X-601-71-166
PREPRINT

NASA TM X- 65553

THE NORMALIZATION OF SOLAR X-RAY DATA FROM MANY EXPERIMENTS

CHARLES D. WENDE

MAY 1971

JUN 1971
RECEIVED
NASA STI FACILITY
INPUT BRANCH

GSFC — GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

FACILITY FORM 602
N71-27644 (ACCESSION NUMBER) (THRU)
86 (PAGES) G-3 (CODE)
TMX-65553 (NASA CR OR TMX OR AD NUMBER) 29 (CATEGORY)

X-601-71-166

THE NORMALIZATION OF SOLAR X-RAY DATA FROM MANY EXPERIMENTS

Charles D. Wende

May 1971

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

ACKNOWLEDGMENTS

The experimenters J. A. Van Allen (Explorers 33 and 35 and Mariner 5), R. W. Kreplin (OGO 4), and R. G. Teske (OSO 3) are to be thanked for their frank and helpful discussions and for making unpublished data available. This analysis of detector characteristics was based on window areas, window and gas thicknesses, and window and gas constitutions supplied by them. Dr. J. I. Vette and L. R. Davis are to be thanked for their help and encouragement and for making the facilities available for doing this work.

CONTENTS

ILLUSTRATIONS

PRECEDING PAGE BLANK NOT FILMED

ILLUSTRATIONS (continued)

ILLUSTRATIONS (continued)

ILLUSTRATIONS (continued)

ABSTRACT

A conversion factor is used to convert Geiger (GM) tube count rates or ion chamber currents into units of the incident X-ray energy flux in a specified passband. A method is described which varies the passband to optimize these conversion factors such that they are relatively independent of the spectrum of the incident photons. This method was applied to GM tubes flown on Explorers 33 and 35 and Mariner 5 and to ion chambers flown on OSO 3 and OGO 4. Revised conversion factors and passbands are presented, and the resulting absolute solar X-ray fluxes based on these are shown to improve the agreement between the various experiments. Calculations have shown that, although the GM tubes on Explorer 33 viewed the sun off-axis, the effective passband did not change appreciably, and the simple normalization of the count rates to the count rates of a similar GM tube on Explorer 35 was justified. It is also shown that for the type 6213 GM tube, the passband is effectively 3 to 12 A although gas pressure and window thickness variations may give rise to numerical conversion factor differences of a factor of 6. The programs used in these calculations are given in the appendixes.

PRECEDING PAGE BLANK-NOT FILMED

SECTION 1. APPROACH TO THE PROBLEM

Geiger (GM) tubes and ion chambers have been used to measure the broadband X-ray flux from the sun. However, in using these devices it is necessary to convert the instrumental output from GM tube count rates or ion chamber currents into scientific units in the form of incident energy fluxes. Usually, the instrumental output is multiplied by a conversion factor to effect this change. Because of different approaches to solving the problem of determining the conversion factor, data appearing in the literature are not always consistent.

One approach to finding the conversion factor has been to assume both a spectral shape for the incident spectrum and a passband for the detector and then to calculate the conversion factor using these assumptions (e.g., Kreplin, 1961). But the conversion factor/passband combination so chosen may be very sensitive to the shape of the input spectrum, which in turn is not measured by the detector and is known to vary.

In another approach, developed by Van Allen (1967), a number of input spectra are chosen and the passband limits are varied until limits are found which minimize the spectral dependence of the conversion factor. This approach, which minimizes the dependence of the resulting data upon a quantity which cannot be measured by the instrument, was chosen for further development.

The efficiency of a GM tube or ion chamber, ε, equals the probability that a photon of wavelength λ will be transmitted through the window and absorbed by the filler gas. The efficiency is determined theoretically by the product of the window transmission, $T_w(\lambda)$, and the absorption of the filler gas, $A_g(\lambda)$. For a nonreflecting gas, the sum of the gas transmission, $T_g(\lambda)$, and the absorption, $A_g(\lambda)$, is unity. Thus,

$$\varepsilon(\lambda) = T_w(\lambda) \left[1 - T_g(\lambda)\right] \qquad (1)$$

where

$T(\lambda)$ = exp - $(\mu/\rho)\rho x$,
(μ/ρ) = the mass absorption coefficient, and
ρx = the thickness of the window, w, or the gas, g, in g cm^{-2}.

The mass absorption coefficients used in this study were taken from Henke and Elgin (1970) and McMaster et al. (1969) and were interpolated where necessary assuming a power law relationship between the mass absorption coefficient and the wavelength. (Appendix A briefly describes the programming language used, APL on the IBM system 360; Appendix B describes the program.)

The conversion factor, γ, is defined as the ratio of the energy flux within a specified passband to either the GM tube count rate or the ion chamber current. The energy flux is the integral, between specified limits (e.g., a and b), of the differential energy flux, $dE/d\lambda$. The GM tube count rate equals the window area, A, times the integral from zero to infinity of the differential photon flux, $dN/d\lambda$, times the probability that a photon is absorbed, ε. The conversion factor, γ, resulting for a GM tube is given by

$$\gamma_G = \frac{\int_a^b (dE/d\lambda)\, d\lambda}{A \int_0^\infty (dN/d\lambda) \cdot \varepsilon \cdot d\lambda} \tag{2}$$

The ion chamber current equals the window area, A, times the gain of the gas, G, which is the number of ion-electron pairs formed per erg of absorbed energy, times the charge of an electron, e, times the energy per absorbed photon, hc/λ, times the integral from zero to infinity of the differential photon flux $dN/d\lambda$, times the probability of absorption ε. The conversion factor, γ, resulting for an ion chamber is given by

$$\gamma_I = \frac{\int_a^b (dE/d\lambda)\, d\lambda}{A \cdot G \cdot E \int_0^\infty (hc/\lambda)\, (dN/d\lambda) \cdot \varepsilon \cdot d\lambda} \tag{3}$$

Physically, the differential energy flux is equal to the product of the differential photon flux and the energy of each photon, or

$$dE/d\lambda = (hc/\lambda)\, (dN/d\lambda) \tag{4}$$

Applying equation (4), equations (2) and (3) can be cast entirely in terms of either the differential photon flux $dN/d\lambda$ or the differential energy flux $dE/d\lambda$. The conversion factor γ is a function of the

shape of the spectrum used and not of the magnitude of the spectrum since constant factors cancel out. The difference in spectral response between using a tube as a GM tube or as an ion chamber appears as a factor of $1/\lambda$ in the integral in the denominator since equation (3) is equivalent to equation (2) if the efficiency in equation (2) is replaced by the product of the efficiency, the photon energy hc/λ, the gas gain G, and the electronic charge e.

A computer program has been written which uses the Van Allen criterion (i.e., spectral independence of the conversion factor) to determine the best conversion factor/passband combination for a given detector. This program is described in Appendix C. The input spectra chosen are free-free and blackbody spectra at nominal temperatures of 1-, 2-, 5-, 10-, 20-, 50-, and 100-million °K. When compared to typical solar spectra, these spectra vary from those which are too soft through those which are too hard. Two flat spectra, $dE/d\nu = 1$ and $dE/d\lambda = 1$, are used as controls. Given the efficiency of the tube, the window area of the tube, the gas gain if applicable, and the estimated limits, the program calculates 16 conversion factors, γ's (one for each spectrum), their mean, the standard deviation, σ, and the relative error, σ/mean. The program varies the limits and finds the passband which gives the minimum relative error. Integrals are calculated by summing in 0.25-A steps; the denominator is integrated from 0.25 to 20 A.

While the spectra used are not ideal models of the solar spectrum, it is expected that the conversion factor/passband combinations determined through their use will not vary significantly from those obtained using actual solar spectra. The nominal uncertainty in the passbands is ±0.25 A, and the relative error is given for each conversion factor. The effect of using these conversion factor/passband combinations on actual data from different experiments is shown in the next section.

Two further questions need to be asked, particularly in the case of the University of Iowa experiments flown on Explorers 33 and 35. These experiments used the same type of GM tube (EON 6213), but none was accurately calibrated for soft X-ray use prior to launch. All of the data were normalized to the data from one GM tube on Explorer 33. Although these GM tubes viewed the sun off-axis, the data were again normalized using empirically determined angular correction factors. The validity of these procedures is dependent upon the passband of the GM tubes remaining relatively constant regardless of the known window thickness and gas pressure variations and regardless of the angle between the sun and the axis of the GM tube.

The program explained in Appendix D has been developed to determine the effects of window thickness and gas pressure variations. In this program, a passband is assumed and conversion factors and their corresponding relative errors are calculated. The results of using this program are discussed in Section 2.

The program explained in Appendix E has been developed to determine the effect of viewing a source off-axis. In this program, the geometry of the aperture covering the GM tube is used for input, and the relative efficiency and area are calculated as a function of the off-axis angle. From these variables, the relative off-axis response is calculated. Again, the program validates the procedures used in the reduction of Explorer 33 and Explorer 35 data.

SECTION 2. APPLICATIONS

Data from six detectors on four spacecraft were used in this study. The detectors were the GM tubes flown on Explorer 35 and Mariner 5 which were used to observe solar X rays (University of Iowa experiments; J. A. Van Allen, Principal Investigator), the ion chamber experiment flown on OSO 3 (University of Michigan experiment; Prof. R. Teske, Principal Investigator), and three ion chambers flown on OGO 4 (Naval Research Laboratory experiment; R. Kreplin, Principal Investigator). Although data from Explorer 33 were not used in this study, the GM tubes it carried were the same type as those flown on Explorer 35. All of these detectors measured the broadband solar X-ray flux in different passbands.

First, the optimal passband/conversion factor combinations were calculated. These are listed in Table 1. Applying the revised passband/conversion factor combinations yields the following changes.

- The Explorer 35 fluxes should be multiplied by a factor of 1.37 and the passband decreased to 2.75 to 12 A.
- The Mariner 5 fluxes should be multiplied by a factor of 1.85 and the passband increased to 2 to 9.75 A.
- The OSO 3 flux values should be multiplied by a factor of 2.5 and the passband increased to 1.5 to 13 A.
- The OGO 4 8- to 20-A fluxes should be multiplied by a factor of 1.05 and the passband shifted to 2 to 14 A.
- The OGO 4 1- to 8-A fluxes should be multiplied by a factor of 0.28 and the passband decreased to 0.75 to 7 A.
- The OGO 4 0.5- to 3-A fluxes should be multiplied by a factor of 0.83 and the passband increased to 0.25 to 3 A.

The reason for the drastic change in the passbands of the OSO 3 8- to 12-A detector and the OGO 4 8- to 20-A detector is that the peak in the efficiency at wavelengths shorter than the K- edge of the aluminum in the window (7.94 A) had been neglected previously due to the soft spectrum (2 million °K gray body) used. This spectrum contained relatively few photons at the shorter wavelengths.

Both new and old conversion factor/passband combinations were applied to data obtained during times of low and high solar activity and during a flare. (See Table 2.) Using the old conversion factor/passband combinations, numerous inconsistencies occurred. Typically, the flux observation by OSO 3 in the 8- to 12-A passband was about

Table 1. Old and Revised Conversion Factor/Passband Combinations

Experimenter Spacecraft	Old Passband (A)	Old Conversion Factor	Revised Passband (A)	Revised Conversion Factor	Revision Factor, ϕ (1)
Univ. of Iowa					
Explorers 33,35	2-12	1.8×10^{-6} ergs $cm^{-2}ct^{-1}$ (2)	2.75-12	$(2.47\pm.29)\times10^{-6}$	1.37
Mariner 5	2-9	3.5×10^{-6} ergs $cm^{-2}ct^{-1}$ (3)	2-9.75	$(6.46\pm.51)\times10^{-6}$	1.85
Univ. of Mich.					
OSO 3	8-12	7.71×10^{8} ergs $cm^{-2}amp^{-1}$ (3)	1.5-13	$(1.93\pm.43)\times10^{9}$	2.5
Naval Research Laboratory (NRL)					
OGO 4	8-20	5.25×10^{10} ergs $cm^{-2}amp^{-1}$	2-14	$(5.5\pm1.25)\times10^{10}$	1.05
OGO 4	1-8	5.7×10^{8} ergs $cm^{-2}amp^{-1}$	0.75-7	$(1.59\pm.6)\times10^{8}$	0.28 (5)
OGO 4	0.5-3	1.48×10^{8} ergs $cm^{-2}amp^{-1}$ (4)	0.25-3	$(1.23\pm.17)\times10^{8}$	0.83

(1) (Old flux value) $\times\phi$ =(revised flux value)

(2) Uncertainty of ±50%

(3) Uncertainty of ±30%

(4) Spectral temperatures 5, 7.5, 10, 25, 50, 75, & 100 x 10^6 °K; all others, 1, 2, 5, 10, 20, 50, & 100 x 10^6 °K

(5) Using the temperature in (4), the relative error decreases from .38 to .057, the passband decreases to 1- to 5.75- A, the conversion factor becomes $(1.145\pm.065)\times10^8$, the revision factor becomes 0.20.

Table 2. Typical Observed X-Ray Fluxes

Detector	Old Passband	Old Flux	New Flux	New Passband
Low Activity Sept. 8, 1967 0000 to 0100 UT				
OGO 4	8 - 20	$8.7\ E^{-3}$	$9.2\ E^{-3}$	2 - 14
OSO 3	8 - 12	$(2.1\ E^{-3})$	$5.25\ E^{-3}$	1.5 - 13
EXPL 35	2 - 12	$1\ E^{-3}$	$1.37\ E^{-3}$	2.75 - 12
MAR 5	2 - 9	$3.85\ E^{-4}$	$7.1\ E^{-4}$	2 - 9.75
OGO 4	1 - 8	$3\ E^{-4}$	$8.6\ E^{-5}$	.75 - 7
OGO 4	.5 - 3	$1.4\ E^{-6}$	$1.16\ E^{-6}$	.25 - 3
Moderate Activity Aug. 20, 1967 1200 to 1300 UT				
OGO 4	8 - 20	$3.1\ E^{-2}$	$3.25\ E^{-2}$	2 - 14
OSO 3	8 - 12	$(7.8\ E^{-3})$	$1.95\ E^{-2}$	1.5 - 13
EXPL 35	2 - 12	$3.7\ E^{-3}$	$5.1\ E^{-3}$	2.75 - 12
MAR 5	2 - 9	$1.6\ E^{-3}$	$2.95\ E^{-3}$	2 - 9.75
OGO 4	1 - 8	$1.49\ E^{-3}$	$4.15\ E^{-4}$	.75 - 7
OGO 4	.5 - 3	$4.5\ E^{-6}$	$3.75\ E^{-6}$	.25 - 3
Flare Aug. 19, 1967 0730 to 0800 UT				
OGO 4	8 - 20	$1.24\ E^{-1}$	$1.3\ E^{-1}$	2 - 14
OSO 3	8 - 12	$3.5\ E^{-2}$	$8.75\ E^{-2}$	1.5 - 13
EXPL 35	2 - 12	$1.7\ E^{-2}$	$2.3\ E^{-2}$	2.75 - 12
MAR 5	2 - 9	$9.1\ E^{-3}$	$1.7\ E^{-2}$	2 - 9.75
OGO 4	1 - 8	$3.36\ E^{-2}$	$9.4\ E^{-3}$	.75 - 7
OGO 4	.5 - 3	$8.63\ E^{-5}$	$7.4\ E^{-5}$	.25 - 3

twice the flux observed by Explorer 35 in the 2- to 12-A passband, implying the nonphysical condition of a negative solar X-ray flux between 2 and 8 A. Further, the OGO 4 1- to 8-A flux nearly equalled the Mariner 5 2- to 9-A flux during high activity (i.e., the flux came to within the uncertainty in the Mariner 5 flux). During a flare, the OGO 4 1- to 8-A flux exceeded both the 2- to 9-A flux observed by Mariner 5 and the 2- to 12-A flux observed by Explorer 35. Employing the revised conversion factor/passband combinations eliminates these inconsistencies. These integral fluxes are illustrated in Figures 1, 2, and 3.

Since the solar spectrum in this region decreases strongly with decreasing wavelength, these flux measurements can be combined to yield crude integral spectra. It must be assumed that the major contribution of the incident flux to the detector response occurs at the long wavelength edge of the detector passband. Such integral spectra are shown in Figure 4. The error bars in Figure 4 indicate the relative error calculated. Using the ratios of GM tube count rates or ion chamber currents, crude spectral temperatures can be determined. (See Appendix F.) Temperatures obtained using this method range from 4 to 8 million °K for wavelengths less than 9 A to about 1 million °K for longer wavelenths.

Having shown that the data bases can be made consistent, the integrity of the Explorer 33 and Explorer 35 data bases is questioned next. Figure 5 shows the conversion factor/passband combinations derived for a number of different window thicknesses and gas pressures. The passbands seem to cluster about a nominal passband of 3 to 12 A. Setting a nominal passband of 3 to 12 A increases the worst case of relative error to about 0.4, roughly the value estimated by Van Allen. Given this relative error, the normalization of all the data to the data from one GM tube on Explorer 33 seems justified.

The question concerning the off-axis response was answered using the program given in Appendix E. The angular correction factor for GM2 and GM3 on Explorer 33 was matched against the inflight value every 2.5° from 0° to 25°. The geometry was varied slightly until the match was within 10 percent at every point. The effective efficiency determined every 2.5° was used as input to the program, which determined the best conversion factor/passband combination, and the optimal passband was found. The results are shown in Figure 6. Note that the passband shifted only 0.25 A at the short wavelength edge and that a shift of this level is at the level of integration noise since the integrations were summations in 0.25-A steps. An alternative means of showing the spectral independence of the off-axis response is by taking the ratio of the effective efficiency obtained when viewing the tube at different angles to the effective efficiency obtained when viewing the tube head-on. In the region of 3 to 12 A, these ratios (Figure 7) are relatively constant. Thus, the empirical normalization procedure is valid.

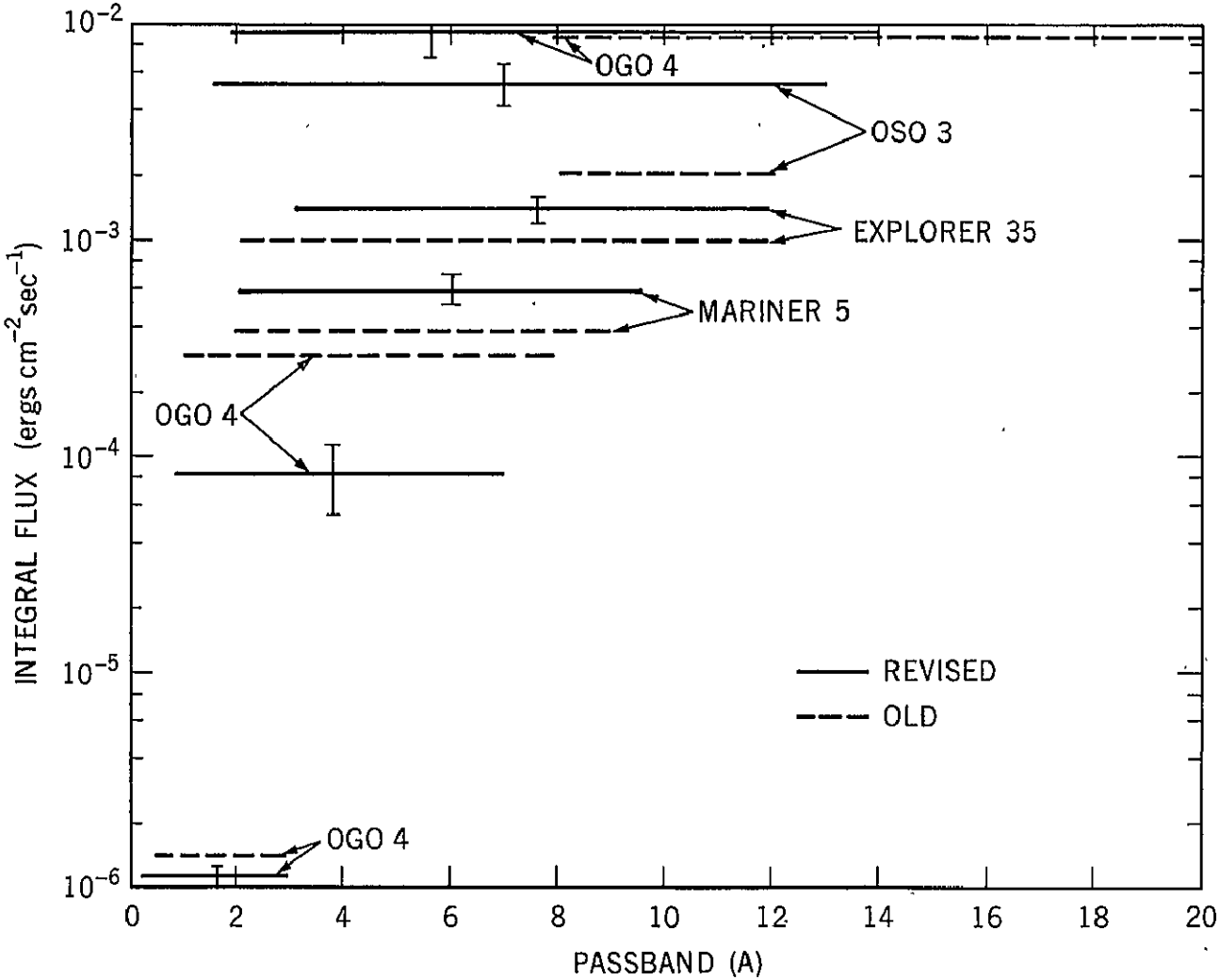

Figure 1. Integral Flux on September 8, 1967, 0000 to 0100 UT (Low Solar Activity).

