```

unit dgbujo;

interface
uses graph, declara.dgrafica;

procedure dibuja(c1,c2,c3,escala,D,dacentro,db : real;
titulo : string);
var
radio1,
radio2,
radio3,
x1,y1,
x2,y2,
x3,y3 : word;
arccoords : arccoordstype;
r1,
r2,
r3 : real;

implementation
procedure dibuja;
begin
modografico(0);
pintaorilla(7);
setcolor(7);
escala := D*maxY/2;
D := maxY/2;
dacentro := dacentro*escala;
db := db*escala;
r1 := 1/c1;
r2 := 1/c2;
r3 := 1/c3;
radio2 := round(abs(r2*escala));
x2 := round(maxX/2-db-radio2);
y2 := round(maxY/2);
radio1 := round(abs(r1*escala));
x1 := round(maxX/2-db-dacentro+radio1);
y1 := y2;
radio3 := round(abs(r3*escala));
x3 := round(maxX/2-radio3);
y3 := y2;
if radio1 > 2500 then

```

```

line(round(maxX/2-dacentro-db),round((maxY-D)/2),
      round(maxX/2-dacentro-db),round((maxY+D)/2)+10)
else
  begin
    if r1 > 0 then
      arc(x1,y1,90,270,radio1)
    else
      arc(x1,y1,270,90,radio1);
    end;
  if radio2 > 2500 then
    line(round(maxX/2-dacentro),round((maxY-D)/2),
          round(maxX/2-dacentro),round((maxY+D)/2)+10)
  else
    begin
      if r2 > 0 then
        arc(x2,y2,90,270,radio2)
      else
        arc(x2,y2,270,90,radio2);
      end;
    if radio3 > 2500 then
      line(round(maxX/2),round((maxY-D)/2),
            round(maxX/2),round((maxY+D)/2)+10)
    else
      begin
        if r2 > 0 then
          arc(x3,y3,90,270,radio3)
        else
          arc(x3,y3,270,90,radio3);
        end;
      setviewport(12,11,maxX-20,round((maxY-D)/2)+10,true);
      floodfill(15,15,11);
      setviewport(12,round((maxY+D)/2)+10,maxX-12,maxY-12,true);
      floodfill(15,15,11);
      outtextXY(440,maxY-round((maxY+D)/2)-50,titulo);
      readln;
      closegraph;
      restorecrtmode;
    end;
  end.

```

Unidad de calculo y despliegue de constantes paraxiales

```

unit dtabpar;

interface
uses crt,declara,dFocal;

var a11,
    a12,
    a22
    : real;

procedure tablparaxial(n11,nt1,n12,nt2,n13,nt3,dacentro,db,c1,c2,c3,D :
    real; var fpencontrada : real);

implementation

procedure tablparaxial:

begin
  fpencontrada := ef1(n11,nt1,n12,nt2,n13,nt3,dacentro,db,c1,c2,c3,
    a11,a12,a22);
  h12 := (1-a11)/(-a12);
  h21 := (a22-1)/(-a12);
  clrscr;
  gotoXY(20,5);
  write('Datos paraxiales');
  gotoXY(5,8);
  write('ef1 = ',fpencontrada:1:6);
  gotoXY(5,10);
  write('Primer punto nodal = ',h12:1:6);
  gotoXY(5,12);
  write('Segundo punto nodal = ',h21:1:6);
  gotoXY(5,14);
  write('Hf = ',fpencontrada/D:1:6);
  readln;
end;

end.

```

Unidad para la tabla de datos marginales

```

unit dtabmar;

interface
uses crt,printer,declara,dfuncion,dcal_X,dtabla;

var
  X1,                                (sagita de la superficie 1 )
  X2                                  (sagita de la superficie 2 )
  : real;
  trazo1C,                             (trazo en la sup 1 linea C )
  trazo2C,                             (trazo en la sup 2 linea C )
  trazo3C,                             (trazo en la sup 3 linea C )
  trazo1D,                             (trazo en la sup 1 linea D )
  trazo2D,                             (trazo en la sup 2 linea D )
  trazo3D,                             (trazo en la sup 3 linea D )
  trazo1F,                             (trazo en la sup 1 linea F )
  trazo2F,                             (trazo en la sup 2 linea F )
  trazo3F,                             (trazo en la sup 3 linea F )
  : rayo;

procedure tablamarginal(naC,nbC,naD,nbD,naF,nbF,da,db,via,D,c1,c2,c3 : real;
  var Lp,dacentro : real; var imaginaria : boolean);

implementation

procedure tablamarginal;

begin
  calculaX(c1,D,X1);                    {procedure tablamarginal }
  calculaX(c2,D,X2);                    {cal sagita de la superficie 1 }
  dacentro := grosorcentro(da,X1,X2);   {cal el grosor en el centro de a}
  clrscr;                                {limpia la pantalla }
  datosmarginal(naC,nbC,naD,nbD,naF,nbF,dacentro,db,via,D,c1,c2,c3,X1,Lp,
  imaginaria,trazo1C,trazo2C,trazo3C,trazo1D,trazo2D,
  trazo3D,trazo1F,trazo2F,trazo3F);     {calcula y despliega la tabla }
  readln;
end;                                     {procedure tablamarginal }

end.                                     {unit dtabmar }

