USER'S MANUAL FOR
UNIVERSITY OF ARIZONA APART PROGRAM
(ANALYSIS PROGRAM-ARIZONA RADIATION TRACE)

Robert P. Breault

NASA contract NAS 8-27804

August 1975

Optical Sciences Center
University of Arizona
Tucson, Arizona 85721

FOREWORD

This report describes an important new optical system analysis tool, the Analysis Program, Arizona Radiation Trace (APART). This is a computer program that is able to efficiently and accurately predict the off-axis rejection characteristics of unwanted stray radiation for complex optical systems. This program was developed by the University of Arizona under NASA contract NAS 8-27804 (MSFC--Mr. Don Griner, Technical Monitor) to aid in the design of the Large Space Telescope (LST).

The LST is a large-aperture stellar telescope intended for operation in a low-altitude earth orbit. For maximum utilization, it is necessary that the LST operate while on the "bright" side of the earth, where it is exposed to strong interfering radiation from both the sun and the bright earth. The desired performance goal for the LST is the detection of M_V 29 objects, so there is clearly a need for sophisticated light rejection systems that reduce the transmittance of the radiation by a factor of 10^{14}.

Dr. Roland Shack of the Optical Sciences Center originally conceived the unique approach used for the development of APART and has contributed invaluable aid and guidance during the development. Mr. Tom Sargent of Steward Observatory contributed a great deal to developing the more complex portions of the computer program, and Mr. Gary Hunt, a Sperry-Rand subcontractor at MSFC, was very helpful in the final stages of the program development.

Special thanks go to Mr. Bill Fannin, Research Associate, whose patience helped me endure some of the more trying moments.

ABSTRACT

The Analysis Program, Arizona Radiation Trace (APART) demonstrates the capability of using modern-day, high-speed computers to do deterministic radiation calculations on rotationally symmetric optical systems. The advent of computers with large central memory and/or random access disk files has made the necessary computations feasible. The basic problems have been solved and an extension of the same techniques to general systems, i.e., nonrotationally symmetric systems, can be foreseen.

Routines to model vane structure proved to be both convenient to use and instructive in clarifying the scattering mechanism of the vane structure subsystem including the effect of vane edge scatter.

Although the run time of the program was to be considerably reduced compared to that of other programs in existence that measure stray radiation, this is not necessarily true. Program three may run a relatively long time. But once generated, subsequent runs will, in fact, be shorter as that portion of the analysis need not be rerun.

Caution should be exercised when using refracting optical systems. The program has not been tried on any such system although it was intended to be able to handle them. Later versions presently being developed are designed to handle this problem along with asymmetric systems in general.

CONTENTS

I. INTRODUCTION

Before the start of the development of this program, most of the sophisticated available stray radiation programs used the unmodified Monte-Carlo statistical techniques. For systems with high attenuation this involves long computer runs. Subsequent runs are just as expensive. Since that time the GUERAP programs have been developed or released thereby reducing many of the disadvantages of earlier approaches. The Arizona program (APART) is a unique deterministic program because it utilizes the y-$\bar{y}$ techniques to determine the critical objects or areas that scatter radiation to the image plane either directly or through imaging elements. Thus, even before expensive radiation calculations begin, one has the opportunity to modify the design so that the number of critical areas "seen" by the image plane is reduced or the radiation to these critical areas is minimized.

Once radiation calculations are made, APART not only gives the power distribution reaching the image plane (as do most other programs), but also gives a sectional power map of all internal surfaces, so one can understand the power distributions involved. Any surfaces that scatter power onto the image plane are identified, and the increment of power is printed out. With this information it is possible to determine which sections of the optical system are the major contributors of scattered radiation, so that any system redesign can concentrate on the problem areas. In addition to the power mapping information, angular information is provided that relates the angle by which the radiation came into a surface to the angle by which the radiation is scattered out of the surface; this permits one to determine if a redesign of the baffle system would be beneficial.

Although APART is technically *not* a design program, its analysis is so complete that it usually clearly defines what must be done to improve the system. Most other programs give final answers only, with little or no diagnostic information, so that attempts to improve system performance are largely a trial and error approach, with little understanding of the basic principles involved. In addition, APART is broken down into five subprograms so that additional computer runs can usually be made at very low cost if no major changes are made in the system. For example, changes in baffle vane structure or surface characteristics may be evaluated without regenerating the power transfer relationships, so that reruns require a few dollars of computer time rather than the hundreds of dollars for a complete evaluation.

Because APART was developed specifically to analyze proposed designs for the LST, in its present form it is in several respects unsuited for the analysis of general optical systems. The most important of the limitations are

(1) The optical system must be rotationally symmetrical about its central axis (i.e., no off-axis mirrors, no noncircular apertures, etc.). However, a number of nonsymmetrical designs may be handled by such techniques as introducing "dummy" surfaces to complete the symmetry, and assigning these surfaces zero reflection characteristics, etc.

(2) APART is not intended to handle transmission optical elements (refracting elements); the optical system is no problem, but the program has never been checked out for such systems.

(3) APART has no provisions for predicting the diffraction effects of a bright radiation source close to the field of view impinging on a baffle edge. (This problem does not arise with the LST because the look angle of the telescope is restricted to angles somewhat removed from bright objects such as the sun or bright earth.)

It is hoped that an improved version of APART will be developed that will remove these (and other) restrictions and will make the program a more universal tool.

II. GENERAL OVERVIEW OF THE APART PROGRAM

There are five basic programs to the University of Arizona computer analysis program for the study of stray radiation suppression. It is intended that programs two through five be run separately but sequentially as one job train, each creating files for the succeeding programs. Program one is the only one that does not create nor depend on any external files. Figure 1 shows the interrelationships of the programs in APART.

Program one looks out from a point in the image plane and determines those objects that are seen from this point. These are the sources of scattered radiation reaching that point in the image plane. This program, along with the remaining programs, can handle only rotationally symmetric objects. Program one indicates those surfaces that are seen and the angle at which they are seen in *real* space. Any surface seen is divided into five segments; the position and angle of each segment is printed out. This initial program helps to quickly see any major flaws in the design (from a scattered radiation standpoint), and will reduce the amount of calculations needed to be executed by the succeeding programs.

Program two is similar to program one; both use the same data deck input and both lay out the imaged space as seen from the image plane. Program two does not calculate what segments are seen or any angular information; instead, it lays out the image space as seen from object space. That is, "Looking into the first imaging aperture, what do the imaged spaces look like?" This is necessary to determine to what sections the radiation can scatter. This is also done for the space between the primary and secondary, which turns out to be a partial combination of the other two cases. Although this may appear redundant in the case of two-mirror systems, it may not be so for other systems. The output is plotted for each space so that one may see how and where real space objects are imaged or reimaged. The printout gives the numerical data of the imaged objects. This includes distances, heights, y-$\bar{y}$ bar values,

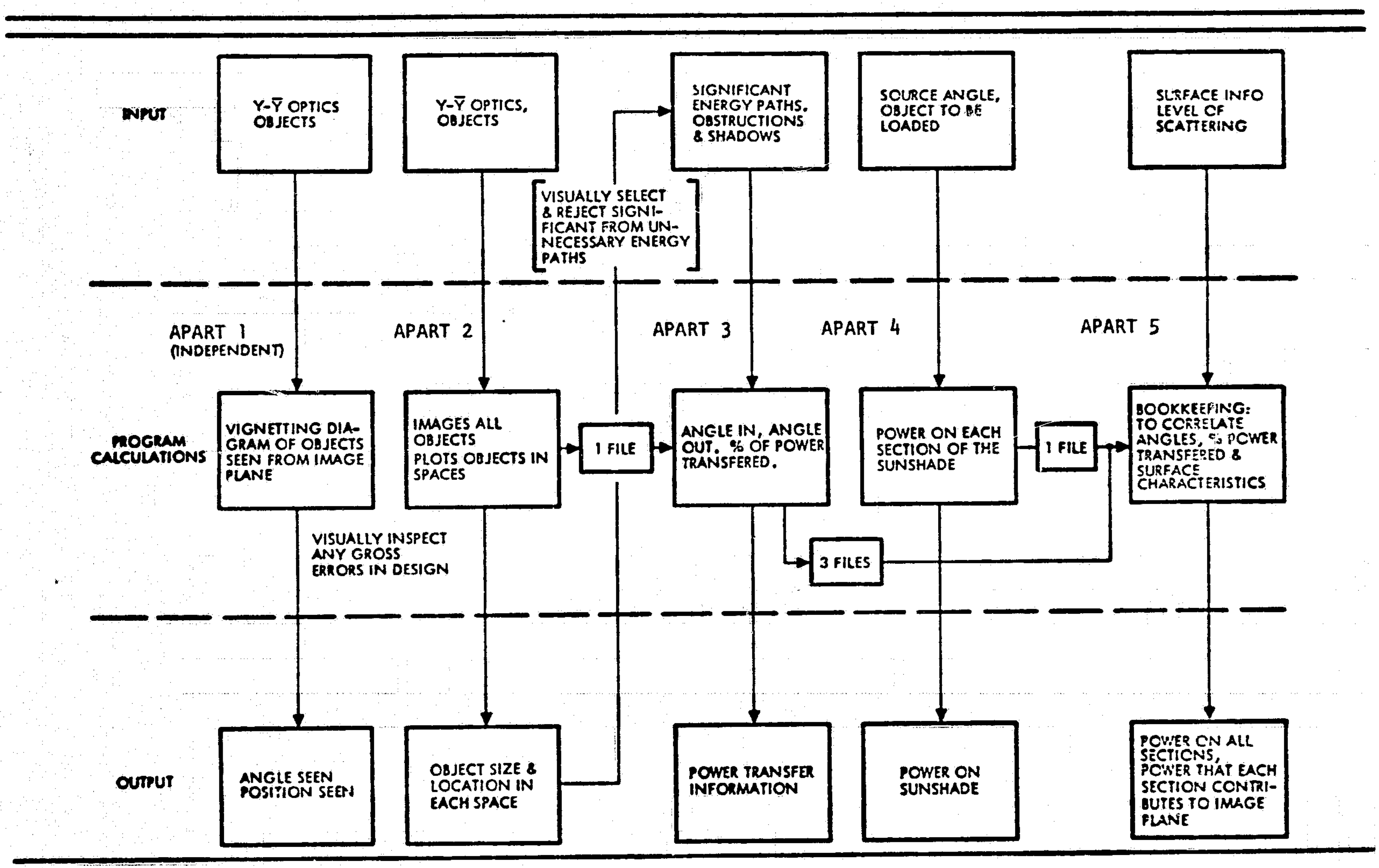

INPUT
PROGRAM CALCULATIONS
OUTPUT
Y-$\bar{Y}$ OPTICS OBJECTS
Y-$\bar{Y}$ OPTICS, OBJECTS
SIGNIFICANT ENERGY PATHS, OBSTRUCTIONS & SHADOWS
SOURCE ANGLE, OBJECT TO BE LOADED
SURFACE INFO LEVEL OF SCATTERING
VISUALLY SELECT & REJECT SIGNIFICANT FROM UNNECESSARY ENERGY PATHS
APART 1 (INDEPENDENT)
APART 2
APART 3
APART 4
APART 5
VIGNETTING DIAGRAM OF OBJECTS SEEN FROM IMAGE PLANE
IMAGES ALL OBJECTS PLOTS OBJECTS IN SPACES
1 FILE
ANGLE IN, ANGLE OUT, % OF POWER TRANSFERED.
POWER ON EACH SECTION OF THE SUNSHADE
1 FILE
BOOKKEEPING: TO CORRELATE ANGLES, % POWER TRANSFERED & SURFACE CHARACTERISTICS
3 FILES
VISUALLY INSPECT ANY GROSS ERRORS IN DESIGN
ANGLE SEEN POSITION SEEN
OBJECT SIZE & LOCATION IN EACH SPACE
POWER TRANSFER INFORMATION
POWER ON SUNSHADE
POWER ON ALL SECTIONS, POWER THAT EACH SECTION CONTRIBUTES TO IMAGE PLANE

Fig. 1. Interrelationships of the programs in APART.

and any objects whose image passes through infinity. It creates an important file (BASICA) that is necessary for the succeeding programs.

Program three attaches most of the information it needs from BASICA. It will take any object from program two and treat it as a source of radiation to any other object, including imaged objects. The data input at this stage can be as simple as two numbers--source object number and collector object number. This program creates four important files: (1) the geometrical percentage of power transferred from each segment of the sources to each segment of the collectors (BDRDF), (2) the angle out of the source (ANGOUT), (3) the angle into the collector (ANGIN), and (4) more information is added to BASICA.

Program four is one of several programs that loads the sun shield with power. One version is a general power-loading program whereas others are more specialized and intended for specific optical designs or to provide a model earth, etc. This information is passed on tape two.

Program five is the driver program for determining the scattered radiation levels throughout the system. The amount of power reaching the image plane from scattered radiation is printed out so that it can be determined which surfaces are the major contributors and just how much power is received from each section. It requires the following files: (1) file BDRDF, file ANGIN, and file ANGOUT from program three, (2) file BASICA from programs two and three, and (3) the input file. The input to this part of the program can be large. The program considers only those radiation paths that contribute scattered radiation to the image plane for the level of scattering being considered; this avoids unnecessary and expensive calculations of scattered energy to surfaces that are not seen from the image plane and do not scatter to surfaces that are seen. This provides the opportunity to study individual radiation paths.

Program five also needs information as to vane configuration--position, vane angle, reflectance of the coating, etc. The total output map of all the energy transfer to each segment is put on tape by a utility program for later analysis.

A. System Flow Chart--Program Functions

Program One

Objects seen from the image plane
Angle seen
Position seen

Program Two
Real and imaged spaces calculated and plotted

Program Three
Calculates the geometrical power transferred
Calculates two necessary angles to specify the angle out of the source
Calculates two necessary angles to specify the angle into the collector

Program Four
Loads power (stray radiation) into the system

Program Five
Calculates the scattered radiation distribution.

B. System Flow Chart--File Generation and Requirements

Program One
Independent

Program Two
Gives plot
Creates file 10 = BASICA

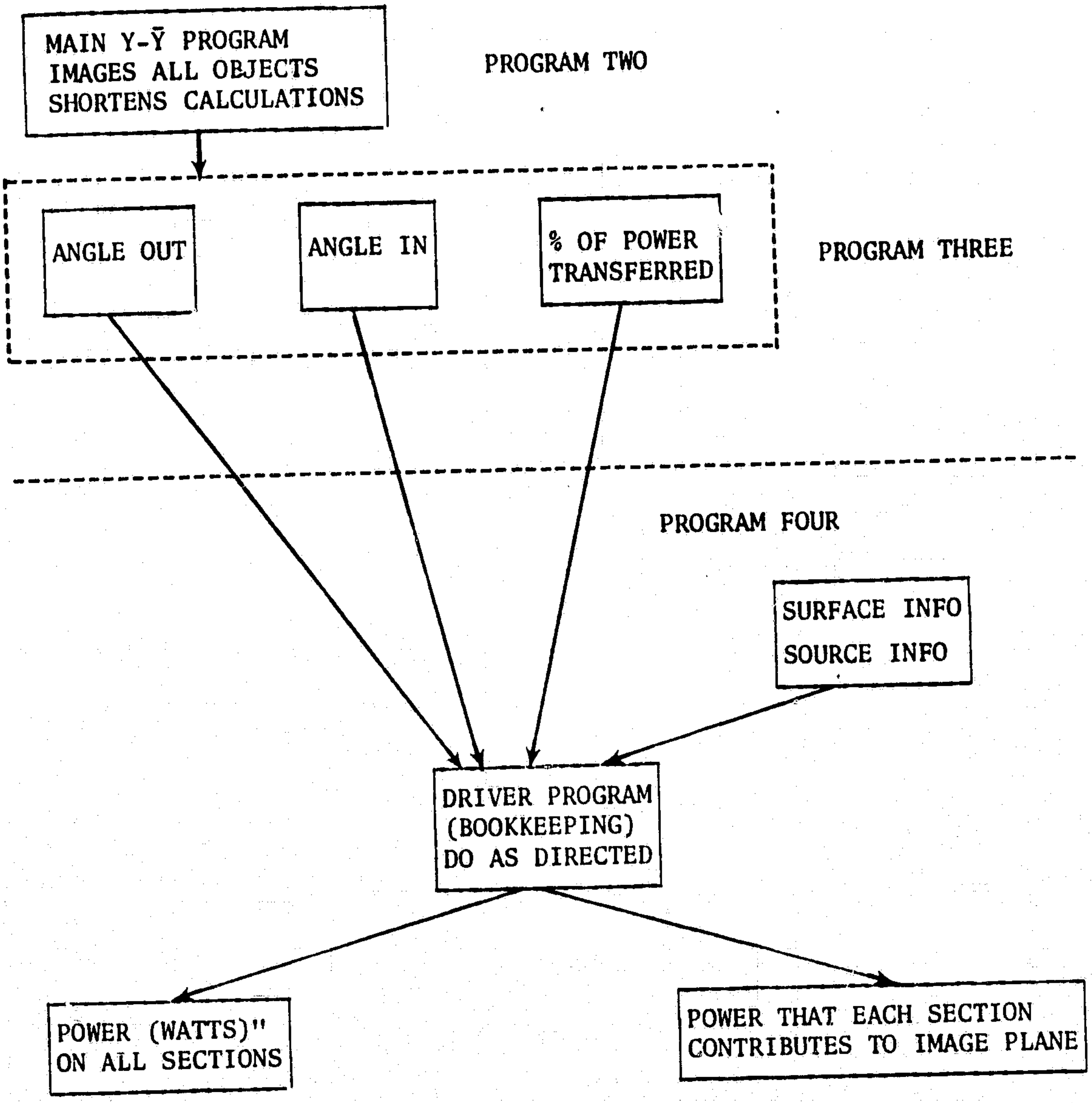
Program Three
Requires tape 10 = BASICA
Adds information to tape 10
Creates file 5 = BDRDF = geometrical power transfered
file 8 = angles out (ANGOUT)
file 9 = angles in (ANGIN)

Program Four
Creates tape 2 = initial power loaded

Program Five
Requires tape 2 = initial power loading
Requires tape 5 = BDRDF
Requires tape 8 = ANGOUT
Requires tape 9 = ANGIN
Requires tape 10 = BASICA
Adds information into tape 10
Adds information into tape 2
Creates tape 3

C. Block Diagram of Programs 2 through 5

(Program one is independent)



D. Program and Subroutine Functions

1. Program One--Object Seen from Image Plane

Program Baffle

Reads in system information:

(1) Baffles
(2) Apertures
(3) Image plane reference point

Calls the subroutines in the desired order

Subroutine SLOPES

Determines the slopes of the lines in the y-$\bar{y}$ diagram

Subroutine YYBARIN

Fills in the missing y and $\bar{y}$ information on the objects

Subroutine SPACES

Determines which space the objects are in, as seen from the image plane

Subroutine PRIMEX

Calculates the imaged positions and y-$\bar{y}$ information

Subroutine POLARS

Puts the spaces seen from the image plane into polar coordinates as seen from the reference point

Subroutine SPLIT

Splits any objects whose image passes through infinity

Subroutine APTOBST

Any surface passing behind a mirror is obstructed by the aperture and any surface behind an imaging element (mirror) cannot be imaged by that surface

Subroutine CLAPT

All imaged objects must pass through the appropriate apertures

Subroutine ORDERP

Orders polar space by angle

Subroutine BAFSEEN

Determines those objects that are seen and that obscure other objects

Subroutine OBJSEEN

Prints out which objects are seen and their imaged space information

Subroutine ANGSEEN

Determines the real space angles and positions of the baffles that are seen

Subroutine ELLIPSE

Projects all the objects into the exit pupil to produce a vignetting type diagram

Subroutine DOUBLEP

Stores the necessary information for OBJSEEN AND ANGSEEN

Subroutine TPOLARX

When an object is split or shortened due to some other baffle or aperture, the new end points are determined

Subroutine ENLARGE

When an object is split, this program enlarges all the necessary arrays, and stores the new information

Subroutine LOWER

Looks for objects seen in the lower half (below the optical axis) of the imaged spaces

2. Program Two--Real and Imaged Spaces

Program CALRAD
Reads the input and calls the subroutines in the desired order

Subroutine SLOPES
Calculates the slopes of the lines in the y-$\bar{y}$ diagram.

Subroutine LAGRANG
Calculates the Lagrange invariant

Subroutine YYBARIN
Determines the missing y-$\bar{y}$ information in real space

Subroutine SPACES
Determines which space the real space objects fall in.

Subroutine IMGLINE
Determines the y value in the plane of the image of the stop as seen from the space being considered. Also determines the $\bar{y}$ value in the plane of the image of the collector object in the space being considered. These two values are very useful in the y-$\bar{y}$ approach.

Subroutine IMAGPRI
The work horse of this program. Projects all objects along the desired image line. It determines the heights and distances as seen from this space. It splits any objects whose image passes through infinity and enlarges the arrays to store them.

Subroutine SEE
Plots the image space as determined by IMAGPRI. The scale is internally set but can be overridden to "blow" up points of interest.

Subroutine LOWER
Restores the original ALAYOUT array to start the next image space.

Subroutine SCALES
Sets the scale of plot if one is not read in.

Subroutine CHANGEA
Changes the relative distances in ALAYOUT to distances from the reference plane in the final space. This information is required for program three.

3. Program Three--Geometrical Power Transfer

Program VIEWFAC

Reads in the sources and collectors, and proceeds to section them in the desired manner. From these subsections the configuration power transfer factors are calculated and stored. The vectors of the line connecting points in real space and imaged space are passed to BDRDF and the important angles--angle out of the source and the angle into the collector--are stored.

Subroutine SETAPT

Determines the critical aperture, if one exists, that the radiation must pass through. If it is an imaged aperture it determines the imaged distance and height.

Subroutine BDRDF

Changes the vector information into a single angle to be used by subroutine SURFACE in program five.

Subroutine VECTOR

Called by BDRDF to create the necessary local system of basis vectors.

Subroutine NORMAL

Normalizes the vectors.

Subroutine MIRRORS

Images all the objects into any desired image space. This 14-line program does the work of a ray-tracing program. It is the basic y-$\bar{y}$ diagram that facilitates the calculations.

Subroutine MIRRORA

Images the apertures for SETAPT.

Subroutine APERT

Determines if any intermediate object or its image shadows the collector so that the energy is not transfered.

4. Program Four--Loader Program

Program SPOTS

This program reads in parcels of power loaded onto the initial source section. Any section on the object can be specified with a specified amount of power. When finished loading the source object, the program processes the data and creates tape 2 for use by program five. The data need to be in a special format acceptable to program five. There are no subroutines to this program.

5. Program Five--Calculation of Radiation Distribution

Program DRIVER

Does the bookkeeping for the power transferred throughout the system. The systems of calculations are controlled by input cards for any level of scattered radiation desired. The benefits realized by this are (1) for the final level of scatter only radiation scattered to the image plane need be calculated, (2) if a change has been made to one surface that was the main source of power reaching the image plane, the radiation path can be run separately at a considerable savings.

Subroutine ARRANGER and ARANGE

Takes the data from tape 2 or 3 and arranges them in the array as required.

Subroutine WHERE

Shows where the power is distributed on all the objects considered.

Subroutine SHOWIT

If any power reaches the image plane, the array of power from that source is printed out. This tells how each parcel of power reached the image plane and how much it was.

Subroutine SURFACE

From the vane information and the angle into the source and angle out of the source an apparent reflectance coefficient is calculated for every section of every collector specified for this source.

Subroutines FFACTOR and GFACTOR

Calculates the power transferred within the vane structure itself to determine the radiant exitance of the surfaces of the vanes.

III. INPUT DATA REQUIRED

A. Programs One and Two Inputs

The data deck for program two is compatible with the data input deck for program one. More information is required for the second program, but this information is not read by the first program. Unless otherwise noted, any variables mentioned apply to both programs.

CARD 1 FORMAT (5X,F15.8,R7,3X,R7)

YZERO The reference point in the image plane. It is at this height in the image plane that the images are seen. Only program one reads these data.

PRINTS If this word reads PRINTIT, an extended printout is given. It has been used mainly to debug the program as it prints out most of the arrays whenever any major change has been made. f it reads any other word, the short form of output is printed.

PLOTSIT If this word is PLOT, program two will plot the layouts of each space.

CARD 2 FORMAT (10AB)

HID This prints a header or identification line.

CARD 3 FORMAT (8X,R8)

VERSION If VERSION is read as relative or eight blanks, the program will assume that the distances that will be read into column four of ALAYOUT will be the distance to the next object plane. If version reads the word ABSOLUTE, all such distances are from a common plane.

CARD 4 FORMAT (F3.0,F12.6,3E12.6,F7.4,F5.0)

ALAYOUT Column 1
A code to the program. A zero means that this is not a refracting nor reflecting element, but the y-$\bar{y}$ heights are known. A -1. is a refracting or reflecting surface and the y-$\bar{y}$ values are known. A +1. means that this is the position of some object and the program should calculate the y-$\bar{y}$ data.

Column 2
The y value at this plane in this space, if it is known.

Column 3
The $\bar{y}$ value, if it is known.

Column 4
The distance to the next object plane. For any given space the objects can appear in any order, or, if VERSION is absolute, the distance from a common plane.

Column 5
The radius of the object in this plane.

Column 6
The index of refraction of this space.

Column 7
Each object is tagged with an object number. A cone is specified by two positions along the optical axis and two radii. Hence there are two lines in ALAYOUT with the same object number.

Column 8
The program needs to know the direction of the surface normal. For cones a -1. is the inward normal and a +1. is the outer surface. For a disk the positive z direction is a +1. Read only by program two.

Column 9
The number of sections that this surface will be subdivided into. Read only by program two.

CARD 5 FORMAT (10X,F15.7,5X,F15.7,5X,F5.0)

APERTUR Reads the maximum radius of the aperture and the minimum radius (the radius of the hole in the primary) and the line position in ALAYOUT. If this information is not included, the program will determine the maximum radius as the sum of y plus $\bar{y}$. Zero will be the minimum value.

CARD 5 FORMAT (7X,F13.10,7X,F13.10,7X,F13.10,7X,F13.7)

YREF Value of the marginal ray at a reference plane in this space. This plane becomes the zero distance. This and remaining information read only by program two.

YBARREF The $\bar{y}$ value in the reference plane in this space.

NDEX The index of this space.

SCALE The override scale factor of the plot. If blank, all objects will be scaled such that even the largest object will fit. Sometimes this is not desirable as other detail becomes too small.

B. Program Three Inputs

This program calculates the geometrical transfer factors.

CARD 1 FORMAT (5X,F5.1,5X,F5.1,5X,R7,3X,R3)

DELTA The number of subdivisions of each section. Default is 3 x 3.

SECJ The number of rib sections involved in the symmetry. Default value is five, but can be any odd number. It should be greater than one, and it would be unwise to be very large. If the value is other than five, program five needs to be altered slightly.

PRINT Print should normally be left blank, If the word is PRINTIT then for each factor stored there is a printout of about 30 words. It is only for the curious or for checking source coordinates and imaged collector positions.

NEWITAB If any punch is made in these columns the program will read off the mass storage file, the array ITAB. If a file is being partially redone, the final ITAB values are printed out and can be checked.

CARD 2 FORMAT (R6,4X,F3.0,3X,F3.0,4X,F3.0,3X,F3.0,3X,F2.0, 4X,F3.0,4X,F3.0,4X,F3.0,2X,F3.0,2X,F3.0,7X,F3.0,I2)

SOURCE C If the word SOURCE is not punched in these columns, the program will treat the data as an obstruction. If the word is SOURCE, the card contains the source collector combination and any other pertinent information.

**********If the word is not SOURCE, the following**********
meanings apply to the data card

OS1 If the shadow is a hole or disk, OS1 refers to the row in ALAYOUT that is the obstruction. If the shadow is a cone OS1 is the object number in ALAYOUT of the cone.

OS2 If OS2 is greater than zero, the shadow is caused by a disk. If OS2 is less than zero, the device causing the shadow is a cone. If OS2 equals zero, the shadow is caused by an aperture (hole).

**********If the word is SOURCE, the following************
meanings apply to the data cards

OBJS OS1 If the object has been defined in program two, then this is the source object, and the following word need not be filled in.

OBS OS2 If the desired object was not numbered in program two, but there are lines from which the desired object can be made, the two lines of interest go in OBJS and OBS. In particular a disk may be made by specifying a single line twice. If the object number is from program two, then fill in only OBJS.

OBJC
OC1 This is like OBJS but is the collector number.

OBC
OC2 This is like OBJSS but is the second surface of the collector.

