Optimization of Radiometric Channel Solar Calibration for the Clouds and the Earth's Radiant Energy System (CERES) Using the Monte-Carlo Method

by

Tai K. Nguyen

Thesis submitted to the faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

APPROVED:

Dr. J. R. Mahan, Chairman

Dr. E. F. Brown

Dr. E. P. Scott

June, 1994

Blacksburg, VA

C.2

LD
5655
V855
1994
N489
c. 2

Optimization of Radiometric Channel Solar Calibration for the Clouds and the Earth's Radiant Energy System (CERES) Using the Monte-Carlo Method

by

Tai K. Nguyen

J. Robert Mahan, Chairman

Mechanical Engineering

(ABSTRACT)

Recent satellite measurements have found the range of solar radiation flux at the mean Earth-Sun distance to be from 1365 to 1372 W/m^2, or 1368 W/m^2 to within ±0.5 percent. This regularity is considered sufficient to permit the use of solar radiation as a source of energy for inflight calibration of radiometers designed to detect electromagnetic radiation in the solar spectrum. But direct viewing of the Sun would provide a flux considerably greater than the operating range of radiometers designed to observe typical Earth scenes. Therefore an attenuator is required. The Earth Radiation Budget Experiment (ERBE) radiometers, operational since 1984, relied on a mirror attenuator mosaic (MAM) to attenuate the solar energy. The ERBE MAM is an array of 105 tightly packed concave spherical mirrors with a black

mask covering the surface between the hemispherical cavities and partially covering the cavities themselves. In principle, the reflection of solar energy by the MAM was anticipated to be independent of the solar incidence direction. Unfortunately, flight data revealed a variation with solar incidence angle of as much as 20 percent for flux arriving at the detector during solar calibration. An improvement of the ERBE MAM design, suppression of the black mask and reduction of the surface area of the spherical mirrors, has been achieved for the Clouds and the Earth's Radiant Energy System (CERES) radiometers, which will be operational sometime in the late-1990's.

The topic of this thesis is the creation of a thermal-radiative model, based on the Monte-Carlo ray-trace method, to characterize the performance of the CERES MAM. The radiative analysis suggests that the current CERES MAM design is still somewhat less than optimal. The desired specifications are that the fraction of solar energy reflected by the MAM to the radiometer aperture be independent of the solar vector, and that the distribution of this reflected energy be uniform across the entire surface area of the radiometer aperture. The work reported here establishes that these specifications can be met by a simple reorientation of the MAM diffuser plate.

Acknowledgements

I would like to thank my professor, Dr. J. R. Mahan, for giving me the opportunity to work on this challenging research. I would like to thank him for being my professor, my advisor, and my friend.

I would like to acknowledge Dr. E. F. Brown and Dr. E. P. Scott for their willingness to serve on my advisory committee.

I am thankful to all graduate students in the Thermal Radiation Group, especially Pierre Villeneuve and Kory Priestley, for their help and kindness.

I am grateful to my Mother and my Step Father, Khang Dao and Charles Allen, for showing me the value of education.

I am thankful to my brothers, Steve Nguyen and Truc Nguyen, for supporting me financially and sharing my life through all these years.

Table of Contents

1.0 Introduction 1

1.1 The Earth Radiation Budget 1

1.2 A Brief History of Earth Radiation Budget Measurements 3

1.3 The Earth Radiation Budget Experiment (ERBE) 5

1.4 The Clouds and the Earth's Radiant Energy System (CERES) . . 7

1.5 CERES Radiometric Instruments 9

1.6 Inflight Calibration Facilities 10

1.7 Motivation and Goals of this Work 13

2.0 Monte-Carlo Formulation 17

2.1 The Distribution Factor 18

2.2 The Monte-Carlo Ray-Trace Method 21

3.0 The CERES MAM Model 32

3.1 Description of the CERES MAM and Its Operation 32

3.2 Application of the Monte-Carlo Method 35

4.0 Results and Discussion 39

4.1 Validation of the CERES MAM Model 39

4.2 Radiative Analysis 41

4.3 Diffuser Plate Reflection Diagnosis 43

4.4 Diagnosis of the Diffuser Plate Reflection for different Diffuser Plate Orientations 47

4.5 Radiative Analysis Using the Diffuser Plate Orientation as a Parameter 49

4.5.1 Comparison of CERES MAM Model Results with Measured Data 49

4.5.2 Parametric Radiative Analysis 51

4.5.3 Study of the CERES MAM Performance Between the 0 and -4 deg Diffuser Plate Orientations 54

4.6 Option to Improve the CERES MAM Performance 56

5.0 Conclusions and Recommendations 59

5.1 Conclusions 59

5.2 Recommendations 60

References 108

Appendix A. Computer Program CERESMAM 112

Vita 253

List of Tables

Table 1. CERES Program Chronology 62

Table 2. Convergence of Radiometer Aperture Distribution Factor Due to Collimated Radiation Entering Parallel to the Baffle Axis 63

List of Illustrations

Fig. 1. Radiative Fluxes Contributing to the Earth Radiation Budget 64

Fig. 2. The CERES Scanning Module 65

Fig. 3. Exploded View of CERES Radiometer Instrument 66

Fig. 4. The MAM Design: (a) MAM, (b) MAM Detail, and (c) MAM Support 67

Fig. 5. The CERES Inflight Calibration Systems 68

Fig. 6. The Scan Positions 69

Fig. 7. Surface Subdivision of CERES MAM Model 70

Fig. 8. Triangular Surface Subdivision 71

Fig. 9. Direction of Specular Reflection 72

Fig. 10. Area Element Subdivision 73

Fig. 11. Convergence of Radiometer Aperture Distribution Factors Due to Collimated Radiation Entering the Baffle Opening Along the Baffle Axis 74

Fig. 12. ERBE MAM (a) Ground and (b) Flight Data 75

Fig. 13. Definitions of Solar Angles 76

Fig. 14. Comparison of ERBE MAM Flight Data with Numerical Results from CERES MAM Model 77

Fig. 15. Distribution of Solar Radiation on the MAM Enclosure for Collimated Radiation Incident Along the Baffle Axis and with the Diffuser Plate Oriented at 15 deg Below the Baffle Axis, Viewed from the Front of the Enclosure 78

Fig. 16. Distribution of Solar Radiation on the MAM Enclosure for Collimated Radiation Incident Along the Baffle Axis and with the Diffuser Plate Oriented at 15 deg below the Baffle Axis, Viewed from the Back of the Enclosure 79

Fig. 17. Imaginary Hemispherical Surface Added to the CERES MAM Model Geometry for Diffuser Plate Reflection Diagnoses 80

Fig. 18. Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented at 15 deg Below the Baffle Axis, Observed from the Front of the Diffuser Plate 81

Fig. 19. Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented at 15 deg Below the Baffle Axis, Observed Directly from the side of the Diffuser Plate 82
Fig. 20. Multiple Reflection Effect 83
Fig. 21. The Multiple Mirror Effect 84
Fig. 22. Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented 12 deg Below the Baffle Axis 85
Fig. 23. Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented 8 deg Below the Baffle Axis 86
Fig. 24. Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented 4 deg Below the Baffle Axis 87
Fig. 25. Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented Along the Baffle Axis 88
Fig. 26. Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented 4 deg Above the Baffle Axis 89
Fig. 27. Illustration of the Limit of Incidence Direction to Avoid Multiple Reflection 90
Fig. 28. Fraction of Radiation Reflected to the Radiometer Aperture as a function of the Solar Angle, Defined by the CERES MAM Model and Experimental Data, for Diffuser Plate Oriented Along the Baffle Axis 91

Fif. 29. Fraction of Radiation Reflected to the Radiometer Aperture as a Function of Solar Angle for the Diffuser Plate Orientation Varying from 15 deg to 8 deg Below the Baffle Axis 92

Fig. 30. Fraction of Radiation Reflected to the Radiometer Aperture as a Function of Solar Angle for the Diffuser Plate Orientation Varying from 6 deg Below to 8 deg Above the Baffle Axis 93

Fig. 31. Contour Plot of the Radiometer Aperture Distribution Factor as a Function of the Solar Angle and the Diffuser Plate Orientation 94

Fig. 32. Contour Plot of the Radiometer Aperture Distribution Factor as a Function of the Solar Angle and the Diffuser Plate Orientation, Observed Along the Solar-Angle-Axis 95

Fig. 33. Worst Case of Solar Angle 96

Fig. 34. Mean Value and Standard Deviation of the Distribution Factor of the Radiometer as a Function of the Solar Angle for the Diffuser Plate Oriented at 15 deg Below the Baffle Axis 97

Fig. 35. Mean Value and Standard Deviation of the Distribution Factor of the Radiometer Aperture as a Function of the Solar Angle for the Diffuser Plate Oriented at 4 deg Above the Baffle Axis 98

Fig. 36. Distribution of Solar Radiation on the MAM Enclosure for the Diffuser Plate Normal to the Baffle Axis and with the Solar Vector Along the Baffle Axis 99

Fig. 37. Distribution of Solar Radiation on the MAM Enclosure for the Diffuser Plate Oriented 4 deg Above the Baffle Axis and with the Solar Vector Along the Baffle Axis 100

Fig. 38. Fraction of Solar Radiation Reflected Back Out of the Baffle Opening as a Function of the Solar Angle for the Diffuser Plate Oriented at 0 and 4 deg Above the Baffle Axis 101

Fig. 39. Normalized Distribution Factor of the Radiometer Aperture as a Function of Solar Angles in the Both Vertical and Horizontal planes for the Diffuser Plate Oriented at 4 deg Above the Baffle Axis and Translated 1 cm Forward 102

Fig. 40. Distribution of Solar Radiation Reflected on the MAM Enclosure for the Diffuser Plate Oriented 4 deg Above the Baffle Axis and Translated 1 cm Forward and for the Incident Solar Angle Along the Baffle Axis 103

Fig. 41. Distribution of Solar Radiation Reflected onto the Radiometer Walls for Diffuser Plate Oriented 4 deg Above the Baffle Axis and Translated 1 cm Forward and for Various Solar Angles in the Vertical Plane 104

Fig. 42. Distribution of Solar Radiation Reflected on the MAM Enclosure for the Diffuser Plate Oriented 4 deg Above the Baffle Axis and Translated 1 cm Forward and for a Solar Angle of -2 deg in the Horizontal PLane 105

Fig. 43. Distribution of Solar Radiation Reflected onto the Radiometer Walls for the Diffuser Plate Oriented 15 deg Below the Baffle Axis and for a Solar Angle of -2 deg in the Horizontal Plane 106

Fig. 44. Distribution of Solar Radiation Reflected on the MAM Enclosure for the Diffuser Plate Oriented 4 deg Above the Baffle Axis and Translated 1 cm Forward, Determined by a Monte-Carlo Ray-Trace Using 10^9 Energy Bundles. Incident Radiation is Along the Baffle Axis 107

Nomenclature

A_i	Area of surface i (m^2)
D_{ij}	Radiation Distribution Factor (-)
n	Number of surfaces in the enclosure (-)
N_i	Number of energy bundles emitted from surface i (-)
N_{ij}	Number of energy bundles emitted by surface i that are absorbed by surface j (-)
q	Radiation flux ($W \cdot m^{-2}$)
Q_{ij}	Power emitted from surface i and absorbed by surface j (W)
Q_i	Net power leaving surface i (W)
R	Reflectivity ratio, $\rho^s/(\rho^d+\rho^s)$ (-)
T	Temperature (K)

Greek

ϵ	Emissivity (-)
ρ	Reflectivity (-)
ρ^s	Specular component of reflectivity (-)
ρ^d	Diffuse component of reflectivity (-)
σ	Stefan-Boltzmann constant (5.6696×10^{-8} W· m^{-2}· K^{-4})
θ	Cone angle of emission (deg)
ϕ	Circumferential angle of emission (deg)

1.0 Introduction

1.1 The Earth Radiation Budget

Since the beginning of this century, progress in our society of consumption can be in part associated with chemical processes. From fossil fuel burning for the production of power so crucial to our existence, to the use of aerosols in our everyday life, these processes release great quantities, and at increasing rates, of unnatural substances into the fragile atmosphere surrounding our planet. The question has been raised about the effects of this rapid change in chemical composition of the atmosphere on our climate, and many studies attempting to answer this question have been carried out by scientists from around the world.

Although these studies involve different methods and differ in their origins and approaches, all the resulting predictions are based on one common phenomenon: the greenhouse effect, or more specifically, the systematic alteration of the Earth radiation budget.

A simple model for the Earth radiation budget is illustrated in Figure 1. In this figure, q_S and q_{ER} are, respectively, the shortwave solar flux and the Earth-reflected solar flux at the top of the atmosphere, and q_{EM} is the outward Earth-emitted radiation in the longwave band. The global net radiation flux at the top of the atmosphere, q_G, can be expressed as

$$q_G = q_S - (q_{ER} + q_{EM}). \tag{1.1}$$

For equilibrium over a long period of time, the global net radiation flux must be equal to zero. But a change in the chemical composition of the atmosphere could alter the shortwave Earth-reflected and the longwave Earth-emitted radiative fluxes such that the three terms on the right-hand side of Equation 1 no longer add up to zero.

This simple model clearly shows the preponderant role of the radiative fluxes in the long-term stability of the Earth's climate. The Earth Radiation Budget Experiment (ERBE) and the Clouds and the Earth's Radiant Energy System (CERES) radiometric instruments are used to, among other things, provide better knowledge about the shortwave Earth-reflected solar radiation and the longwave Earth-emitted radiation energies. In order to ensure high accuracy of the

measurements, the responsivities of the radiometric instruments are regularly evaluated by inflight calibration facilities such as the Mirror Attenuator Mosaic (MAM) whose performance is evaluated in this thesis.

1.2 A Brief History of Earth Radiation Budget Measurements

The brief history of Earth radiation budget measurements before the development of space observing systems given in this paragraph is based on a paper by Hunt *et al.* [1]. According to Hunt *et al.*, Pouillet obtained an estimate of the solar constant based on observations nearly 150 years ago. Early in this century, Abbot and Fowle made the first attempt to determine the components of the Earth radiation budget. In the 1920's, Dines carried out an investigation on the longwave radiation from the atmosphere, and a few years later, Mügge and Möller developed a graphical method for calculating the radiation flux in the atmosphere. From the 1940's to the 1960's, more complete consideration was given to the role of atmospheric constituents, such as water vapor and carbon dioxide.

The arrival of spacecraft marked a new era for Earth radiation budget measurements. The first observation of the Earth from space was obtained by

cameras launched by sub-orbital rockets in the late 1940's. On Feb. 17, 1959, the first satellite picture of the Earth's cloud cover was taken by Explorer 6. Later, data from Explorer 7, the Television Infrared Observational Satellite (TIROS), and the National Oceanic and Atmospheric Administration (NOAA) satellites provided further improved measurements [2].

In modern Earth observing systems incident radiative fluxes are converted into electronic signals. These signals are then transmitted to ground stations and eventually are interpreted as top-of-the-atmosphere net fluxes. The final step is to average these signals over time and space to provide a measurement of the Earth radiation budget [3].

The evolution of Earth radiation budget instruments can be characterized by: (1) spacecraft constraints of power and attitude control, (2) viewing angles of radiometric instruments, (3) spectral bands that isolate the Earth radiation budget into shortwave and longwave components, (4) and on-board calibration. These factors allow Earth radiation budget observational systems to be classified into three generations.

The early Earth radiation budget instruments consisted of black and white hemispheres bonded to mirror surfaces. The black hemisphere was sensitive to the total radiation, while the white hemisphere was sensitive only to longwave radiation. Four thermometers positioned around the hemispheres determined their temperatures. In the second generation of Earth radiation budget missions, from the 1960's to the 1970's, scanning and nonscanning radiometers were used on

spacecraft, such as the Nimbus satellite, to measure the shortwave and the longwave regions of the electromagnetic spectrum. The longwave radiometer employed cold space and hot internal blackbody sources for inflight calibrations, while the shortwave channel had no calibration capability. The third generation of observational systems, currently in operation, use a complete instrument package which measures radiation in three wavelength bands: shortwave, longwave, and total. Inflight calibration systems are provided for each channel. This third-generation technology was used during the 1980's by the Earth Radiation Budget Experiment (ERBE) to investigate the influence of the Earth's radiative energy budget [2].

1.3 The Earth Radiation Budget Experiment (ERBE)

The ERBE program was initiated in 1979 [4] in the interest of providing a continuing record of radiation budget changes and to obtain more accurate measurements through improved sampling. The main objectives of ERBE were to: (1) determine for a minimum of one year, the monthly average radiation budget on regional, zonal, and global scales; (2) determine the equator-to-pole energy transport gradient; (3) determine the average diurnal variation of the radiation

budget on a regional and monthly scale; and (4) examine the measured data to ensure the integrity of instrument calibration procedures, sampling efficiency, and conversion algorithms [2,5].

In order to obtain sufficient coverage, a three-satellite system carrying Earth radiation budget instruments was required. The Earth Radiation Budget Satellite (ERBS) was placed in orbit by the space shuttle Challenger on October 5, 1984; the National Oceanic and Atmospheric Administration's satellite NOAA-9 was launched on December 12, 1984; and NOAA-10 was launched on September 17, 1986 [6]. After only a few months of operation, data from ERBE had been used to provide important results. First, they confirmed the positive feedback of water vapor and revealed that water vapor is one of the major sources of positive feedback in the climate system. Then, they validated the essential physics of the greenhouse effect as the absorption of infrared radiation by certain gases, chiefly H_2O and CO_2. Most significantly, however, they showed that low-level clouds cool the current climate through reflection of the shortwave solar radiation, whereas high-level clouds cause heating through absorption and reflection of the longwave radiation emitted by the Earth's surface (the so-called "cloud forcing" phenomenon) [7].

Although the importance of cloud forcing on the Earth's climate is known, the duration of the ERBE measurements was too short to completely quantify its effect. Clouds remain the least understood phenomenon in the climate system [8]. To further investigate the effect of clouds and to provide Earth radiation budget

measurements at a higher accuracy, a new measurement program has been proposed: The Clouds and the Earth's Radiant Energy System (CERES).

1.4 The Clouds and the Earth's Radiant Energy System (CERES)

The Clouds and the Earth's Radiant Energy System (CERES) mission is a central element of the Earth Observing System (EOS), a United States contribution to the international scientific research effort to understand how the Earth works as a system [9]. EOS will be the first program to provide coordinated measurements of the atmosphere, oceans, solid earth, and biogeochemical cycles to the international scientific community. In order to meet observational needs of the Earth study, EOS is designed to operate for 15 years, a sufficiently long of time to include major environment changes. This period ranges over four to five *el niño* events to sample the pseudo-periodic character of this phenomenon, and over an entire sunspot cycle. The distribution of EOS information is supported by the EOS Data and Information System (EOSDIS), a program whose policy specifies that all data be available to all users.

CERES is among the first missions of the EOS program. As early as 1997, the Tropical Rainfall Measurement Mission (TRMM) spacecraft will be placed into

orbit with one module of CERES radiometers on board. Two other CERES radiometer modules having different scanning patterns will be aboard the EOS-AM satellite, which is scheduled to be launched in 1998. The chronology of other EOS satellites carrying CERES radiometers is shown in Table 1 [10].

The CERES mission is a continuation to the Earth Radiation Budget Experiment (ERBE). The objectives of this program are to advance our knowledge about the effect of clouds on the radiation budget by providing instantaneous radiation fluxes and cloud identification at the top of the atmosphere, and to recover the radiation field from the top of the atmosphere to the Earth's surface. This will enhance the inputs into atmospheric and oceanic models, which in turn will increase the range of weather forecasting in the short term, and should greatly improve our ability to understand the role of clouds in the global change of our climate in the long term. These data have been requested by the World Climate Research Program (WCRP), the Tropical Ocean Global Atmosphere (TOGA) campaign, the World Ocean Circulation Experiment (WOCE), and the Global Energy and Water Cycle Experiment (GEWEX) [11].

The instruments used by CERES are scanning thermistor bolometer radiometers with inflight calibration systems. Each module consists of three radiometric channels, as shown in Figure 2. The shortwave channel will be sensitive to Earth-reflected solar radiation from 0.3 μm to 5.0 μm, the total channel will be sensitive to both the Earth-emitted longwave radiation (from approximately 5.0 μm to greater than 200 μm) and the shortwave Earth-reflected solar radiation,

and the third radiometric channel will be sensitive to the water vapor "window" band between 8 μm and 12 μm. The atmosphere is essentially transparent in this wavelength interval.

In the case where two CERES instrument modules are used in a single platform, one instrument module will scan in the cross-track direction to detect radiation fluxes in a plane perpendicular to the orbital plane, and the second instrument module will operate in a biaxial rotating plane, in which the scanning plane will be rotated from the reference cross-track plane to the along-track plane. The measurements from the biaxially scanning instrument will be used to provide angular flux data that will be used to refine the Bidirectional Reflectivity Distribution Function (BRDF) models. Radiation fluxes measured at the satellite altitude are then transformed to the top of the atmosphere using BRDF models [12].

1.5 CERES Radiometric Instruments

An exploded view of a CERES radiometer appears in Figure 3. This instrument is based on the Cassegrain telescope concept to focus incoming radiation on a thermistor bolometer, called the active flake, which makes up one arm of a deflection bridge network. A second thermistor bolometer, called the

compensating flake, is identical to the active flake but is mounted on the opposite side of the heat sink. This detector makes up an adjacent arm of the bridge circuit. The radiation absorbed by the active flake is converted into heat and causes the temperature of this element to increase. This temperature rise is sensed by a thermistor embedded in the flake. The resistance difference between the thermistors in the active and compensating flakes causes the bridge network to deflect. The deflection voltage is then used to deduce the incident radiation.

The three channels of the radiometer are identical to each other in conception, except that the shortwave channel has a Dynasil™ quartz filter, while the longwave channel has two filters made of 1 mm thick zinc sulfide (ZnS) and 0.5 mm thick cadmium telluride (CdTe). The total channel has no filter [13].

1.6 Inflight Calibration Facilities

In order to allow CERES radiometers to sense only the shortwave Earth-reflected solar radiation or only the water vapor window channel, filters are added to the radiometer design, as described above. But experience has shown that the transmissivity of these filters may decrease when exposed to solar ultraviolet radiation and to a hard vacuum environment. The changes of the instrument

responsivities, due to the degradation of the filters or changes in electronic components, are regularly evaluated by inflight calibrations in which the radiometers observe one of three internal calibration sources:

- the Mirror Attenuator Mosaic (MAM)
- the Internal Calibration Module (ICM)
- the Shortwave Internal Calibration Source (SWICS).

The MAM is an array of tightly packed concave spherical mirrors etched into an aluminum plate, also called the diffuser plate, within a baffle which limits the field of view of the diffuser plate, as shown in Figure 4 [14]. The diffuser plate is expected to reflect solar radiation uniformly regardless of the solar incidence direction. The ICM consists of blackbody sources for the total and atmospheric window radiometers, and a tungsten lamp which is used for the shortwave calibration system. The tungsten lamp is called the Shortwave Internal Calibration Source (SWICS). The SWICS also has a photodiode to monitor the long-term stability of the tungsten lamp. Illustrations of the calibration systems appear in Figures 5 and 6 [14].

During a normal Earth scan of four seconds, the radiometers initially detect the near-zero radiation from dark space at the 18-deg space-look position, then measure radiation fluxes from the Earth between 26 and 154 deg. At the end of the scan, the radiometers rotate to the 194-deg position to look at the deactivated ICM sources, then scan down to face dark space at 162 deg, and finally scan back to measure radiation from the Earth between 154 and 26 deg. During the longwave

internal calibration mode, once every two weeks, the radiometers follow the same scanning profile except that the ICM sources are activated.

During the shortwave and total solar calibration mode, also once every two weeks, the radiometers scan dark space initially at 162 deg, then scan past the ICM to stare at the MAM with no Sun in the MAM field of view. This represents the near-zero power input level. Then, the Sun crosses through the field of view of the MAM baffle. The fraction of solar radiation reflected by the MAM and arriving at the radiometer aperture is measured. Next, the Sun passes out of the field of view of the MAM baffle, the radiometers continue to look at the MAM with only dark space in its field of view. The difference between solar and cold space inputs through the MAM while the radiometers are fixed at the 236-deg solar view position represents the radiation noise from the MAM and its enclosure. The difference between solar reflection to the detector by the MAM (radiometers at the 236-deg position) and direct dark space observation (radiometers at the 162-deg space-look position) defines the solar radiation input to the radiometers. The above differences and the solar radiation flux at the mean Sun-Earth distance are then used to define the changes in the responsivity of the radiometers.

1.7 Motivation and Goals of this work

The Thermal Radiation Group (TRG) of Virginia Polytechnic Institute and State University (VPI&SU), under the direction of Dr. J. Robert Mahan, has been building analytical and numerical models for extremely complex thermal problems for 20 years [15,16,17]. Eskin [18] was the first in the TRG to apply the Monte-Carlo ray-trace technique to create a radiative model for the ERBE nonscanning radiometers. Gardiner [19] and Tira [20] also studied the ERBE nonscanning radiometers. In 1993, Priestley [21] concluded an end-to-end model for the ERBE nonscanning radiometers, Haeffelin [22] completed a study of equivalence in the ERBE/CERES scanning radiometers, and Bonjiovi [23] presented a parametric study of the ERBE/CERES scanning radiometers including the work of Haeffelin [22] and Meekins [24]. The work of the TRG provided important knowledge about the characteristics of ERBE/CERES radiometers. The body of this work will be a valuable tool for radiometer design improvement in the future.

Regardless of the performance of a radiometer, its responsivity is expected to change with time in operation. This is especially true for the shortwave channel, whose filter's transmissivity may decrease not only when exposed to solar ultraviolet radiation and hard vacuum, but may vary with altitude due to the variation of charge particle concentration in the vicinity of the Earth. To overcome this problem, inflight

calibration systems were designed to ensure the repeatability of radiometers in service since 1984. Based on ERBE data, it is believed that solar calibrations provide the best measure of these effects.

An examination of ERBE data showing the responses to solar calibration using the MAM revealed that the radiometer output signals were strongly sensitive to the incident solar direction during calibration [25]. In fact, the fraction of solar radiation arriving at the radiometer had to be averaged to define the power input reference. This averaged power was then used to correct the responsivity of the radiometer in question.

An improved MAM design based on the ERBE MAM has been achieved for the CERES MAM. The performance of the new design is greatly superior to that of ERBE MAM [26]. This improvement was achieved based on experimental results and by taking measurements only in the interval ±2.25 deg instead of the entire ±4 deg field of view (FOV) of the MAM baffle. What are the complete reflecting characteristics of the CERES MAM?An answer to this question was not obvious because no valid analytical or numerical models for the MAM were available.

The effort described in this thesis is the development of a thermal radiative model for the CERES MAM using the Monte-Carlo ray-trace technique. The model permits characterization of the performance of the CERES MAM without resorting to costly experimental testing. The model is a computer code, and so all the dimensional and physical characteristics of the model can be varied with ease to facilitate parametric studies.

The goal of the work reported in this thesis is to build a thermal model and then to use the model to determine the optimally designed configuration for the CERES MAM. Achievement of this goal involves developing a FORTRAN computer code to perform the required radiative analysis. The output of the radiative analysis is the distribution of solar radiation on the entire MAM enclosure. The central point of interest is the strength and distribution of solar energy reflected to the radiometer aperture. However, a knowledge of the complete distribution of energy on the MAM enclosure helped understand the physics of the problem and permitted errors in the computer code to be detected during its development. The results of the parametric study were validated by comparison with flight data from the ERBE MAM operation and experimental results from the engineering model of the CERES MAM. These results were also used to diagnose the reflection pattern from the diffuser plate and thus to determine the best orientation for this diffuser plate. In this thesis the optimal configuration of the diffuser plate is defined as the configuration that satisfies the following objectives:

(1) The fraction of solar radiation captured by the MAM and guided to the radiometer aperture should be independent of the solar vector incident to the MAM aperture.

(2) The distribution of this energy should be uniform across the radiometer aperture area.

(3) The fraction of incident solar radiation reflected back out to the baffle aperture should be maximized.

In the future, the current work could be coupled with other current research projects of the TRG, such as the ongoing Earth Bidirectional Reflectivity Distribution Function model work of P. Villeneuve, and the ongoing advanced CERES data reduction work of K. Priestley, to build a more complete model including Earth scenes, radiometers, and inflight solar calibration schemes. The result would be a magnificent design tool for the next generation of scanning radiometers.

2.0 Monte-Carlo Formulation

Most solutions of thermal radiation heat transfer problems consist of defining the radiation exchange among surfaces of an enclosure. The standard technique for accomplishing this is to compute the configuration factors, based purely on the geometry of the enclosure and the assumptions that all surfaces behave as diffuse and gray emitters, absorbers and reflectors of radiation. Because most surfaces are neither purely diffuse nor gray in reality, the utility of techniques using configuration factors is limited. In 1961, Eckert and Sparrow [27] introduced the exchange factor to determine the radiation exchange among surfaces having both diffuse and specular components of reflectivity. But calculations of the exchange factor have proven to be tedious, if not impossible, when curved surfaces are involved, and their use requires the inversion of a matrix. Simplifying assumptions must be invoked to make the configuration or exchange factor approaches practical for many problems of engineering interest.

In a discussion of a paper presented by Sparrow *et al.* [28], Seban suggested that reflections from most real surfaces can be approximated as the sum of two

components, one diffuse and one specular. In this case the reflectivity can be written

$$\rho = \rho^d + \rho^s , \qquad (2.1)$$

where ρ^d is the diffuse component of the reflectivity and ρ^s is the specular component of the reflectivity. Based on this idea, Mahan and Eskin [29] introduced the distribution factor, D_{ij}. The distribution factor can be defined for any surface or volume element of an enclosure taking into account the directional emissivity and absorptivity, the bidirectionnal reflectivity, and the wavelength dependence of emissivity, reflectivity, and transmissivity.

In the domain of instrumentation involving complex geometries and where high-accuracy is required, the radiation distribution factor defined by the Monte-Carlo ray-trace method, or at least a variation on this approach, is the only method available for radiative analysis.

2.1 The Distribution Factor

In this work the total distribution factor, D_{ij}, is defined as the fraction of

energy diffusely emitted by surface i that is absorbed by surface j due both to direct radiation and to all possible diffuse and specular reflections. Because the distribution factor is determined using the Monte-Carlo method, it is possible to include the directional–spectral emissivity, the bidirectional–spectral reflectivity, and the directional–spectral absorptivity of surfaces in the determination of the distribution factor. In this case we define the monochromatic distribution factor D_{ijk} as the distribution factor in wavelength interval $\Delta\lambda_k$. In addition to its fully directional and monochromatic characteristics, the distribution factor has three useful properties:

$$\sum_{j=1}^{n} D_{ij} = 1.0, \qquad i = 1, 2, ..., n, \tag{2.2}$$

$$\epsilon_i A_i D_{ij} = \epsilon_j A_j D_{ij}, \qquad i = 1, 2, ..., n, \quad j = 1, 2, ..., n, \tag{2.3}$$

and

$$\sum_{i=1}^{n} \epsilon_i A_i D_{ij} = \epsilon_j A_j, \qquad j = 1, 2, ..., n, \tag{2.4}$$

where *n* is the number of surfaces in the enclosure, ϵ is the total hemispherical emissivity of a given surface, and *A* is its surface area. Equation 2.2 is a statement of conservation of energy and Equation 2.3 is the reciprocity relation. Equation 2.4 is obtained by summing Equation 2.3 over *i* and then substituting Equation 2.2 into the result. These equations can be used to detect and eliminate errors during the

determination of the distribution factors of an enclosure.

Using the total distribution factor, D_{ij}, the amount of radiation energy emitted by surface i and absorbed surface j, denoted Q_{ij} and expressed in watts, is defined as

$$Q_{ij} = \epsilon_i A_i \sigma T_i^4 D_{ij}, \tag{2.5}$$

where ϵ_i is the emissivity of surface *i*, A_i is the surface area of surface *i* in m^2, σ is the Stefan-Boltzmann constant (5.6696×10^{-8} $Wm^{-2}K^{-4}$), T_i is the temperature of the emitting surface in kelvins. It should be noted that if the radiative heat loss Q_i of the emitting surface is known instead of its temperature, then the part of Q_i absorbed by each surface *j* of the enclosure Q_{ij} can be computed directly as

$$Q_{ij} = q_i A_i \cos\theta D_{ij}, \tag{2.6}$$

where q_i is the radiative flux emitted from surface A_i, and θ is the angle between the direction of emission and the normal direction of A_i.

The degree of difficulty in calculating the distribution factors increases tremendously with the enclosure geometric complexity, and with any variation of the radiative properties of the enclosure. Since the enclosure representing the CERES MAM consists of many spherical surface segments intersecting irregular plane surfaces, the Monte-Carlo ray-trace technique is believed to be the only practicable technique for determining the associated distribution factors.

2.2 The Monte-Carlo Ray-Trace Method

According to Reference [29], the Monte-Carlo ray-trace technique is a statistical and numerical method by which the real process is replaced by "a game whose outcome is the same as the real problem being modeled but is easier to play." In the new game, radiation energy is *uniformly* divided into an arbitrary number N_i of discrete energy bundles that are followed from emission from surface element *i* to absorption. The properties of the enclosure and the laws of probability are used to determine the number of energy bundles N_{ij} absorbed by a given surface element *j* in the enclosure. The total distribution factor is then defined as the ratio of N_{ij} to N_i in the limit as N_i tends to infinity.

A preliminary work before applying the Monte-Carlo method, and perhaps the most time-consuming task in the development of the current model, is to define the geometry of the problem. This geometry is defined by Cartesian coordinates (x,y,z) of points making up the enclosure, as illustrated in Figure 7.

The enclosure representing the CERES MAM is then subdivided into surfaces based on shapes such as spheres, disks, rectangles, and triangles. Imaginary black surfaces are also used to define openings such as the baffle and the radiometer apertures.

This enclosure includes 105 spherical sector surface elements representing the mirrors of the diffuser plate and 51 other plane surface elements. Figure 7 shows the subdivision of the enclosure into surfaces. From this figure, it is obvious that there is no symmetry in the elevation plan. Although the geometry is symmetric in the plane perpendicular to that elevation plane, this geometric symmetry cannot be exploited to simplify the Monte-Carlo method. The direction of incident solar radiation through the baffle aperture can vary in the plane perpendicular to the elevation plane, and this variation in direction of incidence invalidates the symmetry.

When all the surfaces of emission and absorption of the enclosure have been defined, we are ready to proceed with determining the distribution factors using the Monte-Carlo method. The method consists of the following five steps for each energy bundle emitted.

Step 1: Determine the Location and Direction of Emission of an Energy Bundle

Recall that the Monte-Carlo method consists of following each individual energy bundle along an optical path from emission to absorption to represent the radiation exchange between the emitting surface and all other surfaces of the enclosure. It is possible to define this optical path directly in the global coordinate system, in which all surfaces of the enclosure are related to each other. But an intermediate step renders the Monte-Carlo method easier to apply. This step

involves defining the geometrical characteristics, such as location and direction of emission, for a given surface in a local coordinate system, defined by the dimensional characteristics of that surface. Then, the geometrical characteristics are transformed to the global coordinate system. Based on this approach, the determination of all characteristics in the following paragraphs is described in the local coordinate system. The transformation from one coordinate system to another simply involves a vector-matrix multiplication.

Step 1.1: Determine the Location of Emission

The key assumption of the Monte-Carlo method is that surface emission can be uniformly divided into energy bundles, i. e. the sites of emission are uniformly distributed over the surface of emission. This subsection describes the use of uniformly distributed random numbers to achieve uniformity of emission from a given surface of the enclosure.

In the special case where the surface of emission is a planar, rectangular surface, the site of emission is defined by the appropriate pair of the following equations:

$$x = x_{min} + (x_{max} - x_{min})\, R_x , \tag{2.7}$$

$$y = y_{min} + (y_{max} - y_{min})\, R_y , \tag{2.8}$$

and $$z = z_{min} + (z_{max} - z_{min})\, R_z \,. \qquad (2.9)$$

In Equations 2.7 through 2.9 the subscripts *min* and *max* denote the extreme values of the coordinates on the surface. The quantities R_x , R_y , and R_z are three random numbers uniformly distributed between zero and unity. Note that only two coordinates of the site of emission need to be determined by any pair of the set of Equations 2.7, 2.8, and 2.9. The third coordinate is calculated from the analytical equation of the plane, which is

$$n_x (x - x_0) + n_y (y - y_0) + n_z (z - z_0) = 0 \,, \qquad (2.10)$$

where n_x, n_y, and n_z are the components of the unit normal vector to the plane and x_0, y_0, and z_0 are the coordinates of a point of the surface.

In the case where the emitting surface is a planar polygon, the surface is subdivided into triangles, as shown in Figure 8. This idea was born out of a conversation between the author and P. Villeneuve. In each triangle, *ABC*, the triplet containing the coordinates of point A and the two vectors **AB** and **AC**, denoted (*A*, **AB**, **AC**), is considered as a local coordinate system. In this coordinate system, the site of emission, *P*, can be expressed as the linear combination of **AB** and **AC**,

$$\mathbf{AP} = q\mathbf{AB} + r\mathbf{AC} \,, \qquad (2.11)$$

where the coordinates *q* and *r* of point *P* along the AB- and AC-directions are

defined by

$$\mathbf{q} = |\mathbf{AB}|\, R_q\,, \tag{2.12}$$

and

$$\mathbf{r} = |\mathbf{AC}|\, R_r\,. \tag{2.13}$$

In Equations 2.12 and 2.13 R_q and R_r are two random numbers uniformly distributed between zero and unity. This implies that point *P* is uniformly distributed on the parallelogram defined by **AB** and **AC**, and that point *P* can be located outside of the triangle *ABC*. Mathematically, this situation happens when the sum of R_q and R_r *is* greater than one. In this case point *P* is transformed to the inside of the triangle ABC by setting

$$R_q = 1 - R_r \tag{2.14}$$

and

$$R_r = 1 - R_q\,. \tag{2.15}$$

This transformation would be a symmetric orthogonal projection with respect to the BC-side of the triangle if (*A*, **AB**, **AC**) were an orthonormal coordinate system. Physically, this transformation folds two uniform distributions and places one on top of the other, so the resulting distribution is uniform.

In the case where the emitting surface is a disk, the site of emission is defined by randomly locating a point in the circumferential (ϕ) and radial (r)

directions by

$$\phi = 2\pi R_{\phi} \tag{2.16}$$

$$r = r_0 \sqrt{R_r} \ , \tag{2.17}$$

and

where R_{ϕ} and R_r are two random numbers uniformly distributed between zero and unity, and r_0 is the radius of the disk.

The uniformity of the site of emission on the surface of the disk can be shown by considering two following cases.

In the first case, consider a number N_1 of energy bundles, corresponding to N_1 random numbers ranging from 0 to R_{r1}, and falling within an area limited by R_{r1}, given by $r_1 = r_0\sqrt{R_{r1}}$. In the second case, a number of energy bundles m times greater than N_1, or $N_2 = m\ N_1$,corresponding to N_2 random numbers ranging from 0 to R_{r2} falling within an area defined by $r_2 = r_0\sqrt{m\,R_{r1}}$.

The number of energy bundles per surface area in the two cases are given by

$$\frac{N_1}{A_1} = \frac{N_1}{\pi r_o^2 R_{r1}} \ , \tag{2.18}$$

$$\frac{N_2}{A_2} = \frac{m\,N_1}{\pi r_o^2\, m\, R_{r1}} = \frac{N_1}{A_1} \ . \tag{2.19}$$

Since Equations 2.18 and 2.19 are the same for arbitrarily chosen numbers of energy bundles *m*, it can be concluded that the sites of emission are uniformly distributed over the surface area of the disk.

Step 1.2: Determine the Direction of Emission

The direction of emission is characterized by two components: the angle in the circumferential plane, ϕ, and the angle in the azimuthal plane, θ. For collimated emission, such as in the case of solar radiation entering through the MAM baffle aperture, θ and ϕ are known *a priori*. For diffuse emission, these two angles are given in Reference [30] as

$$\phi = 2\pi R_{\phi} \tag{2.20}$$

and

$$\theta = \sin^{-1}\sqrt{R_{\theta}}\ , \tag{2.21}$$

where R_{ϕ} and R_{θ} are two random numbers uniformly distributed between zero and unity. The two angles, ϕ and θ, are then used in conjunction with the normal and tangent vectors at the point of emission to determine the direction cosines of the line along which the energy bundle travels. Note that in the current effort the normal vector is chosen as the inward normal vector, and the tangent vector is chosen such

that it is always parallel to the x-axis of the local coordinate system.

Step 2: Determine the Point of Intersection of the Emitted Energy Bundle with the Enclosure Wall

All possible points of intersection between the energy bundle and the enclosure are found by solving analytical equations describing surfaces of the enclosure and the line along which the energy bundle is traveling. This line can be expressed as

$$\frac{x_2 - x_1}{l} = \frac{y_2 - y_1}{m} = \frac{z_2 - z_1}{n} \equiv T, \tag{2.22}$$

where (x_1, y_1, z_1) is the point of emission, (x_2, y_2, z_2) is the point of intersection of the line with the intercepting surface, and l, m, n are the direction cosines of the line. In order to avoid the singular case where one of the direction cosines is equal to zero, an intermediate variable T is used to rewrite Equation 2.22 as

$$x_2 = T\,l + x_1, \tag{2.22.1}$$

$$y_2 = T\,m + y_1, \tag{2.22.2}$$

$$z_2 = T\,n + z_1. \tag{2.22.3}$$

and

The substitution of Equations 2.22.1 through 2.22.3 into the analytical

equation describing the surface of interception, e. g. $f(x_2,y_2,z_2) = 0$, yields the determination of the intermediate variable T. Finally, the substitution of T back into Equations 2.22.1 through 2.22.3 yields the coordinates (x_2,y_2,z_2) of the point of intersection.

Since only one point of intersection is physically possible, all others are eliminated. This elimination consists of two steps: (1) points of intersection that are not in front of the emitting surface, as well as those that are located outside or on the opposite side of the intercepting surface are first rejected; then, (2) the distances between the point of emission and the remaining possible points of intersection are found, and the shortest of these corresponds to the point of intersection.

Step 3: Determine Whether the Energy Bundle is Absorbed or Reflected

Each surface of the enclosure is characterized by its total absorptivity. This absorptivity may be interpreted as the probability that an incident energy bundle will be absorbed. Each time an energy bundle intercepts a surface, a random number uniformly distributed between zero and unity is drawn. If that random number is less than or equal to the value of the absorptivity, the energy bundle is absorbed by the surface in question. A counter, denoted N_{ij}, containing the number of energy bundles emitted by surface i and absorbed by surface j, is incremented and the Monte-Carlo method returns to Step 1. If the random number is greater than the

absortivity, the energy bundle is reflected and the type of reflection is determined in Step 4.

