COMPUTER ASSISTED DESIGN OF WELL-BAFFLED
AXIALLY SYMMETRIC OPTICAL SYSTEMS

by

Alan Wayne Greynolds

A Thesis Submitted to the Faculty of the
COMMITTEE ON OPTICAL SCIENCES (GRADUATE)
In Partial Fulfillment of the Requirements
For the Degree of
MASTER OF SCIENCE

In The Graduate College
THE UNIVERSITY OF ARIZONA

1 9 8 0

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: _____

Alan W. Drysdale

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Roland V. Shack

R. V. SHACK

Professor of Optical Sciences

12/3/79

Date

ACKNOWLEDGMENTS

The author wishes to thank the faculty and staff of the Optical Sciences Center, especially my advisor, Dr. Roland V. Shack.

The encouragement from and invaluable discussions with my co-workers, Robert Breault and Steven Lange, were greatly appreciated.

The computer funds for this thesis were provided by the University of Arizona Computer Center and Breault Research Organization, Tucson, Arizona.

And finally, I would like to thank my parents and wife, Vivian, for without their moral support none of this would have been possible.

TABLE OF CONTENTS

	Page
LIST OF ILLUSTRATIONS	vi
LIST OF TABLES	viii
ABSTRACT	ix
1. INTRODUCTION	1
Background and Thesis Content	1
Nomenclature	2
2. BASIC PRINCIPLES OF EFFECTIVE STRAY RADIATION SUPPRESSION	4
Baffle Alignment and Field Stops	7
Aperture and Obscuration Stops	9
3. COMPUTER ASSISTED DESIGN	14
Determining the Unvignetted Beam	14
Paraxial Approximation	17
Baffle Design with Real Rays	21
GOSBOP: A Baffle Design Program	22
Basic Cassegrain Telescope Designs	26
4. EVALUATING A DESIGN	39
The APART Computer Program in General	40
The Mechanics of Using APART	41
A New Version of APART1	45
5. OPTIMIZATION OF A DESIGN	51
Cassegrain Example	51
First Iteration	51
Second Iteration	53
Third Iteration	55
Cassegrain with a Lens Reimager	55
6. SUMMARY AND SUGGESTIONS FOR FUTURE RESEARCH	61

TABLE OF CONTENTS, Continued

	Page
APPENDIX A: APPROXIMATE PARAXIAL SOLUTION TO THE BAFFLING OF A CASSE- GRAIN TELESCOPE	62
APPENDIX B: LISTING OF THE CYBER FORTRAN. EXTENDED CODE FOR THE GOSBOP COMPUTER PROGRAM	72
LIST OF REFERENCES	107

LIST OF ILLUSTRATIONS

Figure	Page
1. A Minimally Baffled Cassegrain Telescope	5
2. Vane Structure on the Inside of the Conical Baffle	8
3. Main Baffle Length Required to Shield the Inner Conical Baffle	10
4. A Larger Central Obscuration Shields the Inner Conical Baffle	11
5. The Stops as Viewed from the Image Plane	13
6. The Six Rays Used in Determining the Boundaries of the Beam	18
7. Iterative Calculations Required in the Design of Baffles	23
8. Block Diagram of the GOSBOP Computer Program . .	25
9. Output from GOSBOP for a Typical Cassegrain Telescope	30
10. Varying the Position of the Image Plane in a Cassegrain for a Fixed Primary and Overall Focal Length	33
11. There Exists an Image Plane Location That Minimizes the Required Obscuration Ratio . . .	34
12. One Way of Varying the Secondary Magnifica- tion on the Y-Y Diagram	36
13. Relative Appearances of the Three Designs . . .	38
14. Surfaces in a Cassegrain Imaged into the First or Object Space	43
15. Surfaces Imaged into Third and Final Space . . .	44

LIST OF ILLUSTRATIONS, Continued

Figure	Page
16. Comparison of APART1 and APART Analyses of a Typical Cassegrain Telescope	48
17. GOSBOP-APART1 Input Deck for the Design and Analysis of a Cassegrain	50
18. APART1 Output Showing the Major Contributor at Each Source Angle	52
19. Results of the Two Methods for Reducing the Inner Conical Baffle Path	54
20. Reduction of the PST at Each Optimization Iteration	56
21. Configuration of Design at the End of Each Iteration	57
22. Comparison of the Final Design with the Reimager Design	59
23. A Cassegrain Telescope Equipped with a Reimaging Lens	60
A.1. Definition of Coordinate System and Parameters Used in the Approximate Analysis of a Baffled Cassegrain Telescope	64
A.2. Absolute Minimum Obscuration Ratio in a Cassegrain	69
A.3. Required Obscuration Ratio for a Minimally Baffled Cassegrain with a Relative Image Height of 0.1	70
A.4. Obscuration Ratio for a Minimally Baffled Cassegrain with a Normalized Image Plane Location of 0.1	71

LIST OF TABLES

Table	Page
1. Properties of the Three GOSBOP Solutions to an F/10 Minimally Baffled Cassegrain . . .	37

ABSTRACT

Basic principles of stray radiation suppression are employed in a FORTRAN computer program that assists in the design of well-baffled axially symmetric optical systems. Called GOSBOP (General Optical System Baffle Optimization Program), this program can design the baffle structure of a typical system according to the user's specifications in less than a second on a CYBER 175 computer. The program can also be used in conjunction with a new version of the first APART (Arizona's Paraxial Analysis of Radiation Transfer) subprogram to design and evaluate a baffled optical system in a single computer run with a minimal amount of input. The programs enable the user to identify and eliminate troublesome single diffuse-scattering paths so that a preliminary design can be optimized in terms of both optical performance and stray radiation suppression in only a few successive computer runs. As an example, the technique is applied to optimizing the baffle structure of a typical Cassegrain telescope such that the stray radiation reaching the image plane is reduced by up to three orders of magnitude. The stray radiation rejection of the final design compares favorably with relatively more complicated designs.

CHAPTER 1

INTRODUCTION

The task of designing an opto-mechanical system with known low stray radiation characteristics can be difficult and time-consuming. The purpose of the research summarized in this thesis was to minimize this effort by computerizing the tedious calculations and developing a straightforward step-by-step procedure for reducing the stray radiation to acceptable levels.

Background and Thesis Content

The field of stray radiation suppression has grown rapidly in recent years mostly due to the increasing number of radiation sensors going into orbit where scattered out-of-field Earthshine is the limiting factor in the performance of these instruments, especially at the longer infrared wavelengths. The present state of stray radiation suppression methods has been nicely summarized in a paper by Breault (1977). Some of the concepts presented in that paper are covered again in a slightly different form and emphasis in Chapter 2.

This thesis grew out of a research project that explored the stray radiation characteristics of various

optical designs. As part of that project the author wrote a small 100-line FORTRAN computer program that determined the locations of the baffles in a Cassegrain telescope. Over the years this program grew to be over 2000 lines of code capable of handling any axially symmetric optical system. The present version of the program is described in Chapter 3.

Soon after the completion of this general baffle design program, the author realized that any type of design program, no matter how sophisticated, is of little value unless some sort of figure-of-merit is generated that establishes how good the design really is. Therefore, an already existing stray radiation analysis program was modified such that it could essentially operate as a subroutine of the design program. This particular figure-of-merit calculation is described in Chapter 4.

Now with the ability to quickly and easily design and evaluate a baffled optical system, it became a straightforward matter to optimize a particular design with respect to stray radiation rejection. This technique is illustrated in Chapter 5 using a Cassegrain telescope as an example.

Nomenclature

The nomenclature used in this thesis may be somewhat different than that routinely used by the reader. For instance, the term "baffle" refers to any large-scale mechanical structure that is intended to block the propagation of

stray radiation. Another commonly used word for this is "shield". The fine-scale structure such as circular annuli found on some "baffle" surfaces are called "vanes" or "vane structure" and are used to reduce the effective reflectivity of the "baffle". This definition may possibly lead to confusion because some workers in the field of stray radiation refer to these circular annuli as "ring baffles" or just "baffles". Therefore, some readers may find it easier to substitute "shield" for "baffle" and "baffles" for "vanes" throughout this thesis.

CHAPTER 2

BASIC PRINCIPLES OF EFFECTIVE STRAY RADIATION SUPPRESSION

When designing a well-baffled optical system or radiation sensor, the first, and most obvious, situation to avoid is allowing radiation from an external out-of-field source, such as the sun, to directly impinge upon the image or detector plane. A good example of this is found in a properly baffled Cassegrain telescope. Two baffles, the secondary and conical, extend toward each other from the two mirrors such that they just border the unvignetted beam. Any line through the edges of the two baffles intersects the image plane outside the field. Therefore, no object space radiation can reach the detectors directly but must scatter at least once off some internal surface (see Fig. 1).

Once all such direct stray radiation paths are eliminated, then a designer should proceed to examine those paths that involve one intermediate scattering surface between the source and detector. After these paths are minimized or entirely eliminated, multiple scattering paths of increasing number are considered in turn. This thesis will be primarily concerned with single diffuse scattering

Fig. 1. A Minimally Baffled Cassegrain Telescope.

paths not only because they are easier to analyze but also because they are usually dominate over multiple scattering paths due to the usually high absorption that occurs at each scattering surface involved.

The amount of stray radiation reaching a detector due to a single scattering off a small element of an internal surface, such as a baffle or mirror, is the product of three factors; the amount of radiation incident upon the surface, the reflectivity of the surface for the particular incoming and outgoing directions, and the projected solid angle of the detector as seen from the element. Reducing any one of these factors to zero will of course completely eliminate the contribution of that path. However, the reflectivity is always a finite number for any real material.

Many times when a radiation sensor is designed, the optical and mechanical configuration is set before any serious thought is given to possible stray radiation problems. At this point a computer analysis of the proposed system or an extensive test of the already built sensor will sometimes reveal major flaws in the baffle design, and since the project is usually too far along for major redesigns, only minor fixes can be made. In most cases this means putting more expensive black coatings or vane structure on the offending surfaces, thereby reducing their effective reflectivity.

For the optimal design of baffles, effective stray radiation suppression methods must be incorporated into the opto-mechanical design phase from the beginning. If this is done, then the designer can entirely eliminate the major propagation paths by blocking either the transfer from the source to the scattering surface or the one from the surface to the detector.

Baffle Alignment and Field Stops

The first transfer in a propagation path can be blocked so that radiation does not reach the scattering surface by either proper baffle placement or if possible, the use of a field stop.

We have already seen how lining up baffle edges can prevent direct radiation from reaching the image plane of a Cassegrain telescope. In these minimally baffled Cassegrain designs, the resulting dominant stray radiation path is usually the one in which the direct radiation strikes the inside of the conical baffle and then scatters to the nearby detector plane. The contribution from this path can be reduced by placing vanes on the inner conical baffle surface (Fig. 2). This forces the radiation to scatter twice within the cavities between the vanes thereby reducing the effective reflectivity of the fictitious baffle surface defined by the edges of the vanes. Alternatively, radiation can be kept completely off the inner conical baffle by aligning the edges

Fig. 2. Vane Structure on the Inside of the Conical Baffle.

of the two baffles with the edge of the entrance port at the end of the main baffle as shown in Fig. 3. However, the main tube would usually have to be impractically long to accomplish this. The same baffle alignment can be had with a main tube of reasonable length at the expense of an increased obscuration (see Fig. 4) or altering the basic first-order optical design.

If the optical system has an intermediate real image, then a field stop can be inserted there. The advantage of this is that radiation from any external out-of-field sources is imaged outside the stop and cannot propagate further into the system.

Aperture and Obscuration Stops

Aperture or obscuration stops can be used to block the final transfer in a path, i.e., the one from the scattering surface to the detector. The aperture stop is the aperture in an optical system that limits the outside of the unvignetted signal beam. Any axially symmetric optical system with at least one reflecting element must have a central obscuration and, therefore, an obscuration stop which limits the inside of the beam.

Both of these stops have an effect analogous to the field stop in that they determine what surfaces bordering the beam can be seen by an in-field detector (the field stop determines what surfaces receive radiation from an out-of-field source). To be more precise, if one looks back into

Fig. 3. Main Baffle Length Required to Shield the Inner Conical Baffle.

Fig. 4. A Larger Central Obscuration Shields the Inner Conical Baffle.

the system from a point in the image plane that is inside the unvignetted field-of-view, one typically sees essentially a vignetting diagram as shown in Fig. 5. Since the stops are the limiting apertures, no surfaces except optical surfaces can be seen beyond the stops, i.e., no surfaces bordering the outside of the beam can be seen beyond the aperture stop and no surfaces bordering the inside are visible beyond the obscuration stop. Therefore, shifting of these stops is a valuable tool in the suppression of stray radiation in optical systems just as stop shifts are important in the reduction of aberrations. In fact the closer the stops are to the image plane the smaller the number of "critical" or visible surfaces and the lesser the chance of stray radiation reaching the image plane.

Fig. 5. The Stops as Viewed from the Image Plane.

CHAPTER 3

COMPUTER ASSISTED DESIGN

The methods and procedures outlined in the previous chapter have been used with varying degrees of success in several optical systems (Bartell et al., 1976). The problem in the past has been the substantial amount of time and effort required to design an opto-mechanical system along such principles. This is mostly due to the fact that there has been no one efficient set of computer programs written to do all the tedious work for the designer. In this chapter one such program will be described.

Determining the Unvignetted Beam

The first task involved in designing the baffle structure of an optical system is the determination of the boundaries of the unvignetted beam. This requires the tracing of rays from the edges of the object to the edges of the stops. For axially symmetric optical systems, we need only consider rays in the meridional plane. Then the unvignetted beam must be bounded on the outside, in general, by four rays that originate from the upper and lower edges of the object and pass through the upper and lower edges of the aperture stop. Let Y be the height of a ray in some plane perpendicular to the optical axis and the superscript sign indicate

the edge of the object from which the ray originated. Likewise, the subscript sign pertains to the aperture stop edge through which the ray passes. Then the height of the outer beam boundary in that plane is given by

$$H = \max \left\{ Y_+^+, Y_+^-, Y_-^-, Y_-^+ \right\} . \quad (1)$$

The rays are shown in this particular order for a specific reason. The upper boundary can be determined by starting from the object and following the first ray until it intersects the next in the sequence. Then one follows that ray and so forth until the final image plane is reached. The sequence is cyclical. Therefore, one can see that the beam boundary in the meridional plane will be composed of a series of line segments connected at either an optical surface, the stop or a real image of the stop or object.

Actually, these four rays are not completely independent of each other for axially symmetric systems, i.e.,

$$Y_-^- = -Y_+^+ \quad (2)$$

$$Y_-^+ = -Y_+^- . \quad (3)$$

Therefore, Eq. (1) can be written in a more compact form by using absolute values.

$$H = \max \left\{ |Y_+^+| , |Y_+^-| \right\} . \quad (4)$$

If the axially symmetric optical system has at least one reflecting optical element, then there will be an

obscuration stop that determines the inner boundary of the beam. A similar formula for the inner boundary can be derived where the rays must now pass through the edges of the obscuration stop. The sequence of rays that must be followed is:

$$0, y_+^-, y_+^+, 0, y_-^+, y_-^- . \quad (5)$$

We have adopted the notation here that upper case variables are associated with the upper or outer boundary and lower case with the lower or inner. The ray with constant zero height is, of course, the optical axis. As before, the boundary is composed of a series of line segments. However, in addition to being joined at optical surfaces and the stop or a real image of it, the segments will also be connected where one of the rays crosses the optical axis instead of, as with the upper boundary, at an image of the object.

The mathematical representation for the height of the inner boundary at any given axial location as derived from the above sequence is somewhat more complicated than the one for the outer boundary.

$$h = \max \left\{ 0, \min \{y_+^-, y_+^+\}, \min \{y_-^+, y_-^-\} \right\}. \quad (6)$$

Note that we have defined the boundary height such that it is never less than zero. Again because of axial symmetry, the rays involved are related by the following expressions:

$$y_-^- = -y_+^+ \quad (7)$$

$$y_-^+ = -y_+^- \quad (8)$$

so that the sequence reduces to

$$0, y_+^-, y_+^+, -0, -y_+^-, -y_+^+ \quad (9)$$

and Eq. (6) becomes

$$h = \max \left\{ 0, \min \{y_+^-, y_+^+\}, -\max \{y_+^-, y_+^+\} \right\} . \quad (10)$$

To summarize, four rays are needed to determine the two boundaries of the beam in an axially symmetric optical system. In addition, the two rays from the center of the object to the edges of the two stops are useful in determining the sizes of the stops when they are located after the first optical surface. These six rays are shown in Fig. 6 in the case of a Cassegrain telescope.

Paraxial Approximation

Equations (4) and (10) can be further simplified in the case of paraxial rays. If Y is the height of the paraxial marginal ray from the center of the object to the top of the aperture stop and $\bar{Y}$ is the paraxial chief ray from the bottom of the object through the center of the aperture stop, then due to the principle of linear superposition of paraxial rays

$$y_+^+ = Y - \bar{Y} \quad (11)$$

$$y_+^- = Y + \bar{Y} , \quad (12)$$

Fig. 6. The Six Rays Used in Determining the Boundaries of the Beam.

and the paraxial form of Eq. (4) is

$$H = \max \{ |Y - \bar{Y}|, |Y + \bar{Y}| \} . \quad (13)$$

For two arbitrary real numbers, a and b , the triangle inequality states that

$$|a \pm b| \leq |a| + |b| . \quad (14)$$

Therefore, Eq. (13) reduces to

$$H = |Y| + |\bar{Y}| \quad (15)$$

which is a well-known result in Gaussian optics.

The derivation for the paraxial form of the inner boundary of the beam is similar. Let y and $\bar{y}$ be the marginal and chief rays through the obscuration stop, respectively. Then

$$y_+^+ = y - \bar{y} \quad (16)$$

$$y_+^- = y + \bar{y} \quad (17)$$

and using the following relationship for any two real numbers

$$a - |b| \leq a \pm b \leq a + |b| , \quad (18)$$

Equation (10) reduces to

$$h = \max \{ 0, |y| - |\bar{y}| \} . \quad (19)$$

An advantage of a paraxial analysis is that stop shifts can be handled in a straightforward manner. A stop shift will not affect the paraxial marginal ray. The new

chief ray can be expressed as a linear combination of the marginal ray and the original chief ray. To be more specific,

$$\bar{Y}' = \bar{Y} - KY \quad (20)$$

where K is the ratio of the heights of the original chief ray to the marginal ray at the plane of the new stop position (Lopez-Lopez, 1973). The boundary equations for arbitrary stop locations become

$$H = |Y| + |\bar{Y} - KY| \quad (21)$$

$$h = \max \{0, |y| - |\bar{Y} - ky|\} \quad (22)$$

The above formulas still require us to trace four rays; the marginal and chief rays for each stop. The aperture stop rays play an important role in the aberrations of optical systems. On the other hand, optical designers do not usually concern themselves with the obscuration stop rays. However, it is possible to express these rays in terms of the more common aperture stop rays. We can define the obscuration ratio e such that

$$y = eY \quad (23)$$

The obscuration stop chief ray can be thought of as just a stop shifted aperture stop chief ray, i.e.,

$$\bar{y} = \bar{Y} - kY \quad (24)$$

where now $k = \bar{Y}/Y$ at the obscuration stop plane. Therefore, Eq. (19) becomes

$$h = \max \{0, e|Y| - |\bar{Y} - kY|\} . \quad (25)$$

To summarize, given the standard marginal and chief ray heights, Y and $\bar{Y}$, at some axial location in an axially symmetric optical system with obscuration ratio e , then in the paraxial approximation the outer and inner boundaries of the unvignetted beam at that location are given by

$$H = |Y| + |\bar{Y} - K(S)Y| \quad (26)$$

$$h = \max \{0, e|Y| - |\bar{Y} - K(s)Y|\} \quad (27)$$

where $K(z) = \bar{Y}/Y$ at a plane of axial location z , and S and s are the axial locations of the aperture and obscuration stop respectively.

Baffle Design with Real Rays

Real rays must be used in designing any system where the positioning of the baffles could be critical to its performance. The tracing of a real ray through an aspheric optical surface of order greater than four (other than a simple conic) must be done by an iterative process. In addition, since the stops can be located anywhere in the system and not necessarily before the first optical surface, the rays through the stops must, in general, be found by an iterative aiming technique. All this can be done on a programmable calculator or with one of the large optical design programs that have such a ray tracing capability.