NOTE: If both the object and collector numbers are known, none of the following information is necessary.

ISS=XSS=
HOLD(5) Maximum sections into which this newly defined object SOURCE is divided.

IMAX=XMAX
HOLD(6) Maximum sections into which this newly defined collector object is divided.

SNORMAL
HOLD(7) The normal of the newly defined source

DNORMAL
HOLD (8) The normal of the newly defined collector.

NS=SN
HOLD(9) The number this newly defined SOURCE is called.

NC=CN The number this newly defined COLLECTOR is called.

BLOCKIN
BLOCK
HOLD(11) If this word is left blank, the program assumes that this information is to be stored into block one and the next source will go into block two unless its block is specified. This feature allows for the extension of files or replacement of blocks of information.

C. Program Four Inputs

Program SPOTS

Card 1
FORMAT(5I5)

ICOL — The random access mass storage BLOCK into which the data will be stored.

MS — The total number of radial sections around the source. (With the present version of the program, it must be 5.)

NS — The number of axial sections on the source. It can have 1. to 10. sections, but they must correspond to the number of sections it had in program three.

MLOAD — Like MS, it must be 5. It represents the maximum number of radial sections on the fictitious object that is putting power onto the so-called "source" object.

NLOAD — The number of axial sections on the object that load the "source object."

The first set of power transfers calculated by program three should be to any object that will be considered as a possible source. Any power transfer will do. If one is clever he can make these be some of the required transfers in some cases. Program five needs to fetch an array for how the power was loaded onto the source and an ANGIN array. Program four writes the power array. Program three sets up the ANGIN array. If this is a dummy transfer choose the object with the least number of axial sections (in the same space as the source) as the MLOAD-NLOAD object to load the source.

Card 2
FORMAT(2I5,E20.10)

IR — The radial section of the source to be loaded.

IZ — The axial section of the source to be loaded.

X — The amount of power to be loaded on section (IR,IZ) of the source.

As long as power is being loaded onto the same "source" object, more cards may be read in. To terminate the loading of this object, the program will check for an end-of-file card. Having found one it will then check for another object to load as an additional source. If none is found the program will terminate normally. Any number of objects may be considered as sources. ICOL must correspond to the power transfer ICOL of program three.

D. Program Five Inputs

This program is mostly bookkeeping of many thousands of words in several large arrays.

CARD 1 FORMAT (I5,F5.0,2I5,F10.8,I5)

LEVEL The level of scatter from which the program is starting. The program can be run through level one and stopped, if all looks well, and the files have been preserved; second-order scatter can be considered by starting at that level. The program can presently go up to fourth-order scatter. An order of scatter may in fact be two scatters. One "scatter" means the radiation came into a section and left the section after any number of internal scatters.

ANG The ANG of radiation on the initial surface. For diffuse scatter 90° is a starting value. It will not accept a zero value.

L The program will start with level LEVEL scatter and do up to and including level L scatter.

IM The object that will be treated as the special collector and all increments of power onto this object will be printed out. Usually this should be the image plane.

RAD The radius of the edge of the vanes in the same units as the data in the previous programs.

IS The number of the object that has been sliced: like the cutaway portion of a cone used as a sun shield.

CARD 2 FORMAT (16I5)

INKEY The number of objects being considered.

CARD 3 FORMAT (16I5)

KEY(I) An integer array with INKEY number of words in it. Each word is the number of sections in object numbered I. I is the Ith word in KEY.

CARD 4 FORMAT (16I5)

ITAB2 The number designation of the collectors corresponding in order to the number designation of the sources in IEMIT.

CARD 5 FORMAT (16I5)

IEMIT The sources as emitters to the corresponding collectors in ITAB2.

CARD 6 FORMAT (16I5)

LIMITS(I) An integer array informing the program to use as sources at level I the sources in IEMIT up to the value LIMITS(I).

Card 9 SURFACE	CARD 7 FORMAT (2F20.10,4F10.5) Contains the basic average characteristics of each section of the system. Word 1--the vane angle. Word 2--the vane separation. Word 3--the vane depth. Word 4--the total diffuse reflectance coefficient of the coating. Word 5--the section length. This gives information about the number of edges involved. Word 6--the radius of the section involved. Do not confuse it with any edge radius. It is the radius of the cone, or whatever, at the location of this section.
Card 10 LOOK4	CARD 8 FORMAT (5I5) A two-dimensional array; the level of scatter being considered is read across (columns), then at this level of scatter the objects whose numbers fall in that column are considered scatterers at this level.
Card 11 SLICE	CARD 9 FORMAT (5F15.10) The percent of area of each section of object IS that physically exists. It slices away sections or portions of sections when there is, for example, a 45° cut in a cylindrical or cone-shaped baffle.

IV. OUTPUT

Although many months can be spent on the design of a baffle system and considerable foresight can be put into its design to utilize the proper principles of baffling, there are few data that will predict its performance *quantitatively*, unless a quantitative analysis program has been run. There are several approaches that can and have been used to indicate *how* the system performs. However, one would like to know *why* the system performed as it did. And, ideally, all this would be done relatively inexpensively. APART does this by using five programs as levels of improvement. These programs are compatible and can be run as a single program, but that is not desireable from a cost standpoint. It may seem a formidible task to learn five programs, however, this is an illusion. It is really one big program with five parts in series instead of overlayed. By running them separately, one can make improvements without having made costly computer runs that will no longer be of any use as the system is changed.

Briefly, the five levels of information are

(1) Program one--What are the areas that are seen from the image plane, and at what angle are they seen in real space?

(2) Program two--What are the principal paths of power to the critical areas that are seen from the image plane?

(3) Program three--What is the power transfer relationships and the angles that the power leaves a source section and enters a collector section?

(4) Program four--What is the loaded power distribution?

(5) Program five--How is the scattered radiation distributed throughout the system for the given conditions of power loading from the sources? What are the parcels of power from each contributing section that reach the image plane?

A. Program One Output

Because there is very little knowledge about the performance of the system to be analyzed, this program needs to establish those areas of the system that assume a special importance. These "critical" areas are those areas that are seen from the image plane. Besides knowing the exact locations of these critical areas, it is helpful to know the angle at which these objects (or imaged objects) are seen. This will assist in determining if a different vane structure design would be beneficial or whether the radiation to these areas must be minimized so as to reduce their contribution. Changes in the system can now be made without even knowing *how much* power is being transferred because one will develop a "sense" that it is going to be too much.

A general overview of the different arrays of output will help keep things at a simple level. Later the discussion will be about the interrelation of these arrays. For the purpose of this discussion, certain definitions are created that may not exist in the program but that will help to separate the overall output into modules of information.

Term — Definition

TITLE PAGE — The first page of information, it may contain some or all of the following:

(1) The title or identification of the run.
(2) The reference point in the image plane.
(3) The value of the Lagrange invariant.
(4) The array SLOPE, containing the slope of the line in the y-$\bar{y}$ diagram along which each object is positioned.

ALAYOUT — The printout of the array ALAYOUT during its various stages of development. The printout may appear under the heading of REAL SPACE or RESECTIONED REAL SPACE. This array contains the majority of the input from which the program works. It contains the location and size of the baffles and the y-$\bar{y}$ information at various points. See the input for program one for a more detailed discussion.

SPACE — The array SPACE numbers the spaces in which each object is located.

PRIME — The array PRIME contains the imaged objects as seen from the image plane space. It appears in the output under the headings of PRIMED SPACE IN RECTANGULAR COORDINATES or RESECTIONED PRIMED SPACE.

POLAR The array POLAR is the array PRIME put into polar coordinates *relative to the reference point* in the image plane. It will appear under the headings PRIMED SPACE IN POLAR COORDINATES or THE POLAR SPACE AS SEEN FROM THE DESIGNATED POINT IN THE IMAGE PLANE.

PPRIME The majority of this page is the array PPRIME, which is that set of imaged objects seen from the reference point in the image plane, to the exclusion of all other objects. The heading of this page is THE NUMBER OF OBJECTS SEEN FROM THIS IMAGE POINT IS I, where I is the number of objects in PPRIME.

RECAP A list of all the objects in numerical order, indicating whether an object is seen or not. If it is seen, the coordinates of the two ends of the imaged object are printed.

REAL SEEN The objects that are seen are printed in the order that they are seen. For each object seen, certain real space information is printed. In PPRIME and RECAP the information was about the imaged space. REAL SEEN yields information such as what portion of the real object is seen from the image plane and at what angle is it seen in real space.

ELLIPSE The last array of program output. It is the information about the projection of all the objects onto the exit pupil plane.

There are two versions possible for output. For simplicity they will be called the long and short versions. If the word PRINT reads PRINTIT on the input card, the long version will be printed. With any other word, the short form will be printed. The long version includes all of the output of the short version, so the long version will be described.

TITLE PAGE

(1) The program will first print the title or identification of the system as it is input into the array HID.

(2) The slope of the line in the y-$\bar{y}$ diagram that *follows* the object plane I (row I) in the array ALAYOUT.

Object plane I	Slope I to I+1
1	0.00000
2	0.00000
3	-32.70133
4	-32.70133
5	-1.40676
6	-1.40676

(3) In the case above, the slope of the line between points one and two is 0.00000, whereas for point three the slope of the line *after* it is -32.70133 and *before* it it is 0.00000. Point three is a break point, so at this point there is some refracting or reflecting element.

(4) The reference point used is printed as follows:

THE REFERENCE IMAGE POINT IS (0.0, 90.00).

The value of YZERO read in was 90.00, which designated the point in the image plane from which the system is looking out. Because the program presently assumes rotational symmetry, the x coordinate will always be 0.0.

(5) The Lagrange invariant is next printed.

CHI = 10.42500000.

ALAYOUT is printed with all previously unknown y-$\bar{y}$ information included. Because the output is in Fortran F format, and usually the object plane is a large distance from the first surface, the first line of ALAYOUT may have values too large to be printed.

SPACE is the next array to be printed.

REF SUR	SPACE
1	1
2	1
3	2
4	2
5	3
6	3

Each row in ALAYOUT is considered as an object plane and is in a specified space. If the object is along the first line in the y-$\bar{y}$ diagram, it is considered to be in space one, along the second line it is in space two, etc. Two other variables printed on this page are:

YEPRIME is the y' value (the height of the marginal ray) at the exit pupil.

IMGLNES is the slope of the last or image line in the y-$\bar{y}$ diagram.

The array PRIME is printed giving the distance that the imaged object planes in ALAYOUT appear to be from the image plane, the height of the images, and the y-$\bar{y}$ of the imaged planes. The first column is the row number, identical to the corresponding object (in real space) in ALAYOUT. The last column is the object number. At this stage there may be negative object numbers, which were not there before. The program has checked and found that the images of these objects pass through infinity. The negative object number codes the program to split this object into two objects; one object will retain the original object number and the other will be 100 larger.

PRIMED SPACE IN POLAR COORDINATES is the next printout (see p. 29). The information in the array PRIME, which is rectangular coordinates, is now put into polar coordinates relative to the point (0.,YZERO,ZIMAGE). YZERO is the reference point in the image plane, and ZIMAGE is the z coordinate of the image plane. Column one is the reference row in the array POLAR. The reference row in the array PRIME corresponding to this row is in column four. It is important that the two are not interchanged. Whereas any row of object information in POLAR can be *relocated* into the order that the objects are seen, the arrays ALAYOUT and PRIME *are enlarged only* to make rows available for split objects. Column two is the distance between the reference point in the image plane and the edge of the imaged object. Column three is the angle from the plus z axis to the edge of the imaged object. Column four is the reference row in PRIME (or ALAYOUT) for the referenced object number. Column five is the object number. Column six is the status of this object.

(1) A zero means that this object number will no longer be considered. This may be because from the start it was only a reference plane or because it has been processed and for one reason or another it

is not seen from the image plane.

(2) There is no status number one.

(3) A two means that the object was seen up to this edge of the object. As explained previously, there are two rows of information for each object. The status number two will appear in the row where the object number appears for the second time in the array POLAR. The status for the object the first time it appears will be a three or four.

(4) A three means that the object is seen starting from this edge of the object *and* object space is seen between the start of this object and the end of the previous object. By checking the last edge seen one will notice that its angle is not the same as the start of this new object. For example, in a Cassegrainian system with an inner conical baffle in the hole of the primary, the surface beyond the edge of the primary is seen in reflection by the secondary (this surface may be the tube wall or the annulus surface between the primary and the tube wall). The next surface seen in reflection is the primary (which both scatters and reflects the object field). Then the inner conical baffle is seen in reflection. When the start of this object is seen, its status will be three because of the intervening object space.

(5) A four means that the object is seen starting with this point. The angle in column three should be the same as the angle at the end of the previous object after all the objects are processed. Keep in mind that there is no previous object for the first status number four and the entire column is initialized to the value of four to start with, and only by processing are the numbers changed to threes, twos, or zeros.

The row order that the objects are *finally* assigned has only minor significance. Objects whose status becomes zero are passed over by the program from that point on and will no longer be rearranged. Hence, they may appear anywhere in POLAR. For objects that are seen, the row with status four or three must appear before the end of the same object, which will have a status two. It must also appear before the start of the next object seen. POLAR is ordered by increasing angle (column three). Because in most cases the angle that ends one object is the same angle that starts the next object (because the first object shadowed the second object) the row that ends the first object may appear before *or* after the start of the second object.

Primed space in polar coordinates

SUR NUM	RADIUS	ANGLE	REF NUM	OBJ NUM	STATUS
1	0.0000	0.000	0.	0.	0.0
2	90.0055	-1.570	65.	33.	0.0
3	2390.5952	2.511	62.	34.	4.0
4	10.0005	1.561	64.	33.	0.0
5	1944.6777	3.041	60.	26.	4.0
6	1940.3274	3.041	63.	34.	2.0
7	5413.4261	3.100	58.	26.	2.0
8	7571.4417	3.107	29.	14.	3.0
9	7213.1160	3.107	42.	19.	0.0
10	7213.1260	3.107	43.	19.	0.0
11	7602.3658	3.108	30.	14.	2.0
12	0.0000	0.000	0.	0.	0.0

Up to this point in the program, information has been filled in, imaged, and transformed, but not processed in the sense that it has started to determine which objects are seen. Most of the information discussed so far will appear in both the long and short versions of output. It is essentially the starting conditions for the calculations. The user should check this information against other sources to verify the accuracy of the input. For instance, one may check the value of the Lagrange invariant computed by this program with that from the optical system design program. The value and position of the exit pupil are another example of an easy check.

In the short version of the output the final values of the arrays ALAYOUT, PRIME, and POLAR are printed after all (not each) steps of the process are completed. In the long version, which we will now follow through, the arrays are printed after *each* step.

The objects whose image passes through infinity have been flagged with negative object numbers. The program sorts for these objects and resections them into two objects, one whose image approaches minus infinity, and the other whose image approaches plus infinity. One of the two sections will retain the original object number, whereas other sections will be given an object number 100 larger. Presently, the arrays can hold 100 rows of information, so all object numbers greater than 100 are recognized as split objects. After an object that passes through infinity is split, it may again be split later by the subroutine BAFSEEN. In this case, an object number in the hundreds is incremented by 100 again. Subroutine BAFSEEN determines which baffles are seen and which are obscured. It is possible that the image of an object will be seen and then have its image shielded behind the image of some other object and then appear again at the termination of the shielding object. In this case, the object will be split again. BAFSEEN will be discussed in greater detail later.

When an object is split because its image passes through infinity, the program checks for three maximum values of the objects with positive object numbers. These values are the maximum size, the maximum z coordinate, and the minimum z coordinate of any object or image. The program will attempt to image the split objects from one plane (end point of the object) toward pluš or minus infinity, whichever is appropriate. If the image is going in the plus z direction, it will use the maximum z value as the other "end" of the imaged object. The program will check for the case where this would give an image of zero length. This would happen when the original end of the object happens to have the maximum z coordinate. If this is the case, the maximum z value will be increased, and this value will be used as the "end" of the imaged object. The imaged object height is determined for this new end point. If it is larger than the maximum height of all the other objects, and the maximum height value is not that of its end point, the program will use the maximum height to determine the z coordinate of the imaged object.

The approach above does not throw out any lines, provides enough information to plot the layout, but yet does not print out values that are unmeaningfully large. A word of caution is in order because the image of the object may be seen beyond this arbitrary cutoff point. Usually, awareness of this fact is sufficient, but one has the option to split the object on just each side of the plane that is imaged at infinity (say a hundredth of a millimeter), and then the program will not put a large gap about that plane.

After the new end points are determined, two new rows of information are added to the arrays ALAYOUT, PRIME, and POLAR. In ALAYOUT the information is for the real space positions of the two new end points, in PRIME the image space data are entered, and in POLAR the image space information is entered in polar coordinates. After this is completed, the three arrays are printed out so they can be checked for accuracy.

The program then calls the subroutine APTOBST, which stands for aperture obstruction. There are two ways that an imaging aperture can act as an obstruction. One example is that of a simple lens inserted halfway down a tube. As one looks into the tube, one sees the tube wall as an object in real space until the lens. Then the tube wall is seen as imaged by the lens. From a given end of the tube the lens will image only that space beyond it. From the other end of the tube the lens will image the other half of the tube. In the y-$\bar{y}$ diagram it will do both,so the program checks for the appropriate space to reimage. The second way that the aperture acts as an obstruction is more straightforward. If the aperture is a mirror, then objects in its real space that go behind it are obscured. You cannot see objects through the mirror, only reflections in the mirror. Real space objects must appear outside the aperture obstruction, whereas imaged objects must appear inside the aperture. The program takes into account such things as the image of the primary that is perforated. Objects, in this case the image plane, will be seen through the hole in the primary even though it is beyond the plane of the imaging device and inside the outside clear aperture.

The program starts with the real image space and finds the first imaging device. If this happens to be a mirror, it checks for objects in real image space that appear behind it. These objects, or portions thereof, will be obscured, and the array POLAR will be changed to indicate the new end points seen or that the object is totally obscured (such an object will be given a status of zero). As each object is checked for each space, the object number in ALAYOUT (column 8 in the printout) is changed to its negative value to indicate that the check has been completed. Only the first row in which the object number appears has its value changed. A similar flag had been set for objects whose image passed through infinity. That flag has been cleared by now and this will not cause any problems for the program.

As each space is completed, the three arrays ALAYOUT, PRIME, and POLAR, are printed and can be checked if desired. Under certain circumstances the arrays will be enlarged because a real space object can appear outside the maximum clear aperture and inside the minimum clear aperture. Under these circumstances the object will be split into two objects.

Just before the program clears this subroutine, the flags (negative object numbers) are cleared, and the three arrays are printed. This is an aid to determine if the correct information is stored in the arrays for the next subroutine CLAPT and to check for accuracy. Subroutine CLAPT (clear aperture) ensures that all images that are seen are seen through the appropriate series of imaging apertures. This subroutine does not check for clear apertures that do not image. This will be done in subroutine BAFSEEN. After this is completed the long version of the output will print the three arrays ALAYOUT, PRIME, and POLAR.

The program has now arranged the array POLAR so that the objects it sees are legitimate objects (or imaged objects). Starting with the object with the largest positive angle in POLAR it proceeds to determine which objects are seen and what portions are obscured. Again there are cases where an object may be seen, then obscured, and then seen again. As before, the necessary arrays are enlarged and in the long printout the new arrays are printed for checking. At the termination of this program the array SPACE will be printed in the long version. From this point on there is no difference between the long and short versions of the output.

The final form of the arrays ALAYOUT, PRIME, and POLAR are printed. One will notice that in the array POLAR the status of the objects has been changed from almost all four-four combinations (four for the first row in which the object number appears and also the second row), to four-two or zero-zero combinations. If the status of the object is now zero, it cannot be seen from the given reference point. For all other objects, the row in which its object number first appears must have a status of four or three. For these objects the row in which its object number next appears must have a status number of two. The difference between a status of three or four is that there are three signals that the object field (or some unaccounted space) intervenes between the end of the last object seen and the start of the new one. This would signal such cases as where the object field is seen directly from the image plane. It is *necessary* that the object field *be seen*, so the *user* needs to determine whether the baffles need redesign or not.

The program calls the subroutine OBJSEEN (objects seen), which will process the data in POLAR slightly and make the necessary changes in the array PPRIME (not the array PRIME). The meaningful results of the program are now printed, starting with

THE NUMBER OF OBJECTS SEEN FROM THIS IMAGE POINT IS N

where N is the number of objects seen. In reality, one may actually see fewer objects because the program may have split an object. It is easy for the user to determine if this is the case because there will be an object with an object number 100 larger than any object initially read in.

Then the RECAP is printed. A list of all the objects in numerical order indicates whether an object is seen or not. If the object is seen, the coordinates of the two ends of the imaged object are printed.

The data printed from PPRIME and in the RECAP are for image space. In the subroutine ANGSEEN, which the program calls next, the conjugate image points of the imaged objects are found in real space. The y-$\bar{y}$ coordinates and the object's real space coordinates (of the section that is seen) are printed, along with the angle from which the object is seen in real space. This angle will normally be different than the angle made with the imaged object, but it is the angle at which the radiation is scattered to reach the next collector. The long version also prints the direction cosines of the real space object (OBJM, OBJN) and the direction cosines (CRITM, CRITN) of the ray connecting the reference point and the end point of the collector section.

```
FOR OBJ  N.
     OBJM     OBJN     CRITM     CRITN
     1.00     0.00     -.100     0.995
     1.00     0.00     -.250     0.968
     1.00     0.00     -.384     0.923
     1.00     0.00     -.497     0.868
     1.00     0.00     -.590     0.808

                       Distance to                        Angle of
       Y        Ȳ      next imaging sur   Baffle height   reflection
     80.43   193.02       1930.50            285.00          95.76
     80.43   193.02       1930.50            588.75         104.49
     80.43   193.02       1930.50            892.50         112.57
     80.43   193.02       1930.50           1196.25         119.81
     80.43   193.02       1930.50           1500.00         126.14
 FOR OBJ  M.

       etc.
```

In the example above, N and M would be object numbers. In this case object N is a vertical object. For all objects OBJL, the direction cosine from the x axis is zero because all calculations are done in the meridional plane. The y value (the marginal ray) in this plane in real space is 80.43, and the $\bar{y}$ (the chief ray) value is 193.02. The distance to the next imaging surface (if this is the real image space, the distance to the image plane) is given as 1930.50 units, the units being whatever was input. The baffle in real space varies from 285. to 1500. units. The program has taken the coordinates of the two end points and split the length of the object into four sections. To describe the four sections, it needs five sets of coordinates, hence the five rows of printout at this stage. The last column of data is the ANGLE OF REFLECTION at which this point on the object is seen in real space.

By knowing which sections are seen, at what angle, and some basic concepts of baffling, one can determine some of the more blatant flaws in the design of the baffle system. An ideal design would keep stray radiation off all the sections that are seen from the image plane. Because this is impossible in a practical case, the design should be such that as many scatters as possible are made before power reaches these areas. If there is any vane structure, the design should also attempt to keep the power off the critical surfaces (those seen by the image plane). Sometimes, better coatings or redesigning the vane structure on these critical areas will give the necessary improvement in performance. If this is not the case, one will have to determine the path of the radiation to the critical area and attempt to remove or reduce the amount of power transferred to it. Sometimes this may be obvious, but if it is not, then the third program will be the next level to use. It will give the percent of power transferred from section to section and angular information (see the discussion of program three for more detailed information).

The program has considered only objects whose image lies above the optical axis. It now returns to do the calculation for the lower half. This was done for computer convenience, as all that needs to be done is to establish the original ALAYOUT array and to change the YZERO value to -YZERO. The sequence of printouts repeats itself with the new objects that are seen in the lower half.

The final array to be printed contains the necessary information to plot the projection of all the apertures onto the exit pupil. These plots help visualize the series of apertures that are seen from the image plane. The format of the printout is as follows:

```
REF   CENTER   RADIUS   XMAX   YMAX   YMIN   I
```

REF is the row in ELLINFO in which the information for this aperture is stored. CENTER is the point in the exit pupil to which the center of the aperture is projected. Remember that everything has been done in the meridional plane so the value given is the y value, the x coordinate being zero. RADIUS is the radius of the aperture at its projection in the exit pupil. The next three values are easily calculated from columns two and three and are printed for convenience. XMAX is the maximum x coordinate that this aperture will appear to have when projected onto the exit pupil. It obviously must be the value of the radius. YMAX is the maximum height of the projected aperture and is equal to CENTER+RADIUS. YMIN is the minimum height of the projected aperture in the exit pupil and is equal to CENTER-RADIUS. I is the row in ALAYOUT for which the information for this aperture was taken.

This completes the types of output that the user can presently expect from program one. It may be desirable to further limit the output of the short version for most users because the pertinent information in many cases is contained in the last four types of information printed. At the present time, the arrays ALAYOUT, PRIME, and POLAR are helpful in confirming that the data were input correctly and in the explanation of "why" certain objects are seen or not. Another aid in checking the accuracy of the input data is program two's array that is printed out as CHANGEA. (See the explanation of the data output for program two.) The data decks for programs one and two are compatible and are relatively inexpensive to run, so running both programs to check the data input is sensible.

B. Program Two Output

The purpose of program two is to establish the principal radiation paths *to* the critical areas, i.e., those areas that are seen from the image plane as annotated in program one's output. Program one has already established the radiation path from these critical areas to the image plane, be it by reflection, refraction, or directly. There are two important aids in program two to help determine these radiation paths, the plots of all the objects as seen from each space, and the file BASICA, which this program creates, prints out, and catalogs for use by program three. Most of the data that are printed have the format of some program one output. In fact, some of the information is identical. Program two does more than program one in the sense that it not only looks back from the image plane, but it also looks into the system from object space and all intermediate spaces. By looking into the optical system one can establish most of the radiation paths that propagate the radiation through the system. This program is not the one that will determine *how much* power is transferred, but it does eliminate the necessity for many calculations by the third program, which is the most time consuming. Unfortunately, program two cannot do as much as program one because it does not determine which sections see other sections as program one does. One could write such a program, but in most cases it would not be worth the time to run it. This program needs to establish the radiation paths to the image plane, which can, in most cases, be done by looking at the plotted output of program two. In the cases where one would not be sure, it can be left to program three to determine. It may turn out that no power is transferred, in which case one can ignore this set of arrays.

A general overview of the various arrays of output will help keep things in perspective. As in program one, there are two forms of output, a long one and a short one. The long one is a debugging aid, and the short one contains all the necessary information for understanding what the program has done and for determining the radiation paths. Anyone familiar with program one output will easily recognize most of the output formats in program two. The modules of information are

Term | Definition

TITLE PAGE — The first page of printout. The short version does not contain the items that have an asterisk.

(1) The title or identification of the run.

*(2) The array SLOPE, containing the slope of the line in the y-$\bar{y}$ diagram along which each object is positioned.

(3) The value of the Lagrange invariant.

ALAYOUT — The printout of the array ALAYOUT during its various stages of development. The printout will appear under the heading REAL space, CHANGEA, or without the heading. The differences will be explained later. This array contains the majority of information that the program works with, including the location and size of the baffles and the y-$\bar{y}$ information for the various points in real space.

SPACE — The array SPACE numbers the space each object is in.

PRIME — The array PRIME contains the imaged objects as seen from the space presently being considered. Unlike program one, this program images all the objects for *each* of the spaces. The array will always be printed under the heading POSITIVE PRIMED SPACE IN RECTANGULAR COORDINATES. Under this heading the program prints out which line was used in the y-$\bar{y}$ diagram to image the objects. It also indicates the space from which the program is looking. Both are the same, *line one meaning space one*, etc. But for those less familiar with the y-$\bar{y}$ diagram the duplication in different words is helpful.

CHANGEA — Although this page of output has already been mentioned as one of the various forms of ALAYOUT, it is of such importance that it will also be considered as a separate module. This array is puton file BASICA for later use by program three.

TITLE PAGE

(1) The program will first print the title or identification of the system as input into the array HID.

(2) The slope of the line in the y-$\bar{y}$ diagram that follows the object plane I (row I) in the array ALAYOUT.

Object plane I	Slope I to I+1
1	0.00000
2	0.00000
3	-32.70133
4	-32.70133
5	-1.40676
6	-1.40676

(3) In the case above, the slope of the line between points one and two is 0.00000, whereas for point three the slope of the line after it is -32.70133 and before it it is 0.00000. Point three is a break point, so at this point there is some refracting or reflecting element. This array, SLOPE, is printed in the long version only.

(4) The Lagrange invariant is next printed.