Step 4: Determine Whether the Reflection is Diffuse or Specular

Each surface of the enclosure has a reflectivity ratio, R. The reflectivity ratio is the ratio of the specular component of the reflectivity to the sum of the specular and diffuse components of reflectivity, that is

$$R \equiv \frac{\rho^s}{\rho^s + \rho^d} = \frac{\rho^s}{\rho} . \tag{2.23}$$

This reflectivity ratio may be interpreted as the probability that a reflection from the surface in question is specular. A random number uniformly distributed between zero and unity is drawn to determine the type of reflection. If that random number is less than or equal to the reflectivity ratio, the reflection is specular and the Monte-Carlo method continues to Step 5. Otherwise, the reflection is diffuse and the process returns to Part 1.2 of Step 1.

Step 5: Determine the Direction of Specular Reflection

In a specular reflection, the angle between the incident energy bundle and the local surface normal is equal to the angle between the reflected energy bundle and the local surface normal, and the paths of the incident and reflected energy

bundles and the surface normal lie in the same plane. From Figure 9 and vector addition, the unit vector in the direction of specular reflection is

$$\mathbf{U_2} = \mathbf{U_1} - 2(\mathbf{U_1} \cdot \mathbf{n})\ \mathbf{n}\ , \tag{2.24}$$

where $\mathbf{U_1}$ is a unit vector along the path of the incident energy bundle, $\mathbf{U_2}$ is a unit vector along the path of the reflected energy bundle, and $\mathbf{n}$ is the unit surface normal at the point of incidence. Once the direction of specular reflection is known, the process returns to Step 2.

The five steps described above are repeated for a large number of energy bundles. The final results are the distribution factors, estimated by

$$D_{ij} = \frac{N_{ij}}{N_i}\ , \tag{2.25}$$

where N_{ij} is the number of energy bundles emitted by surface i that is absorbed by surface j, and N_i is the total number of energy bundles emitted by surface i.

3.0 The CERES MAM Model

3.1 Description of the CERES MAM and Its Operation

A layout of the CERES mirror attenuator mosaic (MAM) is given in Figure 6. A perspective view of the solar calibration module containing two identical MAM's (one for the shortwave channel and one for the total channel) is shown in the upper-right corner of the figure. Solar calibration is not used for the atmospheric window channel because energy in this band represents only 0.5 percent of the solar spectrum. The essential element of the MAM facility is the diffuser plate, which is an array of 105 spherical mirrors, manufactured on a 3.81 × 3.81 cm aluminum plate. Each spherical mirror has a radius of 1.5875 mm, and the distance between the center and the base of each mirror is 1.1938 mm. The surface area between the spherical mirrors is coated with black paint. In this study, the mirrors

are considered perfectly specular, and the black paint covering the diffuser plate is considered perfectly black. Neither assumption is true in reality, and deviations from these two assumptions may be expected to lead to deviations between experimental data and results predicted using the current model.

The diffuser plate is located inside an enclosure which consists of a baffle, a housing, and a mounting plate between the MAM housing and the radiometer aperture.

The baffle is 25.40 cm in length, and it has a field of view of ± 4 deg. This is to ensure that radiation striking the diffuser plate is not contaminated by radiation emitted from or reflected by the spacecraft. The baffle also has seven apertures regularly spaced along its length to reflect or to absorb radiation. The optical axis of the baffle is at an angle of 11 deg from the baseline of the spacecraft. This baffle axis is used as a reference to measure other angles characterizing the MAM geometry. The angle measurement also follows an unexpected sign convention: angles below the baffle axis are considered positive, while angles above this axis are negative.

The MAM housing is basically a rectangular box with a circular opening on one surface to allow radiation reflected from the diffuser plate to reach the radiometer aperture. Because the surface containing this circular opening is located on the inside wall of the enclosure, it is referred to here as the "interior wall."

Between the MAM housing and the radiometer is the mounting plate, which consists of four planar surfaces and one circular opening. This circular opening

represents the radiometer aperture, whose diameter is 2.21 cm and whose axis is at 45 deg with respect to the baffle axis. Because of the proximity of these surfaces with the radiometer, they are referred to in this thesis as the "radiometer walls."

Except for the spherical mirrors on the diffuser plate, all other surfaces of the MAM enclosure are coated with Chemglaze™ black paint, which is assumed to have a nominal absorptivity of about 0.90 and a reflectivity ratio of 0.90.

During solar calibration, the shortwave and total radiometers first look at dark space at the 162-deg position for a "zero" radiation reference, then rotate to the 263-deg solar view position to stare at the diffuser plate. At this time, the Sun begins drifting inside the field of view of the baffle, from + 4 deg (below the baffle axis) to - 4 deg (above the baffle axis). During the Sun crossing period, solar radiation enters the baffle and illuminates the diffuser plate, which reflects solar radiation back into the hemispherical space in front of it. Only a fraction of the reflected solar radiation reaches the radiometer aperture. This fraction of energy is measured as a function of the solar angle with respect to the baffle axis. Then as the Sun crosses out of the baffle field of view, the radiometers still look at the diffuser plate with no solar radiation incident on it. This is used to detect the thermal radiation noise from all the surfaces of the MAM enclosure. The difference between the thermal noise and the signal produced under full sunlight is used to re-evaluate the radiometer responsivities.

As mentioned in Chapter 2, preliminary work is required before the Monte-Carlo method can be applied. This work consists of defining the coordinates of

points making up the geometry of the enclosure. Then the enclosure is subdivided into basic surfaces such as rectangles, triangles, and disks to ensure uniform emission from each surface of the enclosure, the CERES MAM has 51 of these surfaces in addition to 105 spherical surfaces representing the mirrors. This surface subdivision is illustrated in Figure 7. Finally, each surface is subdivided into area elements. A maximum of 1200 area elements can be automatically generated for each surface, i. e. the enclosure can be subdivided into a total of about 60,000 area elements. An example of an about 5,000-area-element subdivision is shown in Figure 10.

3.2 Application of the Monte-Carlo Method

From this point the Monte-Carlo method can be applied to determine the fraction of solar radiation reflected by the diffuser plate onto the MAM enclosure. This is accomplished by emitting energy bundles from a given surface, and by following these energy bundles until they are absorbed by area elements of the enclosure. The results of this process are the distribution factors of the enclosure.

Noting that the power of solar radiation entering the enclosure, denoted Q_0 and expressed in watts, can be defined as

$$Q_0 = A_0 \, q_{sun} \cos\theta \, , \tag{3.1}$$

where A_0 is the surface area of the baffle opening in m^2, q_{sun} is the normal solar flux in the vicinity of the top of the atmosphere (1368 Wm^{-2}), and θ is the angle between the normal of the baffle aperture and the direction of the solar flux vector. The distribution factors, D_{0j}, representing the fraction of energy emitted from the baffle opening in the direction of the solar vector and absorbed by an area element j, can be used to calculate the amount of solar power absorbed by that area element j, denoted Q_{0j}; that is

$$Q_{0j} = A_0 \, q_{sun} \cos\theta \, D_{0j} \, , \tag{3.2}$$

where A_0 is the baffle aperture area and q_{sun} is the magnitude of the solar flux vector incident to the baffle opening. Thus, this amount of solar power divided by the surface area of the area element j, A_j, is the solar flux absorbed by that area element, which is

$$q_{0j} = \frac{Q_{0j}}{A_j} = q_{sun} \frac{A_0}{A_j} \cos\theta \, D_{0j} \, . \tag{3.3}$$

A FORTRAN program, named CERESMAM and presented in Appendix A, has been written to determine the distribution factors of the CERES MAM geometry and the solar flux distribution on that geometry according to Equation 3.3. This program can be operated in two modes: calculation of the fraction of solar radiation

that reaches the radiometer as a function of the angle of solar radiation entering the baffle aperture, and determination of the distribution of solar radiation reflected by the diffuser plate on the entire MAM enclosure for a given direction of solar radiation incident on the diffuser plate. Only a limited number of variables such as the total number of energy bundles emitted, the position or orientation of the diffuser plate and the radiometer axis, and the direction of incident solar radiation are user-defined; however, all other dimensional as well as radiative characteristics of the CERES MAM may be easily varied for a parametric study.

The computer code CERESMAM can emit 60 million energy bundles to determine the distribution factors of all the area elements of the MAM enclosure, for a given incident solar direction entering the MAM baffle, in less than 2 hr running on an IBM RS/6000 model 560. Note that 60 million is not the minimum number of energy bundles needed to calculate with precision the fraction of solar radiation reflected to the entire surface of the radiometer aperture. The result converges to an acceptable accuracy when about 10 million energy bundles are emitted, as shown in Table 2 and in Figure 11. That means that the program CERESMAM is capable of calculating the distribution factors of the radiometer in less than 20 min! However, a correct presentation of the distribution of energy on the enclosure requires a great number of subdivided area elements. For a subdivision of about 5000 area elements, 60 million is the number of energy bundles needed for all distribution factors to converge.

For a parametric study, for example a study of the solar radiation absorbed

by the radiometer as a function of the solar angle, Program CERESMAM can be implemented on a network of several CPU's using multi-CPU processing. On such a network, numerical values of the parameters can be defined separately for each CPU to run independently of all the others on the network. Then the results from each terminal are written to a common output file. For example, on a network consisting of one IBM RS/6000 model 560, one IBM RS/6000 model 530, and eight IBM RS/6000 model 220's, a parametric analysis using 960 million energy bundles takes 8 hr to run. The same parametric study would take 32 hr to complete on one IBM RS/6000 model 560. The gain of time is only on the order of four because the operating speed of the 220 models is only one-third that of the model 560.

4.0 Results and Discussion

4.1 Validation of the CERES MAM Model

A complete performance characterization of the CERES MAM has been performed using the results from the thermal model described in Chapter 3. This characterization can be classified into two types of study: (1) prediction of the fraction of solar radiation reflected to the radiometer aperture, and (2) prediction of the distribution of reflected radiation on the entire CERES MAM enclosure. The first study is used to orient the diffuser plate to minimize the depence of the solar radiation reflected to the radiometer on the direction of the solar vector. The second type of study predicts the radiation flux absorbed by each area element of the enclosure and is used to verify the second specification of the MAM, which is that the distribution of radiation received at the radiometer aperture must be uniform across the area of the aperture. These two types of study play important roles in predicting the performance of the MAM, including predicting the fraction of solar radiation reflected back into space by the diffuser plate. This latter is important in

order to minimize heating of the MAM cavity. A hot MAM cavity leads to thermal contamination of the calibration signal.

Because the CERES MAM is still under development—its first use is scheduled for 1997—neither ground nor flight data are available at this time. However, ground and flight data from the ERBE MAM are available in the literature. The main difference between the CERES MAM and the ERBE MAM is the presence of a black mask covering the area between the spherical mirrors of the diffuser plate in the case of the ERBE MAM. With this in mind, data from the ERBE MAM can be used as a first check of the results from the CERES MAM model. Figures 12(a) and 12(b) show the response of the ERBE radiometers as a function of the solar angle in degrees, where the surface normal of the diffuser plate is at 15 deg below the baffle axis. Measurements from ground test cases are presented in Fig. 12(a), and those from flight data are presented in Fig. 12(b). All measurements are normalized by the maximum response of the radiometers. Although the three MAM's used on NOAA-9, NOAA-10, and ERBS are supposed to be identical, the response from each MAM varies from one to the other. This is probably due to manufacturing tolerances, especially due to slight differences in orientation of the diffuser plate. But these data, either from ground test or from flight, clearly show a decreasing radiometer response for increasing solar angle in the vertical plane.

The Figure 13 shows the convention used for specifying the angles between the diffuser plate and the incoming solar energy with respect to the baffle axis. This figure also defines the vertical plane and the horizontal plane.

For comparison purposes, the CERES MAM model is used to calculate the fraction of solar radiation reflected to the radiometer aperture as a function of the solar angle varying from -10 to +5 deg. The orientation of the diffuser plate is also set at 15 deg below the baffle axis to preserve the similarity between the ERBE and CERES MAM configurations.

ERBE MAM flight data from Figure 12 (b) and results from the CERES MAM ray-trace model are plotted in the same graph in Figure 14. The solar angle in degrees is shown along the x-axis, while the fraction of solar radiation received by the radiometer aperture, normalized by the highest value and expressed in percent, is plotted on the y-axis. The dashed lines represent the ERBE flight data and the solid curve is the result from the CERES MAM model. Although the results from the model do not perfectly conform the flight data, the trend of decreasing response with increasing solar angle is seen both in the flight data and in the prediction.

4.2 Radiative Analysis

A radiative analysis consisting of defining the heat flux received by all area elements of the MAM enclosure due to solar incident to the baffle opening along the baffle axis is illustrated in Figures 15 and 16. Figure 15 shows a cut-away view of the CERES MAM for an observer located in front of and above the enclosure, while Figure 16 is viewed from the back and above the same enclosure. These two

figures are presented to verify that all internal surfaces of the enclosure have two sides and that they are correctly included in the geometry of the model. In these figures, the heat flux absorbed by each area element is presented by a color scale from dark blue to red, where red shows areas with high heat fluxes and dark blue shows areas with relatively low heat fluxes. Observation of these figures reveals that the distribution of solar energy reflected by the diffuser plate is clearly not as uniform as was assumed in the original design of the diffuser plate. Indeed, the distribution of reflected energy on the sensor walls exhibits a hot spot. This peak of energy is not at the center of the radiometer aperture. As the Sun moves across the baffle field of view, in the vertical plane from the bottom to the top, the position of the hot spot also moves across the radiometer aperture toward the rear of the MAM enclosure.

The fraction of energy arriving at the aperture opening is sensitive to the solar angle through the baffle field of view. As is shown later, this sensitivity is a result of multiple reflections in the spherical mirrors of the diffuser plate. Because this sensitivity is the same for the CERES and ERBE MAM's, it can be concluded at this point that the decreasing response of the ERBE radiometer with increasing solar angle was not caused entirely by the presence of the black mask, and that the 15-deg orientation of the diffuser plate carried over from ERBE MAM design cannot be used to meet the specifications of the CERES MAM.

4.3 Diffuser Plate Reflection Diagnosis

A deeper knowledge about reflection from the diffuser plate is needed in order to define the optimum orientation of the diffuser plate that satisfies the specifications of the CERES MAM. Recall that these specifications are: (1) the fraction of solar radiation reflected by the diffuser plate to the radiometer aperture must be independent of the solar vector incident to the MAM aperture, (2) the distribution of radiation energy on the radiometer aperture must be uniform, and (3) the fraction of solar radiation reflected back out to the baffle opening must be maximized.

For this diagnosis, three imaginary concentric hemispherical surfaces are added to the geometry of the model. These surfaces are centered at the center of the diffuser plate, and each surface is subdivided into 1200 equal-area elements. An example of the location and the area subdivision of one of these surfaces is illustrated in Figure 17. Using Equation 3.3 and the distribution factors, the heat flux through each imaginary hemispherical surface area element can be determined by counting energy bundles reflected from the spherical mirrors of the diffuser plate that pass through each imaginary area element. When that number is divided by the total number of energy bundles traced and by the element surface area, the result is a quantity that is proportional to the radiative flux through each area element. A three-dimensional plot of these fluxes can be used to illustrate the

spatial distribution of radiation energy reflected by the diffuser plate.

Figure 18 shows the spatial distribution of radiation reflected by the diffuser plate for the case where the solar angle is along the baffle axis and the normal direction of the diffuser plate is oriented at 15 deg below the baffle axis. The reflected energy is presented here as viewed from an observer standing in front and slightly to the right of the diffuser plate. Figure 19 shows the same distribution but for an observer located directly on the right side of the diffuser plate. A color scale from blue to red is used to differentiate the intensity of the heat flux, with blue color corresponding to low heat fluxes and red corresponding to relatively high fluxes. These two figures clearly show that the reflection by the diffuser plate is not uniform. In fact, the reflection has three irregular features. First, the reflection pattern does not completely fill the 2π-space in front of the mirror mosaic. Rather it is concentrated in a cone angle limited by the dotted lines in Figure 19. Second, there exists a specular hot spot; that is, the angle between the line passing through the center of this spot and the normal direction of the plate is equal to the angle between the solar direction and the normal direction of the plate. Third, the heat flux in the lower half of the reflection cone is higher than that in the upper half.

To understand the reflection pattern of the diffuser plate, which is the combination of 105 spherical mirrors, a study of the reflection by one spherical mirror is needed. Figure 20 shows a schematic of such a study where a section of the spherical mirror is presented at the top of the figure. Below this mirror is an imaginary surface of emission, from which collimated energy bundles are emitted

within a beam limited by two points A_1 and C_1. Energy bundles, or rays, emitted at point A_1 strike the right edge of the mirror at point A_2, and are reflected to point A_3 according to the law of specular reflection. Other rays emitted from a particular point, denoted B_1 in the current diagram, are reflected by the mirror at point B_2 in such a way that the reflected ray passes just below the right edge of the mirror to point B_3. Thus the two paths $A_1A_2A_3$, and $B_1B_2B_3$ represent the limit of the reflection cone. All rays emitted within these two paths are directly reflected by the spherical mirror to the hemispherical space in front of the mirror. There exists a region on the mirror, limited by two points B_2 and C_2, where all incident energy bundles are multiply reflected within the mirror, as indicated by the path $C_1C_2C_3C_4$, for example. This multiple reflection within each mirror is the cause of the specular hot spot, or the multiple reflection spot, as it will be called from now on.

Within the reflection cone and except for the multiple reflection spot, it is reasonable to consider the direct reflection by a spherical mirror to be uniform. In this case the magnitude of reflection can be represented by a sphere centered at the center of the mirror. Note that the distance from the center of the mirror to a given point on the sphere is proportional to the radiative power in the direction defined by that point. Then the reflection by several spherical mirrors adjacent to each other can be described graphically as shown in Figure 21. In this figure an example of only two mirrors is used to construct the reflection pattern in question, where the dotted lines represent the reflection pattern from two mirrors centered at O_1 and O_2, and the solid line represents the reflection from both mirrors. The

construction of the solid line consists of drawing a line from the center of both mirrors (point O) and in an arbitrary direction, OA for example. Then the length OB is the addition of OB_1 and OB_2. The resulting reflection is obtained by repeating the graphic construction in all possible OA-directions. Considering the normal of the diffuser plate passing through point O as a reference line which divides the reflection cone into the upper half and the lower half regions. The model, although simple, clearly demonstrates that the radiation reflected by the diffuser plate is higher in the lower region of the reflection cone. This difference in intensity is called the "multiple mirror effect," because it is due to the addition of reflections from several mirrors.

Recall that the multiple reflection spot and the multiple mirror effect are the causes of the sensivity of the radiometer response with varying solar angle. But the multiple reflection spot and the multiple mirror effect occur only in the lower region of the reflection cone, whereas the upper region has no irregular features. That means that if the radiometer axis were located in the upper region, the reflected energy received by the radiometer would be independent of the solar vector. But instead of moving the radiometer axis from the lower region to the upper region, it would be simpler to keep the direction of the radiometer axis unchanged with respect to the baffle axis and rotate the diffuser plate, so that its normal direction is above the baffle axis. In this case the unwanted multiple reflection effects would occur in the upper region and the radiometer would receive only direct reflections from the diffuser plate. In other words, the response of the radiometer

should be less sensitive to the solar angle!

4.4 Diagnosis of the Diffuser Plate Reflection for Different Diffuser Plate Orientations

In order to exploit the unexpected phenomenon revealed in the previous section by rotating the diffuser plate to avoid the multiple reflection effect on the radiometer aperture, further investigations of the reflection by the diffuser plate are needed. The results from the previous section are based only on one particular orientation of the diffuser plate: 15 deg below the baffle axis.

Figures 22 through 26 show the spatial distribution of radiative energy reflected from the diffuser plate for various plate angles (12, 8, 4, 0, and -4 deg). For these cases, energy enters the MAM enclosure parallel to the baffle axis. These figures are presented as viewed by an observer located directly to the side of the diffuser plate.

Comparison of Figures 23 and 19 reveals that the multiple mirror effect becomes more significant with smaller diffuser plate angles. The area of multiple reflections is larger in the upper reflection region for a plate angle of 8 deg (Figure 23) than for an angle of 15 deg (Figure 19). In fact, as the angle between the diffuser plate normal and the baffle axis becomes smaller, the reflection cone becomes wider, which explains the increasing area of the multiple mirror effect. But

the most relevant information from Figure 23 is the almost complete absence of the multiple reflection spot. The radiative energy on the hemispherical surface decreases regularly. This indicates that the existence of the multiple reflection spot depends on the angle of solar incidence.

The limit of incident angle needed to avoid multiple reflection can be found using the diagram illustrated in Figure 27, which shows a section of a spherical mirror where the radius is R (1.5875 mm), the distance between the center and the base of the mirror is d (1.1938 mm), and the line LC represents the normal direction of the mirror. In this diagram, the angle α can be expressed as

$$\alpha = \cos^{-1}\left(\frac{d}{R}\right). \qquad (4.1)$$

Numerically, the angle α is 41.23 deg, and β, the complement of α, is 48.76 deg. The line AL represents a limiting case in which an energy bundle emitted from point A strikes the mirror at point L and is then reflected in the direction LR. Since the reflections from the mirror are specular, the direction of incidence is the angle (β-α), or 7.53 deg. This means the multiple reflections exist only for incidence angles greater than $\beta - \alpha$ = 7.53 deg. Recall that Figure 23 is for incidence angle equal to 8 deg, thus the effect of multiple reflections is hardly visible in this figure.

Figures 23 through 26 show the reflections from the diffuser plate as the orientation of this plate varies from +8 deg to -4 deg, or, in other words, as the plate normal moves gradually from below the baffle axis to above. Observations of these figures confirm the absence of multiple reflections for diffuser plate orientations of

less than 8 deg with respect to the baffle axis. The results from this section are important because the elimination of the multiple reflection spot in the vicinity of the radiometer aperture is one of the objectives of the current research.

4.5 Radiative Analysis Using the Diffuser Plate Orientation as a Parameter

4.5.1 Comparison of CERES MAM Model Results with Measured Data

Although the set of valid diffuser plate angles has been greatly reduced as described in the previous section, a more detailed study about the reflection properties of the diffuser plate is needed to define the optimal orientation of the plate. This can be done by using the CERES MAM ray-trace model to determine the distribution factors of the radiometer aperture for all possible solar angles and diffuser plate orientations. This task is a time-consuming process since the determination of the distribution factor for one set of parameters takes almost 2 hr. A complete radiative analysis taking into account 16 values for the solar angle (from -10 to + 5 deg) and 12 values for the diffuser plate orientation (from +15 to -8 deg), or 192 sets of parameters, would require about 400 hr of CPU run time for an IBM RISC model 560 to complete! But thanks to a network of ten workstations of the Mechanical Engineering Computing and Analysis Center of VPI&SU, the above-

mentioned task can be completed in about 72 hr!

Before using the simulated results from the CERES MAM model in a parametric study, it is necessary to check these results with a new set of measured CERES MAM data, which became available after preliminary results of the model were presented to the Radiation Sciences Branch of NASA Langley Research Center. Figure 28 shows a comparison between the CERES MAM model and the measurements from the CERES MAM enginnering model manufactured by TRW, the NASA contractor charged with the design and the construction of the CERES MAM. The measurements of radiation reflected by the diffuser plate and guided to the radiometer aperture as a function of solar angle have been performed for the case where the normal direction of the diffuser plate is set along the baffle axis. Then the measured data have been normalized by the maximum value of the set of measurements. Accordingly, the CERES MAM model has been set for the same configuration and the calculated distribution factors have been normalized by the highest value of all the distribution factors. In Figure 28, the measured data are represented by the curve joining the triangular symbols, whereas the results from CERES MAM model are shown by the curve joining the circular symbols. As expected, the predicted results fit the measured data very well.

4.5.2 Parametric Radiative Analysis

Determination of the normalized distribution factor of the radiometer aperture as a function of the solar angle using the orientation of the diffuser plate as a parameter, has been performed with the CERES MAM model. This study consists of emitting energy bundles from the baffle opening in the direction defined by a given solar angle for a given orientation of the diffuser plate. The distribution factor of the radiometer aperture for each configuration, found by the Monte-Carlo ray-trace method, is then weighted by the value of the distribution factor corresponding to the -3-deg solar angle and +15-deg diffuser plate orientation. ***This particular configuration is considered the reference point for comparison of all simulated results from the CERES MAM model. The weighted ditribution factor is called the normalized distribution factor.***

Figure 29 shows the results of the current parametric study, where the normalized distribution factor of the radiometer, expressed in percent, is plotted against the solar angle, expressed in deg. As observed, for diffuser plate orientations decreasing from +15 deg to +8 deg, the curves representing the distribution factor in general decrease with decreasing plate angle. As this happens, the curves become much flatter over a wider range of angles. This is because the multiple reflection effect of the diffuser plate is reduced as the angle between the plate normal and the baffle axis decreases. This effect vanishes for

a diffuser plate orientation approaching +8 deg as demonstrated in Section 4.4. However, the uniformity of the distribution factor function is important only for solar angles varying from -4 to +4 deg. This is because for all solar incidence angles outside the baffle field of view, the diffuser plate is not completely illuminated by solar radiation. Consequently, the reflection from the plate would deteriorate for solar angles beyond ±4 deg.

Other results showing the distribution factor of the radiometer aperture function are presented in Figure 30. In this figure, the orientation of the diffuser plate varies from +6 to -8 deg. The curves in this figure are plotted with a scaling factor in the vertical axis different from that of Figure 29 so that the individual lines could be observed. In reality, these curves would have been plotted on top of each other and it would have been difficult to differentiate between them. The two extreme curves joining the circular symbols correspond to the 6 deg and -8 deg orientations of the diffuser plate. Intermediate angles are represented by the dotted lines. A striking remark here is that the distribution factor function first decreases as the diffuser plate normal approaches the baffle axis from below (from +15 to +8 deg in Figure 29), and then increase as the plate normal approaches and passes above the baffle axis (from +6 to -4 deg in Figure 30).

The above remark suggests another way to analyze the set of data used to construct Figures 29 and 30. Figure 31 is a contour plot of the distribution factor of the radiometer aperture as a function of both the solar angle and the diffuser plate angle. In this figure, the solar angle is varied from -10 to +5 deg along the

horizontal axis, the diffuser plate angle is varied from +15 to - 8 deg along the vertical axis, and the distribution factor is shown as a percentage by a color scale increasing from blue to red. The color scale is concentrated in the 50 to 70 percent interval to show more detail. Three-dimensional representation of Figure 32 is also shown in Figure 32. Figure 32 is from the point of view of an observer located almost along the horizontal axis. It is worth noting that the color scale in Figure 32 has 20 levels, in an interval ranging from 50 to 75 percent, so that each layer between two contour lines represents approximately one percent of variation.

Study of these figures indicate that the distribution factor function peaks at the 15-deg diffuser plate orientation. Then as this angle decreases to 8 deg, the function approaches a minimum. For diffuser plate orientations less than 8 deg, the distribution factor function starts to increase, and becomes almost constant over a wider range of solar angles for diffuser plate angles varying from 0 to -8 deg. In other words, the surface representing the distribution factor function has a plateau between diffuser plate orientations of 0 and - 8 deg. For diffuser plate angles less than -8 deg, the radiometer is outside of the reflection cone, and therefore the signal arriving at the radiometer aperture decreases sharply. Thus, it can be concluded that any diffuser plate orientation between 0 and -8 deg will provide a reflection from the diffuser plate to the radiometer aperture that is relatively invariant with solar angle. It must be taken into account that the distribution factor function increases with decreasing diffuser plate angle. The optimal orientation of the diffuser plate is at the center of the [0, -8 deg] interval, which is the -4-deg orientation.

To further justify the -4-deg configuration, recall that the baffle axis has been used so far in this research as the reference axis to determine the diffuser plate orientation, which is the angle between the normal of the plate and the reference baffle axis. But the solar angle incident to the diffuser plate can vary within the field of view of the MAM baffle, which is ±4 deg (or 4 deg above and below the baffle axis). The angle between the diffuser plate normal and the angle of incoming solar energy must fall within the interval [0, -8 deg]. This guarantees that the distribution factor to the radiometer aperture will fall on the plateau discussed previously. As a worst case, shown in Figure 33, the solar angle will be 4 deg above the baffle axis. Then in order for the diffuser plate normal to remain above, or at least parallel to, the solar vector, the optimal diffuser plate angle is 4 deg above the baffle axis.

4.5.3 Study of the CERES MAM performance Between the 0 and -4 Deg Diffuser Plate Orientations

In order to quantify the CERES MAM performance for comparative purposes the surface area of the radiometer aperture has been subdivided into 80 equal-area elements. The CERES MAM ray-trace model is used to calculate the distribution factors of all area elements of the radiometer aperture, as well as the mean value and the standard deviation of these distribution factors. The mean value of the distribution factors can be used to compare the fraction of solar radiation received by the radiometer aperture for different diffuser plate orientations, while the standard

deviation is an excellent tool to show how the received energy is distributed over the surface area of the radiometer aperture.

Figures 34 and 35 represent the mean value and the standard deviation of the distribution factors on the radiometer aperture as a function of the solar angle in the vertical plane. Figure 34 shows results for a diffuser plate angle of 15 deg, while Figure 35 shows results for plate angles of 0 and -4 deg. The 15-deg configuration is used here as a reference for comparison. As can be observed, there is no difference in the standard deviations calculated for the 0- and -4-deg configurations, and their mean values are almost constant compared to that of the 15-deg configuration. The uniformity of the distribution of energy on the surface area of the radiometer aperture is acceptable for both 0- and -4-deg configurations. Figure 35 also shows that the fraction of solar radiation received by the radiometer aperture in the -4-deg configuration is greater than that of the 0-deg configuration. This result clearly favors the -4-deg configuration over the 0-deg configuration.

Other simulated results from the CERES MAM model, such as the distribution of solar radiation reflected on the MAM enclosure and the fraction of solar radiation reflected back out to the baffle opening by the diffuser plate as a function of solar angle, are compared for both configurations in Figures 36 through 38. As indicated by these figures, there is no discernible difference between the two configurations. Figure 38 shows that relatively little power—7 percent at the most—is reflected back out of the baffle aperture.

From the comparative study in this section, it can be concluded that both the

0- and -4-deg configurations can be used to meet the CERES MAM specifications. But because the -4-deg configuration provides a higher level of reflected power to the radiometer aperture, this configuration may be considered the optimum design of the CERES MAM.

4.6 Option to Improve the CERES MAM Performance

The results from the previous section have shown that a configuration in which the diffuser plate is oriented 4 deg above the baffle axis could be used to meet the CERES MAM specifications. And although the solar radiation received by the radiometer aperture is insensitive to the solar angle for that particular orientation of the diffuser plate, this fraction of radiation received by the radiometer aperture was only 68 percent of the maximum of the reference case, where the diffuser plate is oriented at 15 deg below the baffle axis.

This fraction of radiation depends strongly on the distance between the diffuser plate and the radiometer aperture. It is then possible to increase the fraction of the received radiation by reducing this distance. It is not possible to move the location of the radiometer aperture, as this would require major changes in the geometry of the instrument module. Therefore, the best option is to translate the diffuser plate forward and along the baffle axis. For example, a 1-cm translation of the diffuser plate would increase the fraction of radiation received by the

radiometer from 68 to about 75 percent of the reference value. Results of the proposed translation are presented in Figure 39. In this figure, the normalized distribution factor of the radiometer aperture is plotted against the solar angle. The dotted line joining the square symbols is the result from the case where the diffuser plate is oriented at 15 deg below the baffle axis. This result is used here as a reference. The solid line joining the light circular symbols represents the normalized distribution factor of the radiometer as a function of the solar angle varying in the vertical plane for the case where the diffuser plate is oriented at 4 deg above the baffle axis and translated 1 cm forward along this axis. The line joining the dark circular symbols is the plot for the same conditions but plotted against the variation of the solar angle in the horizontal plane. Two important observations can be drawn from this figure. First, within the baffle field of view, the normalized distribution factor is constant regardless of the solar angles. Second, the normalized distribution factor is increased to a level of about two-thirds of the maximum reference.

Other improvements are demonstrated by the following results. Figure 40 shows the distribution of solar radiation reflected by the diffuser plate onto the entire enclosure of the CERES MAM with the proposed configuration and for solar radiation entering parallel to the baffle axis. Compared to other configurations, as shown in Figures 15 and 16, the reflection by the diffuser plate is now more concentrated on the radiometer aperture, and the distribution of energy received by this aperture is much more uniform. Figures 41 and 42 show the distribution of solar radiation reflected onto the radiometer walls for various solar angles in the

vertical and horizontal planes. These figures reveal that the distribution of energy remains unchanged as the solar angle varies in both the vertical and horizontal directions. Figure 43 is the result from the case where the diffuser plate is oriented at 15 deg below the baffle axis. This is used here as a reminder of how nonuniform the distribution of energy is for the original ERBE diffuser plate angle.

Recall that all numerical results obtained so far have been calculated with a ray-trace model using 60 million energy bundles. It is possible to improve the graphic representation of the results from the CERES MAM model by increasing the number of surface-area elements and the number of energy bundles. Figure 44 shows the distribution of solar radiation on the MAM enclosure subdivided into about 40,000 area elements and for one billion of collimated energy bundles emitted from the baffle opening. The resolution of this figure is greatly improved compared to that of Figure 41. This refined result takes about 29 hr to run on an IBM RS/6000 model 560.

5.0 Conclusions and Recommendations

5.1 Conclusions

A radiative thermal model based on the Monte-Carlo ray-trace method has been developed for the CERES MAM. Results obtained from this model show that:

1. The Monte-Carlo method used to model the CERES MAM provides realistic radiative characteristics of the instrument.
2. The decrease in the amount of solar radiation reflected to the radiometer aperture with increasing solar angle is not caused entirely by the presence of the black mask on the ERBE MAM as had been previously hypothesized.
3. The 15-deg orientation of the diffuser plate carried over from the ERBE MAM design cannot be used to meet the specifications of the CERES MAM.
4. Multiple reflections within each spherical mirror of the diffuser plate exists for

all diffuser plate orientations greater than 8 deg.

5. The optimum orientation of the diffuser plate was found to be 4 deg above the baffle axis. With this configuration, the reflection of energy from the diffuser plate to the radiometer aperture is insensitive to variations of the solar incidence direction in both the vertical and horizontal planes.

6. The distribution of radiation energy across the radiometer aperture is uniform for all diffuser plate orientations ranging from 0 to 8 deg above the baffle axis.

7. The fraction of solar radiation reflected back out to the baffle opening is essentially the same for the diffuser plate oriented at 0 and 4 deg above the baffle axis.

8. The reflected radiation received by the radiometer aperture can be increased by translating the diffuser plate forward along the baffle axis. This translation does not affect the uniformity of the reflected of energy.

5.2 Recommendations

1. It is recommeded that the CERES MAM diffuser plate be oriented 4 deg above the baffle axis. This particular orientation of the diffuser plate is

chosen because it provides:

- The maximum amount of reflected energy from the diffuser plate to the radiometer aperture which is at the same time insensitive to the incident solar vector.
- A uniform distribution of reflected radiation across the surface of the radiometer aperture.
- A maximum of radiation reflected back out of the enclosure through the baffle opening to minimize the thermal noise emission from the enclosure walls.

2. A determination of the distribution factors for emission from the radiometer aperture to all other surfaces of the enclosure should be performed. Then using reciprocity of the distribution factors as described by Equation 2.3, the radiation exchange between the radiometer aperture and all other surfaces of the enclosure, or the thermal radiation noise from these surfaces, could be quantified without difficulty.

3. The radiative thermal model of the CERES MAM should be linked to the end-to-end model of the CERES radiometric instrument to evaluate the actual response of the radiometric instrument during solar calibration and to compare this calibrated response with that of a representative Earth scene viewed by the radiometric instrument.

Table 1. CERES Program Chronology [10].

Satellite	Sponsor	Number of CERES Instrument Sets	Mission Period
TRMM[1]	NASA	1	1997-2000
EOS-AM1[2]	NASA	2	1998-2003
EOS-PM1[3]	NASA	2	2001-2005
TRMM-2	NASDA[4]	2	2000-2003
EOS-AM2	NASA	1	2003-2008
EOS-PM2	NASA	1	2005-2010
EOS-AM3	NASA	1	2008-2013
EOS-PM3	NASA	1	2010-2015

[1] Tropical Rainfall Measurement Mission

[2] Earth Observing System - AM crossing time

[3] Earth Observing System - PM crossing time

[4] National Space Development Agency (Japan)

Table 2. Convergence of Radiometer Aperture Distribution Factor, DF, Due to Collimated Radiation Entering Parallel to the Baffle Axis

Number of Energy Bundles Emitted	Distribution Factor	Convergence[1] in percent	CPU Run Time[2]
1,000	1.000×10^{-3}	N/A	0.3 s
2,000	0.500×10^{-3}	50.00	0.4 s
3,000	0.333×10^{-3}	33.40	0.5 s
5,000	0.600×10^{-3}	80.18	0.7 s
50,000	0.640×10^{-3}	6.66	5.4 s
3,000,000	0.771×10^{-3}	20.51	5 m 40 s
5,000,000	0.760×10^{-3}	1.46	9 m 30 s
10,000,000	0.763×10^{-3}	0.39	17 m 57 s
20,000,000	0.764×10^{-3}	0.13	35 m 20 s
30,000,000	0.762×10^{-3}	0.26	53 m 20 s
40,000,000	0.763×10^{-3}	0.13	1 h 4 m
50,000,000	0.762×10^{-3}	0.13	1 h 26 m
60,000,000	0.762×10^{-3}	0.00	1 h 44 m
80,000,000	0.761×10^{-3}	0.13	2 h 21 m

[1] $\text{Convergence} = \frac{\left| DF_{i-1} - DF_i \right|}{DF_i} \times 100$

[2] h: hour, m: minute, s: second

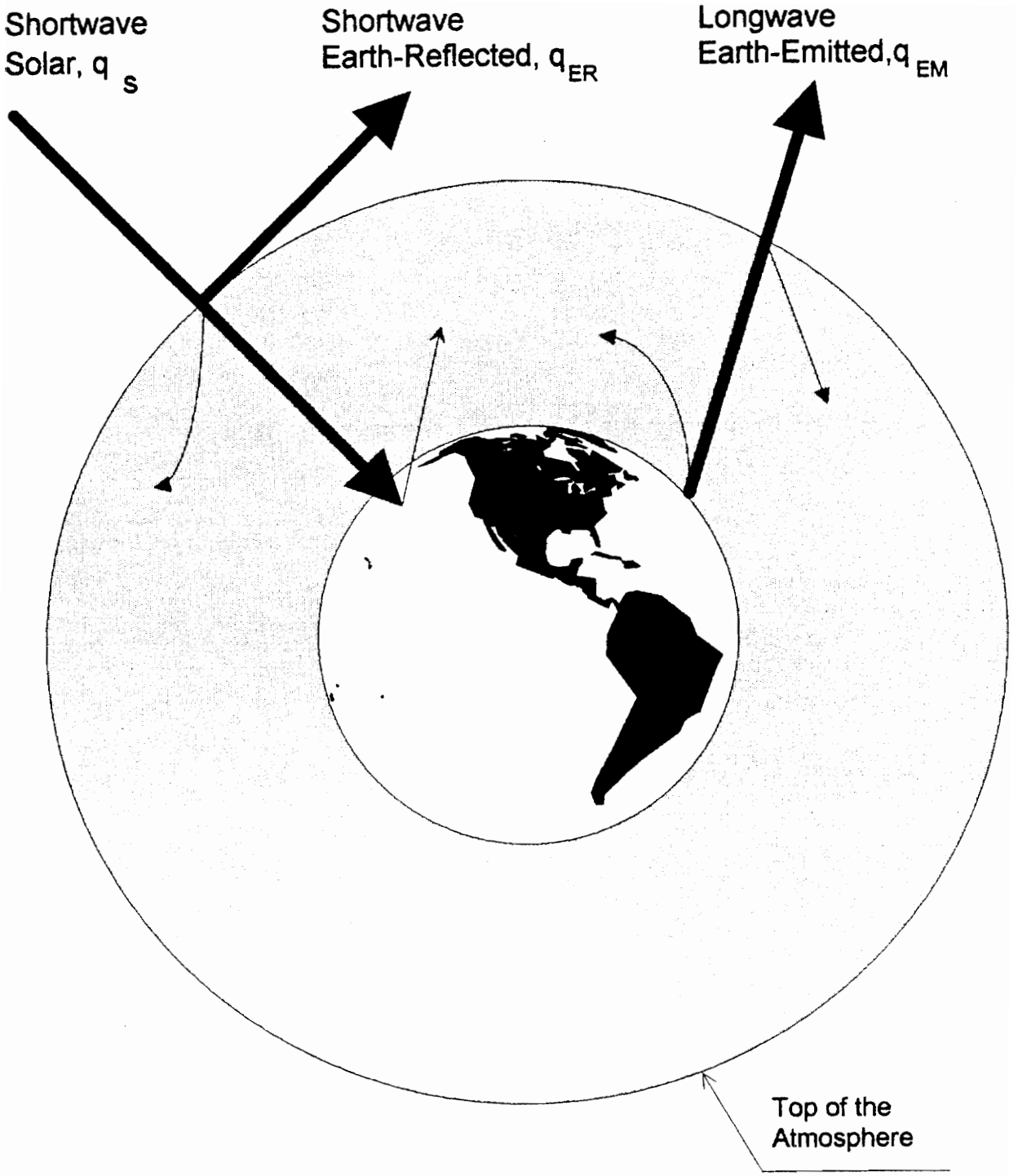


Fig. 1 Radiative Fluxes Contributing to the Earth Radiation Budget.

