AN ABSTRACT OF THE THESIS OF

Nicholas Liepins for the degree of Master of Science in Mathematics presented on May 8, 1968.

Title: Lens Design on a Digital Computer

Abstract approved: ........................................
Dr. Harry E. Goheen

The enclosed thesis presents an algorithm, programmed in FORTRAN IV, which generates the radii for the objective of an astronomical refracting telescope. The results are analyzed by the Trigonometric Ray Trace, and found to have tolerable amounts of spherical aberration, chromatic aberration and coma.
Lens Design on a Digital Computer

by

Nicholas Liepins

A THESIS
submitted to
Oregon State University

in partial fulfillment of
the requirements for the
degree of
Master of Science
June 1968
APPROVED:

Professor of Mathematics in charge of major

Head of Department of Mathematics

Dean of Graduate School

Date thesis is presented ...May 8, 1968...

Typed by Kristine M. Liepins for Nicholas Liepins
Acknowledgment

I wish to thank Clark and Groff Engineers, Inc., Salem, Oregon for allowing me to use their IBM 1130 computer to develop and test these computer programs.

My special thanks go to Dr. Harry E. Goheen, my advisor and friend, for all his invaluable encouragement.
I. The Problem

The problem of designing a high quality telescope objective suitable for astronomical use has occupied the minds of starry-eyed scientists, both professional and lay, since the inception of critical observational astronomy. During the 17th century Galileo pointed his crude "ocular tube" toward the sky and for the first time saw "Great World Systems." (See page 133 of reference 2.)

The purpose of the present project has been to investigate the possibility of automating the design of an astronomical objective.

The designing of an astronomical objective separates into two parts: (1) actual generation of curves for the components of the lens system, and (2) analysis of a given lens system. In part (1) the designer accepts given requirements for the overall system such as focal length, diameter, and certain desired tolerances and then applies some rules of optics to generate the proper radii for the components of the objective. Unfortunately, there are no set algorithms, and the designer approaches the problem heuristically. And in spite of the existence of lens systems with revered names attached to them, such as Zeiss and Cooke, these have usually been the products of many
painstaking years of trial and error.

This leads us to the second part of the design problem. In part (2) the designer accepts a proposed lens design and evaluates its performance by ascertaining what aberrations are present and of what magnitude. Such analysis is accomplished by either of two means. One way is to set up a chart with certain geometric patterns and to examine the images produced by the system. If deviations from the chart are found, some approximate conclusions about the aberrations may be drawn. Furthermore, a complete system must be built, requiring many hours of grinding and polishing. If the results are unsatisfactory, a few surfaces at best may be reground, but usually the entire system is scrapped and a new one reground. In spite of the inherent inadequacies of the method, it is widely used.

The other method of analysis (due to Gauss and others) is the so-called Trigonometric Ray Trace. This involves mathematically calculating the exact path of two or more light rays through different points in the proposed lens system, which is completely specified by the radii, diameters, indices of refraction and the geometric locations of the component lenses. The points of final intersection of these rays with the central axis of the system, or with one another, give precise information on the aberrations which may be
present in the system. This enables the lens designer to use established theory to predict its optical performance, and in the case of a doublet considered in this paper this theory is accurate enough.

Drawbacks to this method are that it is very slow and requires tedious and distracting effort on the designer's part. The Trigonometric Ray Trace is a very unstable procedure. Small errors in the computations will produce large errors in the results. Consequently, six to nine place logarithm tables are required to keep the accuracy. It is not uncommon therefore, to spend six to eight hours to manual computation in tracing two light rays through four surfaces, for example. With the advent of high speed digital computers, accurate to ten or more places, the Trigonometric Ray Trace, easily coded in one of the many available computer languages, could be transformed into a very powerful tool for optical design. This has been done in the present project and is used in conjunction with the design section.

The author, having been an active telescope maker for the past twelve years, has undertaken this project partly with amateur astronomy in mind. Many of the characteristics of the designing algorithm in the project have been geared toward ease of construction of the final lens system, so that the amateur tele-
scope maker is motivated in attempting the construction of a refracting telescope as opposed to a reflecting telescope which employs a concave, parabolic mirror to focus light rays. Furthermore, notwithstanding the merits of the reflector, its light-gathering power, its colorless image, its simplicity and ease of manufacture, there will always be a majority of the public to whom the word "telescope" means a refractor. And the refractor is certainly the most fool-proof form of the instrument. It has nothing about it which time can deteriorate, and its adjustments, once made, are permanent, barring accidents or gross carelessness.
II. The Project

The project consists of one mainline program and two subroutines, written in FORTRAN IV. The mainline program, OPTIK, generates four radii for an achromatic objective (such as the one in Figure 1.) suitable for an astronomical refracting telescope. The subroutine

![Diagram of a telescope with radii and angles labeled](image)

**Figure 1.**

TRACE uses the Trigonometric Ray Trace (see page 33 of reference 3) for intermediate calculations during the generation of radii by OPTIK and evaluates the final system at the end of the mainline program. The "function" subroutine ISINE calculates the inverse sine for both OPTIK and TRACE.