CHI = 10.42500000

ALAYOUT

When the array ALAYOUT was read from the data cards, certain blocks of information may have been missing. In particular the y-$\bar{y}$ values probably will not be known for all the planes of the objects. Before the program starts imaging the objects, it proceeds to fill in the missing data. The full array is then printed. The power of using the y-$\bar{y}$ diagram is stored in this array.

The array will appear several times for different reasons. These reasons will be discussed in the sequence in which they appear. It should also be noted that the output is in Fortran F format, and usually the object plane is a large distance from the first surface. Therefore, the first line of ALAYOUT may have values that overflow the allotted space.

SPACE is the next array printed in the long version.

REF SUR	SPACE
1	1
2	1
3	2
4	2
5	3
6	3

Each row in ALAYOUT is considered as an object plane and is in a specified space. If the object is along the first line in the y-$\bar{y}$ diagram, it is considered to be in space one, along the second line, it is in space two, etc.

After the array SPACE, the long version may have the array ALAYOUT printed twice before the image space (the array PRIME) is printed. The program has taken the object line, line one in the y-$\bar{y}$ diagram, and has checked to see if any imaged objects pass through infinity. If so, the program will print ALAYOUT as the subroutine was entered (the array will not have the words REAL SPACE printed at the top of the page), and again print ALAYOUT as it exits the subroutine (REAL SPACE will head the page). The user can then check that the object was split correctly and that the correct y-$\bar{y}$ values were assigned.

If no object needs to be split or if the short version is requested, the ALAYOUT arrays mentioned above will not be printed and the array PRIME will be. This page of information has the heading:

```
     POSITIVE PRIMED SPACE IN RECTANGULAR COORDINATES
     ALL OBJECTS ARE IMAGED ALONG THE Y-YBAR LINE NUMBER  1.
     OBJECTS AND IMAGES APPEAR AS THEY WOULD BE SEEN FROM SPACE  1.
SUR   Y PRIME   YBAR PRIME   Z FROM REF PLANE   IMAGE BAFFLE HT   OBJ NUM
```

The array PRIME is the array ALAYOUT imaged along the line mentioned in row two of the output for this page. This will be done for all the (imaging) lines in the y-$\bar{y}$ diagram, or, in other words, for all spaces. The information for the plot is taken from these arrays.

The first column (SUR) is really the reference row in PRIME (also ALAYOUT) that stores the information about this image object (or real object). Y PRIME is the y' value of the *imaged* object, while YBAR PRIME is its $\bar{y}'$ value of the *imaged* object. The program has read in the reference y-$\bar{y}$ values of the plane that will be used as the origin for the calculation of distances Z FROM REF PLANE. IMAGE BAFFLE HT is the imaged baffle height, the height at which the object appears to be when looking from the space noted in line two. OBJ NUM identifies the object number that was imaged. The plotted output will be discussed later, but it is worthwhile to point out that it is this array that the plot routine attempts to plot. Because the scale of the plot can be selected, the possibility exists that the entire image of the object will be outside the bounds of the plotting paper and hence cannot be plotted. If part of the image is on the paper, that portion will be plotted, and a message in the printout will inform the user. It is for the above two reasons that sometimes not all the objects in the array PRIME are plotted.

If a plot has been requested, the following message will be printed beneath the array PRIME:

```
IF ANY INFORMATION FOLLOWS IT PERTAINS TO THE PLOT AS SEEN FROM SPACE 1
```

If all the images have dimensions that will plot on the paper, there will not be any further messages. Examples of all the possible message formats are

```
THE MIN HGT FOR OBJ num EXCEEDS THE PLOTTING PAPER SIZE FOR THE DESIRED SCALE
Z1 AND Z2 FOR OBJECT NUMBER num IS LESS THAN -60.  INCHES FOR THIS SCALE
Z1 AND Z2 FOR OBJECT NUMBER num IS GREATER THAN +60. INCHES FOR THIS SCALE
ONLY A PORTION OF OBJECT num WAS PLOTTED
```

The first three messages shou.d be self-explanatory. If the end points of the object have a y value (height) greater than 5.5 in., it is entirely off the paper. The plotting paper that has normally been used is a scroll 11 in. wide. With the optical axis along the center line of the paper, this allows scaled objects to be 5.5 in. high and many feet long. Sixty in. was chosen as the maximum distance from the origin (in either direction) that any object would be plotted, for *each* space. For a Cassegrainian type system there are three spaces, so the plot could be 30 ft long. The 60 in. can be changed if one finds that desirable.

Experience has shown that in most cases plots do not reach the maximum length because the images tend to go off the paper in the vertical direction and therefore are self-limiting.

The fourth possible message (ONLY A PORTION OF OBJECT num WAS PLOTTED) can lead to some confusion because of the present logic of the program. If only one of the two end points of the imaged object has a y value of less than 5.5 in. the program will print the fourth message implying that a portion of the object will always be plotted. The program will calculate the coordinates where the plot of the image would go off the plotting paper in the vertical direction. Next it checks the z coordinates for that portion to be plotted. It can happen that both these points are beyond 60 in. from the origin. Then message two or three will be printed indicating that the image was not even partially plotted. Because the sequence is first message four and then message two or three for the given object, it should not cause too much confusion.

After all the plotting is done, the program prints ALAYOUT corresponding to the array PRIME that was printed and plotted. The Ith row in PRIME is the image of the Ith row in ALAYOUT for this space. For each space the program will print the image space (array PRIME) and the real space (the array ALAYOUT). One may question why the real space would be changing. From each space, different objects have their images pass through infinity, so the array ALAYOUT will have different objects split.

After each space has been plotted, the program reestablishes the original ALAYOUT array. The objects that had been split are merged to give the original object. In the long version this array will be printed. The program then proceeds to the next space until all spaces have been plotted. It will start by checking for images that pass through infinity and proceed as described above.

After all the spaces have been considered, the program will make two important changes to the array ALAYOUT. Up until now the program has taken the data from column four to be the distance from the plane of this object to the plane of the object as referenced in the next row of

ALAYOUT. The program now changes this to the distance from the reference plane *in the last space considered.* For example, if row I indicated that the distance to the object in row I+1 was 7000 mm, but the object itself (the object in row I) was -13,000 mm from the reference plane, the value in the column will be changed from 7000 to -13,000.

The other change interchanges rows with the same object number if the first row in which the object number appears has a greater value in column four than the value in column four of the row the second time the object number appears.

These two changes arrange ALAYOUT for use by program three. Program three will subdivide each object and keep track of all the images of each section and the real space section to which they correspond. Because a single reflection will reverse the ordering of the section numbers, the program orders the direction in which the sections appear in real space and keeps track from there.

Consider the case of a baffle divided into 10 subsections, for instance, the baffle around a primary mirror. The mirror is oriented such that the object field is to the left, which is in the minus z direction. If the baffle is 6000 mm long and extends from the plane of the vertex of the primary toward the object, the data will be stored in ALAYOUT with the plane that is at -6000 mm from the vertex appearing before its other end, which will have a zero z value. Section one will be the section from -6000 mm to -5400 mm, and section 10 will be from -600 mm to 0.0 mm. The image of this baffle will extend from the primary to some positive z value (assuming here that the image does not pass through infinity). Section 10 (prime) will start at the plane of the primary and go to the plane that is conjugate to the z = -600 mm plane. Section one (prime) of the image will be still further away. Hence, the sequence of numbers will be from one to 10 in real space, but will be from 10 to one in image space. Power transferred to the leftmost section of the image (section 10 prime) corresponds to power added to section 10 of the real object.

The last physical piece of output to discuss is the plot. It has been discussed several times already in connection with the printed output, and one might think that it is simple to understand, interpret, and use. After all, it is just the cross-sectional plot of an object or image. In fact, the lines are also numbered according to their object number in the plots. However, as easy as it is to talk about such plots, it can be difficult to interpret them from a multiple mirror system. Images can appear anywhere, to the right of the real space, to the left of it, or on top of it. Any given plot may have images in all those areas. Hence, it takes considerable understanding to interpret what the images are and, more importantly, which apertures these images must pass through to be seen.

If the user numbers all the objects in space one with values less than 20 (or some other number), and then starts using 21 to 40 for the second space (even though there were only 12 objects in space one), and 41 to 60 for space three, etc., it will aid in interpreting the plots. One will then easily recognize that the low numbered objects are from space one and therefore must pass through certain apertures. The program does not draw in a vertical line for the aperture, but the user is advised to do so with a colored pen. Take care when drawing the image of an aperture that you ink in only that portion that exists, i.e., the image of a hole in a primary does no imaging and can act as an obstruction.

For each space there will be a plot and at the bottom of the plot, near the z = 0.0 plane, the following message is printed on the plot:

OBJECTS AS SEEN FROM SPACE num

This identifies from which space we are looking. If it is space one, then we are seeing the images as they would appear as we looked into the system from object space.

As mentioned previously, the lines in the plots are numbered according to their object numbers and should therefore be easily identifiable. There is no provision in the program for not over-writing previously written numbers. Therefore, sometimes a line cannot be identified by the printed number. Several options can be exercised: first, try to determine visually what the image is; second, enlarge the scale factor and rerun the job hoping that at the larger scale the numbers will not superimpose themselves; and third, for a particular run, eliminate some of those objects that might be the ones overlapping.

Ideally, one would like the program to have a routine that keeps track of such things but at this stage the need is not justified.

Just priorto termination the program writes onto disk a large file of several arrays and two numbers. This file is cataloged as BASICA. It is used by programs three and five. Twenty blocks of storage space are allocated, of which the first eight are filled in by this program. The eight blocks are

Variables		Total number of words	Block number
CHI (the Lagrange invariant)		1.	1.
ALAYOUT(100,9)	ARRAY	900.	2.
OMEGA(10,2)	ARRAY	20.	3.
REFER(10,2)	ARRAY	20.	4.
APERTUR(10,3)	ARRAY	30.	5.
SPACENO		1.	6.
SPACE(100)	ARRAY	100.	7.
N		1.	8.

Several of these variables may be unfamiliar, so they will be explained. The array OMEGA contains the normalized u and $\bar{u}$ angles of each line in the y-$\bar{y}$ diagram. The program calculates these values and stores them for later use by program three. The limit of 10 lines in the y-$\bar{y}$ diagram has been chosen arbitrarily and can be changed. REFER is the array that stores the reference points, specified by the y and $\bar{y}$ values, for each space. These values were read in by program two and are being passed along. It has already been explained that care must be taken to make the planes of reference coincide with the reference plane of the last space because CHANGEA has referenced all object distances from it. APERTUR is the array that contains the maximum clear aperture size, the size of any hole in the aperture, and the row in ALAYOUT in which the information for this object can be found. SPACENO is the maximum mumber of spaces in the y-$\bar{y}$ diagram for the set of points considered. SPACE is the array that annotates the space in which each row of ALAYOUT is located. There are 100 rows allocated to ALAYOUT, and 100 for SPACE. N is the number of rows in ALAYOUT that have been initialized.

When the output of programs one and two is understood, little more can be learned without doing the power transfer calculations. The programs have established the critical areas that are seen from the image plane, calculated some of the preliminary angular information, and established most, if not all, of the paths of radiation from the source to the image plane. The array CHANGEA that is printed at the end of program two will be needed to punch the data for program three. Most of the radiation paths have been established, but program three will need to be told what objects it will have to pass through or around. The user will do this by looking at the plot from program two and reading the data CHANGEA. This is explained in the writeup on program three's input.

C. Program Three Output

After analyzing the output from programs one and two, the user should be well aware of the critical areas of the system and the principal paths of radiation. Up until now the programs have been restricted to calculations and relationships in the meridional plane. We have learned about as much as we can under these restrictions and must now face the three-dimensional problem that the system poses. We need relationships between the sections that will help determine how much power is being transferred and at what angles this power is leaving one section (the source) and the angle that it enters the other sections (the collectors). Program three makes these calculations and stores them in three files: BDRDF contains the "fraction" of power transferred from each section to all other sections; ANGOUT contains the "adjusted" angle that the radiation leaves the source section; and ANGIN contains the "adjusted" angle that the radiation enters the collector sections. These files are then cataloged for later use by program five.

The above paragraph alludes to "adjusted" angles and the "fraction" of power transferred, which will need considerable explanation to understand. If one checks the input for the first three programs he finds that nowhere is the surface structure described, the programs have not read in any reflectivities, nor has any power been put on any surface. Yet, this program is calculating the "fraction" of power transferred.

To understand what is meant it is necessary to discuss the information that has been passed to this program and the data that it reads in, and then discuss the three files separately in great detail.

There will be five blocks of data discussed:

(1) BASICA--the basic information that is passed on by program two and added to by this program.

(2) BDRDF--the percent of power transferred factors.

(3) ANGOUT--the "adjusted" angles out of the source.

(4) ANGIN--the "adjusted" angles into the source.

(5) ITAB--the number of words stored in each block of the files BDRDF, ANGOUT, and ANGIN.

When requested, by using the code word CHECKPO (check passed on information), all the information that has been stored in BASICA is printed out. It is usually desirable to request this printout, especially if one is running several systems with the files cataloged with different cycle numbers. It will establish which file the system used for its calculations. Almost all the information has already been scrutinized, and the user will learn little more from studying it. But it is handy to confirm that the correct data are being used and for making corrections to the run or making additions to the run. One need not go back to the program two data as BASICA contains all the necessary information. At present the data are printed without any headings, and therefore an understanding of the output from program two and the sequence in which the data are printed, as shown below, is required.

Variables		Total number of words	Block number
CHI (the Lagrange invariant)		1.	1.
ALAYOUT(100,9)	ARRAY	900.	2.
OMEGA(10,2)	ARRAY	20.	3.
REFER(10,2)	ARRAY	20.	4.
APERTUR(10,3)	ARRAY	30.	5.
SPACENO		1.	6.
SPACE(100)	ARRAY	100.	7.
N		1.	8.

CHI is the Lagrange invariant.

ALAYOUT is the array that contains the real space information about the system. This is the changed form of ALAYOUT called CHANGEA. From this array the user can check that the correct rows of data were used when calculating disk type obstructions or apertures.

The array OMEGA contains the normalized u and $\bar{u}$ angles of each line in the y-$\bar{y}$ diagram ($\Omega = nu/\mathcal{K}$, $\bar{\Omega} = n\bar{u}/\mathcal{K}$). These are of no great use until the user needs to check certain calculations, at which time they are very helpful.

REFER is the array that stores the y-$\bar{y}$ values of the reference plane for each space. Again their usefulness is when one is checking the data.

APERTUR is the array that contains the characteristics of each imaging aperture and the row number in ALAYOUT that contains the other data on this aperture.

SPACENO is the number of spaces in the system.

SPACE is the array that lists the space in which each row of data in ALAYOUT is located.

N is the number of rows in ALAYOUT that have been initialized. This number is not specifically printed out but is easily determined by checking the number of rows of ALAYOUT that are printed. Each row is numbered in the left-most column, so the largest row number value is the value of N.

* * *

BDRDF

The most significant data stored in this section of the printout are the percent of power transferred. There are three important aspects to learn about this type of number:

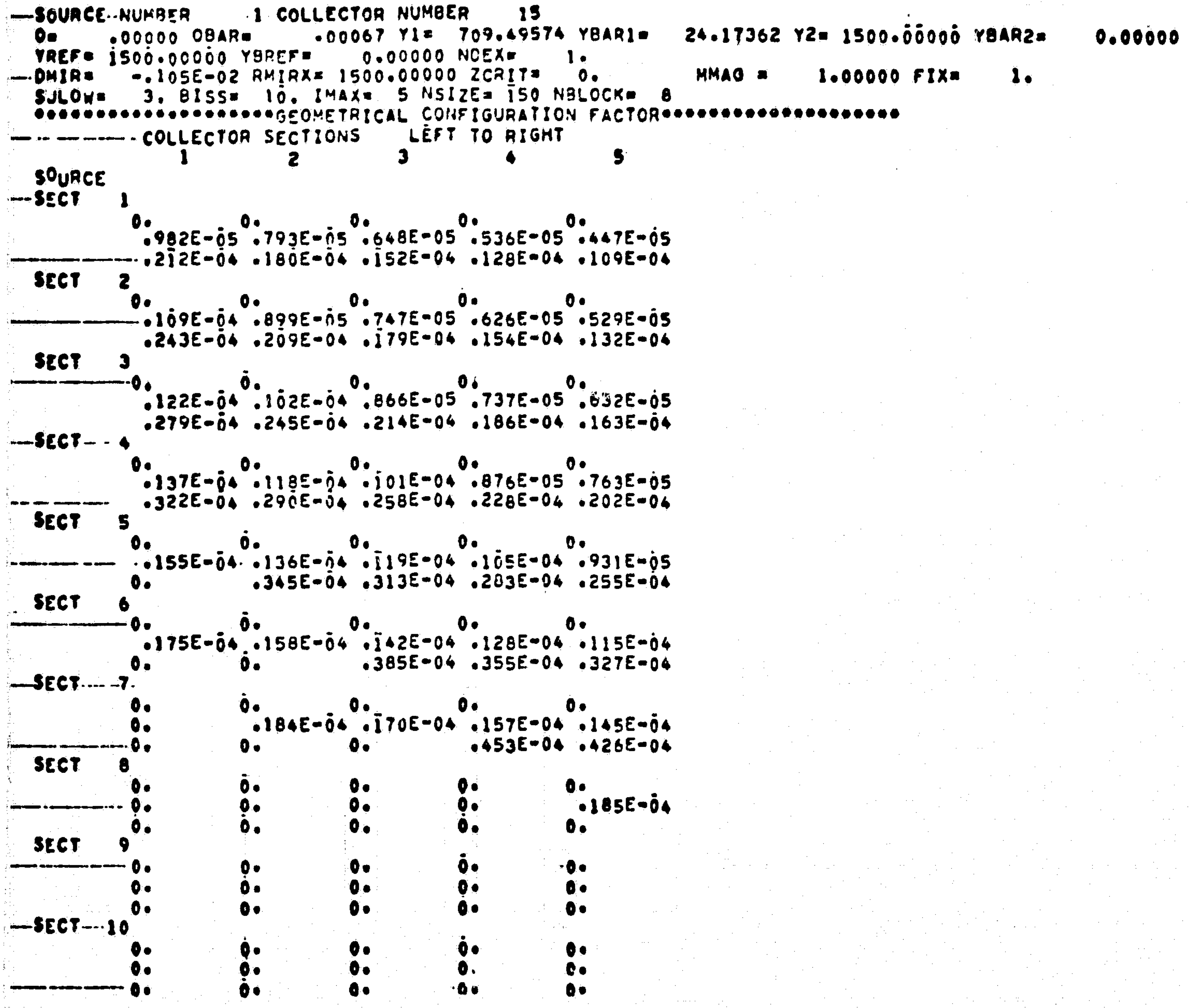
(1) Most important of all is, what does the number mean, what does it represent that one can understand?

(2) How are the data printed? The format on the page is not any trivial matter.

(3) How does the fifth program use this number?

To help explain these aspects it is beneficial to analyze some sample data output (see Fig. 2).

SOURCE NUMBER 1--This implies that the power calculations will be from all the sections (and possibly subsections) of the object that the user has numbered as number one. There are several ways that the source can be numbered; it can retain the number that it had in program two, it can be changed from what it was in program two, or it can be an object that did not exist before, and so it is just given the number. It is possible that the same number could mean two different objects, but that is not recommended because the user is more likely to make a mistake.



```
SOURCE NUMBER     1 COLLECTOR NUMBER    15
O=    .00000 OBAR=     .00067 Y1=  709.49574 YBAR1=   24.17362 Y2= 1500.00000 YBAR2=    0.00000
YREF= 1500.00000 YBREF=    0.00000 NOEX=    1.
DMIR=   -.105E-02 RMIRX= 1500.00000 ZCRIT=   0.        MMAG =    1.00000 FIX=    1.
SJLOW=   3. BISS=  10. IMAX=  5 NSIZE= 150 NBLOCK=  8
********************GEOMETRICAL CONFIGURATION FACTOR*******************
         COLLECTOR SECTIONS    LEFT TO RIGHT
             1         2         3         4         5
SOURCE
SECT   1
         0.        0.        0.        0.        0.
          .982E-05  .793E-05  .648E-05  .536E-05  .447E-05
          .212E-04  .180E-04  .152E-04  .128E-04  .109E-04
SECT   2
         0.        0.        0.        0.        0.
          .109E-04  .899E-05  .747E-05  .626E-05  .529E-05
          .243E-04  .209E-04  .179E-04  .154E-04  .132E-04
SECT   3
         0.        0.        0.        0.        0.
          .122E-04  .102E-04  .866E-05  .737E-05  .632E-05
          .279E-04  .245E-04  .214E-04  .186E-04  .163E-04
SECT   4
         0.        0.        0.        0.        0.
          .137E-04  .118E-04  .101E-04  .876E-05  .763E-05
          .322E-04  .290E-04  .258E-04  .228E-04  .202E-04
SECT   5
         0.        0.        0.        0.        0.
          .155E-04  .136E-04  .119E-04  .105E-04  .931E-05
         0.         .345E-04  .313E-04  .283E-04  .255E-04
SECT   6
         0.        0.        0.        0.        0.
          .175E-04  .158E-04  .142E-04  .128E-04  .115E-04
         0.        0.         .385E-04  .355E-04  .327E-04
SECT   7
         0.        0.        0.        0.        0.
         0.         .184E-04  .170E-04  .157E-04  .145E-04
         0.        0.        0.         .453E-04  .426E-04
SECT   8
         0.        0.        0.        0.        0.
         0.        0.        0.        0.         .185E-04
         0.        0.        0.        0.        0.
SECT   9
         0.        0.        0.        0.        0.
         0.        0.        0.        0.        0.
         0.        0.        0.        0.        0.
SECT  10
         0.        0.        0.        0.        0.
         0.        0.        0.        0.        0.
         0.        0.        0.        0.        0.
```

Fig. 2. Sample data output from BDRDF.

ORIGINAL PAGE IS
OF POOR QUALITY

COLLECTOR NUMBER 15--The power will be calculated from all the sections on source number 1 to all the sections on the collector number 15. If the sections are subdivided into "subsections," then the power transfer is calculated from all subsections to all subsections. As for the source number, the collector can also be numbered in three ways.

O--O is the omega value for the line along which the images are seen. Omega is the normalized u (marginal ray angle) value OMEGA = nu/$\mathcal{K}$. In the example shown, the images will be seen from the space in which object one is located, and therefore along the line that represents that real space.

OBAR--The omega bar value is the normalized $\bar{u}$ (chief ray angle) value equal to n$\bar{u}$/$\mathcal{K}$, for the line along which the images are projected in the y-$\bar{y}$ diagram.

Y1 YBAR1--The y-$\bar{y}$ values for the plane in real space that represents the end of the object (to be imaged) that has the lowest z coordinate. This point can now be imaged along the "imaging" line.

Y2 YBAR2--The y-$\bar{y}$ values for the plane in real space that represents the end of the object to be imaged that has the largest z coordinate.

YREF YBREF--The y-$\bar{y}$ values of the reference plane in the space of the source.

NDEX--The index of refraction of the space of the source.

DMIR--Looking from the source toward the collector, DMIR is the imaged distance of the last imaging aperture. The program automatically checks thatthe radiation passes through this imaged aperture.

RMIRX--The imaged radial size of the last imaging aperture between the source and the collector.

ZCRIT--The distance from the section on the source of the first imaging aperture (if one exists for this source-collector combination). The program will check that the radiation passes through this aperture.

MMAG--The magnification of the last imaging aperture (if imaging is done). If the radiation does pass through the imaged aperture, its position will be found on the imaged aperture. MMAG will then be used to determine its location on the same aperture in real space. Knowing this point and the collector point, the real space angular information can be calculated.

FIX--Whenever a surface is specified, the normal to the surface will determine on which side of the cone or disk is the object. If a disk is reflected by a mirror, the normal to the surface reverses direction. With two reflections it is back to the same direction as real space. FIX keeps track of the direction of the normal for disk-type surfaces.

SJLOW--There are 2*SJLOW-1. radial sections on the objects. Both the source and collector object will be divided in the same manner.

BISS--The number of axial sections into which the source is sectioned. If the source object is not a disk, the axial sections will be of equal *axial length*. If the source object is a disk, BISS is the number of *radius* sections. The radius sections are divided into BISS sections all of *equal area*. Because the area goes as R^2, the difference in radius sections goes as R^2.

IMAX--The number of *axial* sections on the collector. If the collector is a disk, then IMAX is the number of *radius* sections on the collector. For nondisk-type collectors the collector sections are of equal axial length. For disk-type collectors the radii of the IMAX sections go as $R_i{}^2$ so that each ith section has equal area.

NSIZE--The number of words in this block of transfer information. NSIZE = SJLOW*BISS*IMAX.

NBLOCK--The NSIZE words of information are stored into block NBLOCK of the following files:

Geometrical configuration factor	into	BORDF
Angle out of the source	into	ANGOUT
Angle into the collector	into	ANGIN

GEOMETRICAL CONFIGURATION FACTOR--The numbers in this array represent the fraction of incident power on the source section that would be transferred to the collector section if the source section had unit power and unit Lambertian reflectivity. If the source has a radiance of $L(\theta,\phi)$, incremental area dA_S, an angle θ_S between its normal and the vector from the source to the collector, and the collector being a distance r from the source, of incremental area dA_C, and at an angle θ_C from the vector from the collector to the source, then the incremental power transferred dP is

$$dP_c = L(\theta,\phi)*dA_S*\cos(\theta_S)*dA_C*\cos(\theta_c)/r^2. \tag{1}$$

If the source is Lambertian

$$L = M/\pi, \tag{2}$$

where M is the radiant exitance, the power per unit area of the source. The power on the source is

$$dP_S = E*dA_S = \rho*dP_s + \alpha*dP_s + t*dP_s, \tag{3}$$

where E is the irradiance (watts per square cm), ρ is the reflectivity, α is the absortion, and t is the transmittance. The scattered power is $\rho*dP_S$ and is equal to

$$\rho*dP_S = M*dA_S. \tag{4}$$

Substituting Eqs. (4) and (2) into (1) gives

$$dP_C = \rho*dP_s*\cos(\theta_s)*\cos(\theta_C)*dA_C/\pi r^2.$$

The geometrical configuration factor calculated in the program is

$$gcf = \cos(\theta_s)*\cos(\theta_c)*dA_c/\pi r^2,$$

therefore

$$dP_c = \rho*dP_s*gcf.$$

The bidirectional reflectance distribution function (BRDF) will be determined in program five as will the incremental powers dP_c to complete the equation for power transferred. Unless the physical shape of the collector or source is changed, the gcf need not be recalculated. The power loadings of the system and the coatings can be changed, and these factors are still valid. Because ρ is a function of the angle in θ_i,ϕ_i and the angle out θ_o,ϕ_o, it is desirable that ρ *is not* included at this stage.

If the coating were truly Lambertian for all input angles it could be incorporated in gcf. Also, the $\rho(\theta_i,\phi_i;\theta_o,\phi_o)$ for a section with vane structure is significantly a function of the input and output angles.

Program three prints out the gcf in the following format. The *axial* collector sections are listed from left to right (see Fig. 2). *Axial* source sections are listed in the left column. In the case shown there are three numbers listed for each source section-collector section combination. SJLOW is equal to three, which means that there are five radial sections on the objects. The program takes advantage of two symmetry conditions that reduce the number of calculations to 3/25. The source section is always considered to be the radial section in the six o'clock position, in this case section three. Figure 3a shows that the power transferred *from* section three *to* section one and five should be the same owing to the symmetry. The same applies to sections two and four. In Fig. 3b, the power transferred *from* radial section two *to* sections five and four should be the same factor as the power transferred from section three to sections one and five. All the sections have rotated one position. Obviously this rotation can be done five times, reducing the calculations 4/5. The first symmetry condition implies that the program need calculate only three of the power transfers from section three on the source to the five collector sections.

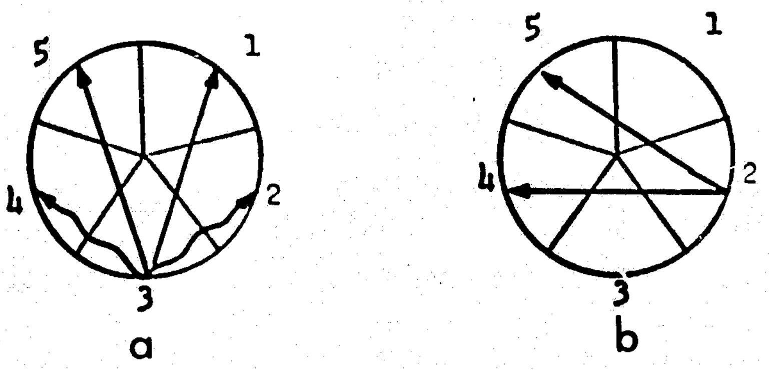


Fig. 3. Power transfer symmetries of rotationally symmetric objects.