Once the boundaries of the beam are known, then baffles may be inserted along these boundaries without encroaching on the beam. The intersection of two boundaries that occurs when the beam folds back on itself due to a reflecting optical element determines the edge of the baffle that can be placed there. As pointed out in the last chapter, it is the alignment of these intersections or baffle edges that is of the most interest to the designer.

GOSBOP: A Baffle Design Program

Although there are approximate closed-form analytical solutions (Cornejo and Malacara, 1968), graphical methods (Prescott, 1968), and computer programs (Young, 1967) for the baffling of specific optical designs with specific stop locations, there are none available for arbitrary axially symmetric systems. In the past the designer had to go through a time-consuming series of trial-and-error calculations in order to arrive at a satisfactory design that achieved the desired baffle alignment. On top of this, if it was also advantageous to make a baffle edge one of the stops (e.g., the secondary baffle edge in a Cassegrain is usually the obscuration stop), then trial-and-error was, in general, the only way. Therefore, overall there could be up to four levels of iterative calculations that had to be performed (see Fig. 7). At the very least it could take a designer,

Fig. 7. Iterative Calculations Required in the Design of Baffles.

even with the aid of a sophisticated ray trace program, several hours to perform all the iterations necessary to arrive at a satisfactory design.

In order to remedy this situation, a FORTRAN computer program called GOSBOP (General Optical System Baffle Optimization Program) has been written by the author that will design a baffled axially symmetric optical system according to the user's specifications in less than a second on a CYBER 175 machine. The input is concise, straightforward and unformatted so that it takes only a few minutes for the user to set up a deck. In addition the program generates an input deck for the APART stray radiation analysis program. The analysis of a design will be covered in more detail in the next chapter.

Figure 8 is a block diagram of GOSBOP showing its general structure and the calculations it performs. Each block corresponds to a separate routine in the program. The paraxial equations (26) and (27) are used as the starting guesses for the aiming of the real rays. The boundaries of the beam are determined by the ray sequence method outlined in the paragraph following Eq. (1). The desired baffle alignments and stop locations are reached by a modified Newton-Raphson iterative process.

Instead of going into any more detail about the program, we will proceed to demonstrate its capabilities by applying it to a specific optical configuration.

Fig. 8. Block Diagram of the GOSBOP Computer Program.

Basic Cassegrain Telescope Designs

As an illustrative example, the Cassegrain telescope is a good choice because its many advantages has made it one of the most widely used optical designs. However, the Cassegrain does have one important fault in that the image plane faces the object space. Therefore, external sources outside the field-of-view can be seen directly from the image plane unless proper baffling is provided. We have already seen how the insertion of two baffles will prevent out-of-field radiation from directly striking the image plane (Fig. 1). It only remains to determine the precise locations of these baffles for a particular Cassegrain design. In this section we will use the GOSBOP baffle design program to design a typical Cassegrain telescope such that no out-of-field external sources can be seen from the unvignetted image plane. We will then look at some variations on this basic design.

For reflecting telescope systems, GOSBOP requires the following initial input data.

1. Entrance aperture diameter
2. Angular field-of-view
3. Axial locations, radii of curvature, and aspheric coefficients of mirrors and image plane
4. Obscuration ratio
5. Locations of aperture and obscuration stops.

In order to determine these parameters, we must first select a specific system f-number. The sags on the mirrors are nonlinear and therefore will prevent us from accurately scaling the results of one f-number to another as we could with a simple paraxial analysis. All the following GOSBOP solutions are accurate only for a typical Cassegrain with an overall f-number of ten (10).

Besides the overall f-number, the other system parameters that are usually fixed by the particular application intended for the telescope are the entrance aperture diameter and the field-of-view. The height of the entrance aperture will be assumed to be unity. Therefore, the overall focal length of the telescope is 20. A typical height for the final image is one-tenth of this entrance aperture height. This corresponds to a full angular field-of-view of 0.01 radians or 0.573 degrees.

The focal length of the primary mirror determines roughly the overall length of the telescope. Usually one tries to make it as small as possible without leading to manufacturing difficulties posed by an overly steep aspheric surface. A primary f-number of 2.5 is reasonable even for large systems. Therefore, the vertex radius of curvature of the primary mirror must be equal to -10.

The separation of the mirrors and curvature of the secondary mirror depend on the desired placement of the final image plane. Most systems would have the image located

behind the primary mirror for easy access. A typical distance would be one-half the height of the primary. It then follows that the separation of the mirrors must be 3.9 [see Eq. (A.11) in Appendix A]. It is not necessary for us to input the vertex radius of curvature of the secondary.

GOSBOP can solve for the radius of curvature of the last optical element in a system along with the axial location of the final image that results in a given final f-number.

Since in most cases the first-order optical properties of a design will be fixed by the constraints placed on the baffle geometry, the only remaining parameters that can affect the aberrations in the image are the aspherics on the optical surfaces. For this reason, GOSBOP was given the capability of calculating the conic constants or fourth-order deformation coefficients that result in the elimination of certain third-order monochromatic aberrations. For example, it can determine the aspheric on one of the mirrors in a Cassegrain necessary to eliminate third-order spherical aberration. Or the program could solve for the aspheric coefficients of both mirrors that simultaneously eliminate spherical aberration and coma (i.e., the aplanatic Ritchey-Cretien telescope). This is made possible by the fact that the aspheric contributions to the third-order aberrations are linearly related to the fourth-order deformation coefficients, and therefore, can be determined by the solution of a system of linear equations (see, for example, Welford, 1974). Thus,

it will not be necessary for us to input the conic constants needed to make this system aplanatic.

Finally, we must specify the locations of the stops and an initial value for the obscuration ratio. Nearly all Cassegrain telescopes have the aperture stop at the primary mirror in order to minimize its size. Most Cassegrains, however, do not have their obscuration stop located at the primary mirror but instead at the edge of the secondary baffle. Since GOSBOP has to go through an iterative process to accomplish this, we need a good initial value for the axial location of this edge. First, we can determine an initial value for the required obscuration ratio from Eq. (A.21) of Appendix A. Then, we substitute this value into Eq. (A.16) to get the approximate axial location of the secondary baffle.

Now that all the required initial system parameters are set, we can instruct GOSBOP to vary the obscuration ratio until the upper edges of the two baffles are aligned with the bottom edge of the image plane. Figure 9 is the output from GOSBOP for the above system. Most of the output is self-explanatory. Four major iterations and one-tenth of a second on the CYBER 175 were required to arrive at a solution for which the final obscuration ratio turned out to be 0.33623, about ten percent higher than our approximate initial value of 0.30563. The coordinates of the two baffle edges can be found under the "Intersections" heading.

```

ITERATION 4
LAGRANGE INVARIANT= .50000E-02

SURF  Y      YBAR      ZV      RD      TYPE
1  0.      -.50000E+06  -.10000E+09  INFINITE  OBJC
2  1.0000  0.      0.      -10.000  REFL
3  .22000  .19500E-01  -3.9000  -2.9333  REFL
4  0.      .10000E+00  .50000  -1.1210  IMAG

SPACE  N      U      UBAR
1  1.0000  .10000E-07  .50000E-02
2  -1.0000  .20000  -.50000E-02
3  1.0000  -.50000E-01  .18295E-01

F/= 10.000  FULL FIELD(DEG)= .57296  MAG=-.20000E-06

OPTI  CC      AD      AE      AF      AG
1  -1.0353
2  -3.1576

RESIDUAL BLUR ON IMAGE SURFACES
CENTROID  DIAMETER
0.  .99928E-01  .56191E-06
.99928E-01  .98050E-04

UPPER BOUND:
H      Z
1  .50000E+06  -.10000E+09  OBJCT
2  1.0000  -.49995E-01  REFL  STOP
3  .24044  -3.9098  REFL
4  .99911E-01  .49457  IMAGE

LOWER BOUND (OBSCUR=.33623):
H      Z
1  0.  -70.670  AXIS
2  .33623  -3.4231  STOP
3  .31914  -.50926E-02  REFL
4  .50697E-01  -3.9004  REFL
5  0.  -2.4210  AXIS

INTERSECTIONS:
H      Z
1  .33623  -3.4231  UPPER LINE 2 WITH LOWER LINE 1
2  .18019  -2.0214  UPPER LINE 3 WITH LOWER LINE 3

```

Fig. 9. Output from GOSBOP for a Typical Cassegrain Telescope.

Note that the program generates an optical figure-of-merit in the form of an approximate blur diameter for the image of an on-axis object point and a point at the extreme edge of the field. This allows the user to monitor the image quality so that he or she can determine the tradeoffs between optical performance and baffle structure. The blur diameter is defined here as the greatest difference in the image surface heights of the meridional rays used in determining the beam. In this case, GOSBOP also determined the axial location and curvature of the image surface that minimizes the RMS spot size due to third-order aberrations.

If the obscuration ratio determined by the program is too high because of, say, diffraction effects, then one would have to vary some first-order optical property of the system in order to achieve the same baffle alignment for a smaller obscuration. The method employed in GOSBOP for varying the first-order properties of an arbitrary optical system is based upon experience with the $Y-\bar{Y}$ diagram (Delano, 1963; Shack, 1973). Essentially, the $Y-\bar{Y}$ diagram depicts a skew ray, whose orthogonal projections are the marginal and chief rays, as it would appear when viewed down the optical axis. A constraint placed on an optical design will many times result in confining a point that represents an optical surface to move along some fixed line on the diagram. For example, suppose we would like to move the final image plane in our Cassegrain

system keeping the primary focal length, and therefore the secondary magnification, fixed. On the $Y-\bar{Y}$ diagram of Fig. 10 the focal length of the primary mirror is proportional to the shaded area. Therefore, the secondary mirror point must remain on the line between the primary mirror and the primary image.

This particular variation was attempted on GOSBOP for an obscuration ratio of 0.3 but the program failed to converge to a solution. At first it was thought that there was something wrong with the program code or the author's modification of the Newton-Raphson algorithm. However, a clue to the cause of the divergence was provided by the results of our approximate analytical solution as shown in Fig. A.3 of Appendix A. There is a position of the final image plane for which the obscuration is minimum, and since we have found this analytical prediction to be low, it is possible that this minimum obscuration ratio is greater than 0.3. A series of GOSBOP runs were generated in which the obscuration ratio was again used to achieve the baffle alignment, but the final image position was varied for each run. The results in Fig. 11 show that indeed the actual minimum obscuration ratio is greater than 0.3 whereas the analytical minimum is below this. By first estimating the actual minimum to occur at a back working distance of -1.625 , we can then use GOSBOP to calculate the precise obscuration ratio at this point, i.e., 0.30926.

Fig. 10. Varying the Position of the Image Plane in a Cassegrain for a Fixed Primary and Overall Focal Length.

Fig. 11. There Exists an Image Plane Location That Minimizes the Required Obscuration Ratio.

As we just learned, failure to converge to a valid solution in a reasonable number of iterations usually indicates that one is trying to do the impossible, and a different method of attack should be taken. For instance, we could move the secondary mirror point on the $Y-\bar{Y}$ diagram along another line that does not necessarily keep the secondary magnification fixed. A line through the origin of the diagram would be the simplest (Fig. 12). This corresponds to keeping the ratio of the mirror separation to the back focal distance constant so that the telescope maintains its relative appearance. In this case GOSBOP did converge to a solution for an obscuration ratio of 0.3 and a primary f-number of 1.971.

The most important properties of the three above computer solutions to a minimally baffled Cassegrain telescope are summarized in Table 1 and shown in Fig. 13 with entrance ports located at their primary focii. The program has also been applied successfully to a variety of other optical systems including Gregorian, three-mirror, four-mirror, and catadioptric telescopes.

Fig. 12. One Way of Varying the Secondary Magnification on the Y - $\bar{Y}$ Diagram.

Table 1. Properties of the Three GOSBOP Solutions to an F/10 Minimally Baffled Cassegrain.

Entrance Pupil Diameter = 2; Full Field-of-view = 0.01 Radians

		1	2	3
Obscuration Ratio	e	.33623	.30926	.3
Secondary Magnification	m	4	4	5.0742
Back Working Distance	z_3	.5	-1.625	.41341
Primary Mirror:				
Radius of Curvature	R_1	-10	-10	-7.883
Conic Constant	CC_1	-1.0353	-1.0195	-1.0173
Inner Height	h_1	.31914	.29148	.28585
Secondary Mirror:				
Vertex Axial Location	z_2	-3.9	-4.325	-3.2245
Radius of Curvature	R_2	-2.9333	-1.8	-1.7859
Conic Constant	CC_2	-3.1576	-3.1203	-2.4062
Outer Height	H_2	.24044	.15733	.19904
Secondary Baffle Edge	r_s, z_s	.33623, -3.4230	.30926, -3.5598	.30000, -2.8357
Conical Baffle Edge	r_c, z_c	.18019, -2.0214	.12203, -2.6785	.15643, -1.6719

Fig. 13. Relative Appearances of the Three Designs.

CHAPTER 4

EVALUATING A DESIGN

When trying to arrive at a design with acceptable background levels in the detector plane, it is not enough for the designer to have at hand a computer program that determines the location of the baffles needed to block certain stray radiation propagation paths. There are two vital pieces of information missing. First, a particular design needs to be characterized by some sort of figure-of-merit so that it cannot only be compared with other designs but also with the performance requirements. Trying to optimize the baffle design of a system with just the GOSBOP program would be like trying to design a lens system with a program that yields the curvatures, thicknesses, and aspherics but no measure of the image quality such as aberration tables, ray fans or spot diagrams. Secondly, the use of GOSBOP to block propagation paths presupposes that the user knows what paths are involved and their relative importance so that he or she can instruct the program to align the baffles to block the most important paths. Both of these missing pieces can be filled in by doing an analysis of the design on any one of the big stray radiation computer programs. Because the author has had some part in its development and is, therefore, most

familiar with it, this thesis will deal with only one of these programs; APART (Arizona's Paraxial Analysis of Radiation Transfer).

The APART Computer Program in General

Unlike the statistical ray trace programs, APART uses a deterministic approach which is basically a numerical integration of the energy scattered from the internal surfaces of the system (Lange, Breault, and Greynolds, 1977). Each surface is divided into small sections. Scattered energy is transferred from one section to another until radiation originating at the external source finally reaches the detector surface. If the radiation must pass through one or more imaging elements then the collector section is imaged in a Gaussian or ideal manner. The program keeps track of the amount and origin of the power falling onto each section in the system including those on the detector plane. From this information a suitable figure-of-merit and the major propagation paths can be determined.

Like all stray radiation programs APART is more difficult to use than a lens design program because there are additional physical processes that must be modeled, and, therefore, much more information that must be fed into the computer, i.e., not only the optical prescription but also the locations and scattering characteristics of all non-imaging surfaces. In addition, most of the programs require

the user to also specify what propagation paths the computer is to consider in its calculations. Although some of the programs can be operated in a "one shot" manner and consider all paths, the computational time is immense and thus, in many cases, prohibitive. APART requires the user to set up the propagation paths but provides sufficient information to the user to allow one to intelligently pick those paths that should be most important. This method has both its advantages and disadvantages. It is computationally more efficient and forces the user to understand the underlying mechanisms better, but requires more effort from the user and opens the analysis to an additional possibility of human error. Experience has shown that in most cases, major errors in an analysis are due to the omission of a significant propagation path.

The Mechanics of Using APART

APART is a semi-interactive program divided into three separate batch subprograms. The geometry of the optomechanical system is entered into the first subprogram, APART1. For complicated systems this can require a significant effort by the user because dimensions on a blueprint must be converted to the absolute coordinate system used by the program. However, this work is avoided when using the GOSBOP design program since it creates an APART1 input file by placing baffle surfaces along the boundaries of the signal beam.

In the second subprogram, the user specifies from what surfaces to what other surfaces energy will be transferred. APART2 then calculates the projected solid angle that each collector section subtends from each source section, and the angles out of the source and into the collector section. Besides specifying the source and collector surfaces, the user must also designate the surfaces that might possibly block the line of sight between the source and collector sections.

All this necessary input to APART2 is generated by the user with the aid of two forms of output from APART1. The first is a graphical two-dimensional representation of the surfaces in the system as imaged into a particular space. For example, Fig. 14 is a plot of the surfaces in a Cassegrain as seen from the first or object space. From this the user can determine the surfaces illuminated by a distant off-axis source. Likewise, the objects seen by a detector in the image plane can be determined from a plot of the surfaces imaged into the final space (Fig. 15). Only these surfaces can transfer energy to the detector. Intermediate transfers between the illuminated and seen areas can be set up using plots of the intermediate image spaces. Although the plots are indispensable when thoroughly analyzing a system, they are sometimes difficult to interpret, especially for the novice user. For instance, they are a two-dimensional

Fig. 14. Surfaces in a Cassegrain Imaged into the First or Object Space.

Fig. 15. Surfaces Imaged into Third and Final Space.

representation of a three-dimensional system. This can sometimes lead to conceptual problems for the user. Secondly, the lines on a plot can become confusing. One object may have to be split into two when its image passes through infinity, i.e., the object itself passes through a focus. Also, images of different objects may overlap on the same plot. To remedy this, APART1 was given the capability of scanning the final image space from a designated point in order to determine the precise locations of the areas seen by a detector.

The scattering characteristics of each surface are entered into the third subprogram by means of either a table of measured data or any one of the scattering models available. The transfers calculated in APART2 are then connected in a sequence designated by the user to form multi-scattering paths from the external source to the detectors. Finally, APART3 calculates the final distribution of scattered energy and prints out the contribution from each section in the system.

A New Version of APART1

Even with the geometrical input already provided by GOSBOP, the analysis of a preliminary design could take several hours using the APART program. What is needed is a quick stray radiation calculation that can be done in the same batch run as the GOSBOP calculation and with a minimal

amount of user effort. To accomplish this a new version of the first APART subprogram was developed that can analyze all single scattering paths in a GOSBOP design given only the scattering characteristics of the internal surfaces, the angular position of the distant off-axis point source, and the detector location. This calculation was made possible by expanding the scanning capability of APART1 to include an arbitrary observation point in any space. Therefore, the subprogram can both look into the system from the point source to determine the illuminated areas and look out from the detector to determine the final scattering areas. These areas are then matched up to determine if they overlap. If they do, the relative amount of energy scattered to the detector is calculated. In the case of multiple scattering paths, the user must resort to a complete APART analysis. However, a quantitative knowledge of the usually more important single scattering paths can be very useful when doing a first pass optimization of a particular design or comparing radically different designs.

APART1 can handle multiple scattering in the cavities between vanes. Like APART, the subprogram assigns the fictitious baffle surface formed by the edges of the vanes a special scattering characteristic that effectively mimics the actual multiple scattering. In addition, APART1 can determine for the designer the angle, depth, and separation

of the vanes that minimizes both the scatter and the number of vanes on the surface.

Since the stray radiation analysis technique employed in APART1 is fundamentally different from the numerical integration method of the complete APART program, a cross-check analysis was performed on our first GOSBOP solution to a minimally baffled Cassegrain telescope found in Chapter 3. The baffles and optical surfaces were assumed to have Lambertian scattering characteristics with total hemispherical reflectivity of 0.01 and 0.0001, respectively. These reflectivities correspond to near state-of-the-art surfaces operating in the infrared. APART1 possesses more complicated and realistic scattering models. However, besides being considerably more efficient from a computational standpoint, a Lambertian model is sufficient for our purposes. The program cannot at this time calculate the stray radiation contributions due to diffraction from apertures and obstacles in the system. Experience indicates that these effects are usually important only at the far infrared wavelengths.