```


implementation

procedure leedatos;

```

begin
    (procedure leedatos
    )
    textbackground(7);
    clrscr;
    textcolor(0);
    repeat
        writeln (' Desea abrir un nuevo archivo de datos de entrada? (S/N)');
        write (' ');
        readln (nuevo);
    until
        upcase(nuevo) in [S,N];
    if (nuevo = 's') or (nuevo = S) then
        begin
            writeln (' Escriba los siguientes datos');
            writeln (' Valor de la longitud focal del doblete');
            write (' ');
            readln (fpdeseada);
            writeln (' Tolerancia en la longitud focal');
            write (' ');
            readln (fptolerancia);
            writeln (' Valor de la aberracion esferica permitida');
            write (' ');
            readln (LApdeseada);
            writeln (' Tolerancia en la aberracion esferica permitida');
            write (' ');
            readln (LAptolerancia);
            writeln (' Valor de na en la linea C');
            write (' ');
            readln (naC);
            writeln (' Valor de na en la linea D');
            write (' ');
            readln (naD);
            writeln (' Valor de na en la linea F');
            write (' ');
            readln (naF);
            writeln (' Valor de nb en la linea C');
            write (' ');
            readln (nbC);
            writeln (' Valor de nb en la linea D');
            write (' ');
            readln (nbD);
            writeln (' Valor de nb en la linea F');
            write (' ');
            readln (nbF);
            writeln (' Valor del grosor de la lente positiva en la orilla');
            write (' ');
            readln (da);
            writeln (' Valor del grosor de la lente negativa en el centro');
            write (' ');
            readln (db);
        end
    end

```

```

writeln (' Valor del diametro libre de la lente');
write (' ');
readln (D);
repeat
  writeln (' ¿Es infinito la distancia del objeto');
  writeln (' a la primera superficie? (S/N)');
  write (' ');
  readln (infinito);
until
  upcase(infinito) in [S,N];
If (infinito = 's') or (infinito = S) then
  via := 0
else
  begin
    writeln (' Valor de la distancia objeto');
    write (' ');
    readln (l);
    via := 1/l;
  end;
writeln (' Numero maximo de iteraciones deseadas');
write (' ');
readln (maxiteracion_a);
guarda(fpdeseada,1,fp tolerancia,LApdeseada,1,LAptolerancia,1,1,1,
naC,naD,naF,nbC,nbD,nbF,da,1,db,D,via,maxiteracion_a,
nombre);
end
else
  begin
    leearchivo(fpdeseada,muda1,fp tolerancia,LApdeseada,muda1,
LAptolerancia,muda1,muda1,muda1,naC,naD,naF,nbC,nbD,
nbF,da,dacentro,db,D,via,maxiteracion_a,imaginaria,
nombre,1);
  end;
end;
nd. (procedure leedatos )
(unit datos )

```

Unidad de calculo de c2 usando las sumas de las G's

```

unit dsumaGc2;

interface
uses dpresent;

var
  Va,Vb : real;

function abbe(nC,nD,nF : real) : real;           {calcula el numero de abbe}
                                                {de los vidrios      }
procedure ca_cb(fpdeseada,Va,Vb,naC,naF,nbC,nbF : real; var ca,cb : real);
{calcula el valor de las }
{curvaturas ca y cb     }
procedure minesferica(naD,nbD,ca,cb,vla : real; var c1,c2,a,b : real; var
  imaginaria : boolean); {calcula c1 minimizando }
                          {aberracion esferica   }

const
  k = -1;           {constante auxiliar      }
  P = 'P';         {opcion de raiz positiva }
  N = 'N';         {opcion de raiz negativa }

var
  G1a,G2a,G3a,G4a,G5a,G6a,           {coef G del ele a del dobl}
  G1b,G2b,G3b,G4b,G5b,G6b,           {coef G del ele b del dobl}
  v1b,                                 {inver de dist obj para b }
  c,                                    {sol c1 con raiz positiva }
  c2pos,                                {sol c1 con raiz negativa}
  c2neg,
  signo : real;                        {op del signo de la raiz }
  : char;

procedure calcula_c1_c2(naC,naD,naF,nbC,nbD,nbF,vla,fpdeseada : real;
  var ca,cb,c1,c2,a,b : real;
  var imaginaria : boolean);
{cal c1 y c2 minimizando }
{aberracion esferica     }

implementation

function abbe;