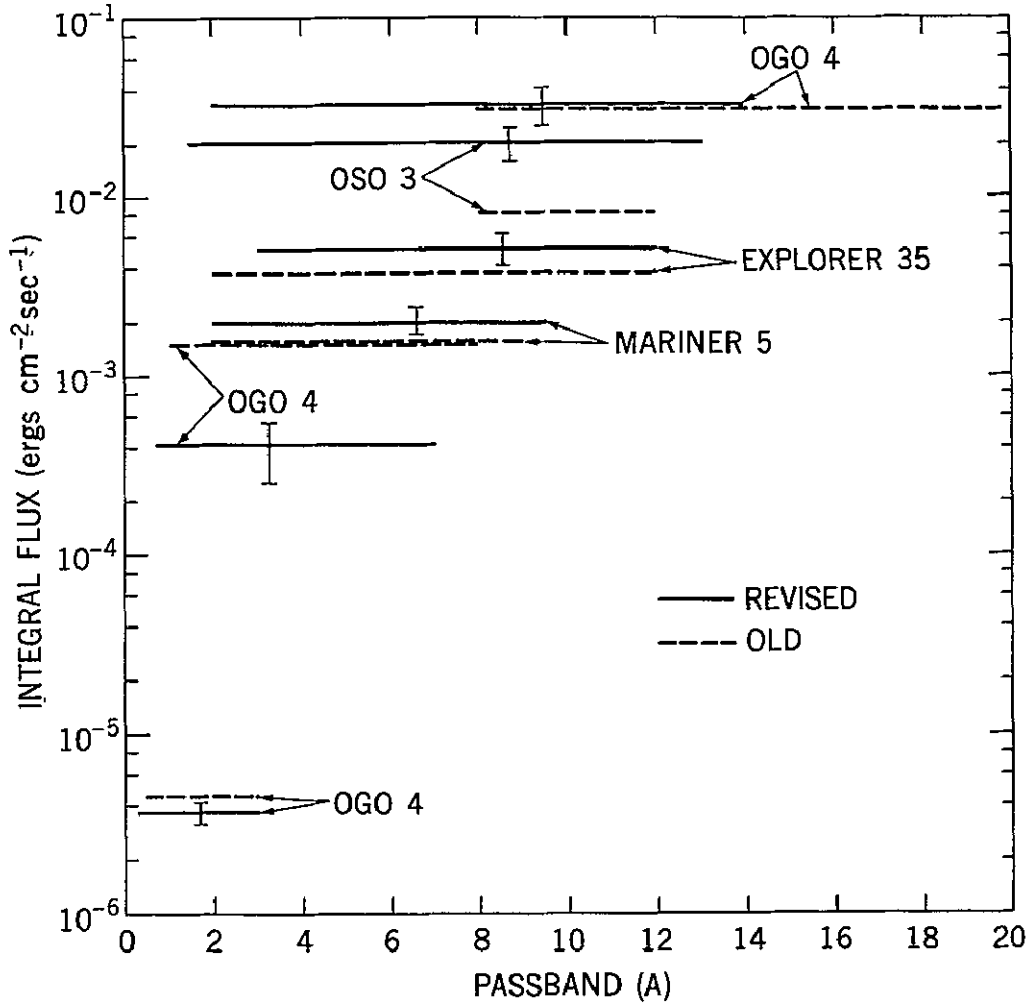

Figure 2. Integral Flux on August 20, 1967, 1200 to 1300 UT (Moderate to High Activity, No Flares)

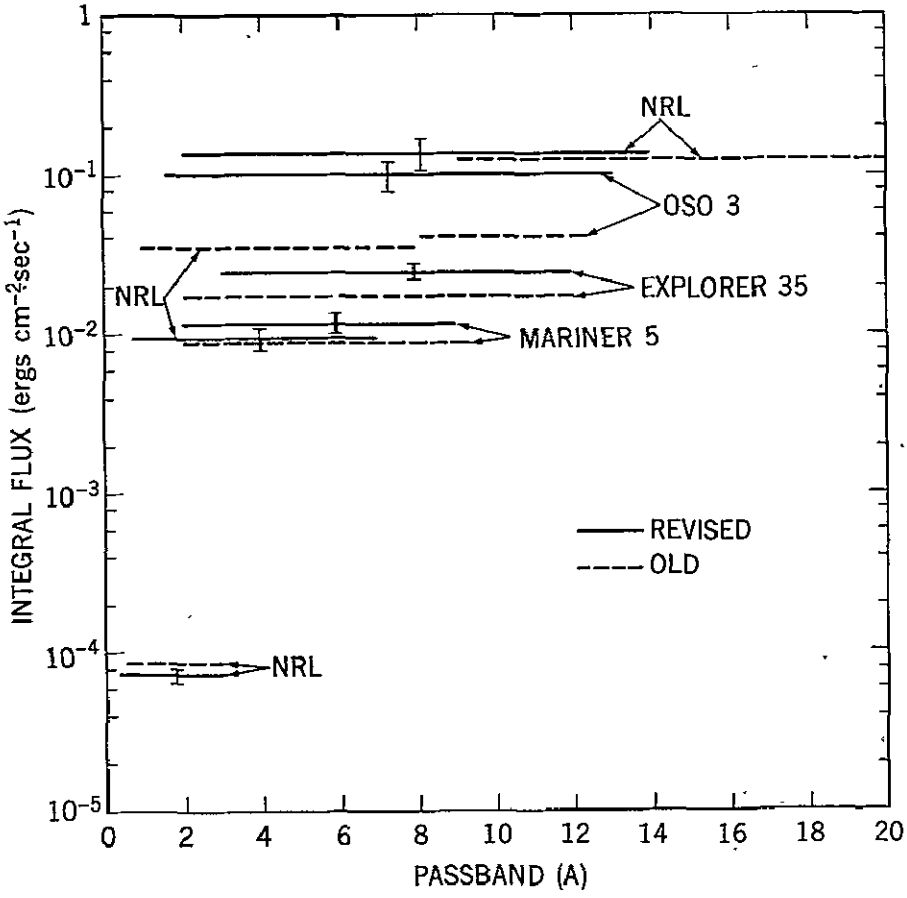

Figure 3. Integral Flux on August 19, 1967, 0730 to 0800 UT
(Post-Peak Decreasing Phase,
3-12 A Flare 0557 - 0615 - 0833;
2B Optical Flare 0530 - 0605 - 0630;
2N Optical Flare 0730 - 0735 - 0800)

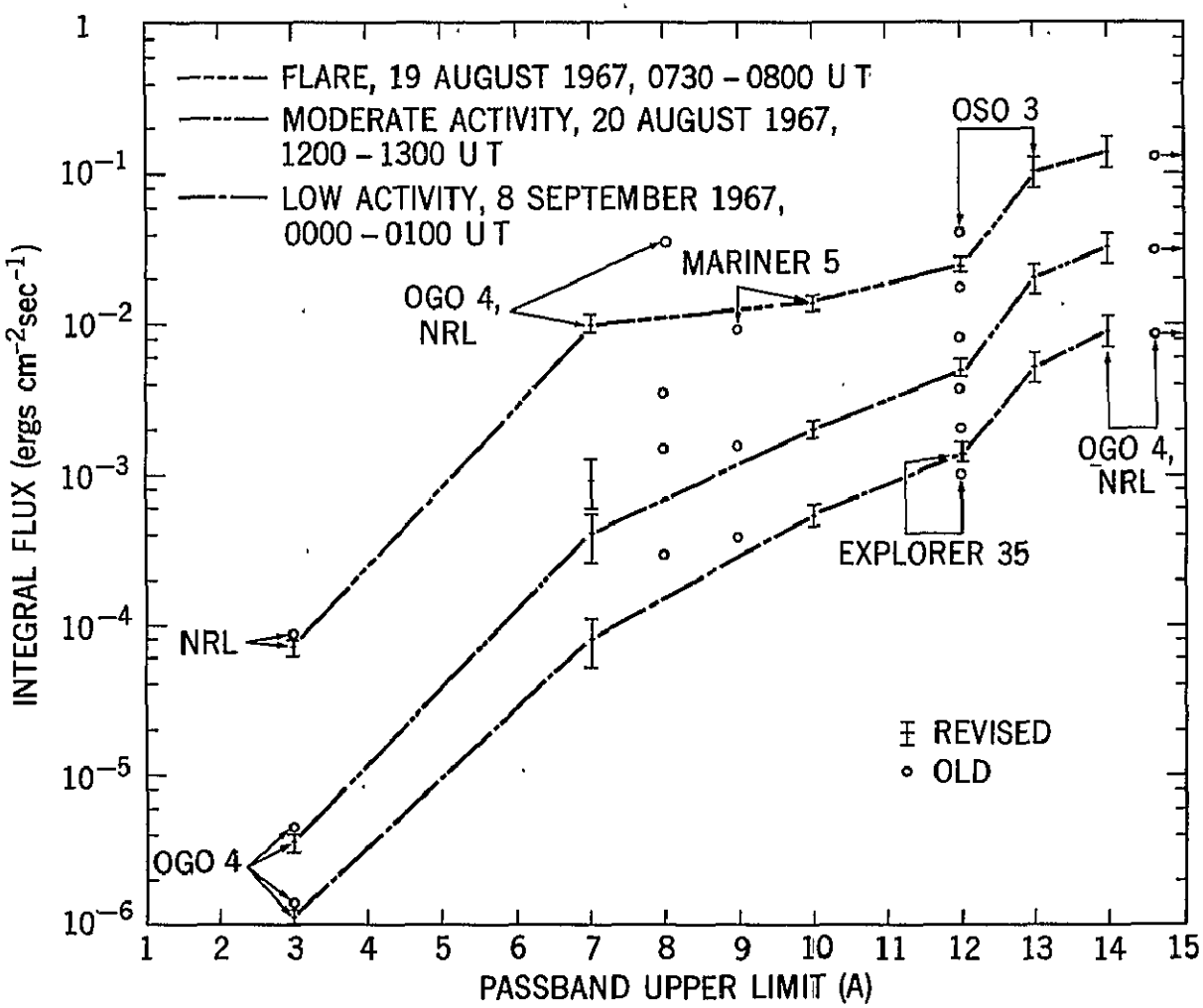

Figure 4. Integral Spectra

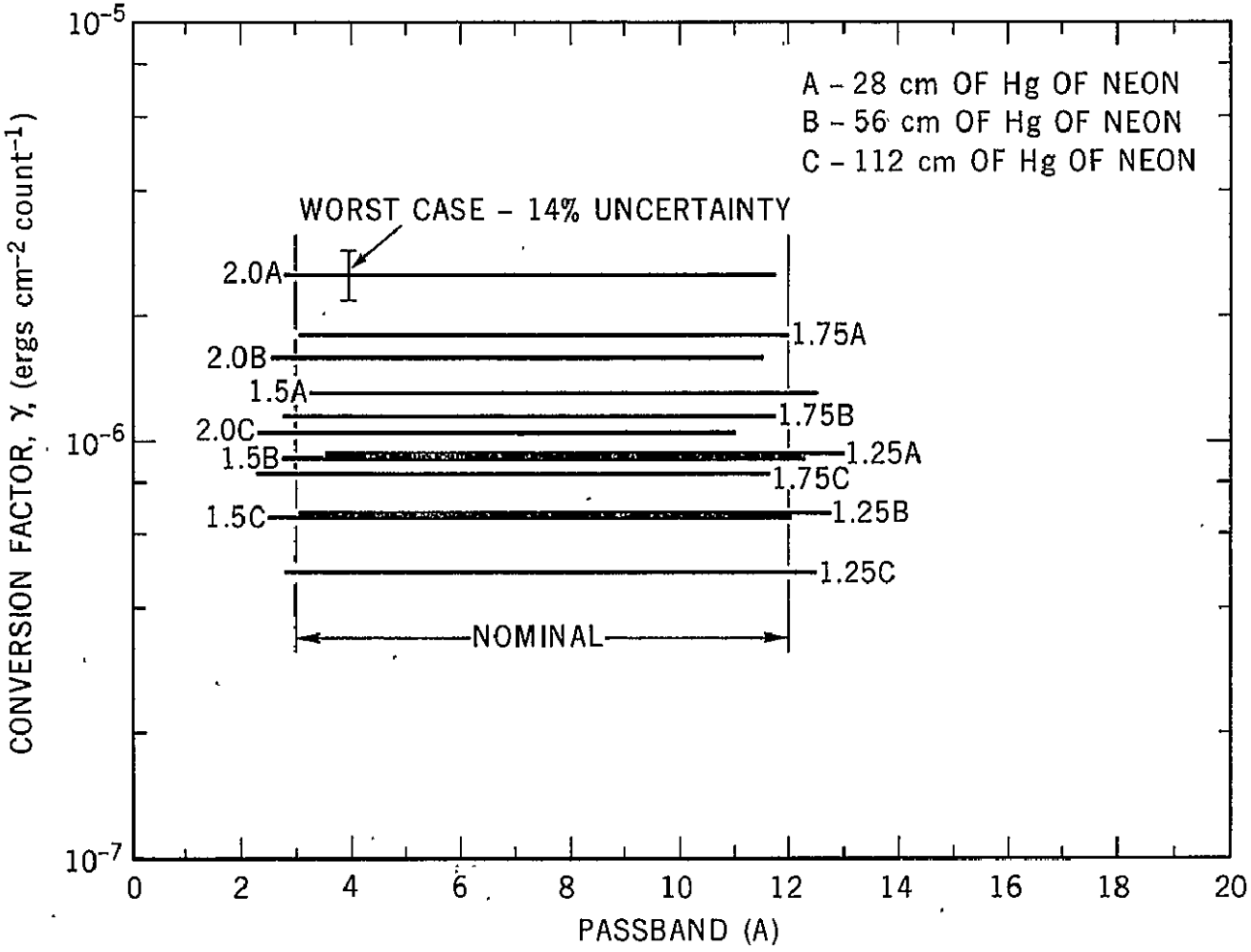

Figure 5. Comparison of Responses of EON 6213 and Similar GM Tubes.
EON 6213 and Anton 213 – 1.25 mg cm^{-2} Mica Windows
LND 704 – 1.5 to 2.0 mg cm^{-2} Mica Windows

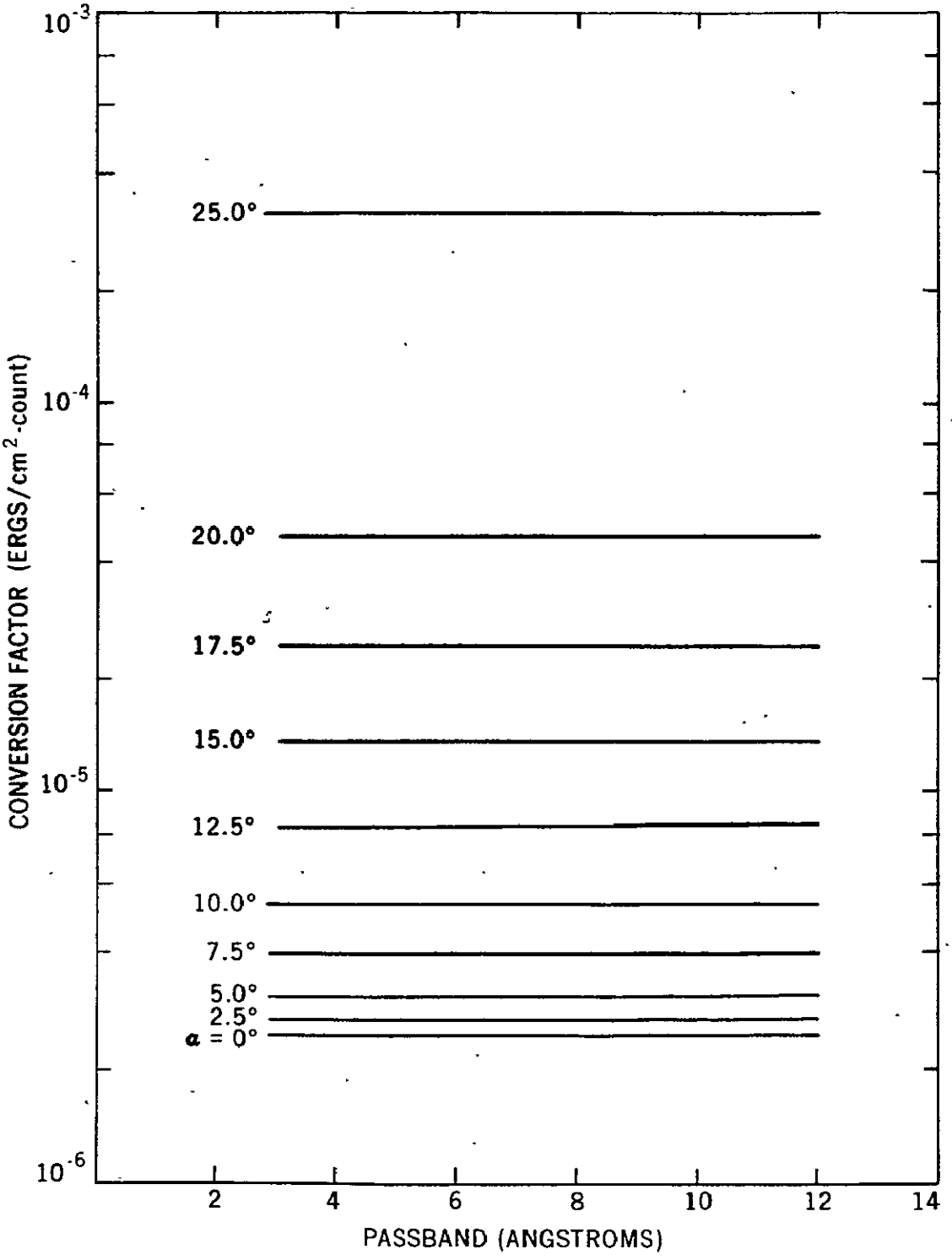

Figure 6. The Passband and Conversion Factor as a Function of Off-Axis Angle α (Explorer 33, GM2 and GM3)

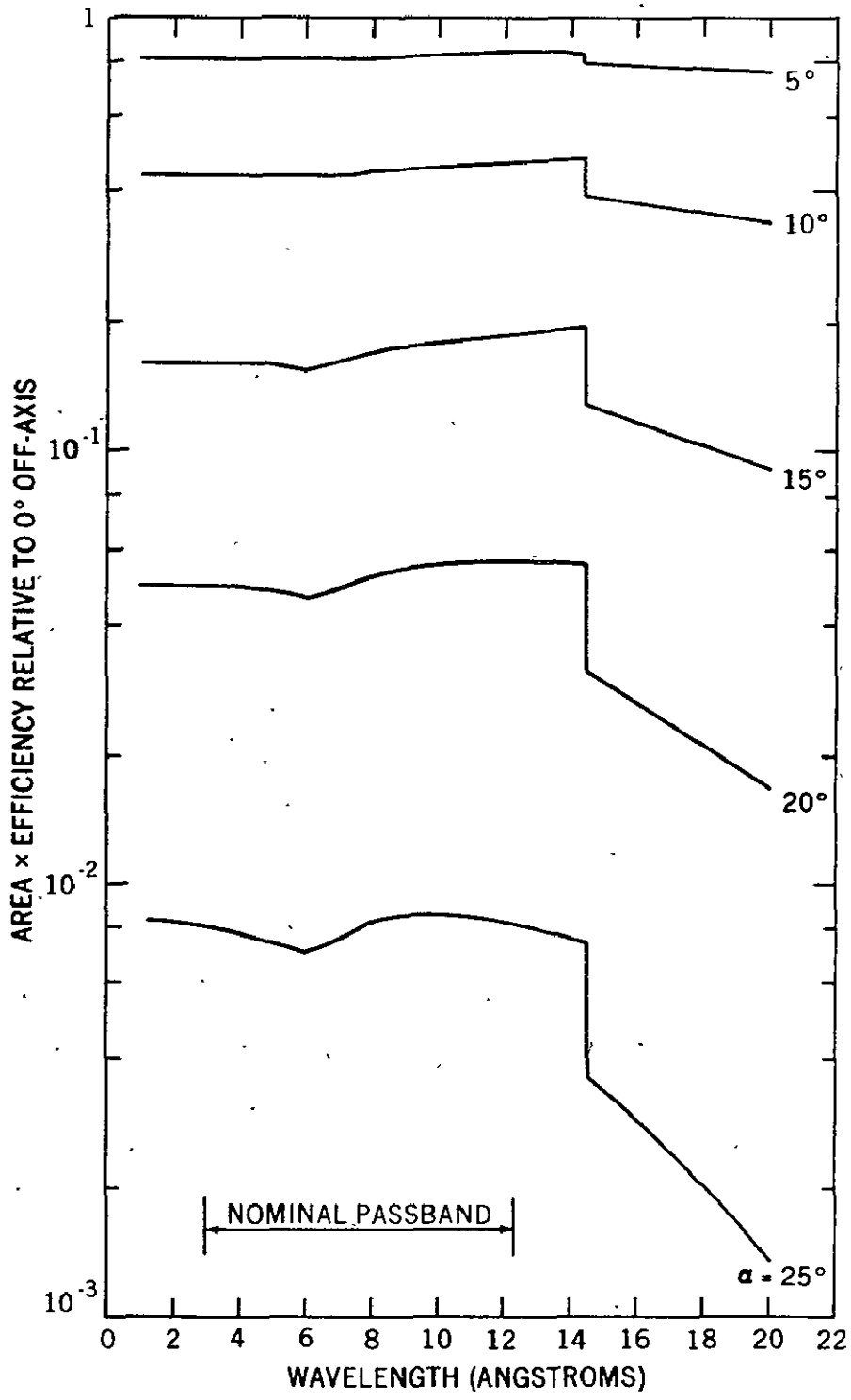

Figure 7. Relative Effective Efficiency versus Wavelength for a Point Source at Various Off-Axis Angles (Explorer 33, GM2 and GM3)

REFERENCES

1. Gilman, L., and A. J. Rose, APL/360, An Interactive Approach, John Wiley and Sons, New York, 1970.

2. Henke, B. L., and R. L. Elgin, "X-Ray Absorption Tables for the 2- to 200-A Region," Advances in X-Ray Analysis, 13, Plenum Press, New York, 1970.

3. Kreplin, R. W., "Solar X-Rays," Ann. Geophys., 17, 151, April 1961.

4. McMaster, W. H., N. K. Del Grande, J. H. Mallett, and J. H. Hubbell, "Compilation of X-Ray Cross Sections, Revision 1," Lawrence Radiation Laboratory document UCRL-50174, University of California, Berkeley, 1969.