The following is a brief description of the TRACE algorithm. The names of the variables in Figure 2 are identical to those used in the subroutine.
The ray is always assumed to be incident from the left. The solid line is the path of the ray. The dotted line $R(K)$ is normal to the refracting surface and its length is the radius of curvature of the surface. $S$ is the distance of the object from the lens, and $SS$ is the image distance. $H$ is the vertical distance from the incident point of the ray to the principal axis. The ray makes an angle $T$ with the axis before refraction and $UU$ after refraction; and likewise, the ray makes an angle of incidence $Z$ with the normal before refraction and $ZZ$ after refraction. $Z(K)$ is the index of refraction of the medium on the left of the surface and $Z(K+1)$ on the right.

The path of a light ray which is at some positive
distance from the axis (called a marginal ray) is traced through the lens system, surface by surface, with the following formulae:

\[ \text{SZ} = \sin(Z) = (S - R(K)) \times \sin(T)/R(K) \] (1)

\[ \text{SZZ} = \sin(ZZ) = Z(K) \times \sin(Z)/Z(K+1) \] (2)

\[ \text{UU} = T + Z - ZZ \] (3)

\[ \text{SS} = (R(K) \times \sin(ZZ)/\sin(UU)) + R(K) \] (4)

In the event that the incident light rays are parallel to the axis, which is the case in an astronomical telescope (at the first surface only), formula (1) must be replaced with

\[ \text{SZ} = \sin(Z) = H/R(K) \] (5)

Thus the first part of the subroutine is repeated \( n \) times, where \( n \) is the number of surfaces in the system. Finally the algorithm obtains the marginal ray intercept distance with the axis, \( \text{SS} \). At the end of each loop, the image distance \( \text{SS} \) less the thickness of the previous component \( \text{TH}(K) \) is used as the new object distance \( S \),

\[ S = \text{SS} - \text{TH}(K) \] (6)

Likewise, the new angle \( T \) has the value of \( \text{UU} \) just computed,

\[ T = \text{UU} \] (7)

For a ray which enters the system along the axis or very close to it (paraxial ray), the intercept angles are so small that they are practically identical with
their sines. Such a ray is traced in the second part of TRACE by the following formulae:

\[
\begin{align*}
PSZ &= PU*(PS - R(K))/R(K) \\
PSZZ &= Z(K)*PSZ/Z(K+1) \\
PUU &= PU + PSZ - PSZZ \\
PSS &= R(K)*PSZZ/PUU + R(K)
\end{align*}
\]

(8)  
(9)  
(10)  
(11)

At the end of each loop we have as before

\[
\begin{align*}
PS &= PSS - TH(K) \\
PU &= PUU
\end{align*}
\]

(12)  
(13)

Likewise, when the rays are parallel for the first surface, we replace (8) by

\[
PSZ = H/R(K)
\]

(14)

Radii for a given lens system are generated by the mainline program OPTIK. The main input consists of the following data:

- **KSURF** = number of surfaces in the system
- **R(K)** = predetermined radii (if an option is chosen in the program to by-pass radii-generation and evaluate a given set of curvatures)
- **TH(K)** = thickness of the various components of the system
- **FL** = desired focal length of objective
- **DIAM** = diameter of objective
- **H** = marginal ray height
- **Z(K)** = indices of refraction for the components
- **DISP(K)** = dispersions for the components
- **SPTOL** = desired tolerance.
The program, during the radii-generation, minimizes the three most detrimental aberrations that plague an astronomical refractor: chromatic aberration, spherical aberration, and coma.

Chromatic aberration results from differences in final image distance between rays of different colors. If the two components of the objective are so designed that the following condition is met,

\[
\frac{\text{DISP}(1)}{F_2} + \frac{\text{DISP}(2)}{F_1} = 0
\]  

(15)

then the system is rendered achromatic (see page 118 of reference 8). In expression (15), \(\text{DISP}(K)\), \(K=1,2\), is the dispersion and \(F_1\) and \(F_2\) are the focal lengths of the first and second components respectively. But from fundamental optics we have

\[
\frac{1}{F_L} = \frac{1}{F_1} + \frac{1}{F_2} 
\]  

(16)

\[
\frac{1}{F_1} = (Z(1) - 1)*(X_1 - X_2) 
\]  

(17)

\[
\frac{1}{F_2} = (Z(2) - 1)*(X_3 - X_4) 
\]  

(18)

where \(X_K\) is the reciprocal of the radius for \(K=1,2,3,4\).