The results of this comparative analysis are shown in Fig. 16, a plot of the Point Source Transmittance (PST) as a function of ten angular positions of the distant off-axis point source. The PST is defined as the irradiance at the center of the image plane divided by the irradiance (normal to the point source direction) incident upon the

Fig. 16. Comparison of APART1 and APART Analyses of a Typical Cassegrain Telescope.

instrument. All the curves have a rather prominent hump in the 8 to 20 degree region because the inner conical baffle is directly illuminated. The choppiness of the one APART curve is due to the quantum nature of the numerical integration technique, i.e., the radiation from the point source walks on and off the sections of the inner conical baffle as the off-axis angle changes. To minimize this effect, each section was divided into nine smaller subsections. As a result the third PST curve is much smoother. All three curves agree as well as can be expected. More importantly, APART1 and APART agree on which are the dominant scattering paths at each angle. The main difference between the analyses is in the effort put forth by both the user and the computer. The GOSBOP-APART1 design and analysis required less than 3 seconds on the CYBER 175 and 35 input cards (see Fig. 17), considerably less than the more than 40 seconds and 200 cards for the complete APART analysis.

```

JOBNAME,BN12345678,CM67K,T3.
ATTACH(GOSBOP,GOSBOP,ID=ONE)
GOSBOP.
ATTACH(APART1,APART1,ID=SIX)
APART1.

```

7/8/9

```

* MINIMALLY BAFFLED RITCHEY-CRETIEN CASSEGRAIN TELESCOPE
FNUMBER 10
APERTURE 2 (ENTRANCE PUPIL DIAMETER)
FOV .01 RAD (.573 DEGREE FULL FIELD)
REFL 0. (PRIMARY MIRROR)
RD -10 (VERTEX RADIUS OF CURVATURE)
CC SOLVE (SOLVE FOR CONIC CONSTANT)
REFL -3.9 (SECONDARY MIRROR)
CC SOLVE
SURF (FINAL IMAGE PLANE)
BEST (FIND MINIMUM RMS SPOT SIZE IMAGE SURFACE)
APSTOP 1 (APERTURE AT PRIMARY MIRROR)
OBSTOP 1 -3.4718 ZINTER 1 (OBSCURATION AT SECONDARY BAFFLE)
DBSCUR .30563 (STARTING GUESS)
VARY DBSCUR
ALIGN INTER 1 INTER 2 UPPER -4 (MINIMAL BAFFLING CONDITION)
INTERSECT UPPER 2 LOWER 1 (SECONDARY BAFFLE)
INTERSECT UPPER 3 LOWER 3 (CONICAL BAFFLE)
ZPORT -5 (ENTRANCE PORT AT PRIMARY FOCUS)
EQUATE HLOWER 4 0. (NO HOLE IN SECONDARY MIRROR)
APART (GENERATE APART INPUT DECK)
STOP

```

7/8/9

```

SANGLE 2 4 6 8 10 12 14 16 18 20 (SOURCE ANGLES IN DEGREES)
COATING BAFFLES DIFFUSE .01 (LAMERTIAN BAFFLE SURFACES)
COATING OPTICS DIFFUSE .0001 (LAMBERTIAN OPTICAL SURFACES)
READ (READ DECK GENERATED BY GOSBOP)
XEQ

```

6/7/8/9

Fig. 17. GOSBOP-APART1 Input Deck for the Design and Analysis of a Cassegrain.

CHAPTER 5

OPTIMIZATION OF A DESIGN

Besides a PST figure-of-merit, the most useful information provided by APART1 is a summary of the dominant single scattering paths at each angle. This output for the Cassegrain analyzed in the last chapter is shown in Fig. 18. From this the user can determine what path needs to be eliminated in order to improve the instrument's performance most significantly. He or she can then return to the GOSBOP program to either align the baffles or shift the stops. This procedure of design, evaluation, and redesign can be repeated until the system meets spec or all single scattering paths, except for those involving scatter off the optical surfaces, are eliminated. Depending on the number of optimization iterations needed, a typical system can be optimized with GOSBOP and APART1 in a few hours.

Cassegrain Example

First Iteration

As we discovered from Fig. 18, the path involving the directly illuminated inner conical baffle dominates over most of the angles. We could reduce the effective reflectivity of this surface by placing vanes on it or we could

SUMMARY FOR DETECTOR POSITION (0. 049990)									
SOURCE	ANGLE	PST	PERCENT	MAJOR CONTRIBUTOR	DESCRIPTION				
				OBJN					
1	2.0	.19086E-04	88.68	2.	INNER	SECONDARY	BAFF	CONE	
2	4.0	.15675E-04	96.09	2.	INNER	SECONDARY	BAFF	CONE	
3	6.0	.10287E-05	75.10	2.	INNER	SECONDARY	BAFF	CONE	
4	8.0	.16303E-04	98.56	3.	INNER	CONICAL	BAFFLE	CONE	
5	10.0	.82023E-05	97.59	3.	INNER	CONICAL	BAFFLE	CONE	
6	12.0	.61698E-05	96.94	3.	INNER	CONICAL	BAFFLE	CONE	
7	14.0	.42520E-05	95.97	3.	INNER	CONICAL	BAFFLE	CONE	
8	16.0	.37217E-05	96.18	3.	INNER	CONICAL	BAFFLE	CONE	
9	18.0	.17554E-05	93.81	3.	INNER	CONICAL	BAFFLE	CONE	
10	20.0	.15852E-05	96.22	3.	INNER	CONICAL	BAFFLE	CONE	

Fig. 18. APART1 Output Showing the Major Contributor at Each Source Angle.

prevent radiation from reaching it by varying the obscuration to align the two baffle edges with the edge of the main baffle. As one can see from Fig. 19, the baffle alignment was more effective. Even though this solution required an increase in obscuration ratio from 0.33623 to 0.40652, the resulting decrease in signal is more than offset by the reduction in stray radiation level so that overall the signal-to-noise ratio increases.

Second Iteration

The major paths in the new design involve the inner secondary baffle at the small angles and the area of the main baffle near the primary mirror at the larger ones. Both of these surfaces can be eliminated from view of the detector by shifting the aperture stop to the aperture formed by the end of the conical baffle. There are drawbacks to shifting the stop this close to the image plane. The clear aperture of the primary mirror must be enlarged to prevent vignetting for the same entrance pupil size. Also, the field aberrations tend to, in general, worsen. However, the oversizing of the primary is tolerable for the small field-of-view of this system and since the system has been made aplanatic by aspherizing, the only third-order aberration affected by the stop shift is distortion.

Fig. 19. Results of the Two Methods for Reducing the Inner Conical Baffle Path.

Third Iteration

After the aperture stop has been shifted, the major contributor becomes the outside of the conical baffle. By making the end of the conical baffle the obscuration stop also, the outer conical baffle can no longer be seen in reflection through the secondary mirror. Remember that during the shifting of the stops, the obscuration ratio must be continually varied in order to maintain the alignment of the three baffle edges. This final design is "optimum" in the sense that the remaining dominant paths involve scattering off of optical surfaces only.

The PST curves corresponding to each of the preliminary designs leading to and including this optimum design are shown in Fig. 20. The baffle alignment, stop locations, obscuration ratio, and dominant path for each of these optimization iterations are shown in Fig. 21. In three iterations and less than an hour of the author's time, the stray radiation of the starting design was reduced by nearly three orders of magnitude at the larger off-axis angles. Before the creation of the GOSBOP-APART1 computer code, the same task required several man-days of effort.

Cassegrain with a Lens Reimager

The addition of a reimaging system to an instrument is a standard stray radiation suppression technique. This makes it possible to place the stops very close to the final

Fig. 20. Reduction of the PST at Each Optimization Iteration.

Fig. 21. Configuration of Design at the End of Each Iteration.

e = Obscuration ratio
 A = Aperture stop
 O = Obscuration stop

image plane without oversizing the optical components. It also allows the introduction of a field stop. Figure 22 is a comparison of a Cassegrain equipped with a one-to-one reimager (Fig. 23) with our optimum Cassegrain design. Since both are limited by scattering from the optical surfaces, one would expect them to have comparable stray radiation characteristics. However, the addition of the reimager complicates the system both optically and mechanically, adding to its expense without significantly improving its performance over that of our final Cassegrain design.

Fig. 22. Comparison of the Final Design with the Reimager Design.

Fig. 23. A Cassegrain Telescope Equipped with a Reimaging Lens.

CHAPTER 6

SUMMARY AND SUGGESTIONS FOR FUTURE RESEARCH

Given the ability to calculate the magnitude of the stray radiation and the relative contributions from all scattering paths, it is a straightforward task for the optical systems designer to significantly reduce the stray radiation. The GOSBOP-APART1 code is an easy-to-use tool that minimizes the designer's time and effort. The code is limited to the design and analysis of axially symmetric optical systems and single scattering paths. Many state-of-the-art systems are asymmetric and limited at some off-axis angles by multiple scattering. In addition, at infrared wavelengths and longer, diffraction, instead of scattering, can be the dominant mechanism. Future research should be aimed at removing these restrictions.

APPENDIX A

APPROXIMATE PARAXIAL SOLUTION TO THE BAFFLING OF A CASSEGRAIN TELESCOPE

The limited analysis presented here was done basically for two reasons: to develop an understanding of the interaction between the optical and baffle parameters in a Cassegrain and to provide starting values for GOSBOP that could be easily done on a hand calculator. Although it is possible to derive much more accurate closed-form equations for the baffling of a Cassegrain telescope, they are much too complicated to be of practical use. It is easier to go ahead and find an exact solution with the GOSBOP computer program.

Given the marginal ray, a chief ray, the obscuration ratio, and the axial locations of the aperture and obscuration stops we could now write down the equations for the heights of the beam at the two mirrors and the final image plane using Eqs. (26) and (27) of the main text. But before we do the necessary substitutions, the equations for the boundaries of the mirrors can be simplified considerably by noting that the second term in each equation, i.e., the magnitude of the new chief ray height, will be proportional to the semi-field angle. Assuming that this angle is small, and that the stops

are confined to the first two spaces and not too far removed from the system, then the boundary heights for the two mirrors can be reasonably approximated by their marginal ray heights.

$$H_1 \approx Y_1 \quad (\text{A.1})$$

$$h_1 \approx e Y_1 \quad (\text{A.2})$$

$$H_2 \approx Y_2 \quad (\text{A.3})$$

$$h_2 \approx e Y_2 \quad (\text{A.4})$$

In other words, the two mirrors will be allowed to vignette slightly for off-axis points in the image plane. However, we do not assume that the final image height is negligible so that

$$H_3 = \bar{Y}_3 \quad (\text{A.5})$$

Under the above conditions, the boundary equations are independent of the stop locations and Fig. A.1 is a valid representation of the baffled system. For mathematical simplicity, a normalized cylindrical coordinate system is used such that the focal length and marginal ray height of the primary mirror are unity and the origin is at its vertex. However, the results of the analysis are applicable to a Cassegrain of any f-number. To get the actual coordinates, multiply the normalized height by the actual entrance aperture height and the normalized z-coordinate by the primary focal length.

Fig. A.1. Definition of Coordinate System and Parameters Used in the Approximate Analysis of a Baffled Cassegrain Telescope.

The optical system can be uniquely characterized by three parameters; the secondary magnification m , the back working distance d , and the final image height I . Therefore, at the final image plane

$$\bar{Y}_3 = I \quad (\text{A.6})$$

$$Z_3 = d. \quad (\text{A.7})$$

The marginal ray height of the secondary mirror and the mirror separation, t , can be found by simultaneously solving the equations for lines A and B in Fig. A.1.

$$\begin{cases} r = Z + 1 \\ r = \frac{d-Z}{m} \end{cases} \quad (\text{A.8})$$

$$r = \frac{d-Z}{m} \quad (\text{A.9})$$

$$Y_2 = \frac{1+d}{m+1} \quad (\text{A.10})$$

$$Z_2 = -t = -\frac{m-d}{m+1} \quad (\text{A.11})$$

The input to the GOSBOP program requires us to be able to calculate the focal lengths or radii of the mirrors from the normalized optical parameters. The radius of curvature of the primary mirror is by definition

$$R_1 = -2f_1 = -2. \quad (\text{A.13})$$

The secondary focal length can be found by applying the lens law since the secondary mirror images the primary image into the final. To do this, we note that the back focal distance is just equal to the magnification times the marginal ray height at the secondary mirror in our normalized coordinate

system. Therefore,

$$\frac{1}{f_2} = \frac{1}{m\left(\frac{1+d}{m+1}\right)} + \frac{1}{\left(\frac{m-d}{m+1}\right) - 1} \quad (\text{A.14})$$

and

$$R_2 = 2f_2 = - \frac{2m(1+d)}{m^2 - 1} . \quad (\text{A.15})$$

We now have all the information we need in order to find the locations of the two baffle edges. Substituting $r = e$ into the equation for line A yields the secondary baffle edge z -coordinate.

$$(r_s, z_s) = (e, e-1) . \quad (\text{A.16})$$

Finally, the conical baffle edge is determined by the intersection of lines C and D.

$$\begin{cases} r = e(z+1) & (\text{A.17}) \\ r = \frac{[I(m+1) - (1+d)](z-d)}{m(1+d)} + I & (\text{A.18}) \end{cases}$$

$$(r_c, z_c) = \frac{(e(1+d)(1+d-I), -(1+d)(em-d) + I(m-d))}{(1+d)(em+1) - I(m+1)} . \quad (\text{A.19})$$

In general, it will be desirable to have the edges of these two baffles line up with some other point in the system. To do this, we first write down the equation of the line through the two baffle edges:

$$r = \frac{(r_c - e)(z + 1 - e)}{z_c + 1 - e} + e . \quad (\text{A.20})$$

If the point (r, z) is now fixed, then the above becomes a constraint equation that can be solved for one system parameter given the other three. For example, given the three optical parameters, m , d , and I , then the equation can be written as a quadratic in the one baffle parameter e and solved using the quadratic formula.

$$e = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

$$A = (1+d)[(1+Z-r)m + (1+d-I)]$$

$$B = [1+r+d(2+Z)](1+d-I) + I(1+Z-r)m$$

$$C = r(1+d)(1+d-I) . \quad (A.21)$$

The selection of the proper root depends on the values of r and z . In general, the smallest positive root is the correct one.

The minimum baffling condition, in which no direct object space radiation reaches a final image of height I , requires that:

$$(r, Z) = (-I, d) . \quad (A.22)$$

In addition, if $I = 0$, then the obscuration formula reduces to:

$$e = \frac{(1+d)}{m+1} = Y_2 . \quad (A.23)$$

In other words, the secondary mirror itself becomes the limiting central obscuration, and there is no secondary baffle. This is also the absolute minimum allowable

obscuration ratio under any circumstances in a Cassegrain (see Fig. A.2).

Although the equations for the baffle edge locations are of limited accuracy, the above analysis can still be used to baffle actual systems because the alignment of these edges is handled exactly. We first solve Eq. (A.21) for the obscuration ratio that yields the desired baffle alignment. We then substitute this value into Eqs. (A.16) and (A.19) to find the baffle locations. The system will then be properly baffled but some off-axis vignetting will be present.

Another advantage of Eq. (A.21) is that it gives us an idea of how, in general, e varies as a function of m , I , and d . Figures A.3 and A.4 are plots of the required obscuration as a function of these three parameters for the minimum baffling condition specified by Eq. (A.22). From Fig. A.3, we see that the obscuration decreases with increasing secondary magnification for most image locations. But of most interest is the fact that there is a position of the final image plane for which the obscuration is minimum. In Fig. A.4 the obscuration increases as the height of the image increases.

Fig. A.2. Absolute Minimum Obscuration Ratio in a Cassegrain.

Fig. A.3. Required Obscuration Ratio for a Minimally Baffled Cassegrain with a Relative Image Height of 0.1.

Fig. A.4. Obscuration Ratio for a Minimally Baffled Cassegrain with a Normalized Image Plane Location of 0.1.

APPENDIX B

LISTING OF THE CYBER FORTRAN EXTENDED CODE FOR THE GOSBOP COMPUTER PROGRAM

```

PROGRAM GOSBOP (INPUT=65, OUTPUT=130, TAPE1=INPUT, TAPE4=OUTPUT,
1 TAPE21=65, PLOT=130, TAPE13=PLOT)
COMMON /CBEAM/ ZUPPER(20), HUPPER(20), ZLOWER(20), HLOWER(20),
1 ZINTER(20), HINTER(20), ZPTS(20), HPTS(20), NUPPER, NLOWER, NINTER,
2 LINES(20,2), ZPORT(2), KUPPER(20), KLOWER(20), MOPTIC(20,3)
3 , NWARN, IWARN(20,2), NAMINT(2,20)
COMMON /CSTOP/ ZASTOP, ZOSTOP, KASTOP, KOSTOP, LASTOP, LOSTOP,
1 PDZSTP(2)
COMMON /COPTIC/ OBSCUR, FOVDEG, DAPERT, YSURF(20), YBSURF(20,2)
1 , ZHE, RINDEX(20), ZVSURF(20), NSURF, NSPACE, NOPTI, NAMSUF(2,20)
2 , OMEGA(20), OMEGAB(20), RDSURF(20), KFIGR(4), NFIGR, KSFIGR, NSFIGR
3 , CVSURF(20), CCSURF(20), KSURF(20), ASURF(20,4), NASURF(20)
COMMON /CVARY/ KVARIB(3), LVARIB(3), SVARY(3), YVARY(3), FDIFR(5,5),
1 YBVARY(3), KQUANT(3,10), XDIFR(5,5), PQUANT(3), NVARY, NSOLV, ITER, DEL
2 , KSPACE(3,10), INDVAR
COMMON /CEXTRA/ BIG, SMALL, CONV, FNUMBR, APTFAC, FOVFAC, WAVEL, IMBEST,
1 NITER, DELMAX, XEQ, ITRACE, KBEAM, DELONE, DVMAX, NINT, KDIM, OBFOV,
2 IVIGN, TOLER, IPRINT, DAXIS, DFIELD, HFIELD, SMAG, MASKS(2), HSEC, ZSEC
3 , HPLOT
COMMON /H/ IHID(8,10), NCMNTS, KI, KO

```

```

C
C "GOSBOP" (GENERAL OPTICAL SYSTEM BAFFLE OPTIMIZATION PROGRAM)
C THIS PROGRAM BAFFLES A GENERAL SYSTEM IN WHICH THE APERTURE
C AND OBSCURATION STOPS ARE LOCATED ARBITRARILY OR AT A BAFFLE TIP.
C VERSION 1 CAPABILITIES:
C 1. AXIALLY SYMMETRIC OPTICAL SYSTEMS
C 2. AXIALLY SYMMETRIC OR MERIDONAL PLANE ECCENTRIC PUPIL BAFFLING
C 3. CONIC AND GENERAL ASPHERIC OPTICAL SURFACES
C 4. ARBITRARY STOP LOCATIONS
C 5. CAN VARY OBSCURATION RATIO, FIELD OF VIEW, APERTURE OR THE
C Y-YBAR VALUES OF ANY ONE ELEMENT IN ORDER TO ALIGN OR EQUATE
C CERTAIN BAFFLE POINTS
C 6. CORRECTION OF THIRD-ORDER ABERRATIONS AND UP TO NINTH ORDER
C SPHERICAL BY ASPHERIC SOLVES
C 7. GENERATES APART INPUT
C 8. PLOTS BOUNDARY RAYS
C

```

```

DATA KSURF/5ROBJCT,19*0/
KDIM=20
BIG=1.E8
SMALL=1.E-8
DELMAX=1.E-4
CONV=3.1415926535/180.
NSURF=NSOLV=NVARY=NFIGR=LASTOP=LOSTOP=NITER=KBEAM=NINT=0
KSFIGR=NWARN=IMBEST=ITRACE=IPRINT=INDVAR=0
IVIGN=1
KASTOP=KOSTOP=1
XEQ=ZPORT(1)=ZPORT(2)=ZASTOP=ZOSTOP=OBSCUR=HSEC=ZSEC=0.
WAVEL=TOLER=DAXIS=HPLOT=0.
OBFOV=FOVFAC=APTFAC=1.
DAPERT=2.
DVMAX=DELONE=.1