5. Pakin, S., APL/360 Reference Manual, Science Research Associates, Chicago, 1968.

6. Van Allen, J. A., "The Solar X-Ray Flare of July 7, 1966," J. Geophys. Res., 72, 5903, Nov. 1967.

PRECEDING PAGE BLANK NOT FILMED

APPENDIX A. A BRIEF DESCRIPTION OF THE APL LANGUAGE

This appendix is intended to provide the definitions necessary to change the programs listed in the following appendixes into other computer languages. More complete descriptions of the APL language and its use are found in Pakin (1968) and Gilman and Rose (1970).

General Characteristics:

Accuracy - Double precision is used except for integers, which do not require it.

Order of Execution - Operations are processed from right to left, with operations enclosed within parentheses being performed first. Thus, A - B - C would be equivalent to A - (B - C).

Classes of Operators - There are two classes of APL operators, monadic (right argument only) and dyadic (both right and left arguments). The same symbol may be used for both a monadic operator and a dyadic operator, but the meanings will be different. For example, *X means $\log_e X$, while 10*X means $\log_{10} X$.

Classes of Variables - Variables may be scalars, vectors, or matrices. They are described as follows:

Scalars - rank 0, no dimensions, no components.
For example, X = 9

Vectors - one dimension, one or more components.
For example, X = 1 3 6 4 5

Matrices or Tensors - more than one dimension, one or more components in each dimension.
For example, X = 1 3 6
2 7 4
9 5 7

Most operators will function with or on any variable. If two or more variables are involved, their ranks must match (both dimensions and components). The exception to this requirement is the scalar, which may be used with vectors or matrices (e.g., in multiplication, addition, etc.). In this case, the scalar is automatically matched with the vector or matrix.

Table A-1 describes the operators used in the following programs. The variables used are A and B, which may be scalars, vectors (with components A_I and B_I), or matrices (with components A_{IJ} and B_{IJ}, although matrices may have more than two dimensions). Components are identified as follows:

Scalar - no components
Vector - A[I]
Matrix - A[I;J]
Tensor - A[I;J;K;1]

Table A-1. APL Operator Definitions (Sheet 1 of 3)

OPERATOR	RESULT STORED AS C
C←+A	THE VALUE OF B
C←A+B	THE SUM OF A AND B
C←-B	THE VALUE OF B WITH THE OPPOSITE SIGN
C←A-B	THE DIFFERENCE OF A MINUS B
C←×B	THE SIGN OF B - ¯1 IF B<0, 0 IF B=0, 1 IF B>0
C←A×B	A MULTIPLIED BY B
C←÷B	THE RECIPROCAL OF B
C←A÷B	A DIVIDED BY B
C←*B	EXPONENTIAL TO THE BASE E OF B
C←A*B	A TO THE POWER B
C←⍟B	THE LOG. TO THE BASE E OF B
C←A⍟B	THE LOG. TO THE BASE A OF B
C←⌈B	THE NEXT INTEGER ≥B
C←A⌈B	THE LARGER OF A AND B
C←⌊B	THE NEXT INTEGER ≤B
C←A⌊B	THE SMALLER OF A AND B
C←|B	THE ABSOLUTE VALUE OF B
C←A|B	THE REMAINDER OF B DIVIDED BY A
C←○B	PI TIMES B
C←1○B	SIN B, B IN RADIANS
C←2○B	COS B, B IN RADIANS
C←3○B	TAN B, B IN RADIANS
C←¯1○B	SIN¯1 B, C IN RADIANS
C←¯2○B	COS¯1 B, C IN RADIANS
C←¯3○B	TAN¯1 B, C IN RADIANS
C←A<B	C=1 IF A IS LESS THAN B, IF NOT, C=0
C←A≤B	C=1 IF A IS LESS THAN OR EQUAL TO B, IF NOT, C=0
C←A=B	C=1 IF A EQUALS B, IF NOT, C=0
C←A≠B	C=1 IF A DOES NOT EQUAL B, IF NOT, C=0
C←A≥B	C=1 IF A IS GREATER THAN OR EQUAL TO B, IF NOT, C=0
C←A>B	C=1 IF A IS GREATER THAN B, IF NOT, C=0

Table A-1. APL Operator Definitions (Sheet 2 of 3)

OPERATOR	OPERATION
$C\leftarrow A\wedge B$	LOGICAL A AND B (A,B = 1'S,0'S)
$C\leftarrow A\vee B$	LOGICAL A OR B OR BOTH (A,B = 1'S,0'S)
$C\leftarrow A\neq B$	LOGICAL A OR B BUT NOT BOTH (A,B = 1'S AND 0'S)
$C\leftarrow\sim B$	INVERSE (NOT) B, INTERCHANGES 1'S AND 0'S
$C\leftarrow\grave{A}B$	C← SUBSCRIPTS OF THE VECTOR B FOR AN ASCENDING SORT
$C\leftarrow\grave{\forall}B$	C← SUBSCRIPTS OF THE VECTOR B FOR A DESCENDING SORT
$C\leftarrow\iota B$	C← A VECTOR OF INTEGERS FROM 1 TO B
$C\leftarrow A\iota B$	C← THE INDEX OF THE FIRST ELEMENT IN THE VECTOR A EQUAL TO THE VALUE OF B
$C\leftarrow,A$	C← A VECTOR OF THE ELEMENTS WHICH COMPOSED THE SCALAR, MATRIX OR TENSOR A
$C\leftarrow A,B$	C← A VECTOR COMPOSED OF THE VECTOR B APPENDED TO THE END OF THE VECTOR A
$C\leftarrow\rho B$	C← THE DIMENSION OF B, A BLANK IF B IS A SCALAR, A SINGLE NUMBER IF B IS A VECTOR, A VECTOR OF NUMBERS IF B IS A MATRIX OR TENSOR
$C\leftarrow A\rho B$	C← A TENSOR, MATRIX, OR VECTOR OF DIMENSION A FROM THE VALUES OF THE SCALAR OR VECTOR B
$C\leftarrow\phi B$	$C[I]\leftarrow B[N-I]$, WHERE N IS THE NUMBER OF COMPONENTS IN B (REVERSES THE ORDER OF ELEMENTS IN B), OR $C[I;J]\leftarrow B[I;N-J]$, WHERE N IS THE NUMBER OF COMPONENTS IN THE ROWS OF MATRIX B (REVERSES THE ELEMENTS IN EACH ROW OF A MATRIX)
$C\leftarrow\Theta B$	$C[I;J]\leftarrow B[J;I]$, TRANSPOSES THE MATRIX ROWS/COLUMNS
$C\leftarrow A+.\times B$	$C[I]\leftarrow \sum_J A[J]\times B[J;I]$ OR $C[I;K]\leftarrow \sum_J A[I;J]\times B[J;K]$
$C\leftarrow A\circ .\div B$	$C[I;J]\leftarrow A[I]\div B[J]$, GENERATES A MATRIX WITH VALUES FOUND IN VECTORS, OTHER OPERATORS MAY BE USED
$C\leftarrow +/B$	C← $A[I]$ OR $C[I]\leftarrow \sum_J A[I;J]$
$C\leftarrow +\not/B$	C← $A[I]$ OR $C[J]\leftarrow \sum_I A[I;J]$
$C\leftarrow \times/B$	C← $B[I]$ OR $C[I]\leftarrow \prod_J B[I;J]$
$C\leftarrow \times\not/B$	C← $B[I]$ OR $C[J]\leftarrow \prod_I B[I;J]$

Table A-1. APL Operator Definitions (Sheet 3 of 3)

OPERATOR	OPERATION
C←⌈/B	C IS THE LARGEST ELEMENT IN THE VECTOR B, OR A VECTOR OF THE LARGEST ELEMENTS IN THE ROWS OF THE MATRIX B
C←⌈⌿B	C IS THE LARGEST ELEMENT IN THE VECTOR B, OR A VECTOR OF THE LARGEST ELEMENTS IN THE COLUMNS OF THE MATRIX B
C←⌊/B	C IS THE SMALLEST ELEMENT IN THE VECTOR B, OR A VECTOR OF THE SMALLEST ELEMENTS IN THE ROWS OF THE MATRIX B
C←⌊⌿B	C IS THE SMALLEST ELEMENT IN THE VECTOR B, OR A VECTOR OF THE SMALLEST ELEMENTS IN THE COLUMNS OF MATRIX B
X←□	CALLS FOR INPUT FROM THE TERMINAL OF A VALUE FOR X
□←X	DISPLAYS X ON THE TERMINAL
NAME:	PRECEEDS AN INSTRUCTION AND LABELS A LOCATION OR LINE NUMBER WITHIN A PROGRAM (THE NAME USED IS ARBITRARY)
→NAME	BRANCHES THE PROGRAM TO THE LOCATION CALLED NAME
→(LOGIC)/NAME	IF THE LOGICAL STATEMENT IS TRUE, THE PROGRAM BRANCHES TO THE LOCATION CALLED NAME, IF NOT TRUE THE PROGRAM PROCEEDS TO THE NEXT LINE
→(LOGIC J←J+1)/NAME	INCREMENTS J, TESTS THE VALUE AND BRANCHES TO LOCATION CALLED NAME IF THE LOGIC IS TRUE, IF THE LOGICAL STATEMENT IS UNTRUE, THE PROGRAM PROCEEDS TO THE NEXT LINE
I20	TIME OF DAY IN 1/60 SEC.
I21	CENTRAL PROCESSOR UNIT (CPU) TIME USED THIS SESSION (1/60 SEC.)
I23	NUMBER OF TERMINAL USERS ON SYSTEM

APPENDIX B. MASS ABSORPTION COEFFICIENTS

The program called INTERP interpolates or extrapolates the needed mass absorption coefficients that are not already listed in the tables of Henke and Elgin (1970) or McMaster et al. (1969). The program is based on the assumption that the mass absorption coefficients are related to the wavelength through a power law. The geometry employed is illustrated in Figure B-1. In Regions I and III, the values are extrapolated while, in Region II, the values are interpolated. The program is flowcharted in Figure B-2 and listed at the end of this appendix.

The operation of the program is as follows. The data are entered and converted to natural logarithms (i.e., the data are linearized). The slope of the graph of known mass absorption coefficients versus wavelengths is determined between each successive pair of points. The slopes are corrected for the effect of absorption edges, and a multiplier is determined to correct for edges found in Regions I and III and for the extrapolation in Region IV. Using the slopes, the multipliers, and the known points, the required coefficients are calculated.

In detail, the program operates as follows.

Steps 1 - 4 require the entry of the known coefficients and their corresponding wavelengths and convert them to natural logarithms. The data must be in the form of vectors.

Steps 5 & 6 order the above data by ascending wavelength.

Steps 7 - 12 require the entry of the known edges, the corresponding jump ratios, and the wavelengths of the wanted coefficients and convert them to vectors of natural logarithms.

Steps 13 - 15 determine the number of successive pairs of points, the slopes (a vector) between successive pairs of points, and set the multiplier as a vector of zeros, one component corresponding to each coefficient wanted.

Steps 16 & 17 set up a loop which cycles through each absorption edge and makes the necessary corrections to the slope or to the multiplier. Step 17 branches to the end of the loop if the edge lies outside the region of interest.

Step 18 corrects the multiplier if the edge falls in either Region I or Region III (dashed lines in Figure B-1) by adding (Region I) or subtracting (Region III) the logarithm of the jump ratio to coefficients at wavelengths shorter (Region I) or longer (Region III) than the edge.

Step 19 branches to the end of the loop if the edge falls within Region I or Region III.

Steps 20 - 22 modify the slope and the multiplier for edges falling in Region III and wanted points (multiplier) falling in Region IV.

Step 23 increments the index J, tests it, and loops back to step 17 if necessary.

Steps 24 - 26 calculate the answer using the slopes, the known coefficients, and the multipliers and convert the answer out of its logarithmic form.

Steps 27 - 31 convert the initial data out of its logarithmic form.

Step 32 announces that the program is finished.

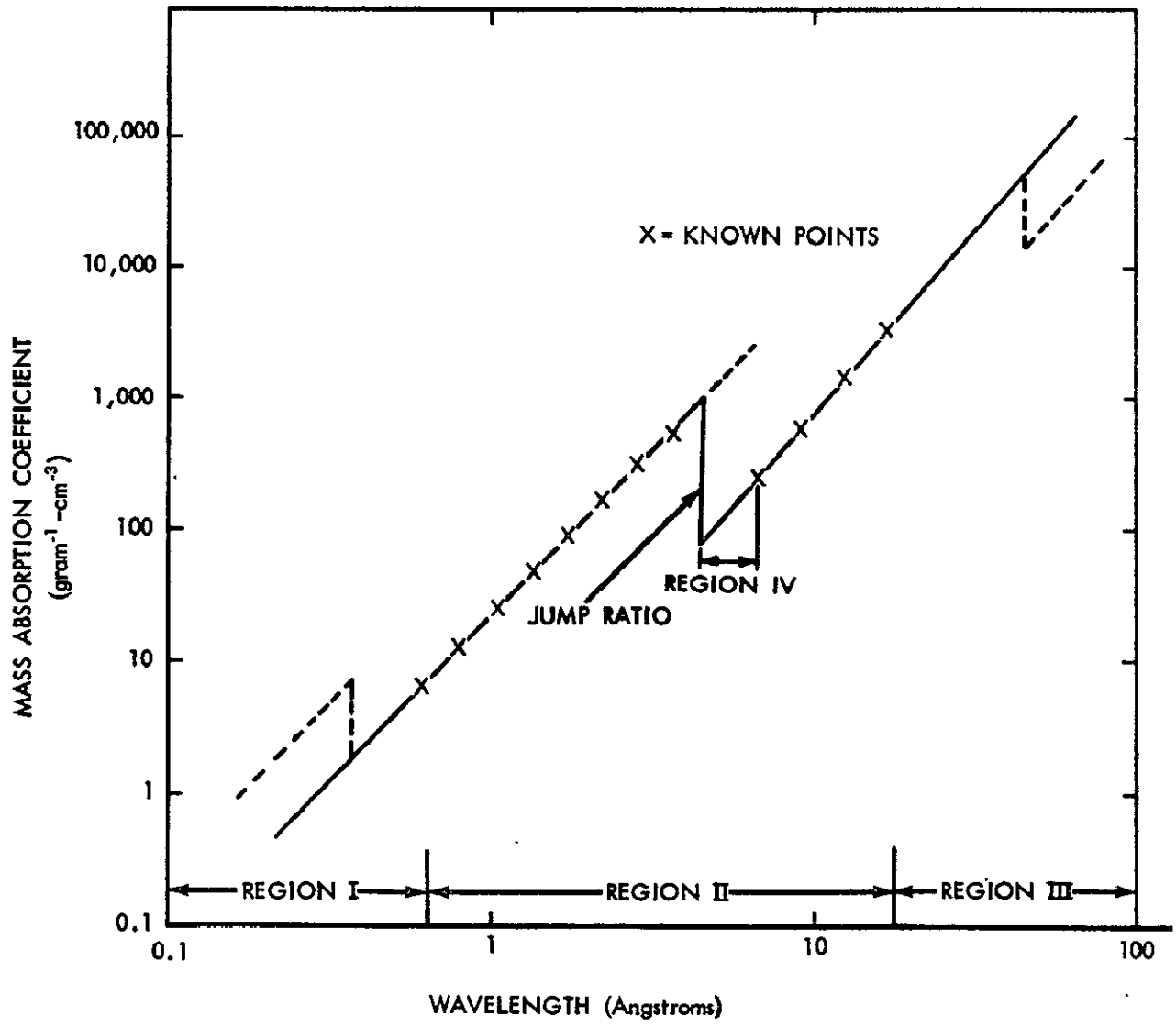

Figure B-1. Geometry for the Interpolation of Mass Absorption Coefficients

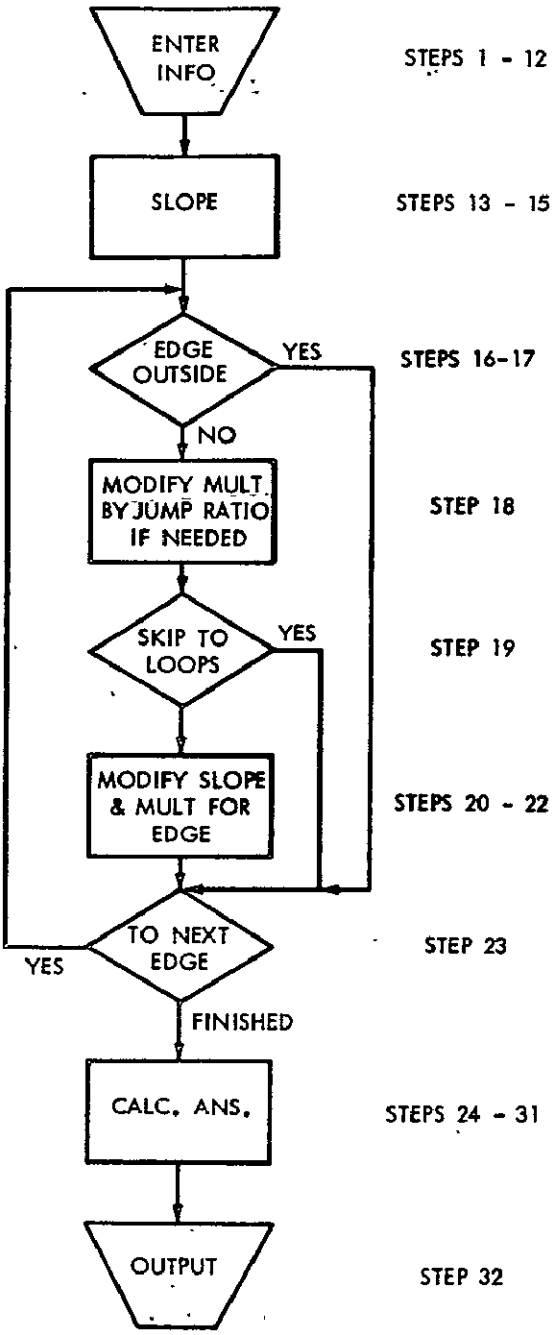

Figure B-2. Flowchart for INTERP Program

```
    ∇ INTERP
[1]   'ENTER KNOWN COEFFICIENTS:'
[2]   CKN←⊛CKN←⍬
[3]   'ENTER CORRESPONDING WAVELENGTHS:'
[4]   WKN←⊛WKN←⍬
[5]   CKN←CKN[⍋WKN]
[6]   WKN←WKN[⍋WKN]
[7]   'ENTER WAVELENGTHS OF EDGES:'
[8]   EDG←,⊛EDG←⍬
[9]   'ENTER CORRESPONDING JUMP RATIOS:'
[10]  JR←,⊛JR←⍬
[11]  'ENTER WAVELENGTHS OF WANTED COEFFICIENTS:'
[12]  W←,⊛W←⍬
[13]  J←¯1+⍴CKN
[14]  SL←(CKN[1+⍳J]-CKN[⍳J])÷(WKN[1+⍳J]-WKN[⍳J])
[15]  MULT←(⍴W)⍴0
[16]  J←1
[17] SLMOD:→((EDG[J]<W[1])+(EDG[J]>W[⍴W]))/ENDMOD
[18]  MULT←MULT+JR[J]×(((EDG[J]<WKN[1])×(EDG[J]>W))-(EDG[J]
      >WKN[⍴WKN])×(EDG[J]<W))
[19]  →((EDG[J]<WKN[1])+(EDG[J]>WKN[⍴WKN]))/ENDMOD
[20]  I←+/WKN<EDG[J]
[21]  SL[I]←SL[I]+(JR[J]÷(WKN[I+1]-WKN[I]))
[22]  MULT←MULT-((W>EDG[J])×(W<WKN[I+1]))×JR[J]
[23] ENDMOD:→((⍴EDG)≥J←J+1)/SLMOD
[24]  I←(I=0)+I←+/W∘.≥WKN
[25]  SL←SL,SL[⍴SL]
[26]  ANS←⍎(MULT+(CKN[I]+SL[I]×(W-WKN[I])))
[27]  W←⍎W
[28]  CKN←⍎CKN
[29]  WKN←⍎WKN
[30]  JR←⍎JR
[31]  EDG←⍎EDG
[32]  'ANSWER IS STORED AS ''ANS''.'
    ∇
```

APPENDIX C. PASSBAND DETERMINATION PROGRAM

The program called PASSBAND determines the optimum passband for a GM tube or an ion chamber following the procedure outlined in Section 1, "Approach to the Problem." The program is flowcharted in Figure C-1 and listed at the end of this appendix.

The operation of the program is as follows. The GM tube efficiency or the ion chamber efficiency (multiplied by the gas gain, electronic charge, and photon energy) is entered followed by the area and the estimated limits. The temperature range is chosen, and lines are added to the spectra, if appropriate. For each pair of limits, set up as a matrix with the lower limit varying from the estimated lower limit -0.5 A to +0.5 A (in 0.25-A steps) versus the estimated upper limit -0.5 A to +0.5 A (in 0.25-A steps), the program determines the 16 conversion factors, γ's, the mean conversion factor, the standard deviation, σ, and the relative error, σ/mean. A matrix of the relative error is derived (in effect, if not in fact), the minimum relative error is determined, and the corresponding limits are found. If the limits occur on any edge of the matrix, the estimated limits are jumped so that the opposite edge of the next matrix will correspond to the edge, the estimated limits are set equal to the limits corresponding to the minimum, the matrix is recalculated and displayed, and the final limits are printed out followed by the linearly averaged conversion factor and its corresponding standard deviation, the logarithmically averaged conversion factor and its standard deviation multiplier, and the 16 conversion factors (one for each spectrum) calculated for the optimum passband. If the minimum occurs twice in succession and has the same value, the program sets the estimated limits to the limits corresponding to the minimum and cycles through to the final results described above.

This procedure prevents any infinite oscillation from occurring should the final result lie on the common edge of two adjoining matrices, while permitting the program to jump in the direction of the absolute minimum rather than increment one step at a time. In the rare case where one limit falls within 0.25 A of either 0 or 20.25 A, the final minimum cannot occur in the center element of the relative error matrix. (This condition would require undefined values of spectra $dN/d\lambda$ and efficiency.) This condition is recognized because the minimums are identical three successive times, and the program prints out the off-centered matrix with the final results.