Combining (15) and (16) we get

\[
\frac{1}{F_1} = \frac{\text{DISP}(1)}{F_L}*(\text{DISP}(1) - \text{DISP}(2)) 
\]  

(19)

\[
\frac{1}{F_2} = \frac{\text{DISP}(2)}{F_L}*(\text{DISP}(1) - \text{DISP}(2)) 
\]  

(20)

Applying (17) and (18) to (19) and (20) respectively we get

\[
X_1 - X_2 = C_1 
\]  

(21)

\[
X_3 - X_4 = C_2 
\]  

(22)

where
\[ CK = \text{DISP}(K) \times (-1)^{K+1} / \text{FL} \times \text{DISP}(1) - \text{DISP}(2) \times (Z(K) - 1) \]

for \( K = 1, 2 \). \hspace{1cm} (23)

Since we are designing this system with the amateur telescope builder in mind, we take \( X2 = X3 \), not only to reduce the number of different radii, but also to eliminate chromatic aberration, as can be deduced from (15), (17), and (18). Hence throughout the later program we shall preserve this condition. Initially furthermore, we take \( X1 \) to be the negative of \( X2 \). These considerations lead to the first approximations,

\[ X1 = C1/2 \] \hspace{1cm} (24)
\[ X2 = -X1 \] \hspace{1cm} (25)
\[ X3 = X2 \] \hspace{1cm} (26)
\[ X4 = X3 - C2 \] \hspace{1cm} (27)

By algebraic manipulation we deduce

\[ X1 = C1 + C2 + X4 \] \hspace{1cm} (28)

This relation will be used throughout the program, but we shall allow the common value of \( X2 \) and \( X3 \) and the value of \( X4 \) to vary independently.

Spherical aberration occurs when the marginal rays come to a different focus from the paraxial rays. It may be reduced to a minimum by minimizing the expression

\[ Y2 * H2^2 / F2^3 + Y1 * H^2 / F1^3 \] \hspace{1cm} (29)

(see page 120 of reference 8), where

\[ H = \text{marginal ray height for first component} \]
\( H_2 = \text{marginal ray height for second component} \)

\[ Y_1 = A(1)S_1^2 - B(1)S_1 + C(1) + D(1) \]

\[ Y_2 = A(2)S_2^2 + B(2)S_2P_2 + C(2)P_2^2 + D(2) \]

(subscripted \( A, B, C, D \) being constants)

\[ S_1 = \frac{(R(2) + R(1))}{(R(2) - R(1))} \]

\[ S_2 = \frac{(R(4) + R(3))}{(R(4) - R(3))} \]

\[ P_2 = -2F_2/F_1 - 1. \]

Coma results when image points which focus off the axis of the system become blurred. It may be reduced by minimizing

\[ (H_2^2(Q(1)S_1 - P(1))/F_1^2 + \]

\[ (H_2^2(Q(2)S_2 + P(2)P_2)/F_2^2) \] (30)

(see page 101 of reference 8) where subscripted \( Q \) and \( P \) are constants. As indicated above, \( H \) is the ray height at the first component, but since we have no ready-made formula for determining the ray height at the second component, TRACE is used with a parameter "2" that specifies to the subroutine to return after tracing through the first three surfaces with values for \( T \) and \( SZ \) declared in a COMMON statement. Then, (see page 593 of reference 6),

\[ H_2 = R(3)\sin(T + SZ) \] (31)

yields the ray height necessary for the coma and spherical aberration conditions in (30) and (29).

In the program listing for OPTIK, statements 799 through 800 (note that the labels do not follow a consecutive numerical order) define two nested loops which
minimize coma and spherical aberration under the conditions imposed by the chromatic aberration equation. The correction parameters DEL and EPS are small incrementals for radii two and three; and one and four, respectively. The above nested loops, which in effect cycle nine times, provide the following possibilities for the radii incrementals:

<table>
<thead>
<tr>
<th>cycle</th>
<th>incremental for radii 2 &amp; 3</th>
<th>incremental for radius 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>+EPS</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-EPS</td>
</tr>
<tr>
<td>4</td>
<td>+DEL</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>+DEL</td>
<td>+EPS</td>
</tr>
<tr>
<td>6</td>
<td>+DEL</td>
<td>-EPS</td>
</tr>
<tr>
<td>7</td>
<td>-DEL</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>-DEL</td>
<td>+EPS</td>
</tr>
<tr>
<td>9</td>
<td>-DEL</td>
<td>-EPS</td>
</tr>
</tbody>
</table>