One should note that this procedure, which limits the program to rotationally symmetric objects, reduces the computer run time. Asymmetrical objects *could* be handled in a straightforward manner, and plans are under way to modify the program for those cases.

In the example shown (Fig. 2), the gcf from source section (3,1) to collector section (1,1) or (5,1) is 0.; from source section (3,1) to collector section (2,1) or (4,1) it is .982E-05; from source section (3,1) to collector section (3,1) it is .212E-04.

Because of obstructions (apertures, disks, cones) between the source sections and the collector sections, certain transfer factors are reduced or go to zero. In the example shown, this occurs with source sections 8 through 10. Section 10 transfers no energy to any of the collector sections of object 15.

The NSIZE words in this array are stored into block NBLOCK of the random access file BDRDF (tape 5). Note that the gcf is neither the solid angle of the collector as seen from the source nor the percent of power transferred from the source to the collector. It has aspects of both. The calculation assumes that the reflectivity of the source is uniform as seen from any position of any section on the collector. For very specular coatings, this will not be valid, and the calculation will be erroneous.

ANGLE OUT OF THE SOURCE--For each source section-collector section combination the program will calculate two angles that specify the angle that the radiation left the source section. If the source sections are subdivided (DELTA $\neq$ 1.), the average values will be calculated. Figure 4 is an example of the output. The ANGLE OUT data are shown in Fig. 5. The angles are relative to the local section coordinate system; one for the source section and one for the collector section. For a cone, the Z' axis vector in the surface of the cone points either toward or away from the vertex of the cone, whichever direction is in the +z direction. The Y' axis is the outward normal of that surface. The X' axis is tangent to the cone and orthogonal to the Y'Z' plane. For a disk, the Z' axis is the outward radius vector, the Y' axis is the

```
                         ANGLE OUT. OF THE SOURCE
               COLLECTOR SECTIONS   LEFT TO RIGHT
                   1         2          3         4         5
  SOURCE
  SECT   1
                  0.00000   0.00000   C.00000   0.00000   0.00000
                174.98239174.29079173.68502173.15014172.67448
                176.23966175.51814174.88590174.32746173.83070
  SECT   2
                  0.00000   0.00000   0.00000   0.00000   0.00000
                174.79357174.04893173.39115172.80603172.28226
                176.09782175.32776174.64720174.04157173.49924
  SECT   3
                  0.00000   0.00000   0.00000   0.00000   0.00000
                174.59003173.78574173.06873172.42572171.84596
                175.94488175.12052174.38520173.72545173.13033
  SECT   4
                  0.00000   0.00000   0.00000   0.00000   0.00000
                174.36999173.49831172.71342172.00324171.35777
                175.77948174.89409174.09633173.37407172.71727
  SECT   5
                  0.00000   0.00000   0.00000   0.00000   0.00000
                174.13135173.18313172.31995171.53121170.80794
                  0.00000174.64569173.77624172.98120172.25168
  SECT   6
                  0.00000   0.00000   0.00000   0.00000   0.00000
                173.87166172.83599171.88185171.00044170.18412
                  0.00000   0.00000173.41962172.53909171.72293
  SECT   7
                  0.00000   0.00000   0.00000   0.00000   0.00000
                  0.00000172.45183171.39115170.39935169.47049
                  0.00000   0.00000   0.00000172.03793171.11735
  SECT   8
                  0.00000   0.00000   0.00000   0.00000   0.00000
                  0.00000   0.00000   0.00000   0.00000168.64642
                  0.00000   0.00000   0.00000   0.00000   0.00000
  SECT   9
                  0.00000   0.00000   0.00000   0.00000   0.00000
                  0.00000   0.00000   0.00000   0.00000   0.00000
                  0.00000   0.00000   0.00000   0.00000   0.00000
  SECT  10
                  0.00000   0.00000   0.00000   0.00000   0.00000
                  0.00000   0.00000   0.00000   0.00000   0.00000
                  0.00000   0.00000   0.00000   0.00000   0.00000
```

Fig. 4. Angle out of the source sections.

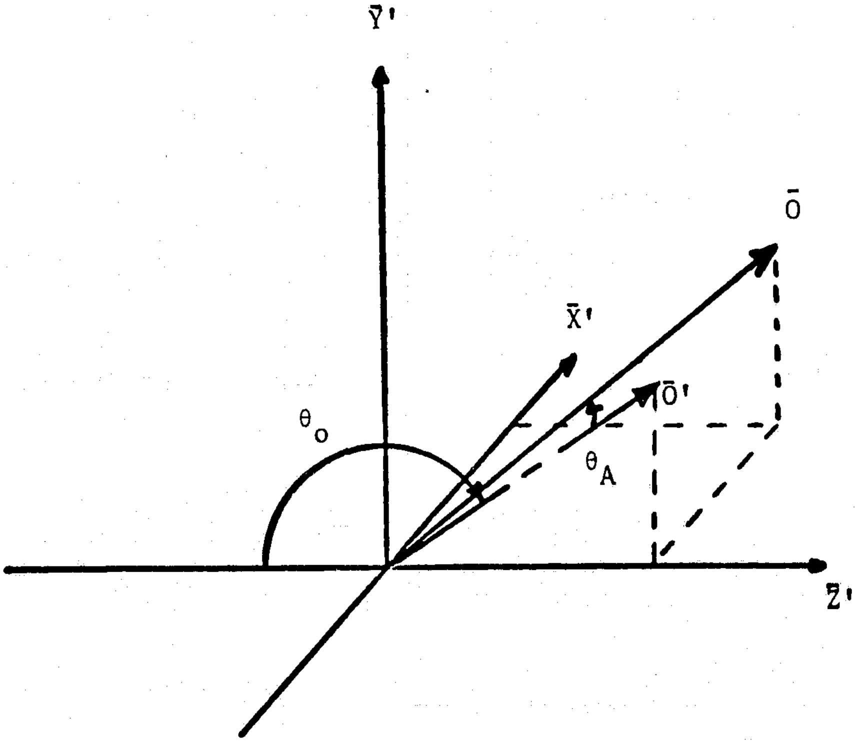


Fig. 5. The vector 0 is the vector from the source section to the collector section. 0' is its projection into the Y'Z' plane. 0_o is the angle from the -Z' vector to the 0' vector, both are in the Y'Z' plane. 0_A is the azimuthal angle and is the angle from 0_A to 0.

normal to the disk, and the X' axis is tangent to the radius and orthogonal to the Y'Z; plane (see Figs. 6 and 7).

The format of the rows and columns in which the two arrays are printed corresponds to their geometrical configuration factor (see the description of the geometrical configuration factor output for a further description).

ANGLE INTO THE COLLECTOR--Because there were two angles that described the angle out of the source, there are two angles that describe the angle into the collector for each source-collector section combination. Figure 8 is a sample output. Figure 5 also represents how these two angles are measured. The $\bar{O}$ vector points toward the source section. The local coordinated systems are the same as described for the angle out of the source.

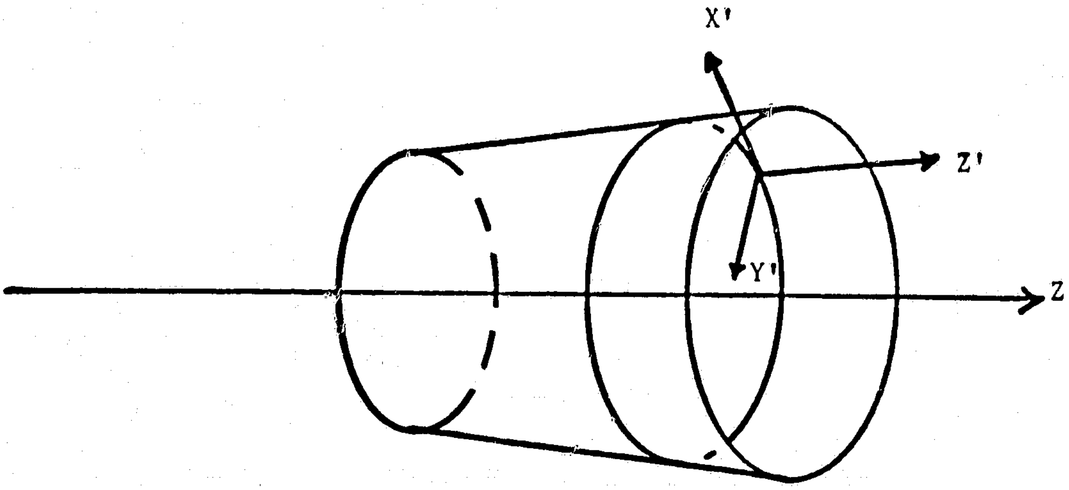


Fig. 6. Local coordinate system for a section on a cone. If the surface was the outside surface of the cone Y' and X' would point in opposite directions. Z' would remain the same.

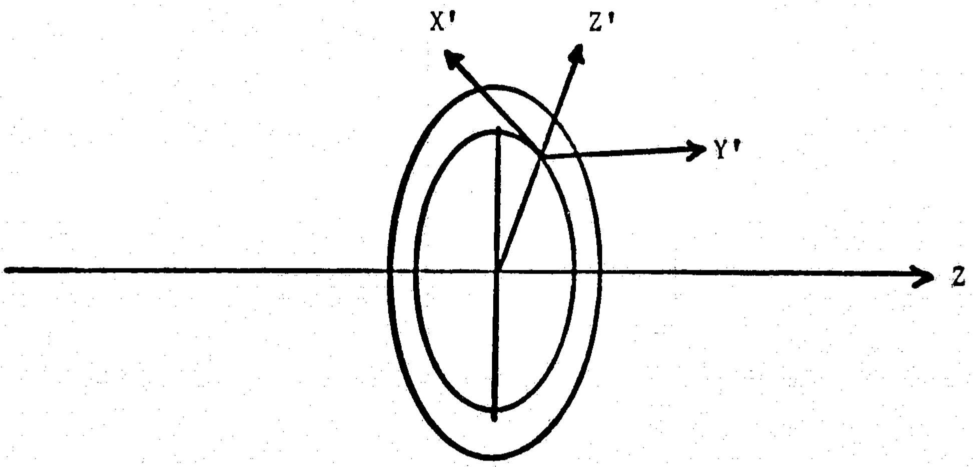
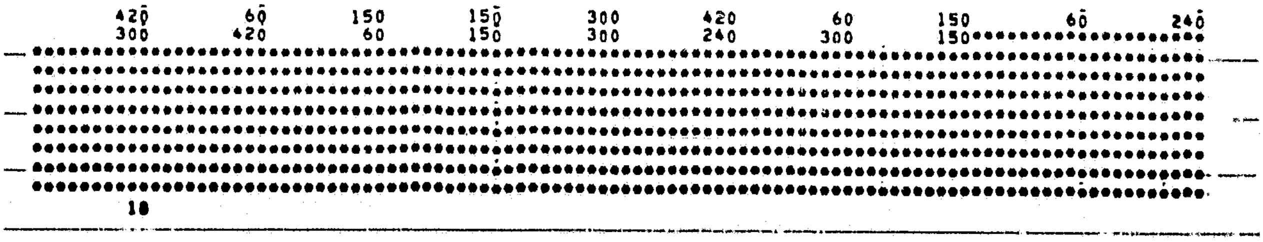


Fig. 7. Local coordinate system of a disk. The +z side of the disk is the source. If the other side is the source, Y' and X' would change directions. Z' would remain the same.

```
                     ANGLE INTO THE COLLECTOR
          COLLECTOR SECTIONS   LEFT TO RIGHT
              1         2         3         4         5
 SOURCE
 SECT   1
         0.00000   0.00000   0.00000   0.00000   0.00000
       176.10286176.35524176.57695176.77327176.94830
       168.92364169.62500170.24363170.79325171.28468
 SECT   2
         0.00000   0.00000   0.00000   0.00000   0.00000
       176.11008176.34513176.55342176.73928176.90614
       168.66518169.33398169.92902170.46176170.94140
 SECT   3
         0.00000   0.00000   0.00000   0.00000   0.00000
       176.11786176.33412176.52758176.70167176.85915
       168.38698169.01772169.58420170.09566170.55967
 SECT   4
         0.00000   0.00000   0.00000   0.00000   0.00000
       176.12628176.32208176.49906176.65981176.80646
       168.08669168.67283169.20465169.68929170.13268
 SECT   5
         0.00000   0.00000   0.00000   0.00000   0.00000
       176.13542176.30886176.46743176.61295176.74696
         0.00000168.29525168.78488169.23565169.65195
 SECT   6
         0.00000   0.00000   0.00000   0.00000   0.00000
       176.14537176.29429176.43214176.56012176.67924
         0.00000   0.00000168.31818168.72606169.10673
 SECT   7
         0.00000   0.00000   0.00000   0.00000   0.00000
         0.00000176.27813176.39253176.50012176.60146
         0.00000   0.00000   0.00000168.14957168.48321
 SECT   8
         0.00000   0.00000   0.00000   0.00000   0.00000
         0.00000   0.00000   0.00000   0.00000176.51125
         0.00000   0.00000   0.00000   0.00000   0.00000
 SECT   9
         0.00000   0.00000   0.00000   0.00000   0.00000
         0.00000   0.00000   0.00000   0.00000   0.00000
         0.00000   0.00000   0.00000   0.00000   0.00000
 SECT  10
         0.00000   0.00000   0.00000   0.00000   0.00000
         0.00000   0.00000   0.00000   0.00000   0.00000
         0.00000   0.00000   0.00000   0.00000   0.00000
```

Fig. 8. Angle into the collector sections.

ITAB--The last array printed is the ITAB array. An example is shown in Fig. 9. The array reads from left to right, the numbers being the number of words, ITAB(I), in the Ith block of stored data on files BDRDF, ANGOUT, and ANGIN. The number of blocks that now have significant data is printed last. In the case shown, 18 blocks of data were written, and the second block has 60 words.

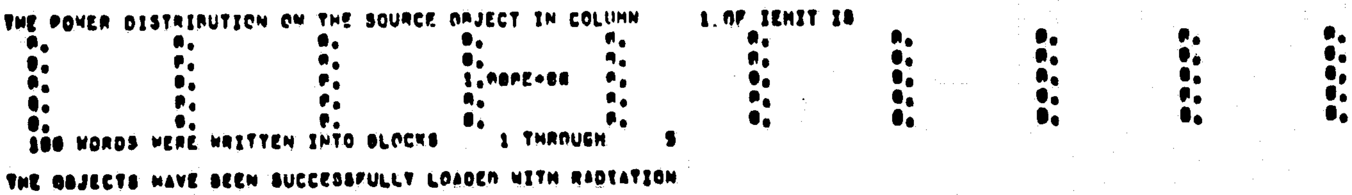


```
         420          60         150         150         300         420          60         150          60         240
         300         420          60         150         300         240         300         150●●●●●●●●●●●●●●●●●●●●●●●●
 ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
 ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
 ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
 ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
 ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
 ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
 ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
 ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
          18
```

Fig. 9. ITAB, the number of words stored on the random access mass storage files.

D. Program Four Output

Program four is the simplified loader program. An array such as shown in Fig. 10 is printed. It depicts the power distribution on this particular source object. Because there are, by default, five radial sections, the data are stored on disk in five blocks, one for each radial section. It is the disk file created by this program as output that primes program five for the stray radiation calculation.



```
THE POWER DISTRIBUTION ON THE SOURCE OBJECT IN COLUMN     1.OF IEMIT IS
 0.         0.         0.         0.         0.        0.        0.        0.        0.        0.
 0.         0.         0.         0.         0.        0.        0.        0.        0.        0.
 0.         0.         0.         1.0000E+00 0.        0.        0.        0.        0.        0.
 0.         0.         0.         0.         0.        0.        0.        0.        0.        0.
 0.         0.         0.         0.         0.        0.        0.        0.        0.        0.
 500 WORDS WERE WRITTEN INTO BLOCKS      1 THROUGH      5
THE OBJECTS HAVE BEEN SUCCESSFULLY LOADED WITH RADIATION
```

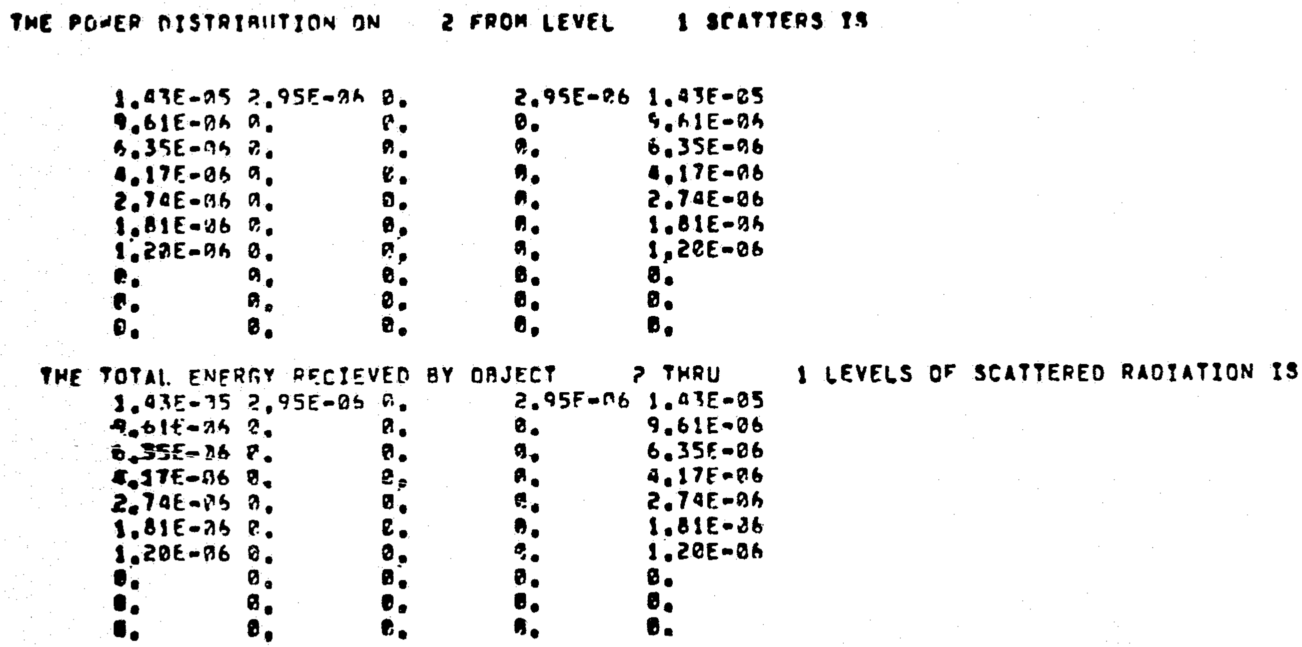
Fig. 10. Program four printout of the power on the sections of the objects loaded.

E. Program Five Output

The analysis of a system would not be complete until the power distribution in the image plane was determined. Program five completes the sequence of programs by doing this calculation. It has basically two parts, a bookkeeping routine and several surface reflectivity routines, the largest being the routine SURFACE that calculates the apparent reflectivity from baffles with vane structure. The program's basic role is to multiply the gcf from program three by the incremental unit of power and the apparent reflectivity of the sections in the system. This product is the power on the succeeding collector section. The calculation is continued until the power distribution on the image plane is determined.

A series of three types of arrays will be printed. One will give the power, added section by section, to all of the objects that were considered collectors at each level. The second array type will give the running total power, section by section, of all the objects input into the system. The third array type will print all the increments of power from *each section* of a source object that are added to *each section* of the "important" surface that is specified in the input deck. This surface is usually the image plane, but can be any other object. Examples of the three arrays are discussed in the following paragraphs.

POWER ADDED and TOTAL POWER arrays are so similar in form they will be discussed as a pair, which is also the way they always appear in the output. Figures 11, 12, and 13 are examples of the POWER ADDED and TOTAL POWER arrays for an object number two at three levels of scattered radiation. Figure 11 shows the power distribution on object number two from level one scatters. Regardless of the sources at this level of scatter, this is the calculated power distribution on object two, section by section. This power distribution is for the programmed radiation paths, which may not be all the significant paths. Object two has 10 axial sections and five radial sections. The first radial section of the first axial section [section (1,1)] has 1.43E-05 watts of power

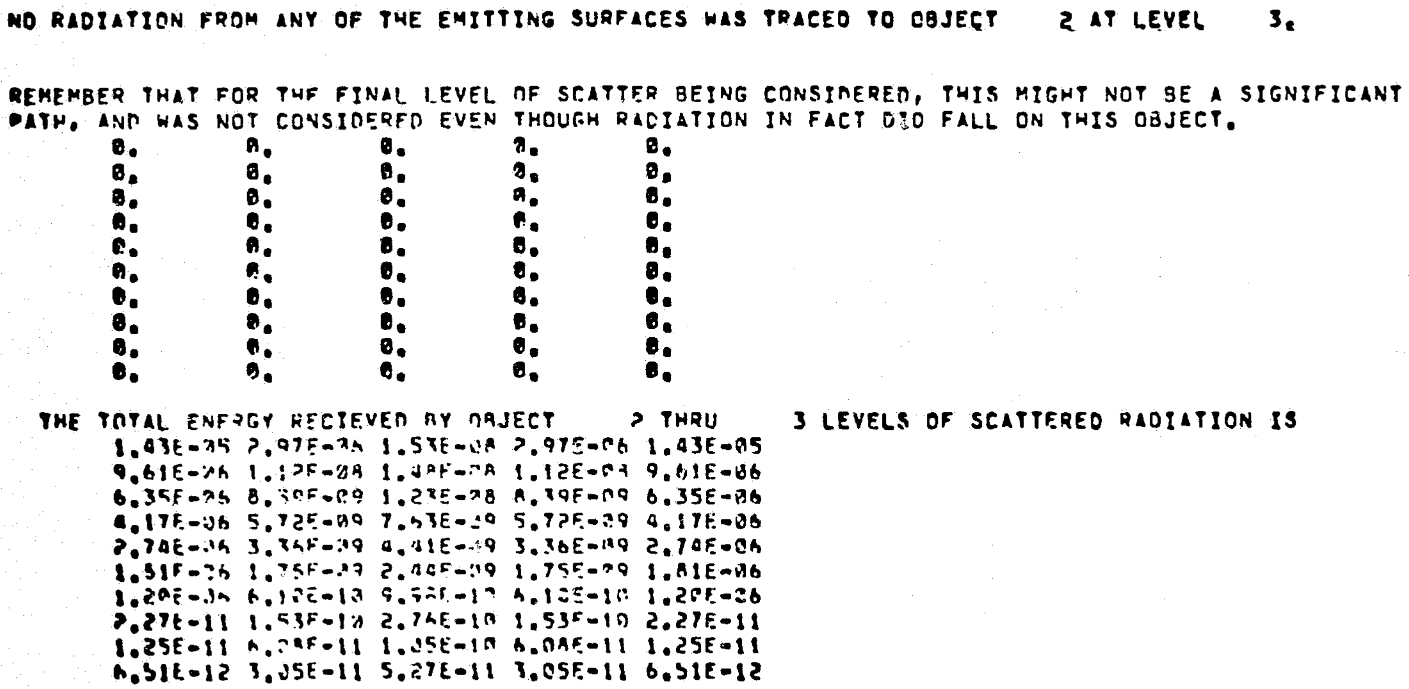


```
THE POWER DISTRIBUTION ON    2 FROM LEVEL    1 SCATTERS IS

        1.43E-05 2.95E-06 0.       2.95E-06 1.43E-05
        9.61E-06 0.       0.       0.       9.61E-06
        6.35E-06 0.       0.       0.       6.35E-06
        4.17E-06 0.       0.       0.       4.17E-06
        2.74E-06 0.       0.       0.       2.74E-06
        1.81E-06 0.       0.       0.       1.81E-06
        1.20E-06 0.       0.       0.       1.20E-06
        0.       0.       0.       0.       0.
        0.       0.       0.       0.       0.
        0.       0.       0.       0.       0.

  THE TOTAL ENERGY RECIEVED BY OBJECT     2 THRU     1 LEVELS OF SCATTERED RADIATION IS
        1.43E-05 2.95E-06 0.       2.95E-06 1.43E-05
        9.61E-06 0.       0.       0.       9.61E-06
        6.35E-06 0.       0.       0.       6.35E-06
        4.17E-06 0.       0.       0.       4.17E-06
        2.74E-06 0.       0.       0.       2.74E-06
        1.81E-06 0.       0.       0.       1.81E-06
        1.20E-06 0.       0.       0.       1.20E-06
        0.       0.       0.       0.       0.
        0.       0.       0.       0.       0.
        0.       0.       0.       0.       0.
```

Fig. 11. Power distribution on object two after one level of scatter.

```
THE POWER DISTRIBUTION ON    2 FROM LEVEL    2 SCATTERS IS

          7.19E-09 1.46E-08 1.53E-08 1.46E-08 7.19E-09
          4.53E-09 1.12E-08 1.48E-08 1.12E-08 4.53E-09
          3.33E-09 8.39E-09 1.23E-08 8.39E-09 3.33E-09
          2.42E-09 5.72E-09 7.63E-09 5.72E-09 2.42E-09
          1.32E-09 3.36E-09 4.41E-09 3.36E-09 1.32E-09
          5.72E-10 1.75E-09 2.44E-09 1.75E-09 5.72E-10
          1.29E-10 6.10E-10 9.98E-10 6.10E-10 1.29E-10
          2.27E-11 1.53E-10 2.76E-10 1.53E-10 2.27E-11
          1.25E-11 6.08E-11 1.05E-10 6.08E-11 1.25E-11
          6.51E-12 3.05E-11 5.27E-11 3.05E-11 6.51E-12

 THE TOTAL ENERGY RECIEVED BY OBJECT     2 THRU     2 LEVELS OF SCATTERED RADIATION IS
          1.43E-05 2.97E-06 1.53E-08 2.97E-06 1.43E-05
          9.61E-06 1.12E-08 1.48E-08 1.12E-08 9.61E-06
          6.35E-06 8.39E-09 1.23E-08 8.39E-09 6.35E-06
          4.17E-06 5.72E-09 7.63E-09 5.72E-09 4.17E-06
          2.74E-06 3.36E-09 4.41E-09 3.36E-09 2.74E-06
          1.81E-06 1.75E-09 2.44E-09 1.75E-09 1.81E-06
          1.20E-06 6.10E-10 9.98E-10 6.10E-10 1.20E-06
          2.27E-11 1.53E-10 2.76E-10 1.53E-10 2.27E-11
          1.25E-11 6.08E-11 1.05E-10 6.08E-11 1.25E-11
          6.51E-12 3.05E-11 5.27E-11 3.05E-11 6.51E-12
```

Fig. 12. Power distribution on object two after two levels of scatter.



```
NO RADIATION FROM ANY OF THE EMITTING SURFACES WAS TRACED TO OBJECT     2 AT LEVEL     3.

REMEMBER THAT FOR THE FINAL LEVEL OF SCATTER BEING CONSIDERED, THIS MIGHT NOT BE A SIGNIFICANT
PATH, AND WAS NOT CONSIDERED EVEN THOUGH RADIATION IN FACT DID FALL ON THIS OBJECT.
          0.         0.         0.         0.         0.
          0.         0.         0.         0.         0.
          0.         0.         0.         0.         0.
          0.         0.         0.         0.         0.
          0.         0.         0.         0.         0.
          0.         0.         0.         0.         0.
          0.         0.         0.         0.         0.
          0.         0.         0.         0.         0.
          0.         0.         0.         0.         0.
          0.         0.         0.         0.         0.

  THE TOTAL ENERGY RECIEVED BY OBJECT      2 THRU      3 LEVELS OF SCATTERED RADIATION IS
          1.43E-05 2.97E-06 1.53E-08 2.97E-06 1.43E-05
          9.61E-06 1.12E-08 1.48E-08 1.12E-08 9.61E-06
          6.35E-06 8.39E-09 1.23E-08 8.39E-09 6.35E-06
          4.17E-06 5.72E-09 7.63E-09 5.72E-09 4.17E-06
          2.74E-06 3.36E-09 4.41E-09 3.36E-09 2.74E-06
          1.81E-06 1.75E-09 2.40E-09 1.75E-09 1.81E-06
          1.20E-06 6.12E-10 9.58E-10 6.12E-10 1.20E-06
          2.27E-11 1.53E-10 2.74E-10 1.53E-10 2.27E-11
          1.25E-11 6.08E-11 1.05E-10 6.08E-11 1.25E-11
          6.51E-12 3.05E-11 5.27E-11 3.05E-11 6.51E-12
```

Fig. 13. Power distribution on object two after three levels of scatter.

incident on it. The second radial section (1,2) has 2.95E-06 watts of power, and so on for the remaining sections. The initial loading was symmetrical, so the power distribution on the object is also symmetrical. The program would handle an asymmetrically loaded system just as easily. The higher numbered sections, 8 to 10, receive no power because an entrance aperture prevented the radiation from reaching those sections.