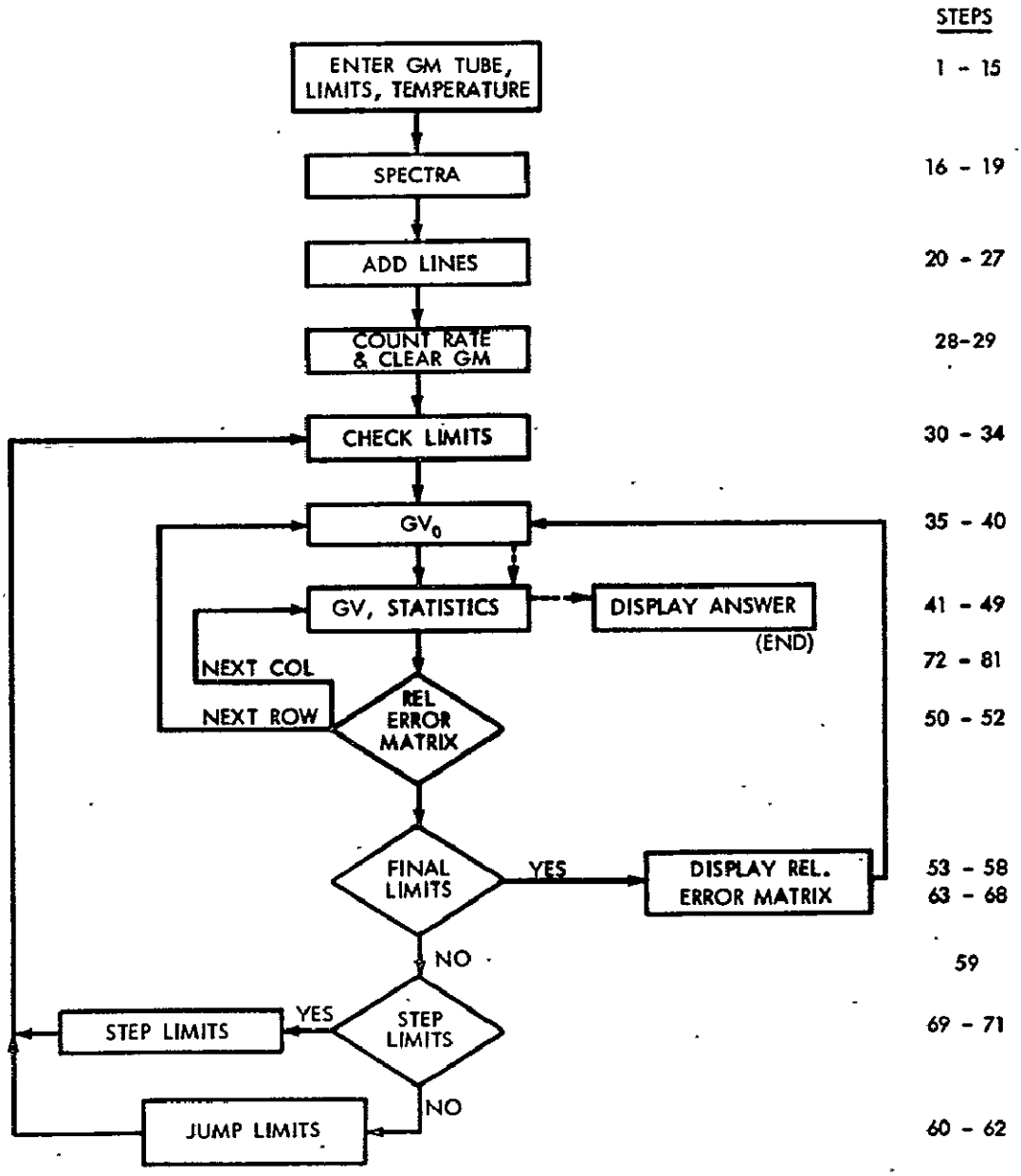

Figure C-1. Flowchart for PASSBAND Determination Program

In detail, the program operates as follows.

Steps 1 - 4 initialize variables.

Steps 5 - 8 require the entry of the GM tube efficiency (or the ion chamber equivalent), multiply it by the inverse of the photon energy λ/hc, and convert it to an integer accurate to one half part in 10^8 with a corresponding normalization factor XY.

Steps 9 - 12 require the entry of estimated limits (a two-element vector) and determine the corresponding indices in a standard vector of wavelengths.

Steps 13 - 19 require the entry of the desired temperature range (entering the letter Y enters the values set in step 13) and calculate relative free-free and blackbody spectra for the temperatures. Step 16 determines a matrix of values of hc/WkT, with the wavelength W in rows versus the temperature T in columns. Step 17 sets a maximum value of 170 on independent elements in the matrix, and steps 18 and 19 determine relative spectra (free-free spectra are determined in step 18, blackbody in step 19) in integer form accurate to one half part in 10^8.

Steps 20 - 23 require the entry of the wavelengths of lines (each, in turn, determining the corresponding index of the wavelength vector W) and the corresponding line/continuum ratios.

Steps 24 - 27 form a loop (step 24 sets the index as J = 1) which multiplies each element in a row corresponding to a line wavelength by a factor of the line/continuum ratio + 1. Step 27 increments the index J and tests its value against the total number of lines.

Step 28 generates a 16-element vector of count rates, one for each spectrum. The integer form of the efficiency is corrected by the factor $(XY \cdot 10^{-8})$. The terms like (GM +. x FF) are equivalent to

$$\sum_I GM_I \cdot FF_{IJ}$$

The last term adds the spectra $dN/d\lambda = 1$ and $dN/d\lambda = \lambda^{-2}$.

Step 29 clears the value of GM.

Steps 30 - 32 check that the upper and lower limits estimated are not too close to each other or to either 0 or 20.25 A. Step 30 determines the difference between the two limits, X. Step 31 accomplishes several functions. First, if the limits are too close together ($X < 4$) and the estimated lower limit $IO > 8$, then the difference between 4 and X is subtrated from the value of IO. Second, if $IO > 2$, it retains its current value. Finally, if $IO < 3$, it is replaced by a value of 3. Step 32 provides that if $IO \leq 8$, the difference between 4 and X is added to JO. If $JO < 79$, it retains its current value. If $JO > 78$, it is set equal to 78.

Steps 33 - 35 set I and J. Step 35 clears variables.

Steps 36 - 40 determine a 16-element vector (GV) of energy fluxes divided by count rates for the wavelength range between a given lower limit and an upper limit -0.25 A. This calculation is performed by a summation of the energy fluxes times a weighting factor, WT, which has values of unity except for the first element, which has a value of 0.5. The weighting function is used to perform the energy integration as in Figure C-2. The value of X is set as a vector of 16 zeros by step 40.

Steps 41 - 44 add the value of X to GV, element by element. The first time through this addition does nothing, but subsequent times it serves to fill out each increment since $dE/d\lambda$ for the end points was taken to be $0.5 \cdot dE/d\lambda$ for the interior points. Step 42 defines X equal to a 16-element vector of $0.5 \cdot dE/d\lambda$ divided by the count rate vector and adds X to GV, element by element. Then GV is linearly averaged. Step 43 is equivalent to

$$AVE = \frac{\sum_I GV_I}{16}$$

The standard deviation, SIG, is determined by step 44, which is equivalent to

$$SIG = \left[\frac{\sum_I \left(GV_I - AVE \right)^2}{15} \right]^{1/2}$$

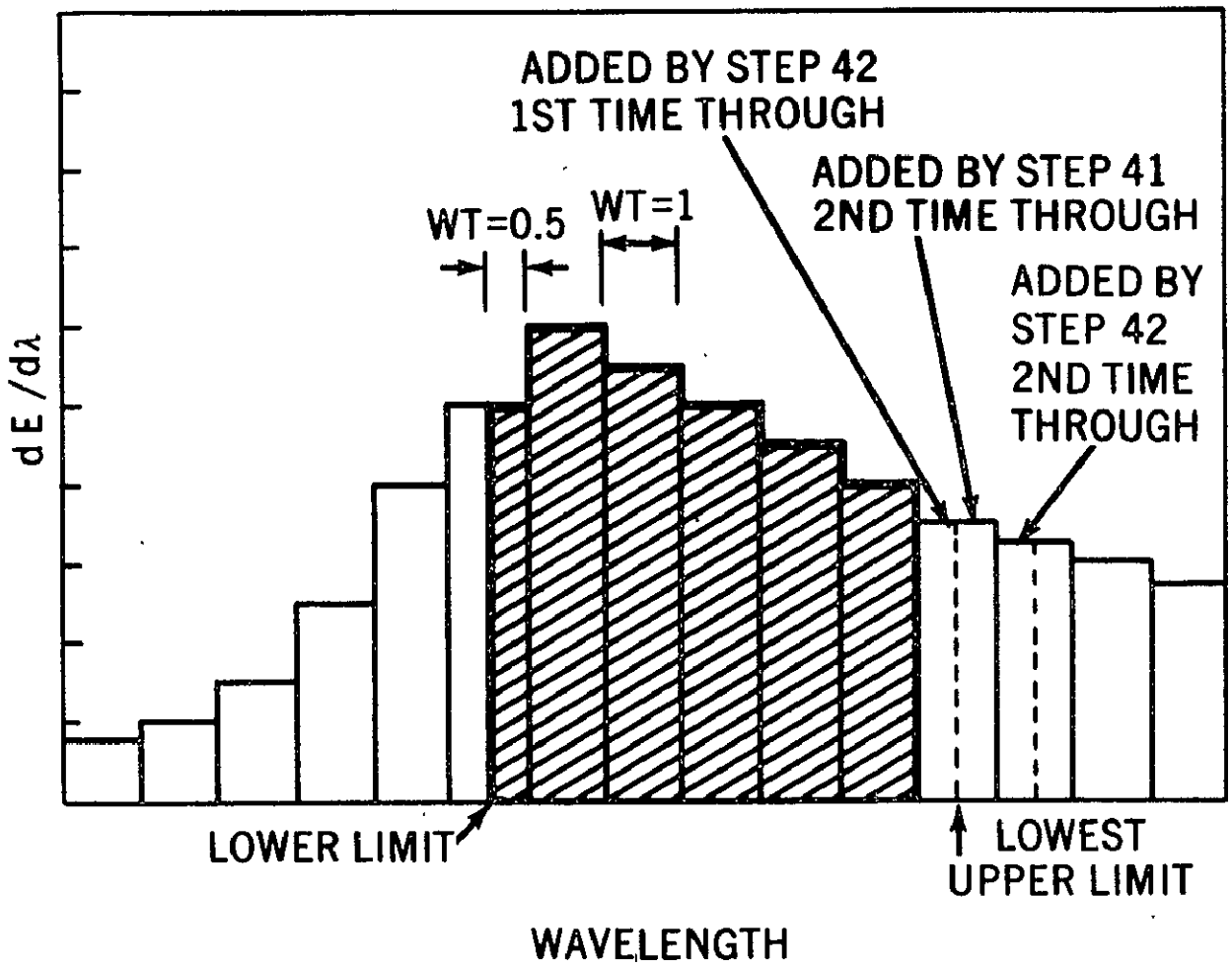

Figure C-2. Schematic Illustration of the Integration Technique Used for Calculating Energy Intervals

Steps 45 - 49 If the value of FLAG = +1, the program branches to step 72. If FLAG = 0, the program continues to step 46. Step 46 adds the new element SIG to the old vector SMAT, and step 47 adds the new element AVE, to the old vector AMAT. Steps 48 and 49 add the new elements J and I to old vectors XJ and XI.

Steps 50 - 52 test the values of J and I (the upper and lower limits), incrementing and looping if necessary. Step 50 loops to step 41 until all upper limits are calculated. Step 51 resets J because the lower limit is now changed. Step 52 loops to step 36 until all lower limits are calculated.

Steps 53 - 57 determine the minimum relative error (SIGM) and the corresponding wavelength indices I and J. Step 53 divides SMAT (a 25-element vector) by AMAT (another 25-element vector), element by element. Step 54 (right to left) determines the minimum value of X and calls it SIGM. Step 55 determines the index of X corresponding to SIGM, and steps 56 and 57 define the corresponding components of XI and XJ as the indices sought.

Steps 58 - 71 test the values of I and J and take appropriate action. Step 58 shunts the program to step 63 if the current value of SIGM equals that determined two iterations ago (stored in SIGM) or if I = IO and J = JO. Otherwise, the program goes to step 59. Step 59 branches the program to step 69 if SIGM equals the SIGM of the previous iteration or I and J do not occur on edges. Otherwise, the program goes to step 60. Steps 60 and 61 jump IO and JO to values equal to the initial values plus twice the difference between I and IO or J and JO. Step 62 branches the program to step 30. Step 63 causes a printout of a 5-by-5 matrix of relative errors. Step 64 prints a title for the matrix. Step 65 causes the terminal to skip a line. Step 66 prints out the optimum passband. In cases where one limit lies near 0 or 20.25 A, the optimal limits are not the same as the ones mentioned in the matrix title. Step 67 sets the value of FLAG = 1, and step 68 branches the program to step 36. Steps 69 - 71 set IO = I and JO = J and branch the program to step 30.

Steps 72 - 77 print out the results on the terminal. Step 72 prints out the values of the linearly averaged conversion factors and the corresponding standard deviation. Steps 73 and 74 calculate the logarithmically averaged conversion factor

and the corresponding multiplicative factor, and step 75 prints them out. Step 76 is a header; step 77 prints out the conversion factors for free-free and blackbody spectra and for power law spectra.

Steps 78 - 81 provide a notation that the program is finished (step 79) and a computation of the CPU time used (steps 80 and 81).

```
     ∇ PASSBAND
[1]    TSTRT←⌶21
[2]    AVE←CR←GV←SIG←SIGM←XY←⍳0
[3]    SIGMA←0,FLAG←0
[4]    W←0.25×⍳80
[5]    'ENTER GM TUBE:'
[6]    GM←⍞
[7]    'ENTER GM TUBE AREA, IN SQ. CM. :'
[8]    GM←⌊0.5+(1000000000)×GM÷XY←⌈/GM←GM×W×(1÷
       1.9862E¯8)×X←⍞
[9]    'ENTER ESTIMATED LIMITS, LOWER(≥.25A),UPPER(≤20A):'
[10]   X←⍞
[11]   IO←W⍳X[1]
[12]   JO←W⍳X[2]
[13]   Y← 1 2 5 10 20 50 100
[14]   'TEMP. RANGE 1-100MILL.∘ OK?(Y=YES,OR ENTER 7 COMP. V
       ECTOR):'
[15]   Y←⍞
[16]   Y←(143.8÷W)∘.÷Y
[17]   Y←*(((Y<170)×Y)+(Y≥170)×170)
[18]   FF←⌊0.5+(1000000000)×FF÷(80⍴1)∘.×X←⌈≠FF←((W*
       ¯2)∘.×7⍴1)÷Y
[19]   BB←⌊0.5+(1000000000)×BB÷(80⍴1)∘.×X←⌈≠BB←((W*
       ¯5)∘.×7⍴1)÷(Y-1)
[20]   'ENTER WAVELENGTHS OF LINE(S) (≥.25A,≤20A):'
[21]   Y←,Y←W⍳Y←⍞
[22]   'ENTER LINE/CONTINUUM RATIO(S):'
[23]   X←,X←⍞
[24]   J←1
[25] LINES:FF[Y[J];]←FF[Y[J];]×X[J]+1
[26]   BB[Y[J];]←BB[Y[J];]×X[J]+1
[27]   →((⍴Y)≥J←J+1)/LINES
[28]   CR←(XY×1E¯9)×((GM+.×FF),(GM+.×BB),(GM+.×⍉ 2
       80 ⍴(80⍴1),W*¯2))
[29]   GM←⍳0
[30] SET:X←JO-IO
[31]   IO←((IO<3)×3)+((IO>2)×IO)-(IO>8)×(4-X)×(X<4)
[32]   JO←((JO>78)×78)+((JO<79)×JO)+(IO≤8)×(4-X)×(X<
       4)
[33]   I←IO-2
[34]   J←JO-2
[35]   AMAT←SMAT←XI←XJ←⍳0
[36] ROW:XY←¯1+I+⍳(J-I)
[37]   Y←⍴XY
[38]   WT←0.5,(Y-1)⍴1
[39]   GV←((WT+.×FF[XY;]),(WT+.×BB[XY;]),(WT+.×⍉(2,Y)⍴(Y⍴1),
       W[XY]*¯2))÷CR
[40]   X←16⍴0
```

```
[41] COL:GV←GV+X
[42]  GV←GV+X←0.5×(FF[J;],BB[J;],1,W[J]*¯2)÷CR
[43]  AVE←+/GV÷16
[44]  SIG←(+/((GV-AVE)*2)÷15)*0.5
[45]  →FLAG/END
[46]  SMAT←SMAT,SIG
[47]  AMAT←AMAT,AVE
[48]  XJ←XJ,J
[49]  XI←XI,I
[50]  →((JO+2)≥J←J+1)/COL
[51]  J←JO-2
[52]  →((IO+2)≥I←I+1)/ROW
[53]  X←(SMAT÷AMAT)
[54]  SIGMA←SIGMA,SIGM←⌊/X
[55]  X←X⍳SIGM
[56]  I←XI[X]
[57]  J←XJ[X]
[58]  →(((I=IO)∧(J=JO))∨(SIGM=SIGMA[¯2+ρSIGMA]))/OUT
[59]  →(((I≠IO-2)∧(I≠IO+2)∧(J≠JO+2)∧(J≠JO-2))∨(SIGM=SIGMA[
      ¯1+ρSIGMA]))/STEP
[60]  IO←IO+(I-IO)×2
[61]  JO←JO+(J-JO)×2
[62]  →SET
[63] OUT: 5 5 ρ(SMAT÷AMAT)
[64]  '             RELATIVE ERROR ABOUT ';W[XI[13]];' AND '
      ;W[XJ[13]];'A.'
[65]  ''
[66]  'OPTIMUM  PASSBAND:  ';W[I];' TO ';W[J];' ANGSTROMS.'
[67]  FLAG←1
[68]  →ROW
[69] STEP:IO←I
[70]  JO←J
[71]  →SET
[72] END:'LIN-AVERAGE GAMMA:  ';AVE;' TO A SIGMA OF ';SIG;'
      .'
[73]  AVE←*((+/⊛GV)÷16)
[74]  SIG←*((+/((⊛GV÷AVE)*2))÷15)*0.5
[75]  'LOG-AVERAGE GAMMA:  ';AVE;' TO A MULT. OF ';SIG;'.'
[76]  'GAMMAS CALCULATED:'
[77]  2 7 ρGV; 1 2 ρGV[15],GV[16]
[78]  ''
[79]  'END'
[80]  X←⌶21
[81]  'CPU TIME:';(X-TSTRT)÷60;' SECONDS.'
```

APPENDIX D. CALCULATION OF CONVERSION FACTORS ASSUMING A KNOWN PASSBAND

The purpose of the program called KPASSBAND is to determine the conversion factor for a GM tube or an ion chamber or a set of GM tubes or ion chambers when the passband is known or assumed. The program is outlined in Figure D-1 and listed at the end of this appendix.

The operation of this program is as follows. The GM tubes are entered serially (i.e., as one long vector with one tube appended on the next). The limits are entered, and the spectra are calculated. (Temperatures may be specified as well as lines.) The incident energy flux within the specified passband is calculated.

The first tube is treated by calculating the count rate for each of the 16 incident spectra. The corresponding conversion factors are calculated, and statistics are run on these conversion factors. The next tube receives the same treatment, and so on. After the last tube is treated, the program prints out the average relative error (linearly and logarithmically) and the resulting conversion factors and uncertainties. Each conversion factor may be printed out, if desired.

In detail, the program operates as follows.

Steps 1 & 2 set initial variables.

Steps 3 & 4 call for and enter the GM tube efficiencies.

Step 5 determines the number of GM tubes inserted in step 4.

Steps 6 & 7 call for the entry of the GM tube area (assumed to be the same for all tubes), calculate 1/hc times the efficiency times the area, normalize it to an integer form accurate to one half part in 10^{10}, and define the normalization constant as NORM.

Steps 8 & 9 call for the entry of the limits, in angstroms.

Step 10 defines the vector, W, of wavelengths, in angstroms.

Steps 11 & 12 determine the indices of W corresponding to the limits entered in step 9.

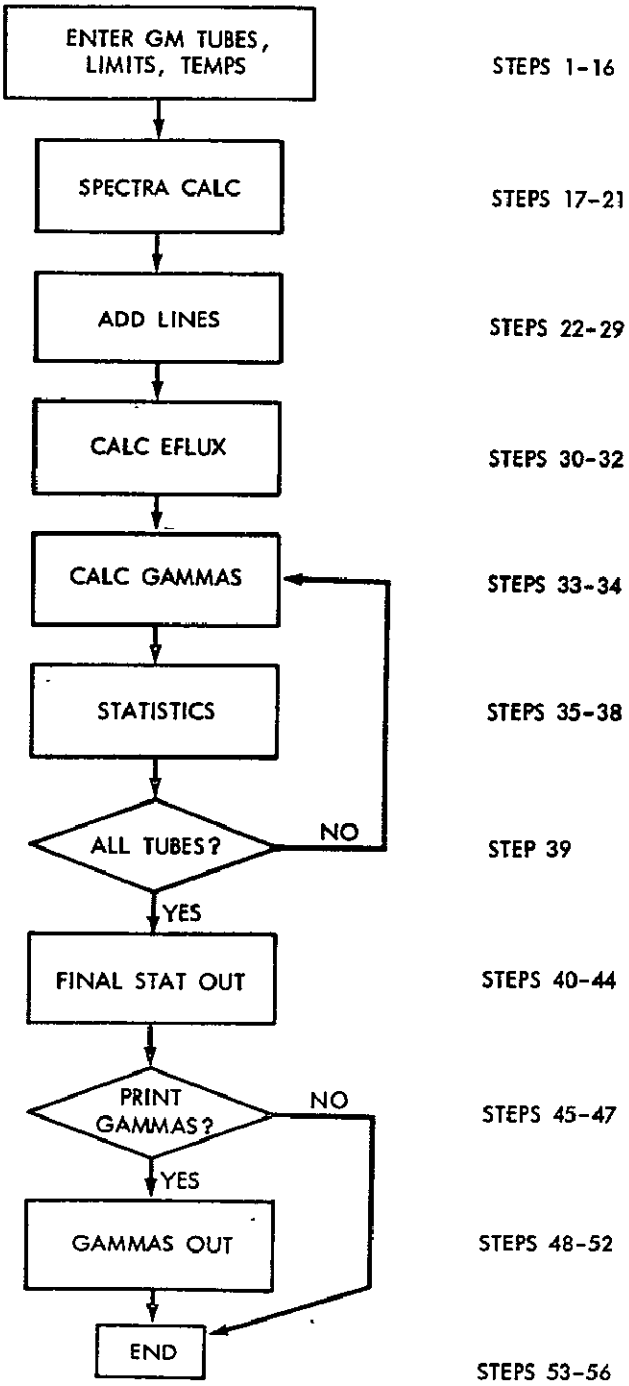

Figure D-1. Flowchart for KPASSBAND Program

Step 13 determines a vector of the indices ranging from the lower limit, I, through the upper limit, J.