```

```

begin                                     {function abbe           }
  abbe := (nD-1)/(nF-nC);                 {def de numero de abbe  }
end;                                       {function abbe           }

procedure ca_cb;

begin                                     {procedure ca_cb         }
  ca := 1/(fpdeседа*(Va-Vb)*(naF-naC));   {curvatura para a       }
  cb := 1/(fpdeседа*(Vb-Va)*(nbF-nbC));   {curvatura para b       }
end;                                       {procedure ca_cb         }

procedure minesferica;

procedure calcula_Gi(n : real; var G1,G2,G3,G4,G5,G6 : real);

begin                                     {procedure calcula_Gi    }
  G1 := (0.5)*n*n*(n-1);                  {valor para el coef G1  }
  G2 := (0.5)*(2*n+1)*(n-1);              {valor para el coef G2  }
  G3 := (0.5)*(3*n+1)*(n-1);              {valor para el coef G3  }
  G4 := (0.5)*(n+2)*(n-1)/n;              {valor para el coef G4  }
  G5 := (n*n-1)/(n*(0.5));                {valor para el coef G5  }
  G6 := (0.5)*(3*n+2)*(n-1)/n;            {valor para el coef G6  }
end;                                       {procedure calcula_Gi    }

begin                                     {procedure minesferica   }
  calcula_Gi(naD,G1a,G2a,G3a,G4a,G5a,G6a); {calculo de G's para a  }
  calcula_Gi(nbD,G1b,G2b,G3b,G4b,G5b,G6b); {calculo de G's para b  }
  vib := v1a + ca * (naD-1);
  a := (G4a*ca)+(G4b*cb);                  {calculo de coeficientes }
                                          {de la ecuacion cuadratica}
  b := (G2a*ca*ca-G2b*cb*cb-G5a*ca*v1b-G5b*cb*v1b);
                                          {para LAp'               }
  c := (G1a*ca*ca*ca)+(G1b*cb*cb*cb)-(G3a*ca*ca*v1b)+(G3b*cb*cb*v1b)+
      (G6a*ca*v1b*v1b)+(G6b*cb*v1b*v1b);

  If (b*b-4*a*c)<0 then                    {si el discriminante de }
  begin                                     {la ec es negativo ent...}
    writeln('PRECAUCION, raiz cuadrada imaginaria en suma Gs');
    c2 := (1.*b)/(2*a);                    {maximo de la parabola  }
    c1 := c2 + ca;                          {val corresp de c2      }
    writeln('El maximo de la parabola da c2 = ',c2:1:6);
    writeln('y por lo tanto c1 = ',c1:1:6);
    espera(24,48)
  end                                       {if                       }
  else                                     {if                       }
  begin                                     {else                     }
    c2pos := (1.*b + (sqrt(b*b-4*a*c)))/(2*a); {cal de las sol de la}
    c2neg := (1.*b - (sqrt(b*b-4*a*c)))/(2*a); {ec cuadratica        }
    writeln('La raiz positiva de suma Gs da c2 = ',c2pos:1:6,
      ' r2 = ',1/c2pos:1:6);
    writeln('La raiz negativa de suma Gs da c2 = ',c2neg:1:6,
      ' r2 = ',1/c2neg:1:6);
  end
end

```

```

repeat
  writeLn ('¿ Escoge ud. la raiz positiva (P) o la negativa (N) ?')
  readLn (signo);
until
  upcase (signo) in [P,N];
If (signo = 'p') or (signo = P) then
  c2 := c2pos
else
  c2 := c2neg;
  c1 := c2 + ca;
end
end;
(else
)
(procedure minesferica
)

procedure calcula_c1_c2;

begin
  Va := abbe(naC,naD,naF);
  Vb := abbe(nbC,nbD,nbF);
  ca_cb(fpdeseada,Va,Vb,naC,naF,nbC,nbF,ca,cb);(cal de curv de a y b )
  minesferica(naD,nbD,ca,cb,via,c1,c2,a,b,imaginaria);(cal de c1 y c2 )
end;
(procedure calucula_c1_c2)
(num de abbe para el a )
(num de abbe para el b )
(cal de curv de a y b )
(cal de c1 y c2 )
(procedure calucula_c1_c2)

end.
(unit dsumagc2
)

```

Unidad de trazo marginal y calculo de c3

```

unit dmargina;

interface
uses
  declara,dfuncion,dcal_x,dtrazmer,Dmenosd;

var
  X1,                                (sagita de la sup 1      )
  X2                                 (sagita de la sup 2      )
    : real;
  trazo1,                             (datos del trazo en la sup 1 )
  trazo2,                             (datos del trazo en la sup 2 )
  trazo3                              (datos del trazo en la sup 3 )
    : rayo;
procedure rayomarginal(naC,neD,naf,nbC,nbD,nbF,da,db,via,D : real;
  var c1,c2,c3,Lp,dacentro : real;
  var imaginaria : boolean);
  (trazo exacto del rayo marginal )

implementation

procedure rayomarginal;

begin
  calculax(c1,D,X1);                (procedure rayo marginal      )
  calculax(c2,D,X2);                (calcula la coord X para la sup1)
  dacentro := grosorcentro(da,X1,X2); (calcula la coord X para la sup2)
  trazo1[7] := arctan((D#via)/(2+(2#X1#via))); (calcula el grosor en el centro )
  trazo1[8] := sin(trazo1[7]);      (para el elemento a          )
  trazo1[9] := cos(trazo1[7]);      (calcula U para el rayo marginal)
  if via = 0 then                   (si los rayos son paralelos al )
    trazo1[1] := D/2;                (eje optico ent...           )
  else
    trazo1[1] := trazo1[8]/via;      (if                             )
  trazomeridional(c1,1,naD,trazo1,imaginaria); (calculo de D cuando los rayos )
  (no son paralelos al eje optico )
  (trazo del rayo atraves de la )
  (primera superficie            )
  if imaginaria = false then        (si no aparecio una raiz imag )
  begin
    trazo2[7] := trazo1[10];         (if                             )
    (actualizacion de U          )
  end
end