The TOTAL POWER received by object two through one level of scattered radiation is the same as the POWER ADDED because there was no power previously scattered to this object.

Figure 12 shows the power added or the power distribution from level two scatters onto object two. Note that now all the sections of object two have power on them. The power added to axial sections one are two or three orders of magnitude less than the power loaded at level one. The TOTAL POWER on object two is the sum of this array with the TOTAL POWER array from level one scatter.

Figure 13 is third-level scatter and indicates that no power was transferred to object two *or* that the program was not asked to trace power to object two. Because no power was added, it is not unexpected that the TOTAL POWER on object two remains the same as for level two scatter. The run from which this example was taken put power onto the image plane at level three. Therefore, the requested power transfers at level three were only from the critical objects to the image plane. Other power transfers were not necessary. In fact, it would be a waste of computer time to do them.

By analyzing the above set of arrays, one can begin to quantitatively understand how the power is being propagated through the system and how the power distribution in the image plane has been determined. However, one does not know the increments of power from all the objects that put power onto the image plane. Which sections of which objects contribute the most power to the sections in the image plane? This information is clearly seen when the third type of array is printed. Every increment of power from every section of every object that transfers power to the image plane is printed. A quick study of these arrays will determine which sections are the major sources of power to the image plane.

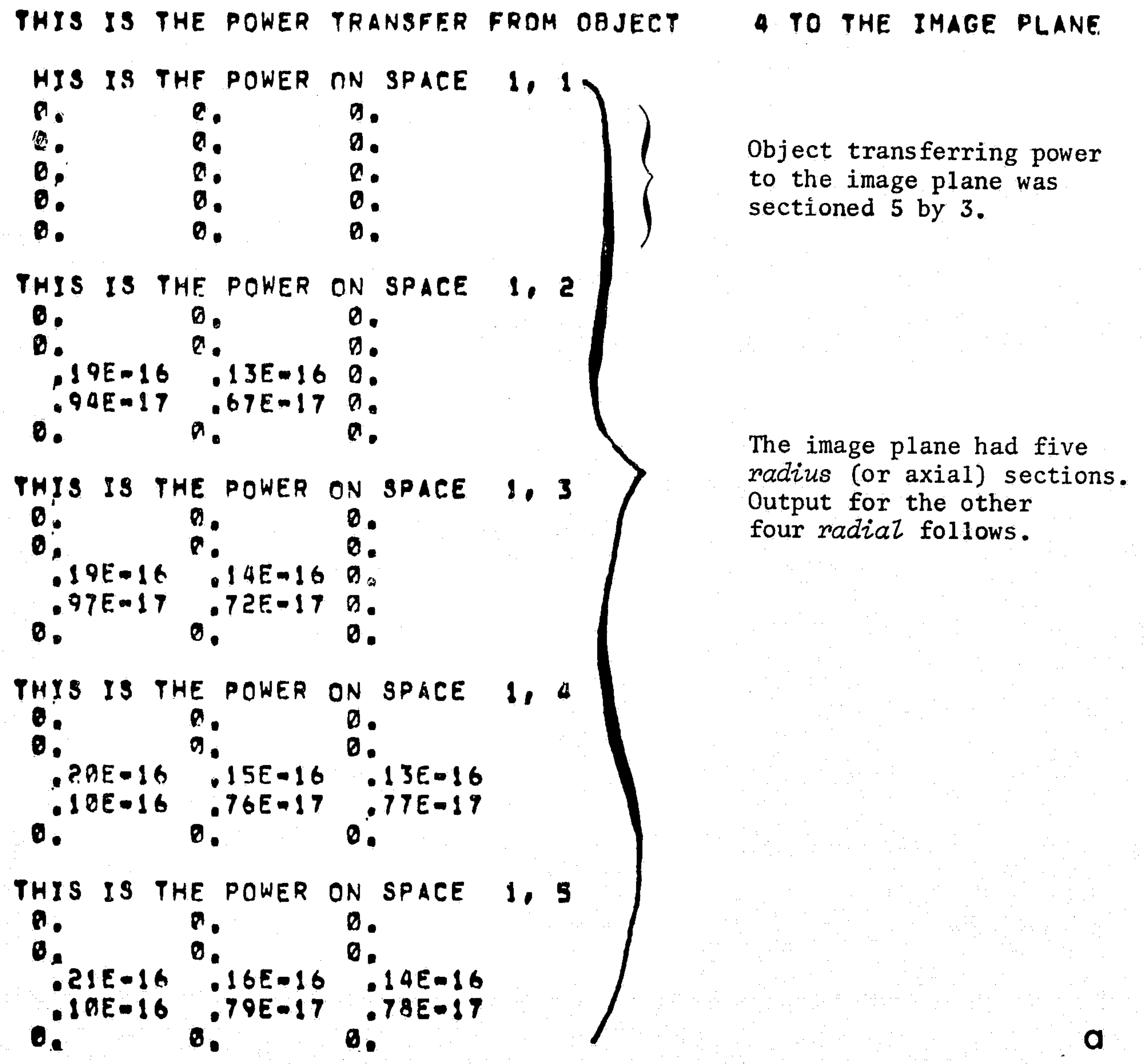
Figure 14 will serve as an example of the type of output that can be expected and how to interpret the data format. For the case shown, the image plane was a disk that had five radii and five radial sections. The source object for Fig. 14a-d was object four, whereas object eight was the source object for Fig. 14e-j.

Looking at Fig. 14a, there are five sets of arrays. They represent the increments of power that are added to five of the 25 sections on the collector. Figure 14b-e represents five more of the sections for a total of 25. The first array is preceded by the statement THIS IS THE POWER ON SPACE 1,1. The five by three array that follows represents the dimension (sections) *of the source object* number 4. Remember the image plane (the collector object) is five by five. The array indicates that no power was transferred to section (1,1) of the image.

The next array in the same figure represents the power added to the one image section (1,2); the first radial section and the second radii section. The third radial-first axial section of the source object contributes .19E-16 watts to the image collector section (1,2). The third radial-second axial section of the source object four contributes .13E-16 watts and so on. The zero power transfers occur for one of two reasons, no power was on that particular source section, or those source sections could not "see" the image plane section.

There are only five radial sections in this example, hence, only five pages of data. Each page contains all the units of power from all the sections on object four to all the *radius* sections of a given *radial* section on the collector.

Figure 14f-j shows the increments of power transferred from object eight to the image plane. The increment of power from section (1,2) to section (1,1) is .22E-16 wa‹ts. From the comparison of the units of power transferred from objects four and eight, one can conclude that object eight sections are contributing about 100 times the power of the sections from object four. Therefore, the system redesign should



```
THIS IS THE POWER TRANSFER FROM OBJECT    4 TO THE IMAGE PLANE

 HIS IS THE POWER ON SPACE  1, 1
 0.          0.          0.
 0.          0.          0.
 0.          0.          0.
 0.          0.          0.
 0.          0.          0.
THIS IS THE POWER ON SPACE  1, 2
 0.          0.          0.
 0.          0.          0.
  .19E-16    .13E-16   0.
  .94E-17    .67E-17   0.
 0.          0.          0.
THIS IS THE POWER ON SPACE  1, 3
 0.          0.          0.
 0.          0.          0.
  .19E-16    .14E-16   0.
  .97E-17    .72E-17   0.
 0.          0.          0.
THIS IS THE POWER ON SPACE  1, 4
 0.          0.          0.
 0.          0.          0.
  .20E-16    .15E-16    .13E-16
  .10E-16    .76E-17    .77E-17
 0.          0.          0.
THIS IS THE POWER ON SPACE  1, 5
 0.          0.          0.
 0.          0.          0.
  .21E-16    .16E-16    .14E-16
  .10E-16    .79E-17    .78E-17
 0.          0.          0.
```

Object transferring power to the image plane was sectioned 5 by 3.

The image plane had five *radius* (or axial) sections. Output for the other four *radial* follows.

a

Fig. 14. Incremental power transfer from object 4 to image plane.

```
THIS IS THE POWER TRANSFER FROM OBJECT    4 TO THE IMAGE PLANE

THIS IS THE POWER ON SPACE  2, 1
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.

THIS IS THE POWER ON SPACE  2, 2
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
  .94E-17   .67E-17 Ø.
  .14E-22   .50E-21 Ø.

THIS IS THE POWER ON SPACE  2, 3
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
  .97E-17   .72E-17 Ø.
  .15E-22   .51E-21 Ø.

THIS IS THE POWER ON SPACE  2, 4
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
  .10E-16   .76E-17   .77E-17
  .15E-22   .52E-21   .10E-17

THIS IS THE POWER ON SPACE  2, 5
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
  .10E-16   .79E-17   .78E-17
  .16E-22   .53E-21   .10E-17
```

b

Fig. 14. Continued.

```
THIS IS THE POWER TRANSFER FROM OBJECT     4 TO THE IMAGE PLANE

THIS IS THE POWER ON SPACE  3, 1
 0.         0.         0.
 0.         0.         0.
 0.         0.         0.
 0.         0.         0.
 0.         0.         0.

THIS IS THE POWER ON SPACE  3, 2
  .14E-22   .50E-21 0.
 0.         0.         0.
 0.         0.         0.
 0.         0.         0.
  .14E-22   .50E-21 0.

THIS IS THE POWER ON SPACE  3, 3
  .15E-22   .51E-21 0.
 0.         0.         0.
 0.         0.         0.
 0.         0.         0.
  .15E-22   .51E-21 0.

THIS IS THE POWER ON SPACE  3, 4
  .15E-22   .52E-21   .10E-17
 0.         0.         0.
 0.         0.         0.
 0.         0.         0.
  .15E-22   .52E-21   .10E-17

THIS IS THE POWER ON SPACE  3, 5
  .16E-22   .53E-21   .10E-17
 0.         0.         0.
 0.         0.         0.
 0.         0.         0.
  .16E-22   .53E-21   .10E-17
```

C

Fig. 14. Continued.

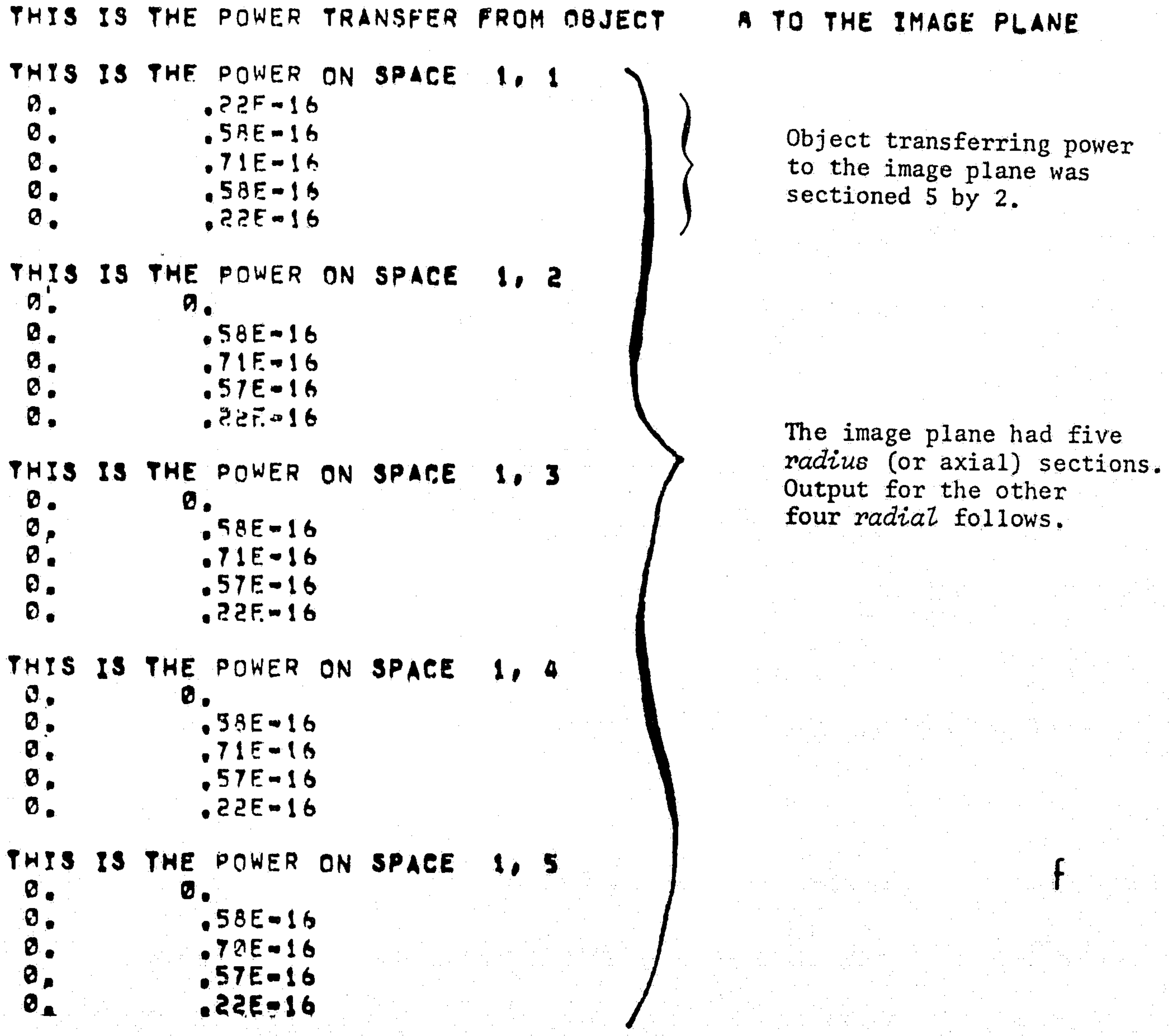
```
THIS IS THE POWER TRANSFER FROM OBJECT    4 TO THE IMAGE PLANE

THIS IS THE POWER ON SPACE  4, 1
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.

THIS IS THE POWER ON SPACE  4, 2
   .14E-22   .5ØE-21 Ø.
   .94E-17   .67E-17 Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.

THIS IS THE POWER ON SPACE  4, 3
   .15E-22   .51E-21 Ø.
   .97E-17   .72E-17 Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.

THIS IS THE POWER ON SPACE  4, 4
   .15E-22   .52E-21   .1ØE-17
   .1ØE-16   .76E-17   .77E-17
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.

THIS IS THE POWER ON SPACE  4, 5
   .16E-22   .53E-21   .1ØE-17
   .1ØE-16   .79E-17   .78E-17
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
 Ø.         Ø.         Ø.
```

d

Fig. 14. Continued.

```
THIS IS THE POWER TRANSFER FROM OBJECT    4 TO THE IMAGE PLANE

THIS IS THE POWER ON SPACE  5, 1
 0.        0.        0.
 0.        0.        0.
 0.        0.        0.
 0.        0.        0.
 0.        0.        0.

THIS IS THE POWER ON SPACE  5, 2
 0.        0.        0.
  .94E-17   .67E-17 0.
  .19E-16   .13E-16 0.
 0.        0.        0.
 0.        0.        0.

THIS IS THE POWER ON SPACE  5, 3
 0.        0.        0.
  .97E-17   .72E-17 0.
  .19E-16   .14E-16 0.
 0.        0.        0.
 0.        0.        0.

THIS IS THE POWER ON SPACE  5, 4
 0.        0.        0.
  .10E-16   .76E-17   .77E-17
  .20E-16   .15E-16   .13E-16
 0.        0.        0.
 0.        0.        0.

THIS IS THE POWER ON SPACE  5, 5
 0.        0.        0.
  .10E-16   .79E-17   .78E-17                         e
  .21E-16   .16E-16   .14E-16
 0.        0.        0.
 0.        0.        0.
   33
```

Fig. 14. Continued.

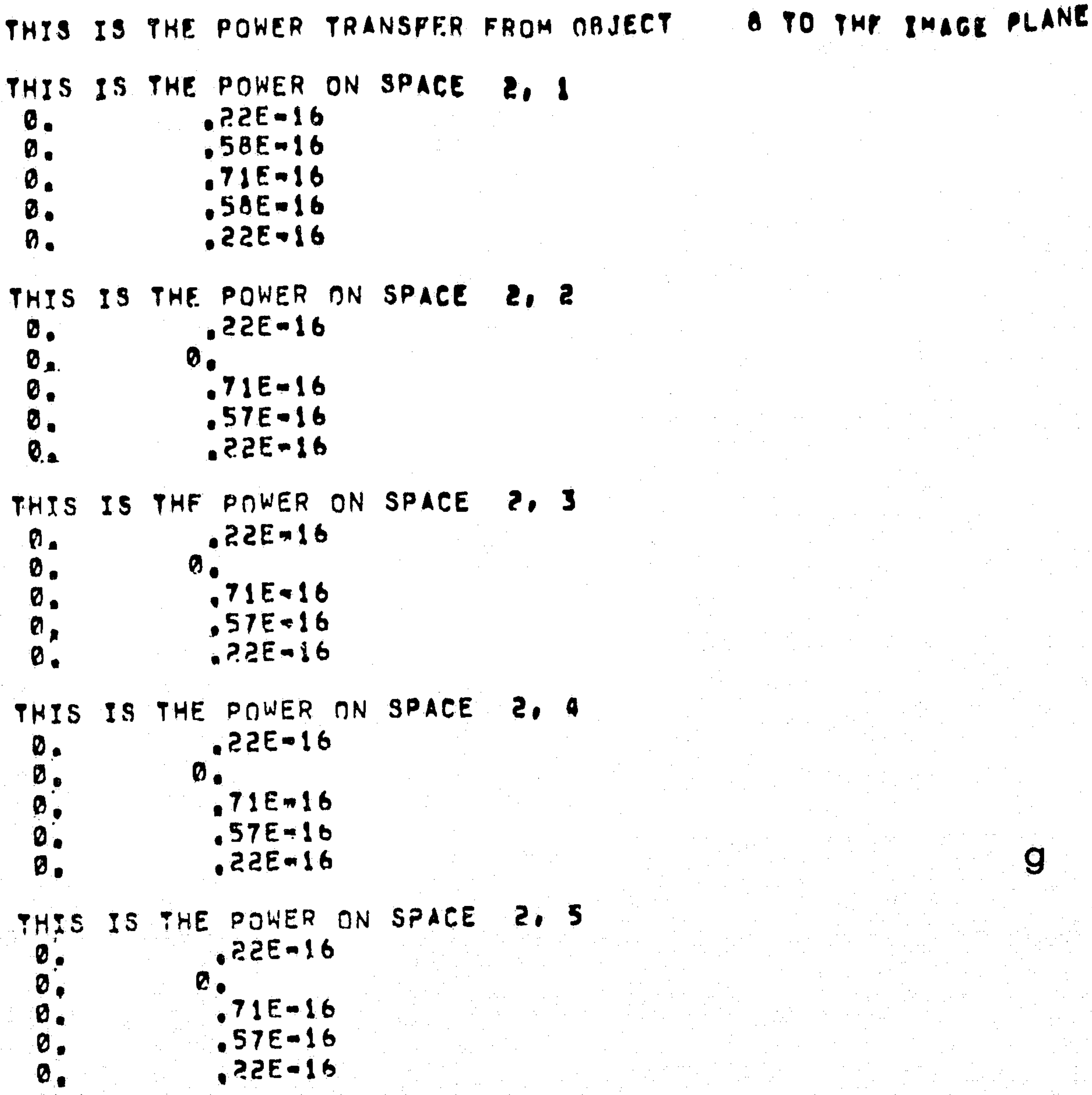


```
THIS IS THE POWER TRANSFER FROM OBJECT    A TO THE IMAGE PLANE

THIS IS THE POWER ON SPACE  1, 1
 Ø.          .22F-16
 Ø.          .58E-16
 Ø.          .71E-16
 Ø.          .58E-16
 Ø.          .22E-16

THIS IS THE POWER ON SPACE  1, 2
 Ø.        Ø.
 Ø.          .58E-16
 Ø.          .71E-16
 Ø.          .57E-16
 Ø.          .22E-16

THIS IS THE POWER ON SPACE  1, 3
 Ø.        Ø.
 Ø.          .58E-16
 Ø.          .71E-16
 Ø.          .57E-16
 Ø.          .22E-16

THIS IS THE POWER ON SPACE  1, 4
 Ø.        Ø.
 Ø.          .58E-16
 Ø.          .71E-16
 Ø.          .57E-16
 Ø.          .22E-16

THIS IS THE POWER ON SPACE  1, 5
 Ø.        Ø.
 Ø.          .58E-16
 Ø.          .7ØE-16
 Ø.          .57E-16
 Ø.          .22E-16
```

Object transferring power to the image plane was sectioned 5 by 2.

The image plane had five *radius* (or axial) sections. Output for the other four *radial* follows.

Fig. 14. Continued.



```
THIS IS THE POWER TRANSFER FROM OBJECT    8 TO THF IMAGE PLANE

THIS IS THE POWER ON SPACE  2, 1
 0.         .22E-16
 0.         .58E-16
 0.         .71E-16
 0.         .58E-16
 0.         .22E-16

THIS IS THE POWER ON SPACE  2, 2
 0.         .22E-16
 0.        0.
 0.         .71E-16
 0.         .57E-16
 0.         .22E-16

THIS IS THF POWER ON SPACE  2, 3
 0.         .22E-16
 0.        0.
 0.         .71E-16
 0.         .57E-16
 0.         .22E-16

THIS IS THE POWER ON SPACE  2, 4
 0.         .22E-16
 0.        0.
 0.         .71E-16
 0.         .57E-16
 0.         .22E-16

THIS IS THE POWER ON SPACE  2, 5
 0.         .22E-16
 0.        0.
 0.         .71E-16
 0.         .57E-16
 0.         .22E-16
```

Fig. 14. Continued.

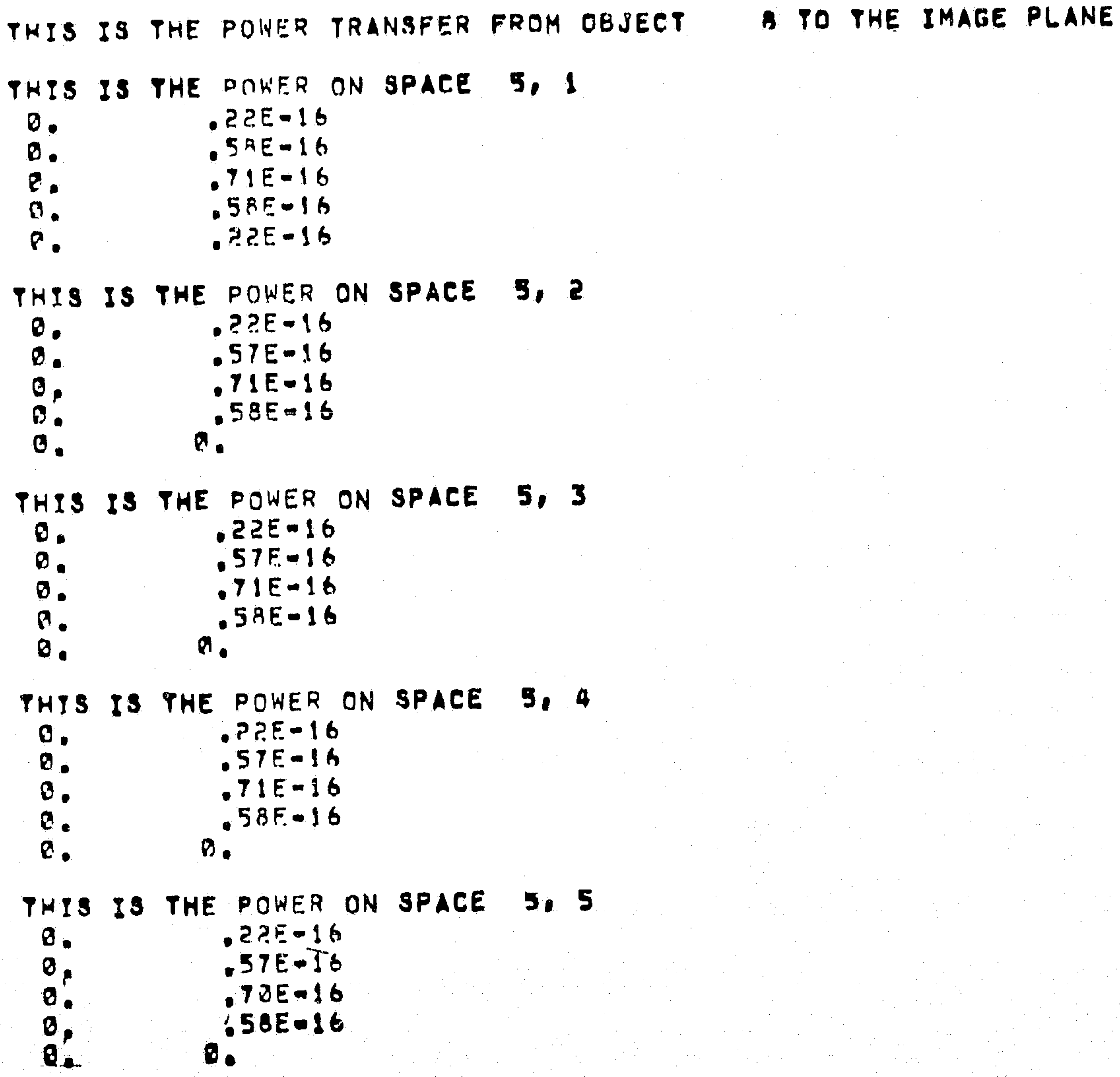
```
THIS IS THE POWER TRANSFER FROM OBJECT    8 TO THE IMAGE PLANE

THIS IS THE POWER ON SPACE  3, 1
 0.         .22E-16
 0.         .58E-16
 0.         .71E-16
 0.         .58E-16
 0.         .22E-16

THIS IS THE POWER ON SPACE  3, 2
 0.         .22E-16
 0.         .58E-16
 0.        0.
 0.         .58E-16
 0.         .22E-16

THIS IS THE POWER ON SPACE  3, 3
 0.         .22E-16
 0.         .58E-16
 0.        0.
 0.         .58E-16
 0.         .22E-16

THIS IS THE POWER ON SPACE  3, 4
 0.         .22E-16
 0.         .58E-16
 0.        0.
 0.         .58E-16
 0.         .22E-16

THIS IS THE POWER ON SPACE  3, 5
 0.         .22E-16
 0.         .58E-16
 0.        0.
 0.         .58E-16
 0.         .22E-16
```

h

Fig. 14. Continued.

```
THIS IS THE POWER TRANSFER FROM OBJECT    8 TO THE IMAGE PLANE

THIS IS THE POWER ON SPACE  4, 1
 0.         .22E-16
 0.         .58E-16
 0.         .71E-16
 0.         .58E-16
 0.         .22E-16

THIS IS THE POWER ON SPACE  4, 2
 0.         .22E-16
 0.         .57E-16
 0.         .71E-16
 0.        0.
 0.         .22E-16

THIS IS THE POWER ON SPACE  4, 3
 0.         .22E-16
 0.         .57E-16
 0.         .71E-16
 0.        0.
 0.         .22E-16

THIS IS THE POWER ON SPACE  4, 4
 0.         .22E-16
 0.         .57E-16
 0.         .71E-16
 0.        0.
 0.         .22E-16

THIS IS THE POWER ON SPACE  4, 5
 0.         .22E-16
 0.         .57E-16
 0.         .71E-16
 0.        0.
 0.         .22E-16
```

Fig. 14. Continued.



```
THIS IS THE POWER TRANSFER FROM OBJECT    8 TO THE IMAGE PLANE

THIS IS THE POWER ON SPACE  5, 1
 0.         .22E-16
 0.         .58E-16
 0.         .71E-16
 0.         .58E-16
 0.         .22E-16

THIS IS THE POWER ON SPACE  5, 2
 0.         .22E-16
 0.         .57E-16
 0.         .71E-16
 0.         .58E-16
 0.        0.

THIS IS THE POWER ON SPACE  5, 3
 0.         .22E-16
 0.         .57E-16
 0.         .71E-16
 0.         .58E-16
 0.        0.

THIS IS THE POWER ON SPACE  5, 4
 0.         .22E-16
 0.         .57E-16
 0.         .71E-16
 0.         .58E-16
 0.        0.

THIS IS THE POWER ON SPACE  5, 5
 0.         .22E-16
 0.         .57E-16
 0.         .70E-16
 0.         .58E-16
 0.        0.
```

Fig. 14. Continued.

focus on reducing the scatter from object eight by improved coatings, vane structure, or by preventing the power from reaching object eight. When the scatter from object eight is reduced by two orders of magnitude, then attention can be focused on improving object four.