Steps 14 - 16 pre-define Y equal to a range of temperatures and ask for the entry of the temperatures wanted.

Step 17 defines Y = hc/WkT.

Step 18 limits Y to values ≤ 170 and then takes the exponent to the base e of Y.

Steps 19 & 20 define the free-free and blackbody continuum spectra. These spectra are normalized to the largest value for each temperature and then converted to an integer form accurate to one half part in 10^8.

Step 21 defines the power law spectra in integer form.

Steps 22 - 25 ask for the entry of the line wavelengths and corresponding line/continuum ratios.

Steps 26 - 29 form a loop which replaces the old spectra values (free-free and blackbody only) at wavelengths which include lines with the old spectral flux value times the sum of the line/continuum ratio and 1.

Steps 30 - 31 define a weighting function, WT, such that the two end increments of the energy flux integral are weighted one half the weight of the other intervals, and define an energy flux vector (16 components, one for each spectrum) using the spectra and the weighting function. The expressions within the parentheses are equivalent to (using the leftmost expression as an example):

$$(WT +.\times FF[XY;]) = \sum_I WT_I \cdot FF_{IJ}$$

Steps 32 - 39 form a loop which calculates the statistics for each individual tube. Step 33 defines GM equal to the 80-component efficiency vector of the Jth tube, times the wavelength, and converted out of normalized units. Step 34 defines the vector GV equal to the energy flux, EFLUX, divided by the count rate (the expressions in parentheses) and appends this vector onto the vector GAM. Steps 35 and 36 calculate the logarithmic average and standard deviation, and steps 37 and 38 calculate the linear average and standard deviation of the resultant conversion factors. Step 39 increments J, tests it, and branches the program to step 33 if J is less than or equal to the number of tubes.

Steps 40 - 44 print out the results of the above calculations.

Steps 45 - 52 inquire if all calculated conversion factors are wanted and, if not, branch to step 53. Otherwise, the program prints out all conversion factors calculated for each tube.

Steps 53 - 56 announce the end of the program and print out CPU time.

```
    ∇ KPASSBAND
[1]   TSTRT←⌶21
[2]   GM←GAM←SIG←LSIG←AVE←LAVE←EFLUX←GV←XY←⍳0
[3]   'ENTER GM TUBES:'
[4]   GMTB←⍞
[5]   N←(⍴GMTB)÷80
[6]   'ENTER GM TUBE AREA, IN SQ. CM. :'
[7]   GMTB←⌊0.5+(10000000000)×GMTB÷NORM←⌈/GMTB←GMTB×(1÷
      1.9862E¯8)×X←⍞
[8]   'ENTER LIMITS, LOWER(≥.25A),UPPER(≤20A):'
[9]   X←⍞
[10]  W←0.25×⍳80
[11]  I←W⍳X[1]
[12]  J←W⍳X[2]
[13]  XY←¯1+I+⍳(1+J-I)
[14]  Y← 1 2 5 10 20 50 100
[15]  'TEMP. RANGE 1-100 MILL. ∘ OK? (Y=YES, OR ENTER 7 COM
      P. VECTOR):'
[16]  Y←⍞
[17]  Y←(143.8÷W)∘.÷Y
[18]  Y←*(((Y≥170)×170)+(Y<170)×Y)
[19]  FF←⌊0.5+(100000000)×FF÷(80⍴1)∘.×X←⌈⌿FF←((W*¯2)∘.×
      7⍴1)÷Y
[20]  BB←⌊0.5+(100000000)×BB÷(80⍴1)∘.×X←⌈⌿BB←((W*¯5)∘.×
      7⍴1)÷(Y-1)
[21]  Z←⍉2 80⍴(80⍴1),X÷Y←⌊/X←W*¯2
[22]  'ENTER LINE WAVELENGTH(S) (A):'
[23]  Y←,Y←W⍳Y←⍞
[24]  'ENTER LINE/CONTINUUM RATIO(S):'
[25]  X←,X←⍞
[26]  J←1
[27] LINES:FF[Y[J];]←FF[Y[J];]×X[J]+1
[28]  BB[Y[J];]←BB[Y[J];]×X[J]+1
[29]  →((⍴Y)≥J←J+1)/LINES
[30]  WT←0.5,((¯2+⍴XY)⍴1),0.5
[31]  EFLUX←(WT+.×FF[XY;]),(WT+.×BB[XY;]),(WT+.×Z[XY;])
[32]  J←1
[33] RERUN:GM←GMTB[((J-1)×80)+⍳80]×W×NORM×
      1E¯10
[34]  GAM←GAM,GV←EFLUX÷((GM+.×FF),(GM+.×BB),(GM+.×Z))
[35]  LAVE←LAVE,*(+/(⍟GV)÷16)
[36]  LSIG←LSIG,*((+/((⍟(GV÷LAVE[⍴LAVE]))*2)÷15)*
      0.5)
[37]  AVE←AVE,(+/GV÷16)
[38]  SIG←SIG,(+/((GV-AVE[⍴AVE])*2)÷15)*
      0.5
[39]  →(N≥J←J+1)/RERUN
[40]  'OVERALL UNCERTAINTY: LINEAR RELATIVE ERROR-';+/(SIG÷
      AVE)÷N;'  LOGARITHMIC  SIGMA-';+/(LSIG÷N);'.'
```

```
[41]  'RESULTING CONVERSION FACTORS AND UNCERTAINTIES:'
[42]  ⍉((5,N)⍴(⍳N),AVE,SIG,LAVE,LSIG)
[43]  '              LIN GAMMA,SIGMA        LOG GAMMA,SIGMA'
[44]  ''
[45]  'PRINT OUT ALL GAMMAS?(1=YES):'
[46]  Y←⍞
[47]  →(Y≠1)/END
[48]  J←1
[49] NXT:'TUBE ';J;':'
[50]  2 7 ⍴GAM[((J-1)×16)+⍳14]; 1 2 ⍴GAM[((J-1)×16)+
      15],GAM[((J-1)×16)+16]
[51]  ''
[52]  →(N≥J←J+1)/NXT
[53] END:''
[54]  'END'
[55]  X←⌶21
[56]  'CPU TIME:';(X-TSTRT)÷60;' SECONDS.'
```

APPENDIX E. RESPONSE BEHIND A CONICAL APERTURE

This appendix describes a program designed to determine the relative response of a GM tube (or ion chamber) located behind a conical aperture. The program includes calculations determining the shadowing of the GM tube window by the aperture. The name of the program is CONE. The program called CONEIN generates the input for CONE.

The CONEIN program is flowcharted in Figure E-1A. CONE is flowcharted in Figure E-1B. The GM tube absorption and efficiency are entered with the GM tube geometry (window area and gas path length), spectra are determined for future use, the aperture geometry (Figure E-2) is entered, and the needed angles are calculated. The overall approach involves placing a 6-by-6 grid of squares over the face of the GM tube. The squares are dimensioned such that the four corner squares lie outside the radius of the window (Figures E-3 and E-4). The area and the center of gravity for each square are calculated. The aperture-shadowing problem then becomes a problem of determining how much of each areal element is left unshadowed and what the approximate center of gravity is, so that the efficiency of each element can be calculated, weighted by the area of that element, and then all elements summed to determine the final effective efficiency.

The program CONEIN is described in detail as follows.

Step 1 clears a number of variables.

Steps 2 - 6 require the entry of the gas absorption of the GM tube and the efficiency of the GM tube and calculate the window transmission.

Steps 7 - 11 determine values $hc/\lambda kT$. Step 7 sets up the standard temperature range required for step 9. Step 10 determines $hc/\lambda kT$, and step 11 checks the maximum limits and then exponentiates.

Steps 12 - 24 require the entry of the GM tube geometry and the aperture geometry (Figure E-2) and define variables R1 and R2.

Steps 25 - 26 require the entry of the off-axis angles wanted.

Step 27 determines the off-axis angles needed (Figure E-5).

Steps 28 - 30 determine the areal elements and the dimensions shown in Figure E-4.

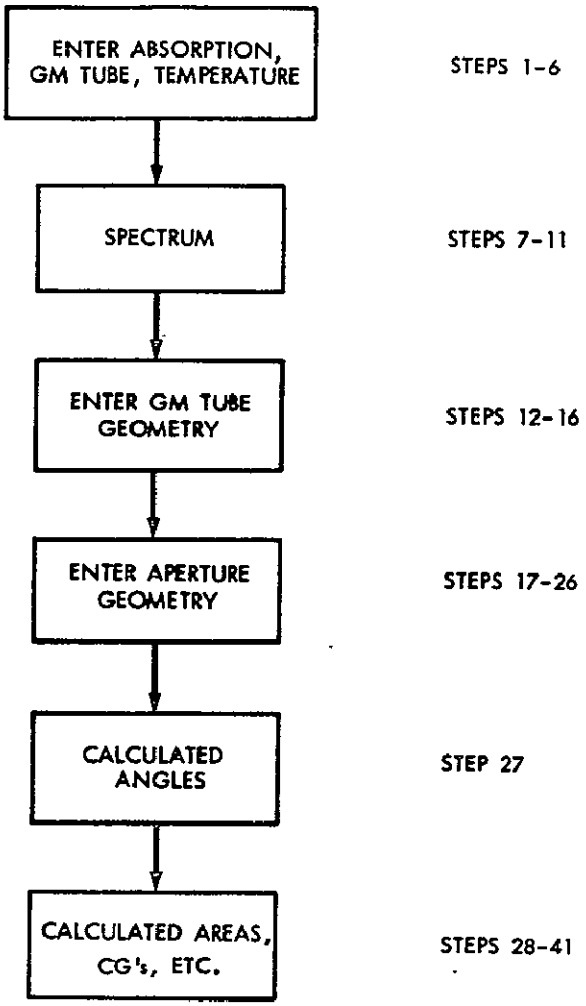

Figure E-1A. Flowchart for CONEIN Program

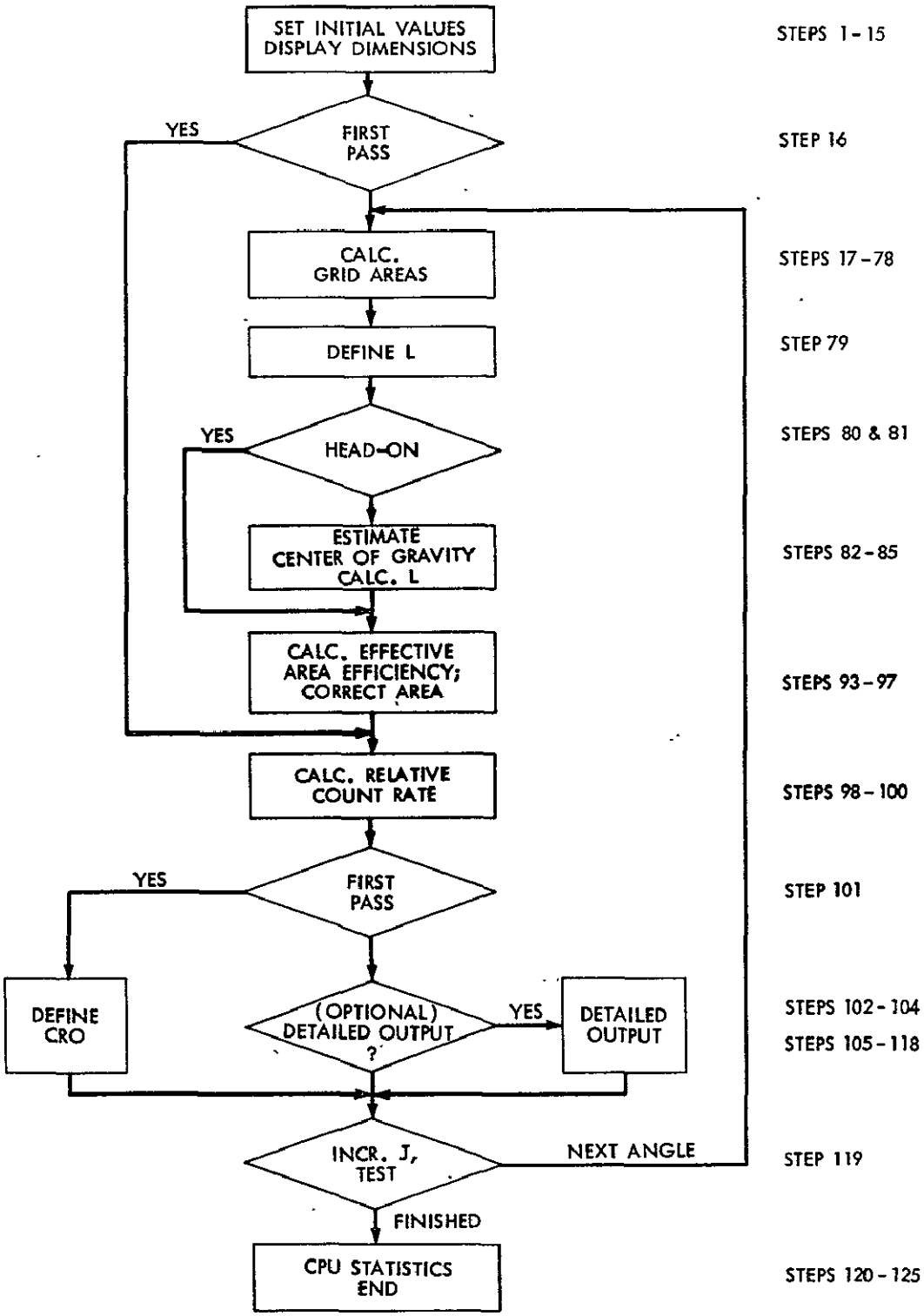

Figure E-1B. Flowchart for CONE Program

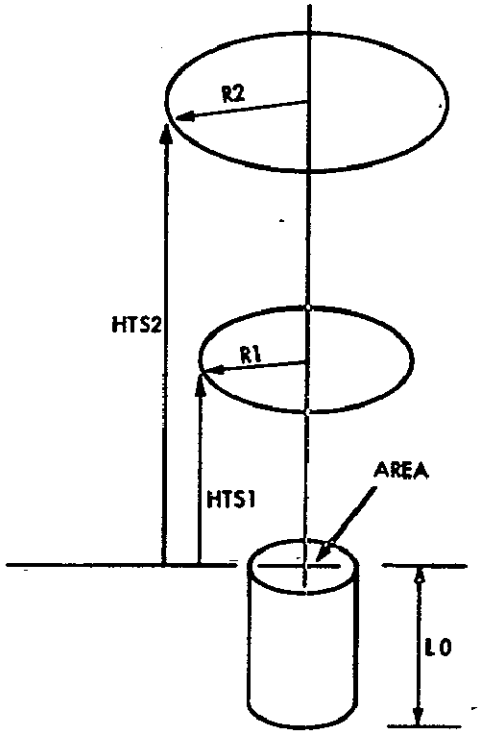

Figure E-2. Aperture Geometry for the CONEIN Program

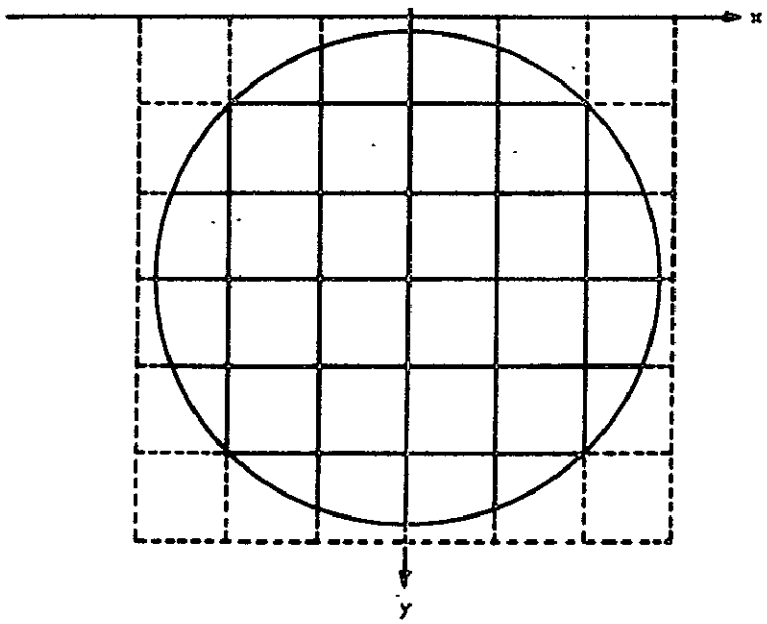

Figure E-3. Grid Scheme for the CONEIN Program

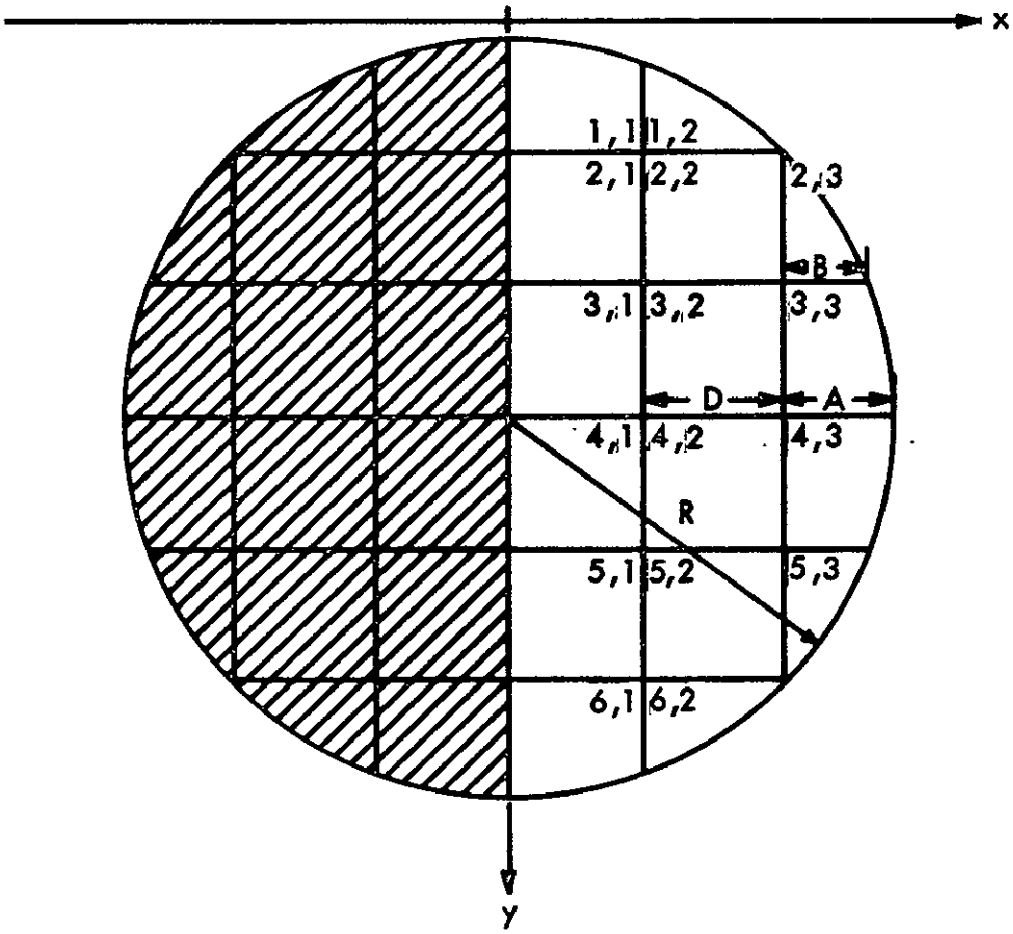

Figure E-4. Dimensions of Grid Network Used in CONEIN and CONE Programs

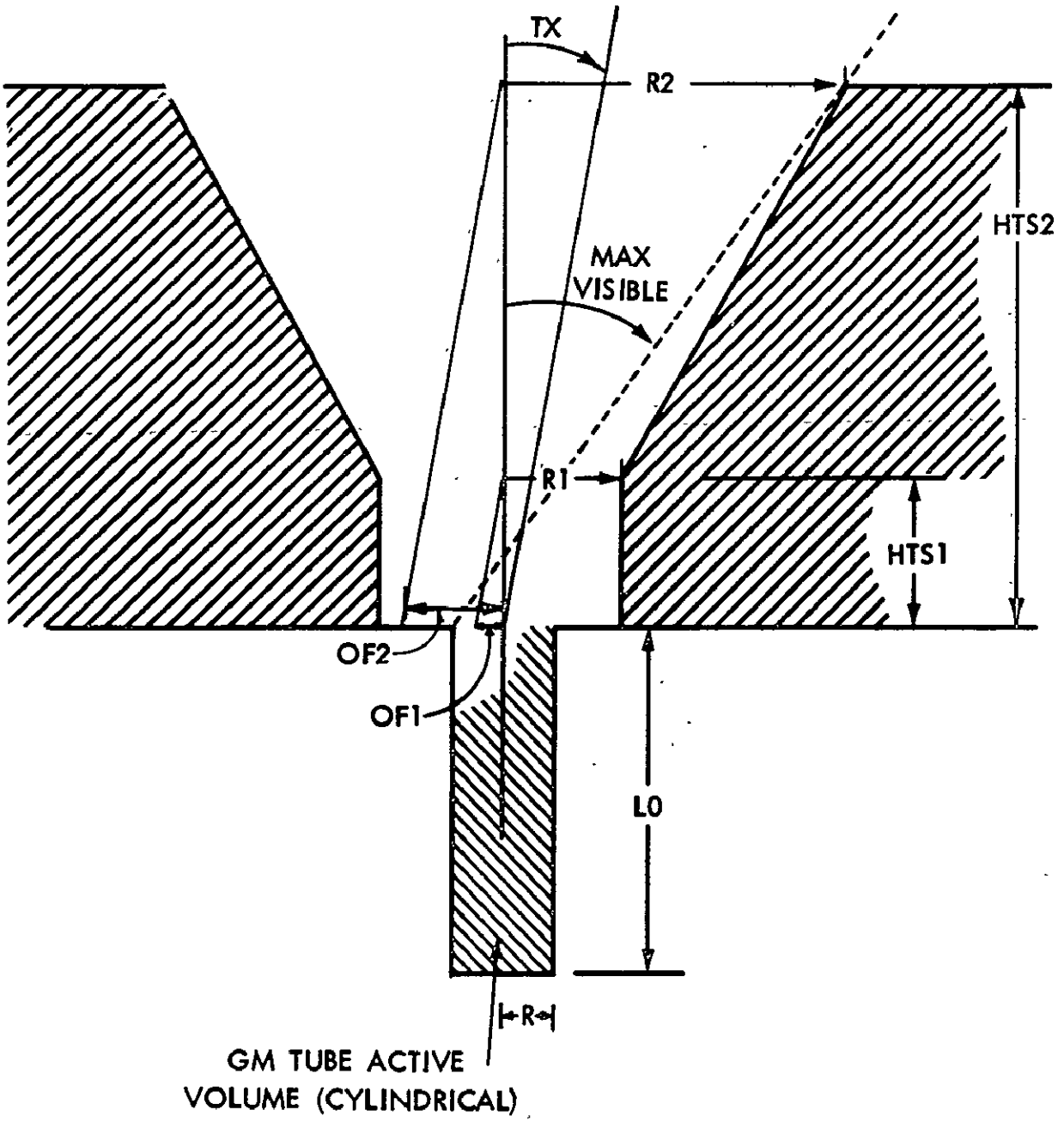

Figure E-5. Side View Illustrating Aperture Geometry and Defining Angles

Steps 31 - 36 determine the X and Y (Figure E-4) coordinates of the center of gravity of the areal elements.