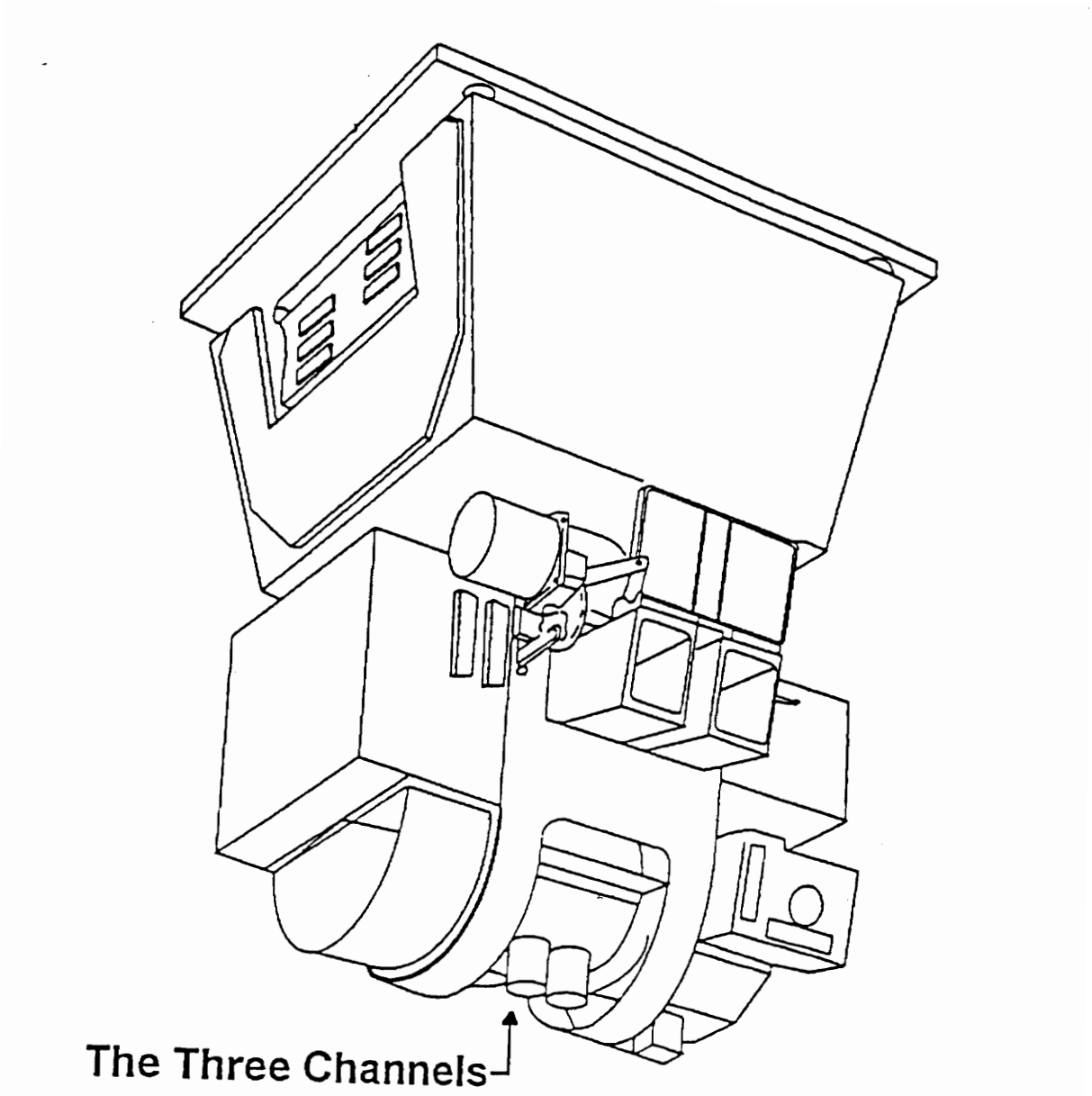


Fig. 2 The CERES Scanning Module [10].

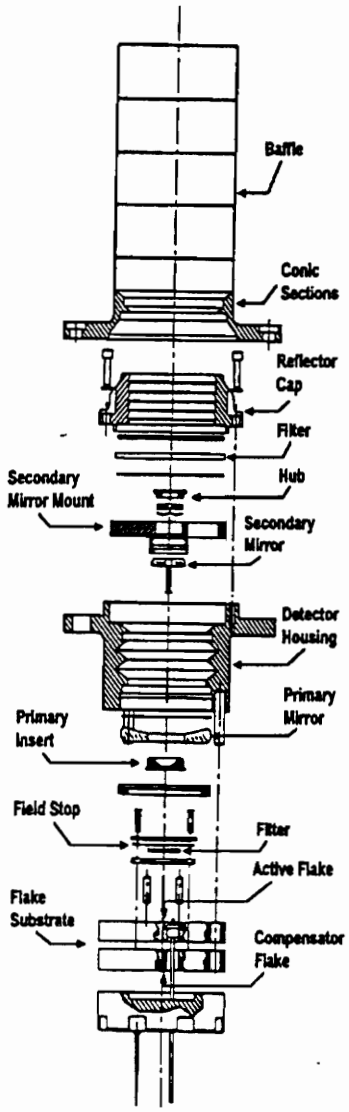
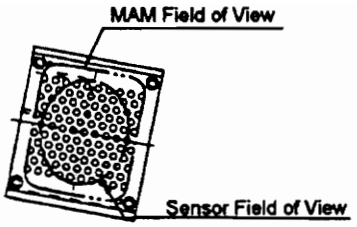
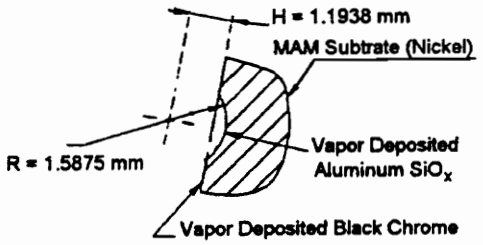


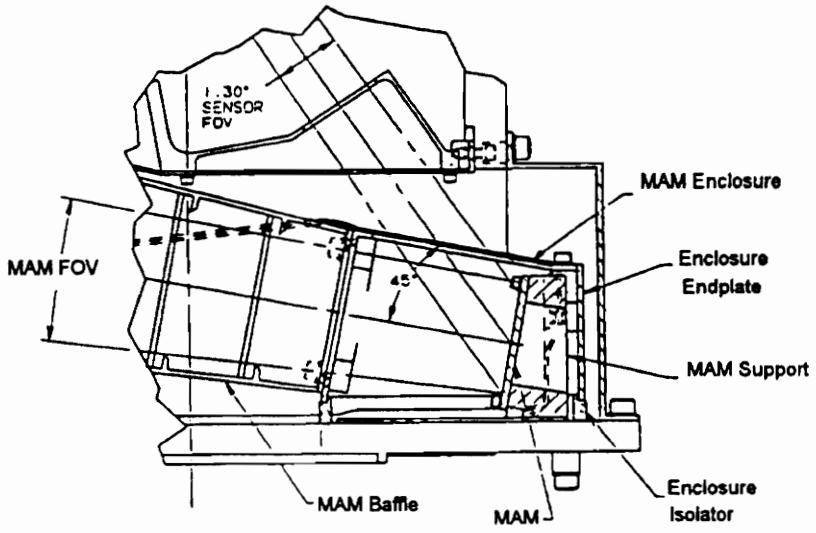
Fig. 3 Exploded View of CERES Radiometer Instrument [14].



(a) Mirror Attenuator Mosaic (MAM)



(b) Solar Attenuator Detail



(c) MAM Support

Fig. 4 The MAM Design [14]: (a) Mirror Attenuator Mosaic, (b) Attenuator Detail, and (c) MAM Support.

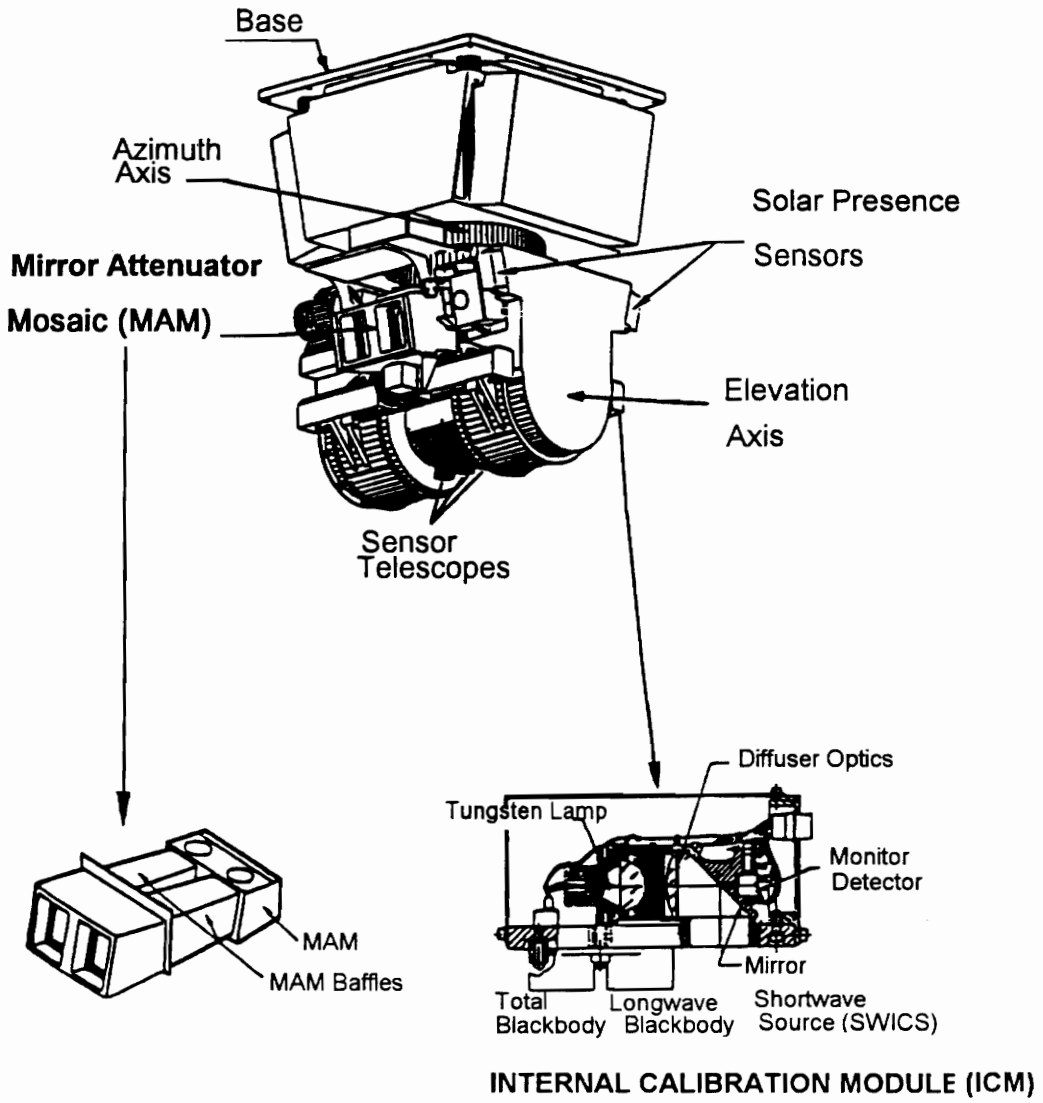


Fig. 5 The CERES In-Flight Calibration Systems [12].

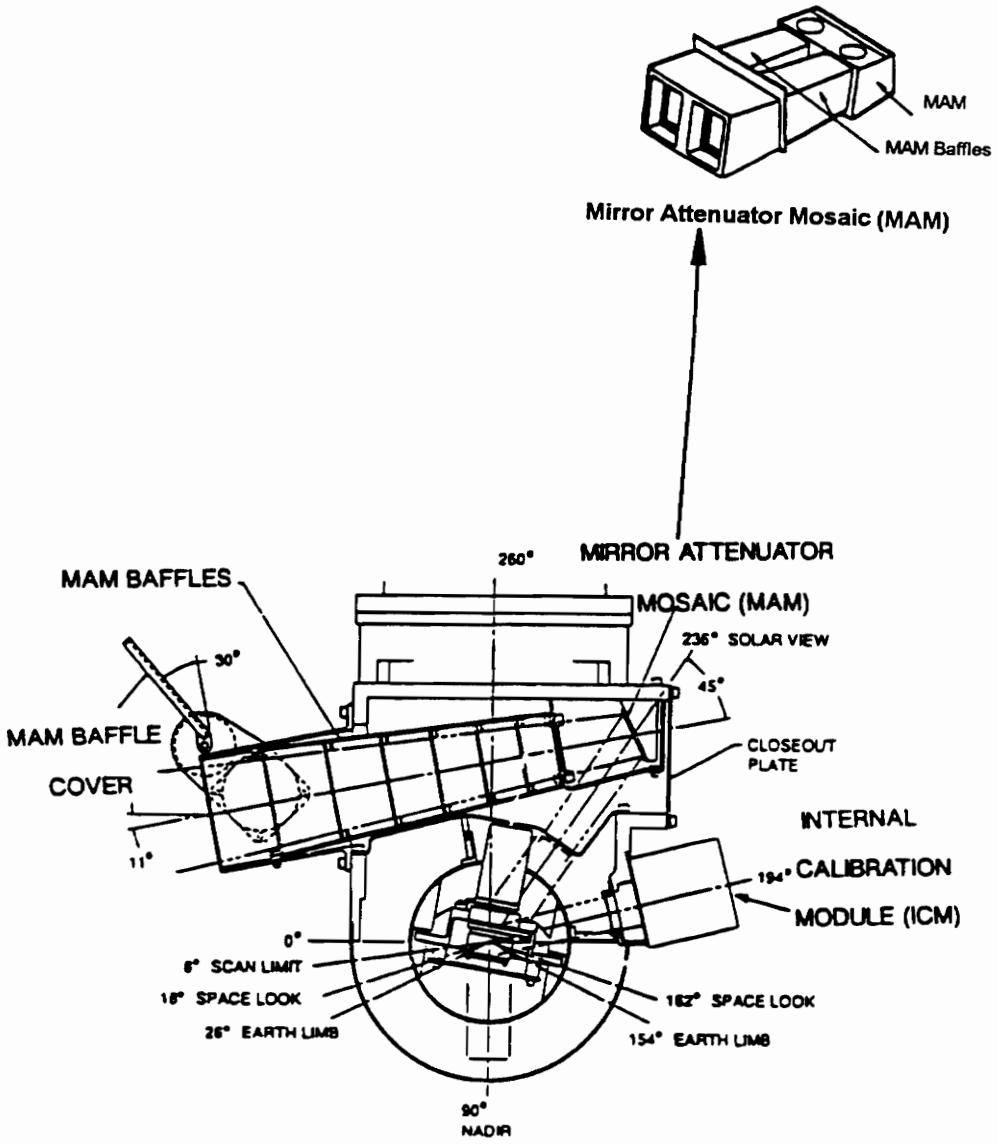


Fig. 6 The Scan Positions [12].

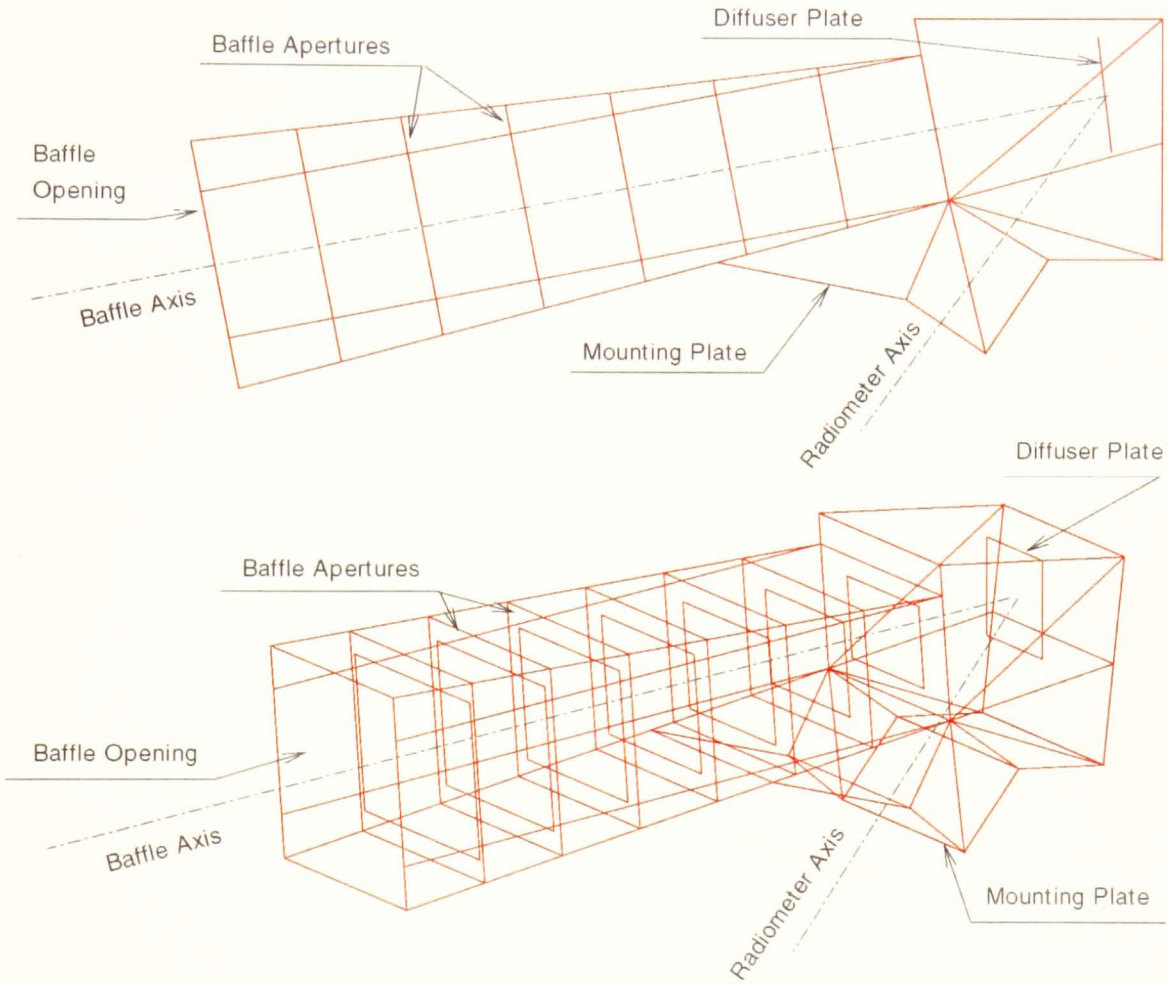


Fig. 7 Surface Subdivision of CERES MAM Model.

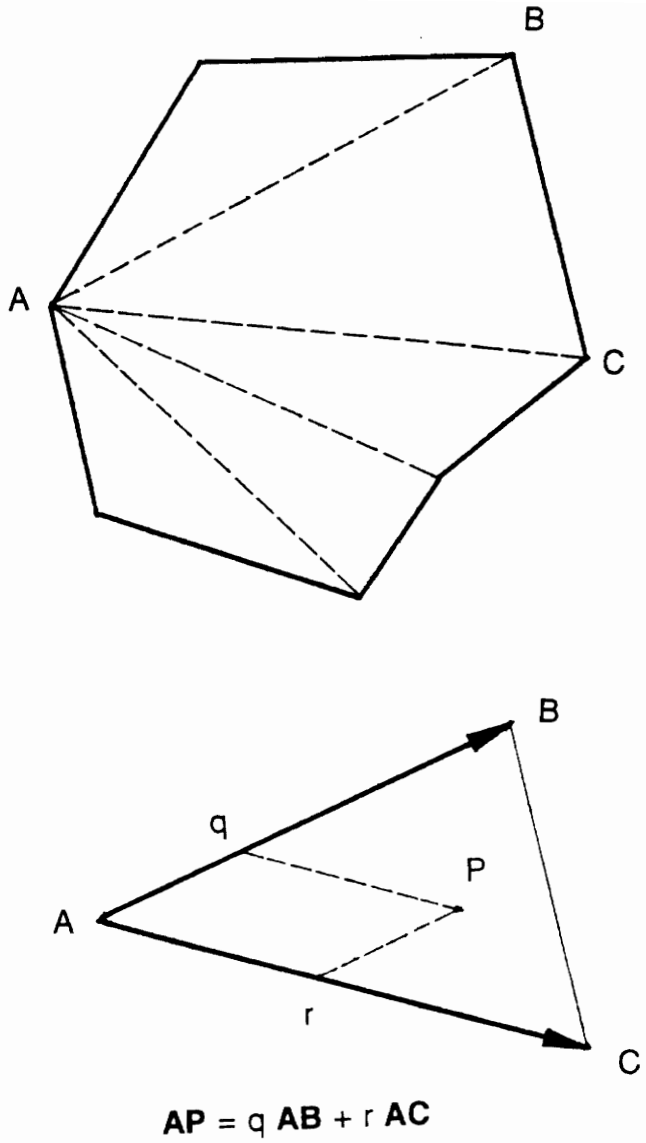


Fig. 8 Triangular Surface Subdivision.

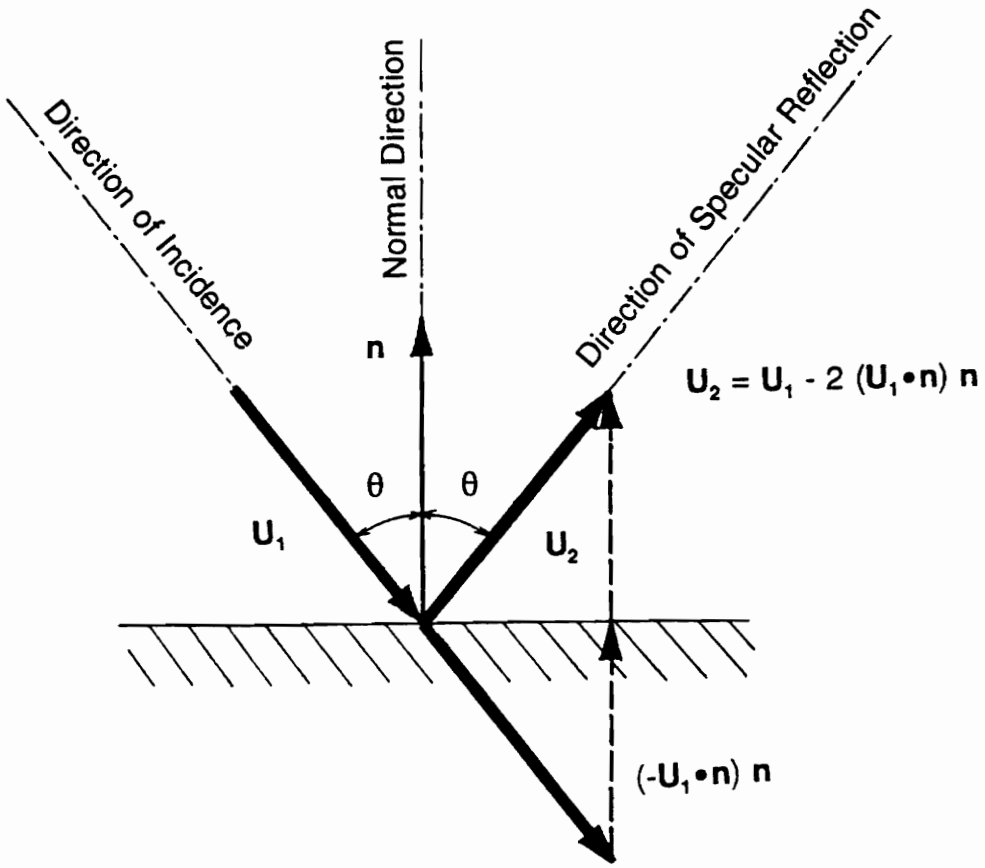


Fig. 9 Direction of Specular Reflection.

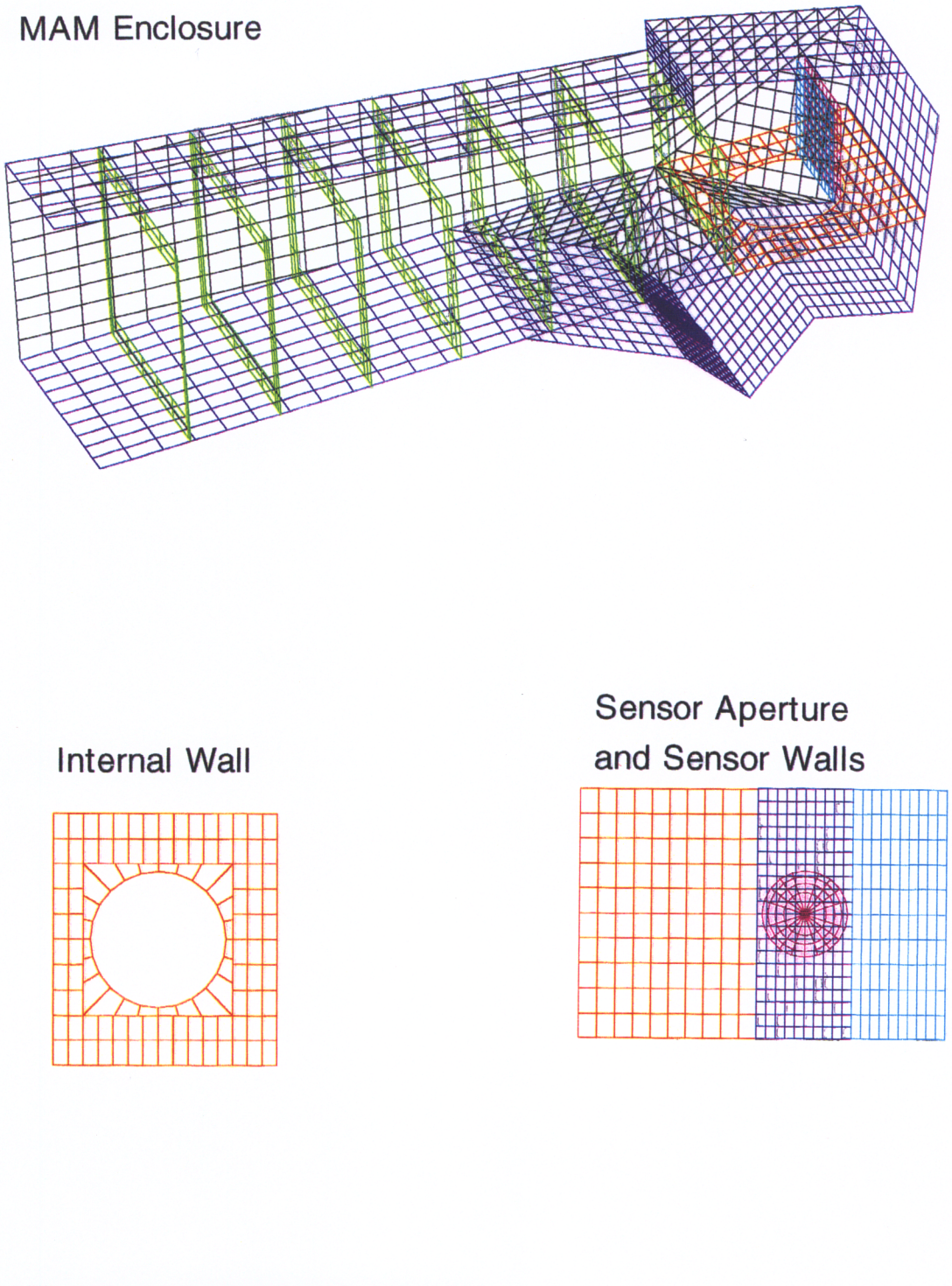


Fig. 10 Area Element Subdivision.

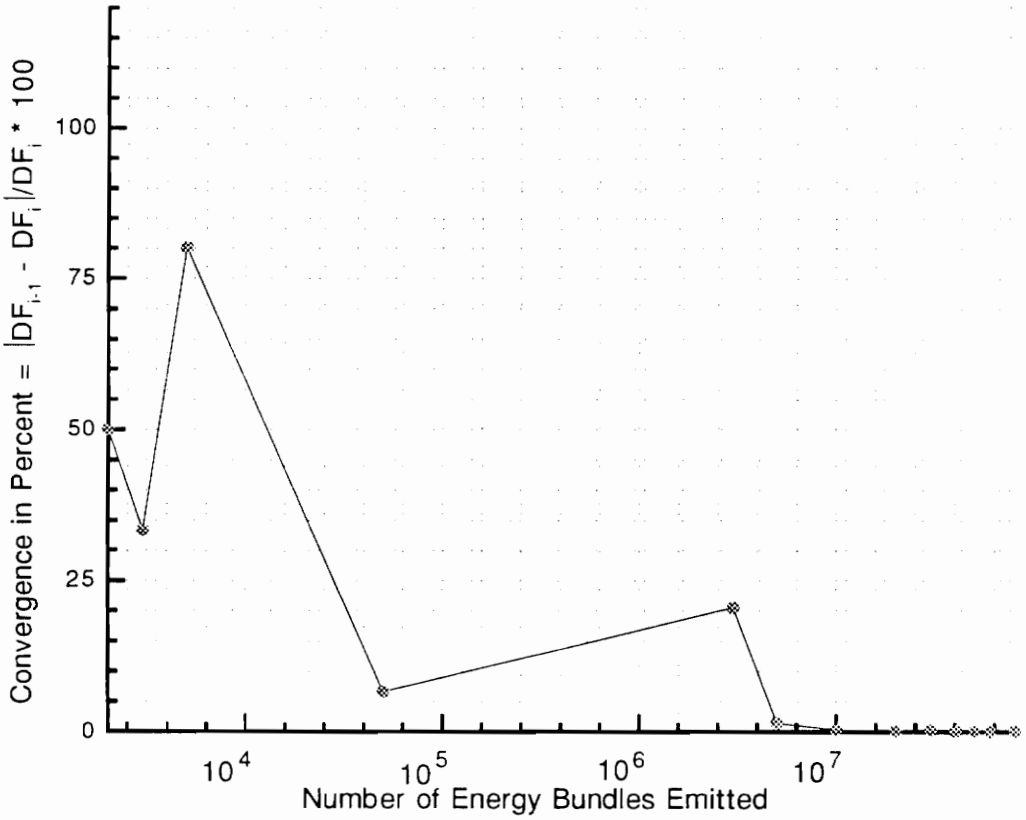
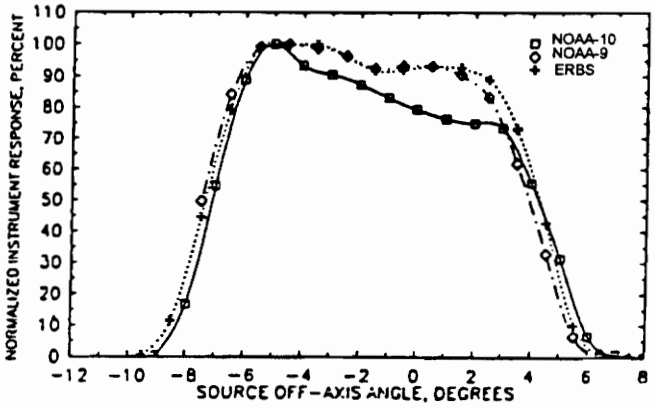
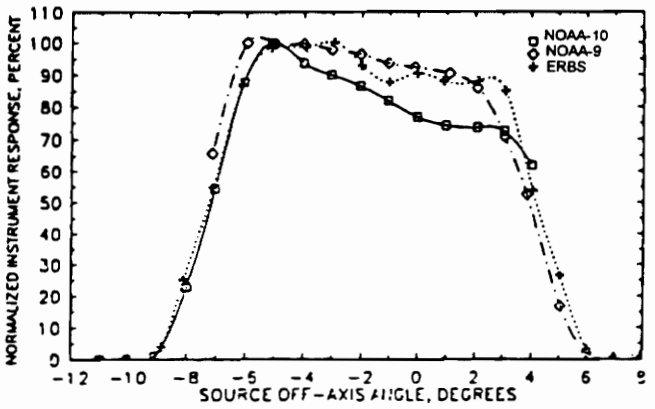


Fig. 11 Convergence of Distribution Factors of Radiometer Aperture Due to Collimated Radiation Entering the Baffle Opening Along the Direction of the Baffle Axis (Data from Table 2).



(a) Ground Data



(b) Flight Data

Fig. 12 ERBE MAM (a) Ground and (b) Flight Data [25].

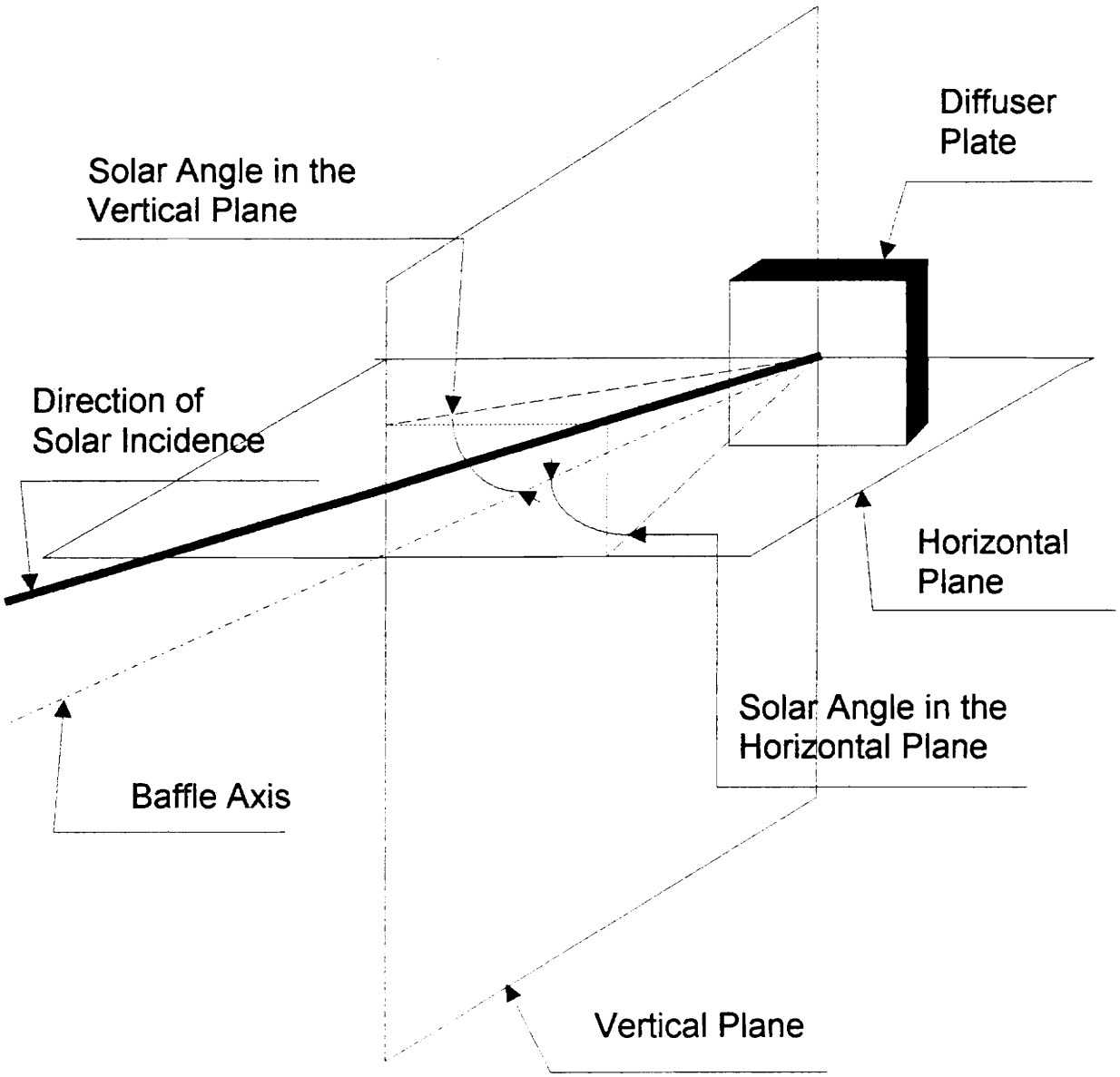


Fig. 13 Definition of Solar Angles (Diffuser Plate Normal Not Shown for Clarity).

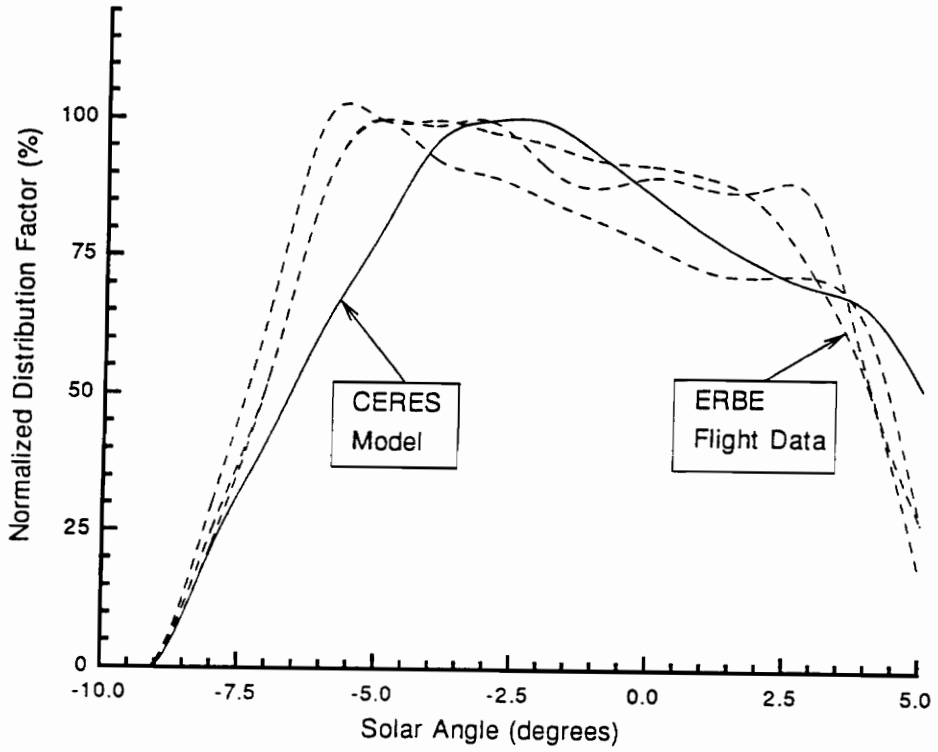


Fig. 14 Comparison of ERBE MAM Flight Data [25] with Numerical Results from CERES MAM Model.

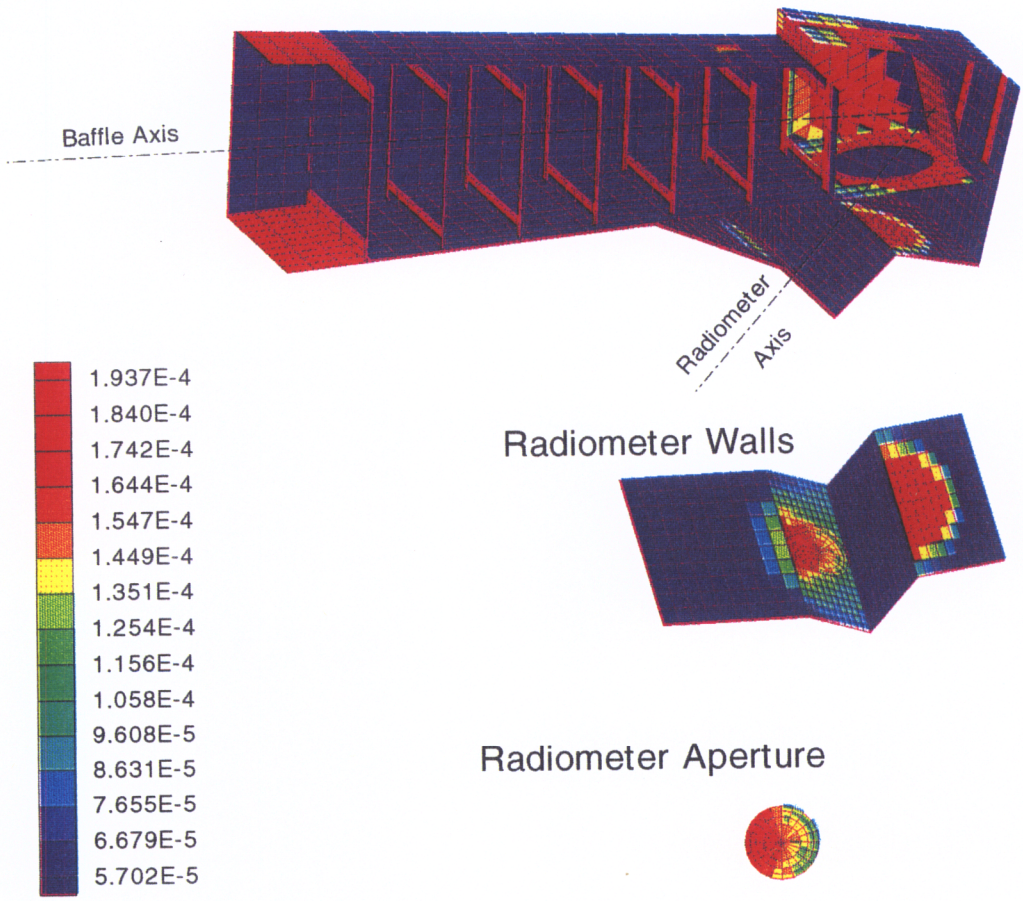


Fig. 15 Distribution of Solar Radiation on the MAM Enclosure for Collimated Radiation Incident Along the Baffle Axis and with the Diffuser Plate Oriented at 15 deg Below the Baffle Axis, Viewed from the front of the Enclosure.

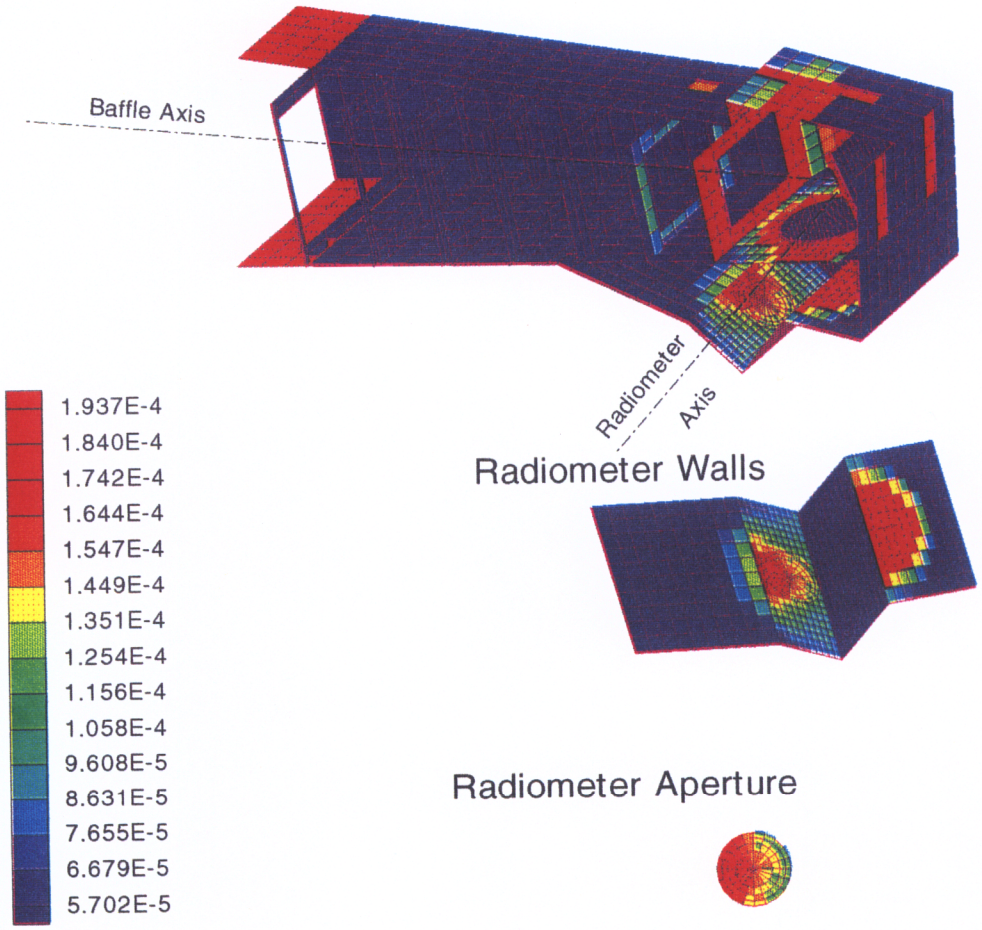


Fig. 16 Distribution of Solar Radiation on the MAM Enclosure for Collimated Radiation Incident Along the Baffle Axis and with the Diffuser Plate Oriented at 15 deg Below the Baffle Axis, Viewed from the Back of the Enclosure.