```

```

5  WRITE(4,1) DATE(XX),TIME(XX)
1  FORMAT(1H1,*GOSBOP (VERSION 1) *,2A10)
    NS=NSURF
    FNUMBR=0.
    CALL READIN
    IF (NS.GT.0) NSURF=NS
    ITER=0
    YSURF(2)=DAPERT/2.
    IF (KBEAM.GT.0) YSURF(2)=YSURF(2)+OBSCUR
    IF (ITRACE.NE.0) CALL PTRACE
    KSURF(NSURF)=5RIMAGE
    IF (ABS(YBSURF(1)).LT.SMALL) GO TO 110
    T12=ZVSURF(2)-ZVSURF(1)
    IF (T12.EQ.0.) STOP "FIRST TWO SURFACES COINCIDE"
    ZHE=RINDEX(1)*(YSURF(1)*YBSURF(2)-YSURF(2)*YBSURF(1))/T12
    IF (NFIGR.GT.4) NFIGR=4
    CALL YYBCAL
    CALL BEAM
    CALL INTPTS
    IF (NVARY.EQ.0) GO TO 110
    IF (IPRINT.NE.0) CALL SHOW
    DO 20 I=1,NVARY
    YVARY(I)=YVARY(I)*APTFAC
20  YBVARY(I)=YBVARY(I)*FQVFAC
    FQVFAC=APTFAC=1.
    ITER=1
    CALL CHANG1
    CALL YYBCAL
    CALL BEAM
    CALL INTPTS
    IF (IPRINT.NE.0) CALL SHOW
    IF (NITER.LT.2) NITER=10+NVARY
    DO 40 I=2,NITER
    ITER=I
    CALL CHANG2,RETURNS(100)
    CALL YYBCAL
    CALL BEAM
    CALL INTPTS
    IF (IPRINT.NE.0) CALL SHOW
40  CONTINUE
    GO TO 110
100  ITER=I-1
110  IF (HPLOT.NE.0.) CALL PLOTBS
    IF (NSOLV.GT.0) CALL SOLVE
    IF (IPRINT.EQ.0) CALL SHOW
    IF (XEQ.EQ.1.) GO TO 5
    IF (NWARN.GT.0) STOP "WARNINGS"
    IF (HSEC.NE.0.) CALL APRTIN
    STOP "NORMAL TERMINATION"
    END

```

```

SUBROUTINE READIN
COMMON /CBEAM/ ZUPPER(20),HUPPER(20),ZLOWER(20),HLOWER(20),
1 ZINTER(20),HINTER(20),ZPTS(20),HPTS(20),NUPPER,NLOWER,NINTER,
2 LINES(20,2),ZPORT(2),KUPPER(20),KLOWER(20),MOPTIC(20,3)
3 ,NWARN,IWARN(20,2),NAMINT(2,20)
COMMON /CVARY/ KVARIB(3),LVARIB(3),SVARY(3),YVARY(3),FQVFR(5,5),
1 YBVARY(3),KQUANT(3,10),XDIFR(5,5),PQUANT(3),NVARY,NSOLV,ITER,DEL
2 ,KSPACE(3,10),INDVAR
COMMON /CSTOP/ ZASTOP,ZOSTOP,KASTOP,KOSTOP,LASTOP,LOSTOP,
1 PDZSTP(2)

```

```

COMMON /COPTIC/ OBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1 ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2 ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3 ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1 NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2 IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3 ,HPLOT
COMMON /CREAD/ IMAGE(8),ICMNT(8),ITYPE(40),KARD(40),CARD(40),
1 NITEMS
COMMON /H/ IHID(8,10),NCMNTS,KI,KO

```

C
C
C

CHECKS THE INPUT CARDS AND READS INTO CORE THE INPUT DATA

```

DIMENSION LIST1(34),LIST3(3),LIST4(6),LIST5(6)
DATA LIST1/3HXEQ,5HSHIFT,4HSURF,6HOBSCUR,6HAPSTOP,6HOBSTOP,
1 5HZPORT,3HFOV,5HAPART,8HAPERTURE,6HCHANGE,9HINTERSECT,
2 4HVARY,5HNITER,6HDELMAX,4HSTOP,5HPRINT,7HFNUMBER,4HBEST,
3 5HINDEX,5HALIGN,6HEQUATE,5HSCALE,6HVIGNET,7HCOMMENT,4HPLOT,
4 6HY-YBAR,2HRD,4HREFL,2HCC,5HGLASS,4HOPTI,10HWAVELENGTH,
5 9HTOLERANCE/
DATA LIST3/5HUPPER,5HLOWER,5HINTER/
DATA LIST4/6HZUPPER,6HHUPPER,6HZLOWER,6HHLOWER,6HZINTER,6HHINTER/
DATA LIST5/6HOBSCUR,3HFOV,8HAPERTURE,6HY-YBAR,1HY,4HYBAR/
DATA IHID,NCMNTS,KI,KO/80*10H ,0,1,4/
NPTS=KILL=0
WRITE(4,1)
FORMAT(// * INPUT CARDS:*/)
10 CALL GENRED
IF (NITEMS.EQ.0) GO TO 10
IF (ITYPE(1).GE.0) GO TO 900
20 DO 30 I=1,34
30 IF (KARD(1).EQ.LIST1(I)) GO TO (210,50,60,70,80,90,100,
1 110,120,130,140,150,170,180,190,200,215,220,230,240,
2 250,260,270,280,40,340,290,300,60,310,60,60,320,330),I
WRITE(4,35)
35 FORMAT(/1H ,* KEYWORD NOT RECOGNIZED*)
GO TO 10
C..COMMENT
40 NCMNTS=NCMNTS+1
DO 45 I=1,8
45 IHID(I,NCMNTS)=IMAGE(I)
GO TO 10
C..SHIFT
50 KBEAM=1
IF (NITEMS.EQ.1) GO TO 900
IF (ITYPE(2).LT.0) GO TO 900
OBSCUR=ABS(CARD(2))
GO TO 10
C..SURF
60 IF (XEQ.EQ.1.) GO TO 67
NSURF=NSURF+1
NSPACE=NSURF-1
NOPTI=NSURF-2
NASURF(NSURF)=0
RINDEX(NSURF)=1.
DO 62 I=1,2
62 NAMSRF(I,NSURF)=ICMNT(I)
IF (NSURF.EQ.1) GO TO 65
RINDEX(NSURF)=SIGN(1.,RINDEX(NSPACE))
IF (ABS(RINDEX(NSPACE)).EQ.1.) GO TO 63

```

```

        KSURF(NSURF)=5LGLAS
        GO TO 65
63      KSURF(NSURF)=5LREFL
        RINDEX(NSURF)=-RINDEX(NSPACE)
65      ZVSURF(NSURF)=CVSURF(NSURF)=RDSURF(NSURF)=CCSURF(NSURF)=0.
        ASURF(NSURF,1)=0.
67      IF (NITEMS.EQ.1) GO TO 10
        IF (ITYPE(2).LT.0) GO TO 900
        IF (ITYPE(2).EQ.0) NSURF=KARD(2)
        IF (ITYPE(2).GT.0) ZVSURF(NSURF)=CARD(2)
        IF (NITEMS.EQ.2) GO TO 10
        IF (ITYPE(3).LE.0) GO TO 900
        ZVSURF(NSURF)=CARD(3)
        GO TO 10
C..OBSCUR
70      IF (NITEMS.LT.2) GO TO 900
        IF (ITYPE(2).LE.0) GO TO 900
        OBSCUR=ABS(CARD(2))
        IF (NITEMS.EQ.2) GO TO 10
        IF (ITYPE(3).LT.0) GO TO 900
        OBFOV=AMIN1(1.,ABS(CARD(3)))
        GO TO 10
C..APSTOP
80      IF (NITEMS.LT.2) GO TO 900
        IF (ITYPE(2).LT.0) GO TO 900
        KASTOP=KARD(2)
        LASTOP=-1-KASTOP
        IF (NITEMS.EQ.2) GO TO 10
        IF (ITYPE(3).LT.0) GO TO 900
        LASTOP=0
        ZASTOP=CARD(3)
        IF (NITEMS.EQ.3) GO TO 10
        IF (ITYPE(4).GE.0) GO TO 900
        IF (NITEMS.LT.5) GO TO 900
        DO 85 I=1,5,2
85      IF (KARD(4).EQ.LIST4(I)) K=KDIM*(I-1)
        IF (ITYPE(5).LT.0) GO TO 900
        LASTOP=K+KARD(5)
        GO TO 10
C..OBSTOP
90      IF ((NITEMS.LT.2).OR.(NITEMS.GT.5)) GO TO 900
        IF (ITYPE(2).LT.0) GO TO 900
        KOSTOP=KARD(2)
        LOSTOP=-1-KOSTOP
        IF (NITEMS.EQ.2) GO TO 10
        IF (ITYPE(3).LT.0) GO TO 900
        LOSTOP=0
        ZOSTOP=CARD(3)
        IF (NITEMS.EQ.3) GO TO 10
        IF (ITYPE(4).GE.0) GO TO 900
        IF (NITEMS.LT.5) GO TO 900
        DO 95 I=1,5,2
95      IF (KARD(4).EQ.LIST4(I)) K=KDIM*(I-1)
        IF (ITYPE(5).LT.0) GO TO 900
        LOSTOP=K+KARD(5)
        GO TO 10
C..ZPORT
100     IF (NITEMS.GT.3) GO TO 900
        IF (ITYPE(2).LT.0) GO TO 900
        ZPORT(1)=ZPORT(2)=CARD(2)
        IF (NITEMS.EQ.2) GO TO 10

```

```

IF (ITYPE(3).LT.0) GO TO 900
ZPORT(2)=CARD(3)
GO TO 10

```

C...FOV

```

110 IF (NITEMS.LT.3) GO TO 900
IF ((ITYPE(2).LT.0).OR.(ITYPE(3).GE.0)) GO TO 900
IF (KARD(3).EQ.3HDEG) FOVDEG=CARD(2)
IF (KARD(3).EQ.3HRAD) FOVDEG=CARD(2)/CONV
FOVDEG=ABS(FOVDEG)
IF (NITEMS.EQ.3) GO TO 115
IF (ITYPE(4).LE.0) GO TO 900
BIG=ABS(CARD(4))
SMALL=1./BIG
IF (DELMAX.EQ.1.E-4) DELMAX=SQRT(BIG)
115 YBSURF(1)=-BIG*FOVDEG*CONV/2.
IF (XEQ.EQ.1.) GO TO 10
NSURF=1
RINDEX(1)=1.
YSURF(1)=0.
ZVSURF(1)=-BIG
RDSURF(1)=0.
GO TO 10

```

C...APART

```

120 HSEC=ZSEC=-5
IF (NITEMS.EQ.1) GO TO 10
DO 125 I=2,NITEMS,2
IF (ITYPE(I)) 126,127,128
126 IF (I.EQ.NITEMS) GO TO 900
J=I+1
IF (ITYPE(J).LT.0) GO TO 900
IF (KARD(I).EQ.4HZSEC) GO TO 129
IF (KARD(I).NE.4HHSEC) GO TO 900
IF (ITYPE(J).EQ.0) HSEC=-KARD(J)
IF (ITYPE(J).GT.0) HSEC=CARD(J)
GO TO 125
129 IF (ITYPE(J).EQ.0) ZSEC=-KARD(J)
IF (ITYPE(J).GT.0) ZSEC=CARD(J)
GO TO 125
127 HSEC=ZSEC=-KARD(I)
GO TO 125
128 HSEC=ZSEC=CARD(I)
125 CONTINUE
GO TO 10

```

C...APERTURE

```

130 IF (NITEMS.LT.2) GO TO 900
IF (ITYPE(2).LT.0) GO TO 900
DAPERT=ABS(CARD(2))
GO TO 10

```

C...CHANGE

```

140 IF (NITEMS.LT.2) GO TO 900
IF (ITYPE(2).LE.0) GO TO 142
DELONE=CARD(2)
GO TO 144
142 INDVAR=1
144 IF (NITEMS.EQ.2) GO TO 10
IF (ITYPE(3).LE.0) GO TO 146
DVMAX=ABS(CARD(3))
GO TO 148
146 INDVAR=1
148 IF (NITEMS.EQ.3) GO TO 10
INDVAR=1

```

```

      GO TO 10
C..INTERSECT
150  IF (NITEMS.LT.5) GO TO 900
      I=2
      J=0
      NINT=NINT+1
      DO 151 K=1,2
151  NAMINT(K,NINT)=ICMNT(K)
152  J=J+1
      IF (J.GT.2) GO TO 10
      IF (ITYPE(I).GE.0) GO TO 160
      IP1=I+1
      IF (ITYPE(IP1).NE.0) GO TO 900
      IF (KARD(I).EQ.5HUPPER) LINES(NINT,J)=KARD(IP1)
      IF (KARD(I).EQ.5HLOWER) LINES(NINT,J)=-KARD(IP1)
      I=I+2
      GO TO 152
160  IP3=I+3
      DO 162 K=I,IP3
162  IF (ITYPE(K).LT.0) GO TO 900
      NPTS=NPTS+1
      LINES(NINT,J)=NPTS+6*KDIM
      ZPTS(NPTS)=CARD(I)
      HPTS(NPTS)=CARD(I+1)
      NPTS=NPTS+1
      ZPTS(NPTS)=CARD(I+2)
      HPTS(NPTS)=CARD(I+3)
      I=I+4
      GO TO 152
C..VARY
170  IF (NITEMS.LT.2) GO TO 900
      IF (ITYPE(2).GE.0) GO TO 900
      NVARY=NVARY+1
      IF (KARD(2).EQ.5HSHIFT) KARD(2)=6HOBSCUR
      DO 175 I=1,6
175  IF (KARD(2).EQ.LIST5(I)) KVARIB(NVARY)=MINO(I,4)
      YVARY(NVARY)=YBVARY(NVARY)=0.
      IF (KVARIB(NVARY).LT.4) GO TO 176
      IF (ITYPE(3).LT.0) GO TO 900
      KVARIB(NVARY)=3+KARD(3)
      IF (KARD(2).EQ.4HYBAR) KVARIB(NVARY)=KVARIB(NVARY)+KDIM
      LVARIB(NVARY)=KVARIB(NVARY)+KDIM
      IF (KARD(2).NE.6HY-YBAR) LVARIB(NVARY)=0
      IF (NITEMS.EQ.3) GO TO 176
      IF (NITEMS.LT.5) GO TO 900
      IF (ITYPE(4).LT.0) GO TO 900
      YVARY(NVARY)=CARD(4)
      IF (ITYPE(5).LT.0) GO TO 900
      YBVARY(NVARY)=CARD(5)
176  CALL GENRED
      IF (ITYPE(1).GE.0) GO TO 900
      KQ=NVARY
      IF (KARD(1).EQ.5HALIGN) GO TO 251
      IF (KARD(1).EQ.6HEQUATE) GO TO 261
      GO TO 900
C..NITER
180  IF (NITEMS.LT.2) GO TO 900
      IF (ITYPE(2).LT.0) GO TO 900
      NITER=IABS(KARD(2))
      GO TO 10
C..DELMAX

```

```

190  IF (NITEMS.LT.2) GO TO 900
      IF (ITYPE(2).LT.0) GO TO 900
      DELMAX=ABS(CARD(2))
      GO TO 10
C...STOP
200  XEQ=0.
      IF (NITEMS.EQ.1) GO TO 1000
      CALL SHOW
      STOP "INPUT CHECK"
C...XEQ
210  XEQ=1.
      GO TO 1000
C...PRINT
215  IPRINT=1
      IF (NITEMS.EQ.1) GO TO 10
      IPRINT=-1
      GO TO 10
C...FNUMBER
220  IF (NITEMS.LT.2) GO TO 900
      IF (ITYPE(2).LT.0) GO TO 900
      FNUMBR=ABS(CARD(2))
      GO TO 10
C...BEST
230  IMBEST=1
      GO TO 10
C...INDEX
240  IF (NITEMS.EQ.1) GO TO 10
      IF (ITYPE(2).LT.0) GO TO 900
      RINDEX(NSURF)=CARD(2)
      IF (NSURF.EQ.1) GO TO 10
      IF (RINDEX(NSURF)*RINDEX(NSPACE).LE.0.) GO TO 10
      KSURF(NSURF)=5LGLAS
      GO TO 10
C...ALIGN
250  NSOLV=NSOLV+1
      KQ=NSOLV+3
251  IF (NITEMS.LT.7) GO TO 900
      I=2
      L=0
253  L=L+1
      IF (L.GT.3) GO TO 10
      KSPACE(L,KQ)=0
      IP1=I+1
      IF (ITYPE(I)) 252,900,254
252  DO 256 J=1,3
256  IF (KARD(I).EQ.LIST3(J)) K=2*KDIM*(J-1)
      IF (ITYPE(IP1).LT.0) GO TO 900
      KQUANT(L,KQ)=ISIGN(K+IABS(KARD(IP1)),KARD(IP1))
      GO TO 255
254  NPTS=NPTS+1
      ZPTS(NPTS)=CARD(I)
      IF (ITYPE(IP1).LT.0) GO TO 900
      HPTS(NPTS)=CARD(IP1)
      KQUANT(L,KQ)=NPTS+6*KDIM
255  I=I+2
      IF (I.GT.NITEMS) GO TO 10
      IF (ITYPE(I).NE.0) GO TO 253
      KSPACE(L,KQ)=KARD(I)
      I=I+1
      GO TO 253
C...EQUATE

```



```

260  NSOLV=NSOLV+1
      KQ=NSOLV+3
261  DO 265 I=2,NITEMS,2
      IF (ITYPE(I).GE.0) GO TO 264
      DO 266 J=1,6
266  IF (KARD(I).EQ.LIST4(J)) K=KDIM*(J-1)
      IF (ITYPE(I+1).LT.0) GO TO 900
      KQUANT(I/2,KQ)=ISIGN(K+IABS(KARD(I+1)),KARD(I+1))
      GO TO 265
264  NPTS=NPTS+1
      ZPTS(NPTS)=CARD(I)
      KQUANT(I/2,KQ)=NPTS+6*KDIM
265  CONTINUE
      KQUANT(3,KQ)=0
      GO TO 10

C..SCALE
270  IF (NITEMS.LT.3) GO TO 900
      IF (NITEMS.GT.5) NITEMS=5
      DO 275 I=2,NITEMS,2
      IF (ITYPE(I).GE.0) GO TO 900
      IF (ITYPE(I+1).LT.0) GO TO 900
      IF (KARD(I).EQ.3HFOV) FQVFAC=CARD(I+1)
275  IF (KARD(I).EQ.8HAPERTURE) APTFAC=CARD(I+1)
      APTFAC=ABS(APTFAC)
      FQVFAC=ABS(FQVFAC)
      GO TO 10

C..VIGNET
280  IVIGN=0
      GO TO 10

C..Y-YBAR
290  YSURF(NSURF)=1.
      YBSURF(NSURF)=0.
      IF (NITEMS.EQ.1) GO TO 10
      IF (ITYPE(2).LT.0) GO TO 900
      YSURF(NSURF)=CARD(2)
      IF (NITEMS.EQ.2) GO TO 10
      IF (ITYPE(3).LT.0) GO TO 900
      YBSURF(NSURF)=KARD(3)
      GO TO 10

C..RD
300  IF (ITRACE.EQ.0) ITRACE=1
      IF (NITEMS.EQ.1) GO TO 10
      IF (ITYPE(2).LT.0) GO TO 900
      RDSURF(NSURF)=CARD(2)
      CVSURF(NSURF)=RDSURF(NSURF)
      IF (CVSURF(NSURF).NE.0.) CVSURF(NSURF)=1./CVSURF(NSURF)
      GO TO 10

C..CC
310  ITRACE=-1
      IF (NITEMS.EQ.1) GO TO 10
      NSFGR=0
      DO 315 I=2,NITEMS
      NASURF(NSURF)=I-2
      IF (ITYPE(I).GE.0) GO TO 318
      IF (I.GT.3) GO TO 319
      NFIGR=NFIGR+1
      KFIGR(NFIGR)=NSURF*(-1)**I
      GO TO 315
318  CCSURF(NSURF)=CARD(2)
      IF (NASURF(NSURF).GT.0) ASURF(NSURF,NASURF(NSURF))=CARD(I)
      GO TO 315

```

```

319  NSFIGR=NSFIGR+1
      KSFGR=NSURF
315  CONTINUE
      GO TO 10
C..WAVELENGTH
320  WAVEL=1.
      IF (NITEMS.EQ.1) GO TO 10
      IF (ITYPE(2).LT.0) GO TO 900
      WAVEL=ABS(CARD(2))
      GO TO 10
C..TOLERANCE
330  IF (NITEMS.EQ.1) GO TO 10
      IF (ITYPE(2).LT.0) GO TO 900
      TOLER=ABS(CARD(2))
      GO TO 10
C..PLOT
340  HPLLOT=3.
      IF (NITEMS.EQ.1) GO TO 10
      HPLLOT=-3.
      IF (ITYPE(2).LT.0) GO TO 10
      HPLLOT=CARD(2)
      GO TO 10
900  WRITE(4,950)
950  FORMAT(/1H ,* WRONG DATA TYPE,NUMBER OR SEQUENCE*)
      KILL=1
      GO TO 10
1000 IF (KILL.EQ.0) RETURN
      STOP "INVALID INPUT CARD(S)"
      END