Steps 37 - 39 determine the differentials of the X and Y center-of-gravity matrices (i.e., the distance between the center of gravity and the edge of the areal element).

Steps 40 - 41 serve to define variables and announce the end of the program.

The operation of program CONE is as follows. The window of the GM tube is overlaid with a 6-by-6 grid of squares such that the four corner squares are outside the window proper (Figure E-4). The amount of shadowing of each square is individually determined, and the shift in the center of gravity of the area of each square is approximated. This effective center of gravity is later used to estimate the effective gas path length. Because of the symmetry of the conical aperture and cylindrical GM tube system, only half of the areas of the squares need be determined (Figure E-4). The uniformity of the incident flux of X-ray photons across the face of the aperture is tacitly assumed. Thus, the product of the area and the efficiency is actually calculated.

In detail, the program CONE operates as follows.

Steps 1, 2, 121, 122, 123, 124, and 125 generate statistics on central processor time (typically 100 seconds on an IBM 360/50), turnaround time (typically the order of minutes), and the number of users on the system.

Steps 3 - 9 set initial values for variables. Step 5 determines the value of the area times efficiency, AXE, were the GM tube viewing the source head-on with no aperture. Step 6 defines the wavelengths used. Steps 7 through 9 refer to geometric constants (defined in Figure E-5).

Steps 10 - 15 list the aperture dimensions used for this particular run and ask if stopping the program or obtaining detailed output is desired. The geometry and the dimensions are defined in Figure E-5.

Step 16 routes the program to step 98 if J = 0. This is done to determine the relative count rate (a 16-element vector, one component for each of 16 spectra--seven free-free, seven blackbody, and two power law) if the GM tube is viewing the source head-on.

Steps 17 - 19 allow the addition of input to the program if additions were requested.

Steps 20 - 22 determine offset values (Figure E-5).

Step 23 sets the relative area matrix, AR, equal to a matrix of zeros. This sets up the program for the condition of total shadowing.

Step 24 determines if the aperture is totally shadowed. This condition occurs when the projection of either aperture on the plane of the GM tube window does not overlap the window (Figure E-6) or when the projections of both apertures on the plane of the tube window are mutually exclusive (Figure E-7). If the window is completely shadowed, the program branches to step 79.

Step 25 redefines the relative area in terms of no shadowing.

Step 26 determines if the window is shadowed (Figure E-8). If the window is not shadowed, the program branches to step 79.

Steps 27 - 29 set values for variables.

Steps 30 & 31 determine if only the lower mask, R1, effectively masks the tube window (Figure E-9); if that condition is true, step 31 branches the program to step 38.

Step 32 branches the program to step 47 if only the upper mask effectively shadows the tube window (Figure E-10).

Steps 33 - 36 determine if the intersection of the projections of the two masks on the plane of the window lies outside the radius of the window (Figure E-11). Steps 33 and 34 calculate X1 and X2 from the law of cosines (Figure E-11). Steps 35 and 36 test whether $X1 \leq X2$, and, if so, branch the program to step 47. This series of steps differs from step 31 in that it tests for different conditions.

Steps 37 & 38 calculate the angles X1 and MAX, defined in Figure E-12, using the law of cosines.

Step 39 is similar to step 36. It branches the program to step 47 when $X1 \geq MAX$; i.e., the mask of radius R2 does all of the effective shadowing.

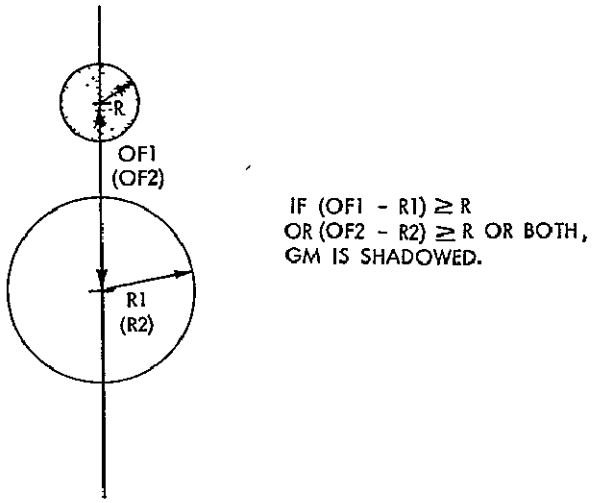

Figure E-6. Condition I for Total Shadowing

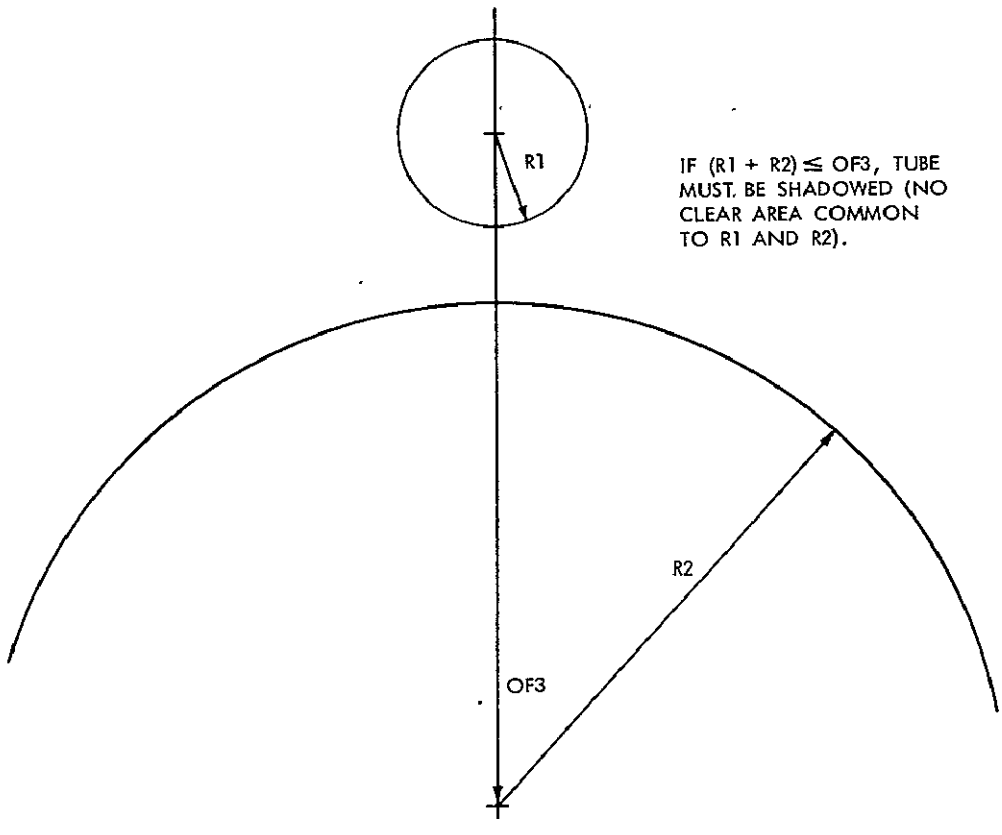

Figure E-7. Condition II for Total Shadowing

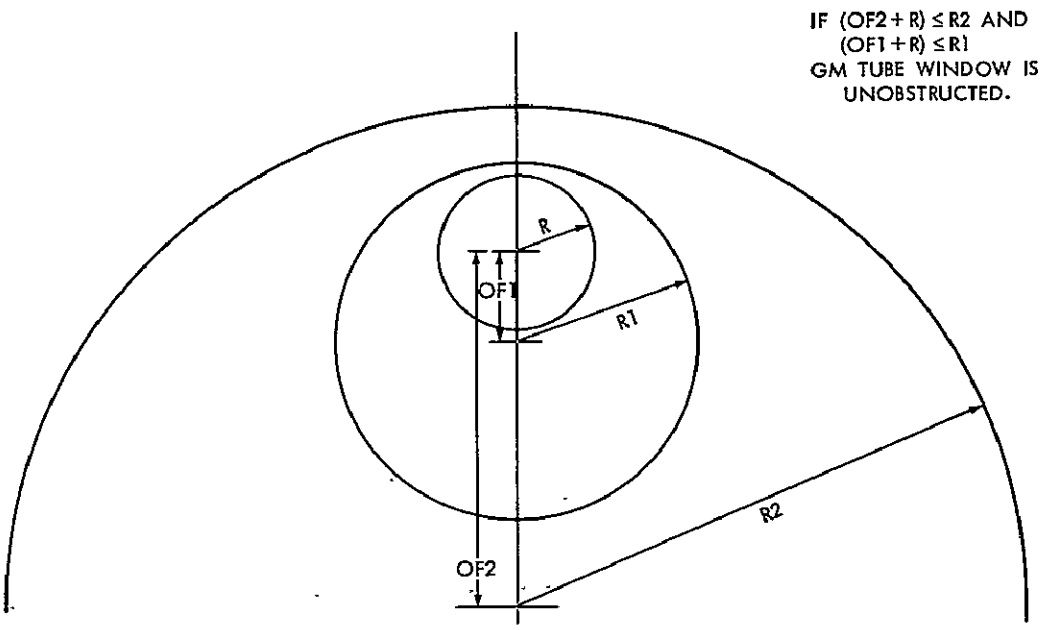

Figure E-8. Condition for No Shadowing

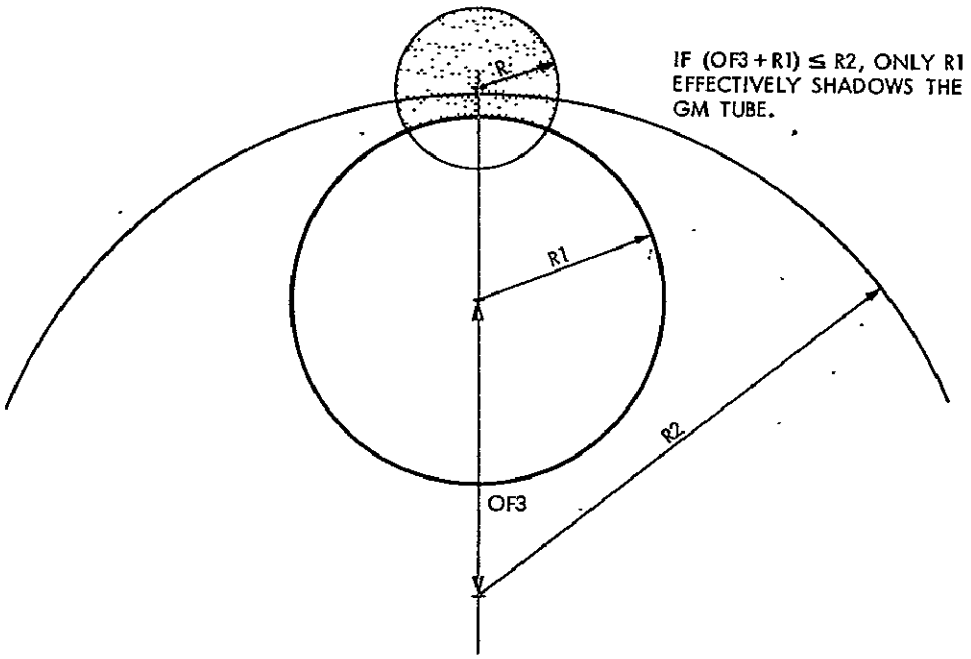

Figure E-9. Condition for Only Lower Mask to Shadow the GM Tube Window

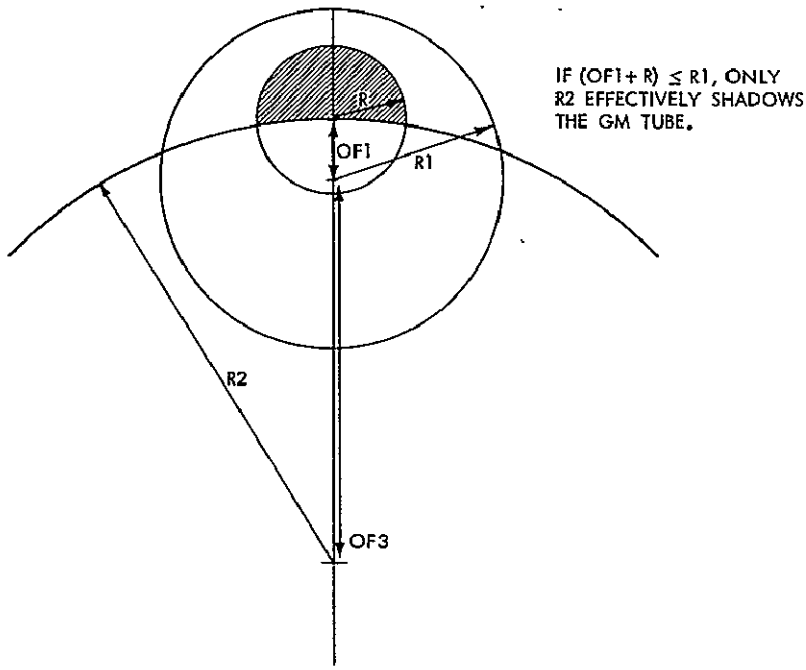

Figure E-10. Condition for Only the Upper Mask to Shadow the GM Tube Window

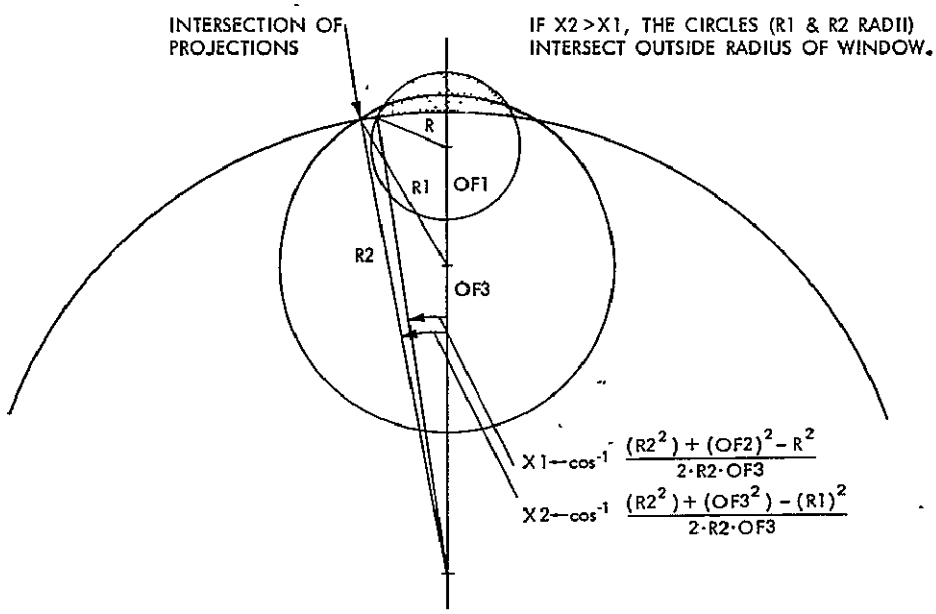

Figure E-11. Condition for the Intersection of the Projections of the Masks Lying Outside the Radius of the GM Tube Window

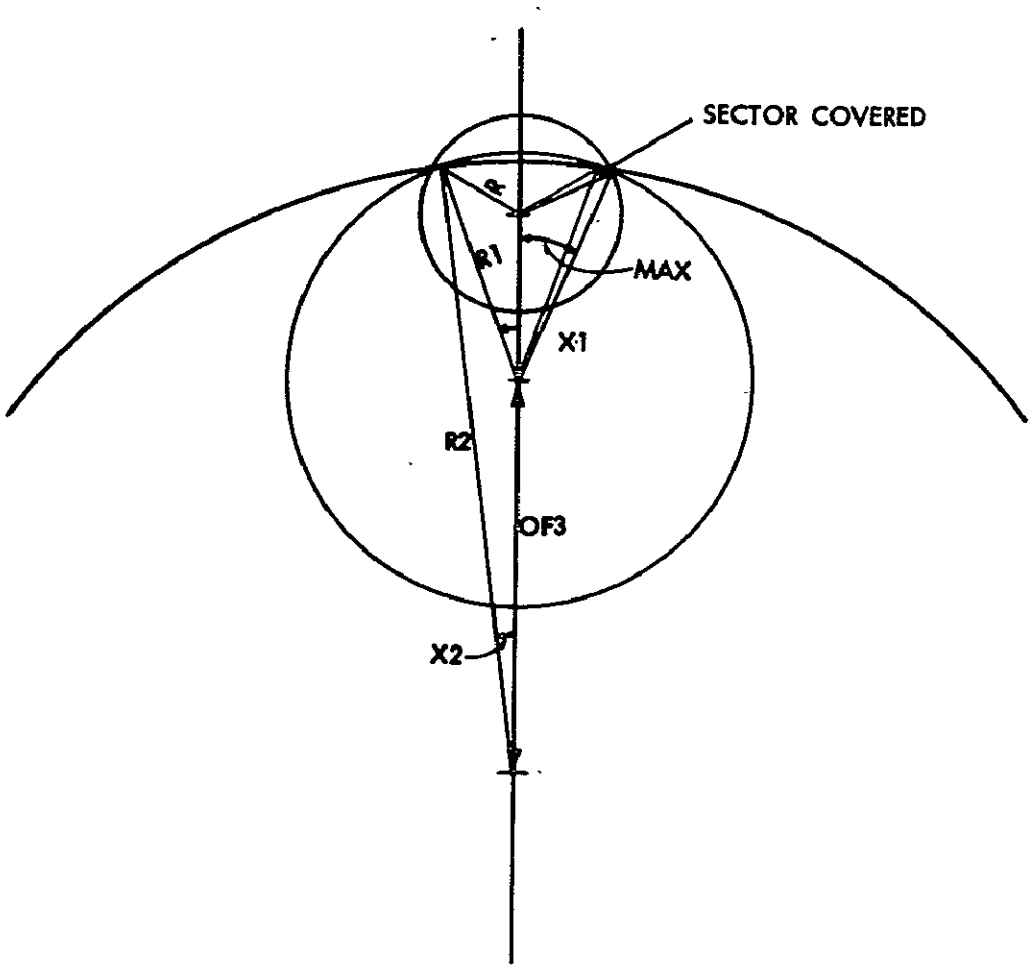

Figure E-12. Geometry Defining Angles X1 and MAX (Steps 37 and 38)

Step 40 determines the angles between the offset line, OF1, and radii drawn to the intersections of the grid lines and the edge of the tube window (Figure E-13). The expression to the right of the comma determines the angles of the intersections with the horizontal grid lines; the expression to the left of the comma determines the angles of the intersections with the vertical lines.

Steps 41 - 43 order the angles in ascending order, truncate the vector of ordered angles so that it includes only angles less than MAX, and then truncate angles less than X1.

Steps 44 & 45 define vectors with the number of components equal to that of the vector ANG and of values of the radius R1 (RAD) and OF1 (OFST).

Step 46 branches the program to step 58 if the mask of radius R1 is the only mask shadowing the window.

Step 47 is the same as step 46 and covers the geometry shown in Figure E-14.

Step 48 is the same as step 40, but the upper mask (of radius R2) is used rather than the lower mask (Figure E-13).

Step 49 orders the elements of vector A2 in ascending order.

Step 50 calculates the angle MAX illustrated in Figure E-15. This is the angle at which the outer edge of the window intersects the projection of the upper mask.

Step 51 truncates the vector A2 so it includes values less than and equal to MAX.

Steps 52 & 53 branch the program to step 55 if only the upper mask shadows the window.