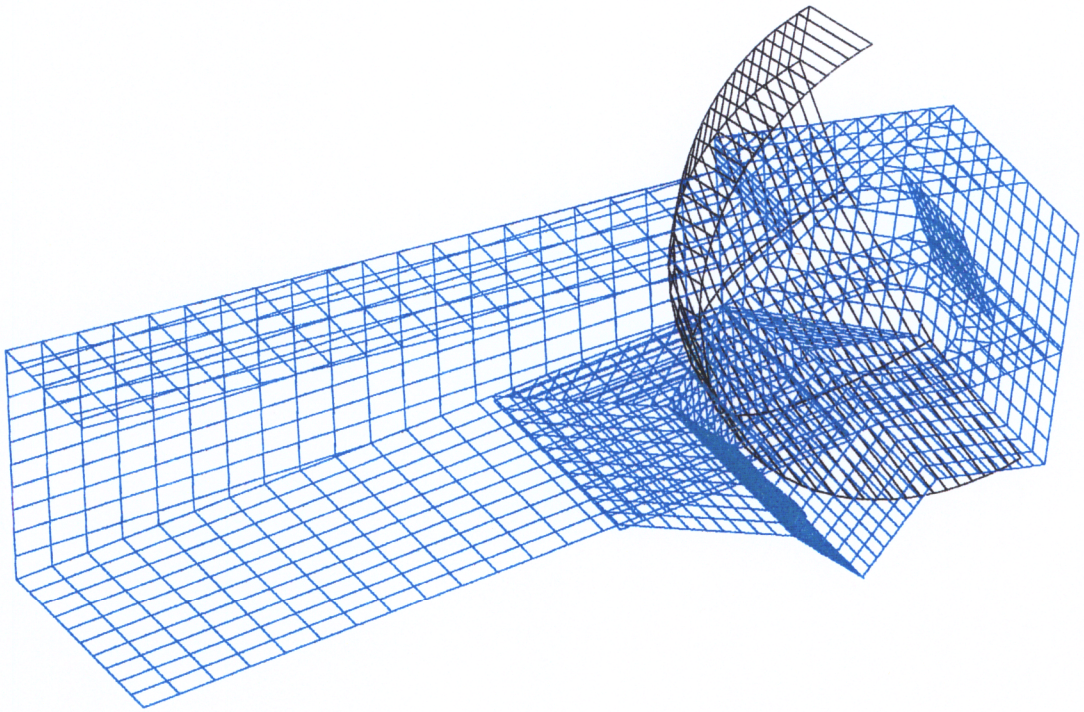


Fig. 17 Imaginary Hemispherical Surface (Shown in Black) Added to the CERES MAM Model Geometry for Diffuser Plate Reflection Diagnoses.

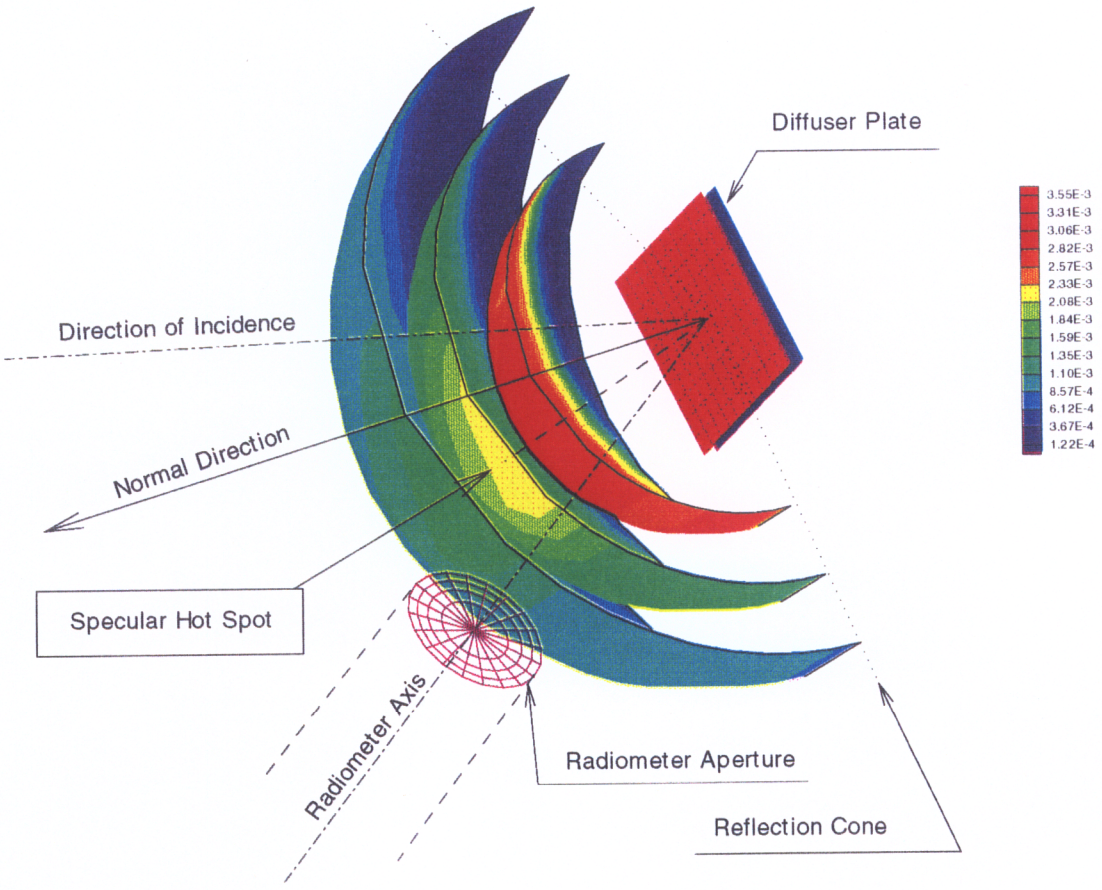


Fig. 18 Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented at 15 deg Below the Baffle Axis, Observed from the Front of the Diffuser Plate.

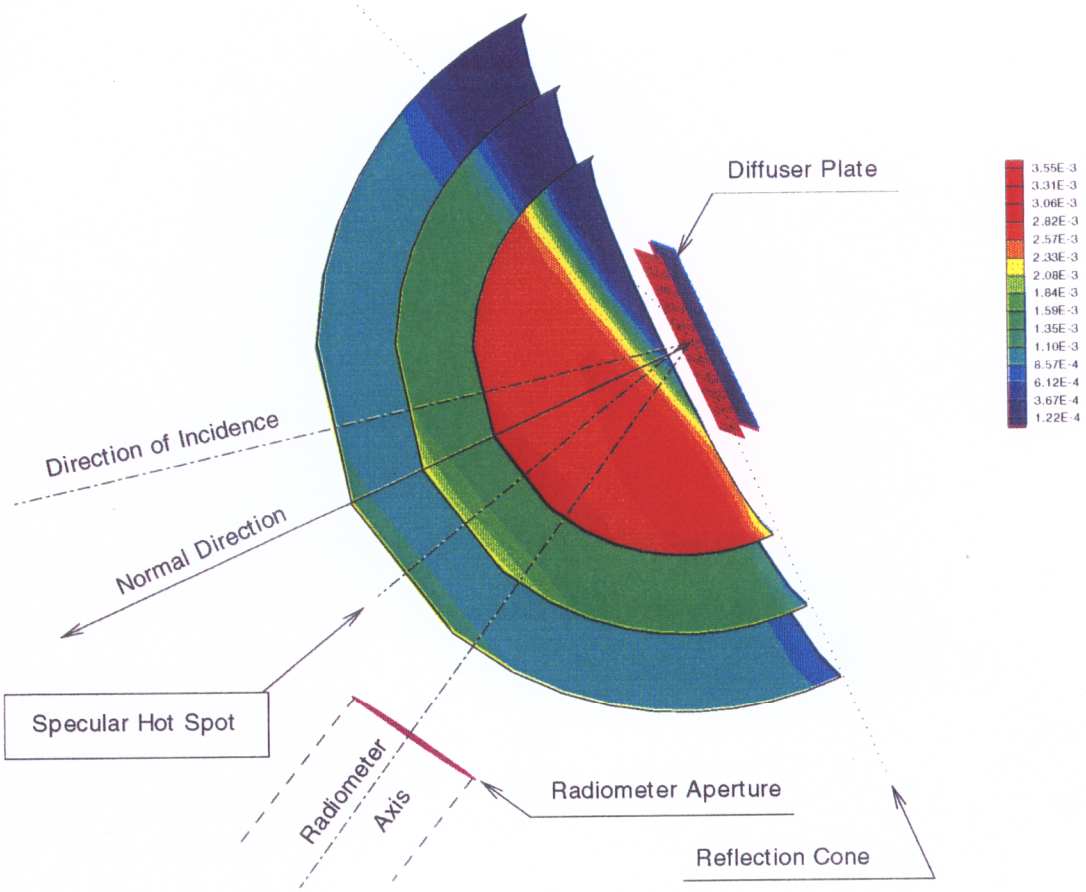


Fig. 19 Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented at 15 deg Below the Baffle Axis, Observed from the side of the Diffuser Plate.

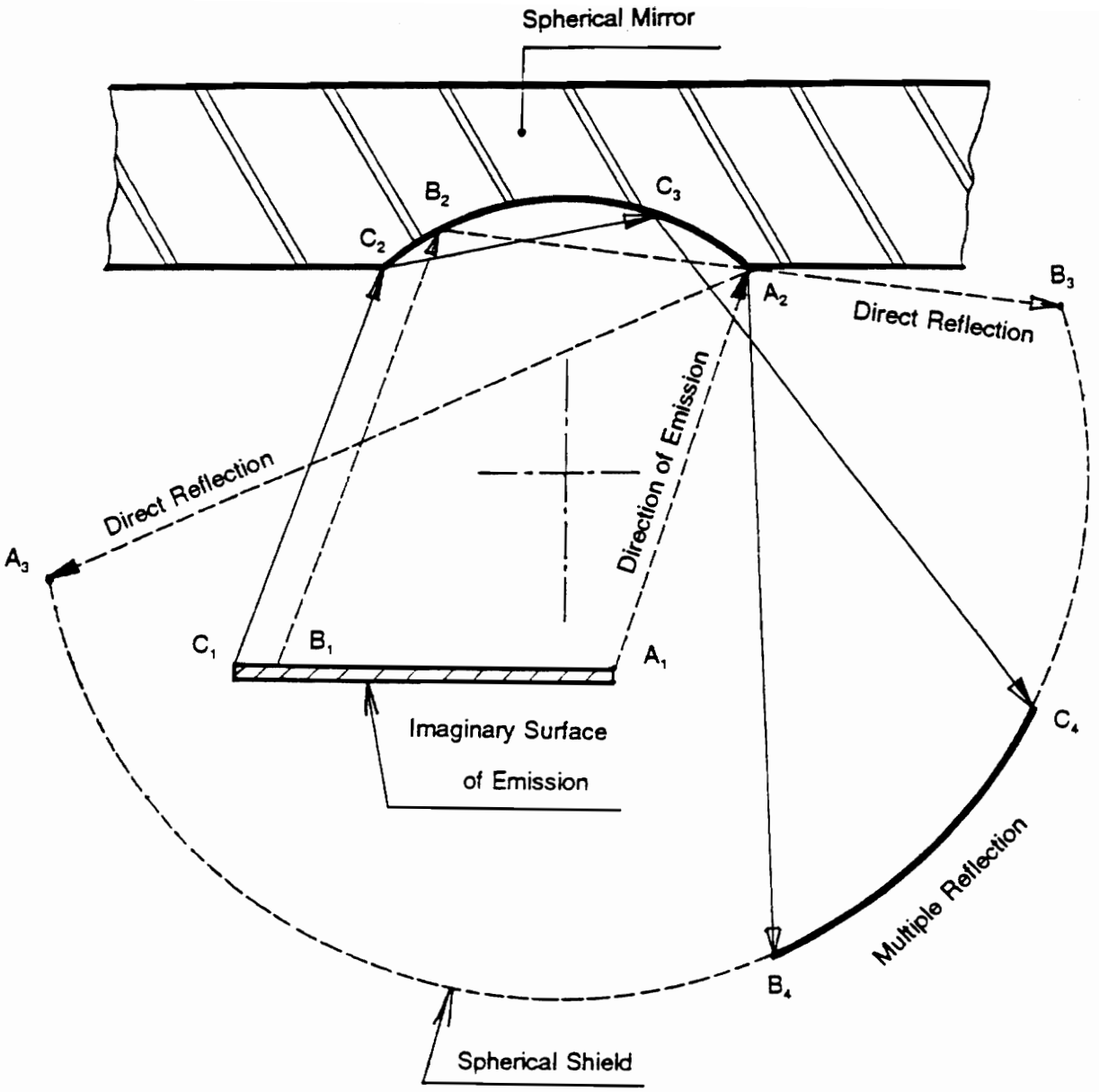


Fig. 20 Multiple Reflection Effect.

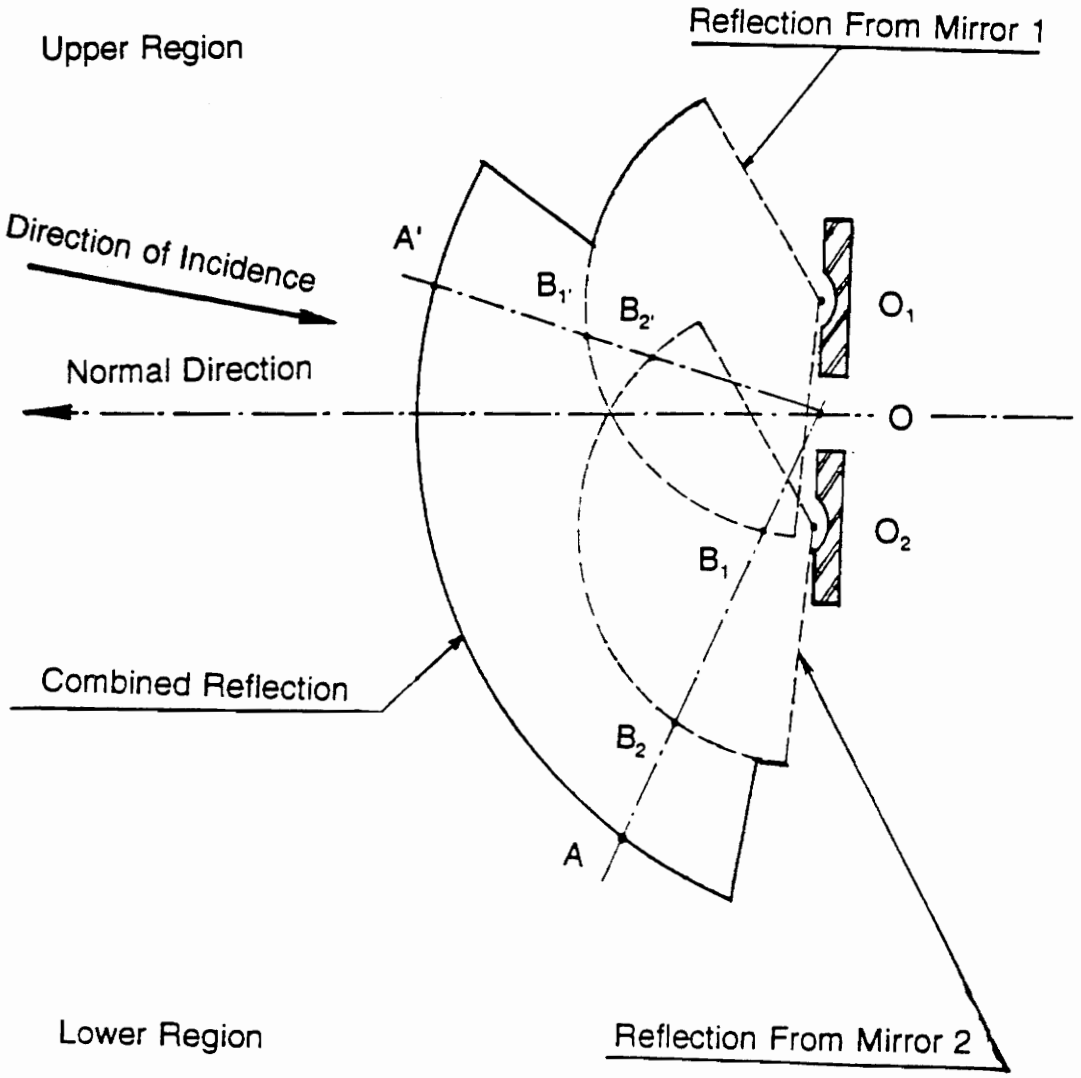


Fig. 21 The Multiple Mirror Effect.

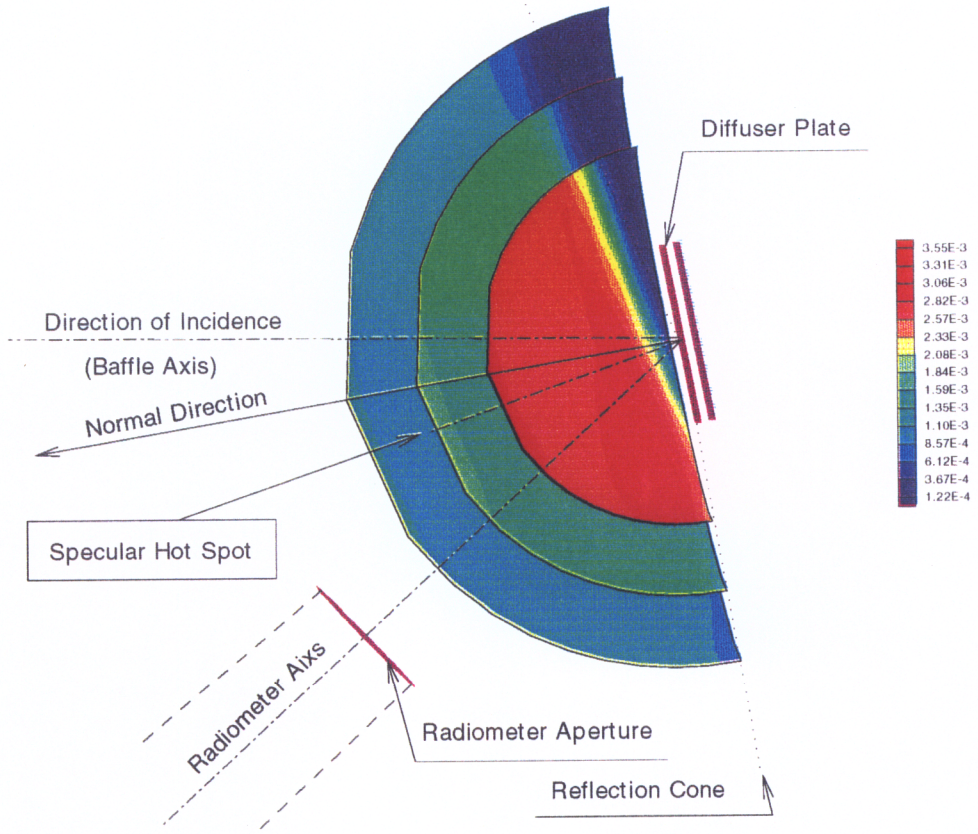


Fig. 22 Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented 12 deg Below the Baffle Axis.

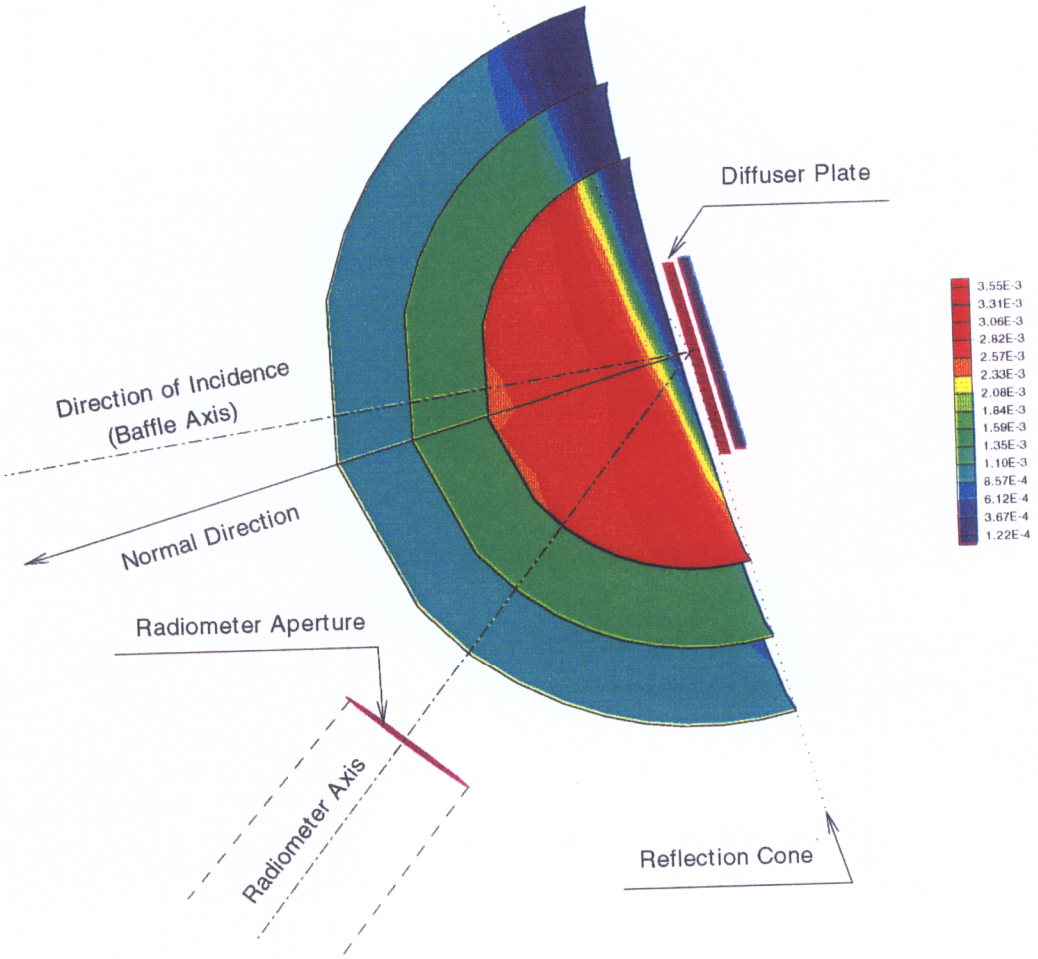


Fig. 23 Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented 8 deg Below the Baffle Axis.

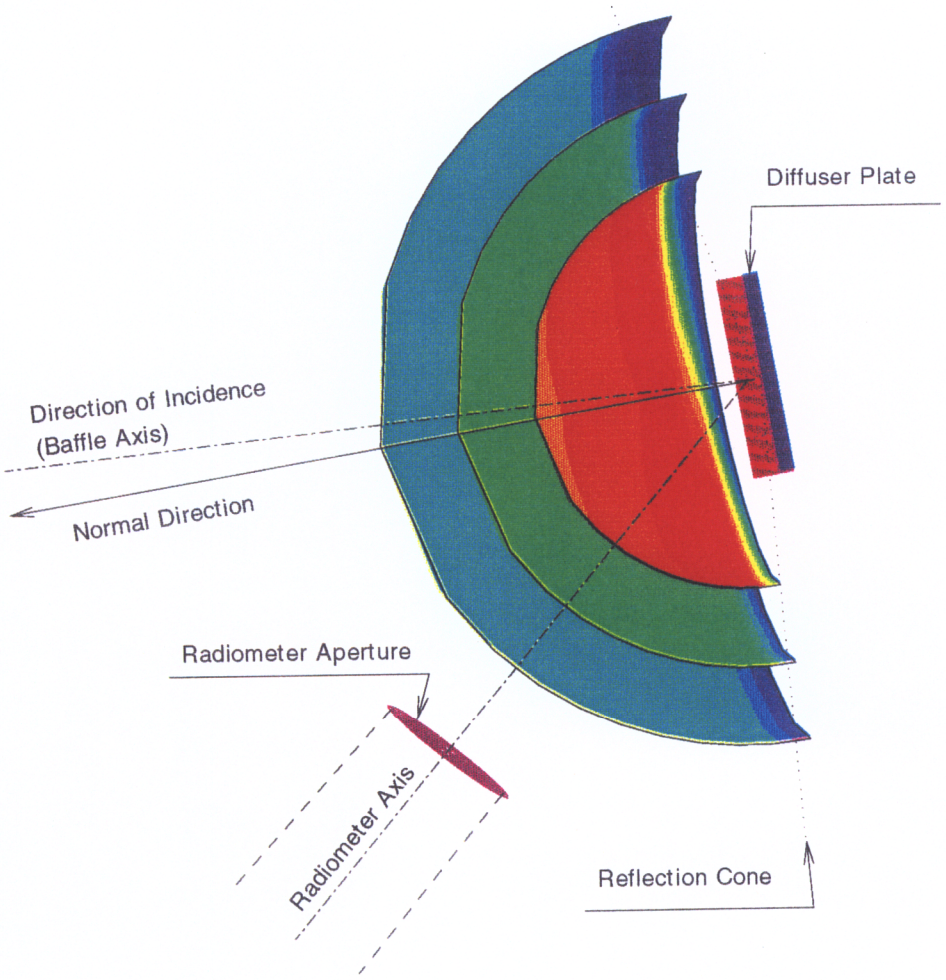


Fig. 24 Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented 4 deg Below the Baffle Axis.

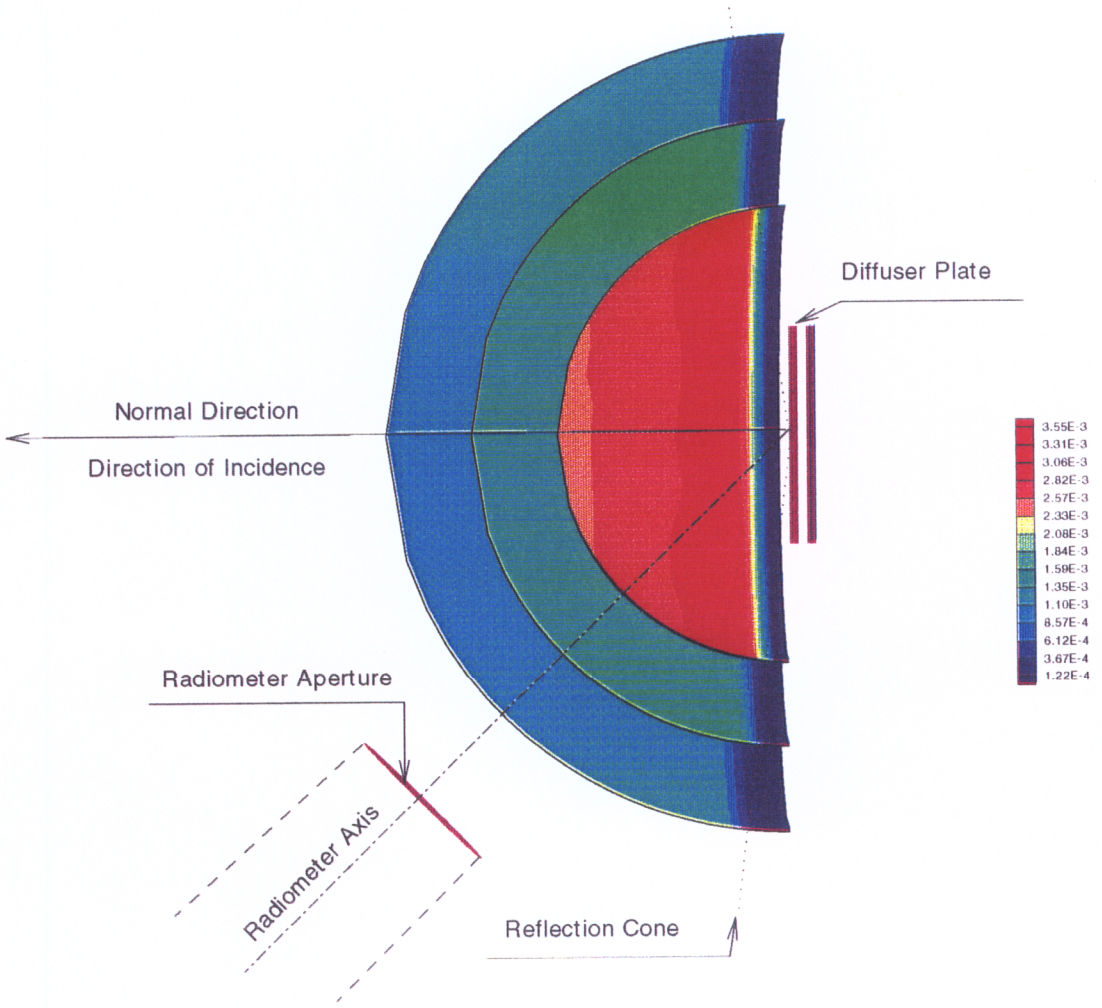


Fig. 25 Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented Along the Baffle Axis.

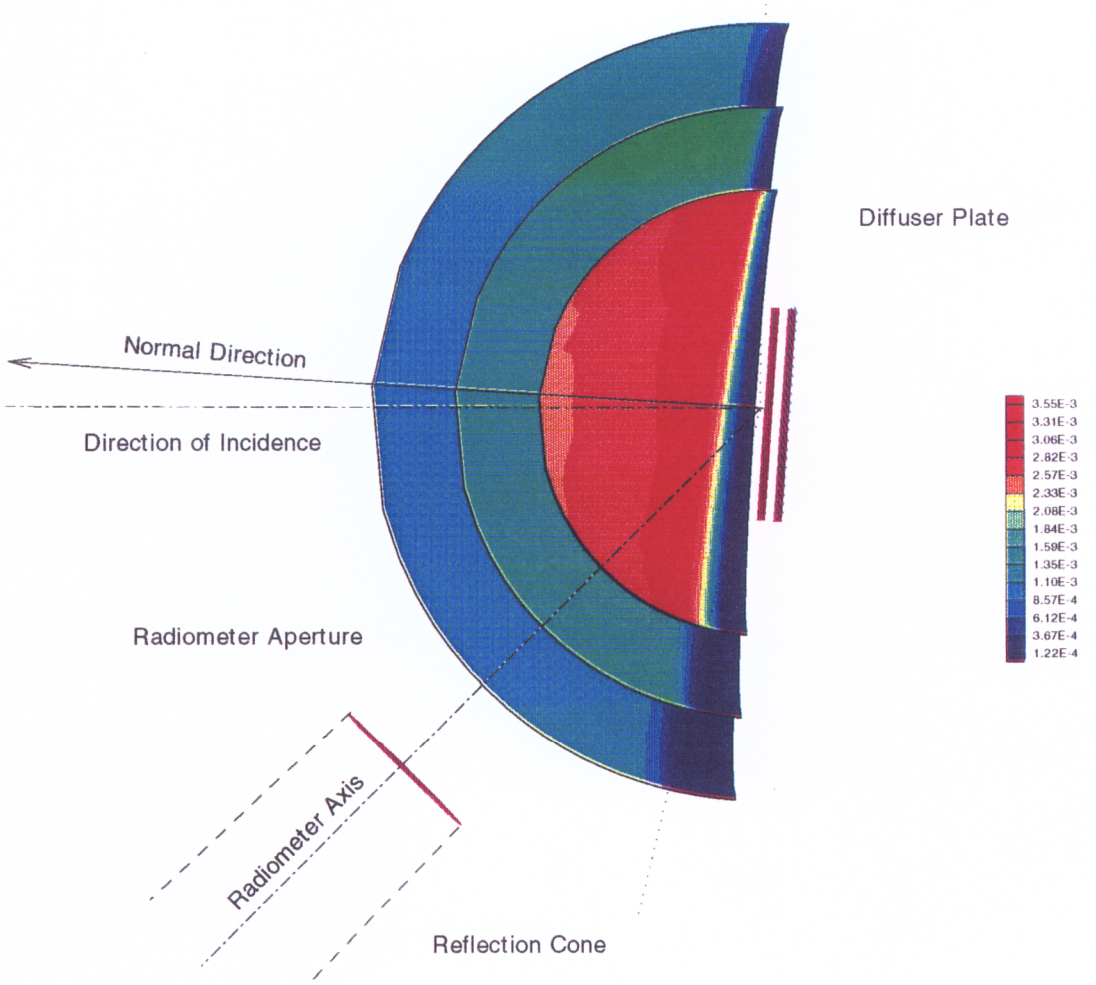


Fig. 26 Spatial Distribution of Radiation Reflected by the Diffuser Plate Oriented 4 deg Above the Baffle Axis.

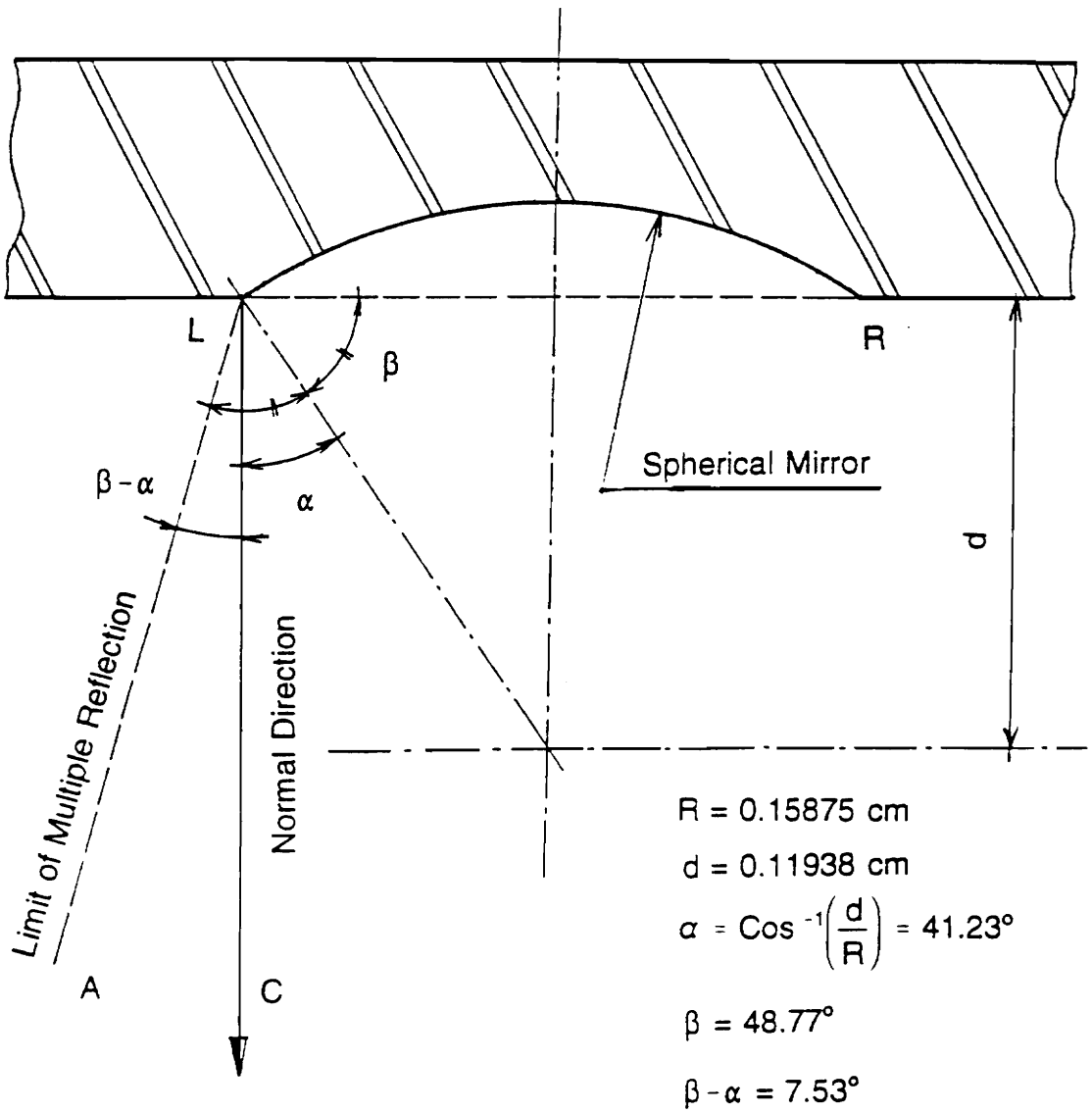


Fig. 27 Illustration of the Limit of Incidence Direction to Avoid Multiple Reflection.

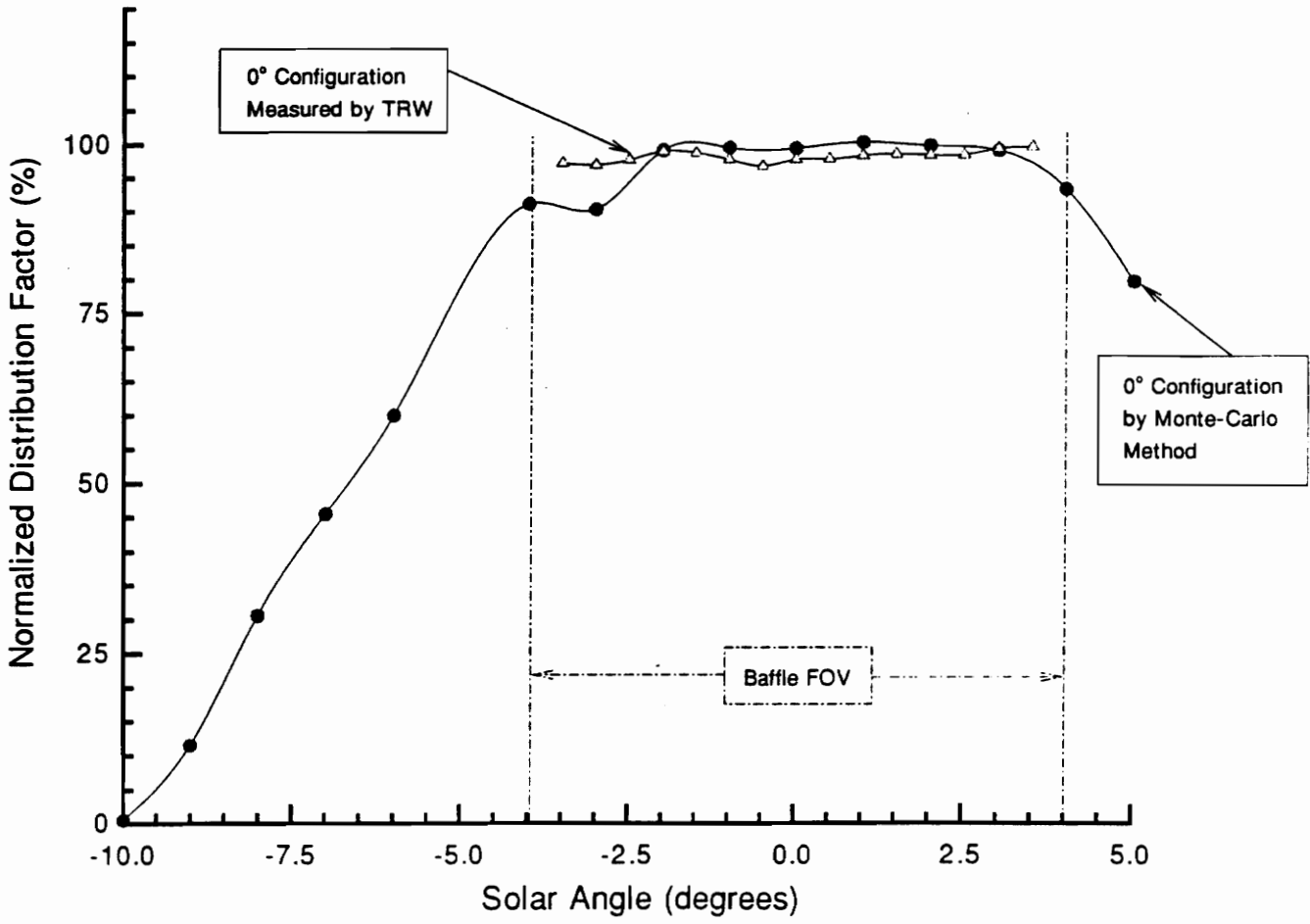


Fig. 28 Fraction of Radiation Reflected to the Radiometer Aperture as a Function of the Solar Angle, Defined by the CERES MAM Model and Experimented Data [26], for Diffuser Plate Oriented Along the Baffle Axis.