```

```

trazo2[8] := sin(trazo2[7]);
trazo2[9] := cos(trazo2[7]);
trazo2[1] := trazo1[2] - dacentro*(trazo2[8]);
trazomeridional(c2,naD,nbD,trazo2,imaginaria);
                                (transferencia a ka sig sup )
                                (trazo del rayo a traves de la )
                                (segunda superficie )
                                (if )
end;
If imaginaria = false then      (si no aparecio una raiz imag )
begin                             (if )
  D_d(naF,naC,nbF,nbC,da,dacentro,db,trazo1[9],trazo2[8],trazo2[9],
    trazo2[12],trazo2[13],c3);    (llamada a D_d para calcular c3 )
  trazo3[7] := trazo2[10];        (actualizacion de U )
  trazo3[8] := sin(trazo3[7]);
  trazo3[9] := cos(trazo3[7]);
  trazo3[1] := trazo2[2] - db*trazo3[8];
  trazomeridional(c3,nbD,1,trazo3,imaginaria)
                                (transferencia a ka sig sup )
                                (trazo del rayo a traves de la )
                                (segunda superficie )
                                (if )
end;
If imaginaria = false then
  Lp := trazo3[1]/sin(trazo3[10]);
end;
end.                               (procedure rayo marginal )
                                   (unit dmargin )

```


Unidad de evaluación de tolerancias

```

unit dtolera;

  interface

    procedure tolerancia1(deseado, encontrado, tolerancia : real;
      var repite : boolean);
      {checa si los valores obtenidos }
      {para la distancia focal estan }
      {dentro del rango deseado }
    procedure tolerancia2(deseado, encontrado, tolerancia : real;
      var repite : boolean);
      {checa si los valores obtenidos }
      {para la aberracion esferca }
      {están dentro del rango deseado }

  implementation

    procedure tolerancia1;
  begin
    if (abs(deseado-encontrado) > tolerancia) then {procedure tolerancia1 }
      repite := true
    else
      repite := false;
  end; {procedure tolerancia1 }

    procedure tolerancia2;
  begin
    if ((abs(deseado)+tolerancia) < abs(encontrado)) then {procedure tolerancia2 }
      repite := true
    else
      repite := false;
  end; {procedure tolerancia2 }

end. {unit dtolera }
  
```


Unidad para guardar los datos en un archivo

```
)
unit dguarda;
```

```
interface
```

```
type
```

```
  datos = real ;
  archivo = file of datos;
```

```
var
```

```
  data : array[1..21] of datos ;
  f : archivo;
  ix : integer;
```

```
procedure guarda(fpdeseada,fpconcontrada,fpptolerancia,LAPdeseada,
  LAPconcontrada,LAPptolerancia,c1,c2,c3,naC,naD,naF,nbC,nbD,
  nbF,da,dacentro,db,D,via,iteracion_a : real; var nombre :
  string);
```

```
      (guarda los datos del diseño )
      (en un arreglo y lo archiva )
```

```
implementation
```

```
procedure guarda;
```

```
begin
```

```
  data[1] := fpdeseada;
  data[2] := fpconcontrada;
  data[3] := fpptolerancia;
  data[4] := LAPdeseada;
  data[5] := LAPconcontrada;
  data[6] := LAPptolerancia;
  data[7] := 1/c1;
  data[8] := 1/c2;
  data[9] := 1/c3;
  data[10] := naC;
  data[11] := naD;
  data[12] := naF;
  data[13] := nbC;
  data[14] := nbD;
  data[15] := nbF;
  data[16] := da;
  data[17] := dacentro;
  data[18] := db;
  data[19] := D;
  data[20] := via;
  data[21] := iteracion_a;
  writeIn(' Escribe el nombre del archivo para estos datos');
```

```
(procedure guarda )
```

```

writein(' ');
readin(nombre);
{!-}
assign(f,nombre);
if (IOResult <> 0) then
  write(' Error al asignar archivo ');
rewrite(f);
if (IOResult <> 0) then
  write(' Error al abrir archivo ');
for ix := 1 to 21 do
  write(f,data[ix]);
close(f);
if (IOResult <> 0) then
  write(' Error al cerrar archivo ');
end;
      (procedure guarda
end.      (unit dguarda