Radius 1, as mentioned before, is determined from radius 4, by formula (28) and, of course, radius 2 equals radius 3. A combined aberration value is calculated by adding equations (29) and (30), and the least one among the nine possible aberration values is stored in the variable ABMIN. In addition, corresponding reciprocals of the radii are stored in XR(K). After the loop, ABMIN is compared with a certain tolerance TOL. If ABMIN does not fall below TOL, a ray trace is made to determine the condition of spherical aberration. If the spherical aberration is below a certain value we are done; if not, we iterate by applying
further corrections on the radii in the nested loop.

The reasons for using spherical aberration as a criterion instead of coma are two. One is that, from past experience, if spherical aberration is reduced, coma is almost always automatically minimized. Two, in experimenting with the equations for coma and spherical aberration, it was found that under identical EPS and DEL incrementals, the coma expression approaches zero much more rapidly than that of spherical aberration.
III. Results

In analyzing our design we perform a ray trace and use

\[
\text{SPHER} = \text{PSS} - \text{SS} \quad (32)
\]
\[
\text{COMA} = 1 - \frac{\text{PSS*PUU}}{\text{SS} \cdot \sin(UU)} \quad (33)
\]
to determine the amount of spherical aberration and coma present in our system (see pages 587 and 589 of reference 6). These amounts are printed out immediately following the radii iterations.

Using four inch diameter, 47.13 inch focal length objective as an example, we see the spherical aberration and coma amount to 0.03710 and 0.00005 respectively. The corresponding permissible aberrations are given by \((93 \cdot 10^{-6}) \cdot (2*\text{FL}/\text{DIAM})^2\) and \((.0025)\) (see page 594 of reference 6). Using our example, the former turns out to be 0.05180. Thus, both aberrations are well within these tolerable values.

Since we are designing the objective in order that the amateur astronomer have little difficulty constructing it, standard lens blanks are used for the two components. Consequently, during the iterations, how do we know that the amount of "bending" that some component goes through, as the corresponding radii get smaller, does not exceed the given thickness of the glass? For this purpose, we use the concept of "SAGITTA" which is defined
by $S = R - X$ in Figure 3.

Combining $R^2 = X^2 + r^2$ and $S = R - X$ we obtain

$$S = R - (R^2 - r^2)^{\frac{1}{2}}$$

$$S = \frac{r^2}{2R} + \frac{r^4}{8R^3} + ...$$

where $S = \frac{r^2}{2R}$ is a very good approximation. A little arithmetic computation shows that the sum of the two sagittas for each component of the objective in the example reduces the thickness at the edges very slightly. However, if we were designing a system of very short focal length, this reduction could amount to a large value. In that case, the mainline program would have to be modified to read in a number of available thicknesses for the two components. Then during each iteration,
after a new set of radii were calculated the sagittas would be checked. If they exceeded a given thickness, a larger one would be introduced and the iterations continued.

These results establish a method whereby a telescope manufacturer as well as an amateur astronomer can design a doublet for his own purposes, free from chromatic aberration, coma and spherical aberration. With the availability of computer time at public rental computer centers the Fortran program of the next section can be used in practical design.
// JOB
// FOR
*EXTENDED PRECISION
*ONE WORD INTEGERS
*IOCS(DISK+132 PRINTER+CARD)
*LIST ALL
*NAME OPTIK

REAL ISINE
DIMENSION IDEC(18),IRAD(18),DISP(2),XX(4),XR(4),X(4),A(2),B(2)
DIMENSION C(2),D(2),E(2),ABR(3,3),W(4),P(2),Q(2)
COMMON T,SZ,SURF,R(4),TH(3),PSS,SS,PUU,UU