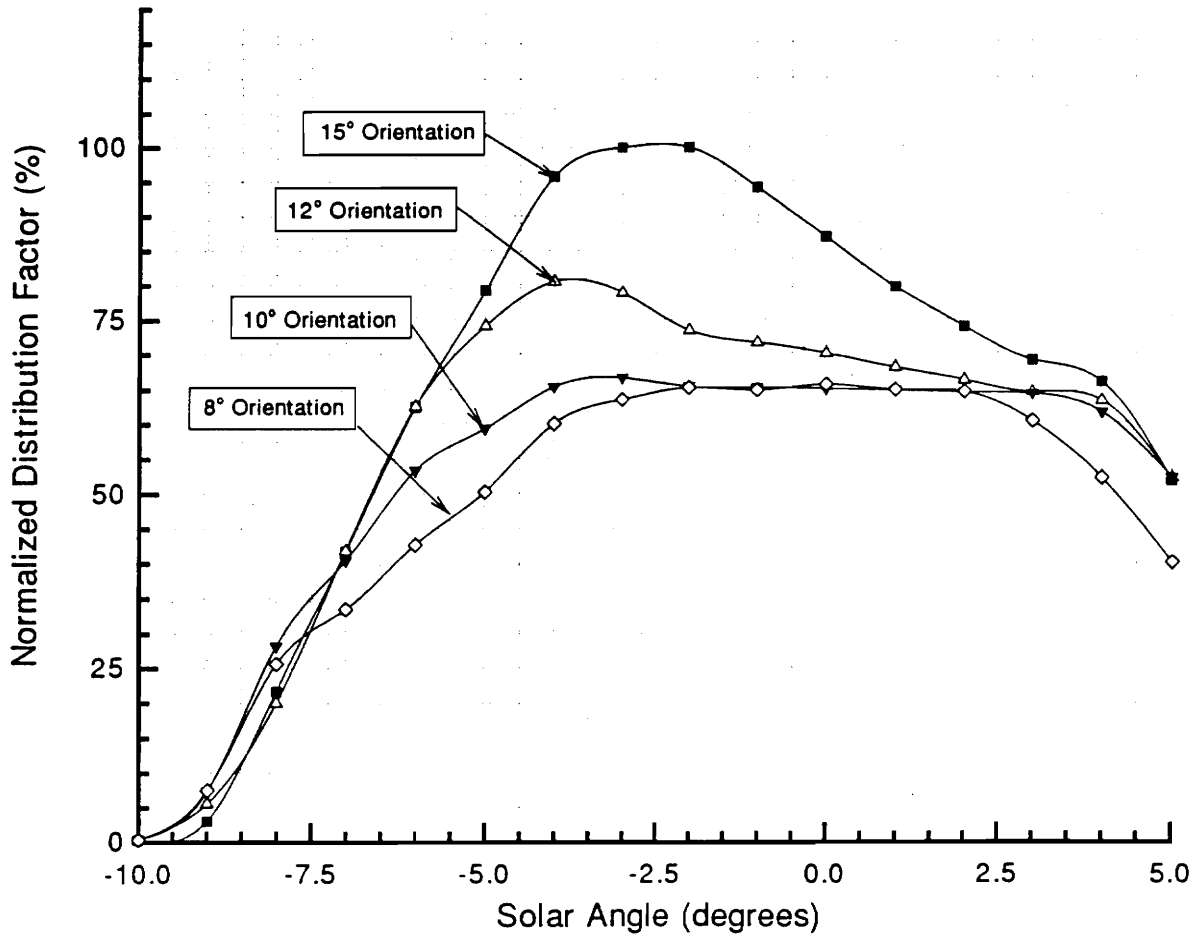


Fig. 29 Fraction of Radiation Reflected to the Radiometer Aperture as a Function of Solar Angle for the Diffuser Plate Orientation Varying from 15 deg to 8 deg Below the Baffle Axis.

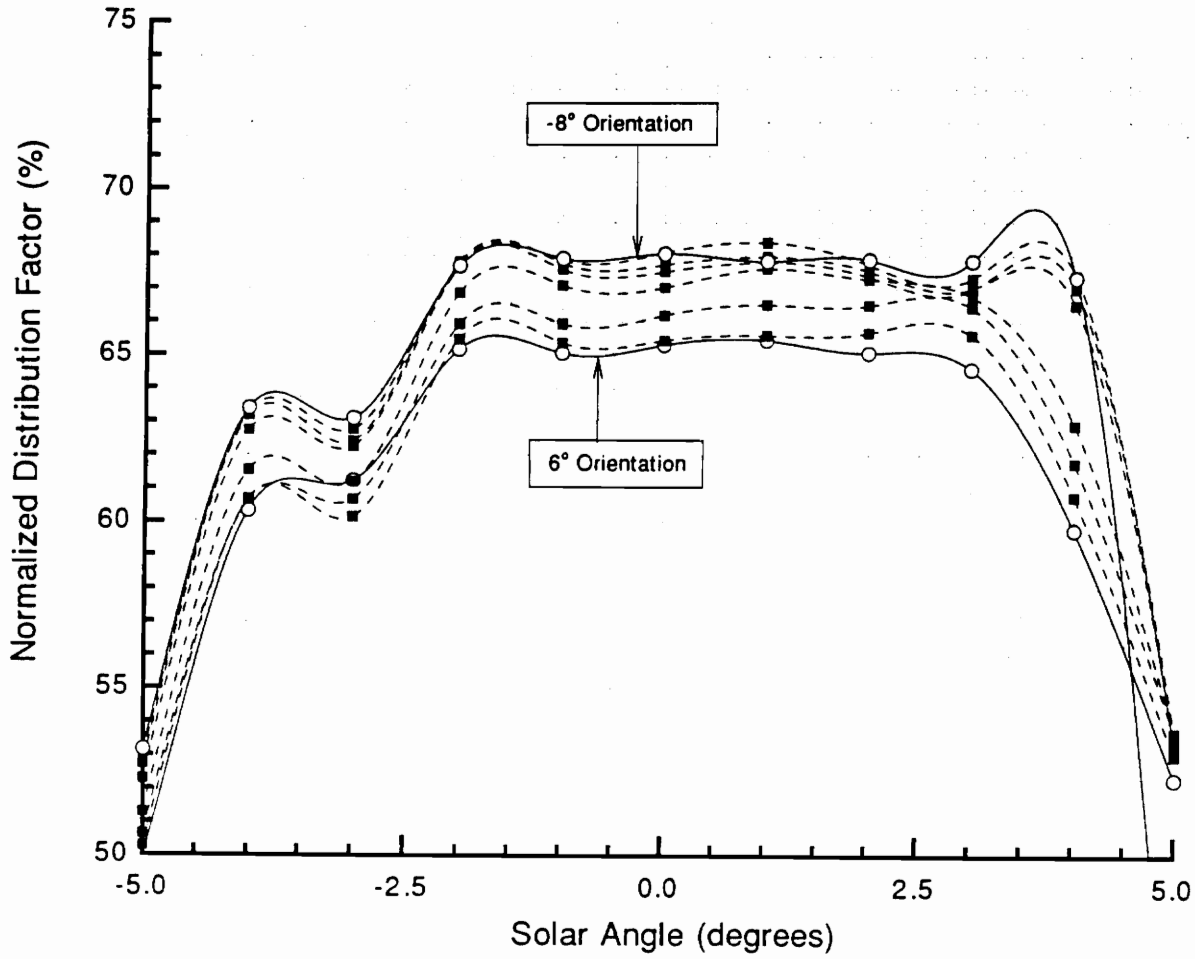


Fig. 30 Fraction of Radiation Reflected to the Radiometer Aperture as a Function of Solar Angle for the Diffuser Plate Orientation Varying from 6 deg Below to 8 deg Above the Baffle Axis.

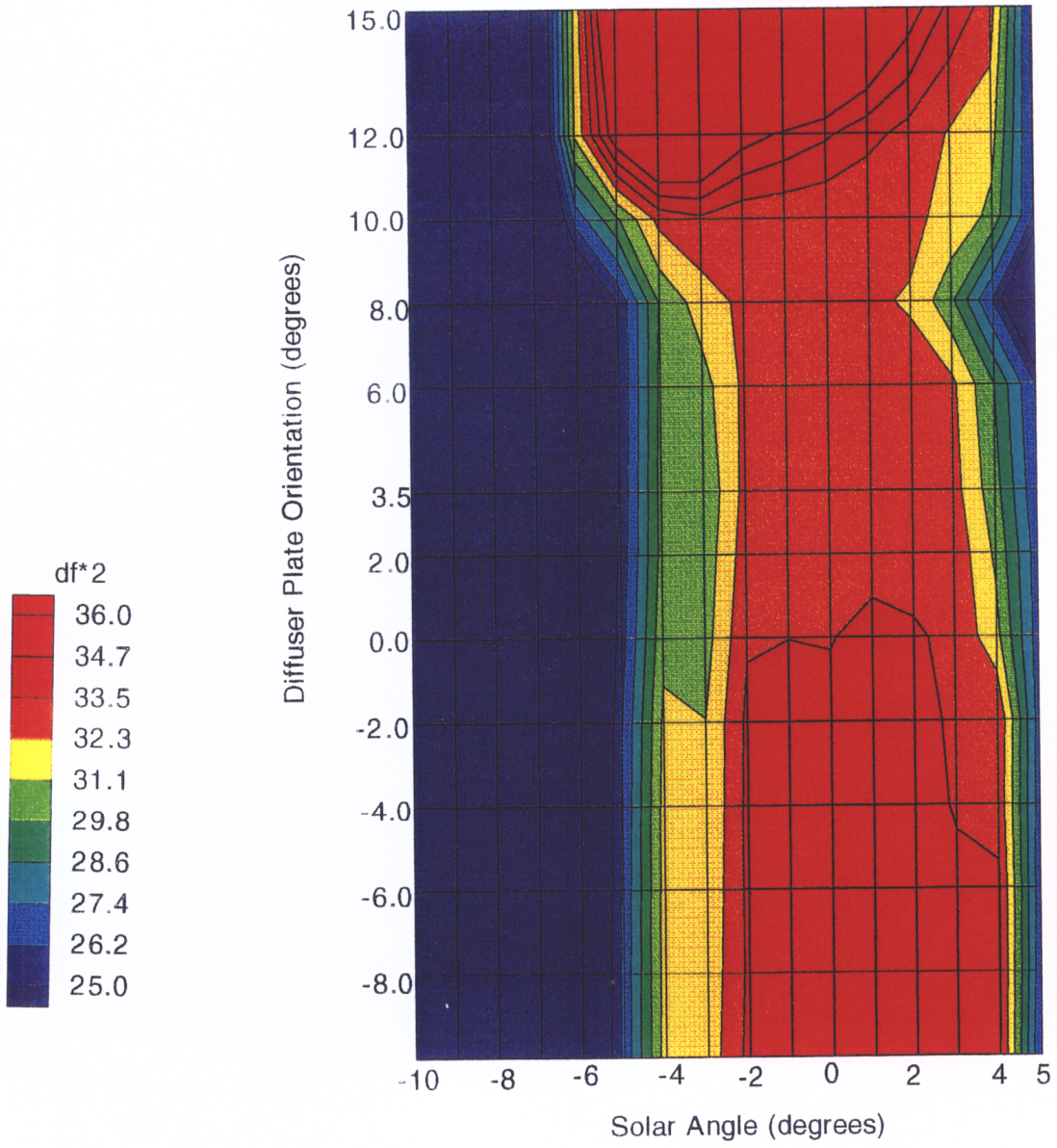


Fig. 31 Contour Plot of the Radiometer Aperture Distribution Factor as a Function of the Solar Angle and the Diffuser Plate Orientation.

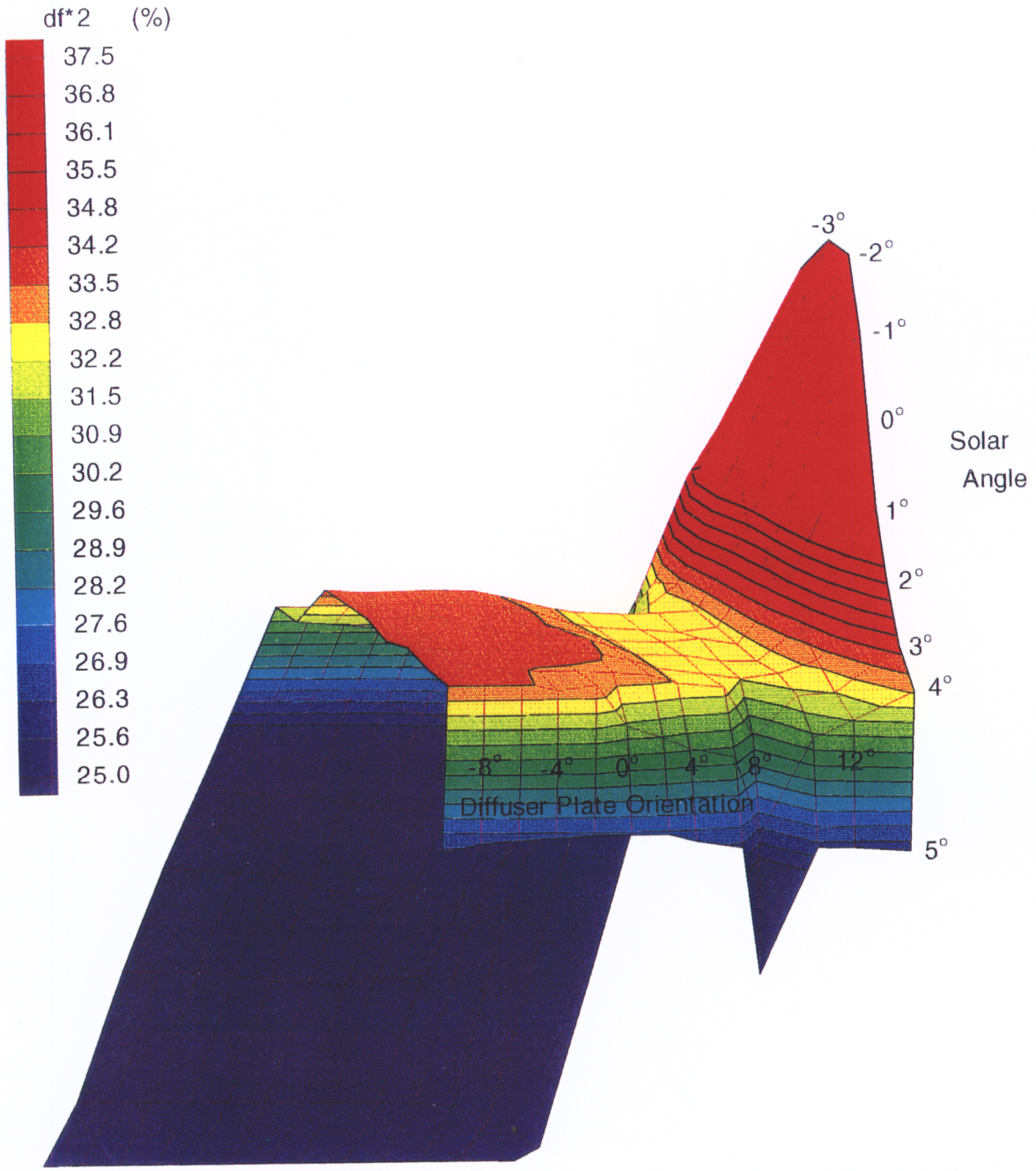
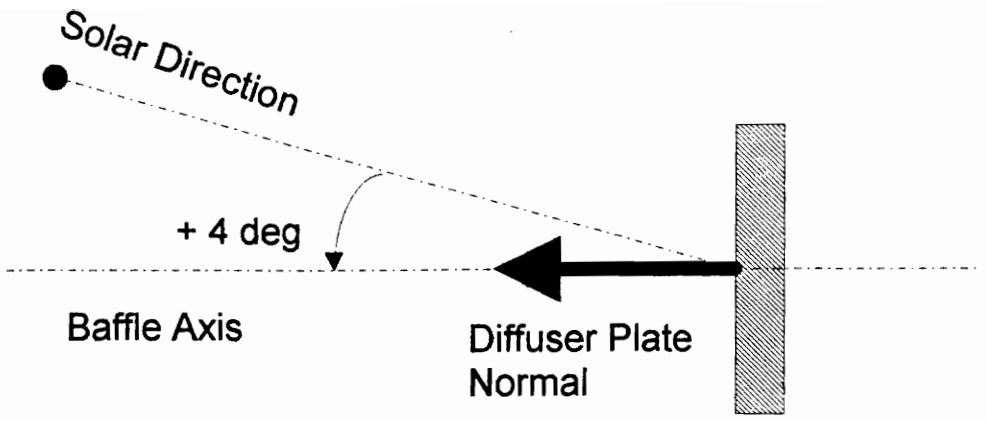
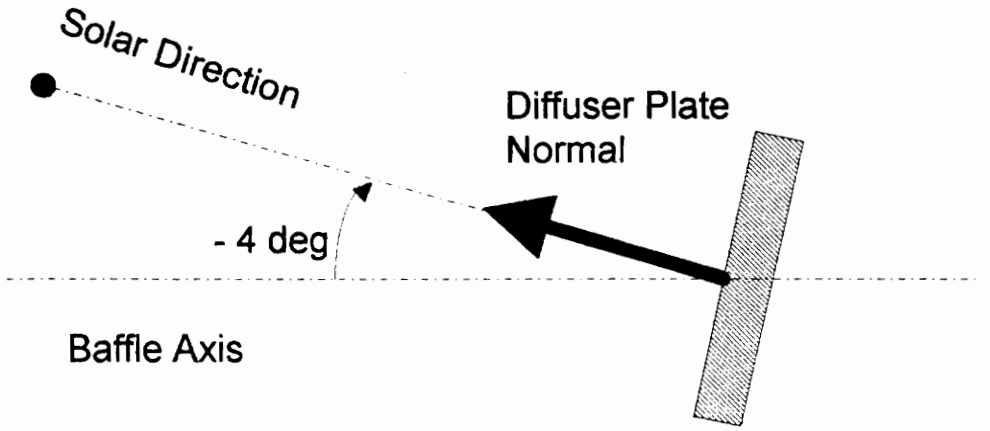


Fig. 32 Contour Plot Radiometer Aperture Distribution Factor as a Function of the Solar Angle and the Diffuser Plate Orientation. The Plot Is Observed Along the Solar-Angle-Axis.



Angle between the solar direction and the diffuser plate normal = + 4 deg



Angle between the solar direction and the diffuser plate normal = 0 deg

Fig. 33 Worst Case of Solar Angle.

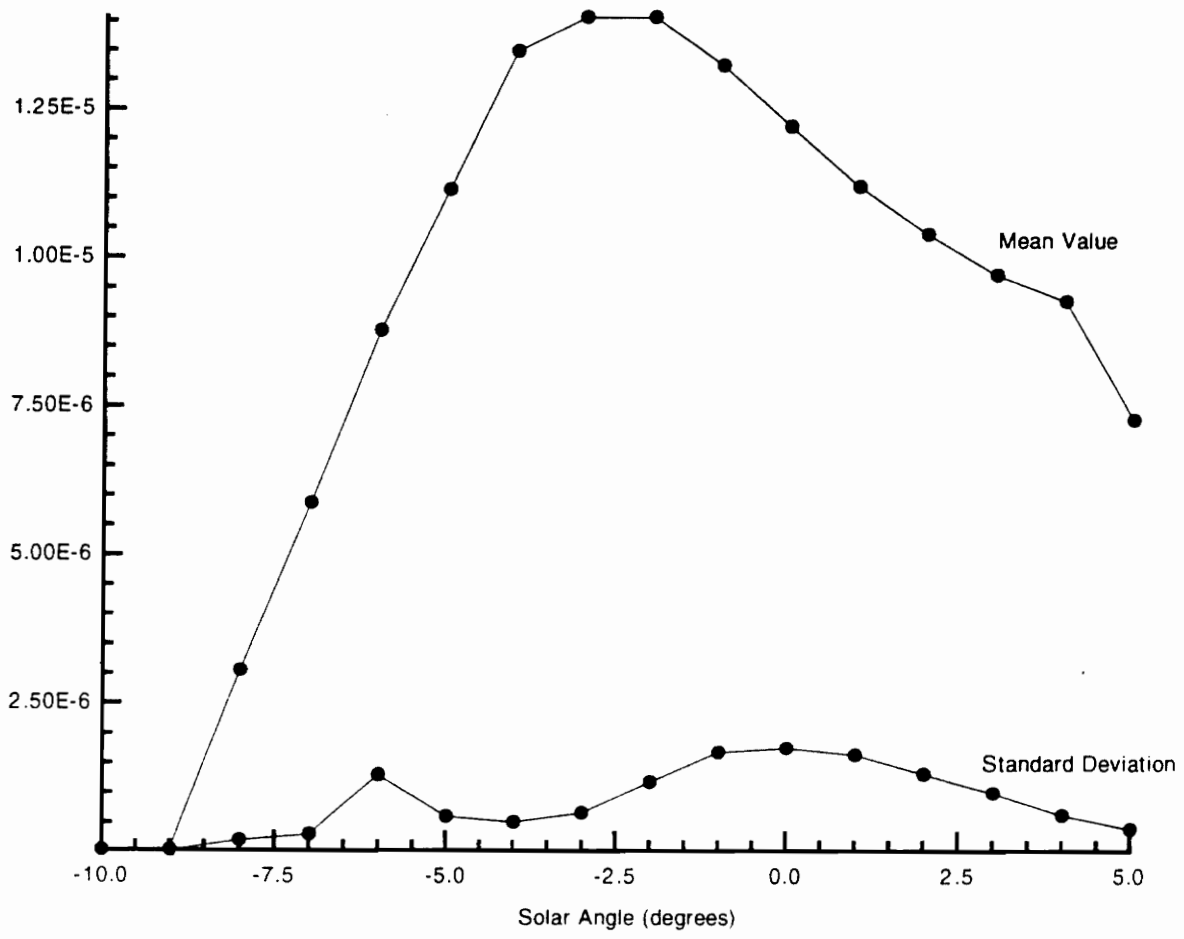


Fig. 34 Mean Value and Standard Deviation of the Distribution Factors of the Radiometer Aperture as a Function of the Solar Angle for the Diffuser Plate Oriented at 15 deg Below the Baffle Axis.

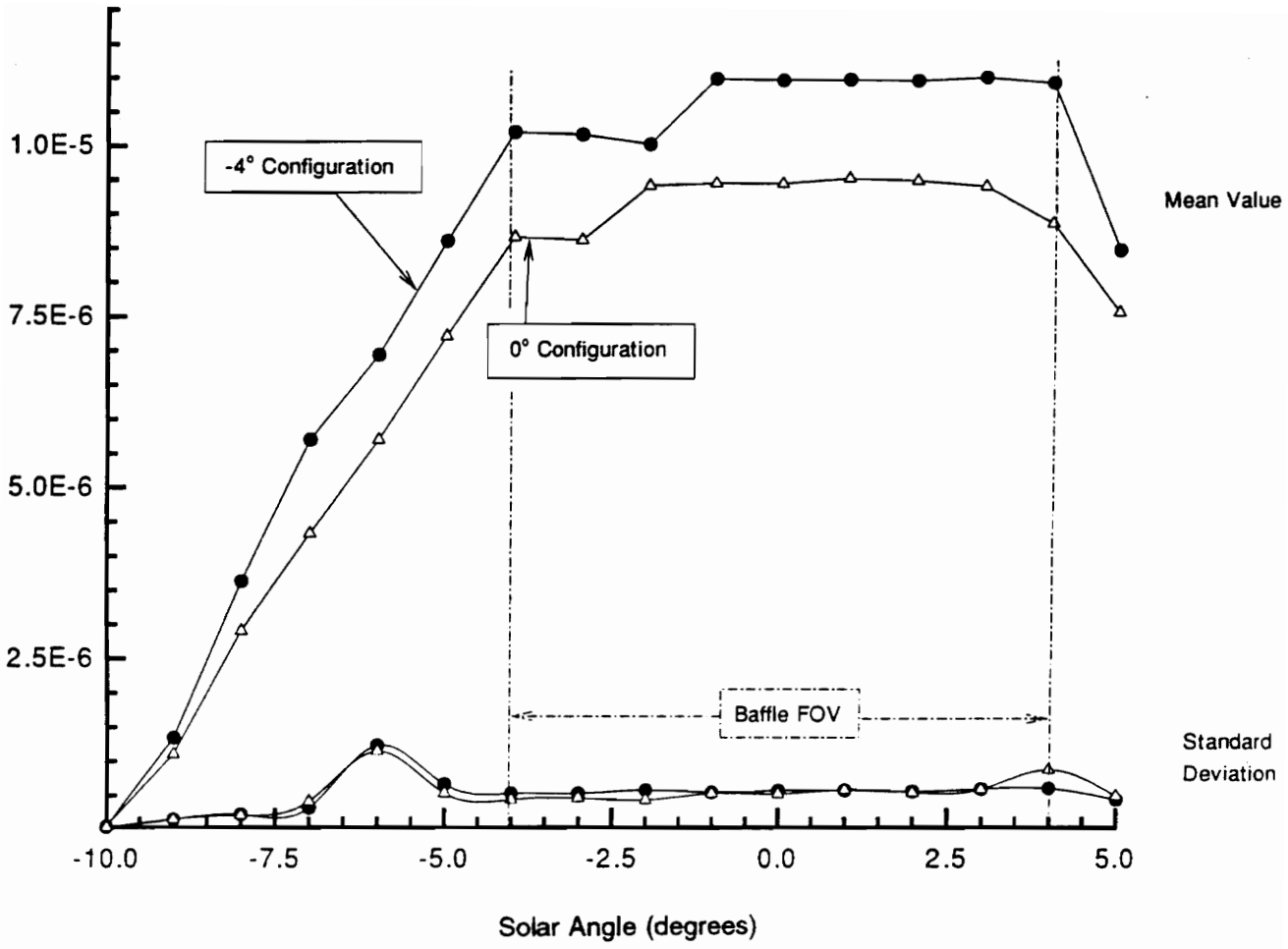


Fig. 35 Mean Value and Standard Deviation of the Distribution Factors of Radiometer Aperture as a Function of the Solar Angle for the Diffuser Plate Oriented at 0 and 4 deg Above the Baffle Axis.

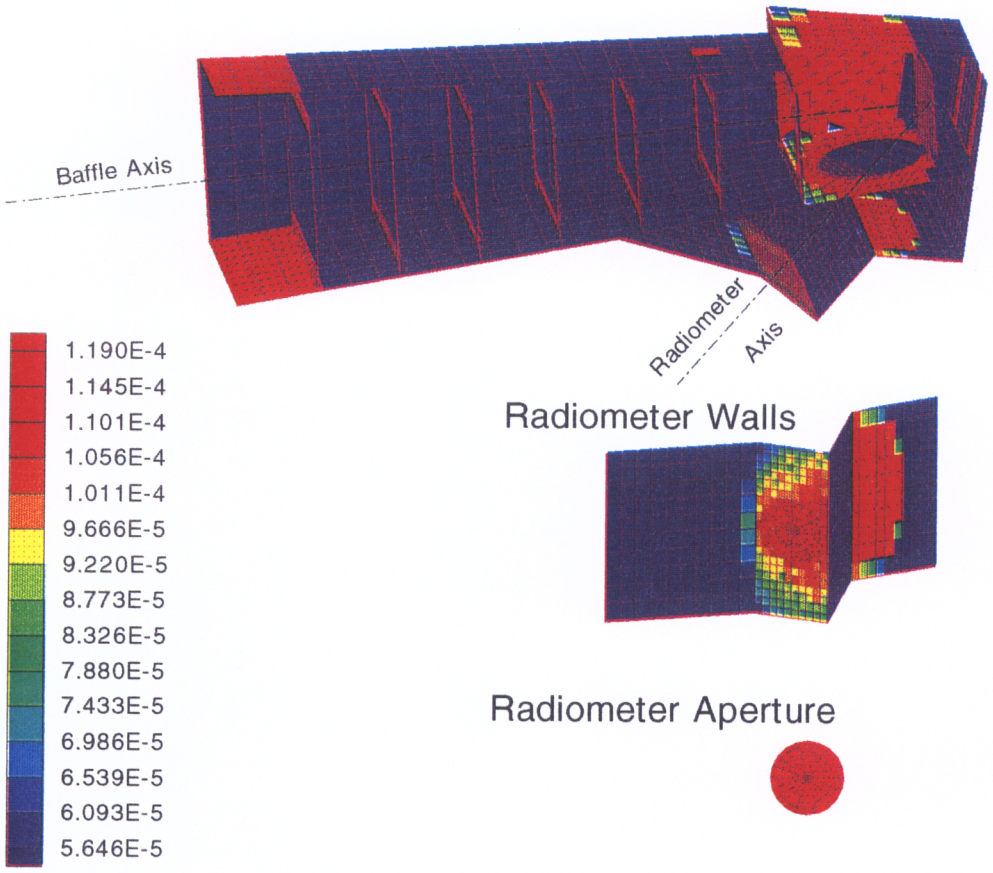


Fig. 36 Distribution of Solar Radiation on the MAM Enclosure for the Diffuser Plate Normal to the Baffle Axis and with the Solar Vector Along the Baffle Axis.

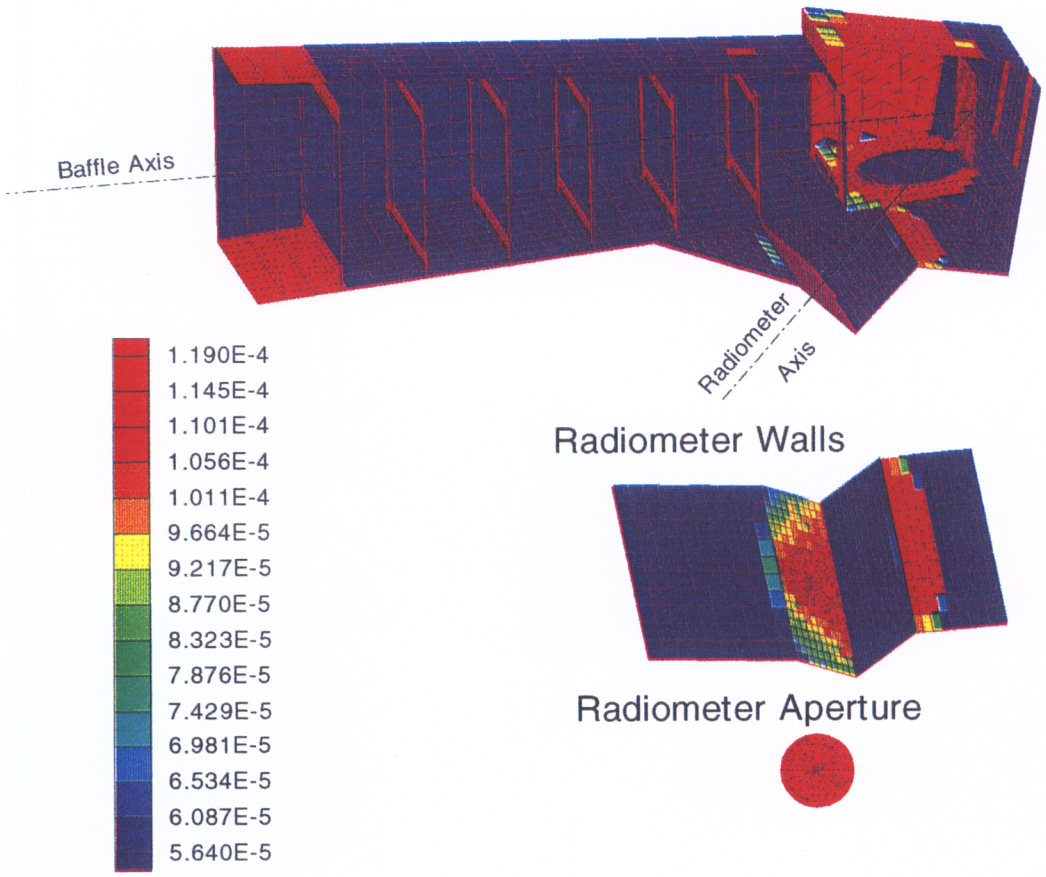


Fig. 37 Distribution of Solar Radiation on the MAM Enclosure for the Diffuser Plate Oriented 4 deg Above the Baffle Axis and with the Solar Vector Along the Baffle Axis.

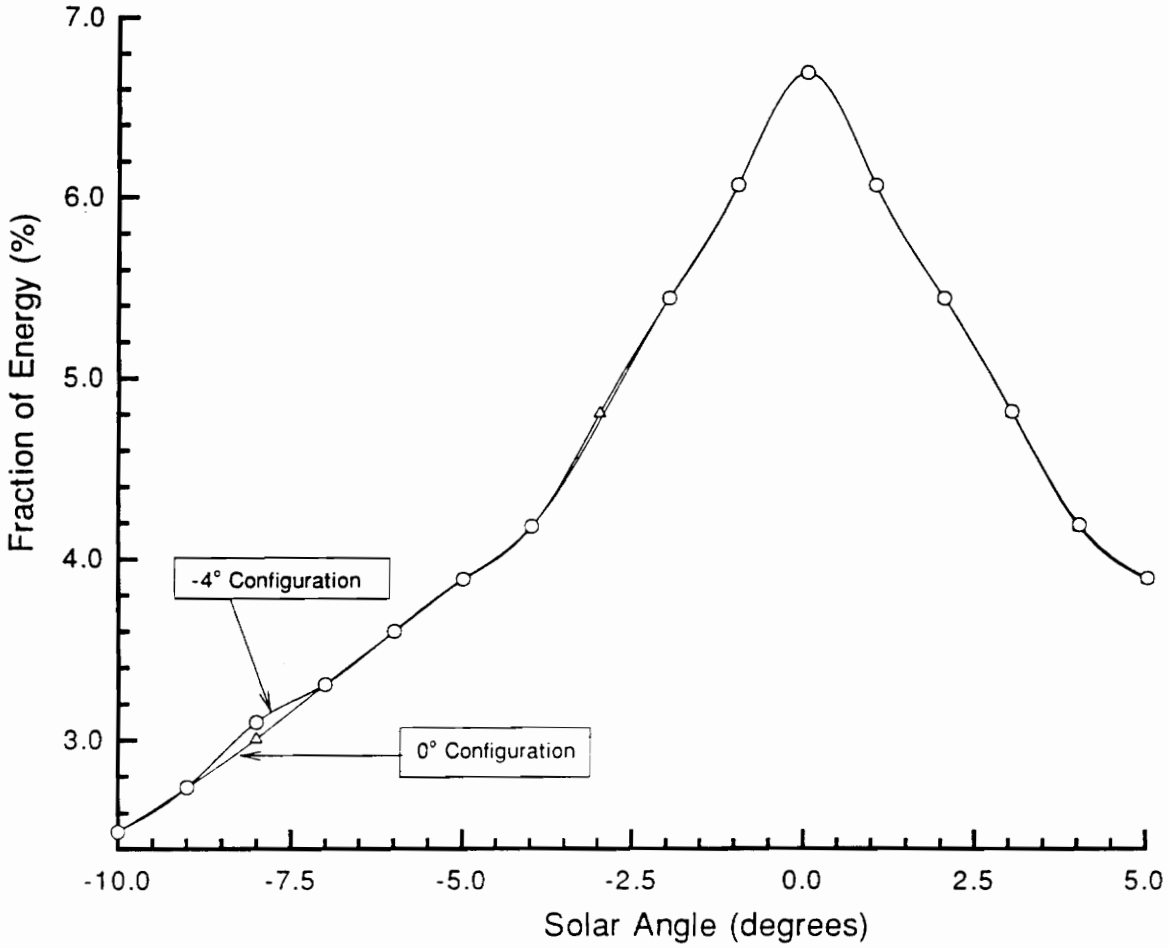


Fig. 38 Fraction of Solar Radiation Reflected Back Out of the Baffle Opening as a Function of the Solar Angle for the Diffuser Plate Oriented at 0 and 4 deg Above the Baffle Axis.

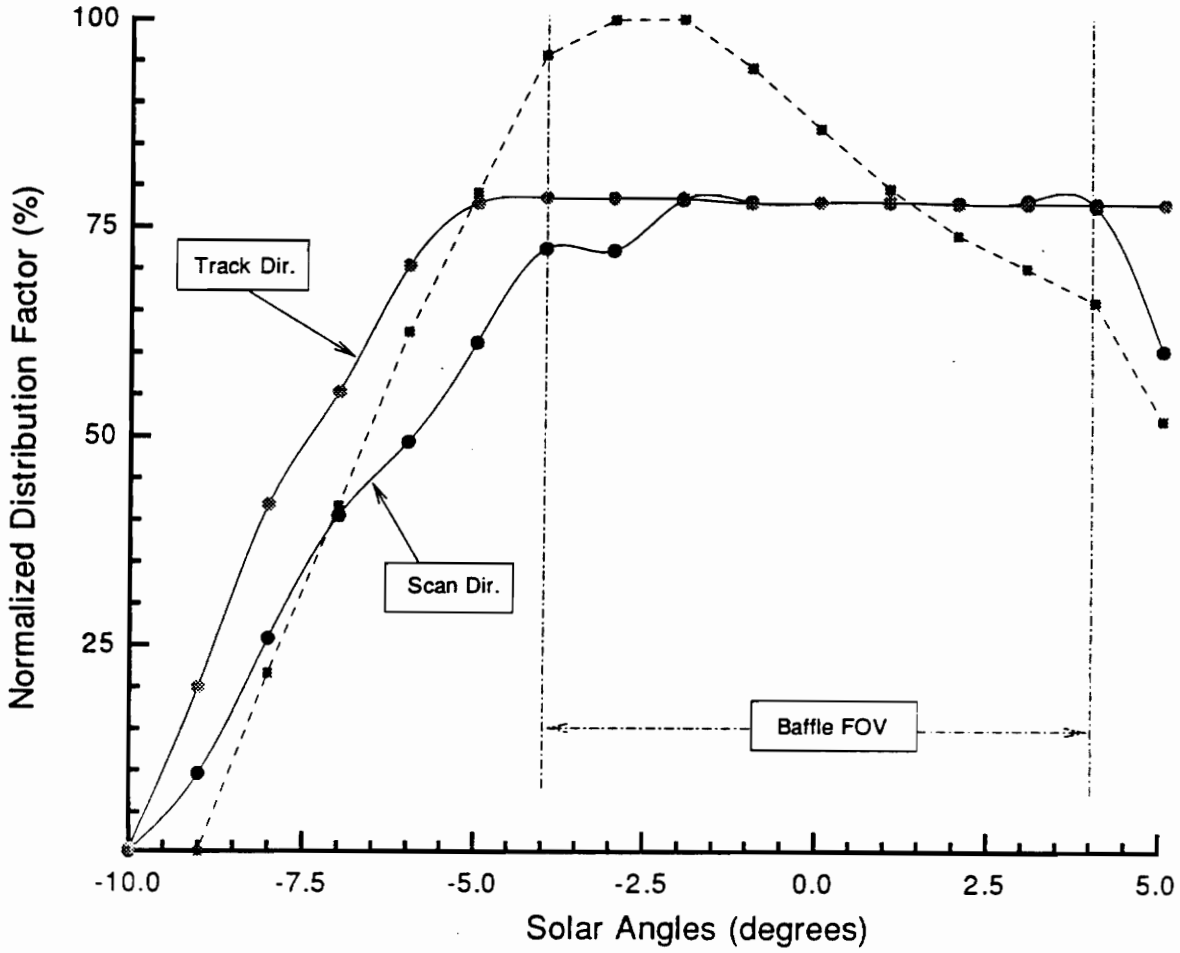


Fig. 39 Normalized Distribution Factor of the Radiometer Aperture as a Function of Solar Angles In Both the Vertical and Horizontal Planes for the Diffuser Plate Oriented at 4 deg Above the Baffle Axis and Translated 1 cm Forward.

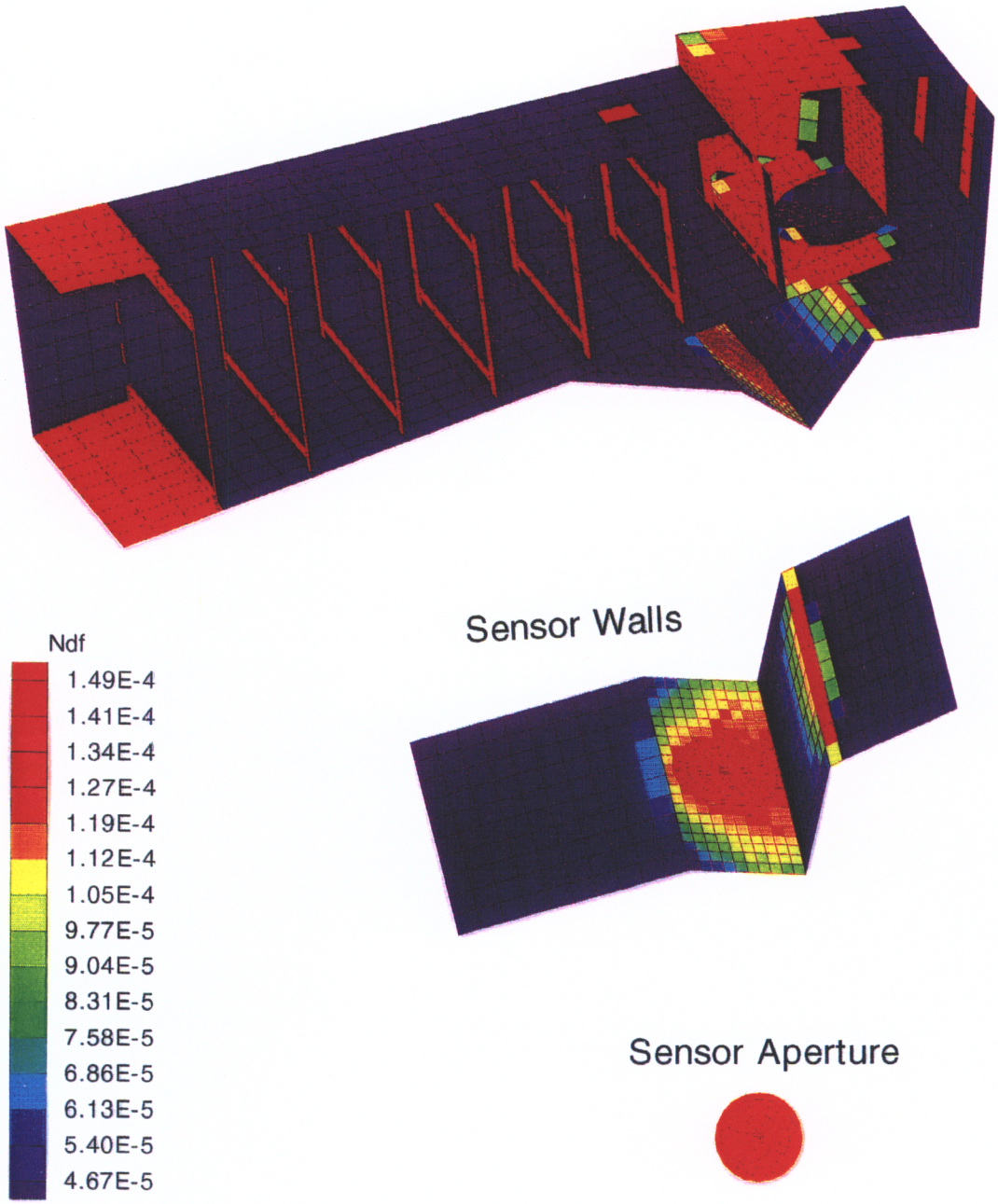


Fig. 40 Distribution of Solar Radiation Reflected on the MAM Enclosure for the Diffuser Plate Oriented 4 deg Above the Baffle Axis and Translated 1 cm Forward and for the Incident Solar Angle Along the Baffle Axis.