```

Unidad de declaración de funciones

```

unit dfuncion;

interface
function grosorcentro(D,X1,X2 : real) : real;      (calcula el grosor cen- )
function angulo(senangulo : real) : real;         (tral de la lente   )
                                                    (calcula el seno inverso)

var
  cosangulo : real;

implementation
function grosorcentro;
begin
  grosorcentro := D + X1 - X2;
end;
function angulo;
begin
  if (senangulo < -1) or (senangulo > 1) then (function angulo      )
  begin
    writeln ('Error en funcion angulo');
    exit
  end
  else
  begin
    cosangulo := sqrt (1 - sqr(senangulo));
    if 0 <= cosangulo (checa el signo de cos )
    then (para asingular correcc-)
    angulo := arctan(senangulo/cosangulo)
    else (tamente a arctan      )
    angulo := -arctan(senangulo/cosangulo)
  end;
end;
end.
(function angulo      )
(unit dfuncion      )

```


Unidad de utilerías para graficación

```

}
unit dgrafica;

interface
uses crt,graph,declara;

procedure modografico(fondo : word);           (inicializa modografico )

var
  graphdriver,           (tipo de drive para graficar )
  graphmode             (modo de la pantalla )
  : integer;
  pathdriver           (directorío para graficar )
  : string;
  palette             (declaración de colores )
  : palettetype;

procedure modotexto(i,j,xini,yini,xfin,yfin : integer);
                                     (inicializa textos dentro del )
                                     (modo grafico )

procedure pintaorilla(colortrazo: word);
                                     (dibuja el marco )

procedure pintaejes;
                                     (dibuja los ejes coordenados )

var
  i,j : integer;
                                     (contadores )

procedure grafica(x,y,max: real; color : word; i : integer);

var
  z,           (coordenada z en la pantalla )
  w,           (coordenada w en la pantalla )
  tempx,
  tempy
  : word;

procedure titulos(max,defoco : real; leyenda,nombre : string);

var
  defocus,           (valor de defoco en grafica )
  maximo : string;  (valor maximo en el eje y )

implementation

procedure modografico;

```

```

begin
graphdriver := 9;
graphmode := 2;
pathdriver := '';
initgraph(graphdriver,graphmode,pathdriver);
with palette do
begin
size := 15;
colors[0] := black;
colors[1] := blue;
colors[2] := green;
colors[3] := cyan;
colors[4] := red;
colors[5] := magenta;
colors[6] := brown;
colors[7] := lightgray;
colors[8] := darkgray;
colors[9] := lightblue;
colors[10] := lightgreen;
colors[11] := lightcyan;
colors[12] := lightred;
colors[13] := lightmagenta;
colors[14] := yellow;
colors[15] := white;
setallpalette(palette);
end;
setbkcolor(fondo);
setfillstyle(emptyfill,fondo);
clearviewport;
end;
)
)

procedure modotexto;

begin
textmode(CB0);
window(xini,yini,xfin,yfin);
textbackground(i);
textcolor(j);
clrscr;
end;
)

procedure pintaorilla;

begin
setviewport(0,0,maxX,maxY,true);
setcolor(7);
rectangle(0,0,638,478);
rectangle(10,10,628,468);
setcolor(colortrazo);
setviewport(10,10,(maxX)-10,(maxY)-10,true);
end;
)

procedure pintaeyes;

```

```

begin
begin
  line(45,40,45,440);
  line(45,240,565,240);
  j := 0;
  for i := 1 to 21 do
    begin
      inc(j,50);
      line(j+10,207,j+10,240);
    end;
  j := 0;
  for i := 1 to 11 do
    begin
      inc(j,40);
      line(42,j,48,j);
    end;
end;

procedure grafica;

begin
  z := round((+520+45);
  w := round((-200*y/abs(max))+240);
  setcolor(color);
  if i < 0 then
    line(tempx,tempy,z,w);
  putpixel(z,w,color);
  putpixel(z+1,w,color);
  putpixel(z,w+1,color);
  putpixel(z+1,w+1,color);
  tempx := z;
  tempy := w;
end;

procedure titulos;

begin
  setcolor(7);
  settextstyle(Default font,horizdir,1);
  setusercharsize(1,7,1,1);
  outtextxy(35,250,'0');
  outtextxy(560,250,'1');
  str(max:16,maximo);
  outtextxy(25,25,maximo);
  outtextxy(245,415,levenda);
  outtextxy(245,445,nombre);
  if defoc=0 then
    begin
      str(defoc:16,defocus);
      outtextxy(245,420,'Defoc = ');
      outtextxy(415,430,defocus);
    end;
end;

```

end.