1001 FORMAT (2(I1*I),2(16A1),I1)
1002 FORMAT (I1*I,F10.5)
1003 FORMAT (I1*I,F7.5)
1004 FORMAT (I1*I,F9.5,2(16x),F4.2)
1006 FORMAT (I1*I,F8.5)
1007 FORMAT (2X,F10.7,3F7.5)
READ (2*1001) I,KSURF,IDEC,IRAD,K
DO 2 L=1,KSURF
  READ (2,1002) I, R(L)
  IS = KSURF - 1
  DO 3 L = 1,IS
    READ (2*1003) I, TH(L)
    READ (2*1004) I, FL, DIAM.H
    IS = KSURF + 1
    DO 5 L = l, IS
      READ (2,1003) I, Z(L)
      IS = KSURF + 1
      DO 6 L = i, IS
        READ (2*1006) I, DISP(L)
        READ (2*1007) TOL.DEL,EPS,SPT4L
        GO TO (80,90), K
    7000 FORMAT (1H1,6X,'DESIGN OF REFRACTOR OBJECTIVE',/1H1,6X,'ITERATION',1,5X,'RADIUS 1',4X,'RADIUS 2',4X,'RADIUS 3',4X,'RADIUS 4',/1H1)
    90 WRITE (3,7000)
    C COMPILE CONSTANTS
    DO 400 K=1,2
      J=2*K
      A(K) = (Z(J)+Z(J)+2.0)/(8.0*Z(J)*(Z(J)-1.0)**2)
      B(K) = (Z(J)+Z(J)+1.0)/(2.0*Z(J)*(Z(J)-1.0))
      C(K) = (3.0*Z(J)+Z(J)+2.0)/(8.0*Z(J))
      D(K) = Z(J)/(8.0*(Z(J)+Z(J)-1.0)**2)
      E(K) = Z(J)+1.0
      P(K) = (6.0*Z(J)+3.0)/(4.0*Z(J))
      Q(K) = (3.0*Z(J)+3.0)/(4.0*Z(J)-(Z(J)-1.0))
      400 CONTINUE
    ITER = 0
    PREV = 9999999.9
    ABMIN = 9999999.0
    C CHROMATIC ABERRATION
    V2 = DISP(1) - DISP(2)
    C1 = DISP(1)/(FL*V2*E(1))
    X(1) = C1/2.0
    X(2) = -X(1)
    X(3) = X(2)
    C2 = DISP(2)/(FL*V2*E(2))
X(4) = X(3) - C2
DO 788 1J = 1, 4
788 W(1J) = 1.0/X(1J)

C CORRECTION TERMS - EPS FOR X(4) - DEL FOR X(M)

799 DO 800 I = 1, 3
   IF (I .EQ. 2) GO TO 804
800 IF (I$2) GO TO 804
     EPS1 = EPS1 - EPS
804 DO 809 J = 1, 4
     EPS1 = EPS1 - EPS
809 DO 800 J = 1, 4
     EPS1 = EPS1 - EPS
808 XX(2) = 1.0/(W(2) + DEL1)
     XX(3) = 1.0/(W(3) + DEL1)
     XX(4) = 1.0/(W(4) + EPS1)
     XX(1) = C1 + C2 + XX(4)
801 DO 809 IJ = 1, 4
809 R(IJ) = 1.0/XX(IJ)
     S1 = (R(2) + R(1))/((R(2) - R(1))
     S2 = (R(4) + R(3))/((R(4) - R(3))
     F1 = 1.0/(E(1)*(XX(1)*XX(2)))
     F2 = 1.0/(E(2)*(XX(3) - XX(4)))
     P2 = -2.0*F2/F1
     SPHHERICAL CORRECTION
     Y1 = A(1)*S1*S1 + B(1)*S1 + C(1) + D(1)
     Y2 = A(2)*S2*S2 + B(2)*S2 + P2
     CALL TRACE(2)
     H2 = R(3)*SIN(T + ISINE(SZ))
     V2 = H*H*Y1/(F1*F1)
     COMA CORRECTION
     V2 = H*H*(W(1)*S1 - P11)/(F1*F1)
     V2 = V2 + H2*H2*(Q(2)*S2 + P(2)*P2)/(F2*F2)
     ABR(I, J) = ABR(I, J) + ABS(V2)
     IF (ABMIN .EQ. 991) 821
820 ABRMIN = ABR(I, J)
     DO 821 1J = 1, 4
821 XX(1J) = XX(1J)
800 CONTINUE