```

SUBROUTINE GENRED

C..THIS ROUTINE IS A GENERAL PURPOSE READING PROGRAM FOR
C..READING UNFORMATTED INPUT CONTROL CARDS.

```

      COMMON /CREAD/ IMAGE(8),ICMNT(8),ITYPE(40),KARD(40),CARD(40),
1  NITEMS
      COMMON /H/ IHID(8,10),NCMNTS,KI,KD
      DIMENSION KCMNT(80),NPUT(80)
      DO 10 I=1,8
10  ICMNT(I)=10H
      DO 20 I=1,40
      KCMNT(I)=KCMNT(I+40)=0
20  ITYPE(I)=KARD(I)=CARD(I)=0.
      READ(KI,2000) IMAGE
      IF (EOF(KI).NE.0.) STOP "END OF INPUT FILE"
      WRITE(KD,2010) IMAGE
      DECODE(80,2020,IMAGE) NPUT
      DO 30 I=1,80
      MCOL=81-I
30  IF(NPUT(MCOL).NE.45) GO TO 40
40  NITEMS=0
      NCMNT=0
      ICOL=0
50  ICOL=ICOL+1
      IF(ICOL.GT.MCOL) GO TO 110
      IF((NPUT(ICOL).EQ.45).OR.(NPUT(ICOL).EQ.46)) GO TO 50
      IF((NPUT(ICOL).LT.39).OR.(NPUT(ICOL).GT.44)) GO TO 70
      IF(NITEMS.GT.0) GO TO 60
      NCMNT=MCOL-ICOL
      ICOL=ICOL+1
      ENCODE(NCMNT,2020,ICMNT) (NPUT(I),I=ICOL,MCOL)
      NITEMS=1

```

```

KARD(1)=7HCOMMENT
ITYPE(1)=-1
RETURN
60 ICOL=ICOL+1
IF(ICOL.GT.MCOL) GO TO 110
IF((NPUT(ICOL).GE.39).AND.(NPUT(ICOL).LE.44)) GO TO 50
NCMNT=NCMNT+1
KCMNT(NCMNT)=NPUT(ICOL)
GO TO 60
70 ITAB=ICOL
NITEMS=NITEMS+1
ITYPE(NITEMS)=0
IDECP=0
IEXP=0
LENGTH=0
IF(NPUT(ICOL).LE.26) ITYPE(NITEMS)=-1
80 IF(NPUT(ICOL).EQ.47) IDECP=1
IF(NPUT(ICOL).EQ.5) IEXP=1
LENGTH=LENGTH+1
ICOL=ICOL+1
IF(ICOL.GT.MCOL) GO TO 90
IF((NPUT(ICOL).LT.45).OR.(NPUT(ICOL).GT.46)) GO TO 80
90 IF(ITYPE(NITEMS).EQ.0) ITYPE(NITEMS)=IDECP+IEXP
IF(ITYPE(NITEMS).GE.0) GO TO 100
DECODE(ICOL,2030,IMAGE) ITAB,LENGTH,KARD(NITEMS)
GO TO 50
100 DECODE(ICOL,2040,IMAGE) ITAB,LENGTH,CARD(NITEMS)
KARD(NITEMS)=CARD(NITEMS)
GO TO 50
110 IF(NCMNT.EQ.0) RETURN
ENCODE(80,2020,ICMNT) KCMNT
RETURN
2000 FORMAT(8A10)
2010 FORMAT(2X,8A10)
2020 FORMAT(80R1)
2030 FORMAT(T=,A=)
2040 FORMAT(T=,E=.0)
END

```

SUBROUTINE SHOW

```

COMMON /CBEAM/ ZUPPER(20),HUPPER(20),ZLOWER(20),HLOWER(20),
1 ZINTER(20),HINTER(20),ZPTS(20),HPTS(20),NUPPER,NLOWER,NINTER,
2 LINES(20,2),ZPORT(2),KUPPER(20),KLOWER(20),MOPTIC(20,3)
3 ,NWARN,IWARN(20,2),NAMINT(2,20)
COMMON /COPTIC/ DBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1 ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NDPTI,NAMSRF(2,20)
2 ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3 ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
COMMON /CVARY/ KVARIB(3),LVARIB(3),SVARY(3),YVARY(3),FDIFR(5,5),
1 YBVARY(3),KQUANT(3,10),XDIFR(5,5),PQUANT(3),NVARY,NSOLV,ITER,DEL
2 ,KSPACE(3,10),INDVAR
COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1 NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,DBFOV,
2 IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3 ,HPLLOT
WRITE(4,1) ITER,ZHE
1 FORMAT(1H1,*ITERATION *,I2//1H ,*LAGRANGE INVARIANT=*,G11.5)
WRITE(4,10)
10 FORMAT(/1H ,*SURF*,3X,*Y*,10X,*YBAR*,9X,*ZV*,9X,*RD*,9X,*TYPE*)
DO 26 I=1,NSURF
IF ((RDSURF(I).EQ.0.).OR.(ABS(RDSURF(I)).GT.BIG)) GO TO 28

```

```

WRITE(4,30) I,YSURF(I),YBSURF(I,2),ZVSURF(I),RDSURF(I),KSURF(I)
GO TO 26
28 WRITE(4,32) I,YSURF(I),YBSURF(I,2),ZVSURF(I),KSURF(I)
26 CONTINUE
30 FORMAT(1H ,I2,4G12.5,1X,A10)
32 FORMAT(1H ,I2,3G12.5,2X,*INFINITE*,3X,A10)
WRITE(4,4)
4 FORMAT(/1H ,*SPACE*,2X,*N*,11X,*U*,10X,*UBAR*)
DO 2 I=1,NSPACE
ZDX=ZHE/RINDEX(I)
2 WRITE(4,3) I,RINDEX(I),ZDX*OMEGA(I),ZDX*OMEGAB(I)
3 FORMAT(1X,I2,6G12.5)
WRITE(4,9) FNUMBR,FOVDEG,SMAG
9 FORMAT(/1H ,*F/=*,G11.5,* FULL FIELD(DEG)=*,G11.5,
1 * MAG=*,G11.5)
WRITE(4,40)
40 FORMAT(/1H ,*OPTI*,2X,*CC*,10X,*AD*,10X,*AE*,10X,*AF*,10X,*AG*)
DO 45 I=1,NOPTI
J=I+1
NA=NASURF(J)
IF (NA.GT.0) WRITE(4,3) I,CCSURF(J),(ASURF(J,K),K=1,NA)
45 IF (NA.EQ.0) WRITE(4,3) I,CCSURF(J)
IF (DAXIS.EQ.0.) RETURN
ENTRY SHOWD
WRITE(4,6) 0.,DAXIS,HFIELD,DFIELD
6 FORMAT(/1H ,*RESIDUAL BLUR ON IMAGE SURFACE:*/
1 4X,*CENTROID*,6X,*DIAMETER*/
2 3X,2G12.5/3X,2G12.5)
IF (NUPPER.EQ.0) RETURN
ENTRY SHOWU
WRITE(4,51) DAPERT
51 FORMAT(/1H ,*UPPER BOUND (DAPERT=*,G12.5,*) :*/7X,*H*,11X,*Z*)
DO 61 J=1,NUPPER
61 WRITE(4,15) J,HUPPER(J),ZUPPER(J),KUPPER(J)
15 FORMAT(1H ,I2,2G12.5,1X,A10)
IF (NLOWER.EQ.0) RETURN
ENTRY SHOWL
ITYPE=6HOBSCUR
IF (KBEAM.GT.0) ITYPE=5HSHIFT
WRITE(4,71) ITYPE,OBSCUR
71 FORMAT(/1H ,*LOWER BOUND (*,A6,*=*,G12.5,*) :*/7X,*H*,11X,*Z*)
DO 81 J=1,NLOWER
81 WRITE(4,15) J,HLOWER(J),ZLOWER(J),KLOWER(J)
IF (NINTER.EQ.0) RETURN
ENTRY SHOWI
WRITE(4,35)
35 FORMAT(/1H ,*INTERSECTIONS:*/7X,*H*,11X,*Z*)
DO 20 I=1,NINTER
ITYPE1=ITYPE2=5HUPPER
IF (LINES(I,1).LT.0) ITYPE1=5HLOWER
IF (LINES(I,2).LT.0) ITYPE2=5HLOWER
LINE1=IABS(LINES(I,1))
LINE2=IABS(LINES(I,2))
20 WRITE(4,25) I,HINTER(I),ZINTER(I),ITYPE1,LINE1,ITYPE2,LINE2
25 FORMAT(1H ,I2,2G12.5,1X,A5,* LINE*,I3,* WITH *,A5,* LINE*,I3)
IF (NWARN.EQ.0) RETURN
WRITE(4,90)
90 FORMAT(/1H ,*WARNINGS:*)
DO 85 I=1,NWARN
ITYPE=5HUPPER
IF (IWARN(I,1).LT.0) ITYPE=5HLOWER

```

```

      LINE=IABS(IWARN(I,1))
85  WRITE(4,95) I, ITYPE, LINE, IWARN(I,2)
95  FORMAT(1H ,I2,2X,A5,* LINE*,I2,* INTERSECTS OPTICAL ELEMENT*,I2)
      RETURN
      END

```

```

SUBROUTINE PTRACE
COMMON /COPTIC/ OBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1  ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2  ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3  ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1  NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2  IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3  ,HPLT

```

```

C
C  TRACES THE MARGINAL AND CHIEF PARAXIAL RAYS
C

```

```

      YTRANS(Y,T,U)=Y+T*U
      UREFR(X1,X2,U,Y,P)=(X1*U-Y*P)/X2
      T=ZVSURF(2)-ZVSURF(1)
      Y=YSURF(2)
      U=Y/T
      UBAR=-YBSURF(1)/T
      YBAR=0.
      NM1=NSURF-1
      DO 7 I=2,NM1
      X1=RINDEX(I-1)
      X2=RINDEX(I)
      P=(X2-X1)*CVSURF(I)
      YSURF(I)=Y
      YBSURF(I)=YBAR
      U=UREFR(X1,X2,U,Y,P)
      UBAR=UREFR(X1,X2,UBAR,YBAR,P)
      IP1=I+1
      T=ZVSURF(IP1)-ZVSURF(I)
      Y=YTRANS(Y,T,U)
      IF (IP1.EQ.NSURF) GO TO 8
7     YBAR=YTRANS(YBAR,T,UBAR)
8     YSURF(NSURF)=0.
      DY=Y-YSURF(NSPACE)
      IF (DY.EQ.0.) STOP "AFOCAL"
      ZVSURF(NSURF)=ZVSURF(NSURF)-Y*(ZVSURF(NSURF)-ZVSURF(NSPACE))/DY
      T=ZVSURF(NSURF)-ZVSURF(NSPACE)
      YBSURF(NSURF)=YTRANS(YBAR,T,UBAR)
      IF (FNUMBR.EQ.0.) GO TO 10
      S=SIGN(1.,YSURF(NSPACE))
      YBSURF(NSURF)=FOVDEG*DAPERT*FNUMBR*S*CONV/2.
10    DAXIS=0.
      IF (IPRINT.LT.0) CALL SHOW
      RETURN
      END

```

```

SUBROUTINE YYBCAL
COMMON /COPTIC/ OBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1  ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2  ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3  ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1  NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2  IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC

```

```

3 ,HPLLOT
COMMON /CSTOP/ ZASTOP,ZOSTOP,KASTOP,KOSTOP,LASTOP,LOSTOP,
1 PDZSTP(2)
COMMON /CRAYS/ YRAYS(20,5),ZRAYS(20,5),YA(20),ZA(20),YO(20),ZO(20)
DIMENSION SEIDEL(5),DSASPH(5,5),ASPH(5)

C
C CALCULATES THE NECESSARY PARAXIAL INFORMATION AND SELECTS THE
C DESIRED CONIC CONSTANT SUCH THAT THIRD-ORDER
C ABERRATION IS ELIMINATED
C
FOVFAC=-BIG*FOVDEG*CONV/(2.*YBSURF(1))
DD2=DAPERT/2.
APTFAC=DD2/YSURF(2)
IF (KBEAM.GT.0) APTFAC=(DD2+OBSCUR)/YSURF(2)
DO 4 I=1,NSURF
YSURF(I)=APTFAC*YSURF(I)
4 YBSURF(I,1)=YBSURF(I,2)=FOVFAC*YBSURF(I,1)
ZHE=FOVFAC*APTFAC*ZHE
DO 18 I=1,NSPACE
J=I+1
T=RINDEX(I)*(YSURF(I)*YBSURF(J,2)-YSURF(J)*YBSURF(I,2))/ZHE
ZVSURF(J)=ZVSURF(I)+T
18 IF (ABS(ZVSURF(J)/T).LT.SMALL) ZVSURF(J)=0.
ZHE2=ZHE**2
IF (LASTOP.LT.0) ZASTOP=ZVSURF(-LASTOP)
IF (LOSTOP.LT.0) ZOSTOP=ZVSURF(-LOSTOP)
YBSTOP=YYBAR(YBSURF,ZVSURF,ZASTOP,KASTOP,NSURF)
IF (YBSTOP.EQ.0.) GO TO 3
YSTOP=YYBAR(YSURF,ZVSURF,ZASTOP,KASTOP,NSURF)
IF (YSTOP.EQ.0.) STOP "TRIED TO SHIFT APSTOP TO IMAGE"
S=YBSTOP/YSTOP
DO 5 I=2,NSPACE
5 YBSURF(I,2)=YBSURF(I,2)-S*YSURF(I)
3 DO 2 I=1,NSPACE
J=I+1
S=BIG
IF (YBSURF(J,2).NE.YBSURF(I,2)) S=(YSURF(J)-YSURF(I))/
1 (YBSURF(J,2)-YBSURF(I,2))
YE=YSURF(J)-S*YBSURF(J,2)
OMEGAB(I)=1./YE
2 OMEGA(I)=S*OMEGAB(I)
EFL=1./(ZHE*(OMEGA(1)*OMEGAB(NSPACE)-OMEGA(NSPACE)*OMEGAB(1)))
SMAG=YBSURF(NSURF)/YBSURF(1)
FNUMBR=ABS(EFL)/DAPERT
FOVDEG=ABS(ZHE*OMEGAB(1)/RINDEX(1))*2./CONV
DO 6 I=2,NSPACE
J=I-1
PHI=(OMEGA(J)*OMEGAB(I)-OMEGA(I)*OMEGAB(J))*ZHE
DN=RINDEX(I)-RINDEX(J)
CVSURF(I)=PHI/DN
RDSURF(I)=0.
6 IF (CVSURF(I).NE.0.) RDSURF(I)=1./CVSURF(I)
IF (ITRACE.GE.0) GO TO 15
DO 12 I=1,4
12 SEIDEL(I)=0.
PETZVL=0.
DO 7 I=1,NFIGR
IF (KFIGR(I).LT.0) ASURF(-KFIGR(I),1)=0.
7 IF (KFIGR(I).GT.0) CCSURF(KFIGR(I))=0.
DO 14 I=2,NSPACE
J=I-1

```

```

DN=RINDEX(I)-RINDEX(J)
A=ZHE*OMEGA(I)+RINDEX(I)*YSURF(I)*CVSURF(I)
B=ZHE*OMEGAB(I)+RINDEX(I)*YBSURF(I,2)*CVSURF(I)
DELUDN=ZHE*(OMEGA(I)/RINDEX(I)**2-OMEGA(J)/RINDEX(J)**2)
YBDY=YBSURF(I,2)/YSURF(I)
DSA=(CCSURF(I)*CVSURF(I)**3+8.*ASURF(I,1))*DN*YSURF(I)**4
YDUDN=YSURF(I)*DELUDN
SEIDEL(1)=SEIDEL(1)+DSA-YDUDN*A**2
DSA=DSA*YBDY
SEIDEL(2)=SEIDEL(2)+DSA-YDUDN*A*B
SIII=YDUDN*B**2
DSA=DSA*YBDY
SEIDEL(3)=SEIDEL(3)+DSA-SIII
SIV=CVSURF(I)*(1./RINDEX(I)-1./RINDEX(J))*ZHE2
PETZVL=PETZVL-SIV
DSA=DSA*YBDY
14 SEIDEL(4)=SEIDEL(4)+DSA-B*(SIII+SIV)/A
IF (NFIGR.EQ.0) GO TO 10
NM1=NFIGR-1
IF (NM1.EQ.0) GO TO 17
DO 16 I=1,NM1
KI=IABS(KFIGR(I))
YBDY=ABS(YBSURF(KI,2)/YSURF(KI))
IP1=I+1
DO 16 J=IP1,NFIGR
KJ=IABS(KFIGR(J))
IF (YBDY.LT.ABS(YBSURF(KJ,2)/YSURF(KJ))) GO TO 16
KT=KFIGR(I)
KFIGR(I)=KFIGR(J)
KFIGR(J)=KT
16 CONTINUE
17 DO 1 J=1,NFIGR
K=IABS(KFIGR(J))
YBDY=YBSURF(K,2)/YSURF(K)
DNY4=(RINDEX(K)-RINDEX(K-1))*YSURF(K)**4
IF (KFIGR(J).LT.0) DSA=8.*DNY4
IF (KFIGR(J).GT.0) DSA=DNY4*CVSURF(K)**3
E=1
DO 1 I=1,4
DSASPH(I,J)=E*DSA
1 E=E*YBDY
N=NFIGR
8 D=DETERM(DSASPH,ASPH,SEIDEL,N)
IF (D.NE.0.) GO TO 9
N=N-1
IF (N.EQ.0) GO TO 10
GO TO 8
9 DO 11 I=1,N
K=KFIGR(I)
IF (K.LT.0) ASURF(-K,1)=-ASPH(I)
11 IF (K.GT.0) CCSURF(K)=-ASPH(I)
DO 13 I=1,4
DO 13 J=1,N
13 SEIDEL(I)=SEIDEL(I)-DSASPH(I,J)*ASPH(J)
10 IF (IMBEST.EQ.0) GO TO 15
CVSURF(NSURF)=-RINDEX(NSPACE)*(2.*SEIDEL(3)+PETZVL)/ZHE2
RDSURF(NSURF)=0.
IF (CVSURF(NSURF).NE.0.) RDSURF(NSURF)=1./CVSURF(NSURF)
CCSURF(NSURF)=-1.
NASURF(NSURF)=0
ZVSURF(NSURF)=ZVSURF(NSURF)-.75*SEIDEL(1)*RINDEX(NSPACE)/(2.*ZHE2

```

```

      1 OMEGA(NSPACE)**2)
15  IF (KSFIGN.EQ.0) GO TO 100
    DO 30 I=1,NSFIGN
30  ASURF(KSFIGN,I+1)=0.
    DN=RINDEX(KSFIGN)-RINDEX(KSFIGN-1)
    DO 40 I=1,NSFIGN
    DO 35 J=1,NSURF
    YA(J)=YSURF(J)
35  ZA(J)=ZVSURF(J)
    CALL RTRACE(YA,ZA,1,NSURF)
    N=2*I+4
    WSA=-OMEGA(NSPACE)*YA(NSURF)/N
40  ASURF(KSFIGN,I+1)=WSA/(DN*YSURF(KSFIGN)**N)
100 DAXIS=0.
    IF (IPRINT.LT.0) CALL SHOW
    RETURN
    END

```

```

      FUNCTION YYBAR(Y,Z,ZS,KS,N)
C
C  FINDS Y OR YBAR GIVEN Z COORDINATE AND SPACE
C
    DIMENSION Y(N),Z(N)
    KSP1=KS+1
    YYBAR=Y(KSP1)+(ZS-Z(KSP1))*(Y(KS)-Y(KSP1))/(Z(KS)-Z(KSP1))
    RETURN
    END