(unit dgrafica

Unidad de graficación de la aberración esférica

```
unit dgesfer;
```

```
interface
```

```
uses
```

```
declara, dfuncion, dtrazmer, desferic, graph, dgrafica;
```

```
var
```

```
trazo1,           (datos del trazo en la sup 1 )
trazo2,           (datos del trazo en la sup 2 )
trazo3           (datos del trazo en la sup 3 )
                : rayos;
```

```
procedure grayoexacto(na,nb,dacentro,db,D,c1,c2,c3,lpparax,X1 : real;
                    var Lp,max,maxz : real; var imaginaria : boolean;
                    var x,y,yy : aberracion);
                (trazo exacto de los rayos )
```

```
implementation
```

```
procedure grayoexacto;
```

```
begin
    (procedure grayoexacto )
    trazo1[7] := 0; (calcula U para el rayo marginal)
    trazo1[8] := 0;
    trazo1[9] := 1;
    If v1a = 0 then (si los rayos son paralelos al )
        begin
            for i := 1 to num do (eje optico ent... )
                begin
                    trazo1[i] := (i/num)*D/2;
                    If imaginaria = false then
                        trazarayo(na,nb,dacentro,db,D,c1,c2,c3,Lp,imaginaria,
                                trazo1,trazo2,trazo3);
                    x[i] := trazo1[i]*2/D; (altura en la 1a. superficie )
                    y[i] := LAP(Lp,lpparax); (aberracion esferica )
                    yy[i] := sin(trazo3[10])*v[i]/cos(trazo3[10]);
                end
            end
        else (if )
            begin
                for i:= 1 to num do
                    begin
                        trazo1[7] := (i/núm)*(arctan((D/2)/(1/v1a+X1)));
                        (calcula U para cada rayo )
                    end
                end
            end
        end
    end
```

```

trazo1[8] := sin(trazo1[7]);
trazo1[9] := cos(trazo1[7]);
trazo1[1] := trazo1[8]/vie;
      (calculo de D cuando los rayos )
if imaginaria = false then
  trazarevo(na,nb,decentro,db,D,c1,c2,c3,Lp,imaginaria,
           trazo1,trazo2,trazo3);
x[i] := (trazo1[10]/D)*2;
y[i] := LAp/Lp/lppara/i;
yy[i] := (1/sin(trazo1[10]))*y[i]/(cos(trazo1[10]));
      (no son paralelos al eje optico )
end;
max := abs(y[1]);
for i := 1 to num do
  (eleccion de la escala para graf)
  if abs(y[i]) > max then
    max := abs(y[i]);
maxz := abs(yy[1]);
for i := 1 to num do
  (eleccion de la escala para graf)
  if abs(yy[i]) > maxz then
    maxz := abs(yy[i]);
end;
      (procedura grayboxacto )
end.
      (unit dgesfer )

```

Unidad de trazo exacto para graficacion de la aberracion cromatica

```

unit dgronom;

interface
uses
  declara,dfuncion,dcal_k,dtrazmer,desferisc,graph,dgrafica;

var
  max : real;
  trazo1,          (datos del trazo en la sup 1 )
  trazo2,          (datos del trazo en la sup 2 )
  trazo3          (datos del trazo en la sup 3 )
  : rayo;

procedure grrayoexacto(na,nb,dacentro,db,D,c1,c2,c3,X1 : real; var Lp
  : real; var imaginaria : boolean;
  var x,y : aberracion);
  (inicializacion de trazo exacto )

implementation

procedure grrayoexacto;

begin
  (procedure grrayoexacto )
  if via = 0 then
  (si los rayos son paralelos al )
  begin
    trazo1[7] := 0;
    trazo1[8] := 0;
    trazo1[9] := 1;
    for i := 1 to num do
      (eje optico ent... )
      begin
        trazo1[i] := (1/num)*D/2;
        if imaginaria = false then
          trazarayo(na,nb,dacentro,db,D,c1,c2,c3,Lp,imaginaria,
            trazo1,trazo2,trazo3);
        x[i] := trazo1[i]*2/D;
        y[i] := Lp;
      end
    end
  else
    (if )
    begin
      for i:= 1 to num do
      begin
        trazo1[7] := (1/num)*(arctan((D/2)/(1/via+X1)));
      end
    end
  end
end

```

```

                                (cálculo U para el rayo marginal)
trazol[8] := sin(trazol[7]);
trazol[9] := cos(trazol[7]);
trazol[1] := trazol[5]/via;
                                (cálculo de D cuando los rayos )
if imaginaria = false then
    trazarayola,nb,dacentro,db,D,c1,c2,c3,Lp,imaginaria,
    trazol,trazo2,trazo3;
x[i] := (trazol[10]/D)*2;
y[i] := Lp;
end;                                (no son paralelos al eje optico )
end;                                (procedura qcrayoevacto )
end.                                (unit dgram )