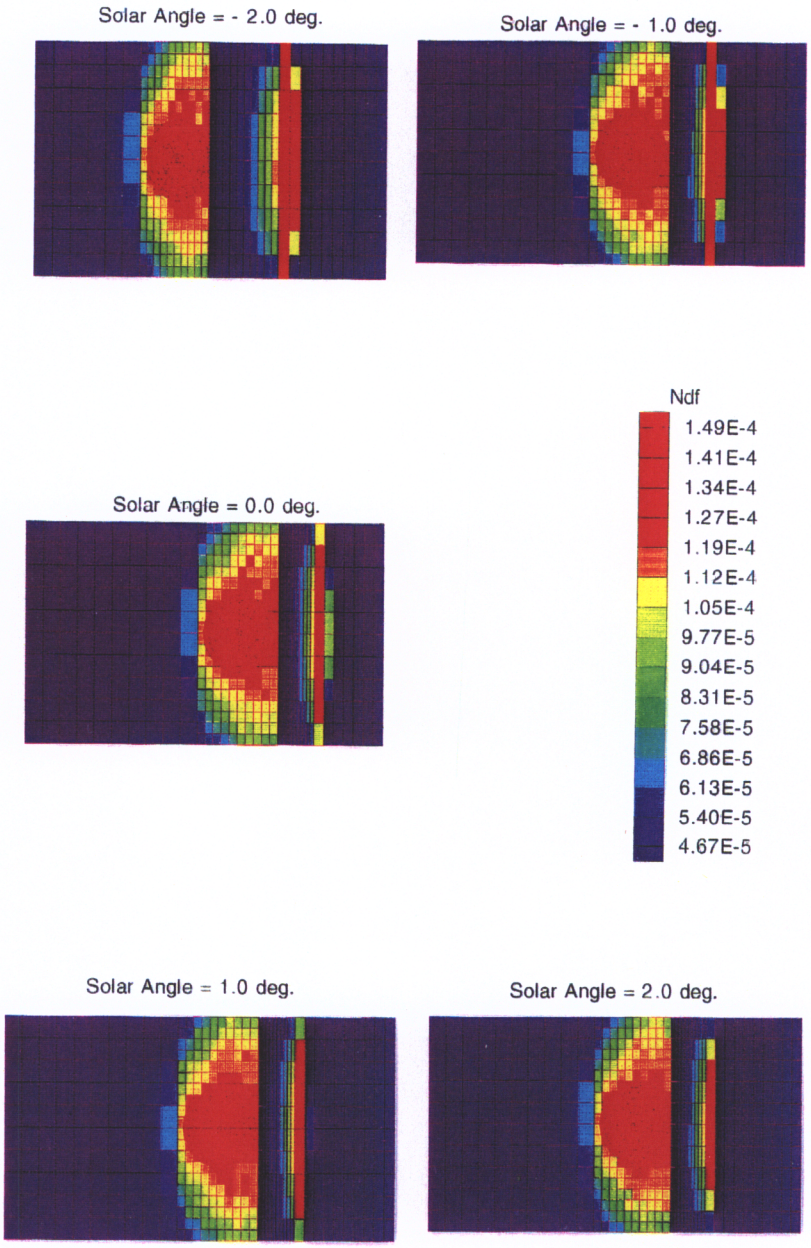


Fig. 41 Distribution of Solar Radiation Reflected onto the Radiometer Walls for the Diffuser Plate Oriented 4 deg Above the Baffle Axis and Translated 1 cm Forward and for Various Solar Angles in the Vertical Plane.

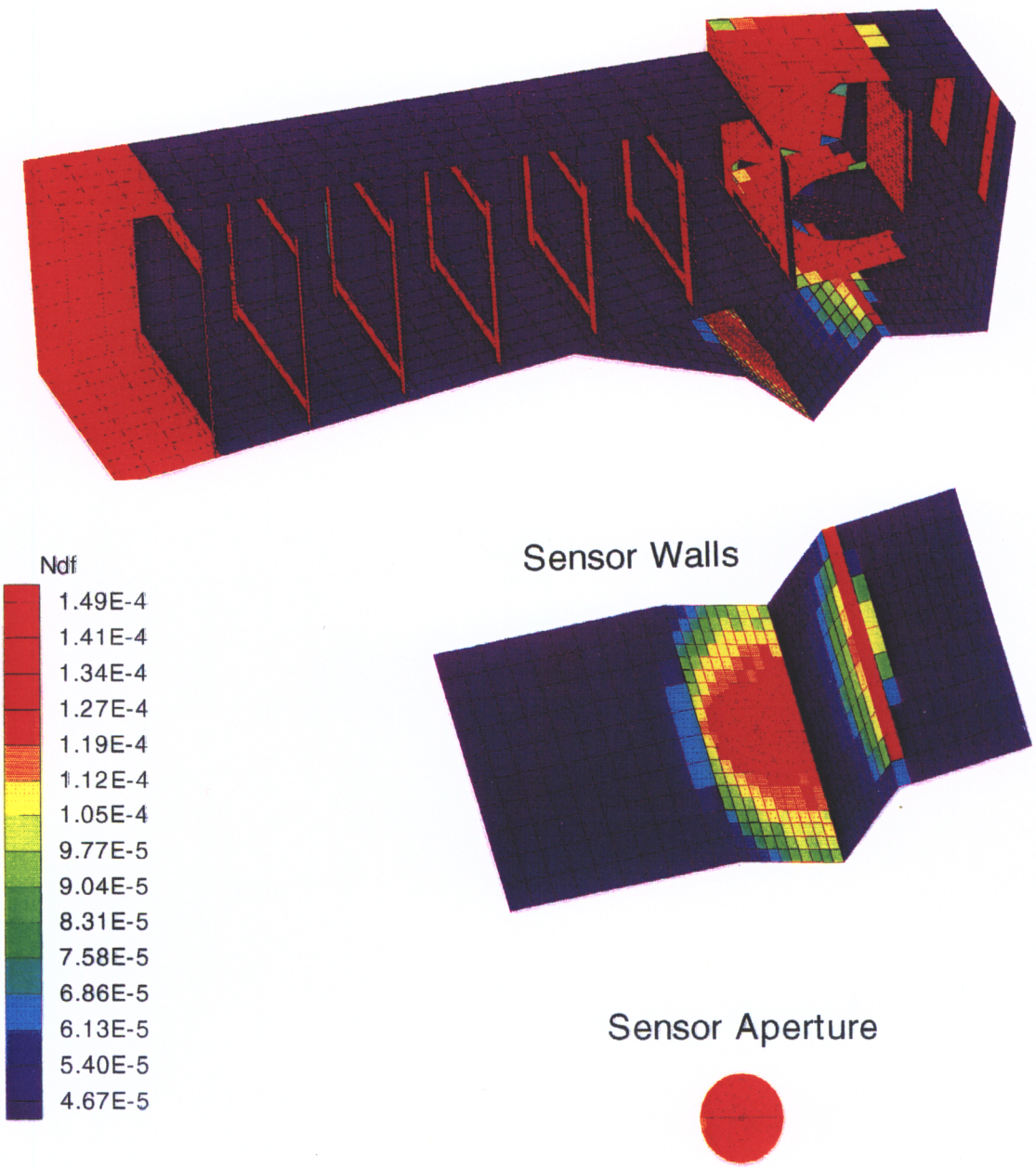


Fig. 42 **Distribution of Solar Radiation Reflected on the MAM Enclosure for the Diffuser Plate Oriented 4 deg Above the Baffle Axis and Translated 1 cm Forward and for a Solar Angle of -2 deg in the Horizontal Plane.**

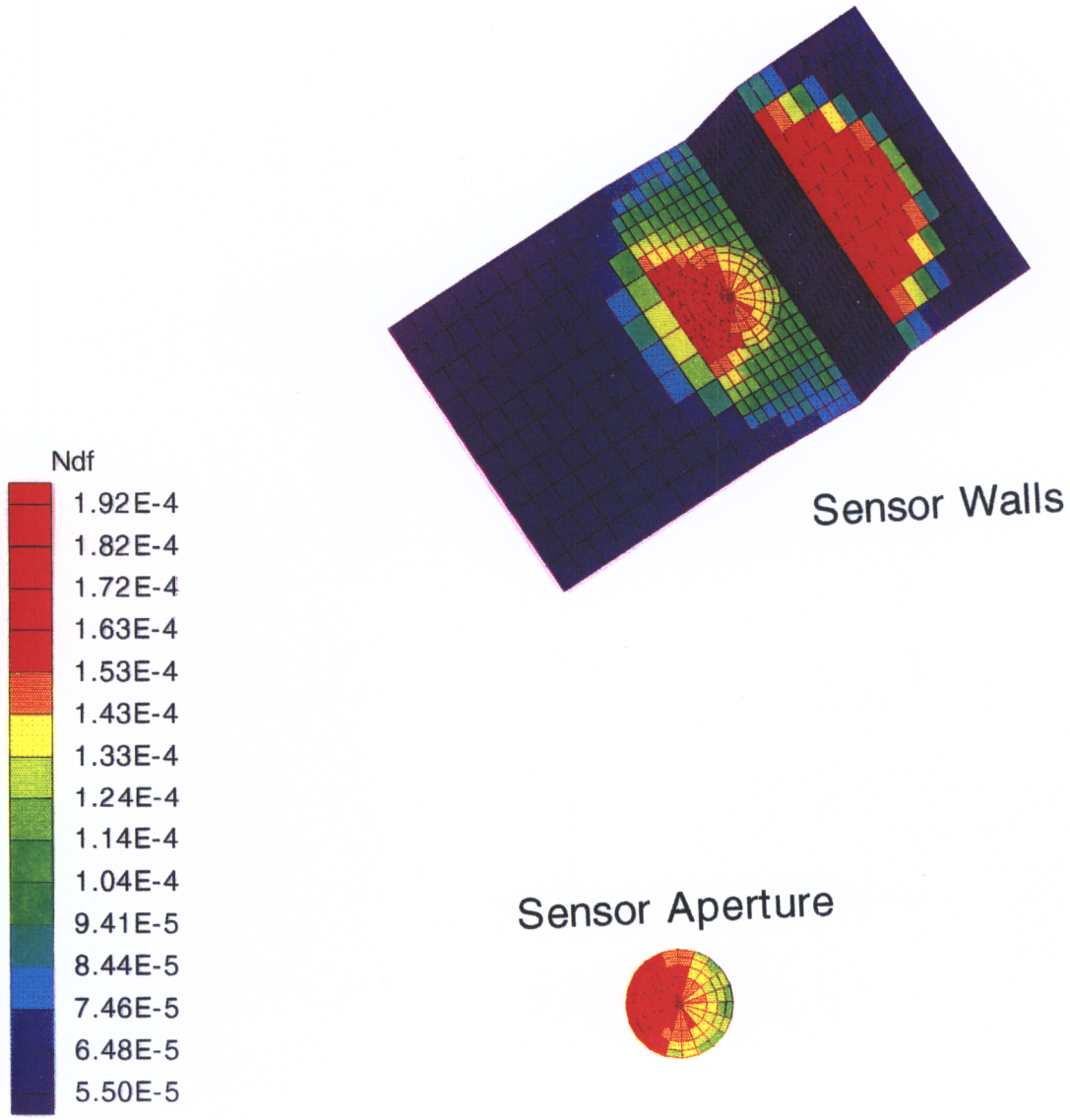


Fig. 43 Distribution of Solar Radiation Reflected onto the Radiometer Walls for the Diffuser Plate Oriented 15 deg Below the Baffle Axis and for a Solar Angle of -2 deg in the Horizontal Plane.

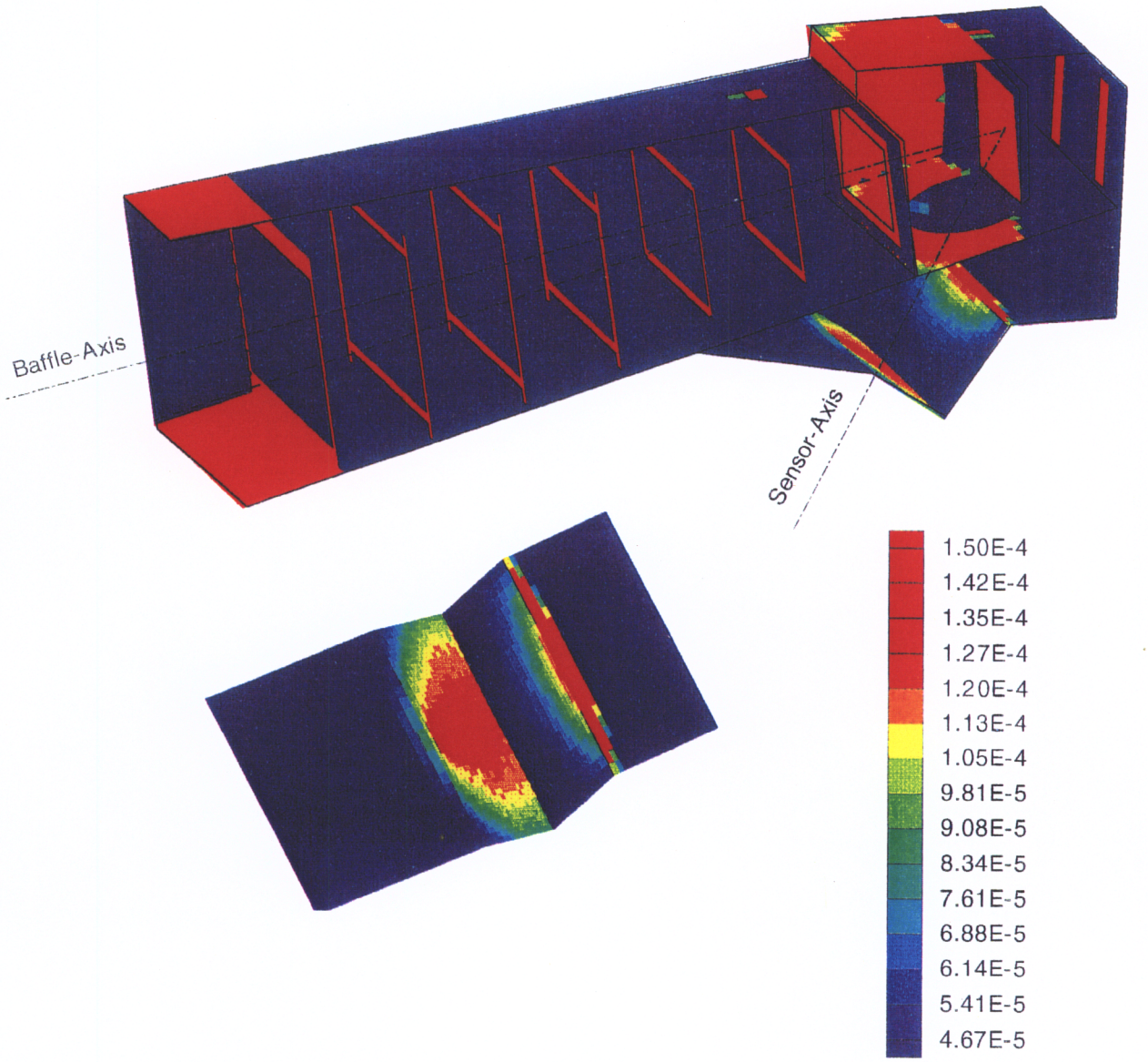


Fig. 44 Distribution of Solar Radiation Reflected on the MAM Enclosure for the Diffuser Plate Oriented 4 deg Above the Baffle Axis and Translated 1 cm Forward, Determined by a Monte-Carlo Ray-Trace Using 10^9 Energy Bundles. Incident Radiation is Along the Baffle Axis.

References

1. Hunt, G. E., R. Kandel, and A. T. Mecherikunnel, *A History of Presatellite Investigations of the Earth's Radiation Budget*, **Reviews of Geophysics**, Vol. 24, No. 2, May 1986, pp 351-356.

2. House, F. B., A. Gruber, G. E. Hunt, and A. T. Mecherikunnel, *History of Satellite Missions and Measurements of the Earth Radiation Budget (1957-1984)*, **Reviews of Geophysics**, Vol. 24, No. 2, May 1986, pp 357-377.

3. Barkstrom, B. R., and J. B. Hall Jr., *Earth Radiation Budget Experiment (ERBE): An Overview*, **Journal of Energy**, Vol. 6, No. 2, April 1982, pp 141-146.

4. Barkstrom, B. R., *The Earth Radiation Budget Experiment (ERBE)*, **Bulletins of the American Meteological Society**, Vol. 65, No. 11, 1984, pp 1170-1185.

5. Barkstrom, B. R., and G. L. Smith, *The Earth Radiation Budget Experiment: Science and Implementation*, **Reviews of Geophysics**, Vol. 24, No. 2, May 1986, pp 379-390.

6. Kopia, L. P., *Earth Radiation Budget Experiment Scanner Instrument*, **Reviews of Geophysics**, Vol. 24, No. 2, May 1986, pp 400-406.

7. Ramanathan, V., R. D. Cess, E. F. Harrison, P. Minnis, B. R. Barsktrom, E. Ahmad, and D. Hartmann, *Cloud-Radiative Forcing and Climate: Results from the Earth Radiation Budget Experiment*, **Science**, Vol. 243, 1889, pp 57-63.

8. Ramanathan, V., B. R. Barskstrom, and E. F. Harrison, *Climate and the Earth's Radiation Budget*, **Physics Today**, Vol. 42, No. 5, May 1989, pp 22-32.

9. Anon., InterNetwork, Inc., **EOS A Mission to Planet Earth**, The National Aeronautics and Space Administration, 1993.

10. Ghassem Asrar, and David Jon Dokken, ed., **1993 EOS Reference Handbook**, The National Aeronautics and Space Administration, 1993.

11. Barkstrom, B. R., *Earth Radiation Budget Measurements: Pre-ERBE, ERBE, and CERES*, **Proceeding of SPIE - The International Society for Optical Engineering**, Vol. 1299, 1990, pp 52-60.

12. Lee III, R. B., B. R. Barkstrom, L. P. Kopia, J. E. Cooper, P. J. Jarecke, and M. A. Folkman, *The Clouds and the Eath's Radiant Energy System (CERES) Experiment, First Earth Observing System (EOS) Instrument*, **Third Annual Space Dynamics Laboratory/Utah State University Symposium on Infrared Radiometric Sensor Calibration**, Utah State University, September 14-17, 1992.

13. Robert B. Lee III, *The Clouds and the Earth's Radiant Energy System (CERES) Experiment, Earth Obverving System (EOS) Instrument and Calibrations*, **SPIE's International Symposium**, Technical Conference Paper 1993-07, Orlando, Florida, April 12-16, 1993.

14. Carman, S., **CERES Preliminary Design Review**, TRW Space & Technology, Los Angeles, CA, June 23-25, 1992.

15. Rasnic, R. L., **A Thermal and Kinematic Model of a Thin-Wall Spherical Shell Satellite**, M. S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1975).

16. Fanney II, A. H., **Experimental Study of the LZEEBE Measurement System under Simulated Flight Conditions**, M. S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1975).

17. Passwaters III, J. O., **Detailed Thermal Analysis of a Thin-Shell, Spherical Radiometer in Earth Orbit**, M. S. Thesis, Mechanical Engineering Department, Virginia Polytechnic and State University, Blacksburg, VA (1976).

18. Eskin, L. D., **Application of the Monte-Carlo Method to the Transient Thermal Modeling of a Diffuse-Specular Radiometer Cavity**, M. S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1981).

19. Gardiner, B. D., **An Analytical Study of Dynamic Response and Nonequivalence of an Absolute Active Cavity Radiometer Operating at Cryogenic Temperatures**, M. S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1986).

20. Tira, N. E., **A Study of the Thermal and Optical Characteristics of Radiometric Channels for Earth Radiation Budget Applications**, Ph. D. Dissertation, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1991).

21. Priestley, K., **An End-to-End Model of the Nonscanning Radiometric Channels of the Earth Radiation Budget Experiment**, M. S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1993).

22. Haefflin, M., **A Numerical Study of the Equivalence in Scanning Thermistor Bolometer Radiometers for Earth Radiation Budget Apllications**, M. S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1993).

23. Bonjiovi II, R. P., **A Parametric Study of the Radiative and Optical Characteristics of a Scanning Radiometer for Earth Radiation Budget Apllications Using the Monte-Carlo Method**, M. S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1993).

24. Meekins, J. L., **Optical Analysis of the ERBE Scanning Thermistor Bolometer Radiometer Using the Monte-Carlo Method**, M. S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1990).

25. Lee III, R. B., *Characterizations of the Mirror Attenuator Mosaic: Solar Diffuser Plate*, **Applied Optics**, Vol. 31, No. 31, November 1992, pp 6643-6652.

26. Folkman, M. A., P. J. Jarecke, T. R. Hedman, J. S. Yun, and R. B. Lee III, *Design of a Solar Diffuser for On-Orbit Calibration of the Clouds and the Earth's Radiant Energy System (CERES) Instruments,* **SPIE's International Symposium**, Technical Conference Paper, Vol. 1939, Orlando, Florida, April 12-16, 1993, pp 72-81.26.

27. Eckert, E. R. G., and Sparrow, E. M., *Radiative Heat Exchange Between Surfaces with Specular Reflections*, **International Journal of Heat Transfer**, Vol. 3, 1961, pp 42-54.

28. Sparrow, E. M., E. R. G. Eckert, and V. K. Jonsson, *An Enclosure Theory for Radiative Exchange Between Specularly and Diffusely Reflecting Surfaces*, **Journal of Heat Transfer**, Vol. 84, November 1962, pp 294-300.

29. Mahan, J. R., and L. D. Eskin, *The Radiation Distribution Factor - Its Calculation Using Monte Carlo and Example of its Application*, **First UK National Heat Transfer Conference**, July 4-6, 1984, Leeds, Yorkshire, England, pp 1001-1012.

30. Siegel, R., and J. R. Howell, **Thermal Radiation Heat Transfer**, Third Edition, Hemisphere Publishing Corporation, Washington, 1992.

Appendix A.