```

```

      FUNCTION DETERM(A,X,B,N)
C..GAUSS-JORDAN SOLUTION OF A SYSTEM OF LINEAR EQUATIONS
    DIMENSION A(5,5),X(5),B(5),C(5,6)
    NP1=N+1
    DO 10 I=1,N
    C(I,NP1)=B(I)
    DO 10 J=1,N
10  C(I,J)=A(I,J)
    DETERM=1.
    DO 40 K=1,N
    DETERM=DETERM*C(K,K)
    IF (DETERM.EQ.0.) RETURN
    NPK=NP1+K
    DO 20 L=K,NP1
    J=NPK-L
20  C(K,J)=C(K,J)/C(K,K)
    DO 40 I=1,N
    IF (I.EQ.K) GO TO 40
    DO 30 L=K,NP1
    J=NPK-L
30  C(I,J)=C(I,J)-C(I,K)*C(K,J)
40  CONTINUE
    DO 50 I=1,N
50  X(I)=C(I,NP1)
    RETURN
    END

```

```

      SUBROUTINE BEAM
      COMMON /CBEAM/ ZUPPER(20),HUPPER(20),ZLOWER(20),HLOWER(20),
1  ZINTER(20),HINTER(20),ZPTS(20),HPTS(20),NUPPER,NLOWER,NINTER,
2  LINES(20,2),ZPORT(2),KUPPER(20),KLOWER(20),MOPTIC(20,3)
3  ,NWARN,IWARN(20,2),NAMINT(2,20)
      COMMON /CSTOP/ ZASTOP,ZOSTOP,KASTOP,KOSTOP,LASTOP,LOSTOP,

```



```

1 PDZSTP(2)
COMMON /COPTIC/ OBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1 ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2 ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3 ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1 NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2 IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3 ,HPLOT
COMMON /CRAYS/ YRAYS(20,5),ZRAYS(20,5),YA(20),ZA(20),YO(20),ZO(20)

```

C DETERMINES THE UPPER AND LOWER BOUNDS ON THE BEAM

```

C
C
1 DIMENSION YBO(20)
YBOSTP=YYBAR(YBSURF,ZVSURF,ZOSTOP,KOSTOP,NSURF)
DO 1 I=1,NSURF
1 YBO(I)=YBSURF(I)
IF (YBOSTP.EQ.0.) GO TO 3
YOSTP=YYBAR(YSURF,ZVSURF,ZOSTOP,KOSTOP,NSURF)
IF (YOSTP.EQ.0.) STOP "TRIED TO SHIFT OBSTOP TO IMAGE"
SK=YBOSTP/YOSTP
DO 2 I=2,NSPACE
2 YBO(I)=YBO(I)-SK*YSURF(I)
3 OBS=OBSCUR
DD2=DAPERT/2.
IF (KBEAM.GT.0) OBS=(OBSCUR-DD2)/YSURF(2)
DO 10 J=1,NSURF
YRAYS(J,1)=YRAYS(J,2)=YA(J)=YSURF(J)
YRAYS(J,3)=YRAYS(J,4)=YO(J)=OBS*YSURF(J)
YBO(J)=OBFOV*YBO(J)
YRAYS(J,5)=0.
ZA(J)=ZO(J)=ZVSURF(J)
IF (IVIGN.EQ.0) GO TO 5
YRAYS(J,1)=YRAYS(J,1)-YBSURF(J,2)
YRAYS(J,2)=YRAYS(J,2)+YBSURF(J,2)
YRAYS(J,3)=YRAYS(J,3)+YBO(J)
YRAYS(J,4)=YRAYS(J,4)-YBO(J)
5 DO 10 I=1,5
10 ZRAYS(J,I)=ZVSURF(J)
IF (ITRACE.GE.0) GO TO 40
CALL RTRACE(YA,ZA,1,NSURF)
CALL RTRACE(YO,ZO,1,NSURF)
KAP1=KASTOP+1
KOP1=KOSTOP+1
YASTOP=YA(KAP1)
ZAS=ZA(KAP1)
IF (ABS(ZASTOP-ZVSURF(KAP1)).LT.DELMAX) GO TO 20
ZAS=ZASTOP
YASTOP=YYBAR(YA,ZA,ZAS,KASTOP,NSURF)
20 YOSTOP=YO(KOP1)
ZOS=ZO(KOP1)
IF (ABS(ZOSTOP-ZVSURF(KOP1)).LT.DELMAX) GO TO 30
ZOS=ZOSTOP
YOSTOP=YYBAR(YO,ZO,ZOS,KOSTOP,NSURF)
30 CALL RAYAIM(YRAYS(1,1),ZRAYS(1,1),YASTOP,ZAS,KASTOP,1,NSURF)
CALL RAYAIM(YRAYS(1,2),ZRAYS(1,2),YASTOP,ZAS,KASTOP,1,NSURF)
CALL RAYAIM(YRAYS(1,3),ZRAYS(1,3),YOSTOP,ZOS,KOSTOP,1,NSURF)
CALL RAYAIM(YRAYS(1,4),ZRAYS(1,4),YOSTOP,ZOS,KOSTOP,1,NSURF)
40 AY1=ABS(YRAYS(NSURF,1))
AY2=ABS(YRAYS(NSURF,2))
AY3=ABS(YRAYS(NSURF,3))

```

```

    AY4=ABS(YRAYS(NSURF,4))
    DAXIS=2.*AMAX1(ABS(YA(NSURF)),ABS(YO(NSURF)))
    AYMAX=AMAX1(AY1,AY2,AY3,AY4)
    AYMIN=AMIN1(AY1,AY2,AY3,AY4)
    HFIELD=(AYMAX+AYMIN)/2.
    DFIELD=AYMAX-AYMIN
    DAIRY=2.44*WAVEL*FNUMBR
    IF (DAIRY.EQ.0.) GO TO 55
    DAXIS=DAXIS/DAIRY
    DFIELD=DFIELD/DAIRY
55  NUPPER=0
    IF (IPRINT.LT.0) CALL SHOWD
    IF (KBEAM.GT.0) GO TO 50
    CALL BOUND(1,2,-1,HUPPER,ZUPPER,KUPPER,NUPPER)
    NLOWER=0
    IF (IPRINT.LT.0) CALL SHOWU
    CALL BOUND(-5,5,-3,HLOWER,ZLOWER,KLOWER,NLOWER)
    NLOWER=NLOWER-2
    DO 60 I=1,NLOWER
    J=I+1
    HLOWER(I)=HLOWER(J)
    ZLOWER(I)=ZLOWER(J)
60  KLOWER(I)=KLOWER(J)
45  NM1=NLOWER-1
    IF ((HLOWER(NLOWER).NE.0.).OR.(HLOWER(NM1).NE.0.)) GO TO 70
    NLOWER=NLOWER-1
    GO TO 45
50  CALL BOUND(1,4,1,HUPPER,ZUPPER,KUPPER,NUPPER)
    NLOWER=0
    IF (IPRINT.LT.0) CALL SHOWU
    CALL BOUND(3,4,1,HLOWER,ZLOWER,KLOWER,NLOWER)
70  NINTER=0
    IF (IPRINT.LT.0) CALL SHOWL
    IF (TOLER.EQ.0.) GO TO 100
    TUPPER=1.+TOLER
    DO 80 I=1,NUPPER
    IF ((KUPPER(I).AND.MASKS(2)).EQ.0) GO TO 80
    HUPPER(I)=TUPPER*HUPPER(I)
80  CONTINUE
    TLOWER=1.-TOLER
    DO 90 I=1,NLOWER
    IF ((KLOWER(I).AND.MASKS(2)).EQ.0) GO TO 90
    HLOWER(I)=TLOWER*HLOWER(I)
90  CONTINUE
    IF (IPRINT.LT.0) CALL SHOWU
100 IF (ZPORT(1).EQ.0.) RETURN
    HUPPER(1)=HUPPER(2)+(HUPPER(2)-HUPPER(1))*(ZPORT(1)-ZUPPER(2))/
    1 (ZUPPER(2)-ZUPPER(1))
    ZUPPER(1)=ZPORT(1)
    KUPPER(1)=5RENTA
    IF (ZPORT(2).LT.ZLOWER(1)) GO TO 105
    HLOWER(1)=HLOWER(2)+(HLOWER(2)-HLOWER(1))*(ZPORT(2)-ZLOWER(2))/
    1 (ZLOWER(2)-ZLOWER(1))
    ZLOWER(1)=ZPORT(2)
    KLOWER(1)=5RENTA
105 ZP=AMIN1(ZPORT(1),ZPORT(2))
    YA(1)=YA(2)+(YA(2)-YA(1))*(ZP-ZA(2))/
    1 (ZA(2)-ZA(1))
    YO(1)=YO(2)+(YO(2)-YO(1))*(ZP-ZO(2))/
    1 (ZO(2)-ZO(1))
    ZA(1)=ZO(1)=ZP

```

```

DO 110 J=1,4
YRAYS(1,J)=YRAYS(2,J)+(YRAYS(2,J)-YRAYS(1,J))*(ZP-ZRAYS(2,J))/
1 (ZRAYS(2,J)-ZRAYS(1,J))
110 ZRAYS(1,J)=ZP
IF (IPRINT.LT.0) CALL SHOWU
RETURN
END

```

```

SUBROUTINE RTRACE(Y,Z,ISTR,ISTOP)

```

```

C
C TRACES A REAL MERIDONAL RAY THROUGH CONIC SURFACES
C

```

```

COMMON /COPTIC/ OBSCUR,FQVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1 ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2 ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3 ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
DIMENSION Y(20),Z(20)
IDIR=ISIGN(1,ISTOP-ISTR)
I=ISTR+IDIR
DY=Y(I)-Y(ISTR)
DZ=Z(I)-Z(ISTR)
IX=MAX(0,IDIR)
IXP=MAX(0,-IDIR)
IF (DZ.EQ.0.) STOP "FIRST TWO SURFACES COINCIDE"
S=DY/DZ
D=SQRT(DZ**2+DY**2)
AN=ABS(DZ/D)
AM=AN*S
YO=Y(I)+AM*(ZVSURF(I)-Z(I))/AN
20 C=CVSURF(I)
CC=CCSURF(I)
Y(I)=YO
Z(I)=0
AB=0.
AC=1.
NA=NASURF(I)
IF ((C.EQ.0.).AND.(NA.EQ.0)) GO TO 70
F=C*YO**2
G=AN-C*AM*YO
HIT=G**2-C**F*(1+CC*AN**2)
IF (HIT.LT.0.) STOP "RAY MISSED CONIC"
D=F/(G+SQRT(HIT))
Y(I)=YO+AM*D
Z(I)=AN*D
E=CC+1
D=SQRT(1-E*(C*Y(I))**2)
DZ=DZP=DZPP=0.
ZCP=C*Y(I)/D
IF (NA.EQ.0) GO TO 60
DO 40 J=1,NA
K=2*(J+1)
DZ=DZ+ASURF(I,J)*Y(I)**K
DZP=DZP+K*ASURF(I,J)*Y(I)**(K-1)
40 DZPP=DZPP+K*(K-1)*ASURF(I,J)*Y(I)**(K-2)
ZP=DZP
ZPP=DZPP
IF (C.EQ.0) GO TO 50
ZCPP=C*D**(-1.5)
ZP=DZP+ZCP
ZPP=DZPP+ZCPP
50 F=DZ

```

```

G=AM*ZP-AN
A=AM*AM*ZPP/2.
HIT=G**2-4.*A*F
IF (HIT.LT.0.) STOP "RAY MISSED ASPHERE"
D=2.*F/(SQRT(HIT)-G)
Y(I)=Y(I)+AM*D
Z(I)=Z(I)+AN*D
DZP=0.
DO 55 J=1,NA
K=2*(J+1)
55 DZP=DZP+K*ASURF(I,J)*Y(I)**(K-1)
60 AB=-(ZCP+DZP)
D=SQRT(1.+AB*AB)
AB=AB/D
AC=1./D
70 Z(I)=ZVSURF(I)+Z(I)
IF (I.EQ.ISTOP) RETURN
COSI=ABS(AM*AB+AN*AC)
X=RINDEX(I-IX)
XP=RINDEX(I-IXP)
CRIT=1+(COSI**2-1)*(X/XP)**2
IF (CRIT.LT.0.) STOP "TOTAL INTERNAL REFLECTION"
COSIP=SQRT(CRIT)
AK=XP*COSIP-X*COSI
AM=(X*AM+AK*AB)/XP
AN=(X*AN+AK*AC)/XP
IP1=I+IDIR
Z(IP1)=ZVSURF(IP1)
YO=Y(IP1)=Y(I)+AM*(Z(IP1)-Z(I))/AN
I=IP1
GO TO 20
END

```

SUBROUTINE RAYAIM(Y,Z,YP,ZP,KP,ISTR,ISTOP)

```

C
C DETERMINES THE RAY THAT INTERSECTS A GIVEN POINT WITHIN THE SYSTEM
C
COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1 NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2 IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3 ,HPLOT
DIMENSION Y(20),Z(20)
KP1=KP+1
KSTOP=KP1
IF (ISTOP.LT.ISTR) KSTOP=KP
IF (KP.GT.1) GO TO 5
Y(2)=YP
Z(2)=ZP
5 CALL RTRACE(Y,Z,ISTR,KSTOP)
IF (KP.EQ.1) GO TO 30
DO 20 I=1,10
DYDZ=(Y(KP1)-Y(KP))/(Z(KP1)-Z(KP))
YI=DYDZ*(ZP-Z(KP))+Y(KP)
PDY=DY
DY=YP-YI
IF (DYDZ.NE.0.) DZ=DY/DYDZ
IF ((I.GT.2).AND.(ABS(DY).LT.DELMAX).AND.(ABS(DZ).LT.DELMAX))
1 GO TO 30
IF (I.GT.1) GO TO 10
PY=Y(2)
Y(2)=YI*Y(2)/YP

```

```

      GO TO 20
10    S=(Y(2)-PY)/(DY-PDY)
      PY=Y(2)
      Y(2)=Y(2)-DY*S
20    CALL RTRACE(Y,Z,ISTRT,KSTOP)
30    CALL RTRACE(Y,Z,ISTRT,ISTOP)
      RETURN
      END

```

```

      SUBROUTINE BOUND(JSTRT,JSTOP,JRST,H,Z,K,N)

```

```

C
C DETERMINES EITHER UPPER OR LOWER BOUND OF BEAM
C

```

```

      COMMON /COPTIC/ OBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1    ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2    ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3    ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
      COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1    NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2    IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3    ,HPLLOT
      COMMON /CRAYS/ YRAYS(20,5),ZRAYS(20,5),YA(20),ZA(20),YD(20),ZD(20)
      DIMENSION H(20),Z(20),K(20),KTYPE(3)
      DATA KTYPE/5RSTOP,5RIMAGE,5RAXIS /
      DATA MASKS/77777777770000000000B,00000000007777777777B/
      I=N-0
      S=SP=ISIGN(1,JSTRT)
      SRST=ISIGN(1,JRST)
      JRST=IABS(JRST)
      NIMAX=JSTOP-JRST+1
      J=IABS(JSTRT)
      JP1=J+1
      IF (JP1.LE.JSTOP) GO TO 1
      SP=SP*SRST
      JP1=JRST
1    I=I+1
      NM1=N
      N=N+1
      H(N)=S*YRAYS(I,J)
      Z(N)=ZRAYS(I,J)
      K(N)=KSURF(I)
      IF (NM1.LT.1) GO TO 2
      IF (((K(N).AND.MASKS(2)).NE.0).AND.(K(N).EQ.K(NM1))) GO TO 6
      IF (ABS(Z(N)-Z(NM1)).GT.DELMAX) GO TO 2
      IF ((K(NM1).AND.K(N)).EQ.0) K(NM1)=K(NM1).OR.K(N)
6    N=NM1
2    IF (I.EQ.NSURF) RETURN
      IP1=I+1
      NI=0
4    IF (NI.EQ.NIMAX) GO TO 1
      Y1=S*YRAYS(I,J)
      Y2=S*YRAYS(IP1,J)
      Z1=ZRAYS(I,J)
      Z2=ZRAYS(IP1,J)
      YP1=SP*YRAYS(I,JP1)
      YP2=SP*YRAYS(IP1,JP1)
      ZP1=ZRAYS(I,JP1)
      ZP2=ZRAYS(IP1,JP1)
      IF ((Y1-YP1)*(Y2-YP2).GT.SMALL) GO TO 1
      ZI=UINTER(Z1,Y1,Z2,Y2,ZP1,YP1,ZP2,YP2)
      DZ=ZI-Z(N)

```

```

IF (DZ*RINDEX(I).LT.-SMALL) GO TO 1
NI=NI+1
L=2-MOD(J,2)
IF ((J.EQ.5).OR.(JP1.EQ.5)) L=3
J=JP1
S=SP
JP1=J+1
IF (JP1.LE.JSTOP) GO TO 3
SP=SP*SRST
JP1=JRST
3 IF (ABS(DZ).GT.DELMAX) GO TO 5
IF ((K(N).AND.KTYPE(L)).EQ.0) K(N)=K(N).OR.KTYPE(L)
GO TO 4
5 N=N+1
H(N)=UINTER(Y1,Z1,Y2,Z2,YP1,ZP1,YP2,ZP2)
Z(N)=ZI
K(N)=KTYPE(L)
IF (ABS(ZI-ZRAYS(IP1,J)).GT.DELMAX) GO TO 4
GO TO 1
END

```

```

FUNCTION UINTER(U1,V1,U2,V2,UP1,VP1,UP2,VP2)

```

```

C
C FINDS EITHER COORDINATE OF THE INTERSECTION OF TWO INFINITE LINES
C

```

```

UINTER=U1
IF (ABS(U2-U1).LT.1.E-10) RETURN
UINTER=UP1
IF (ABS(UP2-UP1).LT.1.E-10) RETURN
S=(V2-V1)/(U2-U1)
SP=(VP2-VP1)/(UP2-UP1)
IF (S.EQ.SP) STOP "LINES PARALLEL"
UINTER=(S*U1-SP*UP1-V1+VP1)/(S-SP)
RETURN
END

```

```

SUBROUTINE INTPTS

```

```

COMMON /CBEAM/ ZUPPER(20),HUPPER(20),ZLOWER(20),HLOWER(20),
1 ZINTER(20),HINTER(20),ZPTS(20),HPTS(20),NUPPER,NLOWER,NINTER,
2 LINES(20,2),ZPORT(2),KUPPER(20),KLOWER(20),MOPTIC(20,3)
3 ,NWARN,IWARN(20,2),NAMINT(2,20)

```

```

C
C DETERMINES THE INTERSECTION POINTS THAT OCCUR WHEN THE BEAM IS
C FOLDED BACK ON ITSELF BY REFLECTING ELEMENTS
C

```

```

COMMON /COPTIC/ OBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1 ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2 ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3 ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1 NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2 IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3 ,HPLT
COMMON /CVARY/ KVARIB(3),LVARIB(3),SVARY(3),YVARY(3),FDIFR(5,5),
1 YBVARY(3),KQUANT(3,10),XDIFR(5,5),PQUANT(3),NVARY,NSOLV,ITER,DEL
2 ,KSPACE(3,10),INDVAR
DIMENSION ZHPTS(160)
EQUIVALENCE (ZUPPER(1),ZHPTS(1))
NWARN=0
KU=KL=1
DO 9 I=1,NOPTI