Step 54 truncates the vector A2 to values less than or equal to X2.

Step 55 redefines the vector RAD. The old vector RAD contains components of values equal to R1, with each component corresponding to an element in the vector ANG. RAD is appended on the end of a vector of components equal to R2, each corresponding to a component of A2. The resulting vector is called RAD.

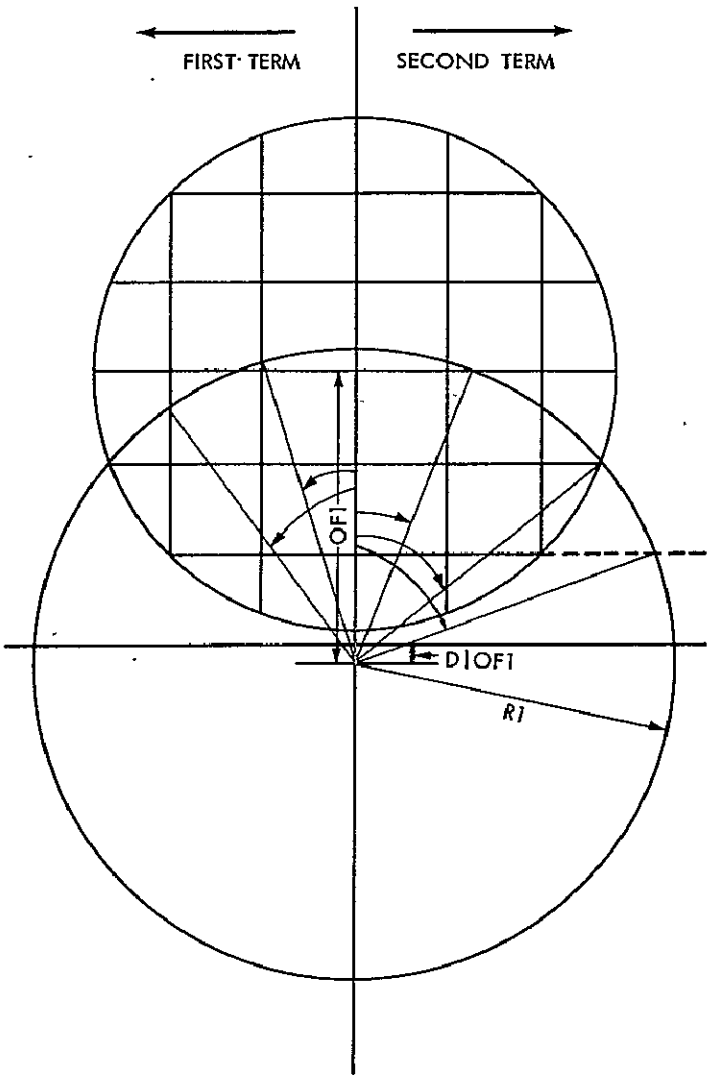

Figure E-13. Geometry Showing the Intersections Between the Grid Lines and the Projection of the Lower Mask

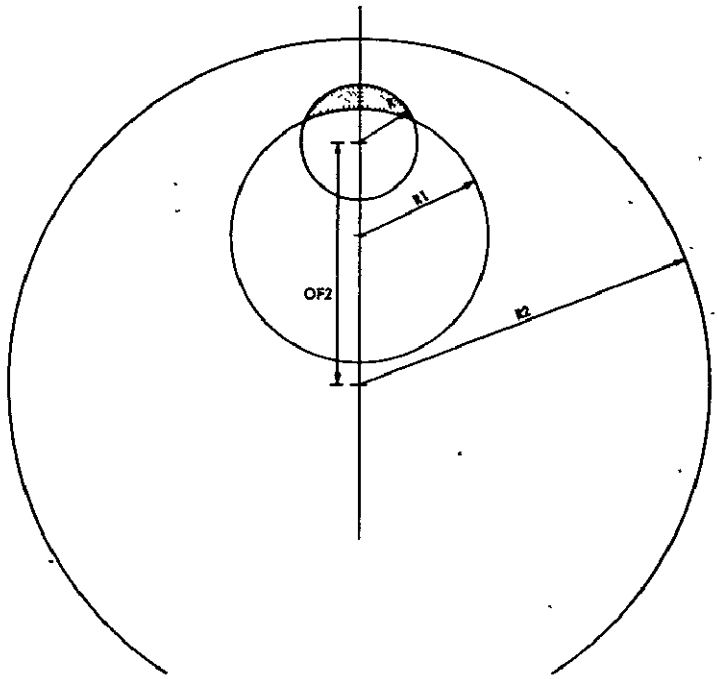

Figure E-14. Geometry for Shadowing by Only Lower Mask

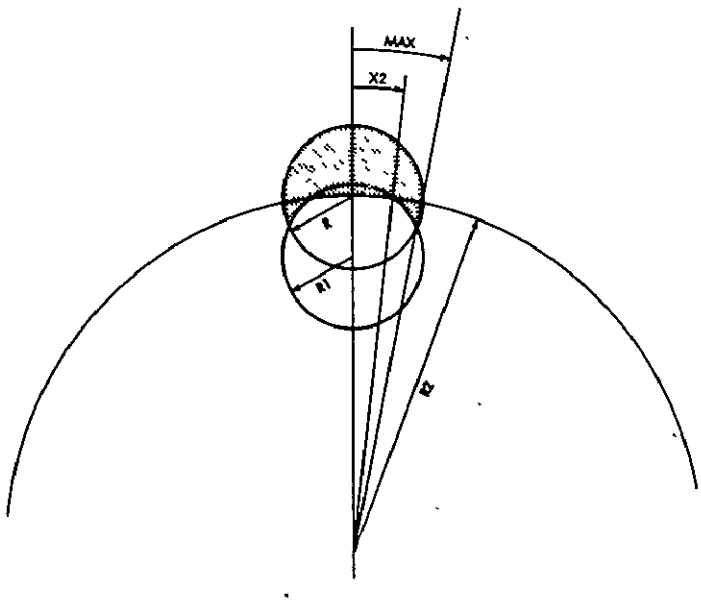

Figure E-15. Definition of the Angle MAX as the Intersection of the Projection of the Lower Mask and the GM Tube Window

Step 56 operates similarly to step 55 except that corresponding values of OF1 or OF2 are used.

Step 57 determines a new vector of angles, ANG, consisting of old vector ANG appended to the end of the vector A2.

Step 58 calculates a vector of values consisting of segments of circles (Figure E-16). These segments correspond to the angles and radii contained in ANG and RAD, respectively. The equation for the area of a segment, seg, is

$$seg = 1/2 \cdot (r^2) \cdot ((2 \cdot \text{half-angle}) - \sin(2 \cdot \text{half-angle}))$$

Steps 59 & 60 calculate the coordinates of the intersections of the grid and radii R1 and R2 based on the system shown in Figure E-4.

Step 61 branches the program to step 64 if the largest value of $Y > (D - B)$. This means that if only grid 1,1 was shadowed (Figure E-4), the program does not branch to step 64.

Step 62 sets the area of grid 1,1 equal to the totally unshadowed value, ARO (1;1), minus the value of segment (2). This result follows because seg (1) = 0, and seg (2) corresponds to the segment that crosses the edge of the window.

Step 63 branches the program to step 79.

Step 64 sets the index K = +1.

Steps 65 & 66 define IY equal to the value of Y(1) divided by D, rounded off to the next higher integer, and branch the program to step 68 if IY = 1.

Step 67 sets the values of all components of the area matrix, AR, which are completely shadowed equal to 0. This step requires that IY be greater than 1.

Step 68 branches to step 75 if ANG(K) = X2.

Steps 69 - 71 define IY and IX as Y(K + 1) and X(K + 1) divided by D and rounded off to the next higher integer and, if IX = 3, branch the program to step 73.

Step 72 sets the areas of grids lying outside the grid in question, but on the same row, equal to 0.

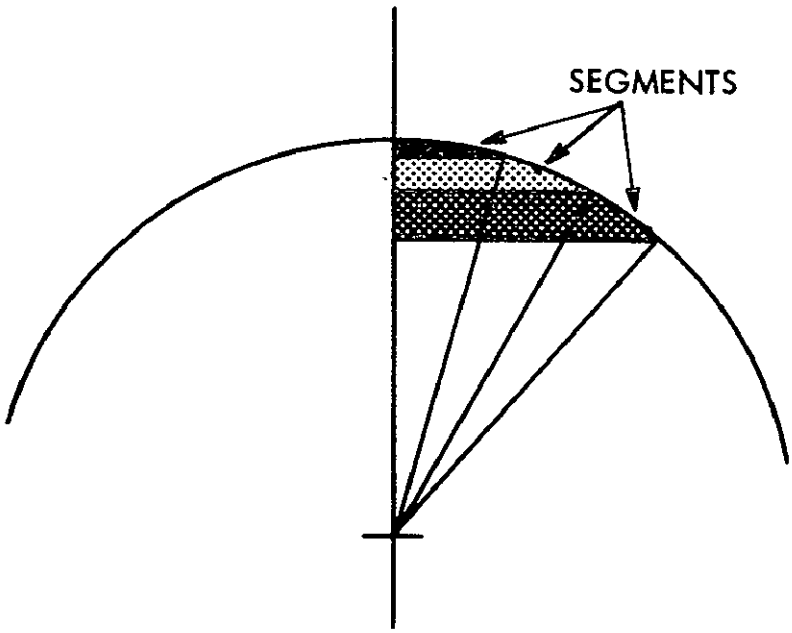

Figure E-16. Definition of Segments

Step 73 defines the dummy variable MAX as the area of a grid (times 2, since by symmetry only half the grids need be calculated) between its maximum Y values (the lesser of either the outer limit of the window (IY · D or R + (3 · D)) and the value of Y(K + 1) provided that the present value of K + 1 does not indicate the transition case in which the shadows from the two masks cross and provided that K + 1 is not the last angle. Otherwise, MAX = 0.

Step 74 determines the area of the grid. This area equals the difference between the areas of the segments K and K + 1 (resulting in a strip across the face of the tube) running perpendicular to the Y axis), plus the prior value of the grid if ANG(K) ≠ X2, plus MAX, and minus a rectangle laid on the strip which has a height equal to (Y(K + 1) - Y(K)) and a length equal to (IX - 1) · D · 2. This step leaves the areas shown in Figures E-17 and E-18.

Step 75 increments K and, if K is still less than the number of components in the vector ANG, branches the program to step 68. Advancing to the last intersection drops the program out of this loop.

Step 76 defines MAX as a vector of two angles (Figure E-19).

Step 77 sets MAX as a vector of the corresponding areas of the segments of the previous vector MAX.

Step 78 determines the area of the last grid which needs calculation. This is the grid which contains the intersection between the shadow and the edge of the window. This area is equal to the previously calculated area, plus the difference between the two segments calculated as MAX (absolute value), minus the area of rectangle contained in the difference of the two segments. See Figure E-19.

Step 79 determines the variable L to be a 6-by-3 matrix of 1's.

Steps 80 & 81 branch the program to step 93 if J = 0 or TX(J) = 0.

Steps 82 & 83 determine approximate center-of-gravity matrices for the Y(CGY) and X(CGX) directions using initial centers of gravity (CGY, CGX) and a fraction of the difference between the center of gravity and the edge of a grid (CGYD, CGXD). A simple linear change was used as an approximation in Figure E-20.

A: $[SEG(K+1) - SEG(K)] - [Y(K+1) - Y(K)] \times 2 \times D \times (IX - 1)$

B: $\left([(IY \times D) \lfloor (R+3 \times D)] - [Y(K+1)]\right) \times 2 \times D$

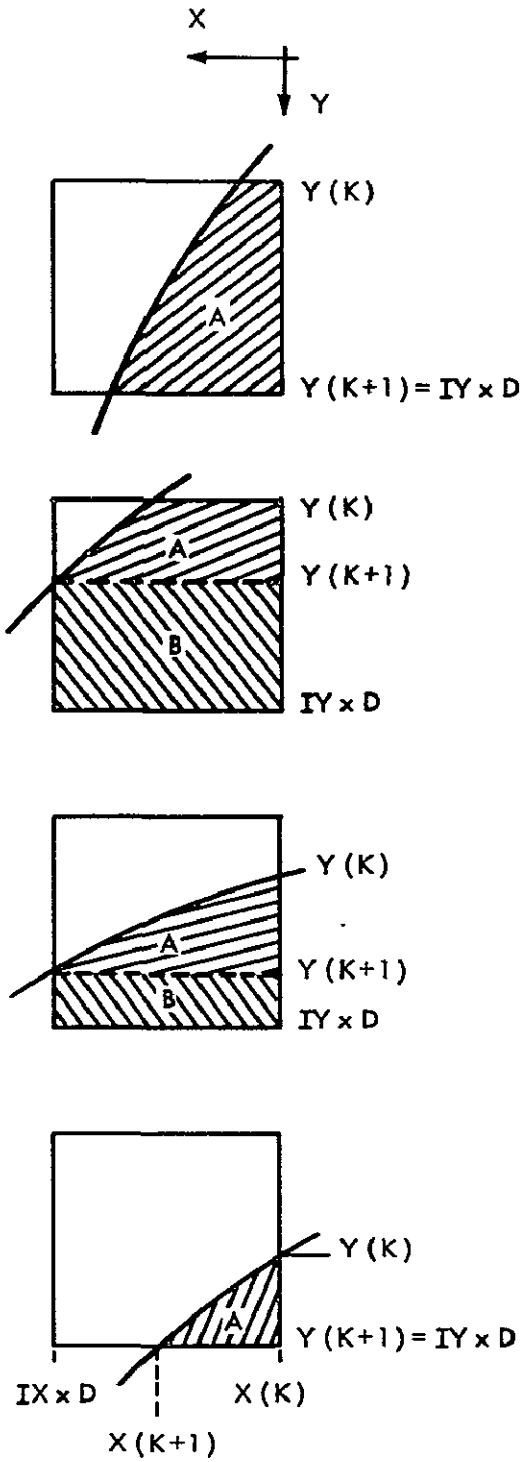

Figure E-17. Areas of Grids Intersected by One Mask

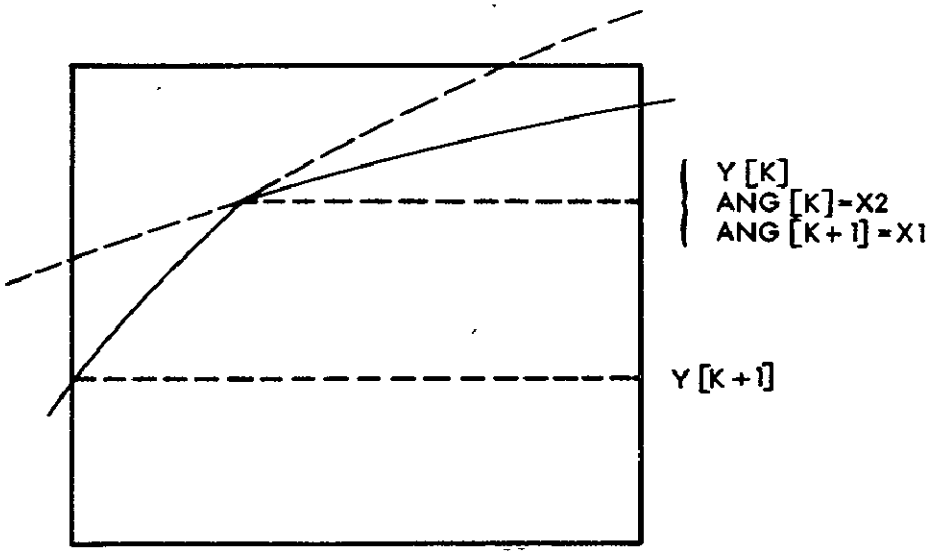

Figure E-18. Area of the Grid Intersected by Both Masks

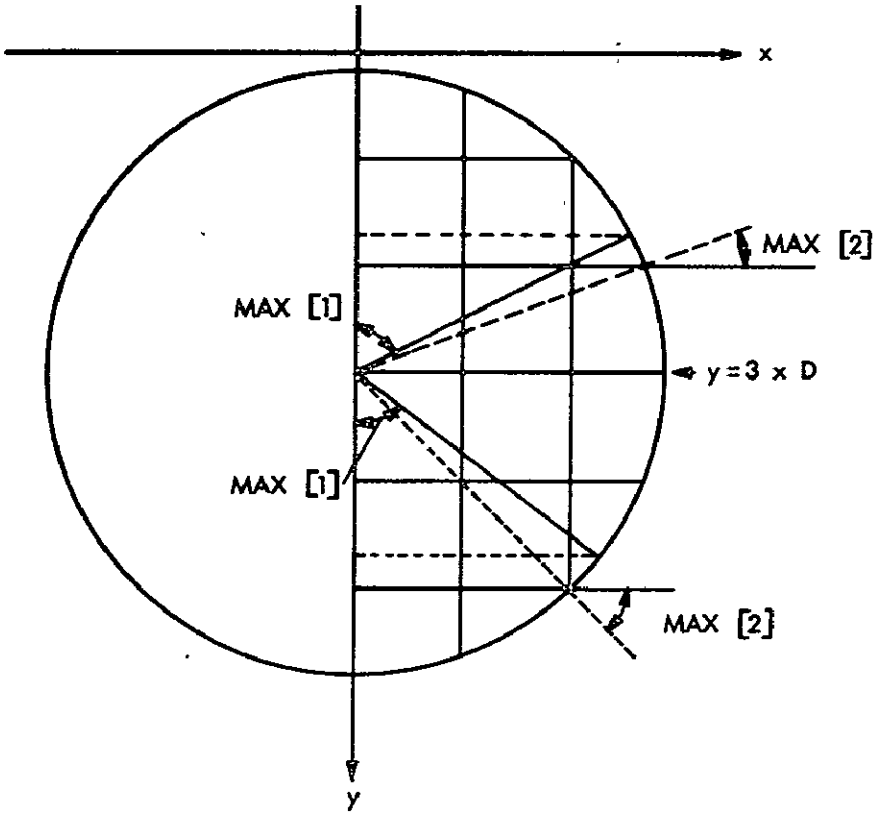

Figure E-19. Definitions of MAX 1 and MAX 2 (2 Cases)

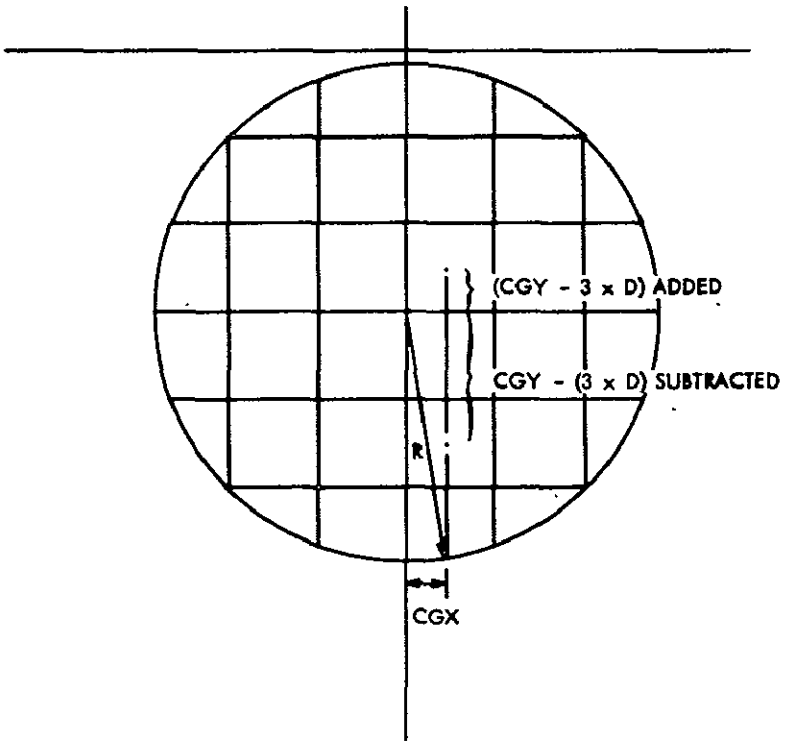

Figure E-20. Definitions of Points of Centers of Gravity

Step 84 defines L as a 6-by-3 matrix of the effective gas path lengths, in units of LO. (See Figure E-21.)

Step 85 sets L equal to the minimum of either L defined in step 84 or the secant of TX(J). (See Figure E-22.)

Steps 86 - 91 determine that if optional program stopping is desired the program stops (steps 86, 87, 88), asks what is desired, and then goes on. If stopping is not desired, the program branches to step 92.

Step 92 erases variables no longer needed.

Step 93 defines X as a vector (each element corresponding to a different wavelength) of the window transmission. The window transmission is defined as the initial transmission to the secant TX(J) power because the change in path length occurs in the exponential argument of the expression for the transmission (i.e., $T = \exp(\mu/\rho)\rho x \sec TX(J)$).

Steps 94 - 96 define the effective area X efficiency, AXE. This is equal to the sum, for each grid, of the window transmission (X, defined in step 93), times the area of the grid, times 1 minus the gas transmission to the L power.

Step 97 corrects for area foreshortening due to off-axis look-angles.