If the system was satisfactory with the present power transfers from object eight, it might be financially beneficial to reduce the efficiency of object four by using less expensive coatings or less structure. In a well designed system, most of the critical objects should be contributing about equal power to the image plane.

ICOL--Besides the three arrays, program five prints out the variable ICOL, when at any level the program has found a source-collector combination in tables ITAB2 and IEMIT that satisfies the request of LIMITS and LOOK4. See above for the description of the input data. ICOL will assist the user in determining if all the desired transfers were input and will indicate approximately where the program was if a time limit is encountered during a computer run. The single integer value printed, ICOL, indicates that the program did or was doing the power transfer from source section IEMIT(ICOL) to collector section ITAB2(ICOL).

V. PROGRAM VARIABLES

A. Program One

The variables that are considered to be of importance are briefly described to aid the programmer in making additions or changes in the program. The variables appear alphabetically by program or subroutine. If a variable appears in common, it will be described in the first routine in which it appears. Sometimes it will be described in a subsequent routine if it is very significant or used frequently.

* * *

Program BAFFLE

ALAYOUT(100,7) An array that stores the information about the layout of the optical system being analyzed. Column one is coded information that tells whether the y-$\bar{y}$ heights are known or not. Column two is the y value (marginal ray height) at the plane in space. Column three is the $\bar{y}$ (chief ray height) at this plane in space. Column four is the distance from the present plane to the plane containing the next object. Column five is the radial height of the object in this plane. Column six is the index of the space containing the plane under consideration. Column seven is the object number. Each object has two planes with the same object number. Two z locations and corresponding radii describe cones, cylinders, and disks. Column eight is the direction of the normal to the surface. Column nine is the number of axial sections on the object.

ALPHA The y value on a conjugate line parallel to the image line where the $\bar{y}$ value is that of the plane being considered.

ANGLE1 The angle from the reference point to the first edge of the aperture as the scan is made in a counterclockwise direction.

ANGLE2 The angle from the reference point to the second edge of the aperture when the scan is in a counterclockwise direction.

APERS The number of apertures in the system. Presently it can handle up to 30 apertures.

APERTUR(30,3) The array that holds the information about the apertures. Column one is the maximum height of the aperture. Column two is the minimum height of the aperture (for example the radius of the hole in the primary of a Cassegrainian system). Column three is the row in ALAYOUT in which this aperture appears.

Program one
BAFFLE (continued)

BETA The y value on a conjugate line parallel to the image line where the $\bar{y}$ value is that of the plane being considered. It is like ALPHA but for a different $\bar{y}$ value.

CHI The Lagrange invariant.

CRITH The critical height at which the imaged baffle is seen at the critical distance CRITL.

CRITL The critical distance at which an imaged baffle is seen.

DMAX The maximum distance at which any object edge is imaged.

DMIN The minimum distance at which any object edge is imaged.

EXTRA EXTRA = 100, an integer constant. Whenever the image of an object is split because it passes through infinity or is partially obscured by another object and is seen as two parts, the second part is made into a separate object and is given an object number 100 more than the original object.

HID An 80-letter heading to be printed out at the start of the output to briefly describe the system being analyzed.

HMAX The maximum height of all the imaged objects.

IMGLNES The slope of the image line in the y-$\bar{y}$ diagram.

JOBJS Counts the number of objects that are seen from the reference point in the image plane.

LIMIT1 The reference line in ALAYOUT into which new object data are entered when an object is split.

LOCAT1 The critical baffle height (imaged or otherwise) of the end of the object that shadows or is shadowed by another object.

LOCAT2 The critical baffle distance (imaged or otherwise) of the end of the object that shadows or is shadowed by another object.

LOWSIDE If its value is one, then the program is checking objects above the optical axis. If the value is two, the reference point in the image plane has been changed to its negative value. By symmetry, the same program is used to check the objects appearing below the optical axis.

N The number of reference planes in ALAYOUT. This number is increased when objects are split.

NOBJS The number of objects in the system as described in ALAYOUT.

Program one
BAFFLE (continued)

NPOINTS — Twice the value of JOBJS. It takes two sets of values of z position and radius to describe a cone.

PHI — The value of PI = 3.14159265358979.

POLAR(100,5) — Holds the values of the imaged objects in polar coordinates relative to the reference point in the image plane.

PPRIME(100,7) — Contains the sections of the imaged objects that are seen from the reference point in the image plane. Column one is the y' value. This is the value of the marginal ray at the image plane of the object. Column two is the $\bar{y}'$ value. Column three is the distance of the image from the image plane. Column four is the imaged baffle height. Column five is the object number. Column six is the status. Column seven is the row number in ALAYOUT by which this object is described.

PRIME(100,4) — The image of all objects in ALAYOUT into image space as seen from the image plane. Column one is the y' value. Column two is the $\bar{y}'$ value. Column three is the distance from the image plane. Column four is the imaged baffIe height.

PRINTS — If this word is read as PRINTIT, then the long version of the printout will run. This should be done only if one wants to follow the calculations of each subroutine.

SIGHT — The imagedheight of the object under consideration minus the reference point height.

SLOPE(100) — The slope of the line in the y-$\bar{y}$ diagram on which this reference plane falls.

SPACE(100) — The space in which this object falls.

SPACENO — The total number of spaces considered. It equals the number of lines in the y-$\bar{y}$ diagram.

TPOLAR1 — The distance from the reference point to a possible critical point along some object or image of an object.

TPOLAR2 — Similar to TPOLAR1 but handles the second point along an object where it may reappear from behind the object shadowing it.

VALUE1 — The code number (2.) for the end of an object that is seen.

VALUE2 — The code number (4.) for the start of an object that is seen.

YEPRIME — The y value at the exit pupil.

YZERO — The height in the reference plane from which the system sees out for the analysis.

Program one

Subroutine YYBARIM

M The row in ALAYOUT being considered.

***** See BAFFLE for other variables

Subroutine PRIMEMX

CONJK The slope of the conjugate line.

DMAX The maximum positive distance of any object edge.

DMIN The longest distance from the image plane in the negative z direction.

HMAX The maximum height of all the imaged objects.

***** See BAFFLE for other variables.

Subroutine POLARSP

SIGHT The image height of the object under consideration minus the reference point height.

X The slope of the line connecting the reference point and the imaged object.

***** See BAFFLE for other variables.

Subroutine ORDERP

***** See BAFFLE for identification of variables.

Subroutine SPACES

***** See BAFFLE for identification of variables.

Subroutine SLOPES

L The line reference number of an aperture as it appears in ALAYOUT.

NMINUS1 NMINUS1 equals the number of reference lines in ALAYOUT (which is N) minus one.

NN The line referenced in APERTUR.

***** See BAFFLE for other variables.

Subroutine SPLIT

B The line referenced in ALAYOUT. Used to check if an object falls before or after an aperture.

CONJK The slope of the conjugate line.

D If an object passes through infinity, the program splits the original object into two objects. In order to do this, it must define two new planes to describe the two cones. One plane describes the distance and radius of

Program one
Subroutine SPLIT
(continued)

the imaged object that is imaged toward minus infinity, and the other plane is the distance and radius as the object is imaged toward the plus infinity distance. D is that distance at which the program calculates the reference plane information. It is the minimum (or maximum) distance of all objects not passing through infinity. If this distance happens to be the distance D, then D is decreased (or increased) by 10% so that the object passing through infinity has a finite length.

DIST When an image passes through infinity, a new plane of information is created in ALAYOUT. DIST is the fourth column entry, which is the distance from one reference plane to the succeeding one.

HPRIME The radius of the imaged object that passes through infinity. HPRIME is that radius at distance D from the image plane.

M M holds the old N value while certain arrays are enlarged to hold the new planes of information.

NOMORE1 The line number minus one in ALAYOUT of the first plane appearing in ALAYOUT of an object passing through infinity.

NOMORE2 The line number minus one in ALAYOUT of the second plane describing an object in ALAYOUT that passes through infinity.

OBJH Object height of the new edge of a split object. This object height is the real object height that appears in ALAYOUT.

S The slope of the real object as taken from ALAYOUT.

SP The slope of the imaged object as taken from PRIME.

X The length of the object in real space.

XFOUR The distance from one object plane in ALAYOUT to the next reference plane. Column four in ALAYOUT.

XONE The distance from the first reference plane after the object to the reference plane under consideration.

XPRIME The distance that is used to terminate an object that passes through infinity.

XTHREE The distance from one reference plane in ALAYOUT to the next.

XTWO The distance from the first reference plane after the object to the reference plane under consideration. It is like XONE but the reference plane is the second one appearing in ALAYOUT for a given object number.

Program one
Subroutine SPLIT
(Continued)

Y — The change in height of a given object. Also the y value for the new plane of information for a split object.

YBAR — The $\bar{y}$ value for the new plane of information for a split object.

YBARP — The $\bar{y}$ prime value of the split object.

YONE — The real object height of the first reference plane of a given object.

YP — The y prime value of the split object.

YTWO — The real object height of the second reference plane of a given object.

Subroutine APTOBST

AA — Object number from ALAYOUT.

APTDIST — Aperture distance.

APTHP1 — Aperture height in real space or image space of the first point (maximum height) in APERTUR.

APTHP2 — Aperture height in real space or image space of the second point (minimum height) in APERTUR.

CUT — The distance of the aperture that precedes the image space under consideration. All imaged objects of this space must pass through it.

IBB — Counter from the line in ALAYOUT of the referenced aperture.

ICC — Counter to the second line in ALAYOUT corresponding to the object number of line NN+IBB, where NN is the line of the referenced aperture.

ICUT — The number of lines in ALAYOUT.

IFIRST — The first line in ALAYOUT to which the object being split is referenced.

IT — The line number in ALAYOUT of the aperture to the next image space. This aperture acts as an obscuration to the preceding image spaces.

ITT — The line number in ALAYOUT of the aperture through which all objects of the space being considered must be seen.

LOKATP1 — The reference line in POLAR in which the reference object first appears.

LOKATP2 — The reference line in POLAR in which the reference object appears for the second time.

Program one

Subroutine APTOBST (Continued)

NN — The line in ALAYOUT while searching for apertures.

SPACEZ — The space that determines the apertures through which all images of objects must appear and by which all real objects in this space are obscured if they pass behind it.

TMAG — The magnification of the aperture that acts as an obscuration for the space under consideration.

XX — The farthest point of an object (imaged or not) from the image plane.

YBARFST — The $\bar{y}$ value of the first aperture, which is the one that acts as an obscuration.

YBARSEC — The $\bar{y}$ value of the second aperture, which is the aperture through which all the imaged objects must pass.

***** See BAFFLE for other variables.

Subroutine CLAPT

APTDPRM — The distance of the referenced aperture in imaged space.

APTHP1 — Aperture height in real or imaged space (whichever is being referenced) of the first point (maximum height) in APERTUR.

APTHP2 — Aperture height in real or imaged space (whichever is being referenced) of the second point (minimum height) in APERTUR.

CLINES — The slope of the line of the imaged object.

IAT — The space in which the referenced object exists.

INS — As the scan is made in POLAR, INS is the reference number appearing in column three of POLAR that tells which space the object is in.

LOKATP1 — The line reference number (column three) in POLAR the first time the object number is referenced.

LOKATP2 — The line reference number (column three) in POLAR the second time that the object number is referenced.

MAPT — The third column of APERTUR that holds the line number in ALAYOUT of the aperture.

SIGHTSL — The slope of the line of sight from the reference point in the image plane to the upper or lower edge of the aperture.

X — The angle of either the front or back edge of the object as in column two of POLAR.

***** See BAFFLE for other variables.

Program one

Subroutine BAFSEEN

CLINES The slope of the line of the imaged object.

IJKL The line number in POLAR the second time the object number appears in column four.

IPK The line number in POLAR the first time the number of the object being shadowed appears (or at least appears after the original object that does the shadowing).

IPP The line number in POLAR the first time the object number appears in column four.

IPPRIME The line into which information is stored in array PPRIME.

K The counter from line I in POLAR to the first time the second object number appears in column four of POLAR.

L The counter from line I+K in POLAR to the second time the second object appears in column four of POLAR.

LA The line in ALAYOUT in which the first object number appears in ALAYOUT for the first time.

LB The line in ALAYOUT in which the first object number appears in ALAYOUT for the second time.

LOCATCP Locates what is considered to be the critical point in POLAR.

LOKATP1 The reference line in ALAYOUT of the first object the first time it appears in POLAR.

LOKATP2 The reference line in ALAYOUT of the first object the second time it appears in POLAR.

LOKATP3 The reference line in ALAYOUT of the second object the first time it appears in POLAR.

LOKATP4 The reference line in ALAYOUT of the second object the second time it appears in POLAR.

M The line in POLAR the second time the first object appears in POLAR.

SIGHTSL The slope of the line from the reference point to the critical point.

TPOLAR The distance from the reference point to the critical point on the second object.

***** See BAFFLE for other variables.

Subroutine ANGSEEN

BAFFSEC One-fourth the axial length of the baffle that is seen.

CONJKPC The slope of the conjugate line through all the subsections of the baffle that is seen.

Program one
<u>Subroutine ANGSEEN</u>

CRITL The direction cosine from the z axis.

CRITM The direction cosine from the y axis.

CRITPTS The array holds the following information: column one has the y value in real space of this section of the baffle that is seen, column two is the $\bar{y}$ value, column three is the distance from the subsection in real space to the next refracting or reflecting surface, column four is the subsection baffle height in real space, column five is the angle at which this baffle subsection is seen (the angle is in degrees).

DENOM The distance from the reference point in the image plane to the different subsections. It is used in the calculation of the direction cosines.

FORSLOP Defined as an integer. The line number in SLOPE of the object.

HPRIMEI The radius at each subsection of the baffle that is seen.

HREFOBJ The height of the ray from the reference point to the subsections on the last (nearest the baffle) refracting or reflecting surface in image space.

HREFPRI The height of the ray from the reference point to the subsection on the last (nearest the baffle) refracting or reflecting surface in real space.

H1 The height of one end of the baffle that is seen in image space.

H2 The height of one end of the baffle that is seen in image space.

IFINDOR The reference line in ALAYOUT of the object being processed. It is the first time that the object has appeared in POLAR, which may or may not be the first time it appears in ALAYOUT.

IFINDSE Finds the line in ALAYOUT corresponding to the other end of the object defined by line IFINDOR.

IPPRIME The line in PPRIME being referenced.

K The number of the subsection. K can equal any integer from one to four.

KONIMGL The y value at the image of the pupil for the image line in the y-$\bar{y}$ diagram being considered.

MM The line number in CRITPTS.

NOMORE1 The line number minus one in ALAYOUT of the first plane describing an object.

Program one
Subroutine ANGSEEN
(Continued)

NOMORE2 — The line number minus one in ALAYOUT of the second plane describing an object.

OBJL — The direction cosine from the z axis of the object.

OBJM — The direction cosine from the y axis of the object.

REFPLAN — The reference line of the last imaging surface for the given object.

RRPRMAG — The magnification of the last imaging surface.

SECTION — One-fourth of the length of the section of the baffle that is seen.

SLOPB — The slope of the line of the imaged baffle.

SMAGINV — The inverse of the magnification of the object.

THOBJ1 — The object number under consideration.

XONE — The distance from the first reference plane after the object to the reference plane under consideration. The distance is the real space distance.

XPRIME1 — The distance of the subsection from the image plane in image space.

XTWO — The distance from the first reference plane after the object to the reference plane under consideration. It is like XONE, but the reference plane is the second one appearing in ALAYOUT for a given object number.

X1 — The distance from the image plane of one edge of the imaged object that is seen.

X2 — The distance from the image plane of one edge of the imaged object that is seen. It is like X1, but the other plane of information defining the object.

YBARFF1 — The $\bar{y}$ value of one end of the object in image space. The value is taken from column two in PPRIME.

YBARFF2 — The $\bar{y}$ value of one end of the object in image space. The value is taken from column two in PPRIME. It is like YBARFF1 but is the other plane of information defining the object.

YBARX — The difference in $\bar{y}$ height of the two planes defining the object. YBARX = BYARFF2 - YBARFF1.

YBFFP — The y value of the subsections of the portion of the imaged baffle that is seen.

YBRBFFP — The $\bar{y}$ value of the subsections of the portion of the imaged baffle that is seen.

Program one

Subroutine ANGSEEN
(Continued)

YFF1 The y value in image space of one of the ends of an imaged object seen from the image plane. The value is taken from column one of PPRIME.

YFF2 The y value in the image space of one of the ends of an imaged object seen from the image plane. The value is taken from column one of PPRIME. It is like YFF1 but the other plane of information defining the object.

YONE The real object height of the first reference plane of the given object.

YSLOPE The slope of the image line in the y-$\bar{y}$ diagram.

YTWO The real object height of the second reference plane of a given object.

***** See BAFFLE for other variables.

Subroutine OBJSEEN

AOBJN An object number. Searches PPRIME for the largest object number in its array.

IA Line number in ALAYOUT of object being processed in POLAR.

IB Line number in ALAYOUT of object being processed in POLAR.

IJ Line number in POLAR being processed.

JOBJS2 Twice the number of objects seen from the given reference point. Because it takes two planes to describe each object there are JOBJS2 lines in PPRIME.

K The line counter in POLAR from the line where the object is first referenced in POLAR to the line where the object is next referenced.

M Line reference number in PPRIME.

MOBJN The maximum object number in PPRIME. It is equivalent to AOBJN.

***** See BAFFLE for other variables.

Subroutine DOUBLEP

I The line number in POLAR of the plane of information being processed. It is that line in POLAR where the referenced object number first appears.

IPPRIME The line in PPRIME being referenced.

J I + J is the line number in POLAR of the object the second time its object number appears in POLAR.

Program one
Subroutine DOUBLEP
(Continued)

X — The polar angle (column two in POLAR) of the object for the line being referenced.

***** See BAFFLE for other variables.

Subroutine TPOLARX

CLINES — Slope of the imaged baffle.

LOKATP1 — The line reference number (column three) in POLAR the first time the object is referenced.

LOKATP2 — The line reference number (column three) in POLAR the second time the object is referenced.

SIGHTSL — The slope of the line of sight from the reference point in the image plane to the point of interest in the image space.

TPOLAR — The distance from the reference point in the image plane to the point of interest in the image space.

***** See BAFFLE for other variables.

Subroutine ENLARGE

BAFHT — For any object split it is the height of the new baffle in the new plane.

CONJKPC — The slope of the conjugate line through the y-$\bar{y}$ point representing the plane where the baffle was split.

D — The distance from the end of the original object to the plane where the object is split.

DIS — The distance from the most forward plane of the object to the plane where the object is split.

IFIRST — The first line in ALAYOUT in which the object is referenced.

IMARK — It marks the increment in line number from line ISEC in ALAYOUT from which the distance D is figured.

IMOVE — Tells how many lines to move the information stored in the various arrays to make room for the newly created information.

ISEC — Either the first or second line in ALAYOUT in which the referenced object occurs.

ISIGN — Either plus or minus one in value. M - ISIGN tells the line number where the information is stored that is being relocated.

Program one
Subroutine ENLARGE
(continued)

KONIMGL — The y value at the image of the pupil for the image line being considered in the y-$\bar{y}$ diagram.

LA — The line in ALAYOUT in which the referenced object number first appears.

LB — The line in ALAYOUT in which the referenced object number appears for the second time.

LC — The line in POLAR in which the referenced object number appears for the first time.

LD — The line in POLAR in which the referenced object number appears for the second time.

LIMIT2 — LIMIT1-1 in value. See BAFFLE for the value of LIMIT1.

LOKATP1 — The line reference number (column three) in POLAR the first time the object number is referenced.

LOKATP2 — The line reference number (column three) in POLAR the second time the object number is referenced.

M — The line reference number in the various arrays from which information is taken or stored. Sometimes this information is stored some small line increment from line M.

NOMORE1 — The line number minus one in ALAYOUT of the first plane describing an object.

NOMORE2 — The line number minus two in ALAYOUT of the second plane describing an object.

SLOPB — The slope of the y-$\bar{y}$ line for the split baffle in the y-$\bar{y}$ diagram.

X — The value of the angle from a line parallel to the z axis through the reference point in the image plane to the point in image space being referenced.

XONE — The distance from the first reference plane after the object to the reference plane under consideration.

XTWO — The distance from the first reference plane after the object to the reference plane under consideration. It is like XONE, but the reference plane is the second one appearing in ALAYOUT for a given object number.

Y — The y value of the new edge in real space.

YBAR — The $\bar{y}$ value of the new edge in real space.

YBAR1 — The $\bar{y}$ value of the edge with the smallest $\bar{y}$ value of the object being split.

Program one
Subroutine ENLARGE
(Continued)

YBAR2 The $\bar{y}$ value of the edges with the largest $\bar{y}$ value of the object being split.

YONE The real object height of the first reference plane of a given object.

YTWO The real object height of the second reference plane of a given object.

Y1 The y value of the edge with the smallest $\bar{y}$ value of the object being split.

Y2 The y value of the edge with the largest $\bar{y}$ value of the object being split.

***** See BAFFLE for other variables.

Subroutine CHECK

***** See BAFFLE for identification.

Subroutine LOWER

IAGAIN If IAGAIN equals one, the program is reinserting the original real and image space data for an object that was split and extracting the extra plane of information that was stored in the line just before the second time the original object was referenced.

INSERT The line where the extra plane of information was stored in ALAYOUT.

***** See BAFFLE for other variables.

Subroutine ELLIPSE

DEXIT The distance to the exit pupil

ELLINFO(100,6) Column one stores the height of the projection of the center of the referenced pupil onto the exit pupil. Similarly, column two is the radius of the projected pupil, column three is also the same radius, column four is the maximum height of the projection, column five is the minimum height of the projection, and column six is the line referenced in ALAYOUT.

NNN Equals N, the number of lines in SLOPE.

RADIUSA The radius of the projected aperture.

SHOWI It is intentionally undefined. It is preset to negative infinity. In this manner the program can reset certain arrays.

XONE If XONE is negative, it is the z distance from the image plane of the imaged aperture. Otherwise XONE equals zero, the location of the image plane.

Program one
Subroutine ELLIPSE
(Continued)

XTWO The z distance of the imaged pupil if the imaged distance is positive (z positive), otherwise the value of XTWO is zero, the location of the image plane.

YBMAX The radius of the imaged pupil projected onto the exit pupil.

YCENTER The height of the projection of the center of the aperture onto the exit pupil.

YMAX The maximum height of the projection of the aperture onto the exit pupil.

YONE The height of the imaged pupil (or YZERO if XONE is zero).

YTWO Equals YZERO unless XTWO is not equal to zero. Then YTWO equals the height of the imaged pupil.

***** See BAFFLE for the other variables.

B. Program Two

Program CALRAD

ALAYOUT(100,7) This array stores the information about the layout of the optical system being analyzed. Column one is coded information that tells whether the y-$\bar{y}$ heights are known or not. Column two is the y value (marginal ray height) at the plane in space. Column three is the $\bar{y}$ (chief ray height) at this plane in space. Column four is the distance from the present plane to the plane containing the next object. Column five is the radial height of the object in this plane. Column six is the index of the space containing the plane under consideration. Column seven is the object number. Each object has two planes with the same object number. Two z locations and corresponding radii describe cones, cylinders, and disks. Column eight is the direction of the normal to the surface. Column nine is the number of axial sections on the object.

APERTUR(30,3) The array that holds the information about the apertures. Column one is the maximum height of the aperture. Column two is the minimum height of the aperture (for example the radius of the hole in the primary). Column three is the row in ALAYOUT in which this aperture appears.

CHI The Lagrange invariant.

DMAX The maximum distance that any object edge is imaged.

DMIN The minimum distance that any object edge is imaged.

HID An 80-letter heading to be printed out at the start of the output to briefly describe the system being analyzed.

HMAX The maximum height of all the imaged objects.

IFST The reference line number in ALAYOUT of the object that starts a new space. For the first space it is line one, which is the object. For the second space it will be the first imaging surface, etc.

IMGLNES The slope of the image line in the y-$\bar{y}$ diagram of the line being imaged along.

INDEXT An index array initialized by the Fortran routine OPENMS and WRITMS. The array contains information as to where on the disk the information is stored so it can be retrieved. DO NOT change the program to write in this array.

N The number of reference planes in ALAYOUT. This number is increased when an object is split.

NAPT The number of apertures in the system. Presently it can handle up to 30 apertures.

NDEX The index of the space from which the program is looking.

Program two
Program CALRAD
(Continued)

NMINUS1 N-1. One less than the number of lines in ALAYOUT.

NO The space from where the images are seen.

OMEGA(10,2) Stores the omega-omega bar information of the lines in the y-$\bar{y}$ diagram. $\Omega = nu/\mathcal{K}$, where n is the index of the space, u is the marginal ray angle, and $\mathcal{K}$ is the Lagrange invariant. $\bar{\Omega} = n\bar{u}/\mathcal{K}$, where $\bar{u}$ is the chief ray angle in this space.

P Equivalenced to I, the line number in ALAYOUT.

PLOTSIT If PLOTSIT equals PLOT the output will include a plot of all the imaged spaces.

PRIME(100,4) The array holding the information of all the objects as they are imaged from the given space. Column one is the y' value, column two is the $\bar{y}$ value, column three is the imaged baffle distance relative to the reference plane for this space, column four is the imaged baffle height.

PRINTS A code word to print out the long or short version of the output. If PRINTS equals PRINTIT, it will print the long version. This should be done only if a mistake is suspected or if one is just learning how the program proceeds.

REFER(10,2) Contains the y and $\bar{y}$ values of the planes that are used as the reference planes for each of the spaces.

SCALE The scale at which the plot will be made.

SLOPE(100) The slope of the line in the y-$\bar{y}$ diagram on which this reference plane falls.

SPACE(100) The space in which this object falls.

SPACENO The total number of spaces considered. It equals the number of lines in the y-$\bar{y}$ diagram.

YBARREF The $\bar{y}$ value at the reference plane for the space being considered.

YEPRIME The y value at the pupil or image of the pupil as seen for the space being considered.

YREF The y value at the reference plane for the space being considered.

Subroutine SLOPES

L Line number in ALAYOUT.

***** See CALRAD for other variables.

Subroutine YYBARIN

***** See CALRAD for identification of the variables.

Program two

Subroutine SPACES

***** See CALRAD for identification of the variables.

Subroutine LAGRANG

T — The distance from the first plane of the system referenced in ALAYOUT to the next plane of information where the y and $\bar{y}$ values are known.

***** See CALRAD for other variables.

Subroutine IMAGPRI

ALPHA — The y value on a conjugate line parallel to the image line where the $\bar{y}$ value is that of the plane being considered.

BETA — The y value on a conjugate line parallel to the image line where the $\bar{y}$ value is that of the plane being considered. It is like ALPHA but for a different $\bar{y}$ value.

BIG — A big number. (1.E40).

CONJK — The slope of the conjugate line.

D — If an object passes through infinity, the program splits the original object into two objects. To do this, it must define two new planes to describe the two cones. One plane describes the distance and radius of the imaged object that is imaged toward minus infinity, and the other plane is the distance and radius as the object is imaged toward the plus infinity distance. D is that distance at which the program calculates the reference plane information. It is the minimum (or maximum) distance of all objects not passing through infinity. If this distance happens to be the distance D, the D is decreased (or increased) by 10% so that the object passing through infinity has a finite length.

DIST — When an object passes through infinity, a new plane of information is created in ALAYOUT. DIST is the fourth column, which is the distance from one reference plane to the succeeding one.

EXTRA — EXTRA = 100, an integer constant. Whenever the image of an object is split because it passes through infinity or is partially obscured by another object and is seen as two parts, the second part is made into a separate object and is given an object number 100 more than the original object.

HPRIME — The radius of the imaged object that passes through infinity. HPRIME is that radius at distance D from the image plane.

M — The line in the various arrays into which line information is put when it is being relocated to make space available for a split object.

NOMORE1 — The line number minus one in ALAYOUT of the first plane appearing in ALAYOUT of an object passing through infinity.

NOMORE2 — The line number minus one in ALAYOUT of the second plane describing an object in ALAYOUT that passes through infinity.

OBJH — Object height of the new edge of a split object. This object height is the real object height that appears in ALAYOUT.

S — The slope of the real object as taken from ALAYOUT.