IF (PREV .NE. ABRMIN) 825
822 IF (ABMIN .LE. TOL) CALL TRACE(1)
PREV=ABMIN
SPHER=PSS-SS
IF (ABS(SPH)-SPTOL) 826*826*888
888 DO 824 IJ=1,4
   R(IJ)=1.0/XR(IJ)
824 W(IJ)=R(IJ)
8825 FORMAT (1H+9X+13*4X+4F12.5)
ITER=ITER+1
IF (ITER=1) 901,900,901
901 IK=ITER/10
IK=IK*10
IF (ITERIK) 799.900.799
900 WRITE (3,8825) IK,IK
GO TO 799
C
NO SOLUTION
825 CALL EXIT
C
DONE---------
826 DO 827 IJ=1,4
827 R(IJ)=1.0/XR(IJ)
WRITE (3*8825) IK,IK
80 CALL TRACE(1)
C
CALCULATE SPHER ABERRATION AND COMA
SPHER=PSS-SS
COMA=1.0-PSS*PUU/(SS*SIN(UU))
C
PRINT OUT
2000 FORMAT (1H+20X,'TRIGONOMETRIC RAY TRACE OF'+/1H+20X+18A2+/1H+120X+18A2+/1H0+13X,'DIAMETER'+8X+F4.2+/1H+13X,'FOCAL LENGTH '+2F9.5)
WRITE (3*2000) IDEC,IRAD,DIAM,FL
2001 FORMAT (1H0+13X,'MEDIUM '+5X,'RADII'+5X,'THICKNESS'+3X,'INDEX'+13X,'DISPERSION')
WRITE (3*2001)
3002 FORMAT (1H+15X,'('+I10)+5X+F10.5+3X+F7.5+3X+F7.5+3X+F8.5+/1H+23X+F10.5)
3003 FORMAT (1H+15X,'('+I10)+18X+F7.5+3X+F7.5)
3001 FORMAT (1H+15X,'('+I10)+28X+F7.5)
MED=1
WRITE (3*3001) MED,Z(1)
MED=2
WRITE (3*3002) MED,R(1)+TH(1)+Z(2)+DISP(1)+R(2)
MED=3
IF (Ksurf=2) 10,20,10
10 WRITE (3*3003) MED,TH(2)+Z(3)
MED=4
WRITE (3*3002) MED,R(3)+TH(3)+Z(4)+DISP(2)+R(4)
MED=5
20 WRITE (3*3001) MED,Z(MED)
2002 FORMAT (1H0+13X,'MARGINAL RAY HEIGHT '+F4.2)
WRITE (3*2002) H
2004 FORMAT (1H0+13X,'SPHERICAL ABERRATION '+F9.5+/1H+13X,'COMA'+18X+1F9.5)
WRITE (3*2004) SPHER,COMA
CALL EXIT
END
VARIABLE ALLOCATIONS

T = 7FFD SIZ = 7FFA KSURF = 7FF9 W = 7FF6 R = 7FF3 Z = 7FE7 TH = 7FD8 PSS = 7FCF SS = 7FCC PUU = 7FC9
UU = 7FC6 Disp = 0003 XX = 000F XR = 0018 X = 0027 A = 002D B = 0033 C = 0039 D = 003F E = 0045
ABR = 0060 W = 006C P = 0072 Q = 0078 FL = 007B DIAM = 007E TOL = 0081 DEL = 0084 EPS = 0087 SPTOL = 008A
PREV = 008B ABRMIN = 0090 V1 = 0093 C1 = 0096 C2 = 0099 EPS1 = 009C DEL1 = 009F S1 = 00A2 S2 = 00A5 F1 = 00A8
F2 = 00AB P2 = 00AE Y1 = 00B1 Y2 = 00B4 H2 = 00B7 SPHER = 00BA COMA = 00BD IDEC = 00DA IRAD = 00EC I = 00ED
K = 00EE L = 00EF IS = 00F0 J = 00F1 ITER = 00F2 IJ = 00F3 IK = 00F4 MED = 00F5

STATEMENT ALLOCATIONS

1001 = 011E 1002 = 0129 1003 = 012D 1004 = 0131 1006 = 0139 1007 = 013D 1008 = 0142 8825 = 0178 2000 = 0183 2001 = 0189
3002 = 01DA 3003 = 01F0 3001 = 01FD 2002 = 0208 2004 = 0216 2 = 025B 3 = 0279 5 = 02A2 6 = 02C1 90 = 02E6
400 = 0397 788 = 03F9 799 = 040C 801 = 0418 802 = 0424 804 = 0429 805 = 0435 806 = 0438 807 = 0441
808 = 0446 809 = 0474 820 = 05B1 821 = 05C6 800 = 05D7 991 = 05EE 822 = 05F5 823 = 05FC 888 = 0611 824 = 0620
901 = 063A 900 = 064E 825 = 0659 826 = 065A 827 = 065E 80 = 067A 10 = 06DA 20 = 0708

FEATURES SUPPORTED
ONE WORD INTEGERS
EXTENDED PRECISION
IOCS

CALLED SUBPROGRAMS

ISINE TRACE ESIN EABS EADD EADDX ESUB ESUBX EMPY EMPYX EDIV EDIVX ELD ELDX ESTO
ESTOX ESBR EDVR EDVRX EAXI SBRED SWRT SCOMP SFIO SIOAI SIOAF SIOFX SIOF SIOI SUBSC
SNR CARDZ PRNZ SDFIO