```

```

1  KU=KU+1
   IF ((KUPPER(KU).AND.MASKS(1)).EQ.0) GO TO 1
2  KL=KL+1
   IF ((KLOWER(KL).AND.MASKS(1)).EQ.0) GO TO 2
   MOPTIC(I,1)=KU
   MOPTIC(I,2)=KL
   MOPTIC(I,3)=KUPPER(KU).AND.MASKS(1)
   NM1=NUPPER-1
   DO 3 L=1,NM1
     LP1=L+1
     IF ((L.EQ.KU).OR.(LP1.EQ.KU)) GO TO 3
     CALL SEGINT(HUPPER(L),HUPPER(LP1),ZUPPER(L),ZUPPER(LP1),
1  HUPPER(KU),HLOWER(KL),ZUPPER(KU),ZLOWER(KL),HI,ZI,INT,0.)
     IF (INT.EQ.0) GO TO 3
     NWARN=NWARN+1
     IWARN(NWARN,1)=L
     IWARN(NWARN,2)=I
3  CONTINUE
   NM1=NLOWER-1
   DO 5 L=1,NM1
     LP1=L+1
     IF ((L.EQ.KL).OR.(LP1.EQ.KL)) GO TO 5
     CALL SEGINT(HLOWER(L),HLOWER(LP1),ZLOWER(L),ZLOWER(LP1),
1  HUPPER(KU),HLOWER(KL),ZUPPER(KU),ZLOWER(KL),HI,ZI,INT,0.)
     IF (INT.EQ.0) GO TO 5
     NWARN=NWARN+1
     IWARN(NWARN,1)=-L
     IWARN(NWARN,2)=I
5  CONTINUE
9  CONTINUE
   IF (NINT.EQ.0) GO TO 7
   DO 8 I=1,NINT
     KZ1=LINES(I,1)
     IF (KZ1.LT.0) KZ1=2*KDIM-KZ1
     KH1=KZ1+KDIM
     KZ2=KZ1+1
     KH2=KH1+1
     Z1=ZHPTS(KZ1)
     H1=ZHPTS(KH1)
     Z2=ZHPTS(KZ2)
     H2=ZHPTS(KH2)
     KZ1P=LINES(I,2)
     IF (KZ1P.LT.0) KZ1P=2*KDIM-KZ1P
     KH1P=KZ1P+KDIM
     KZ2P=KZ1P+1
     KH2P=KH1P+1
     Z1P=ZHPTS(KZ1P)
     H1P=ZHPTS(KH1P)
     Z2P=ZHPTS(KZ2P)
     H2P=ZHPTS(KH2P)
8  ZINTER(I)=UINTER(Z1,H1,Z2,H2,Z1P,H1P,Z2P,H2P)
7  HINTER(I)=UINTER(H1,Z1,H2,Z2,H1P,Z1P,H2P,Z2P)
   NINTER=NINT
   NM2=NUPPER-2
   DO 10 I=2,NM2
     IM1=I-1
     NMI=NUPPER-I
     DO 10 K=2,NMI
       J=I+K
       JM1=J-1
       CALL SEGINT(HUPPER(IM1),HUPPER(I),ZUPPER(IM1),ZUPPER(I),

```

```

1 HUPPER(JM1),HUPPER(J),ZUPPER(JM1),ZUPPER(J),HI,ZI,INT,0.)
  IF (INT.EQ.0) GO TO 10
  NINTER=NINTER+1
  HINTER(NINTER)=HI
  ZINTER(NINTER)=ZI
  LINES(NINTER,1)=IM1
  LINES(NINTER,2)=JM1
10 CONTINUE
  NM2=NLOWER-2
  DO 15 I=2,NM2
  IM1=I-1
  NMI=NLOWER-I
  DO 15 K=2,NMI
  J=I+K
  JM1=J-1
  CALL SEGINT(HLOWER(IM1),HLOWER(I),ZLOWER(IM1),ZLOWER(I),
1 HLOWER(JM1),HLOWER(J),ZLOWER(JM1),ZLOWER(J),HI,ZI,INT,0.)
  IF (INT.EQ.0) GO TO 15
  NINTER=NINTER+1
  HINTER(NINTER)=HI
  ZINTER(NINTER)=ZI
  LINES(NINTER,1)=-IM1
  LINES(NINTER,2)=-JM1
  NAMINT(1,NINTER)=NAMINT(2,NINTER)=10H
15 CONTINUE
  DO 30 I=2,NUPPER
  IM1=I-1
  DO 30 J=2,NLOWER
  JM1=J-1
  CALL SEGINT(HUPPER(IM1),HUPPER(I),ZUPPER(IM1),ZUPPER(I),
1 HLOWER(JM1),HLOWER(J),ZLOWER(JM1),ZLOWER(J),HI,ZI,INT,0.)
  IF (INT.EQ.0) GO TO 30
  NINTER=NINTER+1
  HINTER(NINTER)=HI
  ZINTER(NINTER)=ZI
  LINES(NINTER,1)=IM1
  LINES(NINTER,2)=-JM1
30 CONTINUE
  IF (IPRINT.LT.0) CALL SHOWI
  RETURN
  END

```

SUBROUTINE SEGINT(X1,X2,Y1,Y2,XP1,XP2,YP1,YP2,XI,YI,I,D)

C
C DETERMINES THE INTERSECTION (IF IT EXISTS) OF TWO LINE SEGMENTS
C

```

  I=0
  XMAX=AMAX1(X1,X2)
  XMINP=AMIN1(XP1,XP2)
  IF (XMINP-XMAX.GT.D) RETURN
  XMIN=AMIN1(X1,X2)
  XMAXP=AMAX1(XP1,XP2)
  IF (XMIN-XMAXP.GT.D) RETURN
  YMAX=AMAX1(Y1,Y2)
  YMINP=AMIN1(YP1,YP2)
  IF (YMINP-YMAX.GT.D) RETURN
  YMIN=AMIN1(Y1,Y2)
  YMAXP=AMAX1(YP1,YP2)
  IF (YMIN-YMAXP.GT.D) RETURN
  XI=UINTER(X1,Y1,X2,Y2,XP1,YP1,XP2,YP2)
  IF (((XMIN-XI.GT.D).OR.(XI-XMAX.GT.D)).OR.((XMINP-XI.GT.D).OR.

```



```

1 (XI-XMAXP.GT.D))) RETURN
YI=UINTER(Y1,X1,Y2,X2,YP1,XP1,YP2,XP2)
IF (((YMIN-YI.GT.D).OR.(YI-YMAX.GT.D)).OR.((YMINP-YI.GT.D).OR.
1 (YI-YMAXP.GT.D))) RETURN
I=1
RETURN
END

```

```

SUBROUTINE CHANG1
COMMON /CSTOP/ ZASTOP,ZOSTOP,KASTOP,KOSTOP,LASTOP,LOSTOP,
1 PDZSTP(2)
COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1 NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2 IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3 ,HPLLOT
COMMON /CBEAM/ ZUPPER(20),HUPPER(20),ZLOWER(20),HLOWER(20),
1 ZINTER(20),HINTER(20),ZPTS(20),HPTS(20),NUPPER,NLOWER,NINTER,
2 LINES(20,2),ZPORT(2),KUPPER(20),KLOWER(20),MOPTIC(20,3)
3 ,NWARN,IWARN(20,2),NAMINT(2,20)
COMMON /CVARY/ KVARIB(3),LVARIB(3),SVARY(3),YVARY(3),FDIFR(5,5),
1 YBVARY(3),KQUANT(3,10),XDIFR(5,5),PQUANT(3),NVARY,NSOLV,ITER,DEL
2 ,KSPACE(3,10),INDVAR
COMMON /COPTIC/ OBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1 ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2 ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3 ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)

```

C COMPUTES THE INITIAL CHANGES IN THE VARIABLES

```

C
C
C
1 DIMENSION VARI(43),ZHPTS(160),LSTOP(2),ZSTOP(2)
1 ,H(3),Z(3)
EQUIVALENCE (VARI(1),OBSCUR),(ZUPPER(1),ZHPTS(1))
1 ,(ZSTOP(1),ZASTOP),(LSTOP(1),LASTOP)
DO 3 I=1,NVARY
IF (KVARIB(I).LT.4) GO TO 3
IF (LVARIB(I).EQ.0) GO TO 3
SVARY(I)=(VARI(KVARIB(I))-YVARY(I))/(VARI(LVARIB(I))-YBVARY(I))
IF (ABS(SVARY(I)).GT.1.) GO TO 3
LVARIB(I)=KVARIB(I)
KVARIB(I)=LVARIB(I)+KDIM
SVARY(I)=1./SVARY(I)
T=YVARY(I)
YVARY(I)=YBVARY(I)
YBVARY(I)=T
3 CONTINUE
NX=0
DO 10 I=1,2
IF (LSTOP(I)) 6,10,7
6 ZSTOP(I)=ZVSURF(-LSTOP(I))
GO TO 10
7 NX=NX+1
PDZSTP(I)=ZSTOP(I)-ZHPTS(LSTOP(I))
XDIFR(1,NX)=-PDZSTP(I)
ZSTOP(I)=ZSTOP(I)+XDIFR(1,NX)
10 CONTINUE
DO 100 I=1,NVARY
IF (KQUANT(3,I).NE.0) GO TO 30
K1=KQUANT(1,I)
K2=KQUANT(2,I)
I1=ISIGN(1,K1)
I2=ISIGN(1,K2)

```

```

K1=IABS(K1)
K2=IABS(K2)
PQUANT(I)=(I1*ZHPTS(K1))-(I2*ZHPTS(K2))
GO TO 50
30 DO 35 J=1,3
   KZ=KQUANT(J,I)
   IS=ISIGN(1,KZ)
   KZ=IABS(KZ)
   KH=KZ+KDIM
   H(J)=IS*ZHPTS(KH)
35 Z(J)=ZHPTS(KZ)
   IF (KSPACE(1,I).EQ.0) GO TO 45
   DO 40 J=2,3
40 CALL IMAGE(H(J),Z(J),KSPACE(J,I),KSPACE(1,I))
45 PQUANT(I)=(Z(1)-Z(2))-(H(1)-H(2))*(Z(2)-Z(3))/(H(2)-H(3))
50 NX=NX+1
   XDIFR(1,NX)=DELONE*VARI(KVARIB(I))
   PVARI=VARI(KVARIB(I))
   VARI(KVARIB(I))=VARI(KVARIB(I))+XDIFR(1,NX)
   IF (KVARIB(I).LT.4) GO TO 100
   IF (LVARIB(I).EQ.0) GO TO 100
   VARI(LVARIB(I))=(VARI(KVARIB(I))-YVARY(I))/SVARY(I)+YBVARY(I)
100 CONTINUE
   CALL DISPLA("    DELTA",0)
   RETURN
END

```

```

SUBROUTINE IMAGE(H,Z,K,KP)
C..FIND IMAGE OF POINT USING Y-YBAR DIAGRAM METHOD
COMMON /COPTIC/ OBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1 ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2 ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3 ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
   IF (K.EQ.KP) RETURN
   Y=YYBAR(YSURF,ZVSURF,Z,K,NSURF)
   YBAR=YYBAR(YBSURF,ZVSURF,Z,K,NSURF)
   KP1=KP+1
   YP=UINTER(0.,0.,Y,YBAR,YSURF(KP),YBSURF(KP),YSURF(KP1),YBSURF(KP1
1 )
   YBARP=UINTER(0.,0.,YBAR,Y,YBSURF(KP),YSURF(KP),YBSURF(KP1),YSURF(
1 KP1))
   IF (ABS(Y).LT.ABS(YBAR)) GO TO 10
   H=H*YP/Y
   GO TO 20
10 H=H*YBARP/YBAR
20 Z=ZVSURF(KP1)+RINDEX(KP)*(YSURF(KP1)*YBARP-
1 YBSURF(KP1)*YP)/ZHE
   RETURN
END

```

```

SUBROUTINE CHANG2, RETURNS(DONE)
COMMON /CSTOP/ ZASTOP,ZOSTOP,KASTOP,KOSTOP,LASTOP,LOSTOP,
1 PDZSTP(2)
COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1 NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2 IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3 ,HPLDT
COMMON /CBEAM/ ZUPPER(20),HUPPER(20),ZLOWER(20),HLOWER(20),
1 ZINTER(20),HINTER(20),ZPTS(20),HPTS(20),NUPPER,NLOWER,NINTER,
2 LINES(20,2),ZPORT(2),KUPPER(20),KLOWER(20),MOPTIC(20,3)
3 ,NWARN,IWARN(20,2),NAMINT(2,20)

```

```

COMMON /CVARY/ KVARIB(3),LVARIB(3),SVARY(3),YVARY(3),FDIFR(5,5),
1 YBVAR(3),KQUANT(3,10),XDIFR(5,5),PQUANT(3),NVARY,NSOLV,ITER,DEL
2 ,KSPACE(3,10),INDVAR
COMMON /COPTIC/ OBSCUR,F0VDEG,DAPERT,YSURF(20),YBSURF(20,2)
1 ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2 ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3 ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)

```

```

C
C COMPUTES THE APPROXIMATE CHANGES IN THE VARIABLES
C NEEDED TO APPROACH THE DESIRED CONSTRAINTS BY THE
C NEWTON-RAPHSON METHOD
C

```

```

        DIMENSION VARI(43),ZHPTS(160),LSTOP(2),ZSTOP(2),KSTOP(2)
1      ,H(3),Z(3),FUNCT(5),PDERIV(5,5),TEMP(5)
      EQUIVALENCE (VARI(1),OBSCUR),(ZUPPER(1),ZHPTS(1))
1      , (ZSTOP(1),ZASTOP),(LSTOP(1),LASTOP)
      NS=0
      CALL STACK(FDIFR,NVARY+2)
      DO 10 I=1,2
      IF (LSTOP(I)) 6,10,7
6      ZSTOP(I)=ZVSURF(-LSTOP(I))
      GO TO 10
7      NS=NS+1
      FUNCT(NS)=ZSTOP(I)-ZHPTS(LSTOP(I))
      KSTOP(NS)=I
      FDIFR(1,NS)=FUNCT(NS)-PDZSTP(I)
      PDZSTP(I)=FUNCT(NS)
10     CONTINUE
      DO 55 I=1,NVARY
      NX=NS+I
      IF (KQUANT(3,I).NE.0) GO TO 30
      K1=KQUANT(1,I)
      K2=KQUANT(2,I)
      I1=ISIGN(1,K1)
      I2=ISIGN(1,K2)
      K1=IABS(K1)
      K2=IABS(K2)
      FUNCT(NX)=(I1*ZHPTS(K1))-(I2*ZHPTS(K2))
      GO TO 50
30     DO 35 J=1,3
      KZ=KQUANT(J,I)
      IS=ISIGN(1,KZ)
      KZ=IABS(KZ)
      KH=KZ+KDIM
      H(J)=IS*ZHPTS(KH)
35     Z(J)=ZHPTS(KZ)
      IF (KSPACE(1,I).EQ.0) GO TO 45
      DO 40 J=2,3
      CALL IMAGE(H(J),Z(J),KSPACE(J,I),KSPACE(1,I))
45     FUNCT(NX)=(Z(1)-Z(2))-(H(1)-H(2))*(Z(2)-Z(3))/(H(2)-H(3))
50     FDIFR(1,NX)=FUNCT(NX)-PQUANT(I)
55     PQUANT(I)=FUNCT(NX)
      DELTA=0.
      DO 60 I=1,NX
      IF (ABS(FUNCT(I)).GT.ABS(DELTA)) DELTA=FUNCT(I)
      CALL DISPLA(" ",DELTA)
      IF (ABS(DELTA).LT.DELMAX) RETURN DONE
      DO 70 J=1,NX
      DO 65 I=1,NX
65     PDERIV(I,J)=0.
      PDERIV(J,J)=FDIFR(1,J)/XDIFR(1,J)

```

```

70  FUNCT(J)=-FUNCT(J)
    IF ((ITER.LE.NX).OR.(INDVAR.GT.0)) GO TO 80
    DO 75 J=1,NX
    D=DETERM(XDIFR,TEMP,FDIFR(1,J),NX)
    IF (D.EQ.0.) GO TO 80
    DO 75 I=1,NX
75  PDERIV(J,I)=TEMP(I)
80  D=DETERM(PDERIV,TEMP,FUNCT,NX)
    IF (D.EQ.0.) RETURN DONE
    CALL STACK(XDIFR,NX)
    DO 85 J=1,NX
85  XDIFR(1,J)=TEMP(J)
    IF (NS.EQ.0) GO TO 95
    DO 90 I=1,NS
    IF (ABS(XDIFR(1,I)/ZSTOP(KSTOP(I))).GT.DVMAX) XDIFR(1,I)=
1  SIGN(DVMAX*ZSTOP(KSTOP(I)),XDIFR(1,I))
90  ZSTOP(KSTOP(I))=ZSTOP(KSTOP(I))+XDIFR(1,I)
95  DO 100 I=1,NVARY
    N=NS+I
    IF (ABS(XDIFR(1,N)/VARI(KVARIB(I))).GT.DVMAX) XDIFR(1,N)=
1  SIGN(DVMAX*VARI(KVARIB(I)),XDIFR(1,N))
    PVARIB=VARI(KVARIB(I))
    VARI(KVARIB(I))=VARI(KVARIB(I))+XDIFR(1,N)
    IF (KVARIB(I).LT.4) GO TO 100
    IF (LVARIB(I).EQ.0) GO TO 100
    VARI(LVARIB(I))=(VARI(KVARIB(I))-YVARY(I))/SVARY(I)+YBVARY(I)
100 CONTINUE
    RETURN
    END

```

SUBROUTINE STACK(A,N)

```

DIMENSION A(5,5)

```

```

DO 1 K=2,N

```

```

I=N-K+2

```

```

IM1=I-1

```

```

DO 1 J=1,N

```

```

1  A(I,J)=A(IM1,J)

```

```

RETURN

```

```

END

```

SUBROUTINE SOLVE

```

COMMON /CBEAM/ ZUPPER(20),HUPPER(20),ZLOWER(20),HLOWER(20),

```

```

1 ZINTER(20),HINTER(20),ZPTS(20),HPTS(20),NUPPER,NLOWER,NINTER,

```

```

2 LINES(20,2),ZPORT(2),KUPPER(20),KLOWER(20),MOPTIC(20,3)

```

```

3 ,NWARN,IWARN(20,2),NAMINT(2,20)

```

```

COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,

```

```

1 NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,DBFOV,

```

```

2 IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC

```

```

3 ,HPLLOT

```

```

COMMON /CVARY/ KVARIB(3),LVARIB(3),SVARY(3),YVARY(3),FDIFR(5,5),

```

```

1 YBVARY(3),KQUANT(3,10),XDIFR(5,5),PQUANT(3),NVARY,NSOLV,ITER,DEL

```

```

2 ,KSPACE(3,10),INDVAR

```

```

DIMENSION ZHPTS(160),H(3),Z(3)

```

```

EQUIVALENCE (ZHPTS(1),ZUPPER(1))

```

```

KQ=NSOLV+3

```

```

DO 50 I=4,KQ

```

```

IF (KQUANT(3,I).NE.0) GO TO 30

```

```

K1=KQUANT(1,I)

```

```

K2=KQUANT(2,I)

```

```

I1=ISIGN(1,K1)

```

```

I2=ISIGN(1,K2)

```

```

K1=IABS(K1)
K2=IABS(K2)
ZHPTS(K1)=I1*I2*ZHPTS(K2)
GO TO 50
30 DO 35 K=1,3
   J=4-K
   KZ=KQUANT(J,I)
   IS=ISIGN(1,KZ)
   KZ=IABS(KZ)
   KH=KZ+KDIM
   H(J)=IS*ZHPTS(KH)
35 Z(J)=ZHPTS(KZ)
   IF (KSPACE(1,I).EQ.0) GO TO 45
   DO 40 J=2,3
40 CALL IMAGE(H(J),Z(J),KSPACE(J,I),KSPACE(1,I))
45 ZHPTS(KH)=IS*YYBAR(H,Z,Z(1),2,3)
50 CONTINUE
   IF (IPRINT.LT.0) CALL SHOWU
   RETURN
   END

```

SUBROUTINE PLOTBS

C..PLOT THE BOUNDARIES OF THE BEAM ON THE CALCOMP PLOTTER

```

COMMON /COPTIC/ OBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1 ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2 ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3 ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
COMMON /CBEAM/ ZUPPER(20),HUPPER(20),ZLOWER(20),HLOWER(20),
1 ZINTER(20),HINTER(20),ZPTS(20),HPTS(20),NUPPER,NLOWER,NINTER,
2 LINES(20,2),ZPORT(2),KUPPER(20),KLOWER(20),MOPTIC(20,3)
3 ,NWARN,IWARN(20,2),NAMINT(2,20)
COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1 NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2 IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3 ,HPLOT
COMMON /CRAYS/ YRAYS(20,5),ZRAYS(20,5),YA(20),ZA(20),YO(20),ZO(20)
YMAX=8.8
XMAX=ABS(HPLOT)
HMAX=ZMAX=-BIG
ZMIN=BIG
DO 10 I=1,NUPPER
   IF (HUPPER(I).GT.HMAX) HMAX=HUPPER(I)
   IF (ZUPPER(I).GT.ZMAX) ZMAX=ZUPPER(I)
10 IF (ZUPPER(I).LT.ZMIN) ZMIN=ZUPPER(I)
ZSCALE=(ZMAX-ZMIN)/YMAX
HSCALE=-ZSCALE
IF (HPLOT.GT.0.) HSCALE=-HMAX/XMAX
CALL INITIAL(0,13,0,0,0)
CALL PLOT(XMAX,1,1,-1)
CALL PLOT(0,-1,3)
CALL PLOT(0,YMAX+.1,2)
IF (HPLOT.EQ.-3.) GO TO 15
NP1=NSURF+1
NP2=NSURF+2
ZA(NP1)=ZO(NP1)=ZMIN
YA(NP1)=YO(NP1)=0.
ZA(NP2)=ZO(NP2)=ZSCALE
YA(NP2)=YO(NP2)=HSCALE
CALL LINE(YA,ZA,NSURF,1,0)
CALL LINE(YO,ZO,NSURF,1,0)
DO 20 J=1,4