SP — The slope of the imaged object as taken from PRIME.

X — The length of the object in real space.

XFOUR — The distance from one object plane in ALAYOUT to the next reference plane. Column four in ALAYOUT.

XONE — The distance from the first reference plane after the object to the reference plane under consideration.

XPRIME — The distance used to terminate an object that passes through infinity.

XTHREE — The distance from one reference plane in ALAYOUT to the next.

XTWO — The distance from the first reference plane after the object to the reference plane under consideration. It is like XONE but the reference plane is the second one appearing in ALAYOUT for a given object number.

Y — The change in height of a given object. Also the y value for the new plane of information for a split object.

YBAR — The $\bar{y}$ value for the new plane of information for a split object.

YBARP — The $\bar{y}'$ value of the split object.

YONE — The real object height of the first reference plane of a given object.

YP — The y' value of the split object.

YSS — The y value at the image of the pupil as seen from the space being considered.

YTWO — The real object height of the second reference plane of a given object.

YYYY — A frequently appearing value equal to YREF/YSS-1.

***** See CALRAD for other variables

Subroutine SCALES

***** See CALRAD for the definition of the variables.

Program two

Subroutine LOWER

IAGAIN If IAGAIN equals one, the program is extracting the second line of information that was added for a split object. If IAGAIN equals two it is extracting the first line of added information.

INSERT The line in ALAYOUT where the extra plane of information was stored.

***** See CALRAD for other variables in LOWER.

Subroutine SEE

DMAXS The maximum distance of any of the plotted objects. The objects are first scaled, and if any portion of the plot at this scale falls within the limits of ±5.4 in. in height and ±60 in. in distance, then that portion is plotted.

DMINS The minimum distance of any of the plotted objects. See DMAXS above.

HGTMAX The maximum height of any of the plotted objects. It will be 5.4 in. or less, so if the image of an object is larger than this, it must be scaled down.

NOTE The words to be written on the plot to identify it.

S The slope of the line to be plotted.

SC The program scales all imaged objects and then determines the minimum and maximum scaled distances. If these distances are greater or less than 60 in. the program will rescan the imaged data for objects or portions of objects that fall within these limits. It will set new values for DMAXS and DMINS (usually they will now be +60 in. and - 60 in. but not necessarily). The value of 60 in. was chosen as a large but practical plot size.

SCALE If the scale is not read in, then the program will establish a scale such that ALL imaged objects will fall within the limits of the plotting paper. If there is one very long object, everything will be scaled down accordingly and perhaps be too small. The desired plot size can be read in and overrides the above feature. Those objects outside the limits are not plotted, and a message is written on the output. If only a portion of the object is plotted, then a message is again printed on the output.

X The number of the object being plotted.

XX The space from which this plot is seen.

X1 One of the x coordinates of the plotted object.

Program two
Subroutine SEE
(Continued)

X2 One of the x coordinates of the plotted object.

Y1 One of the y coordinates of the plotted object.

Y2 One of the y coordinates of the plotted object.

ZA The angle at which the letters are plotted onto the plot.

ZB The offset value used to place the numbers along an object in the plotted output.

ZZ The height at which the numbers are written on the plotted output.

***** See CALRAD for identification of other variables.

Subroutine IMGLINE

ISEC One of the two imaging apertures that defines the space that the objects are in. ISEC is line number in ALAYOUT in which the second defining aperture appears.

***** See CALRAD for identification of the other variables.

Subroutine CHANGEA

OFFSET The distance from the image plane to the reference plane of the last image space.

X The object number.

XI The line in ALAYOUT where object X first appears.

XIJ The line in ALAYOUT in which the object X appears a second time.

***** See CALRAD for identification of other variables.

C. Program Three

Program VIEWFAC

AAT The difference in the z coordinate from the source point to the collector point on the image (if not imaged, then on the real collector object).

ACC Each object section can be subdivided into smaller subsections. Then the power transfer is made from each subsection on the source to each subsection on the collector. Sometimes, because of obscurations, not all these possible transfers occur. ACC is the number of such transfers that do occur, and the data are averaged over this number of occurrences.

AL The direction cosine from the x axis of that vector that enters the collector in REAL space as the path of radiation is traced from the source to the collector. This value is not necessarily the same as the vector BL from the source to the IMAGE of the collector section.

AM The direction cosine from the y axis of that vector that enters the collector in REAL space as the path of radiation is traced from the source to the collector.

AN The direction cosine from the z axis of that vector that enters the collector in REAL space as the path of radiation is traced from the source to the collector.

ANGAB The angle (in radians) subtended by each subsections's circular edge.

ANGB090 The cosine of the angle between the vector (AL,AM,AN) and its projection in the y-z plane in the local coordinate system of the collector. The basis vectors of this local coordinate system are ZXB,ZYB,ZZB.

ANGLE(1764,4) Column one is the average computed angle out of the source, that is the average SANG. Column two is the average ANGS090 angle. This is the angle between the outgoing vector and its projection in local y-z plane. Column three is the average computed angle into the collector in real space, that is the average BANG. Column four is the average ANGB090 angle. This is the angle between the incoming vector into the collector in real space and its projection into the local y-z plane.

ANGSS Two PI radians divided by the number of subsections in the circumference of the object.

ANGS090 The angle between the incoming vector into the collector in real space and its projection in the local y-z plane.

APARENP Equals SSCOS, the angle between the normal to the source and the outgoing vector.

APERTUR(10,3) The array holds the information about the apertures. Column one is the maximum height of the aperture. Column two is the minimum height of the aperture (for example the radius of the primary hole).

AXT The difference in the x coordinate of the incoming vector to the collector where one point is the collector point and the other point is the point where the vector penetrates the last imaging surface.

AY The y value at the last imaging surface in real space.

AYB The $\bar{y}$ value at the last imaging surface. The $\bar{y}$ value is the real space $\bar{y}$ value.

AYT The incoming vector is defined by two points, the collector point and the point on the last imaging surface where the projection of this vector reflects at the surface. AYT is the difference in the y coordinate of these two points.

AZT Like AXT and AYT but the difference in the z coordinate of the two points.

B B is the current axial section of the collector.

BAFAREA The area of the imaged baffle subsection.

BAFCOS The cosine of the angle between the normal to the imaged collector surface and the incoming vector.

BANG The incoming vector to the collector in real space is projected into the local y-z plane of the collector point, and the angle between this projection and the plus z vector (in local coordinates) in degrees is the value BANG.

BB The number of axial sections in the collector.

BBT The vector from the source to the image of the collector is defined by two points, the source point and the imaged collector point. BBT is the difference in the y coordinate of these two points.

BISDELT The maximum number of SUBsections along the axis of the source.

BISS The maximum number of sections along the axis of the source. If these sections are subdivided (into delta by delta subsections) then there are BISS*DELTA subsections (BISDELT) along the axis.

BL The direction cosine from the x axis of the outgoing vector from the source.

Program three
Program VIEWFAC
(Continued)

BLOCK The value read in by the program that tells the program into which mass storage block the following information should be stored. If left blank the program proceeds sequentially from one of the last BLOCK values read in. In this manner the indiv:dual blocks can be changed without redoing all the calculations.

BM The direction cosine from the y axis of the vector from the source point to the point on the imaged collector.

BN The direction cosine from the z axis of the vector from the source point to the point on the imaged collector.

BRBA The radius of the back edge (the edge with the largest z value in real space) of the subsection of the collector being calculated.

BSS The current axial *section* on the source (NOT the axial subsection).

BSSUBI The current *sub*section *within* the section on the source being considered. BSSUBI cannot get larger than DELTA.

BSUBI The current *sub*section *within* the section of the collector being considered, BSUBI cannot get larger than DELTA.

BZ Half the axial distance between subsections of the collector.

BZSS Half the axial distance between subsections of the source.

C If C reads the word SOURCE, then a new power transfer set will be calculated. Any other word, including blanks, will make the program search for other cue values and treat them as obstructions.

CCP The sum of all the angles BANG calculated for the power transferred from the section on the source to the section on the collector.

CCT The vector from the source to the image of the collector is defined by two points, the source point and the imaged collector point. CCT is the difference in the x coordinate of these two points.

CN Equals NC, the number of the collector.

CONE(1764) Contains the percent of power transferred from the sections on the source to the sections on the collector.

C90 The sum of all the angles ANGB090 calculated for the power transferred from the section on the source to the section on the collector.

D The distance between subsections on the collector.

DCOSBRX
DCOSBRY
DCOSBRZ The direction cosines from the x, y, and z axes, respectively, of the vector connecting the imaged collector point and the point on the z axis that will make this vector lie in a plane normal to the z axis. In a sense

Program three
Program VIEWFAC
(Continued)

it is the imaged baffle radius vector. If the baffle is a cone, then the direction of this vector is determined by DNORMAL, which denotes whether the surface is the inside of the cone or the outside of it.

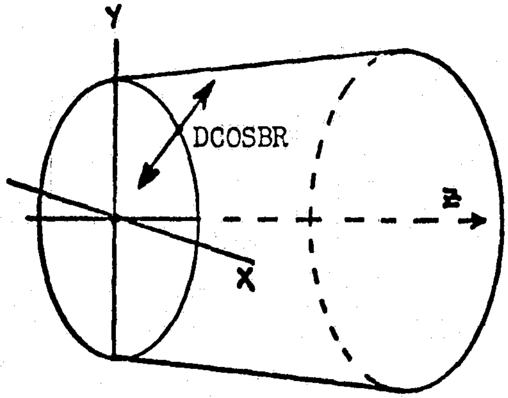


Fig. 15. DCOSBR vector to which the direction cosines, DCOSBRX, DCOSBRY, and DCOSBRZ are calculated.

DCOSBX
DCOSBY
DCOSBZ

The direction cosines from the x, y, and z axes, respectively, of the vector normal to the collector surface in real space at the point being referenced. The direction of the vector determines whether the inside or outside surface of the cone is being considered.

DCOSIX
DCOSIY
DCOSBZI

The direction cosines from the x, y, and z axes, respectively, of the vector that is normal to the *imaged* collector surface at the point being referenced. The direction of the vector determines whether the inside or outside surface of the cone is being considered.

DCOSRX
DCOSRY
DCOSRZ

The direction cosines from the x, y, and z axes, respectively, of the vector that is normal to the z axis, passes through the source point and the z axis. The direction of the vector determines whether the inside or outside surface of the cone is being referenced.

DCOSSX
DCOSSY
DCOSSZ

The direction cosines from the x, y, and z axes, respectively, of the vector that is normal to the surface of the source at the point being referenced. The direction of the vector determines whether the inside or outside surface is being considered.

Program three
Program VIEWFAC
(Continued)

DELTA

Each section of the source and the collector can be subdivided into DELTA by DELTA subsections. Then the power transfer factor is computed from each subsection on the source to each subsection on the collector and then summed for the power transfer from each section to each section. If DELTA is three, that means that there are nine subsections on the source and nine subsections on the collector and the program will make 81 more power transfer calculations. The final power transfer calculations will be more accurate, but the computer costs will go up.

DENOM

The distance between the source point and the image of the collector point.

DENOMBN

The length of the DCOSB vector.

DENOMI

The incoming vector to the collector point in real space is defined by the collector point and the point on the last imaging surface through which the collector point is seen by the source. The length of this vector is DENOMI.

DENOMSN

The length of the vector DCOSS.

DENOMWN

The length of the vector DCOSI.

DMIR

Looking from the source toward the collector, DMIR is the imaged distance of the last imaging aperture. The program automatically checks that the radiation passes through this imaged aperture.

DNORMAL

The normal to the collector surface in real space. If the collector is a cone, a positive DNORMAL value indicates the outside of the cone. A negative DNORMAL indicates the inside of the cone. If the collector is a disk, a positive DNORMAL indicates the plus z direction of the disk is the collector. If DNORMAL is negative, the minus z direction of the disk is the collector.

DSS

The axial distance between subsections on the source.

E

The number of the subsection along the z axis being referenced as the collector.

ENDSIT

A code word set by the program when all the data cards have been read.

EX
EY
EZ

The program calculates a series of vectors that are interrelated to determine the "computed" angle into the collector or out of the source. The angle that is finally used is the projection of the incoming (or outgoing if the source) vector onto the "local" y-z plane at the point on the collector in real space. Three vectors already described are sufficient to establish the local coordinate system and determine the necessary angles. These vectors are $\bar{A}$(AL,AM,AN), $\overline{DCOSB}$(DCOSBX,DCOSBY,DCOSBZ), and $\overline{DCOSBR}$(DCOSBRX,DCOSBRY,DCOSBRZ) for the collector. For the source they are $\bar{B}$(BL,BM,BN), $\overline{DCOSS}$(DCOSSX, DCOSSY,DCOSSZ) and $\overline{DCOSR}$(DCOSRX,DCOSRY,DCOSRZ). If we rename these vectors as $\bar{I}$, $\bar{N}$, and $\bar{R}$, respectively, in either case it will be easier to follow both cases. $\bar{I}$ stands for the incoming or outgoing vector connecting the two points, $\bar{N}$ stands for the normal to the surface vector, and $\bar{R}$ stands for the radius vector. $\bar{R}x\bar{N}=\bar{O}$. $\bar{O}$ has the direction cosines (OX,OY,OZ). $\bar{N}x\bar{O}=\bar{Z}$. $\bar{Z}$ has the direction cosines (ZX,ZY,ZZ). $\bar{I}x\bar{O}=\bar{E}$. $\bar{E}$ has the direction cosines (EX,EY,EZ). $\bar{E}x\bar{O}=\bar{F}$. $\bar{F}$ has the direction cosines (FX,FY,FZ). $\bar{E}$ is the projection of $\bar{I}$ into the plane containing the $\bar{R}$, $\bar{N}$, and $\bar{Z}$ vectors. They determine the "local" y-z plane. $\bar{O}$ is the "local" x axis.

F — The current subsection along the z axis of the source.

FIX — When a disk is reimaged, the normal to that imaged surface can be in the opposite direction of the normal in real space. FIX fixes the normal of the imaged object accordingly.

FRBA — The radius of the front edge (the edge with the smallest z value in real space) of the subsection of the collector being calculated.

G — Equals one-fourth the distance of the closest source subsection-collector subsection combination.

HOLES(20,2) — It contains information as to what the obscurations are and where the information is stored in ALAYOUT. If for obscuration IH HOLES(IH,2) equals zero, the obscuration is an aperture that the radiation must pass through, and HOLES(IH,1) is the line in ALAYOUT that has the size and other information about the aperture. If HOLES(IH,2) is greater than zero, then the shadow is a disk and HOLES(IH,1) is the line in ALAYOUT that has the necessary information about the disk. If HOLES(IH,2) is less than zero, then the obstruction is a cone and HOLES(IH,1) is the object number in ALAYOUT of the cone.

IH — The lines in HOLES being referenced.

IMAX — The number of sections along the axis of the collector.

INANGL — The line in CONE and ANGLE being referenced.

Program three
Program VIEWFAC
(Continued)

INBLOCK The block in the random access storage in which the information will be stored.

INCONE The line in cone being referenced.

INDEX(100,3)
INDEX(100) An index array initialized by the Fortran routines OPENMS and WRITMS. The array contains information as to where on the disk the information is stored so it can be retrieved. DO NOT change the program to write in this array.

IOBJC The line number in ALAYOUT the first time the collector object is referenced.

IOBJCC The line number in ALAYOUT the second time the collector object is referenced.

IOBJS The line in ALAYOUT the first time that the source is referenced.

IOBJSS The line in ALAYOUT the second time that the source is referenced.

ISPC The space the collector is in.

ISPS The space the source is in.

ISS The number of axial sections on the source.

ISS3 The number of sections along the axis of the source times the number of radial sections from section one to the bottom section.

ITAB(100) The number of words or the size of the blocks of information stored by WRITMS.

JLOW There must be an odd number of sections on the circumference of the objects. If one starts at the 12 o'clock position and counts the sections clockwise, the section on the bottom at the six o'clock position is section JLOW.

JMIR The line in ALAYOUT of the last imaging aperture from the source to the collector.

MMAG The magnification of the last imaging aperture.

N The number of lines in ALAYOUT.

NBLOCK The block in mass storage in which the information is stored.

NC The number of the collector.

NDEX The index of the space of the source.

Program three
Program VIEWFAC
(Continued)

NEWITAB Read as input. If blank, it will create a new ITAB index. If it is not blank, then the program will read from mass storage the old ITAB and overwrite where necessary.

NS The number of the source object.

NSIZE The number of words stored in this block of mass storage information.

NTAB Equals 100. This value is set to be compatible with the other programs, DO NOT change it unless you change all programs. Certain information is stored in INDEX and INDEXT in row 100.

O Omega value for the image line of the source. Omega equals $nu/\mathcal{H}$, where n is the index of the space of the source, u is the marginal ray angle in the source space, and $\mathcal{H}$ is the Lagrange invariant.

OBAR Omega bar value for the image line of the source. Omega bar equals $n\bar{u}/\mathcal{H}$, where n is the index of the space of the source, $\bar{u}$ is the chief ray angle in the source space, and $\mathcal{H}$ is the Lagrange invariant.

OBC If the desired collector was not numbered in program two but there are lines in ALAYOUT from which the desired object can be made, then OBJC and OBC are the two line numbers in ALAYOUT containing the information. In particular a disk may be made by specifying a single line twice.

OBJC If the collector is specified in program two, OBJC is the collector number. If a new object is being made by existing lines in ALAYOUT, then OBJC is one of the two line numbers. OBC is the other.

OBJS If the source was specified in program two, OBJS is the source number. If a new object is made from existing lines in ALAYOUT, then OBJS is one of the line numbers.

OBS If the source was not numbered in program two, OBS is one of the two lines in ALAYOUT which form the new source.

OC1 If the data being read in are a "SOURCE" data card, OC1 is the OBJC value, otherwise OC1 should be read in as -0.

OC2 If the data being read in are a "SOURCE" data card, OC2 is the OBC value, otherwise OC2 should be read in as -0.

OMEGA(10,2) Stores the omega-omega bar information of the lines in the y-$\bar{y}$ diagram. $\Omega = nu/\mathcal{H}$ where n is the index of the

Program three
Program VIEWFAC
(Continued)

space, u is the marginal ray angle in this space, and $\mathcal{K}$ is the Lagrange invariant. $\bar{\Omega} = n\bar{u}/\mathcal{K}$, where $\bar{u}$ is the chief ray angle in this space.

OS1 If the data being read in are a "SOURCE" data card, then OS1 is the same as OBJS. If the card is not a "SOURCE" card, then OS1 becomes the value for HOLES(IH,1), which is the line in ALAYOUT of the obstructing disk or aperture or it is the cone object number causing the shadow.

OS2 If the data being read in are a "SOURCE" data card, then OS2 is the same as OBS. If the card is not a "SOURCE" card, then OS2 is the value for HOLES(IH,2), which codes whether the obstruction is a disk, aperture, or cone.

OX
OY
OZ See EX for description.

PCP The percent of power transferred from one section of the source to one section of the collector.

PHI The value of PI=3.14159265358979.

PRINTER Code word for the long or short form of printout. For the long version the code word is PRINTIT. For the power transfer from each section on the source to each section on the collector the following variables are printed: INANGLE, DCOSBRX, DCOSBRY, DCOSBRZ, BL, BM, BN, AL, AM, AN, DCOSBX, DCOSBY, DCOSBZ, DCOSSX, DCOSSY, DCOSSZ, XW, YW, ZW, XP, YP, ZP, XS, YS, ZS, DCOSIX, DCOSIY, DCOSBZI, OX, OY, OZ, VX, VY, VZ, EX, EY, EZ, WX, WY, and WZ. This is enough information to check out the calculations. If the code word is CHECKPO, the program will print out the information passed on by file 10 (BASICA).

PWRTRAN The percent of power transferred from one subsection of the source to one subsection of the collector.

QRBA The radius of the imaged baffle at the collector point.

QZP The distance of the plane containing the imaged collector point from the reference plane.

RBA The radius in real space at the collector point.

REFER(10,2) Contains the y and $\bar{y}$ values of the planes that are used as reference planes for each of the three spaces.

RLONGER A disk is subsectioned into sections of equal area. To do this there must be an inside radius (RSHORT) and an outside (RLONGER) radius.

Program three
Program VIEWFAC
(Continued)

RMAG The lateral magnification of the object in question.

RMIR The distance from the optical axis of the vector from the source point to the collector point as it passes through the first imaging aperture.

RMIRX The imaged radius of the last imaging aperture.

RSHORT A disk is subsectioned into sections of equal area. To do this it finds the appropriate inner and outer radius. The inner radius is RSHORT.

RSS The radius to the source point from the z axis.

RSS1 The radius of the source in the plane that has the lowest z coordinate as referenced in ALAYOUT.

RSS2 The radius of the source in the plane that has the largest z coordinate as referenced in ALAYOUT.

R1 The radius of the collector in the plane that has the lowest z coordinate as referenced in ALAYOUT.

R2 The radius of the collector in the plane that has the largest z coordinate as referenced in ALAYOUT.

SANG The "computed" angle out of the source.

SB The vector $\bar{Z}$ in "local" coordinates is positive as it runs in the positive z direction. SB is set so that when the appropriate cross products are made to determine the "local" system, $\bar{Z}$ will be in the correct direction.

SCP The sum of the SANG angles as the power transferred is made from subsection to subsection.

SEC The number of sections along the z axis on the collector.

SECJ The number of sections around the z axis. The default value is five. The circumference of the objects is divided into SECJ equal segments.

SECSSJ The number of subsections around the z axis. This number equals SECJ times DELTA.

SJ The current radial section of the collector.

SJLOW The number of the section in the six o'clock position of the source. See JLOW.

SJSUBJ The number of the radial subsections *within* the radial section of the source.

SLANTB The slant length of the section of the collector in real space.

Program three
Program VIEWFAC
(Continued)

SLANTL The slant length of the section of the imaged collector.

SLANTS The slant length of the section of the source.

SLONGER If the source is a disk, it is subsectioned into sections of equal area. To do this it calculates the appropriate inner and outer radius. The outer radius is SLONGER.

SLOPAB The slope in the y-z plane of the collector in real space.

SLOPABI The slope in the y-z plane of the imaged collector.

SLOPSS The slope in the y-z plane of the source.

SN The source number.

SNORMAL The codefor the direction of the normal to the surface of the source. If the source is a disk, a value of +1. for SNORMAL implies the source surface is the side of the disk toward the positive z direction. A value of -1. implies the other side. If the source is a cone, a value of +1. for SNORMAL implies the outside of the cone is the source, whereas a value of -1. implies the inside of the cone.

SPACE(100) The space that the object is in.

SS The adjustment factor so that the "local" coordinate Z axis always is in the direction of the plus z axis.

SSAREA The area of the source subsection.

SSCOS The cosine of the angle between the normal to the source and the vector from the source point to the imaged collector point.

SSHORT If the source is a disk, it is subsectioned into areas of equal area. To do this, it calculates the appropriate inner and outer radius. The inner radius is SSHORT.

SSJ The number of the radial subsection on the source.

SSJSUBJ The number of the subsection *within* the section of the source.

SY The y value at the plane containing the source point.

SYBAR The $\bar{y}$ value at the plane containing the source point.

S90 The sum of the angles ANGS090, for the power transfer from section to section.

TYPE The type of collector; a disk or a cone.

TYPEMIT The type of emitter (source): a disk or a cone.

Program three
<u>Program VIEWFAC</u>
(Continued)

VIEW If both the source and the collector are in the same space, the VIEW is real space. If the collector is not in the same space as the source, it will be reimaged and VIEW will be blank filled.

VV The real space distance used as a reference point to calculate the $y-\bar{y}$ values at the collector point in real space. This plane is used because the $y-\bar{y}$ values are known at this position.

VX
VY
VZ VX=ZX VY=ZY VZ=ZZ See EX.

WX
WY
WZ WX=FX WY=FY WZ=FZ See EX.

XMAX The number of sections along the z axis on the collector.

XMIR The x coordinate on the first imaging surface of the vector from the source to the collector.

XP
YP
ZP The (x,y,z) coordinates of the collector point in real space.

XS
YS
ZS The (x,y,z) coordinates of the source point.

XSS The number of axial sections on the source.

XW
YW
ZW The (x,y,z) coordinates of the collector point in image space.

XXX The angle in radians from the 12 o'clock position to the source point.

XXXX The angle in radians from the 12 o'clock position to the collector point.

YBAR1 The $\bar{y}$ value at the plane with the lowest z coordinate as defined in ALAYOUT for the collector.

YBAR2 The $\bar{y}$ value at the plane with the largest z coordinate as defined in ALAYOUT for the collector.

YBREF The $\bar{y}$ reference value for the space containing the source.

YMIR The y coordinate on the first imaging surface of the vector from the source to the imaged collector point.

Program three
<u>Program VIEWFAC</u>
(Continued)

YREF The y reference value for the space containing the source.

YS The coordinates of the source point are (XS,YS,ZS).

YP The coordinates of the collector point in real space are (XP,YP,ZP).

YW The coordinates of the collector point in image space are (XW,YW,ZW).

Y1 The y value at the plane with the lowest z coordinate as defined in ALAYOUT for the collector.

Y2 The y value at the plane with the largest z coordinate as defined in ALAYOUT for the collector.

ZB The z coordinate of the imaged collector plane that has has the largest z coordinate.

ZCRIT The z coordinate of the image of the first imaging aperture. The program checks that the radiation passes through this aperture.

ZF The z coordinate of the imaged collector plane that has the lowest value for its z coordinate.

ZHE The Lagrange invariant.

ZMIR The distance in real space of the last imaging surface. The vector $\bar{A}$ is created from this plane to the collector point in real space.

ZP The coordinates of the collector point in real space are (XP,YP,ZP).

ZPZERO The point on the z axis where the normal to the surface of the collector passes through the z axis.

ZS The coordinates of the source point are (XS,YS,ZS).

ZSS1 The z coordinate of the plane of the source (as defined in ALAYOUT) with the lowest z value.

ZSS2 The z coordinate of the plane of the source (as defined in ALAYOUT) with the largest z value.

ZSZERO The point along the z axis where the normal to the surface of the source passes through the z axis.

ZW The coordinates of the image of the collector point are (XW,YW,ZW).

Z1 The plane of the collector with the lowest z coordinate as defined in ALAYOUT.

Z2 The plane of the collector with the largest z coordinate as defined in ALAYOUT.

Program three

Subroutine SETAPT

IV — The line number in ALAYOUT of the last imaging aperture.

JV — The line number in ALAYOUT of the first imaging aperture for this source collector combination.

***** See VIEWFAC for other variables.

Subroutine BDRDF

ANGB180 — The cosine of the angle BANG, which is the "computed" angle into the collector.

ANGS180 — The cosine of the angle SANG, which is the "computed" angle out of the source.

CONVERS — Equals 180/PI. Converts radians to degrees.

FX
FY
FZ — See EX in VIEWFAC. Vector $\bar{F}$ is for the source.

FXB
FYB
FZB — Vector $\bar{F}$ for the collector. See EX in VIEWFAC.

X — Corrects the sign of the cosine values for the various angles.

ZX
ZY
ZZ — Vector $\bar{Z}$ for the source. See EX in VIEWFAC.

ZXB
ZYB
ZZB — Vector $\bar{Z}$ for the collector. See EX in VIEWFAC.

***** See VIEWFAC for the definition of other variables.

Subroutine VECTOR

***** See VIEWFAC for the definition of the variables.

Subroutine NORMAL

***** See calling routines.

Subroutine MIRRORS

PY — The y value for the plane in real space containing the object point.

PYBAR — The $\bar{y}$ value for the plane in real space containing the object point.

QC — The conversion factor from the y-$\bar{y}$ value of the object plane for the collector to the y-$\bar{y}$ values of its image plane along the imaging line. It is equal to the y value of the image point in its real space times the omega bar

Program three
Subroutine MIRRORS
(Continued)

value (nū/Ж) of the line onto which it is being projected minus the $\bar{y}$ value of the image point in its real space times the omega value (nu/Ж) of the line onto which it is being projected.

QY The y value of the plane containing the image of the collector point.

QYBAR The $\bar{y}$ value of the plane containing the image of the collector point.

RMAG The lateral magnification of the collector plane into its image.

***** See VIEWFAC for the definition of other variables.

Subroutine MIRRORA

PY The y value of the plane in real space containing the object point.

PYBAR The $\bar{y}$ value of the plane in real space containing the object point.

QC The conversion factor from the $y-\bar{y}$ value of the object plane for the collector in its real space to the $y-\bar{y}$ values of its image along the imaging line. See subroutine MIRRORS.

QY The y value of the plane containing the image of the collector point.