Program

CERESMAM

```
**********************************************************************************************************
*       Thermal Model of Radiometric Channel Solar Calibration for the Clouds and the Earth's
*              Radiant Energy System (CERES) Using the Monte-Carlo Method
**********************************************************************************************************
*
*                                File CERESMAM.F  (May 11, 1994)
*
*   This computer code is a thermal model for the solar calibration facility of the Clouds and the Earth's Radiant Energy
*   System (CERES). The program defines the coordinates of the points forming the geometry  of the Mirror Attenuator
*   Mosaic (MAM) from various parameters and subdivides the surfaces of the enclosure into area elements, then emits
*   energy bundles in random direction and at random location  on  the baffle aperture. The MONTE CARLO ray-trace
*   technique is applied  to determine the points where energy bundles intersect the area  elements making up the
*   MAM  enclosure. The  coordinates  of the corners, and the number of rays incident on each area element are
*   written to files 'File_NF.cer', that are then used by TECPLOT to represent the distribution of solar radiation
*   reflected  on the  enclosure, and on the sensor aperture by the MAM.
**********************************************************************************************************

      Program  CERESMAM
      Implicit double precision (A-H, O-Z)

      Character*8  Title
      Character*72 Line, ligne, line1
      Character*20 hname,name1,name2,name3,name4,name5,name6,NFtmp

c     Declare the dimension for points forming the geometry of the MAM:

c     In the local coordinate system:

      Dimension p1p(3),p2p(3),p3p(3),p4p(3),p5p(3),p6p(3),p7p(3),p8p(3)
     *     ,p9p(3),p10p(3),pap(3),pbp(3),pcp(3),p25p(3),p29p(3),
     *      p30p(3),p31p(3),p42p(3)

c     In the global coordinate system:

      Dimension p1(3),p2(3),p3(3),p4(3),p5(3),p6(3),p7(3),p8(3),p9(3)
     *     ,p10(3),p11(3),p12(3),p13(3),p14(3),p15(3),p16(3),p17(3)
     *     ,p18(3),p19(3),p20(3),p21(3),p22(3),p23(3),p24(3),p25(3)
     *     ,p26(3),p27(3),p28(3),p29(3),p30(3),p31(3),p32(3),p33(3)
     *     ,p34(3),p35(3),p36(3),p37(3),p38(3),p39(3),p40(3),p41(3)
     *     ,p42(3),p43(3),p44(3),p45(3),p46(3),p47(3),p48(3),p49(3)
      Dimension p50(3),p51(3),p52(3),p53(3),p54(3),p55(3),p56(3),p57(3)
     *     ,p58(3),p59(3),p60(3),p61(3),p62(3),p63(3),p64(3),p65(3)
     *     ,p66(3),p67(3),p68(3),p69(3),p70(3),p71(3),p72(3),p73(3)
     *     ,p74(3),p75(3),p76(3),p77(3),p78(3),p79(3),p80(3),p81(3)
     *     ,p82(3),p83(3),p84(3),p85(3),p86(3),p87(3),p88(3),p89(3)
     *     ,p90(3),p91(3),p92(3),p93(3),p94(3),p95(3),p96(3),p97(3)
      Dimension p98(3),p99(3),p100(3),p101(3),p102(3),p104(3),p105(3)
     *     ,p106(3),p107(3),p108(3),p109(3),p110(3)

c     Declare the dimension for normal unit vectors of surfaces:

      Dimension dc1(3),dc2(3),dc3(3),dc4(3),dc5(3),dc6(3),dc7(3),dc8(3)
     *     ,dc9(3),dc10(3),dc11(3),dc12(3),dc13(3),dc14(3),dc15(3)
```

```
     *      ,dc16(3),dc17(3),dc18(3),dc19(3),dc20(3),dc21(3),dc22(3)
     *      ,dc23(3),dc24(3),dc25(3),dc26(3),dc27(3),dc28(3),dc29(3)
     *      ,dc30(3),dc31(3),dc32(3),dc33(3),dc34(3),dc35(3),dc36(3)
     *      ,dc37(3)

c (surfaces forming the reflectors on the baffle)
      Dimension dc38(3),dc39(3),dc40(3),dc41(3),dc42(3),dc43(3),dc44(3)
     *      ,dc45(3),dc46(3),dc47(3),dc48(3),dc49(3),dc50(3),dc51(3)

c Normal unit vectors in matrix form:

      Dimension dcm(51,3)

c Group of surfaces:

      Dimension Ngr(51)

c Characteristics of surfaces:
c
c - Surface area:
      Dimension surare(51), ssa(38:51,4)
c
c - Rectangular surfaces perpendicular to the x,z-plane:
      Dimension cXmin(51), cYmin(51),
     *      cXmax(51), cYmax(51), iflag(51)
c
c - Angle between the x'-direction and the x-direction:

      Dimension omega(51)

c - Recflectors on the baffle:
      Dimension rXmin(38:51), rXmax(38:51),
     *      rYmin(38:51), rYmax(38:51), etay(7)

c Maximum number of area elements per surface:

      Dimension Idir(51), Jdir(51)

c Surface properties:

      Dimension Emis(51), RhoDif(51), RhoSpe(51), RhoRef(51)

c Number of rays absorbed by surface:

      Dimension N(51,800), n29(150)
c     N(ns,nsa) where ns : Surface number, nsa : area element number

c Distribution factor Dij:

      Dimension Dfac(51,800)

c Coordinates of the point of intersection :
```

```
      Dimension Xcoor(51), Ycoor(51), Zcoor(51)

c Surfaces of intersection and the distance from the site of emission
c to the point of incidence :

      Dimension Nsurf(51), Dis(51)

c Radii for sensor aperture presentation, and shields :

      Dimension rsap(6), rsh(6)

c Coordinatres of the centers of the five possible spherical mirrors
c in a cell :

      Dimension xc(5),yc(5),xcin(5,5,9),ycin(5,5,9),xcto(2,9),ycto(2,9),
     *          xcbo(2,9),ycbo(2,9),xcle(2,9),ycle(2,9),
     *          xcri(2,9),ycri(2,9)

c Number of rays emitted from a sub-area of the reflector:
      Dimension Nresar(4)

c Logical variables :

      Logical inside, tops, front, face, hit, incirc, v, aper, ok

c List of surfaces that a given surface cannot see:

      Logical see(51,22)

c Put frequently used variables in blocks:

      Common/block1/ Ninter, Nsurf
      Common/block2/ dmX, dmY, dbX, dbY, dsX, dsY, rd, e1, e2, e3, e4
      Common/block3/ Dis, Xcoor, Ycoor, Zcoor
      Common/block4/ hit
      Common/block5/ Xmin, Xmax, Ymin, Ymax
      Common/block7/ Pi, x, y, z, dl, dm, dn, x2, y2, z2, dl2, dm2, dn2
      Common/ bl9  / N, Dfac, Ntre
      Common/ bl11 / cXmin, cXmax, cYmin, cYmax
      Common/ bl12 / rXmin, rXmax, rYmin, rYmax
      Common/ bl13 / xcs, ycs, zcs
      Common/ bl15 / rs, incirc
      Common/ bl16 / rd1, rmh
      Common/ bl17 / a1, dmu, dlambda
      Common/ bl18 / a2, rsa
      Common/ bl19 / aper
      Common/ bl20 / a6, p42, rint
      Common/ bl22 / rsap
      Common/ bl23 / phi1,phi2,phi3,phi4,disab,disac,is,ie,js,je,
     *               idir34,jdir34,NiP
      Common/ bl24 / dlsun, dmsun, dnsun
      Common/ bl25 / pap, pbp, pcp
      Common/ bl26 / xcin,ycin,xcto,ycto,xcbo,ycbo,xcle,ycle,xcri,ycri,
```

```
*            xctl,yctl,xctr,yctr,xcbl,ycbl,xcbr,ycbr
      Common/ bl27 / Nresar
      Common/ bl28 / rsh

      Line='[illegible]
     *[illegible]'


c Parameters from 'Input__.f' :

      Read (*,'(a)') ligne
      Read (*,*) Ntre
      Read (*,'(a)') ligne
      Read (*,*) Dmud
      Read (*,'(a)') ligne
      Read (*,*) Dlad
      Read (*,'(a)') ligne
      Read (*,*) a1deg
      Read (*,'(a)') ligne
      Read (*,*) a2deg
      Read (*,'(a)') ligne
      Read (*,*) Tramir
      Read (*,'(a)') ligne
      Read (*,'(a)') name1
      Read (*,*) NF1
      Read (*,'(a)') name2
      Read (*,*) NF2
      Read (*,'(a)') name3
      Read (*,*) NF3
      Read (*,'(a)') name4
      Read (*,*) NF4
      Read (*,'(a)') name5
      Read (*,*) NF5
      Read (*,'(a)') name6
      Read (*,*) NF6
      Read (*,'(a)') ligne
      Read (*,'(a)') NFtmp

c Determine host name:

      Call system ('hostname > '//NFtmp)
      Open (unit = 10, file = NFtmp, status = 'old')
      Read (10, '(a)') hname
      Close (10)
      Call system ('rm '//NFtmp)

c This file contains the coordinates of the points making up the
c geometry of the MAM-enclosure:

      If (NF1.eq.1) then
        Open (unit = 1, file = name1)
        Write (1,*) 'TITLE = "MIRROR  ATTENUATOR  MOSAIC"'
```

```
      Write (1,*) 'VARIABLES = X, Y, Z, Np'
      Write (1,*) 'ZONE I = 99, J = 44, F = FEPOINT'
    Endif

c This file contains the coordinates of grid-point on the left
c and the lateral-side of the enclosure:

    If (NF2.eq.1) then
      Open (unit = 2, file = name2)
      Write (2,*) 'TITLE = "MIRROR  ATTENUATOR  MOSAIC"'
      Write (2,*) 'VARIABLES = X, Y, Z, Ndf'
    Endif

c This file contains the coordinates of grid-point on the right-side
c of the enclosure:

    If (NF3.eq.1) then
      Open (unit = 3, file = name3)
      Write (3,*) 'TITLE = "MIRROR  ATTENUATOR  MOSAIC"'
      Write (3,*) 'VARIABLES = X, Y, Z, Ndf'
    Endif

c This file contains the coordinates of the points forming the surfaces
c of the geometry:

    If (NF4.eq.1) then
      Open (unit = 4, file = name4)
      Write (4,*) line
      Write (4,*) 'Coordinates of the points forming the MAM enclosure'
      Write (4,*) line

    Endif

c Unit=5 is the keyboard by default.
c Unit=6 is the screen by default.

c This file contains the normalized number of rays absorbed on each
c area element of the enclosure (resulting from Monte-Carlo Method):

    If (NF5.eq.1) then
      Open (unit = 7, file = name5)
      Write (7,*) line
      Write (7,*) 'This file contains the distribution factors'
      Write (7,*) 'of the MAM enclosure.'
      Write (7,*) line
    Endif

c This file contains the results of the current parametric study:

    If (NF6.eq.1) then
      Open (unit = 8, file = name6)
      Write (8,*) line
      Write (8,*) 'This file contains the results of the current'
      Write (8,*) 'parametric study.'
```

```
        Write (8,*)  Line
      Endif
c*****************************************************************
c            List of dimensional parameters
c*****************************************************************
      Pi = dacos(-1.00d0)

c Total number of surfaces:
     Nstot = 51

c Total number of groups:
      Ngtot = 22

c Solar emission at:
      Dmu    = Rad (Dmud)
      Dlambda = Rad (Dlad)

c Seeds for random number generator:
      Call RANDIN (1802,9373)

c ---------------------------------------------------------------
c                    Angles  (degrees)
c ---------------------------------------------------------------
c Direction of the baffle:
      a1 = rad(a1deg)
c Direction of the radiometer:
      a2 = Rad (a2deg)
c Opening angle of the baffle:
      a3 = rad(4.00D0)
c Angle between the direction of the baffle and the
c spacecraft baseline :
      a4 = rad(11.00d0)
c Tilt angle between the sensor wall and the bottom wall :
      a5 = rad (23.00d0)
c some combinations of the given angles:
      a6 = rad(90.00d0) - (a1-a3)
c ---------------------------------------------------------------
c                    Distances (cm)
c ---------------------------------------------------------------
c Baffle reflectors (Thickness : Tau, Width : eta) :
      tau = 0.20d0
      tmir = 0.30d0
      eta = 0.83d0
c Length of the baffle:
      b2 = 24.23d0
c Dimensions of baffle-entrance (B1:x-direction,B4:y-direction):
      b1 = 8.255d0
      b4 = b1
c Scaled baffle-aperture for:
c   ---  Emission on the central spherical mirror:
*         b1p = 0.15875d0
*         b4p = 0.15875d0
c   ---  Emission on the entire MAM:
     b1p = 3.81d0
     b4p = 3.81d0
c Distance from baffle centerline to spacecraft baseline:
```

```
      b3 = 3.651d0
c  Distance from baffle-entrance to MAM (cm):
      b6 = 29.845d0
      b6p = b6 - 1.00d-4
c  Distance from MAM to endbaffle:
      b5 = b6 - b2
c  Length of MAM-enclosure:
      b7 = 8.255d0
c  Width of the closeout plate:
      b8 = 7.937d0
c  Distance from the MAM to the aperture of the radiometer:
      b9 = 9.20d0
c  Length of the sensor wall:
      b10 = 3.175d0

c  ------------------------------------------------------------------
c                          Mirrors
c  ------------------------------------------------------------------
c  Dimension of the MAM (b11 * b11):
      b11 = 3.81d0
      dmX = b11
      dmY = b11
c  Dimensions of the mirrors of CERES-MAM:
      rs = 0.15875d0
      rmh = 0.11938d0
      rd = rmh * dtan(dacos(rmh/rs))
      rd1 = rd
c  ERBE-MAM dimensions needed to determine the distance between the centers
c  of two spherical mirrors (DSX).
      rE = rs
      rmhE = 0.06350d0
      rdE = rmhE * dtan(dacos(rmhE/rE))
c  Configuration of CERES-MAM:
      dsY = 2.00d0 * rdE
      dsp = dsY - 2.00d0 * rd
      dsX = 2.00d0 * dsY * dcos(rad(30.00d0))
      dbX = (dmX - 5.00d0 * dsX) / 2.00d0
      dbY = (dmY - 9.00d0 * dsY) / 2.00d0
c  Radius of the sensor aperture:
      rsa = 2.857d0/2.00d0
c  Radius of MAM aperture:
      Rint = 4.445d0/2.00d0

c  ------------------------------------------------------------------
c                      Surface properties
c  ------------------------------------------------------------------
c  Emissivity, Reflectivity Ratio:
      Do i = 1, nstot
        emis(i) = 0.90d0
        RhoRef(i) = 0.90d0
      Enddo
c
c  The following surfaces are considered perfectly black :
c    baffle entrance, scaled baffle entrance, spherical mirrors,
c    sensor aperture.
      emis( 1) = 1.00d0
```

```
      emis( 2) = 1.00d0
      emis(32) = 1.00d0
      emis(29) = 1.00D0

      RhoRef( 1) = 0.00d0
      RhoRef( 2) = 0.00d0
      RhoRef(29) = 0.00d0
c
c The hemispherical mirrors are considered perfectly specular:
      emis(37)   = 0.00d0
      Rhoref(37) = 1.00d0
c
c-------------------------------------------------------------------------------
c                    Solar  emission
c-------------------------------------------------------------------------------
c The direction of solar emission is a function of 2 angles, Mu and
c Lambda, defined by the position of the sun viewed at the baffle
c aperture. Mu is in the scan direction and Lambda is in the track
c direction. This emission is considered parallel, so the direction
c cosines are constant for given Mu and Lambda.

c Define the direction of emission in the local coordinate system:

      dnume = dsqrt( (dcos(dLambda))**2+(dcos(dMu))**2
     *              *(1.00d0-(dcos(dLambda))**2)   )

      dnp = - dcos(dMu) * dcos(dLambda) / dnume

      if (dLambda.ge.0.00d0) then
        dmp =  dcos(dMu) * dsqrt( 1.00d0-(dcos(dLambda))**2)
      else
        dmp = -dcos(dMu) * dsqrt( 1.00d0-(dcos(dLambda))**2)
      endif

      if (dMu.ge.0.00d0) then
        dlp = -dsqrt (1.00d0 - dmp**2 - dnp**2)
      else
        dlp =  dsqrt (1.00d0 - dmp**2 - dnp**2)
      endif

c Define the direction of emission in the global coordinate system:

      dlsun =  dlp*dcos(a1) - dnp*dsin(a1)
      dmsun = -dmp
      dnsun = -dlp*dsin(a1) - dnp*dcos(a1)


*******************************************************************************************
*                                Definition of surfaces
*******************************************************************************************

* The surfaces forming the geometry of the problem are defined by 22 groups. Each group contains
* different types of surfaces, which are lying on the same plane. A ray emitted from a surface cannot
* hit  another surface in the same group.
*******************************************************************************************************
```

```
c
c  Initialize the groups of surfaces that a surface can "see" :
      Do 35 i = 1, Nstot
         Do 36 j = 1, Ngtot
            see(i,j) = .true.
 36      Continue
 35   Continue
c
c  Initialize the width of the MAM geometry:
      Do i = 1, Nstot
         cYmin(i) = -b4/2.00d0
         cYmax(i) =  b4/2.00d0
      Enddo

c-------------------------------------------------------------------------------
c  GROUP number 1 :
c  The surface S1 is the only element of this group.
      Ngr (1) = 1

c  ..............................................................................
c  Definition of the scaled baffle aperture, surface S1 (34,35,36,37) :
c  (this scaled surface is needed to determine the distribution of
c  solar energy reflected on the enclosure by the central spherical
c  mirror, or by the entire difuser plate).
c
c  List of groups of surfaces that s1 cannot see :
      Do i = 2, Nstot
         see(1,i) = .false.
      Enddo
      see(1,15) = .true.

      epsi = 1.00d-4
c  (in the local coordinate sysytem)
      p1p(1) = b1p / 2.00d0
      p1p(2) = -b4p / 2.00d0
      p1p(3) = epsi
      p2p(1) = -b1p / 2.00d0
      p2p(2) = p1p(2)
      p2p(3) = p1p(3)
      p3p(1) = p2p(1)
      p3p(2) = b4p / 2.00d0
      p3p(3) = p1p(3)
      p4p(1) = p1p(1)
      p4p(2) = p3p(2)
      p4p(3) = p1p(3)
c  (in the global coordinate system)
      Call LOCGLO (p1p, p34, b6, a1)
      Call LOCGLO (p2p, p35, b6, a1)
      Call LOCGLO (p3p, p36, b6, a1)
      Call LOCGLO (p4p, p37, b6, a1)
c  Normal unit vector:
      dc1(1) = dsin(a1)
      dc1(2) = 0.00d0
      dc1(3) = dcos(a1)
```

```
c Normal unit vector in matrix form:
      dcm(1,1) = dc1(1)
      dcm(1,2) = dc1(2)
      dcm(1,3) = dc1(3)
c Subdivision:
      idir(1) = 1
      jdir(1) = 1
      iflag(1) = 1
c Characteristics of S1:
      Call XMIMA (p35, p34, 1)
      cYmin(1) = p34(2)
      cYmax(1) = p36(2)
      omega(1) = datan2 (dc1(1), dc1(3))
      surare(1) = dist(p34, p35) * dist(p35, p36)
c ------------------------------------------------------------------------
c GROUP number 2 :
c There is 1 surface in this group: S2
c ........................................................................
c Definition of the baffle entrance, Surface S2 (1, 2, 3, 4) :
      Ngr (2) = 2
c Groups of surfaces that S2 cannot see :
      see(2,2)  = .false.
      see(2,7)  = .false.
      see(2,11) = .false.
      see(2,12) = .false.
      see(2,13) = .false.
c (in the local coordinate system)
      p1p(1) = b1 / 2.00d0
      p1p(2) = -b4 / 2.00d0
      p1p(3) = 0.00d0
      p2p(1) = -b1 / 2.00d0
      p2p(2) = p1p(2)
      p2p(3) = 0.00d0
      p3p(1) = p2p(1)
      p3p(2) = b4 / 2.00d0
      p3p(3) = 0.00d0
      p4p(1) = p1p(1)
      p4p(2) = p3p(2)
      p4p(3) = 0.00d0
c (in the global coordinate system)
      Call LOCGLO (p1p, p1, b6, a1)
      Call LOCGLO (p2p, p2, b6, a1)
      Call LOCGLO (p3p, p3, b6, a1)
      Call LOCGLO (p4p, p4, b6, a1)
c Normal init vector :
      dc2(1) = dc1(1)
      dc2(2) = dc1(2)
      dc2(3) = dc1(3)
c Normal unit vector in matrix form :
      dcm(2,1) = dc2(1)
      dcm(2,2) = dc2(2)
      dcm(2,3) = dc2(3)
c Subdivision :
```

```
      idir(2) = 1
      jdir(2) = 1
      iflag(2) = 2
c Characteristics of the rectangular perpendicular to the x,z-plane S2:
      Call XMIMA (p1, p2, 2)
      omega(2) = datan2 (dc2(1), dc2(3))
      surare(2) = dist(p1, p2) * dist(p2, p3)
c ------------------------------------------------------------------------------
c GROUP number 3 :
c Group number 3 contains 9 surfaces forming the right side of the
c enclosure, which are : S3, S4, S5, S6, S7, S8, S9, S10, S11.
c ..............................................................................
c Definition of surface S3 (5, 6, 10, 11) :
      Ngr (3) = 3
c groups of surfaces S3 cannot see:
      see(3,3) = .false.
      see(3,7) = .false.
      see(3,11) = .false.
      see(3,12) = .false.
      see(3,13) = .false.
c (in the local coordinate system)
      p1p(1) = b1/2.00d0 - b2*dsin(a3)
      p1p(2) = -b4/2.00d0
      p1p(3) = 0.00d0
      p2p(1) = -b1/2.00d0 + b2*dsin(a3)
      p2p(2) = p1p(2)
      p2p(3) = 0.00d0
      p3p(1) = p2p(1)
      p3p(2) = p1p(2)
      p3p(3) = b2
      p4p(1) = p1p(1)
      p4p(2) = p1p(2)
      p4p(3) = p3p(3)
c (in the global coordinate system)
      Call LOCGLO (p1p, p5, b6, a1)
      Call LOCGLO (p2p, p6, b6, a1)
      Call LOCGLO (p3p, p11, b6, a1)
      Call LOCGLO (p4p, p10, b6, a1)
c Normal unit vector :
      dc3(1) = 0.00d0
      dc3(2) = 1.00d0
      dc3(3) = 0.00d0
c Normal unit vector in matrix form :
      dcm(3,1) = dc3(1)
      dcm(3,2) = dc3(2)
      dcm(3,3) = dc3(3)
c Subdivision :
      idir(3) = 20
      jdir(3) = 6
      surare(3) = dist(p5, p10) * dist(p10, p11)
c ..............................................................................
c Definition of the surface S4 (2, 6, 11 which are all known) :
      Ngr(4) = 3
c Groups of surfaces that S4 cannot see :
```

```
      see(4,3)  = .false.
      see(4,7)  = .false.
      see(4,10) = .false.
      see(4,11) = .false.
      see(4,12) = .false.
      see(4,13) = .false.
c  Normal unit vector :
      dc4(1) = dc3(1)
      dc4(2) = dc3(2)
      dc4(3) = dc3(3)
c  Normal unit vector in matrix form :
      dcm(4,1) = dc4(1)
      dcm(4,2) = dc4(2)
      dcm(4,3) = dc4(3)
c  Subdivision :
      idir(4) = 20
      jdir(4) = 2
      Call SURTRI (p6, p2, p11, surare(4))
c  .....
c  Definition of the surface S5 (1, 5, 10 which are all known) :
      Ngr(5) = 3
c  Groups of surfaces that S5 cannot see :
      see(5,3)  = .false.
      see(5,7)  = .false.
      see(5,10) = .false.
      see(5,11) = .false.
      see(5,12) = .false.
      see(5,13) = .false.
c  Normal unit vector :
      dc5(1) = dc4(1)
      dc5(2) = dc4(2)
      dc5(3) = dc4(3)
c  Normal unit vector in matrix form :
      dcm(5,1) = dc5(1)
      dcm(5,2) = dc5(2)
      dcm(5,3) = dc5(3)
c  Subdivision :
      idir(5) = 20
      jdir(5) = 2
      Call SURTRI (p5, p1, p10, surare(5))
c  .....
c  Definition of the surface S6 (10, 14, 18) :
      Ngr(6) = 3
c  Groups of surfaces that S6 cannot see :
      see(6,3)  = .false.
c  (in the local coordinate system)
      p1p(1) = -b3
      p1p(2) = -b4/2.00d0
      p1p(3) = 0.00d0
      p2p(1) = p1p(1) + b7*dsin(a4)
      p2p(2) = p1p(2)
      p2p(3) = b7*dcos(a4)
c  (in the global coordinate system)
      Call LOCGLO (p1p, p14, b5, a1)
```

```
      Call LOCGLO (p2p, p18, b5, a1)
c  Normal unit vector :
      dc6(1) = dc5(1)
      dc6(2) = dc5(2)
      dc6(3) = dc5(3)
c  Normal unit vector in matrix form :
      dcm(6,1) = dc6(1)
      dcm(6,2) = dc6(2)
      dcm(6,3) = dc6(3)
c  Subdivision :
      idir(6) = 10
      jdir(6) = 10
      Call SURTRI (p10, p11, p18, surare(6))
c.....
c  Definition of the surface S7 (10, 18, 17) :
      Ngr(7) = 3
c  Groups of surfaces that S7 cannot see :
      see(7,3)  = .false.
c  (in the global coordinate system)
      p17(1) = p18(1) + b8*dcos(a1+a4)
      p17(2) = p18(2)
      p17(3) = p18(3) - b8*dsin(a1+a4)
c  Normal unit vector :
      dc7(1) = dc6(1)
      dc7(2) = dc6(2)
      dc7(3) = dc6(3)
c  Normal unit vector in matrix form :
      dcm(7,1) = dc7(1)
      dcm(7,2) = dc7(2)
      dcm(7,3) = dc7(3)
c  Subdivision :
      idir(7) = 10
      jdir(7) = 10
      Call SURTRI (p10, p18, p17, surare(7))
c.....
c  Definition of the surface S8 (10,17,22) :
      Ngr(8) = 3
c  Groups of surfaces that S8 cannot see :
      see(8,3)  = .false.
c  (in the local coordinate system)
      p1p(1) = b10/2.00d0
      p1p(2) = -b4/2.00d0
      p1p(3) = 0.00d0
c  (in the global coordinate system)
      Call LOCGLO (p1p, p23, b9, -a2)

      cm2 = dtan(pi/2.00d0 - (a1+a4))
      cb2 = p17(3) - cm2*p17(1)
      cm3 = dtan(a2-pi/2.00d0)
      cb3 = p23(3) - cm3*p23(1)

      p22(1) = (cb3-cb2)/(cm2-cm3)
      p22(2) = -b4/2.00d0
      p22(3) = cm3*p22(1) + cb3
```

```
c  Normal unit vector :
       dc8(1) = dc7(1)
       dc8(2) = dc7(1)
       dc8(3) = dc7(3)
c  Normal unit vector in marix form :
       dcm(8,1) = dc8(1)
       dcm(8,2) = dc8(2)
       dcm(8,3) = dc8(3)
c  Subdivision :
       idir(8) = 10
       jdir(8) = 10
       Call SURTRI (p10, p17, p22, surare(8))
c.....
c  Definition of the surface S9 (10,22,23) :
       Ngr(9) = 3
c  Groups of surfaces that S9 cannot see :
       see(9,3)  = .false.
       see(9,2)  = .false.
       see(9,16) = .false.
       see(9,17) = .false.
       see(9,18) = .false.
       see(9,19) = .false.
       see(9,20) = .false.
c  Normal unit vector :
       dc9(1) = dc8(1)
       dc9(2) = dc8(2)
       dc9(3) = dc8(3)
c  Normal unit vector in matrix form :
       dcm(9,1) = dc9(1)
       dcm(9,2) = dc9(2)
       dcm(9,3) = dc9(3)
c  Subdivision :
       idir(9) = 10
       jdir(9) = 10
       Call SURTRI (p10, p22, p23, surare(9))
c.....
c  Definition of the surface S10 (10,23,24) :
       Ngr(10) = 3
c  Groups of surfaces that S10 cannot see :
       see(10,3)  = .false.
       see(10,2)  = .false.
       see(10,5)  = .false.
       see(10,16) = .false.
       see(10,17) = .false.
       see(10,18) = .false.
       see(10,19) = .false.
       see(10,20) = .false.
       see(10,21) = .false.
c  (in the local coordinate system)
       p1p(1) = -b10/2.00d0
       p1p(2) = -b4/2.00d0
       p1p(3) = 0.00d0
c  (in the global coordinate system)
       Call LOCGLO (p1p, p24, b9, -a2)
```

```
c Normal unit vector :
      dc10(1) = dc9(1)
      dc10(2) = dc9(2)
      dc10(3) = dc9(3)
c Normal unit vector in matrix form :
      dcm(10,1) = dc10(1)
      dcm(10,2) = dc10(2)
      dcm(10,3) = dc10(3)
c Subdivision :
      idir(10) = 10
      jdir(10) = 10
      Call SURTRI (p10, p23, p24, surare(10))
c.....
c Definition of the surface S11 (10,24,25) :
      Ngr(11) = 3
c Groups of surfaces that S11 cannot see :
      see(11,3)  = .false.
      see(11,2)  = .false.
      see(11,5)  = .false.
      see(11,7)  = .false.
      see(11,16) = .false.
      see(11,17) = .false.
      see(11,18) = .false.
      see(11,19) = .false.
      see(11,20) = .false.
      see(11,21) = .false.
      see(11,22) = .false.
c (in the global coordinate system)

      cm2 = dtan(a2+a5)
      cb2 = p24(3) - cm2*p24(1)
      cm3 = (p10(3)-p1(3))/(p10(1)-p1(1))
      cb3 = p10(3) - cm3*p10(1)

      p25(1) = (cb2-cb3)/(cm3-cm2)
      p25(2) = -b4/2.00d0
      p25(3) = cm3*p25(1) + cb3
c Normal unit vector :
      dc11(1) = dc10(1)
      dc11(2) = dc10(2)
      dc11(3) = dc10(3)
c Normal unit vector in matrix form :
      dcm(11,1) = dc11(1)
      dcm(11,2) = dc11(2)
      dcm(11,3) = dc11(3)
c Subdivision :
      idir(11) = 10
      jdir(11) = 10
      Call SURTRI (p10, p24, p25, surare(11))
c-------------------------------------------------------------------------------
c GROUP number 4 :
c This group consists of 9 surfaces forming the left side of the
c enclosure, which are : S12, S13, S14, S15, S16, S17, S18, S19, S20.
c.....
```

```
c Definition of the surface S13 (8, 9, 12, 13) :
      Ngr(13) = 4
c Groups of surfaces that S13 cannot see :
      see(13,4)  = .false.
      see(13,7)  = .false.
      see(13,11) = .false.
      see(13,12) = .false.
      see(13,13) = .false.
c (in the global coordinate system)
      p8(1) = p6(1)
      p8(2) = +b4/2.00d0
      p8(3) = p6(3)
      p9(1) = p5(1)
      p9(2) = p8(2)
      p9(3) = p5(3)
      p12(1) = p11(1)
      p12(2) = p8(2)
      p12(3) = p11(3)
      p13(1) = p10(1)
      p13(2) = p8(2)
      p13(3) = p10(3)
c Normal unit vector :
      dc13(1) = 0.00d0
      dc13(2) = - 1.00d0
      dc13(3) = 0.00d0
c Normal unit vector in matrix form :
      dcm(13,1) = dc13(1)
      dcm(13,2) = dc13(2)
      dcm(13,3) = dc13(3)
c Subdivision :
      idir(13) = 20
      jdir(13) = 6
      surare(13) = dist(p9, p8) * dist(p9, p13)
c.....
c Definition of the surface S12 (3, 8, 12) :
      Ngr(12) = 4
c Groups of surfaces that S12 cannot see :
      see(12,4)  = .false.
      see(12,7)  = .false.
      see(12,11) = .false.
      see(12,12) = .false.
      see(12,13) = .false.
c Normal unit vector :
      dc12(1) = dc13(1)
      dc12(2) = dc13(2)
      dc12(3) = dc13(3)
c Normal unit vector in matrix form :
      dcm(12,1) = dc12(1)
      dcm(12,2) = dc12(2)
      dcm(12,3) = dc12(3)
c Subdivision :
      idir(12) = 20
      jdir(12) = 2
```

```
      Call SURTRI (p8, p3, p12, surare(12))
c.....
c  Definition of the surface S14 (9, 4, 13) :
      Ngr(14) = 4
c  Groups of surfaces that S14 cannot see :
      see(14,4)  = .false.
      see(14,7)  = .false.
      see(14,11) = .false.
      see(14,12) = .false.
      see(14,13) = .false.
c  Normal unit vector :
      dc14(1) = dc12(1)
      dc14(2) = dc12(2)
      dc14(3) = dc12(3)
c  Normal unit vector in matrix form :
      dcm(14,1) = dc14(1)
      dcm(14,2) = dc14(2)
      dcm(14,3) = dc14(3)
c  Subdivision :
      idir(14) = 20
      jdir(14) = 2
      Call SURTRI (p9, p4, p13, surare(14))
c.....
c  Definition of the surface S15 (13, 16, 20) :
      Ngr(15) = 4
c  Groups of surfaces that S15 cannot see :
      see(15,4)  = .false.
c  (in the global coordinate system)
      p16(1) = p14(1)
      p16(2) = b4/2.00d0
      p16(3) = p14(3)
      p20(1) = p18(1)
      p20(2) = p16(2)
      p20(3) = p18(3)
c  Normal unit vector :
      dc15(1) = dc14(1)
      dc15(2) = dc14(2)
      dc15(3) = dc14(3)
c  Normal unit vector in matrix form :
      dcm(15,1) = dc15(1)
      dcm(15,2) = dc15(2)
      dcm(15,3) = dc15(3)
c  Subdivision :
      idir(15) = 10
      jdir(15) = 10
      Call SURTRI (p13, p16, p20, surare(15))
c.....
c  Definition of surface S16 (13, 20, 21) :
      Ngr(16) = 4
c  Groups of surfaces that S16 cannot see :
      see(16,4) = .false.
c  (in the global coordinate system)
      p21(1) = p17(1)
      p21(2) = b4/2.00d0
```

```
      p21(3) = p17(3)
c  Normal unit vector :
      dc16(1) = dc15(1)
      dc16(2) = dc15(2)
      dc16(3) = dc15(3)
c  Normal unit vector in matrix form :
      dcm(16,1) = dc16(1)
      dcm(16,2) = dc16(2)
      dcm(16,3) = dc16(3)
c  Subdivision :
      idir(16) = 10
      jdir(16) = 10
      Call SURTRI (p13, p20, p21, surare(16))
c.....
c  Definition of the surface S17 (13, 21, 26) :
      Ngr(17) = 4
c  Groups of surfaces that S17 cannot see :
      see(17,4) = .false.
c  (in the global coordinate system)
      p26(1) = p22(1)
      p26(2) = b4/2.00d0
      p26(3) = p22(3)
c  Normal unit vector :
      dc17(1) = dc16(1)
      dc17(2) = dc16(2)
      dc17(3) = dc16(3)
c  Normal unit vector in matrix form :
      dcm(17,1) = dc17(1)
      dcm(17,2) = dc17(2)
      dcm(17,3) = dc17(3)
c  Subdivision :
      idir(17) = 10
      jdir(17) = 10
      Call SURTRI (p13, p21, p26, surare(17))
c.....
c  Definition of the surface S18 (13, 26, 27) :
      Ngr(18) = 4
c  Groups of surfaces that S18 cannot see :
      see(18,4) = .false.
      see(18,2) = .false.
      see(18,16) = .false.
      see(18,17) = .false.
      see(18,18) = .false.
      see(18,19) = .false.
      see(18,20) = .false.
c  (in the global coordinate system)
      p27(1) = p23(1)
      p27(2) = b4/2.00d0
      p27(3) = p23(3)
c  Normal unit vector :
      dc18(1) =dc17(1)
      dc18(2) = dc17(2)
      dc18(3) = dc17(3)
c  Normal unit vector in matrix form :
```

```
      dcm(18,1) = dc18(1)
      dcm(18,2) = dc18(2)
      dcm(18,3) = dc18(3)
c  Subdivision :
      idir(18) = 10
      jdir(18) = 10
      Call SURTRI (p13, p26, p27, surare(18))
c.....
c  Definition of the surface S19 (13, 27, 28) :
      Ngr(19) = 4
c  Group of surfaces that S19 cannot see :
      see(19,4) = .false.
      see(19,2) = .false.
      see(19,16) = .false.
      see(19,17) = .false.
      see(19,18) = .false.
      see(19,19) = .false.
      see(19,20) = .false.
      see(19,21) = .false.
c  (in the global coordinate system)
      p28(1) = p24(1)
      p28(2) = b4/2.00d0
      p28(3) = p24(3)
c  Normal unit vector :
      dc19(1) = dc18(1)
      dc19(2) = dc18(2)
      dc19(3) = dc18(3)
c  Normal unit vector in matrix form :
      dcm(19,1) = dc19(1)
      dcm(19,2) = dc19(2)
      dcm(19,3) = dc19(3)
c  Subdivision :
      idir(19) = 10
      jdir(19) = 10
      Call SURTRI (p13, p27, p28, surare(19))
c.....
c  Definition of the surface S20 (13, 28, 29) :
      Ngr(20) = 4
c  Groups of surfaces that S20 cannot see :
      see(20,4) = .false.
      see(20,2) = .false.
      see(20,5) = .false.
      see(20,7) = .false.
      see(20,16) = .false.
      see(20,17) = .false.
      see(20,18) = .false.
      see(20,19) = .false.
      see(20,20) = .false.
      see(20,21) = .false.
      see(20,22) = .false.
c  (in the global coordinate system)
      p29(1) = p25(1)
      p29(2) = b4/2.00d0
      p29(3) = p25(3)
```

```
c Normal unit vector :
      dc20(1) = dc19(1)
      dc20(2) = dc19(2)
      dc20(3) = dc19(3)
c Normal unit vector in matrix form :
      dcm(20,1) = dc20(1)
      dcm(20,2) = dc20(2)
      dcm(20,3) = dc20(3)
c Subdivision :
      idir(20) = 10
      jdir(20) = 10
      Call SURTRI (p13, p28, p29, surare(20))
c-------------------------------------------------------------------------
c GROUP number 5 :
c The surface S21 is the only surface of this group.
c.........................................................................
c Definition of the surface S21 (2, 3, 11, 12, 7, 38) :
      Ngr(21) = 5
c
c Note : The surfaces forming the top side of the enclosure are defined
c        by 6 points : 4 corners, 2 median points and a flag, denoted
c        iflag, which characterizes of the graphic representation of
c        the surface in order to show the internal structure of the
c        enclosure.
c        If iflag = 1, then the entire surface is represented.
c        If iflag = 2, then only the left side is represented.
c        If iflag = 3, then only the right side is represented.
c
c Groups of surfaces that S21 cannot see :
      see(21,5) = .false.
      see(21,7) = .false.
      see(21,12) = .false.
      see(21,13) = .false.
c (in the global coordinate system)
      p7(1) = p2(1)
      p7(2) = 0.00d0
      p7(3) = p2(3)
      p38(1) = p11(1)
      p38(2) = 0.00d0
      p38(3) = p11(3)
c Normal unit vector :
      dc21(1) = dcos(a1+a3)
      dc21(2) = 0.00d0
      dc21(3) = - dsin(a1+a3)
c Normal uint vector in matrix form :
      dcm(21,1) = dc21(1)
      dcm(21,2) = dc21(2)
      dcm(21,3) = dc21(3)
c Subdivision :
      idir(21) = 20
      jdir(21) = 10
      iflag(21) = 2
c Charactersitics of the rectangular surface perpendicular to the
c x,z-plane, S21 :
      Call XMIMA (p2, p11, 21)
      omega(21) = datan2 (dc21(1), dc21(3))
```

```
      surare(21) = dist(p2, p3) * dist(p2, p11)
c-----------------------------------------------------------------------
c  GROUP number 6 :
c  there are 2 surfaces in this group, which are S22 and S31.
c.....
c  Definition of the front of the baffle bottom side, S22 (1, 4, 10, 13) :
      Ngr(22) = 6
c  Group of surfaces that S22 cannot see :
      see(22,6) = .false.
      see(22,7) = .false.
      see(22,10) = .false.
      see(22,11) = .false.
      see(22,12) = .false.
      see(22,13) = .false.
      see(22,14) = .false.
c  Normal unit vector :
      dc22(1) = - dcos(a1-a3)
      dc22(2) = 0.00d0
      dc22(3) = dsin(a1-a3)
c  Normal unit vector in matrix form :
      dcm(22,1) = dc22(1)
      dcm(22,2) = dc22(2)
      dcm(22,3) = dc22(3)
c  Subdivision :
      idir(22) = 20
      jdir(22) = 10
      iflag(22) = 1
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S22 :
      Call XMIMA (p1, p10, 22)
      omega(22) = datan2 (dc22(1), dc22(3))
      surare(22) = dist(p1, p4) * dist(p1, p10)
c.....
c  Definition of the back of the baffle bottom side, S31 (10, 13, 25, 29):
      Ngr(31) = 6
c  Group of surfaces that S31 cannot see :
      see(31,6) = .false.
      see(31,2) =  .false.
      see(31,5) = .false.
      see(31,7) = .false.
      see(31,16) = .false.
      see(31,17) = .false.
      see(31,18) = .false.
      see(31,19) = .false.
      see(31,20) = .false.
      see(31,21) = .false.
      see(31,22) = .false.
      see(31,8) = .false.
      see(31,14) = .false.
c  Normal unit vector :
      dc31(1) = - dc22(1)
      dc31(2) = - dc22(2)
      dc31(3) = - dc22(3)
c  Normal unit vector in matrix form :
```

```
      dcm(31,1) = dc31(1)
      dcm(31,2) = dc31(2)
      dcm(31,3) = dc31(3)
c  Subdivision :
      idir(31) = 10
      jdir(31) = 10
      iflag(31) = 1
      surare(31) = dist(p25, p29) * dist(p25, p10)
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S31 :
      Call XMIMA (p25, p10, 31)
      omega(31) = datan2 (dc31(1), dc31(3))
c-------------------------------------------------------------------------
c  GROUP number 7 :
c  S23 is the only surface in this group.
c.........................................................................
c  Definition of the surface S23 (11, 12, 14, 16, 38, 15) :
      Ngr(23) = 7
c  Groups of surfaces that S23 cannot see :
      see(23,7) = .false.
      see(23,2) = .false.
      see(23,5) = .false.
      see(23,6) = .false.
      see(23,16) = .false.
      see(23,17) = .false.
      see(23,18) = .false.
      see(23,19) = .false.
      see(23,20) = .false.
      see(23,21) = .false.
      see(23,22) = .false.
      see(23,13) = .false.
c  (in the global coordinate system)
      p15(1) = p14(1)
      p15(2) = 0.00d0
      p15(3) = p14(3)
c  Normal unit vector :
      dc23(1) = dsin(a1)
      dc23(2) = 0.00d0
      dc23(3) = dcos(a1)
c  Normal unit vector in matrix form :
      dcm(23,1) = dc23(1)
      dcm(23,2) = dc23(2)
      dcm(23,3) = dc23(3)
c  Subdivision :
      idir(23) = 5
      jdir(23) = 10
      iflag(23) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S23 :
      Call XMIMA (p11, p14, 23)
      omega(23) = datan2 (dc23(1), dc23(3))
      surare(23) = dist(p11, p14) * dist(p11, p12)
c-------------------------------------------------------------------------
c  GROUP number 8 :
c  The surface S24 is the only surface in this group.
```

```
c.....
c  Definition of the surface S24 (14, 15, 16, 18, 20, 19) :
      Ngr(24) = 8
c  Groups of surfaces that S24 cannot see :
      see(24,8) = .false.
      see(24,5) = .false.
c  (in the global coordinate system)
      p19(1) = p18(1)
      p19(2) = 0.00d0
      p19(3) = p18(3)
c  Normal unit vector :
      dc24(1) = dcos(a1+a4)
      dc24(2) =  0.00d0
      dc24(3) = - dsin(a1+a4)
c  Normal uint vector in matrix form :
      dcm(24,1) = dc24(1)
      dcm(24,2) = dc24(2)
      dcm(24,3) = dc24(3)
c  Subdivision :
      idir(24) = 10
      jdir(24) = 10
      iflag(24) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S24 :
      Call XMIMA (p14, p18, 24)
      omega(24) = datan2 (dc24(1), dc24(3))
      surare(24) = dist(p11, p16) * dist(p11, p18)
c-----------------------------------------------------------------------
c  GROUP number 9 :
c  The surface S25 is the only surface of this group.
c.....
c  Definition of the closeout plate, surface S25 (17, 18, 19, 20, 21, 39)
      Ngr(25) = 9
c  Group of surfaces that S25 cannot see :
      see(25,9) = .false.

c  (in the global coordinate system)
      p39(1) = p17(1)
      p39(2) = 0.00d0
      p39(3) = p17(3)
c  Normal unit vector :
      dc25(1) = -dsin(a1+a4)
      dc25(2) = 0.00d0
      dc25(3) = -dcos(a1+a4)
c  Normal uint vector in matrix form :
      dcm(25,1) = dc25(1)
      dcm(25,2) = dc25(2)
      dcm(25,3) = dc25(3)
c  Subdivision :
      idir(25) = 10
      jdir(25) = 10
      iflag(25) = 1
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S25 :
      Call XMIMA (p17, p18, 25)
      omega(25) = datan2 (dc25(1), dc25(3))
```

```
      surare(25) = dist(p17, p18) * dist(p17, p21)
c-----------------------------------------------------------
c  GROUP number 10 :
c  S26 is the only surface in this group.
c.....
c  Definition of the surface S26 (17, 21, 22, 26):
      Ngr(26) = 10
c  Groups of surfaces that S26 cannot see :
      see(26,10) = .false.
      see(26,2) = .false.
      see(26,16) = .false.
      see(26,17) = .false.
      see(26,18) = .false.
      see(26,19) = .false.
      see(26,20) = .false.
      see(26,11) = .false.
c  Normal unit vector :
      dc26(1) = -dc24(1)
      dc26(2) = 0.00d0
      dc26(3) = -dc24(3)
c  Normal unit vector in matrix form :
      dcm(26,1) = dc26(1)
      dcm(26,2) = dc26(2)
      dcm(26,3) = dc26(3)
c  Subdivision :
      idir(26) = 10
      jdir(26) = 10
      iflag(26) = 1
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S26 :
      Call XMIMA (p22, p17, 26)
      omega(26) = datan2 (dc26(1), dc26(3))
      surare(26) = dist(p22, p17) * dist(p22, p26)
c-----------------------------------------------------------
c  GROUP number 11 :
c  S27 is the only surface of this group.
c.....
c  Definition of the surface S27 (22, 26, 23, 27)
      Ngr(27) = 11
c  Groups of surfaces that S27 cannot see :
      see(27,11) = .false.
      see(27,2) = .false.
      see(27,16) = .false.
      see(27,17) = .false.
      see(27,18) = .false.
      see(27,19) = .false.
      see(27,20) = .false.
      see(27,21) = .false.
      see(27,10) = .false.
c  Normal unit vector :
      dc27(1) = -dcos(a2)
      dc27(2) = 0.00d0
      dc27(3) = -dsin(a2)
c  Normal unit vector in matrix form :
      dcm(27,1) = dc27(1)
```

```
      dcm(27,2) = dc27(2)
      dcm(27,3) = dc27(3)
c  Subdivision :
      idir(27) = 10
      jdir(27) = 10
      iflag(27) = 1
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S27 :
      Call XMIMA (p22, p23, 27)
      omega(27) = datan2 (dc27(1), dc27(3))
      surare(27) = dist(p23, p22) * dist(p23, p27)
c-------------------------------------------------------------------------------
c  GROUP number 12 :
c  There are 2 surfaces in this group, which are the sensor wall (S28)
c  and the sensor aperture (S29).
c.....
c  Definition of the sensor wall S28 (23, 27, 24, 28)
      Ngr(28) = 12
c  Groups of surfaces that S28 cannot see :
      see(28,12) = .false.
      see(28,2) = .false.
      see(28,16) = .false.
      see(28,17) = .false.
      see(28,18) = .false.
      see(28,19) = .false.
      see(28,20) = .false.
      see(28,21) = .false.

c  Normal unit vector :
      dc28(1) = -dsin(a2)
      dc28(2) = 0.00d0
      dc28(3) = dcos(a2)
c  Normal unit vector in matrix form :
      dcm(28,1) = dc28(1)
      dcm(28,2) = dc28(2)
      dcm(28,3) = dc28(3)
c  Subdivision :
      idir(28) = 10
      jdir(28) = 20
      iflag(28) = 1
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S28 :
      Call XMIMA (p24, p23, 28)
      omega(28) = datan2 (dc28(1), dc28(3))
      surare(28) = dist(p24, p23) * dist(p24, p28) - pi*rsa**2
c.....
c  Definition of the sensor aperture, S29 (43) :
      Ngr(29) = 12
c  Groups of surfaces that S29 cannot see :
      see(29,12) = .false.
      see(29,2) = .false.
      see(29,16) = .false.
      see(29,17) = .false.
      see(29,18) = .false.
```

```
      see(29,19) = .false.
      see(29,20) = .false.
      see(29,21) = .false.
c  (in the local coordinate system)
      p1p(1) = 0.00d0
      p1p(2) = 0.00d0
      p1p(3) = 0.00d0
c  (in the global coordinate system)
      Call LOCGLO (p1p, p43, b9, -a2)
c  Subdivision (Nr : radial, Nphi : circumferential) :
      Nr = 4
      Nphi = 20
c  Surface area:
      surare(29) = pi*rsa**2
c  Determine the radii for the numbering of the sensor aperture :
      rsap(1) = 0.00d0
      rsap(2) = rsa / dsqrt( dble(nr) )
      Do i = 3, Nr
         dnu = dble(i-1)
         dde = dble(i-2)
         rsap(i) = rsap(i-1) * dsqrt(dnu/dde)
      Enddo
      rsap(Nr+1) = rsa


c-------------------------------------------------------------------------------
c  GROUP number 13 :
c  S30 is the only surface in this group.
c.....
c  Definition of the bottom wall, surface S30 (24, 28, 25, 29) :
      Ngr(30) = 13
c  Groups of surfaces that S29 cannot see :
      see(30,13) = .false.
      see(30,2) = .false.
      see(30,5) = .false.
      see(30,7) = .false.
      see(30,16) = .false.
      see(30,17) = .false.
      see(30,18) = .false.
      see(30,19) = .false.
      see(30,20) = .false.
      see(30,21) = .false.
      see(30,22) = .false.
c  Normal unit vector :
      dc30(1) = -dsin(a2+a5)
      dc30(2) = 0.00d0
      dc30(3) = dcos(a2+a5)
c  Normal unit vector in matrix form:
      dcm(30,1) = dc30(1)
      dcm(30,2) = dc30(2)
      dcm(30,3) = dc30(3)
c  Subdivision :
      idir(30) = 10
```

```
      jdir(30) = 10
      iflag(30) = 1
c Characteristics of the rectangular surface perpendicular to the
c x,z-plane, S30 :
      Call XMIMA (p25, p24, 30)
      omega(30) = datan2 (dc30(1), dc30(3))
     surare(30) = dist(p25, p24) * dist(p25, p29)
c-------------------------------------------------------------------------------
c GROUP number 14 :
c the 3 elements of the internal wall belong to this group.
c.....
c Definition of the top side of the internal wall, S34 (10,13,40,41,42)
      Ngr(34) = 14
c Groups of surfaces that S34 cannot see :
      see(34,14) = .false.
      see(34,6) = .false.
      see(34,10) = .false.
      see(34,11) = .false.
      see(34,12) = .false.
      see(34,13) = .false.
c (in the global coordinate system)
      cm1 = (p17(3)-p18(3))/(p17(1)-p18(1))
      cb1 = p17(3) - cm1*p17(1)
      cm2 = (p10(3)-p1(3))/(p10(1)-p1(1))
      cb2 = p10(3) - cm2*p10(1)

      p40(1) = (cb2-cb1)/(cm1-cm2)
      p40(2) = -b4/2.00d0
      p40(3) = cm2*p40(1) + cb2

      cm3 = - dtan(pi/2.00d0 - a2)

      p42(1) = - cb2 / (cm2-cm3)
      p42(2) = 0.00d0
      p42(3) = cm3*p42(1)

      p41(1) = p40(1)
      p41(2) = b4/2.00d0
      p41(3) = p40(3)
c Normal unit vector :
      dc34(1) = dc22(1)
      dc34(2) = dc22(2)
      dc34(3) = dc22(3)
c Normal unit vector in matrix form :
      dcm(34,1) = dc34(1)
      dcm(34,2) = dc34(2)
      dcm(34,3) = dc34(3)
c Subdivision :
      idir(34) = 15
      jdir(34) = 10
      idir34 = idir(34)
      jdir34 = jdir(34)
      Nip = 3
```

```
c    ( the area around the circle is sudivided into 8*Nip slices)
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S34 :
      Call XMIMA (p10, p40, 34)
      omega(34) = datan2 (dc34(1), dc34(3))
      surare(34) = dist(p10, p13) * dist(p10, p40) - pi*rint**2
c  Characteristics needed for the numbering of the area elements
c  of the internal wall :
c    "boundary" of the circle :