```

Unidad de trazo exacto para la tabla de rayos marginales

unit dtabla;

interface

uses

crt,declara,dtrazmer;

procedure datosmarginal(naD,nbD,naF,nbF,dacentro,db,via,D,c1,c2,c3,
X1 : real; **var** Lp : real; **var** imaginaria : boolean;
var trazo1C,trazo2C,trazo3C,trazo1D,trazo2D,
trazo3D,trazo1F,trazo2F,trazo3F : rayo);
 {inicializacion de trazo exacto }

var

opcion : word;

procedure tablarayo(na,nb,dacentro,db,c1,c2,c3 : real; **var** Lp : real;
var imaginaria : boolean; **var** trazo1,trazo2,trazo3 :
rayo);

implementation

procedure datosmarginal;

```

begin
  if via = 0 then
    begin
      trazo1C[7] := 0;
      trazo1C[8] := 0;
      trazo1C[9] := 1;
      trazo1C[1] := D/2;
      trazo1D[7] := 0;
      trazo1D[8] := 0;
      trazo1D[9] := 1;
      trazo1D[1] := D/2;
      trazo1F[7] := 0;
      trazo1F[8] := 0;
      trazo1F[9] := 1;
      trazo1F[1] := D/2;
    end
  else
    begin
      trazo1C[7] := arctan((D/2)/(1/via+X1));
      trazo1C[8] := sin(trazo1C[7]);
      trazo1C[9] := cos(trazo1C[7]);
      trazo1C[1] := trazo1C[8]/via;
    end
  end

```

```

trazoID[7] := arctan((D/2)/(1/v1a+X1));
trazoID[8] := sin(trazoID[7]);
trazoID[9] := cos(trazoID[7]);
trazoID[1] := trazoID[8]/v1a;
trazoIF[7] := arctan((D/2)/(1/v1a+X1));
trazoIF[8] := sin(trazoIF[7]);
trazoIF[9] := cos(trazoIF[7]);
trazoIF[1] := trazoIF[8]/v1a;
end;                                     (no son paralelos al eje optico )
repeat
  writeln('      En que linea desea el trazo ?');
  writeln('      (1) Rojo');
  writeln('      (2) Amarillo');
  writeln('      (3) Azul');
  readlnopcion;
until (opcion in [1,2,3]);
case opcion of
  1 : begin
      trazarayo(naC,nbC,dacentro,db,D,c1,c2,c3,Lp,imaginaria,
        trazo1C,trazo2C,trazo3C);
      tablarayo(naC,nbC,dacentro,db,c1,c2,c3,Lp,imaginaria,
        trazo1C,trazo2C,trazo3C);
    end;
  2 : begin
      trazarayo(naD,nbD,dacentro,db,D,c1,c2,c3,Lp,imaginaria,
        trazo1D,trazo2D,trazo3D);
      tablarayo(naD,nbD,dacentro,db,c1,c2,c3,Lp,imaginaria,
        trazo1D,trazo2D,trazo3D);
    end;
  3 : begin
      trazarayo(naF,nbF,dacentro,db,D,c1,c2,c3,Lp,imaginaria,
        trazo1F,trazo2F,trazo3F);
      tablarayo(naF,nbF,dacentro,db,c1,c2,c3,Lp,imaginaria,
        trazo1F,trazo2F,trazo3F);
    end;
end;
end;                                     (procedure datosmarginal )
procedure tablarayo;
begin                                     (procedure tablarayo )
  writeln;
  gotoXY(5,5);
  write('c=1/r');
  gotoXY(20,5);
  write(c1:2:6);
  gotoXY(25,5);
  write(c2:2:6);
  gotoXY(50,5);
  write(c3:2:6);
  gotoXY(5,6);
  write('d');
  gotoXY(27,6);
end;

```

```
write(dacentro:1:4);
gotoXY(42,6);
write(db:1:4);
gotoXY(5,7);
write('a');
gotoXY(27,7);
write(na:2:6);
gotoXY(42,7);
write(nb:2:6);
gotoXY(5,9);
write('D');
gotoXY(20,9);
write(trazo1[1]:1:6);
gotoXY(35,9);
write(trazo2[1]:1:6);
gotoXY(50,9);
write(trazo3[1]:1:6);
gotoXY(5,10);
write('05');
gotoXY(27,10);
write(trazo1[2]:1:6);
gotoXY(35,10);
write(trazo2[2]:1:6);
gotoXY(50,10);
write(trazo3[2]:1:6);
gotoXY(5,11);
write('1');
gotoXY(20,11);
write(trazo1[3]:1:6);
gotoXY(35,11);
write(trazo2[3]:1:6);
gotoXY(50,11);
write(trazo3[3]:1:6);
gotoXY(5,12);
write('1p');
gotoXY(20,12);
write(trazo1[4]:1:6);
gotoXY(35,12);
write(trazo2[4]:1:6);
gotoXY(50,12);
write(trazo3[4]:1:6);
gotoXY(1,13);
write('sin U');
gotoXY(12,13);
write(trazo1[8]:1:6);
gotoXY(27,13);
write(trazo2[8]:1:6);
gotoXY(42,13);
write(trazo3[8]:1:6);
gotoXY(57,13);
write(sin(trazo3[10]):1:6);
gotoXY(5,14);
write('U');
```

```

gotoXY(12,14);
write(trazo1[7]:1:6);
gotoXY(27,14);
write(trazo2[7]:1:6);
gotoXY(42,14);
write(trazo3[7]:1:6);
gotoXY(57,14);
write(trazo3[10]:1:6);
gotoXY(1,15);
write('cos U');
gotoXY(12,15);
write(trazo1[9]:1:6);
gotoXY(27,15);
write(trazo2[9]:1:6);
gotoXY(42,15);
write(trazo3[9]:1:6);
gotoXY(57,15);
write(cos(trazo3[10]):1:6);
gotoXY(5,16);
write('G');
gotoXY(20,16);
write(trazo1[11]:1:6);
gotoXY(35,16);
write(trazo2[11]:1:6);
gotoXY(50,16);
write(trazo3[11]:1:6);
gotoXY(5,17);
write('X');
gotoXY(20,17);
write(trazo1[12]:1:6);
gotoXY(35,17);
write(trazo2[12]:1:6);
gotoXY(50,17);
write(trazo3[12]:1:6);
gotoXY(5,18);
write('Y');
gotoXY(20,18);
write(trazo1[13]:1:6);
gotoXY(35,18);
write(trazo2[13]:1:6);
gotoXY(50,18);
write(trazo3[13]:1:6);
gotoXY(25,20);
write('Lp');
gotoXY(40,20);
write(Lp:1:6);