REAL CONSTANTS
9000000000E 01 = 00FC 9800000000E 01 = 00FF 9100000000E 01 = 0102 9300000000E 01 = 0105 9600000000E 01 = 0108
9000000000E 01 = 010B 9999999990E 06 = 010E 9999999990E 06 = 0111 9000000000E 00 = 0114

INTEGER CONSTANTS
2 = 0117 1 = 0118 3 = 0119 0 = 011A 4 = 011B 10 = 011C 5 = 011D

CORE REQUIREMENTS FOR OPTIK
COMMON 58 VARIABLES 252 PROGRAM 1576

END OF COMPILATION
JOB
// FOR
*EXTENDED PRECISION
*ONE WORD INTEGERS
*LIST ALL
*NAME TRACE
SUBROUTINE TRACE(IW)

REAL ISINE
COMMON T+SZ+KSURF+H+R(4)+Z(5)+TH(3)+PSS+SS+PUU+UU

C MARGINAL RAY TRACE
398 DO 100 L=1+KSURF
99 IF (L-1) 98+99+98
98 SZ=H/R(L)
GO TO 97
97 SZ=SIN(T)*(S.-RtL))'/R(L)
IF (L=3) 97+496+97
496 GO TO (97+497)+IW
87 T=0.0
86 UU=T+ISINE(SZ)-ISINE(SZZ)
SS=R(L)*SZZ/SIN(UU)+R(L)
IF (L=KSURF) 96+101+96
96 S=SS+TH(L)
T=UU
100 CONTINUE

C PARAXIAL RAY TRACE
101 DO 200 L=1+KSURF
199 PSZ=H/R(L)
GO TO 197
198 PSZ=PU*(PS-R(L))/R(L)
197 PSZZ=PSZ*Z(L)/Z(L+1)
IF (L=1) 186+187+186
187 PU=0.0
186 PUU=PU+PSZ=PSZZ
PSS=R(L)+PSZZ/PUU+R(L)
IF (L=KSURF) 196+497+196
196 PS=PSS+TH(L)
PU=PUU
200 CONTINUE

497 RETURN
END

VARIABLE ALLOCATIONS
T =7FFD SZ =7FFA KSURF=7FF9 H =7FF6 R =7FF3 Z =7FE7 TH =7FDB PSS =7FCF SS =7FCC PUU =7FC9
UU =7FC6 S =0000 SZZ =0003 PSZ =0006 PU =0009 PS =000C PSZZ =000F L =0115

STATEMENT ALLOCATIONS
398 =0022 99 =002C 98 =0039 496 =0058 97 =005E 87 =0076 86 =007A 96 =00A7 100 =00B6 101 =00BE
199 =0028 198 =0038 197 =00E4 187 =00FC 186 =0100 196 =011D 200 =012C 497 =0134

FEATURES SUPPORTED
ONE WORD INTEGERS
EXTENDED PRECISION
CALLED SUBPROGRAMS
ISINE  ESIN  EADD  EADDX  ESUB  ESUBX  EMPY  EMPYX  EDIV  EDIVX  ELD  ELDX  ESTO  ESBR  EDVR
SUBSC  SUBIN

REAL CONSTANTS
.000000000E 00=0018

INTEGER CONSTANTS
1=0018  3=001C

CORE REQUIREMENTS FOR TRACE
COMMON  58 VARIABLES  24 PROGRAM  286

END OF COMPILATION
REAL FUNCTION ISINE(A)
  ISINE = ATAN(A/SQRT(1.0-A*A))
RETURN
END

VARIABLE ALLOCATIONS
  ISINE=0000

FEATURES SUPPORTED
  ONE WORD INTEGERS
  EXTENDED PRECISION

CALLED SUBPROGRAMS
  EATAN  ESQRT  EMPY  ELD  ESTO  ESBR  EDVR  SUBIN

REAL CONSTANTS
  .100000000E 01=000A

CORE REQUIREMENTS FOR ISINE
  COMMON 0 VARIABLES 10 PROGRAM 34

END OF COMPILATION
### DESIGN OF REFRACTOR OBJECTIVE

<table>
<thead>
<tr>
<th>ITERATION</th>
<th>RADIUS 1</th>
<th>RADIUS 2</th>
<th>RADIUS 3</th>
<th>RADIUS 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20.49351</td>
<td>-20.51010</td>
<td>-20.51010</td>
<td>-527.13279</td>
</tr>
<tr>
<td>50</td>
<td>25.71922</td>
<td>-21.92010</td>
<td>-21.92010</td>
<td>-84.66279</td>
</tr>
<tr>
<td>59</td>
<td>25.71922</td>
<td>-21.65010</td>
<td>-21.65010</td>
<td>-84.66279</td>
</tr>
</tbody>
</table>

Table 1.