```

```

ZRAYS(NP1,J)=ZMIN
YRAYS(NP1,J)=0.
ZRAYS(NP2,J)=ZSCALE
YRAYS(NP2,J)=HSCALE
20 CALL LINE(YRAYS(1,J),ZRAYS(1,J),NSURF,1,0)
15 NP1=NUPPER+1
NP2=NUPPER+2
ZUPPER(NP1)=ZMIN
HUPPER(NP1)=0.
ZUPPER(NP2)=ZSCALE
HUPPER(NP2)=HSCALE
CALL LINE(HUPPER,ZUPPER,NUPPER,1,0)
NP1=NLOWER+1
NP2=NLOWER+2
ZLOWER(NP1)=ZMIN
HLOWER(NP1)=0.
ZLOWER(NP2)=ZSCALE
HLOWER(NP2)=HSCALE
CALL LINE(HLOWER,ZLOWER,NLOWER,1,0)
DO 30 I=1,NOPTI
YV=(ZVSURF(I+1)-ZMIN)/ZSCALE
CALL PLOT(-.05,YV,3)
CALL PLOT(.05,YV,2)
ML=MOPTIC(I,2)
XL=HLOWER(ML)/HSCALE
YL=(ZLOWER(ML)-ZMIN)/ZSCALE
CALL PLOT(XL,YL,3)
MU=MOPTIC(I,1)
IF (HUPPER(MU)*HLOWER(ML).LT.0.) CALL PLOT(0.,YV,2)
XU=HUPPER(MU)/HSCALE
YU=(ZUPPER(MU)-ZMIN)/ZSCALE
CALL PLOT(XU,YU,2)
IF (MOPTIC(I,3).NE.5LREFL ) GO TO 30
YB=YV+RINDEX(I)*ABS(XU)/5.
CALL PLOT(XU,YB,2)
CALL PLOT(XL,YB,2)
CALL PLOT(XL,YL,2)
30 CONTINUE
X=YBSURF(NSURF)/HSCALE
Y=(ZVSURF(NSURF)-ZMIN)/ZSCALE
CALL PLOT(0.,Y,3)
CALL PLOT(X,Y,2)
CALL ENDPLT
RETURN
END

```

```

SUBROUTINE APRTIN
COMMON /CBEAM/ ZUPPER(20),HUPPER(20),ZLOWER(20),HLOWER(20),
1 ZINTER(20),HINTER(20),ZPTS(20),HPTS(20),NUPPER,NLOWER,NINTER,
2 LINES(20,2),ZPORT(2),KUPPER(20),KLOWER(20),MOPTIC(20,3)
3 ,NWARN,IWARN(20,2),NAMINT(2,20)
COMMON /COPTIC/ OBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1 ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2 ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3 ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1 NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2 IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3 ,HPLQT
COMMON /CDBJCTS/ NOBJS,NCARDS,NBAFF,NREP,IREP(24),YCK(20)
COMMON /CSTOP/ ZASTOP,ZOSTOP,KASTOP,KOSTOP,LASTOP,LOSTOP,

```

```

1 PDZSTP(2)
  COMMON /H/ IHID(8,10),NCMNTS,KI,KO
  DIMENSION ZCK(5),HCK(5),IMAGE(8),IBAF(5)
  NBAFF=0
  NCARDS=NCMNTS
  IF (NCMNTS.EQ.0) GO TO 7
  DO 5 I=1,NCMNTS
5  WRITE(21,90) (IHID(J,I),J=1,8)
7  DO 10 I=1,NSURF
10 YCK(I)=YSURF(I)*(1.+OBSCUR)/2.
    NOPTI=NOPTI+1
    MOPTIC(NOPTI,1)=NUPPER
    MOPTIC(NOPTI,2)=NLOWER
    MOPTIC(NOPTI,3)=5LDISK
    RINDEX(NSURF)=-RINDEX(NOPTI)
    NI=NINTER
    IF (NINT.GT.0) NI=NINT
    NU1=NL1=1
    N1=N2=NOBJS=0
12  N2=N2+1
    IF (N2.GT.NOPTI) GO TO 62
    IF (MOPTIC(N2,3).NE.5LREFL) GO TO 15
    M2M1=MOPTIC(N2,1)-1
    IF ((N2.LT.2).OR.(M2M1.LT.3)) GO TO 12
    KTYPE=KUPPER(M2M1).AND.MASKS(2)
    IF (KTYPE.NE.5RIMAGE) GO TO 12
15  NREP=0
    NU2=MOPTIC(N2,1)
    NL2=MINO(NLOWER,MOPTIC(N2,2))
C..UPPER
    NM1=NU2-1
    DO 40 I=NU1,NM1
    NCK=1
    IP1=I+1
    IBAF(1)=IBAFF=0
    DO 20 J=1,NI
    IF (ABS(ZINTER(J)-ZUPPER(I)).GT.DELMAX) GO TO 18
    IBAF(1)=J
    GO TO 20
18  IF (ABS(ZINTER(J)-ZUPPER(IP1)).GT.DELMAX) GO TO 19
    IBAFF=J
    GO TO 20
19  IF ((LINES(J,1).NE.I).AND.(LINES(J,2).NE.I)) GO TO 20
    NCK=NCK+1
    ZCK(NCK)=ZINTER(J)
    HCK(NCK)=HINTER(J)
    IBAF(NCK)=J
20  CONTINUE
    ZCK(1)=ZUPPER(I)
    HCK(1)=HUPPER(I)
    NCK=NCK+1
    ZCK(NCK)=ZUPPER(IP1)
    HCK(NCK)=HUPPER(IP1)
    IBAF(NCK)=IBAFF
    NCKM1=NCK-1
    DO 40 K=1,NCKM1
    DO 30 L=1,NCKM1
    M=K+L
    IF (M.GT.NCK) GO TO 40
    DZ=ZCK(K)-ZCK(M)
    DH=HCK(K)-HCK(M)

```

```

D=SQRT(DZ**2+DH**2)
IF (D.LT.DELMAX) GO TO 30
IF (ABS(DZ).LT.DELMAX) GO TO 28
IBAFF=MAXO(IBAF(K),IBAF(M))
CALL OBJECT(5L CONE,-1,0,ZCK(K),HCK(K),ZCK(M),HCK(M),IBAFF)
GO TO 30
28  NORM=SIGN(1.,DZ)
    CALL OBJECT(5L DISK,NORM,0,ZCK(K),HCK(K),ZCK(M),HCK(M),0)
30  CONTINUE
40  CONTINUE
    LAST=NOBJS
    IF ((N1.GT.0).OR.(HLOWER(1).EQ.0.)) GO TO 42
    CALL OBJECT(5L DISK,-1,0,ZLOWER(1),HLOWER(1),ZLOWER(1),0.,0)
    NCARDS=NCARDS+1
    WRITE(21,41)
41  FORMAT(* =INNER ENTRANCE PORT*)
C..LOWER
42  IF (NL2.LE.NL1) GO TO 47
    NM1=NL2-1
    DO 45 I=NL1,NM1
        NCK=1
        IP1=I+1
        IBAF(1)=IBAFF=0
        DO 25 J=1,NI
            IF (ABS(ZINTER(J)-ZLOWER(I)).GT.DELMAX) GO TO 48
            IBAF(1)=J
            GO TO 25
48  IF (ABS(ZINTER(J)-ZLOWER(IP1)).GT.DELMAX) GO TO 49
        IBAFF=J
        GO TO 25
49  IF ((LINES(J,1).NE.-I).AND.(LINES(J,2).NE.-I)) GO TO 25
        NCK=NCK+1
        ZCK(NCK)=ZINTER(J)
        HCK(NCK)=HINTER(J)
        IBAF(NCK)=J
25  CONTINUE
        ZCK(1)=ZLOWER(I)
        HCK(1)=HLOWER(I)
        NCK=NCK+1
        ZCK(NCK)=ZLOWER(IP1)
        HCK(NCK)=HLOWER(IP1)
        IBAF(NCK)=IBAFF
        NCKM1=NCK-1
        DO 45 K=1,NCKM1
            DO 35 L=1,NCKM1
                M=K+L
                IF (M.GT.NCK) GO TO 45
                IF ((HCK(K).LT.DELMAX).AND.(HCK(M).LT.DELMAX)) GO TO 35
                DZ=ZCK(K)-ZCK(M)
                DH=HCK(K)-HCK(M)
                D=SQRT(DZ**2+DH**2)
                IF (D.LT.DELMAX) GO TO 35
                IF (ABS(DZ).LT.DELMAX) GO TO 38
                IBAFF=MAXO(IBAF(K),IBAF(M))
                CALL OBJECT(5L CONE,1,0,ZCK(K),HCK(K),ZCK(M),HCK(M),IBAFF)
                GO TO 35
38  NORM=SIGN(1.,-DZ)
        CALL OBJECT(5L DISK,NORM,0,ZCK(K),HCK(K),ZCK(M),HCK(M),0)
35  CONTINUE
45  CONTINUE
C..OPTICS

```



```

47  NS=N1+1
    DO 50 I=NS,N2
      IP1=I+1
      ITYPE=MOPTIC(I,3)
      CALL OBJECT(ITYPE,NORM,I,Z1,H1,Z2,H2,0)
      IF (ITYPE.NE.5LOPTIC) GO TO 50
      NCARDS=NCARDS+2
      IF (NAMSRF(1,IP1).EQ.10H ) GO TO 70
      WRITE(21,71) (NAMSRF(K,IP1),K=1,2)
      GO TO 72
70  WRITE(21,51) I,MOPTIC(I,3)
72  IF (ITRACE.EQ.0) WRITE(21,55) YSURF(IP1),YBSURF(IP1,2)
      IF (ITRACE.NE.0) WRITE(21,52) RDSURF(IP1)
      IF (ABS(RINDEX(IP1)).EQ.1.) GO TO 58
      NCARDS=NCARDS+1
      WRITE(21,56) RINDEX(IP1)
58  IF (CCSURF(IP1).EQ.0.) GO TO 59
      NCARDS=NCARDS+1
      WRITE(21,57) CCSURF(IP1)
59  IF (MOPTIC(I,3).NE.5LREFL ) GO TO 50
      NCARDS=NCARDS+1
      WRITE(21,60) (IREP(J),J=1,NREP)
50  CONTINUE
51  FORMAT(* =OPTICAL SURF*,I3,*,*,A4)
71  FORMAT(* =*,2A10)
55  FORMAT(* Y-YBAR*,2G12.5)
52  FORMAT(* RD*,G12.5)
56  FORMAT(* INDEX*,G12.5)
57  FORMAT(* CC*,G12.5)
60  FORMAT(* REPEAT*,24I3)
    N1=N2
    NU1=MOPTIC(N1,1)
    NL1=MOPTIC(N1,2)
    GO TO 12
62  NCARDS=NCARDS+7
      BIGGER=1.+DELMAX
      SMALLR=1.-DELMAX
      HSORS=-SMALLR*HUPPER(1)
      COBS=BIGGER*OBSCUR*DAPERT/2.
      ZPORT(1)=ZPORT(1)-DELMAX
      ZDET=ZVSURF(NSURF)-SIGN(DELMAX,RINDEX(NSPACE))
      IWIND=ISIGN(LAST,INT(RINDEX(NSPACE)))
      WRITE(21,65) YSURF(NSURF),YBSURF(NSURF,2),COBS,1,NSPACE,
1  HSORS,ZPORT(1),-1,NSPACE,1,0.,ZDET,IWIND
65  FORMAT(* =FINAL IMAGE PLANE*/ * Y-YBAR*,2G12.5/*COBSCUR*,G12.5/
1  *SCAN*/2I3,2G12.5,I3/2I3,2G12.5,I3/*READ 1*)
      REWIND 21
      WRITE(4,110)
110  FORMAT(1H1,*APART1 INPUT (TAPE21):*/ )
      DO 80 I=1,NCARDS
        READ(21,90) IMAGE
80  WRITE(4,100) IMAGE
90  FORMAT(8A10)
100  FORMAT(1X,8A10)
      REWIND 21
      RETURN
      END

```

```

SUBROUTINE OBJECT(ITYPE,NORM,IO,Z1,H1,Z2,H2,IBAFF)
COMMON /CBEAM/ ZUPPER(20),HUPPER(20),ZLOWER(20),HLOWER(20),
1 ZINTER(20),HINTER(20),ZPTS(20),HPTS(20),NUPPER,NLOWER,NINTER,

```

```

2  LINES(20,2),ZPORT(2),KUPPER(20),KLOWER(20),MOPTIC(20,3)
3  ,NWARN,IWARN(20,2),NAMINT(2,20)
   COMMON /COPTIC/ OBSCUR,FOVDEG,DAPERT,YSURF(20),YBSURF(20,2)
1  ,ZHE,RINDEX(20),ZVSURF(20),NSURF,NSPACE,NOPTI,NAMSRF(2,20)
2  ,OMEGA(20),OMEGAB(20),RDSURF(20),KFIGR(4),NFIGR,KFIGR,NSFIGR
3  ,CVSURF(20),CCSURF(20),KSURF(20),ASURF(20,4),NASURF(20)
   COMMON /CEXTRA/ BIG,SMALL,CONV,FNUMBR,APTFAC,FOVFAC,WAVEL,IMBEST,
1  NITER,DELMAX,XEQ,ITRACE,KBEAM,DELONE,DVMAX,NINT,KDIM,OBFOV,
2  IVIGN,TOLER,IPRINT,DAXIS,DFIELD,HFIELD,SMAG,MASKS(2),HSEC,ZSEC
3  ,HPLLOT
   COMMON /COBJCTS/ NOBJS,NCARDS,NBAFF,NREP,IREP(24),YCK(20)
   DIMENSION IMAGE(9),ICOL1(90),ICOL2(80)
   IF (ITYPE.NE.5L CONE) GO TO 35
20  DO 30 I=1,NSPACE
     IP1=I+1
     CALL SEGINT(H1,H2,Z1,Z2,YCK(I),YCK(IP1),ZVSURF(I),ZVSURF(IP1),
1  HI,ZI,INT,0.)
     IF (INT.GT.0) RETURN
     IF (KBEAM.GT.0) GO TO 30
     CALL SEGINT(H1,H2,Z1,Z2,-YCK(I),-YCK(IP1),ZVSURF(I),ZVSURF(IP1),
1  HI,ZI,INT,0.)
     IF (INT.GT.0) RETURN
30  CONTINUE
     GO TO 37
35  IF (ITYPE.NE.5L DISK) GO TO 40
37  NREP=NREP+1
     NOBJS=NOBJS+1
     IREP(NREP)=NOBJS
     GO TO 50
40  IF (IO.EQ.0) GO TO 50
     IOI=IO+1
     Z1=Z2=ZVSURF(IOI)
     H1=HUPPER(MOPTIC(IO,1))
     H2=HLOWER(MOPTIC(IO,2))
     NORM=X=RINDEX(IOI)
     IF (ITYPE.EQ.5LGLAS ) NORM=0
     NOBJS=NOBJS+1
50  NCARDS=NCARDS+1
     HS=-HSEC
     DH=ABS(H1-H2)
     IF (HS.LT.0.) HS=DH/HSEC
     ZS=-ZSEC
     DZ=ABS(Z1-Z2)
     IF (ZS.LT.0.) ZS=DZ/ZSEC
     NSEC=(DH*HS+DZ*ZS)/(DH+DZ)+1
     IF (NSEC.LT.1) NSEC=1
     IF (NSEC.GT.10) NSEC=10
     IF (ITYPE.NE.5L DISK) GO TO 55
     ENCODE(90,60,IMAGE) ITYPE,NOBJS,NORM,NSEC,Z1,H1,H2
     GO TO 100
55  IF (ITYPE.NE.5L CONE) GO TO 70
     ENCODE(90,60,IMAGE) ITYPE,NOBJS,NORM,NSEC,Z1,H1,Z2,H2
60  FORMAT(A5,3I3,6G12.5)
     GO TO 100
70  IF (ITYPE.NE.5L DISK ) GO TO 80
     ENCODE(90,60,IMAGE) ITYPE,NOBJS,NORM,NSEC,Z1,H1,H2
     GO TO 100
80  ITYPE=5LOPTIC
     ENCODE(90,60,IMAGE) ITYPE,NOBJS,NORM,NSEC,Z1,H1,H2
100  DECODE(90,90,IMAGE) ICOL1
90  FORMAT(90A1)

```

```

N=0
DO 130 J=1,89
JP1=J+1
IF ((ICOL1(J).EQ.1H ).AND.(ICOL1(JP1).EQ.1H )) GO TO 130
N=N+1
ICOL2(N)=ICOL1(J)
130 CONTINUE
N=N+1
DO 140 J=N,80
140 ICOL2(J)=1H
WRITE(21,90) (ICOL2(I),I=1,80)
IF (ITYPE.NE.5L CONE) RETURN
NCARDS=NCARDS+1
IF ((IBAFF.EQ.0).AND.(NOBJS.EQ.1)) GO TO 150
KTYPE=5HOUTER
IF (NORM.LT.0) KTYPE=5HINNER
IF (IBAFF.EQ.0) GO TO 170
IF (NAMINT(1,IBAFF).NE.10H ) GO TO 180
GO TO 110
170 NBAFF=NBAFF+1
WRITE(21,125) KTYPE,NBAFF
RETURN
110 WRITE(21,120) KTYPE,IBAFF
120 FORMAT(*  =*,A5,* BAFFLE*,I3.0)
125 FORMAT(*  =*,A5,* BAFFLE*,1X,R1)
RETURN
150 WRITE(21,160)
160 FORMAT(*  =MAIN BAFFLE*)
RETURN
180 WRITE(21,190) KTYPE,(NAMINT(I,IBAFF),I=1,2)
190 FORMAT(*  =*,A5,1X,2A10)
RETURN
END

```

LIST OF REFERENCES

- Bartell, F., R. Breault, A. DeBell, B. Fannin, A. Greynolds, J. Gunderson, J. Harvey, S. Lange, R. Shack, F. Turner, W. Wolfe, R. Wooden, and D. Goodman, A Study Leading to Improvements in Radiation Focusing and Control in Infrared Sensors, Final Report for D/A Project 8X3636604D215, Report No. AMMRC CTR 76-42 (1976).
- Breault, R. P., "Problems and Techniques in Stray Radiation Suppression," Proc. SPIE 107:2-23 (1977).
- Cornejo, A., and N. Malacara, "Direct Design Solution for Cassegrain Shields," D. Bol. de los Obs. Tonantzintia y Tacubaya, 30, Vol. 4 (1968).
- Delano, E., "First Order Design and the Y- $\bar{Y}$ Diagram," Appl. Opt. Vol. 2, #12 (1963).
- Lange, S., R. Breault, and A. Greynolds, "APART, A First-order Deterministic Stray Radiation Analysis Program," Proc. SPIE, Vol. 107:89-97, April (1977).
- Lopez-Lopez, F. J., The Application of the Delano Y- $\bar{Y}$ Diagram to Optical Design, Ph. D. dissertation, University of Arizona, Tucson (1973).
- Prescott, R., "Cassegrainian Baffle Design," Appl. Opt. 7, 479 (1968).
- Shack, R. V., "Analytic System Design with a Pencil and Ruler--The Advantages of the Y-Y Diagram," Proc. SPIE, Vol. 39, August (1973).
- Welford, W. T., Aberrations of the Symmetric Optical System, Academic Press, New York (1974).
- Young, A. T., "Design of Cassegrain Light Shields," Appl. Opt. 6, 1063 (1967).