```

end;

(procedure tabliarayo)

end.

(unit dtabla)

Unidad de trazo meridional

```
unit trazmer;
```

```
interface
```

```
uses
```

```
  crt,declara,dpresent,dfuncion;
```

```
procedure trazomeridional(c,n1,n2 : real; var trazo : rayo; var
  imaginaria : boolean);
  {trazo exacto para rayos }
  {meridionales }
}
```

```
procedure trazarayo(na,nb,dacentro,db,D,c1,c2,c3 : real; var Lp : real;
  var imaginaria : boolean; var trazo1,trazo2,trazo3 :
  rayo);
```

```
implementation
```

```
procedure trazomeridional;
begin
  trazo[5] := (trazo[1]*c - trazo[8]);           {procedure trazomeridional}
  {calculo del seno de I }
  IF (trazo[5]=1) or (trazo[5]=-1) then         {si no esta en el rango }
  begin                                         {if }
    imaginaria := true;                       {se trata de un num. imag.}
    writeln ('El rayo no cruza la superficie');
    espera(25,25);
  end                                           {if }
  else                                         {if }
  begin                                         {else }
    trazo[6] := (trazo[5] * (N1/N2));          {calculo del seno de I' }
    IF (trazo[6]=1) or (trazo[6]=-1) then      {si no esta en el rango }
    begin                                       {if }
      imaginaria := true;                       {se trata de un num. imag.}
      writeln ('Reflexion total interna');
      espera(25,25);
    end                                         {if }
  else                                         {else }
  begin                                         {else }
    trazo[3] := angulo(trazo[5]); {I }
    trazo[4] := angulo(trazo[6]); {I' }
    trazo[10] := (trazo[7] + trazo[3] - trazo[4]); {U' }
    trazo[11] := trazo[1]/(trazo[7] + cos(trazo[3])); {G }
    trazo[12] := trazo[11]*(cos(trazo[10])+cos(trazo[4])); {O' }
    trazo[12] := (1 - cos(trazo[7]+trazo[3]))/c1{k }
    trazo[13] := (sin(trazo[7]+trazo[3])/c1; {Y }
  end;                                         {else }
}
```


Unidad del método D menos d

```
unit Dmenosd;
```

```
interface
```

```
procedure D_d(naF,naC,nbF,nbC,grisorrilla,grisorcentro,db,cosU1,senU2,
cosU2,X1,Y1 : real; var c3 : real);
```

```
var
```

```
r3,
D1, D2,
x2, y2,
Dimenosd1 : real;
```

```
implementation
```

```
procedure D_d;
```

```
begin
D1 := grisorrilla/(cosU1);           (procedure D_d )
Dimenosd1 := D1 - grisorcentro;     (dist recorrida rayo marginal )
D2 := (-1*Dimenosd1*(naF-naC)+db*(nbF-nbC))/(nbF-nbC); (dif en dist entre marg y axia)
                                       (criterio de correccion D-d)
x2 := D2 * (cosU2) + X1 - db;        (coord de intersec del rayo )
y2 := Y1 - D2 * (senU2);             (en la ultima superficie )
r3 := (sqr(x2) + sqr(y2))/(2*x2);    (valor para r3 )
c3 := 1/r3;                          (valor para c3 )
end;
```

```
end.
```

```
(unit Dmenosd )
```

Unidad de trazo paraxial

```
unit dtrazpar;
```

```
interface
```

```
procedure trazoparaxial(y,n1,n2,c : real; var u : real);
```

```
var
```

```
up : real;
```

```
implementation
```

```
procedure trazoparaxial;
```

```
begin
```

```
up := (n1*u + y*(n2-n1)*c)/n2;
```

```
u := up
```

```
end;
```

```
end.
```

```
(procedure trazoparaxial )
```

```
(procedure trazoparaxial )
```

```
(unit dtrazpar )
```