Steps 98 - 100 calculate a 16-component vector of the relative counting rates for free-free, blackbody, and power law spectra. Step 98 is equal to

$$CR = \sum_{I} AXE_I \cdot W_I \cdot T_{IJ}$$

where

$$T_{IJ} = \exp -\left(hc/\lambda_I kT_I\right)$$

Step 99 appends onto the 7-element vector CR the vector

$$\frac{AXE_I \cdot W^{-4}}{T_{IJ} - 1}$$

Step 100 appends onto CR the vectors

$$(AXE_I \cdot W_I) \text{ and } (AXE_I \cdot W_I) \cdot W^{-2}$$

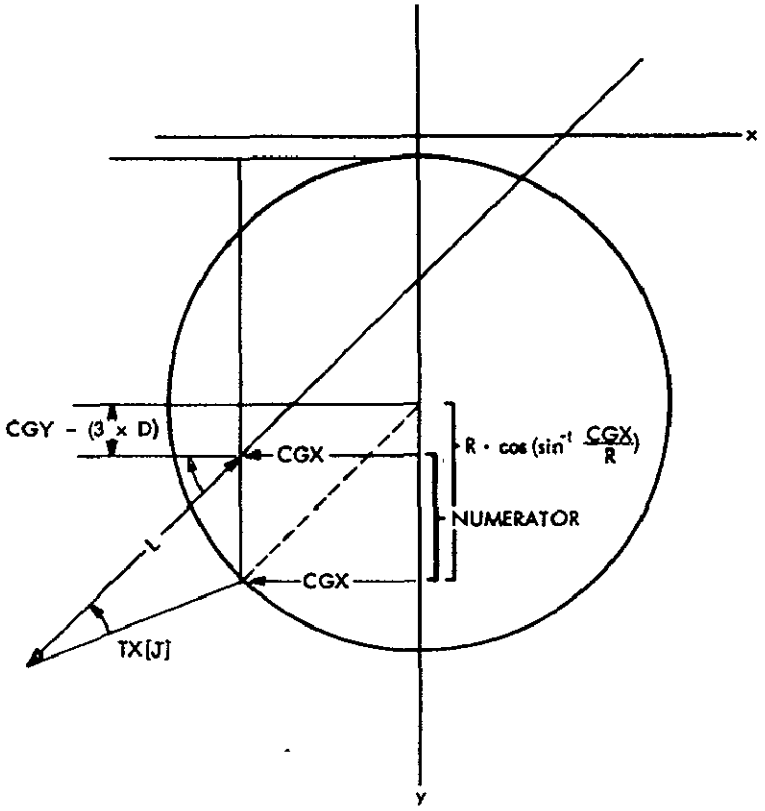

Figure E-21. Geometry for the Effective Gas Path (One of Two Cases)

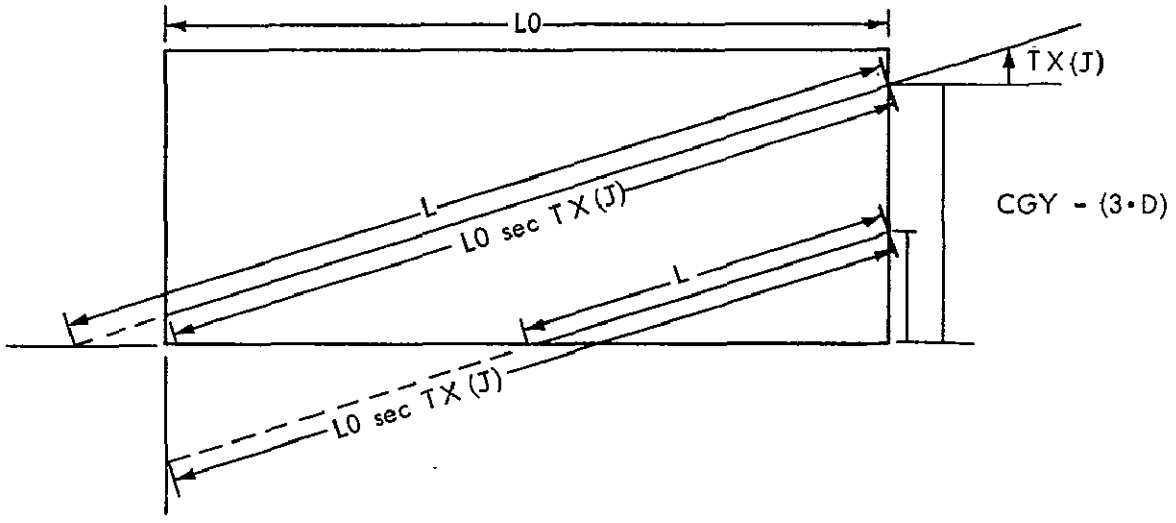

Figure E-22. Determining the Actual Gas Path Length from the Two Cases (LD SEC TX(J) or L)

Step 101 branches the program to step 104 if $J \neq 0$.

Step 102 defines CRO as equal to CR.

Step 103 branches the program to step 119.

Step 104 defines the relative counting rate, RC, as CR/CRO.

Step 105 sets the Jth component of the vector RCR equal to the average value of RC, or

$$\sum_{I} \frac{RC_I}{16}$$

Steps 106 - 118 branch the program to step 119 if the option (steps 14 and 15) is not desired; otherwise, the program asks if a detailed output is desired. If the answer is 1 (yes), the program prints out a list including the angle TX(J) (in degrees), the relative area (compared to looking head-on), the relative counting rates determined for each of the 16 spectra, and the linearly and logarithmically averaged relative count rates and their standard deviations.

Step 119 branches the program to step 17 if, after incrementing J, J is not larger than the number of angles.

Steps 120 - 125 announce that the program is finished and determine time statistics.

```
    ∇ CONEINPUT
[1]   RCR←TX←AROO←CGXO←CGYO←CGXD←CGYD←⍳0
[2]   'ENTER GAS ABSORPTION:'
[3]   TRGAS←1-X←⍞
[4]   'ENTER GM TUBE:'
[5]   X←⍞
[6]   TRWIN←X÷1-TRGAS
[7]   Y← 1 2 5 10 20 50 100
[8]   'TEMP. RANGE 1-100 MILL. ∘ K OK? (Y=YES,OR ENTER 7 CO
      MPONENT VECTOR):'
[9]   Y←⍞
[10]  T←(143.8÷(0.25×⍳80))∘.÷Y
[11]  T←*(((T≤170)×T)+(T>170)×170)
[12]  'ENTER AREA OF GM TUBE (SQ. CM.):'
[13]  X←⍞
[14]  R←(X÷○1)*0.5
[15]  'ENTER LENGTH OF GM TUBE (CM.);'
[16]  LO←⍞
[17]  'ENTER RADII OF TWO MASKS (CM.), ASCENDING ORDER:'
[18]  X←⍞
[19]  R1←X[1]
[20]  R2←X[2]
[21]  'ENTER CORRESPONDING HEIGHTS OF MASKS (CM.):'
[22]  X←⍞
[23]  HTS1←X[1]
[24]  HTS2←X[2]
[25]  'ENTER MAX. THETA, GRID (DEGREES,DEGREES):'
[26]  X←(○÷180)×X←⍞
[27]  TX←0,(X[2]×⍳(⌊X[1]÷X[2]))
[28]  AROO← 6 3 ⍴÷8
[29]  AROO[⍳3;3]←⌽AROO[1;]←AROO[6;]←AROO[3+⍳3;3]←
      0.096043 0.046656 0
[30]  AROO←⌊0.5+(1000000)×AROO
[31]  X← 0.48032 1.3577 0 0.5 1.5 2.23265
      0.5 1.5 2.3861
[32]  CGXO← 6 3 ⍴X
[33]  CGXO[3+⍳3;]←⊖⌽⊖CGXO[⍳3;]
[34]  CGYO← 6 3 ⍴(,⌽⊖(CGXO[⍳3;]+3))
[35]  CGYO[⍳3;]←6-⊖⌽⊖CGYO[3+⍳3;]
[36]  CGYO[1;3]←CGYO[6;3]←0
[37]  CGXD←CGXO-(6 3 ⍴(0 1 2))
[38]  CGXD[1;3]←CGXD[6;3]←0
[39]  CGYD← 6 3 ⍴ 0.3861 0.2327 0 0.5
      0.5 0.3577 0.5 0.5 0.48 0.5 0.5
      0.52 0.5 0.5 0.379 0.401 0.249
      0
[40]  RCR←0×TX
[41]  'SAVE THIS WORKSPACE, LOAD ''CONE'', COPY ''INPUT INC
      ONE'', AND RUN ''CONE''.'
    ∇
```

```
     ∇ CONE
[1]   TTSTRT←⍳20
[2]   TSTRT←⍳21
[3]   J←0
[4]   ANG←X←Y←RAD←OFST←○0
[5]   AXE←(○R*2)×TRWIN×1-TRGAS
[6]   W←0.25×○80
[7]   D←(2*¯1.5)×R
[8]   A←0.29289×R
[9]   B←0.2283×R
[10]  'GM TUBE OF RADIUS ';R;' CM., LENGTH ';LO;' CM., AND
      AREA ';○R*2;' SQ.CM.'
[11]  'MASKS OF ';R1,R2;' CM. RADII AT HEIGHTS OF ';HTS1,
      HTS2;' CM.'
[12]  MAX←((¯3○((R+R2)÷HTS2))⌊(¯3○((R+R1)÷HTS1)))×
      180÷○1
[13]  'MAX. THETA = ';TX[⍴TX]×180÷○1;' DEG.;MAX VISIBLE = '
      ;MAX;' DEG.'
[14]  'OPTIONS WANTED?(1=YES):'
[15]  OPTN←⍞
[16]  →(J=0)/CRCALC
[17] CYCLEANG:→(OPTN≠1)/CYCLEANG+3
[18]  ' J=';J;',OPTION NOW:'
[19]  X←⍞
[20]  OF1←HTS1×3○TX[J]
[21]  OF2←HTS2×3○TX[J]
[22]  OF3←OF2-OF1
[23]  AR← 6 3 ⍴0
[24]  →(((OF1-R1)≥R)∨((OF2-R2)≥R)∨((R1+R2)≤OF3))/LENGTH
[25]  AR←ARO←(○R*2)×AROO÷(+/+/AROO)
[26]  →(((OF2+R)≤R2)×((OF1+R)≤R1))/LENGTH
[27]  A2←ANG←RAD←OFST←○0
[28]  X2←○0.5
[29]  X1←0
[30]  R1ONLY←((OF3+R1)≤R2)
[31]  →R1ONLY/SKIP1
[32]  →((OF1+R)≤R1)/SKIP2
[33]  X2←¯2○(((R2*2)+(OF3*2)-R1*2)÷2×R2×OF3)
[34]  X1←¯2○(((R2*2)+(OF2*2)-R*2)÷2×R2×OF2)
[35]  R2ONLY←X1≤X2
[36]  →R2ONLY/SKIP2
[37]  X1←¯2○((R2×2○X2)-OF3)÷R1)
[38] SKIP1:MAX←¯2○(((R1*2)+(OF1*2)-R*2)÷2×R1×OF1)
[39]  →(MAX≤X1)/SKIP2
[40]  ANG←(¯1○((0,D,2×D)÷R1)),¯2○(((D×⍳⌊(R1-(D|OF1))÷D)+(D|
      OF1))÷R1)
[41]  ANG←ANG[⍋ANG]
[42]  ANG←((ANG<MAX)/ANG),MAX
[43]  ANG←X1,((ANG>X1)/ANG)
```

```
[44]  RAD←(ρANG)ρR1
[45]  OFST←(ρANG)ρOF1
[46]  →R1ONLY/SEGMNT
[47] SKIP2:→((OF2+R)≤R2)/SEGMNT
[48]  A2←(¯1○((0,D,2×D)÷R2)),¯2○(((D×⍳⌊(R2-(D|OF2))÷D)+(D|
      OF2))÷R2)
[49]  A2←A2[⍋A2]
[50]  MAX←¯2○(((R2*2)+(OF2*2)-R*2)÷2×R2×OF2)
[51]  A2←((A2<MAX)/A2),MAX
[52]  →R2ONLY/SKIP3
[53]  →(X2≥MAX)/SKIP3
[54]  A2←((A2<X2)/A2),X2
[55] SKIP3:RAD←((ρA2)ρR2),RAD
[56]  OFST←((ρA2)ρOF2),OFST
[57]  ANG←A2,ANG
[58] SEGMNT:SEG←0.5×(RAD*2)×((2×ANG)-1○(2×ANG))
[59]  Y←(3×D)+OFST-RAD×2○ANG
[60]  X←RAD×1○ANG
[61]  →(Y[ρANG]>(D-B))/KSET
[62]  AR[1;1]←ARO[1;1]-SEG[2]
[63]  →LENGTH
[64] KSET:K←1
[65]  IY←⌈Y[1]÷D
[66]  →(1=IY)/CALC
[67]  AR[⍳(IY-1);]←((IY-1),3)ρ0
[68] CALC:→(ANG[K]=X2)/TCT
[69]  IY←⌈Y[K+1]÷D
[70]  IX←⌈X[K+1]÷D
[71]  →(IX=3)/STCT
[72]  AR[IY;IX+⍳(3-IX)]←(3-IX)ρ0
[73] STCT:MAX←2×D×((((IY×D)⌊(R+3×D))-Y[K+1])×(ANG[K+1]≠X1))
      ×((K+1)≠(ρANG))
[74]  AR[IY;IX]←(SEG[K+1]-SEG[K])+(AR[IY;IX]×(ANG[K]=X2))+
      MAX-(Y[K+1]-Y[K])×2×D×(IX-1)
[75] TCT:→((ρANG)>K←K+1)/CALC
[76]  MAX←¯2○(((|(Y[K]-3×D)),(|(((IY-3)×D)⌊R)))÷R)
[77]  MAX←0.5×(R*2)×((2×MAX)-1○(2×MAX))
[78]  AR[IY;IX]←AR[IY;IX]+(|(MAX[1]-MAX[2]))-2×D×(((IY×D)⌊(
      R+3×D))-Y[K])×(IX-1)
[79] LENGTH:L← 6 3 ρ1
[80]  →(J=0)/NEWEFF
[81]  →(TX[J]=0)/NEWEFF
[82]  CGY←D×CGY←CGYO+CGYD×(1-(AR÷ARO))
[83]  CGX←D×CGX←CGXO-CGXD×(1-AR÷ARO)
[84]  L←((R×2○(¯1○(CGX÷R)))-(CGY-3×D))÷(LO×1○TX[J])
[85]  L←L⌊(÷2○TX[J])
[86]  →(OPTN≠1)/NEWEFF-1
[87]  'PRINT (AR÷ARO;L)?(1=YES),STOP(→=YES), GO ON (9=YES):'
```

```
[88]  K←⍬
[89]  →(K≠1)/NEWEFF-1
[90]  'AR÷ARO';AR÷ARO
[91]  'L';L
[92]  X←Y←ANG←SEG←RAD←CGX←CGY←⍳0
[93] NEWEFF:X←TRWIN*(÷2○TX[J])
[94]  AXE←+/(((X∘.×AR[;1])×(1-(TRGAS∘.*L[;1]))))
[95]  AXE←AXE+(+/((X∘.×AR[;2])×(1-(TRGAS∘.*L[;2]))))
[96]  AXE←AXE+(+/((X∘.×AR[;3])×(1-(TRGAS∘.*L[;3]))))
[97]  AXE←AXE×2○TX[J]
[98] CRCALC:CR←(AXE÷W)+.×(÷T)
[99]  CR←CR,((AXE×W*¯4)+.×(÷T-1))
[100] CR←CR,((AXE÷W)+.×(⍉ 2 80 ⍴(80⍴1),W*-2))
[101] →(J≠0)/SKIPSET
[102] CRO←CR
[103] →RUNON
[104]SKIPSET:RC←CR÷CRO
[105] RCR[J]←+/RC÷16
[106] →(OPTN≠1)/RUNON
[107] 'DETAILED OUTPUT WANTED?(1=YES):'
[108] Y←⍬
[109] →(Y≠1)/RUNON
[110] 'AT THETA = ';TX[J]×180÷○1;' DEGREES.'
[111] '   REL.AREA: ';X←((+/+/AR)÷(○R*2))×2○TX[J]
[112] →(X=0)/RUNON
[113] '   REL. COUNTING RATES:'; 2 7 ⍴RC; 1 2 ⍴(RC[
      15],RC[16])
[114] X←+/RC÷16
[115] '   LIN-AVE: ';X;'  +,- ';(+/(((RC-X)*2)÷15))*
      0.5
[116] X←*((+/⍟RC)÷16)
[117] '   LOG-AVE: ';X;' TO A FACTOR OF ';*((+/(((⍟RC÷X)*
      2)÷15))*0.5)
[118] ''
[119]RUNON:→((⍴TX)≥J←J+1)/CYCLEANG
[120] 'FINISHED'
[121] X←⌶21
[122] 'CPU TIME :';(X-TSTRT)÷60;' SEC.'
[123] X←⌶20
[124] 'REAL TIME:';(X-TTSTRT)÷60;' SEC.'
[125] X←⌶23;' USERS.'
```

APPENDIX F. DETERMINING TEMPERATURES FROM COUNTING RATE RATIOS

The program called RATIOS can be used to determine the free-free and blackbody temperatures corresponding to either count rate ratios or energy flux ratios providing that the energy flux ratios are derived by multiplying count rates by constant conversion factors. The program is flowcharted in Figure F-1 and listed at the end of this appendix.

The operation of the program is as follows. The product of the efficiency and the area of each GM tube is entered, multiplied automatically by λ/hc, and converted to an integer format accurate to one half part in 10^8. If energy flux ratios are to be used, the product of the efficiency (an 80-component vector), the area, and the conversion factor used is entered. The program calculates the ratios obtained for 18 temperatures ranging from 0.4 to 100 million °K. Due to core limitations, these calculations are run in blocks of six temperatures. The spectra for six temperatures are calculated, and then the ratios are calculated. The next block of temperatures is handled in the same way until all three blocks of temperatures have been processed. Then the answer is listed.

In detail, the program operates as follows.

Steps 1 & 2 define a vector W of the wavelengths and set up a matrix of temperatures.

Steps 3 - 7 ask for the input data, multiply it by λ/hc, convert it to integer format, and calculate the overall normalization factor, GX. The two GM tube efficiencies are called GM1 and GM2, although they have been modified.

Steps 8 & 9 initialize the index I and the vectors for temperature (TANS), blackbody ratios (BANS), and free-free ratios (FANS).

Steps 10 - 12 define a matrix $X = \exp hc/\lambda kT$ and the free-free spectra matrix $Y = (W^{-2}) \cdot X$.

Step 13 calculates the free-free ratios corresponding to the temperatures in T. Step 13 is equivalent to

$$FF = \frac{GX\left(\sum_I GM1_I \cdot Y_{IJ}\right)}{\sum_I \left(GM2_I \cdot Y_{IJ}\right)}$$

Step 14 calculates the blackbody spectra matrix Y, where

$$Y = \frac{W^{-5}}{X-1}$$

Step 15 calculates the blackbody ratios for the temperatures in T.

Step 16 increments I, tests it, and cycles the program back to step 9, if appropriate.

Step 17 reshapes the answers for TANS, BANS, and FANS as a matrix with the temperature, corresponding blackbody ratio, and corresponding free-free ratio appearing in a row; it then prints them out.

Step 18 prints a header for the matrix.

Step 19 clears variables.

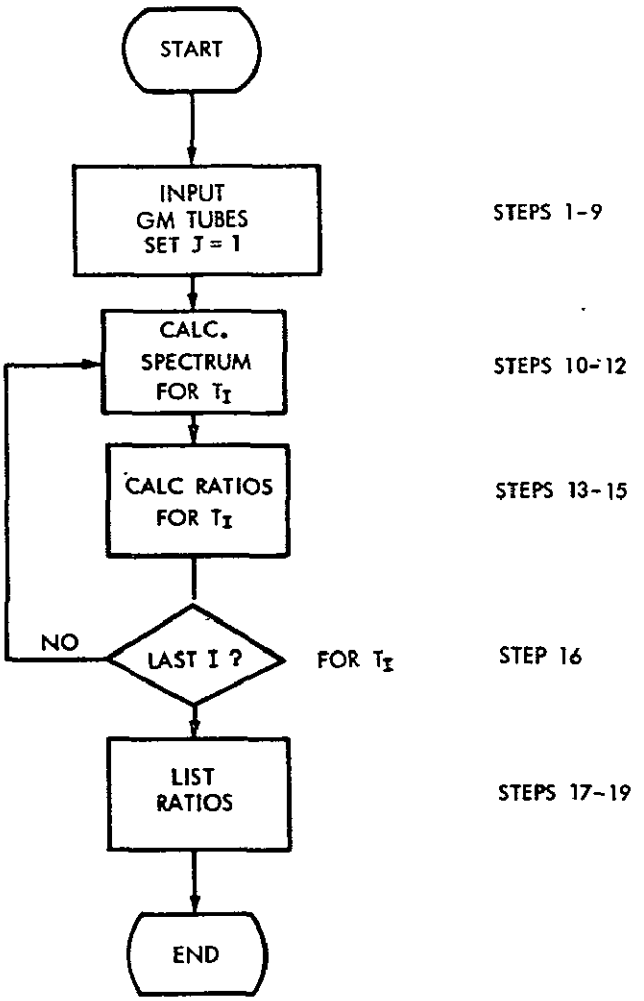

Figure F-1. Flowchart for RATIOS Program

```
    ∇ RATIOS
[1]   W←0.25×⍳80
[2]   TEMP← 3 6 ⍴(0.4 0.6 0.8 1 1.5
      2 3 4 6 8 10 15 20 30 40 60 80
      100)
[3]   'ENTER GM1×AREA×(CONVERSION FACTOR):'
[4]   GM1←⌊0.5+(100000000)×GM1÷GX←⌈/GM1←W×(1÷
      1.9862E¯8)×GM1←⎕
[5]   'ENTER GM2×AREA×(CONVERSION FACTOR):'
[6]   GM2←⌊0.5+(100000000)×GM2÷X←⌈/GM2←W×(1÷
      1.9862E¯8)×GM2←⎕
[7]   GX←GX÷X
[8]   FANS←BANS←⍳0
[9]   I←1
[10] STARTR1:Y←(143.8÷W)∘.÷TEMP[I;]
[11]  X←*(((Y<170)×Y)+(Y≥170)×170)
[12]  Y←((W*¯2)∘.×6⍴1)÷X
[13]  FANS←FANS,Y←GX×(GM1+.×Y)÷(GM2+.×Y)
[14]  Y←((W*¯5)∘.×6⍴1)÷(X-1)
[15]  BANS←BANS,Y←GX×(GM1+.×Y)÷(GM2+.×Y)
[16]  →(4≠I←I+1)/STARTR1
[17]  ⎕←RESULT←⍉(3 18 ⍴((,TEMP×1000000),BANS,FANS))
[18]  'TEMP.,BLACKBODY, AND FREE-FREE RATIOS,GM1/GM2.
[19]  TEMP←BANS←FANS←⍳0
    ∇
```