QYBAR The $\bar{y}$ value of the plane containing the image of the collector point.

RMAR The lateral magnification of the collector plane into its image.

Subroutine APERT

COSC The cosine of CRTANG. See below for CRTANG.

CRTANG If the obstruction is a cone (it could be the image of a cone), then there are two planes tangent to the cone and passing through the source point. Any vector passing between these planes through the cone will be obstructed. The critical angle (CRTANG) is the half angle these planes make where they intersect the x,y plane containing the source point (see Fig. 16).

C1 The radius in real space of the projection of the shadowing cone onto the y-x plane.

I The line number in HOLES.

Program three
Subroutine APERT
(Continued)

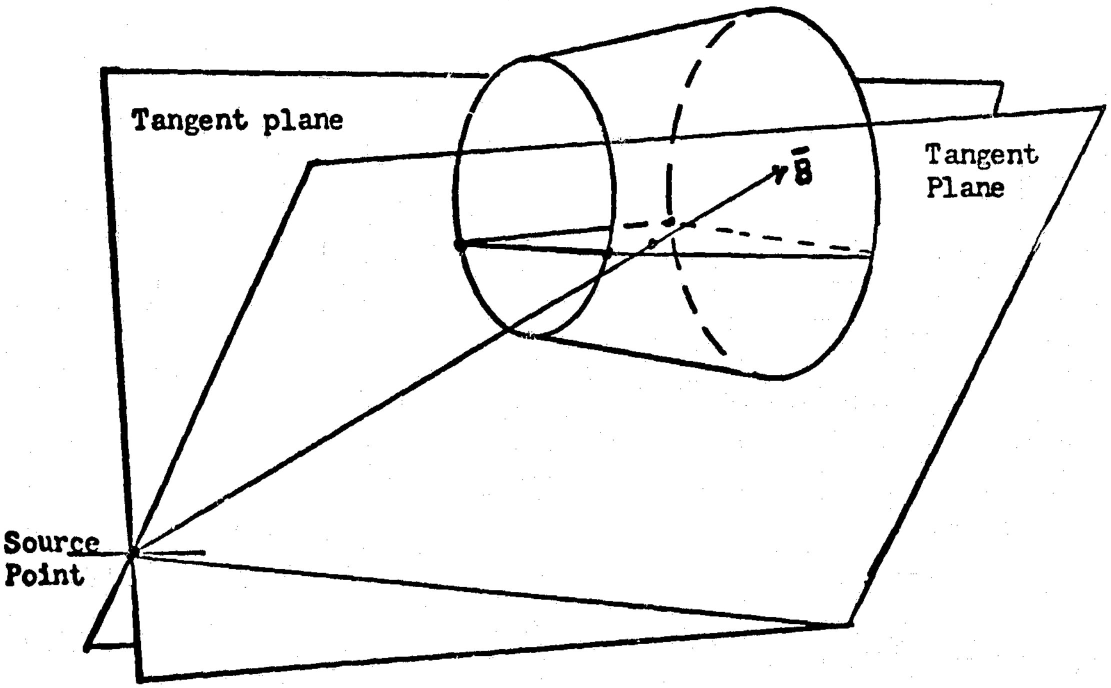


Fig. 16. Obstruction by a cone-shaped baffle.

ILINEA — The line in ALAYOUT containing the information about the shadowing object when the shadow is a disk or an aperture.

IOBJH
IOBJHH — If the shadow is caused by a cone then there are two lines in ALAYOUT that contain all the necessary information. IOBJH is the first line appearing in ALAYOUT, and IOBJHH is the line number of the second time it appears.

ISPA
ISPSH — The space of the shadowing object. The image of the object is projected as seen from the source if the source and shadow are not in the same space.

OBJH — The object number of the cone causing the shadow.

RBAH — The real height of the obscuring aperture or disk.

RHOLE — The radius from the z axis as the vector from the source point to the collector passes through the plane of the obscuring disk, aperture, or the plane of the end of the shadowing cone.

RHOLE1 — The radius from the z axis as the vector from the source point to the collector point passes through one of the ends of the shadowing cone. The radius as it passes through the other end is RHOLE.

RH1
RH2 The radii in real space of the shadowing cone.

R1ATZ The radius of the shadowing cone as it passes through the x-y plane containing the source point. It is in this plane that CRTANG is calculated.

SINC The sine of CRTANG.

SLOPE1 The slope in the y-z plane of the shadowing cone.

T The shadowing cone is described by two z coordinates and two radii. The two planes tangent to the cone and passing through the source point are tangent also to the two radii. The set of four points (each plane tangent to two circles) is coplanar. The vector from the source to the collector point is $\bar{B}$. T is the distance from the source point along $\bar{B}$ to the point where $\bar{B}$ passes through the plane containing the above four points.

TYPEH The type of shadow: a hole or a disk.

X The x coordinate of the $\bar{B}$ vector as it passes through the plane containing the four points. See T above.

XCB The two positive x coordinates of the four coplanar points mentioned in T above.

XCK The x coordinate of the point on the cone and in the plane of the four coplanar points at the z coordinate of the $\bar{B}$ vector as it passes through the plane containing the four points. If X is less than XCK at this value of z, the line of sight is through the cone and is therefore obscured.

XHOLE The x coordinate of the $\bar{B}$ as it passes through the plane of the obscuring disk, hole, or either end of the cone.

YBH The $\bar{y}$ value fcr the plane of the shadow being imaged.

YCF
YCB The two y coordinates of the points that have the positive x coordinates XCB and XCF.

YH The y value for the plane of the shadow being imaged.

YHOLE The y coordinate of the $\bar{B}$ vector as it passes through the plane of the obscuring disk, hole, or either end of the cone.

Program three
Subroutine APERT
(Continued)

Z The z coordinate as the $\bar{B}$ passes through the plane of the four coplanar points. See T above.

ZPH
ZPH2 The z coordinates of the two ends of the cones after they are imaged if that was necessary.

***** See VIEWFAC for the definition of the other variables.

D. Program Four

There are too few variables in program four to be listed here. See p. 61 for more information.

E. Program Five

Program DRIVER

A(980) A dummy array used to rearrange certain arrays taken from or being put into mass storage files.

AFAC A factor that will adjust the radiance distribution of the source to that measured in the lab. It has been found that even for normal incidence the blacks are not Lambertian.

ANG If the load program does not supply the array of angles into each section of the surfaces receiving radiation from the source, the program will allow the value to be read in or default to 90. degrees.

ANGI The particular "calculated" angle into the collector section being referenced. One of the values in the array ANGINS.

ANGINS(42,14) The array of angles into the present emitter from one of its previous sources.

ANGO The particular "calculated" angle out of the source section that is being referenced. One of the values in the array ANGOS.

ANGOS(42,14) The array of angles out of the present emitter to the collector.

ANGV The angle of the vanes (if this section has any) of the section of the source being referenced.

ARHO The apparent reflectivity of the source section for the angle at which the power came into it, and for the angle at which it is being emitted toward the collector section. This value may be greater than RHO, the reflectivity of the type of coating applied to the surfaces because the locus of the vane tips was used to calculate the percent of power transferred. The cosine of the angle between the normal to the surface and the direction toward the collector was used to determine the projected area seen by the collector. The angle of the vanes, the angle at which the power falls on the vanes (not the locus of vane tips), and the angle from the normal to the vanes at which the radiation goes off the vanes is taken into account by the program.

BDR The percent of power transferred from the referenced source section to the referenced collector section.

CONEB(42,14) The array containing all the BDR values for all the sections of the source to all the sections of the collector.

Program five
Program DRIVER
(Continued)

CONE1(70,14) The array containing the power on each section of the source. The power on the source is stored in a very precise manner, and any new loader program must arrange the loaded radiation accordingly. The manner was chosen because all the separate powers must be stored, and for certain operations the present arrangement is faster and has fewer IO calls. An example will help to show how the information is stored. If the source has two axial sections and has received radiation from an object that had three axial sections, the power on the source will be in five arrays that are 10 by 3.

The source for this collector has three axial sections and five radial sections. Hence this array 5x3. See note A.

The information is stored in five arrays of this size. Each array is for one complete axial section (the entire length of the source with the number of the radial section fixed). Since there are five radial sections, there are five of these arrays.

```
Power on space 1,1 of the collector.
 .21E-06    .24E-06    .28E-06
 .48E-06    .54E-06    .56E-06
 .44E-07    .98E-06    .66E-06
 .47E-06    .78E-06    .54E-06
 .24E-06    .29E-06    .28E-06

Power on space 1,2 of the collector.
 .77E-06    .12E-05    .89E-05
 .45E-06    .24E-05    .77E-05
 .56E-06    .44E-06    .58E-05
 .35E-06    .69E-06    .64E-05
 .68E-06    .45E-05    .75E-05

Power on space 2,1 of the collector.

 .22E-06    ....       .....
 .....
 .....
```

Note A: For the example shown the 10 by 3 array has really stored two 5 by 3 arrays. The source is three axial sections long and five axial sections around. The numbers in the array represent the amount of power contributed to this section on the collector. In the case shown the first value of .21E-06 is the amount of power transferred from section 1,1 on the source to section 1,1 on the collector. Going across, the value of .24E-06 is the amount of power transferred from section 1,2 of the source to section 1,1 on the collector. All the power on the collector section 1,1 is the sum of the values in the inner box. It is in this manner that the program can tell the direction from which all the power to a given collector section came.

CONE2(70,14) The power received on the collector is stored in the same array form as described for CONE1. This is necessary because at the next level of scatter, this collector might be a source.

COSV The cosine of the vane angle. The vane angle is the angle between the "local" minus z direction of the locus of vane tips and the vane on the source.

DISTV The distance between vanes at this section of the source.

DLENTH The slant length of this section of the source. There may be several vanes within one section on the source.

FL The length of the source (within the vane structure) being considered divided by the arc length of this section of the source.

FN The length of the collector (within the vane structure) being considered divided by the arc length of the section of the source.

HERE The current block of data from the file ANGIN. If HERE equals IHERE2, the array is already in central memory.

IBDRL
IBDRR (IBDRR,IBDRL) determines the section on the source object that is the present emitter.

IC The number of the collector object.

ICOL The column in ITAB2 (the list of collectors) for a potential collector at this level of scatter.

IDOWN The row in CONE2 to store the power transferred. The column number is IBDRL.

IE The emitter under consideration.

IEE The emitter for the various possible collectors. A check is made against IE. If IE and IEE are equal, then this is considered an acceptable source.

IEMIT(100) The array of emitters read in as data. The numbers correspond to the sequence of numbers run in program three but need not be identical. In program three each object was given a different object number for each space. In this program all these objects should usually have a common number so that the power on each object of the output will be the total power from the various possible paths. Not all numbers need be included in this array.

Program five
Program DRIVER

IEMITER The row number in LOOK4 as a search is made for emitters IE.

IFIND The row number in LOOK4 as a search is made for collectors.

IFOUND The object number in row IFIND and column LEVEL+1 in LOOK4. If IFOUND equals IC then this is an acceptable collector for source IE.

IHERE1 The last block of data called from file BDRDF. If NBLOCK5 equals IHERE1, the IO call need not be made as the information is already properly stored in central memory.

IHERE2 The last block of data called from file ANGIN. If HERE equals IHERE2, the IO call need not be made as the information is already properly stored in central memory.

IHOLD Holds the number of sections in KEY up to the emitter object. The program needs this information to find arrays in storage.

IK Equivalanced to INKEY. The number of objects in KEY.

IM The object that is considered the final collector. Usually this will be the image plane, but it could be any surface of interest. The program automatically prints out all the power put onto this object.

IMGSEC The number of sections on the object that has some portion of it sliced away. It may be the sunshield that is cut away at say 45°.

INDEXT(20)
INDEX1(201)
INDEX2(201)
INDEX3(100)
INDEX4(100)
INDEX5(100) The set of arrays used by the mass storage routines of the programs. They must be one word larger than the number of blocks taht will be stored. The information that is put in these arrays locates the desired blocks of data stored randomly on the disk.

INKEY The number of words in KEY. The number of objects considered for this run.

IOUT The axial section of the collector.

IPASS(40,2) The array that stores the collectors from ITAB2 that received radiation at this level. This array is used by subsequent programs to access any specified array to aid in a complete trace, step by step, of the radiation through the program.

ISBLOCK The block number into which the array A (CONE2: the power on one axial section of the collector) is stored.

IS The object number that has a portion of it sliced away.

Program five
Program DRIVER
(Continued)

ISUM — Equals the sum of IHOLD and IBDRL. This sum yields the line number in VANINFO where the characteristics of this section of the source are described.

ITAB(100) — The number of words in the Ith block of BDRDF, ANGIN, and ANGOUT.

ITAB2(100) — An array containing the sequence of collector numbers corresponding to the source numbers in IEMIT.

IX — The power on the present emitter came from object IX. The program will now find all the parcels of power (and the angle at which they came) on each section of the present emitter to determine how much of this power is transferred to the collector.

IX3 — Three times the number of axial sections of object IX.

I3 — The number of axial sections on object IX.

JA(5,5) — The symmetry relations of the power transfer operations.

JB — The integer in the array JA(I,J). The symmetry relation to be used for this source section and collector section combination.

JJJ — The row in CONEB containing the percent of power transferred from the source section to the collector section; the column is specified by IOUT. Also the row in ANGOS (column IOUT) containing the angle out of this source section to the collector section.

JK — Like JB, but now it describes the symmetry relation to determine the angle at which the power came into this emitter section.

JOUT — The radial section on the collector.

JQQ — The number of sections on the surface that has been loaded with radiation initially.

KC — The number of axial sections on the collector.

KC5 — Five times the number of axial sections on the collector. This number equals the total number of sections on the collector.

KE — The number of axial sections on the emitter.

KEY(40) — Stores the number of axial sections for all objects considered.

KE3 — Three times the number of axial sections on the emitter.

KE5 — Five times the number of axial sections on the emitter. Equals the total number of sections on the emitter.

Program five
Program DRIVER
(Continued)

KK — CONE1(KK,LL) is the power on the source section.

KK1 — JA(IBDRR,KK1) is the symmetry relation to use to determine the ANGI (angle into the source).

KPTL
KPTF — Determines the set of lines (inclusive) that contain the amounts of power on this section of the source. See CONE1, and the inner box is between lines 1=KPTL and line 5=KPTF.

KQQ — Three times the number of axial sections on the initially loaded surface.

KZ — The total number of axial sections of all the objects. The sum of the values stored in KEY.

L — The level of scatter after which when completed the program will terminate. Provisions are made for five levels of scatter. Usually, once radiation reaches the image plane, successive levels of radiation do not significantly increase the final power on the image plane.

LEVEL — The level of scattered radiation being considered.

LIMIT — The limit of the possible number of sources at this level of scatter. The program searches ITAB2 through LIMIT number of words trying to find sources that have received power.

LIMITP — The program will search through LIMITP words in the array ITAB2 looking for a collector at this level of scatter. This speeds program execution by not accessing arrays not yet filled.

LIMITS(5) — For the given level of scattered radiation (LEVEL) the program will search through ITAB2 for suitable emitters and collectors. A collector at one level is a possible emitter at the next level if that path of radiation wants to be continued.

LL — The axial section of object IX (that object that put power on the present emitter section), which determines the fraction of power on the present emitter section and the angle that it came in.

LOOK4(20,4) — In column LEVEL are the possible emitters at this level of scattered light. In column LEVEL plus one are the possible collectors at this level. For the next level of scatter the collectors are allowable emitters. The program has the capability of running through to a specified level of scattered radiation, stopping,

Program five
Program DRIVER
(Continued)

cataloging the present files, and then restarting at the level at which it stopped, or one level above or below. Therefore, there is the possibility that certain objects will be collectors at one level but not emitters at the next for a particular run, but the potential is there to rerun the program from this level and consider them as emitters and continue to the next higher level.

LUM The two large files created by this program are on logical unit numbers two and three. IPASS(ICOL,LUM) stores the blocks loaded on each logical unit number (LUN). When LUN is two, LUM is one, the first column of information in IPASS. When LUN is three, LUM is two, the second column in IPASS.

LUN The logical unit number of the file being accessed, either two or three. One file is the power on the objects considered as sources at this level while the other file stores the amount of power received by the collectors. At the next level the collectors become the sources and the previous source file will be written over by the new set of collectors.

MAXNO The number of words (collectors in ITAB2, emitters in IEMIT) read into the arrays ITAB2 and IEMIT.

NBLOCK5 The block where the power on the present emitter is stored. This array is put into CONE1.

NTAB The size of the index stored for files five (BDRDF), eight (ANGIN), and nine (ANGOUT). NTAB must be less than or equal to the size of the arrays INDEX3, INDEX4, and INDEX5.

NWORDS The number of words stored in block NBLOCK5.

R The radius of the section of the source.

RAD The edge fit factor for determining the amount of scatter contributed by the edge.

RHO The totaldiffuse reflectivity for the sections of the objects.

SINV The sine of the angle of the vanes.

SLICE(5,20) The program handles only rotationally symmetric objects. If one surface is a cone or cylindrical surface that has a segment sliced away, it can still be considered rotationally symmetric. The percent of area of each section of the object that has not been removed is entered in the array SLICE. Then the power received

Program five
Program DRIVER
(Continued)

by the sections is multiplied by the appropriate percent of area remaining. If no area remains that section will have zero power on it and therefore not transfer any power.

VANINFO(100,6) Stores the characteristics of each section in the system. Column one is the angle of the vanes (ANGV). Column two is the average distance between the vanes (DISTV). Column three is the average vertical height of the vanes (VH). Column four is the reflectivity of the section (RHO). Column five is the length of the section (DLENTH). Column six is the radius of the object (R).

VH The average vertical height of the vanes at each section.

WATTS(5,80) The total power on each section in the system considered. All the increments of power on all the sections (regardless of angular information) are summed to get this total power. This is a running total and includes all power from previous levels.

Subroutine ARANGER

B(42,14) The array A (which can be the information for array CONEB, ANGOS, or ANGINS) is rearranged into array B and passed back to the main program (DRIVER).

I The number of rows in the final array B passed back to DRIVER.

J The number of columns in array B as it is passed back to DRIVER.

IA The word in the array A being accessed.

***** See BAFFLE for the definition of other variables.

Subroutine ARANGE

B(70,14) The information taken from the mass storage file and put into array is rearranged properly for CONE1. The power on the sections of the system.

I The number or rows containing pertinent information in array B.

IA The word in A being accessed.

J The number of pertinent columns being passed back in array B.

***** See BAFFLE for the definition of the other variables.

Program five
Subroutine WHERE
(Continued)

I The number of the object being processed.

IDIDC The row in LOOK4 being checked for the collector number. If the collector I is not found, then no radiation was traced to it at this level. It may have recieved radiation at this level but it was desired by the user that this path not be considered.

IFST From word IFST to ISEC in the array A are the powers that must be attenuated by the appropriate value in SLICE.

II
III The arrays containing the parcels of energy on a specified section (see CONE1 in BAFFLE; the inner box represents such an array) are summed. There are III inner arrays that are five by KE, the number of sections that emitted the energy to this collector. III are the number of sections on the collector. The III inner arrays make up one block of data stored on disk. The outer box described by CONE1 in BAFFLE. II varies from one to five for the five blocks that represent all the power transferred from all the sections on the source to all the sections on the collector.

ISIZE The size (number of words) of the block being accessed.

J The location in ITAB2 as a search is made to find collector number I.

JJJ The radial section of the collector.

JS The row in LOOK4 as a search is made for a source to this collector.

JWATT1
JWATT2 The collector I has the power on its sections stored on lines JWATT1 to JWATT2, inclusive, in the array WATTS.

KC The number of axial sections on the object I, the collector.

KE The number of axial sections on the emitter to object I.

KE5 The total number of sections on the emitter to object I. This is five times KE.

NBLOCK The block in mass storage being accessed for power on collector I.

NONE If no power is transmitted to object I, NONE will be true, and a message to that effect will be printed out. In the system power may in fact be transmitted to object I, but in this run this path was not desired.

***** See BAFFLE for the definition of the other variables.

Program five
Subroutine SHOWIT
(Continued)

I — The section on the collector. This is the special collector for which all powers are printed out. Usually this will be the image plane.

II — The radial section on the collector.

IMGSECT — The number of axial sections on the special collector IM.

NL — The number of axial sections on the emitter to the special collector IM.

***** See BAFFLE for the definition of other variables.

Subroutine SURFACE

A — The majoraxis of the ellipse.

ANGW — PI-ANGV, where PI is 3.14159265358979 and is called PHI in the program.

ANG90 — 90 degrees in radians, PI/2.

ARCL — The arc length of this section of the source.

A2 — Equals A^2.

B — The minor axis of the ellipse.

B2 — Equals B^2.

C — The power transfer within the vane structure is being calculated as the sum and differences of certain parallel vane sections. C is the slant length of one of these sections.

CON — Converts the input angles from degrees to radians.

COST — The cosine of the angle TIN, which is the adjusted specular angle out. It has been found that there is significant difference between the angle of incidence and the specular angle out for surfaces like Martin Black.

COSW — The cosine of the angle ANGW (which equals PI-ANGV, the angle of the vanes).

CT — When solving for the value x, in the elliptical equation $(x/a)^2 + (y/b)^2 = 1$, where y is known in terms of x, there is the square root of a set of terms. CT is the value of this set of terms.

CZ — The y value as the radiation is traced off the surface on the rotated ellipse at the value of x equal to zero.

C1 — If the section of the source has vane structure on it, C1 is the length depicted in Fig. 17.

Program five
Subroutine SURFACE
(Continued)

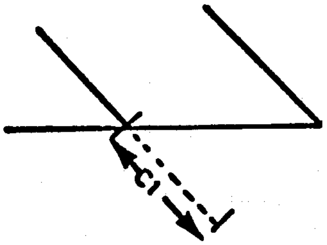


Fig. 17. Length of C1.

If the surface section has no vanes, C1 equals the expression $1/A^2 + (\tan\phi/B)^2$, where A is the major axis and B is the minor axis, and ϕ is the angle of the line through the rotated ellipse.

C2 — If the section has vane structure, C2 is the length of the vane structure receiving radiation directly from the source. If the section has no vanes, C2 equals the expression $2C \tan\phi/B^2$, where C is the y intercept and ϕ is the angle of the line through the rotated ellipse.

C3 — If the section has vane structure on it, C3 is that section of the vane structure not seen by the collector. The slant vane length minus C3 is the section that is seen. If the section has no vanes, C3 equals the expression $(C/B)^2 - 1$, where C is the y intercept of the line through the rotated ellipse, and B is the minor axis of the ellipse.

C4 — If the section has vane structure, C4 is the length of the vane structure not receiving direct radiation, vane slant length minus C2. If the section has no vanes, C4 is the square root of CT.

EDGE — In addition to the scattered radiation from the internal vane structure there is the scatter from the vane edges. EDGE is that fraction added to the apparent reflectivity of the whole section contributed by the edge. The program can be run with no scattered radiation from the edges to determine if this is a significant propagator of the energy to the image plane.

EDGED — The percent of edge compared to the length of the section times the reflectivity.

Program five
Subroutine SURFACE
(Continued)

F
For a surface without vanes. It is the angle of incidence from the normal divided by PI/2. Used to scale the major axis.

FBOT
FP
FT
FVANE
F1
F12
F123
F13
F14
F15
F16
F2
F23
F4
A set of variables that are the "configuration factor" for the sections involved. The sum and difference geometry of sections involved determines the final "configuration factor" between the surfaces receiving radiation directly and those seen by the collector.

HD
HN
When only a fraction of the vane receiving radiation directly is seen directly by the collector, HD is the length of the section receiving the direct radiation. HN is the length of the section seen directly by the collector.

PB
PV
The percent of radiation striking the side wall of the baffle structure: the wall of the tube between the vanes. PV is the percent of radiation striking the vanes.

PCENTH
The percent of the surface receiving radiation directly that is seen from the collector, or a "configuration factor" like FBOT, etc.

PCENTL
1 - PCENTH, or a "configuration factor" like FBOT, etc.

PHIOUT
The adjusted specular angle out of a surface.

PHI2
PHI/2, where PHI=PI=3.14159265358979

PV
See PB.

PX
Because the program is designed to use the locus of vane tips as the surface to calculate the percent of power transferred from any section on a source to any section on the collector, an adjustment must be made to the RHO value (reflectivity) because surfaces within the vane structure are seen at a different angle from its normal; PX is this correction factor.

S
1.E-10 a small value used to check for round-off errors.

SEP
The separation between the vanes along the normal to their surface.

Program five
Subroutine SURFACE
(Continued)

SINI The sine of the angle into the surface.

SINT The sine of the angle TIN, the adjusted specular angle out.

SINT2 Equals $(\mathrm{SINT})^2$.

SQ Equals the expression: $(B^2 \cos^2\phi - A^2 \sin^2\phi)^{\frac{1}{2}}$, where A is the major and B the minor axes of the ellipse, and ϕ is the angle of the line cutting the rotated ellipse.

TANI Tangent of the angle into the surface.

TANI2 The tangent of the angle equal to the angle in -90°.

TANO The tangent of the angle out of the surface.

TANO2 The tangent of the angle equal to the angle out -90°.

TANP The tangent of the adjusted angle out, PHIOUT.

TANV The tangent of the angle of the vanes.

TAN90 The tangent of 90° as stored by the library routine.

TIN The adjustment value for the off-specular reflectance.

VHS The vertical distance from the surface described by the locus of the vane tips down to the point where radiation strikes the vane.

VL The slant length of the vanes.

X The length of the section of the side wall to which or from which radiation will be calculated.

XB The point along the side wall where the radiation passing over one of the vanes would then strike the side wall, regardless of the second vane.

XF
YF
If the radiation strikes the back side of the front vane before striking the side wall, the x coordinate is XF. The y coordinate is YF.

XFNT The x coordinate of the top of the front vane tip.

XFS The x coordinate of the point where the collector section sees into the vane structure the deepest.

XG,YG
XGS,YGS
The x coordinate of the incoming or outgoing ray as it strikes or leaves the front side of the rear vane.

XHS The x coordinate of the out-going ray at its lowest point or either vane surface.

XO
XP
If a ray were to be traced from the collector to this vane structure, such that it would just pass over the first vane it encountered and then continue until it

Program five
Subroutine SURFACE
(Continued)

struck the side wall, then the x coordinate on the side wall would be XO. If the same were done for the incoming ray, the point would have the x coordinate XP as shown in Fig. 18.

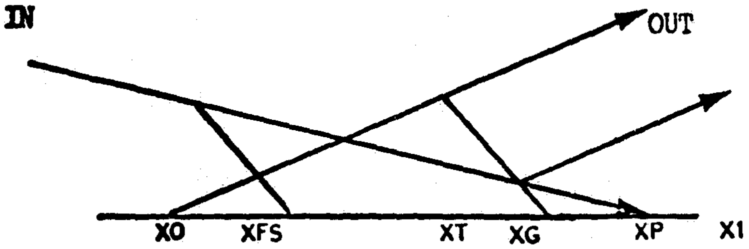


Fig. 18. Significant variables in vane structure analysis.

XPT
YPT — XPT and YPT are the x and y points where the line passes through the ellipse as the scatter point.

XPT2 — Equals XPT^2

XT — The x coordinate of the tip of the rear vane.

X1 — From the point where the incoming ray strikes a surface (vane or side wall) a ray is projected out from this point at the angle out. The x coordinate as it passes through the surface of the locus of vane tips is the point X1.

X1,Y1
X2,Y2 — If the surface has no vane structure, these are the two points where the line passes through the ellipse.

YF — See XF.

YG
YGS — See XG.

YP — Equals -SQ.

YPT — See XPT.

***** For the definition of the other variables see BAFFLE.

Subroutine FFACTOR

FFFF — Stores the intermediate sums while the numerical integration is done.

Program five
Subroutine SURFACE
(Continued)

F12 The value of the function being integrated at its various points of integration.

I The Ith integration point.

PI 3.14159265358979

S 1.E-10

SINV2 SINV^2, the sine of the vane angle squared.

SP
SPP The number of integrations to be done along the function.

S1
S2 The number of integration points.

VROOT
W
XNZ The result of intermediate expressions that reoccur in the numerical integration expression.

XI Equals I+1

XL12(12) The array that stores the values at the S1 integration points.

XNZ See VROOT.

XN2 The length of the collector section considered (XN) squared.

Z10 The length of the source section divided by the number of integrations to be done.

Z2 Equals Z^2, the length of the source section squared.

***** See BAFFLE for the definition of the other terms.

Subroutine GFACTOR

X
Y The length and width of the sections transferring power.

***** See SURFACE for the definition of the other variables.