V. Program

Output
TRIGONOMETRIC RAY TRACE OF ACHROMAT DESIGN ANALYSIS

DIAMETER 4.00
FOCAL LENGTH 47.13000

<table>
<thead>
<tr>
<th>MEDIUM</th>
<th>RADII</th>
<th>THICKNESS</th>
<th>INDEX</th>
<th>DISPERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>1.00000</td>
<td>1.51370</td>
<td>63.60000</td>
<td></td>
</tr>
<tr>
<td>(2)</td>
<td>25.71922</td>
<td>0.52000</td>
<td>1.51370</td>
<td>63.60000</td>
</tr>
<tr>
<td></td>
<td>-21.65010</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3)</td>
<td>0.00000</td>
<td>1.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4)</td>
<td>-21.65010</td>
<td>0.38000</td>
<td>1.61640</td>
<td>36.70000</td>
</tr>
<tr>
<td></td>
<td>-84.66279</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5)</td>
<td>1.00000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MARGINAL RAY HEIGHT 2.00
SPHERICAL ABERRATION 0.03710
COMA 0.00005

Table 2.
<table>
<thead>
<tr>
<th>ITERATION</th>
<th>RADIUS 1</th>
<th>RADIUS 2</th>
<th>RADIUS 3</th>
<th>RADIUS 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26.08605</td>
<td>-26.10269</td>
<td>-26.10269</td>
<td>673.54515</td>
</tr>
<tr>
<td>40</td>
<td>27.24223</td>
<td>-27.27269</td>
<td>-27.27269</td>
<td>321.37516</td>
</tr>
<tr>
<td>50</td>
<td>28.17556</td>
<td>-27.97269</td>
<td>-27.97269</td>
<td>231.07516</td>
</tr>
<tr>
<td>60</td>
<td>30.56626</td>
<td>-27.87269</td>
<td>-27.87269</td>
<td>140.77516</td>
</tr>
<tr>
<td>70</td>
<td>33.04239</td>
<td>-27.75269</td>
<td>-27.75269</td>
<td>104.65516</td>
</tr>
<tr>
<td>80</td>
<td>33.04239</td>
<td>-27.45269</td>
<td>-27.45269</td>
<td>104.65516</td>
</tr>
<tr>
<td>90</td>
<td>33.04239</td>
<td>-27.15269</td>
<td>-27.15269</td>
<td>104.65516</td>
</tr>
<tr>
<td>100</td>
<td>33.04239</td>
<td>-26.85269</td>
<td>-26.85269</td>
<td>104.65516</td>
</tr>
<tr>
<td>101</td>
<td>33.04239</td>
<td>-26.82269</td>
<td>-26.82269</td>
<td>104.65516</td>
</tr>
</tbody>
</table>
TRIGONOMETRIC RAY TRACE OF ACHROMAT
DESIGN ANALYSIS

<table>
<thead>
<tr>
<th>MEDIUM</th>
<th>RADII</th>
<th>THICKNESS</th>
<th>INDEX</th>
<th>DISPERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td></td>
<td></td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>(2)</td>
<td>33.04239</td>
<td>0.52000</td>
<td>1.51370</td>
<td>63.60000</td>
</tr>
<tr>
<td></td>
<td>-26.82269</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3)</td>
<td></td>
<td>0.00000</td>
<td>1.00000</td>
<td></td>
</tr>
<tr>
<td>(4)</td>
<td>-26.82269</td>
<td>0.38000</td>
<td>1.61640</td>
<td>36.70000</td>
</tr>
<tr>
<td></td>
<td>-104.65516</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5)</td>
<td></td>
<td>1.00000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MARGINAL RAY HEIGHT 2.00

SPHERICAL ABERRATION 0.02552
COMA 0.00003
VI. Application of the Program to the Design of a Refractor Objective

We are given a diameter of a telescope, a focal length for the objective, and the thickness and indices of refraction for available optical glass. We wish to establish the details of the doublet.

We know from experience that the first lens should be made out of crown glass \( Z(K) = 1.51370 \) and the second out of flint glass \( Z(K) = 1.61640 \). The initial computations of the radii are carried out with equations (24), (25), (26) and (27) of Chapter II. These then are refined by minimizing expressions (29) and (30) until the results of Table 1 are obtained.

After the radii are computed, the program prints out Table 2, containing detailed information for the benefit of the designer.
BIBLIOGRAPHY