      Call GLOLOC (p40, pcp, p42(1), p42(3), a6)
      Call GLOLOC (p10, pap, p42(1), p42(3), a6)
      Call GLOLOC (p13, pbp, p42(1), p42(3), a6)
      is = int((-Rint-pap(1))/(pcp(1)-pap(1))*idir(34)    )
      ie = int(( Rint-pap(1))/(pcp(1)-pap(1))*idir(34) + 1)
      js = int((-Rint-pap(2))/(pbp(2)-pap(2))*jdir(34)    )
      je = int(( Rint-pap(2))/(pbp(2)-pap(2))*jdir(34) + 1)
c    Angles :
      dex = dble(ie)*(pcp(1)-pap(1))/dble(idir(34)) + pap(1)
      dexm = dabs(dble(is)*(pcp(1)-pap(1))/dble(idir(34)) + pap(1))
      dey = dble(je)*(pbp(2)-pap(2))/dble(jdir(34)) + pap(2)
      phi1 = datan (dey/dex)
      phi2 = pi/2.00d0 - phi1
      phi3 = datan (dexm/dey)
      phi4 = pi/2.00d0 - phi3
      disab = dist (pap, pbp)
      disac = dist (pap, pcp)
c.....
c  Definition of the bottom side of the internal wall, S36 :
      Ngr(36) = 14
c  Groups of surfaces that S36 cannot see :
      see(36,14) = .false.
      see(36,2) = .false.
      see(36,5) = .false.
      see(36,6) = .false.
      see(36,7) = .false.
      see(36,8) = .false.
      see(36,15) = .false.
      see(36,16) = .false.
      see(36,17) = .false.
      see(36,18) = .false.
      see(36,19) = .false.
      see(36,20) = .false.
      see(36,21) = .false.
      see(36,22) = .false.
c  Normal unit vector :
      dc36(1) = - dc34(1)
      dc36(2) = - dc34(2)
      dc36(3) = -dc34(3)
c  Normal unit vector in matrix form :
      dcm(36,1) = dc36(1)
      dcm(36,2) = dc36(2)
      dcm(36,3) = dc36(3)
c  subdivision :
      idir(36) = idir(34)
```

```
      jdir(36) = jdir(34)
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S36 :
      Call XMIMA (p10, p40, 36)
      omega(36) = datan2 (dc36(1), dc36(3))
     surare(36) = surare(34)
c.....
c  Definition of the MAM aperture, surface S35 :
c  This is a circular surface of radius Rint, and centered at p42.
      Ngr(35) = 14
      idir(35) = Nip
c  Groups of surfaces that s35 cannot see :
      see(35,14) = .false.
      see(35,6) = .false.
c-----
c  GROUP number 15 :
c  The 2 sides of the diffuser plate (S32, and S33) and the sperical
c  mirrors are in this group. Note that the 105 sperical surfaces are
c considered as one unique surface (S37).
c.....
c  Definition of the top side of the diffuser plate, S32 (30,31,32,33) :
      Ngr(32) = 15
c  Groups of surfaces that S32 cannot see :
      see(32,15) = .false.
      see(32,9) = .false.
c  Initialize the centers of the mirrors of the cells forming the
c  difuser plate :
      Do i = 1, 5
         xc(i) = 0.00d0
         yc(i) = 0.00d0
      Enddo

c  Initialize the logical variable defining whether or not the ray is
c  incident on a spherical mirror and the coordinates of the center
c  of this mirror :
      Incirc = .false.
      xcs =  0.00d0
      ycs =  0.00d0
      zcs = - Tramir

c  Borders of the difuser plate :
      e1 = -dmX/2.00d0 + dbX
      e2 =  dmX/2.00d0 - dbX
      e3 = -dmY/2.00d0 + dbY
      e4 =  dmY/2.00d0 - dbY

c  (in the global coordinate system)
      p30(1) = b11/2.00d0
      p30(2) = -b11/2.00d0
      p30(3) = Rmh - Tramir

      p31(1) = -b11/2.00d0
      p31(2) = -b11/2.00d0
      p31(3) = Rmh - Tramir

      p32(1) = p31(1)
```

```
      p32(2) = b11/2.00d0
      p32(3) = p30(3)

      p33(1) = p30(1)
      p33(2) = p32(2)
      p33(3) = p30(3)
c  Normal unit vector :
      dc32(1) = 0.00d0
      dc32(2) = 0.00d0
      dc32(3) = -1.00d0
c  Normal unit vector in matrix form :
      dcm(32,1) = dc32(1)
      dcm(32,2) = dc32(2)
      dcm(32,3) = dc32(3)
c  Subdivision :
      idir(32) = 10
      jdir(32) = 10
      iflag(32) = 1
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S32 :
      Call XMIMA (p31, p30, 32)
      cYmax(32) =  b11/2.00d0
      cYmin(32) = -b11/2.00d0
      omega(32) = datan2 (dc32(1), dc32(3))
      surare(32) = dist(p30, p31) * dist(p31, p32)
     *          - 105.00d0*pi*rd**2
c.....
c  Definition of the back side of the diffuser plate, S33 :
      Ngr(33) = 15
c  Groups of surfaces that S33 cannot see :
      see(33,15) = .false.
      see(33,2) = .false.
      see(33,5) = .false.
      see(33,6) = .false.
      see(33,16) = .false.
      see(33,17) = .false.
      see(33,18) = .false.
      see(33,19) = .false.
      see(33,20) = .false.
      see(33,21) = .false.
      see(33,22) = .false.
      see(33,7) = .false.
      see(33,11) = .false.
      see(33,12) = .false.
      see(33,13) = .false.
c  Normal unit vector :
      dc33(1) = -dc32(1)
      dc33(2) = -dc32(2)
      dc33(3) = -dc32(3)
c  Normal vector in matrix form :
      dcm(33,1) = dc33(1)
      dcm(33,2) = dc33(2)
      dcm(33,3) = dc33(3)
```

```
c Subdivision :
      idir(33) = 10
      jdir(33) = 10
      iflag(33) = 1
c Characteristics of the rectangular surface perpendicular to the
c x,z-plane, S33 :
      Call XMIMA (p31, p30, 33)
      cYmax(33) =  b11/2.00d0
      cYmin(33) = -b11/2.00d0
      omega(33) = datan2 (dc33(1), dc33(3))
     surare(33) = dist(p30, p31) * dist(p31, p32)
c.....
c Definition of the sperical surfaces, S37 :
c The surface S37 is considered as one unique surface containing 105
c mirrors.
      Ngr(37) = 15
c Groups of surfaces that S37 cannot see :
      see(37,15) = .false.
      see(37,9) = .false.
c Normal unit vector:
      dc37(1) = dc32(1)
      dc37(2) = dc32(2)
      dc37(3) = dc32(3)
c Normal unit vector in matrix form:
      dcm(37,1) = dc37(1)
      dcm(37,2) = dc37(2)
      dcm(37,3) = dc37(3)
c Frontal surface area:
      surare(37) = 105.00d0*pi*rd**2
c
c determination of the centers of the spherical mirrors for each cell:
c inner cells:

      Do m37 = 1, 5
       Do n37 = 1, 9
            xcin(1,m37,n37) = -2.50d0*dsX + (m37-1)*dsX
            ycin(1,m37,n37) = -4.50d0*dsY + (n37-1)*dsY
            xcin(2,m37,n37) =  xcin(1,m37,n37)
            ycin(2,m37,n37) =  ycin(1,m37,n37) + dsY
            xcin(3,m37,n37) =  xcin(1,m37,n37) + dsX/2.00d0
            ycin(3,m37,n37) =  ycin(1,m37,n37) + dsY/2.00d0
            xcin(4,m37,n37) =  xcin(1,m37,n37) + dsX
            ycin(4,m37,n37) =  ycin(1,m37,n37)
            xcin(5,m37,n37) =  xcin(1,m37,n37) + dsX
            ycin(5,m37,n37) =  ycin(1,m37,n37) + dsY
       Enddo
      Enddo
c
c top and bottom cells:

      Do n37 = 1, 9
        xcto(1,n37) = -2.50d0*dsX
        ycto(1,n37) = -4.50d0*dsY + (n37-1)*dsY
        xcto(2,n37) =  xcto(1,n37)
```

```
          ycto(2,n37) =  ycto(1,n37) + dsY
          xcbo(1,n37) =  2.50d0*dsX
          ycbo(1,n37) = -4.50*dsY + (n37-1)*dsY
          xcbo(2,n37) =  xcbo(1,n37)
          ycbo(2,n37) =  ycbo(1,n37) + dsY
        Enddo
c
c  Left and right cells:
        Do m37 = 1, 5
          xcle(1,m37) = -2.50d0*dsX + (m37-1)*dsX
          ycle(1,m37) =  4.50d0*dsY
          xcle(2,m37) =  xcle(1,m37) + dsX
          ycle(2,m37) =  ycle(1,m37)
          xcri(1,m37) =  -2.50d0*dsX + (m37-1)*dsX
          ycri(1,m37) =  -4.50d0*dsY
          xcri(2,m37) =   xcri(1,m37) + dsX
          ycri(2,m37) =   ycri(1,m37)
        Enddo
c
c  Corner cells:
c   top-left:
          xctl = -2.50d0*dsX
          yctl =  4.50d0*dsY
c   top-right:
          xctr = -2.50d0*dsX
          yctr = -4.50d0*dsY
c   bottom-left:
          xcbl =  2.50d0*dsX
          ycbl =  4.50d0*dsY
c  bottom-right:
          xcbr =  2.50d0*dsX
          ycbr = -4.50d0*dsY
c-------------------------------------------------------------------------------
c  GROUP number 16 :
c  The 2 sides of the baffle reflector number 1 belong to this group,
c  these 2 surfaces are S38, S39.
c.....
c  Definition of the top side of reflector number 1, S38 (44,45,46,47,
c                                                         48,49,50,51):
        Ngr(38) = 16
c  Groups of sufaces that S38 cannot see :
        see(38,16) = .false.
        see(38,17) = .false.
        see(38,18) = .false.
        see(38,19) = .false.
        see(38,20) = .false.
        see(38,21) = .false.
        see(38,22) = .false.
        see(38,7) = .false.
        see(38,8) = .false.
        see(38,9) = .false.
        see(38,10) = .false.
        see(38,11) = .false.
        see(38,12) = .false.
        see(38,13) = .false.
```

```
      see(38,15) = .false.
      see(38,14) = .false.
c
c Determine the dimensions of the reflectors in the y'-direction:
c  ( eta is the width in the x'-direction,
c  ( etay is the width in the y'-direction)
      Do i = 1, 7
        etay(i) = (b2/7.00d0) * dble(i) * dtan(a3)
      Enddo
c
c (in the local coordinate system)
      p1p(1) = b1/2.00d0 - b2/7.00d0*dsin(a3)
      p1p(2) = -b4/2.00d0
      p1p(3) = b2/7.00d0
      p2p(1) = -p1p(1)
      p2p(2) = p1p(2)
      p2p(3) = p1p(3)
      p3p(1) = p2p(1)
      p3p(2) = b4/2.00d0
      p3p(3) = p1p(3)
      p4p(1) = p1p(1)
      p4p(2) = p3p(2)
      p4p(3) = p1p(3)
      p5p(1) = p1p(1) - eta
      p5p(2) = p1p(2) + etay(1)
      p5p(3) = p1p(3)
      p6p(1) = p2p(1) + eta
      p6p(2) = p2p(2) + etay(1)
      p6p(3) = p2p(3)
      p7p(1) = p6p(1)
      p7p(2) = p3p(2) - etay(1)
      p7p(3) = p6p(3)
      p8p(1) = p5p(1)
      p8p(2) = p7p(2)
      p8p(3) = p7p(3)
c (in the global coordinate system)
      Call LOCGLO (p1p, p44, b6, a1)
      Call LOCGLO (p2p, p45, b6, a1)
      Call LOCGLO (p3p, p46, b6, a1)
      Call LOCGLO (p4p, p47, b6, a1)
      Call LOCGLO (p5p, p48, b6, a1)
      Call LOCGLO (p6p, p49, b6, a1)
      Call LOCGLO (p7p, p50, b6, a1)
      Call LOCGLO (p8p, p51, b6, a1)
c Normal unit vector :
      dc38(1) = -dc2(1)
      dc38(2) = -dc2(2)
      dc38(3) = -dc2(3)
c Normal unit vector in matrix form :
      dcm(38,1) = dc38(1)
      dcm(38,2) = dc38(2)
      dcm(38,3) = dc38(3)
c Subdivision :
```

```
      idir(38) = 5
      jdir(38) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S38 :
c    (for numbering)
      Call XMIMA (p45, p44, 38)
      Call rXmima (p48, p49, 38)
      rYmin(38) = p49(2)
      rYmax(38) = p50(2)
c    (for emission)
      omega(38) = datan2 (dc38(1), dc38(3))
      Call SURREF (p44,p45,p46,p48,p49,p50,
     *          ssa(38,1),ssa(38,2),ssa(38,3),ssa(38,3),surare(38))
c.....
c  Definition of the back side of reflector number 1, S39 (44,45,46,47,
c                                                          48,49,50,51):
      Ngr(39) = 16
c  Groups of surfaces that S39 cannot see :
      see(39,16) = .false.
      see(39,2) = .false.
      see(39,7) = .false.
      see(39,10) = .false.
      see(39,11) = .false.
      see(39,12) = .false.
      see(39,13) = .false.
c  Normal unit vector :
      dc39(1) = dc2(1)
      dc39(2) = dc2(2)
      dc39(3) = dc2(3)
c  Normal unit vector in matrix form :
      dcm(39,1) = dc39(1)
      dcm(39,2) = dc39(2)
      dcm(39,3) = dc39(3)
c  Subdivision :
      idir(39) = 5
      jdir(39) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S39 :
      Call XMIMA (p45, p44, 39)
      Call rXmima (p48, p49, 39)
      rYmin(39) = p49(2)
      rYmax(39) = p50(2)
      omega(39) = datan2 (dc39(1), dc39(3))
      Call SURREF (p44,p45,p46,p48,p49,p50,
     *          ssa(39,1),ssa(39,2),ssa(39,3),ssa(39,4),surare(39))
c-----
c  GROUP number 17 :
c  The 2 sides of the baffle reflector number 2 belong to this group,
c  these 2 surfaces are S40, S41.
c.....
c  Definition of the top side of reflector number 2, S40 (52,53,54,55,
c                                                          56,57,58,59):
      Ngr(40) = 17
c  Groups of surfaces that S40 cannot see :
      see(40,17) = .false.
      see(40,18) = .false.
```

```
      see(40,19) = .false.
      see(40,20) = .false.
      see(40,21) = .false.
      see(40,22) = .false.
      see(40,7) = .false.
      see(40,8) = .false.
      see(40,9) = .false.
      see(40,10) = .false.
      see(40,11) = .false.
      see(40,12) = .false.
      see(40,13) = .false.
      see(40,14) = .false.
      see(40,15) = .false.
c  (in the local coordinate system)
      p1p(1) = b1/2.00d0 - b2/7.00d0*2.00d0*dsin(a3)
      p1p(2) = -b4/2.00d0
      p1p(3) = b2/7.00d0*2.00d0
      p2p(1) = -p1p(1)
      p2p(2) = p1p(2)
      p2p(3) = p1p(3)
      p3p(1) = p2p(1)
      p3p(2) = b4/2.00d0
      p3p(3) = p1p(3)
      p4p(1) = p1p(1)
      p4p(2) = p3p(2)
      p4p(3) = p1p(3)
      p5p(1) = p1p(1) - eta
      p5p(2) = p1p(2) + etay(2)
      p5p(3) = p1p(3)
      p6p(1) = p2p(1) + eta
      p6p(2) = p2p(2) + etay(2)
      p6p(3) = p2p(3)
      p7p(1) = p6p(1)
      p7p(2) = p3p(2) - etay(2)
      p7p(3) = p6p(3)
      p8p(1) = p5p(1)
      p8p(2) = p7p(2)
      p8p(3) = p7p(3)
c  (in the global coordinate system)
      Call LOCGLO (p1p, p52, b6, a1)
      Call LOCGLO (p2p, p53, b6, a1)
      Call LOCGLO (p3p, p54, b6, a1)
      Call LOCGLO (p4p, p55, b6, a1)
      Call LOCGLO (p5p, p56, b6, a1)
      Call LOCGLO (p6p, p57, b6, a1)
      Call LOCGLO (p7p, p58, b6, a1)
      Call LOCGLO (p8p, p59, b6, a1)
c  Normal unit vector :
      dc40(1) = -dc2(1)
      dc40(2) = -dc2(2)
      dc40(3) = -dc2(3)
c  Normal unit vector in matrix form :
```

```
      dcm(40,1) = dc40(1)
      dcm(40,2) = dc40(2)
      dcm(40,3) = dc40(3)
c  Subdivision :
      idir(40) = 5
      jdir(40) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S40 :
      Call XMIMA (p53, p52, 40)
      Call rXmima (p56, p57, 40)
      rYmin(40) = p57(2)
      rYmax(40) = p58(2)
      omega(40) = datan2 (dc40(1), dc40(3))
      Call SURREF (p52,p53,p54,p56,p57,p58,
     *          ssa(40,1),ssa(40,2),ssa(40,3),ssa(40,4),surare(40))
c.....
c  Definition of the back side of reflector number 2, S41 (52,53,54,55,
c                                                        56,57,58,59):
      Ngr(41) = 17
c  Groups of surfaces that S41 cannot see :
      see(41,17) = .false.
      see(41,16) = .false.
      see(41,2) = .false.
      see(41,7) = .false.
      see(41,11) = .false.
      see(41,12) = .false.
      see(41,13) = .false.
c  Normal unit vector :
      dc41(1) = dc2(1)
      dc41(2) = dc2(2)
      dc41(3) = dc2(3)
c  Normal unit vector in matrix form :
      dcm(41,1) = dc41(1)
      dcm(41,2) = dc41(2)
      dcm(41,3) = dc41(3)
c  Subdivision :
      idir(41) = 5
      jdir(41) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S41 :
      Call XMIMA (p53, p52, 41)
      Call rXmima (p56, p57, 41)
      rYmin(41) = p57(2)
      rYmax(42) = p58(2)
      omega(41) = datan2 (dc41(1), dc41(3))
      Call SURREF (p52,p53,p54,p56,p57,p58,
     *          ssa(41,1),ssa(41,2),ssa(41,3),ssa(41,4),surare(41))
c-----
c  GROUP number 18 :
c  The 2 sides of the baffle reflector number 3 belong to this group,
c  these 2 surfaces are S42, S43.
c.....
c  Definition of the top side of reflector number 3, S42 (60,61,62,63,
c                                                        64,65,66,67):
      Ngr(42) = 18
```

```
c Groups of surfaces that S42 cannot see :
      see(42,18) = .false.
      see(42,19) = .false.
      see(42,20) = .false.
      see(42,21) = .false.
      see(42,22) = .false.
      see(42,7) = .false.
      see(42,8) = .false.
      see(42,9) = .false.
      see(42,10) = .false.
      see(42,11) = .false.
      see(42,12) = .false.
      see(42,13) = .false.
      see(42,14) = .false.
      see(42,15) = .false.
c (in the local coordinate system)
      p1p(1) = b1/2.00d0 - b2/7.00d0*3.00d0*dsin(a3)
      p1p(2) = -b4/2.00d0
      p1p(3) = b2/7.00d0*3.00d0
      p2p(1) = -p1p(1)
      p2p(2) = p1p(2)
      p2p(3) = p1p(3)
      p3p(1) = p2p(1)
      p3p(2) = b4/2.00d0
      p3p(3) = p1p(3)
      p4p(1) = p1p(1)
      p4p(2) = p3p(2)
      p4p(3) = p1p(3)
      p5p(1) = p1p(1) - eta
      p5p(2) = p1p(2) + etay(3)
      p5p(3) = p1p(3)
      p6p(1) = p2p(1) + eta
      p6p(2) = p2p(2) + etay(3)
      p6p(3) = p2p(3)
      p7p(1) = p6p(1)
      p7p(2) = p3p(2) - etay(3)
      p7p(3) = p6p(3)
      p8p(1) = p5p(1)
      p8p(2) = p7p(2)
      p8p(3) = p7p(3)
c (in the global coordinate system)
      Call LOCGLO (p1p, p60, b6, a1)
      Call LOCGLO (p2p, p61, b6, a1)
      Call LOCGLO (p3p, p62, b6, a1)
      Call LOCGLO (p4p, p63, b6, a1)
      Call LOCGLO (p5p, p64, b6, a1)
      Call LOCGLO (p6p, p65, b6, a1)
      Call LOCGLO (p7p, p66, b6, a1)
      Call LOCGLO (p8p, p67, b6, a1)
c Normal unit vector :
      dc42(1) = -dc2(1)
      dc42(2) = -dc2(2)
```

```
      dc42(3) = -dc2(3)
c  Normal unit vector in matrix form :
      dcm(42,1) = dc42(1)
      dcm(42,2) = dc42(2)
      dcm(42,3) = dc42(3)
c  Subdivision :
      idir(42) = 5
      jdir(42) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S42 :
      Call XMIMA (p61, p60, 42)
      Call rXmima (p64, p65,42)
      rYmin(42) = p65(2)
      rYmax(42) = p66(2)
      omega(42) = datan2 (dc42(1), dc42(3))
      Call SURREF (p60,p61,p62,p64,p65,p66,
     *          ssa(42,1),ssa(42,2),ssa(42,3),ssa(42,4),surare(42))
c.....
c  Definition of the back side of reflector number 3, S43 (60,61,62,63,
c                                                        64,65,66,67):
      Ngr(43) = 18
c  Groups of surfaces that S43 cannot see :
      see(43,18) = .false.
      see(43,16) = .false.
      see(43,17) = .false.
      see(43,2) = .false.
      see(43,7) = .false.
      see(43,11) = .false.
      see(43,12) = .false.
      see(43,13) = .false.
c  Normal unit vector :
      dc43(1) = dc2(1)
      dc43(2) = dc2(2)
      dc43(3) = dc2(3)
c  Normal unit vector in matrix form :
      dcm(43,1) = dc43(1)
      dcm(43,2) = dc43(2)
      dcm(43,3) = dc43(3)
c  Subdivision :
      idir(43) = 5
      jdir(43) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S43 :
      Call XMIMA (p61, p60, 43)
      Call rXmima (p64, p65, 43)
      rYmin(43) = p65(2)
      rYmax(43) = p66(2)
      omega(43) = datan2 (dc43(1), dc43(3))
      Call SURREF (p60,p61,p62,p64,p65,p66,
     *          ssa(43,1),ssa(43,2),ssa(43,3),ssa(43,4),surare(43))
c-----
c  GROUP number 19 :
c  The 2 sides of the baffle reflector number 4 belong to this group,
c  these 2 surfaces are S44, S45.
```

```
c.....
c  Definition of the top side of reflector number 4, S44 (68,69,70,71,
c                                                      72,73,74,75):
      Ngr(44) = 19
c  Groups of surfaces that S44 cannot see :
      see(44,19) = .false.
      see(44,20) = .false.
      see(44,21) = .false.
      see(44,22) = .false.
      see(44,7) = .false.
      see(44,8) = .false.
      see(44,9) = .false.
      see(44,10) = .false.
      see(44,11) = .false.
      see(44,12) = .false.
      see(44,13) = .false.
      see(44,14) = .false.
      see(44,15) = .false.
c  (in the local coordinate system)
      p1p(1) = b1/2.00d0 - b2/7.00d0*4.00d0*dsin(a3)
      p1p(2) = -b4/2.00d0
      p1p(3) = b2/7.00d0*4.00d0
      p2p(1) = -p1p(1)
      p2p(2) = p1p(2)
      p2p(3) = p1p(3)
      p3p(1) = p2p(1)
      p3p(2) = b4/2.00d0
      p3p(3) = p1p(3)
      p4p(1) = p1p(1)
      p4p(2) = p3p(2)
      p4p(3) = p1p(3)
      p5p(1) = p1p(1) - eta
      p5p(2) = p1p(2) + etay(4)
      p5p(3) = p1p(3)
      p6p(1) = p2p(1) + eta
      p6p(2) = p2p(2) + etay(4)
      p6p(3) = p2p(3)
      p7p(1) = p6p(1)
      p7p(2) = p3p(2) - etay(4)
      p7p(3) = p6p(3)
      p8p(1) = p5p(1)
      p8p(2) = p7p(2)
      p8p(3) = p7p(3)
c  (in the global coordinate system)
      Call LOCGLO (p1p, p68, b6, a1)
      Call LOCGLO (p2p, p69, b6, a1)
      Call LOCGLO (p3p, p70, b6, a1)
      Call LOCGLO (p4p, p71, b6, a1)
      Call LOCGLO (p5p, p72, b6, a1)
      Call LOCGLO (p6p, p73, b6, a1)
      Call LOCGLO (p7p, p74, b6, a1)
      Call LOCGLO (p8p, p75, b6, a1)
c  Normal unit vector :
```

```
      dc44(1) = -dc2(1)
      dc44(2) = -dc2(2)
      dc44(3) = -dc2(3)
c  Normal unit vector in matrix form :
      dcm(44,1) = dc44(1)
      dcm(44,2) = dc44(2)
      dcm(44,3) = dc44(3)
c  Subdivision :
      idir(44) = 5
      jdir(44) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S44 :
      Call XMIMA (p69, p68, 44)
      Call rXmima (p72, p73, 44)
      rYmin(44) = p73(2)
      rYmax(44) = p74(2)
      omega(44) = datan2 (dc44(1), dc44(3))
      Call SURREF (p68,p69,p70,p72,p73,p74,
     *             ssa(44,1),ssa(44,2),ssa(44,3),ssa(44,4),surare(44))
c.....
c  Definition of the back side of reflector number 4, S45 (68,69,70,71,
c                                                       72,73,74,75):
      Ngr(45) = 19
c  Groups of surfaces that S45 cannot see :
      see(45,19) = .false.
      see(45,16) = .false.
      see(45,17) = .false.
      see(45,18) = .false.
      see(45,2) = .false.
      see(45,7) = .false.
      see(45,11) = .false.
      see(45,12) = .false.
      see(45,13) = .false.
c  Normal unit vector :
      dc45(1) = dc2(1)
      dc45(2) = dc2(2)
      dc45(3) = dc2(3)
c  Normal unit vector in matrix form :
      dcm(45,1) = dc45(1)
      dcm(45,2) = dc45(2)
      dcm(45,3) = dc45(3)
c  Subdivision :
      idir(45) = 5
      jdir(45) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S45 :
      Call XMIMA (p69, p68, 45)
      Call rXmima (p72, p73, 45)
      rYmin(45) = p73(2)
      rYmax(45) = p74(2)
      omega(45) = datan2 (dc45(1), dc45(3))
      Call SURREF (p68,p69,p70,p72,p73,p74,
     *             ssa(45,1),ssa(45,2),ssa(45,3),ssa(45,4),surare(45))
```

```
c-----------------------------------------------------------------------
c GROUP number 20 :
c The 2 sides of the baffle reflector number 5 belong to this group,
c these 2 surfaces are S46, S47.
c.....
c Definition of the top side of reflector number 5, S46 (76,77,78,79,
c                                                       80,81,82,83):
      Ngr(46) = 20
c Groups of surfaces that S46 cannot see :
      see(46,20) = .false.
      see(46,21) = .false.
      see(46,22) = .false.
      see(46,7) = .false.
      see(46,8) = .false.
      see(46,9) = .false.
      see(46,10) = .false.
      see(46,11) = .false.
      see(46,12) = .false.
      see(46,13) = .false.
      see(46,14) = .false.
      see(46,15) = .false.
c (in the local coordinate system)
      p1p(1) = b1/2.00d0 - b2/7.00d0*5.00d0*dsin(a3)
      p1p(2) = -b4/2.00d0
      p1p(3) = b2/7.00d0*5.00d0
      p2p(1) = -p1p(1)
      p2p(2) = p1p(2)
      p2p(3) = p1p(3)
      p3p(1) = p2p(1)
      p3p(2) = b4/2.00d0
      p3p(3) = p1p(3)
      p4p(1) = p1p(1)
      p4p(2) = p3p(2)
      p4p(3) = p1p(3)
      p5p(1) = p1p(1) - eta
      p5p(2) = p1p(2) + etay(5)
      p5p(3) = p1p(3)
      p6p(1) = p2p(1) + eta
      p6p(2) = p2p(2) + etay(5)
      p6p(3) = p2p(3)
      p7p(1) = p6p(1)
      p7p(2) = p3p(2) - etay(5)
      p7p(3) = p6p(3)
      p8p(1) = p5p(1)
      p8p(2) = p7p(2)
      p8p(3) = p7p(3)
c (in the global coordinate system)
      Call LOCGLO (p1p, p76, b6, a1)
      Call LOCGLO (p2p, p77, b6, a1)
      Call LOCGLO (p3p, p78, b6, a1)
      Call LOCGLO (p4p, p79, b6, a1)
      Call LOCGLO (p5p, p80, b6, a1)
      Call LOCGLO (p6p, p81, b6, a1)
```

```
      Call LOCGLO (p7p, p82, b6, a1)
      Call LOCGLO (p8p, p83, b6, a1)
c Normal unit vector :
      dc46(1) = -dc2(1)
      dc46(2) = -dc2(2)
      dc46(3) = -dc2(3)
c Normal unit vector in matrix form :
      dcm(46,1) = dc46(1)
      dcm(46,2) = dc46(2)
      dcm(46,3) = dc46(3)
c Subdivision :
      idir(46) = 5
      jdir(46) = 2
c Characteristics of the rectangular surface perpendicular to the
c x,z-plane, S46 :
      Call XMIMA (p77, p76, 46)
      Call rXmima (p80, p81, 46)
      rYmin(46) = p81(2)
      rYmax(46) = p82(2)
      omega(46) = datan2 (dc46(1), dc46(3))
      Call SURREF (p76,p77,p78,p80,p81,p82,
     *          ssa(46,1),ssa(46,2),ssa(46,3),ssa(46,4),surare(46))
c.....
c Definition of the back side of reflector number 5, S47 (76,77,78,79,
c                                                        80,81,82,83):
      Ngr(47) = 20
c Groups of surfaces that S47 cannot see :
      see(47,20) = .false.
      see(47,16) = .false.
      see(47,17) = .false.
      see(47,18) = .false.
      see(47,19) = .false.
      see(47,2) = .false.
      see(47,7) = .false.
      see(47,11) = .false.
      see(47,12) = .false.
      see(47,13) = .false.
c Normal unit vector :
      dc47(1) = dc2(1)
      dc47(2) = dc2(2)
      dc47(3) = dc2(3)
c Normal unit vector in matrix form :
      dcm(47,1) = dc47(1)
      dcm(47,2) = dc47(2)
      dcm(47,3) = dc47(3)
c Subdivision :
      idir(47) = 5
      jdir(47) = 2
c Characteristics of the rectangular surface perpendicular to the
c x,z-plane, S47 :
      Call XMIMA (p77, p76, 47)
      Call rXmima (p80, p81, 47)
      rYmin(47) = p81(2)
```

```
      rYmax(47) = p82(2)
      omega(47) = datan2 (dc47(1), dc47(3))
      Call SURREF (p76,p77,p78,p80,p81,p82,
     *          ssa(47,1),ssa(47,2),ssa(47,3),ssa(47,4),surare(47))
c-------------------------------------------------------------------------------
c  GROUP number 21 :
c  The 2 sides of the baffle reflector number 6 belong to this group,
c  these 2 surfaces are S48, S49.
c.....
c  Definition of the top side of reflector number 6, S48 (84,85,86,87,
c                                                        88,89,90,91):
      Ngr(48) = 21
c  Groups of surfaces that S48 cannot see :
      see(48,21) = .false.
      see(48,22) = .false.
      see(48,7) = .false.
      see(48,8) = .false.
      see(48,9) = .false.
      see(48,10) = .false.
      see(48,11) = .false.
      see(48,12) = .false.
      see(48,13) = .false.
      see(48,14) = .false.
      see(48,15) = .false.
c  (in the local coordinate system)
      p1p(1) = b1/2.00d0 - b2/7.00d0*6.00d0*dsin(a3)
      p1p(2) = -b4/2.00d0
      p1p(3) = b2/7.00d0*6.00d0
      p2p(1) = -p1p(1)
      p2p(2) = p1p(2)
      p2p(3) = p1p(3)
      p3p(1) = p2p(1)
      p3p(2) = b4/2.00d0
      p3p(3) = p1p(3)
      p4p(1) = p1p(1)
      p4p(2) = p3p(2)
      p4p(3) = p1p(3)
      p5p(1) = p1p(1) - eta
      p5p(2) = p1p(2) + etay(6)
      p5p(3) = p1p(3)
      p6p(1) = p2p(1) + eta
      p6p(2) = p2p(2) + etay(6)
      p6p(3) = p2p(3)
      p7p(1) = p6p(1)
      p7p(2) = p3p(2) - etay(6)
      p7p(3) = p6p(3)
      p8p(1) = p5p(1)
      p8p(2) = p7p(2)
      p8p(3) = p7p(3)
c  (in the global coordinate system)
      Call LOCGLO (p1p, p84, b6, a1)
      Call LOCGLO (p2p, p85, b6, a1)
      Call LOCGLO (p3p, p86, b6, a1)
```

```
      Call LOCGLO (p4p, p87, b6, a1)
      Call LOCGLO (p5p, p88, b6, a1)
      Call LOCGLO (p6p, p89, b6, a1)
      Call LOCGLO (p7p, p90, b6, a1)
      Call LOCGLO (p8p, p91, b6, a1)
c  Normal unit vector :
      dc48(1) = -dc2(1)
      dc48(2) = -dc2(2)
      dc48(3) = -dc2(3)
c  Normal unit vector in matrix form :
      dcm(48,1) = dc48(1)
      dcm(48,2) = dc48(2)
      dcm(48,3) = dc48(3)
c  Subdivision :
      idir(48) = 5
      jdir(48) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S48 :
      Call XMIMA (p85, p84, 48)
      Call rXmima (p88, p89, 48)
      rYmin(48) = p89(2)
      rYmax(48) = p90(2)
      omega(48) = datan2 (dc48(1), dc48(3))
      Call SURREF (p84,p85,p86,p88,p89,p90,
     *          ssa(48,1),ssa(48,2),ssa(48,3),ssa(48,4),surare(48))
c.....
c  Definition of the back side of reflector number 6, S49 (84,85,86,87,
c                                                        88,89,90,91):
      Ngr(49) = 21
c  Groups of surfaces that S49 cannot see :
      see(49,21) = .false.
      see(49,16) = .false.
      see(49,17) = .false.
      see(49,18) = .false.
      see(49,19) = .false.
      see(49,20) = .false.
      see(49,2) = .false.
      see(49,7) = .false.
      see(49,11) = .false.
      see(49,12) = .false.
      see(49,13) = .false.
c  Normal unit vector :
      dc49(1) = dc2(1)
      dc49(2) = dc2(2)
      dc49(3) = dc2(3)
c  Normal unit vector in matrix form :
      dcm(49,1) = dc49(1)
      dcm(49,2) = dc49(2)
      dcm(49,3) = dc49(3)
c  Subdivision :
      idir(49) = 5
      jdir(49) = 2
c  Characteristics of the rectangular surface perpendicular to the
```

```
c x,z-plane, S49 :
      Call XMIMA (p85, p84, 49)
      Call rXmima (p88, p89, 49)
      rYmin(49) = p89(2)
      rYmax(49) = p90(2)
      omega(49) = datan2 (dc49(1), dc49(3))
      Call SURREF (p84,p85,p86,p88,p89,p90,
     *            ssa(49,1),ssa(49,2),ssa(49,3),ssa(49,4),surare(49))
c-------------------------------------------------------------------------------
c GROUP number 22 :
c The 2 sides of the baffle reflector number 7 belong to this group,
c these 2 surfaces are S50, S51.
c.....
c Definition of the top side of reflector number 7, S50 (92,93,94,95,
c                                                       96,97,98,99):
      Ngr(50) = 22
c Groups of surfaces that S50 cannot see :
      see(50,22) = .false.
      see(50,7) = .false.
      see(50,8) = .false.
      see(50,9) = .false.
      see(50,12) = .false.
      see(50,13) = .false.
      see(50,14) = .false.
      see(50,15) = .false.
c (in the local coordinate system)
      p1p(1) = b1/2.00d0 - b2/7.00d0*7.00d0*dsin(a3)
      p1p(2) = -b4/2.00d0
      p1p(3) = b2/7.00d0*7.00d0
      p2p(1) = -p1p(1)
      p2p(2) = p1p(2)
      p2p(3) = p1p(3)
      p3p(1) = p2p(1)
      p3p(2) = b4/2.00d0
      p3p(3) = p1p(3)
      p4p(1) = p1p(1)
      p4p(2) = p3p(2)
      p4p(3) = p1p(3)
      p5p(1) = p1p(1) - eta
      p5p(2) = p1p(2) + etay(7)
      p5p(3) = p1p(3)
      p6p(1) = p2p(1) + eta
      p6p(2) = p2p(2) + etay(7)
      p6p(3) = p2p(3)
      p7p(1) = p6p(1)
      p7p(2) = p3p(2) - etay(7)
      p7p(3) = p6p(3)
      p8p(1) = p5p(1)
      p8p(2) = p7p(2)
      p8p(3) = p7p(3)
c (in the global coordinate system)
      Call LOCGLO (p1p, p92, b6, a1)
      Call LOCGLO (p2p, p93, b6, a1)
```

```
      Call LOCGLO (p3p, p94, b6, a1)
      Call LOCGLO (p4p, p95, b6, a1)
      Call LOCGLO (p5p, p96, b6, a1)
      Call LOCGLO (p6p, p97, b6, a1)
      Call LOCGLO (p7p, p98, b6, a1)
      Call LOCGLO (p8p, p99, b6, a1)
c  Normal unit vector :
      dc50(1) = -dc2(1)
      dc50(2) = -dc2(2)
      dc50(3) = -dc2(3)
c  Normal unit vector in matrix form :
      dcm(50,1) = dc50(1)
      dcm(50,2) = dc50(2)
      dcm(50,3) = dc50(3)
c  Subdivision :
      idir(50) = 5
      jdir(50) = 2
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S50 :
      Call XMIMA (p93, p92, 50)
      Call rXmima (p96, p97, 50)
      rYmin(50) = p97(2)
      rYmax(50) = p98(2)
      omega(50) = datan2 (dc50(1), dc50(3))
      Call SURREF (p92,p93,p94,p96,p97,p98,
     *          ssa(50,1),ssa(50,2),ssa(50,3),ssa(50,4),surare(50))
c.....
c  Definition of the back side of reflector number 7, S51 (92,93,94,95,
c                                                       96,97,98,99):
      Ngr(51) = 22
c  Groups of surfaces that S51 cannot see :
      see(51,22) = .false.
      see(51,16) = .false.
      see(51,17) = .false.
      see(51,18) = .false.
      see(51,19) = .false.
      see(51,20) = .false.
      see(51,21) = .false.
      see(51,2) = .false.
      see(51,7) = .false.
      see(51,12) = .false.
      see(51,13) = .false.
c  Normal unit vector :
      dc51(1) = dc2(1)
      dc51(2) = dc2(2)
      dc51(3) = dc2(3)
c  Normal unit vector in matrix form :
      dcm(51,1) = dc51(1)
      dcm(51,2) = dc51(2)
      dcm(51,3) = dc51(3)
c  Subdivision :
      idir(51) = 5
      jdir(51) = 2
```

```
c  Characteristics of the rectangular surface perpendicular to the
c  x,z-plane, S51 :
      Call XMIMA (p93, p92, 51)
      Call rXmima (p96, p97, 51)
      rYmin(51) = p97(2)
      rYmax(51) = p98(2)
      omega(51) = datan2 (dc51(1), dc51(3))
      Call SURREF (p92,p93,p94,p96,p97,p98,
     *           ssa(51,1),ssa(51,2),ssa(51,3),ssa(51,4),surare(51))
 ******************************************************************************



c  Initialize the number of rays absorbed by the sensor-aperture:
      Do  i = 1, Nr*Nphi
         N29(i) = 0
      Enddo

c  Initialize the number of rays traced, rays that are lost, and the
c  number of rays that hit the mirrors:
      Ntrt  = 0
      Nlost = 0
      Nmir  = 0


******************************************************************************
*                          Emission
******************************************************************************
c  Initial number of rays absorbed by surface i:
      Do 3 i = 1, Nstot
         Do 4 j = 1, 800
            N(i,j) = 0
 4       Continue
 3    Continue

c  Initialize the number of possible incident surfaces :
      ntot = 0

c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++c
c        Start emission from surface Nse loop
c        (nse : number of the emitting surface)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++c
      Do 5 nse = 2, 2

c  If there is no emission on surfaces Nsj then :
*         if (nse.eq.Nsj1.or.nse.eq.Nsj2...) goto 5

      Write (6,300) nse
  300       Format (2X,'Emitting from Surface S(',i2,')')
c
c  Define the total number of rays emiited:
c
*         Ntre = surare(nse) * Nrepsa
c
c  Initialize the number of rays emitted from each sub-surface
```

```
c  if the emission is from one of the reflectors of the baffle:
c
      If (nse.ge.38.and.nse.le.51) then
        Nresar(1) = Ntre * ssa(nse,1) / surare(nse)
        Do i = 2, 4
          Nresar(i) = Nresar(i-1) + Ntre * ssa(nse,i) / surare(nse)
        Enddo
      Endif

c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++c
c          Start Emitting Ntre energy bundles loop
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++c
c     Ntre: Total number of rays emitted.

      Do 6 i = 1, Ntre

*         If ( mod(i,10000).eq.0 ) then
*           write (6,301) i
*         Endif
 301  Format (i12,2x,'rays emitted')


cxxx
```

```
          Call EMIT3 (p4, p9, p13, dc14(2))
        Else if (nse.eq.15) then
          Call EMIT3 (p13, p13, p20, dc15(2))
        Else if (nse.eq.16) then
          Call EMIT3 (p13, p20, p21, dc16(2))
        Else if (nse.eq.17) then
          Call EMIT3 (p13, p21, p26, dc17(2))
        Else if (nse.eq.18) then
          Call EMIT3 (p13, p26, p27, dc18(2))
        Else if (nse.eq.19) then
          Call EMIT3 (p13, p27, p28, dc19(2))
        Else if (nse.eq.20) then
          Call EMIT3 (p13, p28, p29, dc20(2))
        Else if (nse.eq.21) then
          Call EMIT1 (p2, dc21, 21, omega(21))
        Else if (nse.eq.22) then
          Call EMIT1 (p1, dc22, 22, omega(22))
        Else if (nse.eq.23) then
          Call EMIT1 (p11, dc23, 23, omega(23))
        Else if (nse.eq.24) then
          Call EMIT1 (p11, dc24, 24, omega(24))
        Else if (nse.eq.25) then
          Call EMIT1 (p18, dc25, 25, omega(25))
        Else if (nse.eq.26) then
          Call EMIT1 (p17, dc26, 26, omega(26))
        Else if (nse.eq.27) then
          Call EMIT1 (p22, dc27, 27, omega(27))
        Else if (nse.eq.28) then
          write (*,*)'this emission is not done yet!'
          Stop
*               call EMIT7 (p24, dc28, 28, omega(28), rsa)
c         (S29 is the sensor aperture)
        Else if (nse.eq.30) then
          Call EMIT1 (p24, dc30, 30, omega(30))
        Else if (nse.eq.31) then
          Call EMIT1 (p25, dc31, 31, omega(31))
        Else if (nse.eq.32) then
          Write (6,*) 'This emission is not done yet!'
          Stop
        Else if (nse.eq.33) then
          Call EMIT1 (p30, dc33, 33, omega(33))
        Else if (nse.eq.34) then
          Call EMIT5 (p10, dc34, 34, omega(34))
c         (S35 is the MAM aperture)
        Else if (nse.eq.36) then
          Call EMIT5 (p10, dc36, 36, omega(36))
c         (S37 is the spherical mirrors)
        Else if (nse.eq.38) then
          Call EMIT6 (p44, dc38, omega(38), i,
     *                p44,p45,p46,p48,p49,p50)
        Else if (nse.eq.39) then
          Call EMIT6 (p44, dc39, omega(39), i,
```

```
*                    p44,p45,p46,p48,p49,p50)
      Else if (nse.eq.40) then
        Call EMIT6 (p52, dc40, omega(40), i,
*                    p52,p53,p54,p56,p57,p58)
      Else if (nse.eq.41) then
        Call EMIT6 (p52, dc41, omega(41), i,
*                    p52,p53,p54,p56,p57,p58)
      Else if (nse.eq.42) then
        Call EMIT6 (p60, dc42, omega(42), i,
*                    p60,p61,p62,p64,p65,p66)
      Else if (nse.eq.43) then
        Call EMIT6 (p60, dc43, omega(43), i,
*                    p60,p61,p62,p64,p65,p66)
      Else if (nse.eq.44) then
        Call EMIT6 (p68, dc44, omega(44), i,
*                    p68,p69,p70,p72,p73,p74)
      Else if (nse.eq.45) then
        Call EMIT6 (p68, dc45, omega(45), i,
*                    p68,p69,p70,p72,p73,p74)
      Else if (nse.eq.46) then
        Call EMIT6 (p76, dc46, omega(46), i,
*                    p76,p77,p78,p80,p81,p82)
      Else if (nse.eq.47) then
        Call EMIT6 (p76, dc47, omega(47), i,
*                    p76,p77,p78,p80,p81,p82)
      Else if (nse.eq.48) then
        Call EMIT6 (p84, dc48, omega(48), i,
*                    p84,p85,p86,p88,p89,p90)
      Else if (nse.eq.49) then
        Call EMIT6 (p84, dc49, omega(49), i,
*                    p84,p85,p86,p88,p89,p90)
      Else if (nse.eq.50) then
        Call EMIT6 (p92, dc50, omega(50), i,
*                    p92,p93,p94,p96,p97,p98)
      Else if (nse.eq.51) then
        Call EMIT6 (p92, dc51, omega(51), i,
*                    p92,p93,p94,p96,p97,p98)
      Endif


c  Initialize the surface of reflection as the surface of emission :
c       (Ns : Number of the surface of reflection)
      Ns = Nse

*************************************************************************
*  If there is reflection, the point of incidence becomes the point
*  of emission, the program retraces the ray from this point.
*************************************************************************
   7  Continue

c  Increment the number of ray traced:
```

```
      Ntrt = Ntrt + 1

c  By default the distance between the point of emission and the point
c  of intersection is the longest dimension of the enclosure :
*          Dis(1) = (b2+b7+b5)**2

c  Initially there is no intersection.
      Ninter = 0
      Hit = .false.

*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
*         Find the points of intersection with the enclosure
*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
c  The enclosure is divided into groups, each containing the surfaces
c  that are lying on the same plane.
c     Group 1 : S1 (imaginary surface).
c     Group 2 : S2.
c     Group 3 : S3.
c     Group 4 : S20, S21, S4,...

c-------------------------------------------------------------------
c     Find the point of intersection with group 2 {s2}:
c     (note : s2 has only one side)
c-------------------------------------------------------------------
c     A ray emitted or reflected on surface Ns can see the surfaces
c     in group 2:
      If (see(ns,2)) then
c
c        Define the point of intersection between the ray and the plane
c        containing the surface S2:
         Call PRSURF (p1, dc2)
c
c        Check if the point of intersection is in front of the emitting
c        surface:
         Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
         If (FRONT) then
c
c           Check if the point of intersection is inside S2:
            Call PRCHEC (x2, y2, 2, inside)
            If (INSIDE) then
c
c              the point is inside the surface S2, so record the
c              coordinates of this point as a candidate for the point
c              of intersection:
               Call RECXYZ (2)
            Endif
         Endif
      Endif
c-------------------------------------------------------------------------------
c  Find the point of intersection with group number 3, which has 9 surf-
c  aces {s3,s4,s5,s6,s7,s8,s9,s10,s11}.
c-------------------------------------------------------------------------------
c     A ray emitted or reflected on surface (ns) can see the surfaces
c     in group 3 :
      If (see(ns,3)) then
c        (1)
c           Define the point of intersection with the plane containing
c           the surfaces of group 3 :
            Call XZPLAN (p1(2))
```

```
c
c       Check if the hit point is in front of the emitting point :
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),front)
        If (FRONT) then
c       (2)
c          Check if the point is inside the first element of group 3,
c          which is S3. S3 is now divided into 2 triangular surfaces
c          defined by (p5,p11,p10, and p5,p6,p11)
           Call TRISUR (p5, p11, p10, x2,z2, INSIDE)
           If (INSIDE) then
c             the point is inside the first half of S3, so record
c             this point of intersection, and there is no need to look
c             for the intersection with the other elements of the group:
              Call RECXYZ (3)
              goto 20
           Else
c             The hit point is not on the first half of S3, so continue
c             to check with the second half.
              Call TRISUR (p5, p6, p11, x2, z2, INSIDE)
              If (INSIDE) then
                 Call RECXYZ (3)
                 Goto 20
              Endif
           Endif
c
c       The hit point is not on the first element, so I continue
c       to look for it on the second element, which is S4.
        Call TRISUR (p6, p2, p11, x2, z2, inside)
        If (INSIDE) then
           Call RECXYZ (4)
           Goto 20
        Endif

c     Intersection with the third element of group 3, S5 :
        Call TRISUR (p1, p5, p10, x2, z2, INSIDE)
        If (INSIDE) then
           Call RECXYZ (5)
           Goto 20
        Endif

c     Intersection with the fourth element of group 3, S6 :
        Call TRISUR (p10, p14, p18, x2, z2, INSIDE)
        If (INSIDE) then
           Call RECXYZ (6)
           Goto 20
        Endif

c       Intersection with the fifth element of group 3, S7 :
        Call TRISUR (p10, p18, p17, x2, z2, INSIDE)
        If (INSIDE) then
           Call RECXYZ (7)
           Goto 20
        Endif
```

```
c       Intersection with the sixth element of group 3, S8 :
        Call TRISUR (p10, p17, p22, x2, z2, INSIDE)
        If (INSIDE) then
          Call RECXYZ (8)
          Goto 20
        Endif

c       Intersection with the 7th element of group 3, S9 :
        Call TRISUR (p10, p22, p23, x2, z2, INSIDE)
        If (INSIDE) then
          Call RECXYZ (9)
          Goto 20
        Endif

c       Intersection with the 8th element of group 3, S10 :
        Call TRISUR (p10, p23, p24, x2, z2, INSIDE)
        If (INSIDE) then
          Call RECXYZ (10)
          Goto 20
        Endif

c       Intersection with the 9th element of group 3, S11 :
        Call TRISUR (p10, p24, p25, x2, z2, INSIDE)
        If (INSIDE) then
          Call RECXYZ (11)
          Goto 20
        Endif

      Endif
c     (2)
     Endif
c    (1)
 20   Continue


c-------------------------------------------------------------------------------
c  Find the point of intersection with group number 4, which has 9 surf-
c  aces {s13,s12,s14,s15,s16,s17,s18,s19,s20}.
c-------------------------------------------------------------------------------
c    A ray emitted or reflected on surface (ns) can see the surfaces
c    in group 4 :
      If (see(ns,4)) then
c     (1)
c       Define the point of intersection with the plane containing
c       the surfaces of group 4 :
        Call XZPLAN (p3(2))
c
c       Check if the hit point is in front of the emitting point :
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),front)
        If (FRONT) then
c       (2)
c         Check if the hit point is inside the first element of
c         group 4, which is S13. S13 is now subdivided into 2
c         triangular surfaces defined by (p9,p8,p12, and p9,p12,p13).
```

```
      Call TRISUR (p9, p8, p12, x2, z2, INSIDE)
      If (INSIDE) then
c       The ray is inside S13, so record the point of intersection
c       and there is no need to look for intersection with the
c       other elements of the group.
        Call RECXYZ (13)
        Goto 21
      Else
c       The hit point is not on the first half of S13, check
c       again with the second half.
        Call TRISUR (p9, p12, p13, x2, z2, INSIDE)
        If (INSIDE) then
          Call RECXYZ (13)
          Goto 21
        Endif
      Endif

c     The hit point is not on the first element, so I continue
c     to look for it on the second element, which is S12.
      Call TRISUR (p8, p3, p12, x2, z2, INSIDE)
      If (INSIDE) then
        Call RECXYZ (12)
        Goto 21
      Endif

c     Intersection with the third element of group 4, S14 :
      Call TRISUR (p4, p9, p13, x2, z2, INSIDE)
      If (INSIDE) then
        Call RECXYZ (14)
        Goto 21
      Endif

c     Intersection with the fourth element of group 4, S15 :
      Call TRISUR (p13, p16, p20, x2, z2, INSIDE)
      If (INSIDE) then
        Call RECXYZ (15)
        Goto 21
      Endif

c     Intersection with the fifth element of group 4, S16 :
      Call TRISUR (p13, p20, p21, x2, z2, INSIDE)
      If (INSIDE) then
        Call RECXYZ (16)
        Goto 21
      Endif

c     Intersection with the sixth element of group 4, S17 :
      Call TRISUR (p13, p21, p26, x2, z2, INSIDE)
      If (INSIDE) then
        Call RECXYZ (17)
        Goto 21
      Endif
```

```
c       Intersection with the 7th element of group 4, S18 :
        Call TRISUR (p13, p26, p27, x2, z2, INSIDE)
        If (INSIDE) then
          Call RECXYZ (18)
          Goto 21
        Endif

c       Intersection with the 8th element of group 4, S19 :
        Call TRISUR (p13, p27, p28, x2, z2, INSIDE)
        If (INSIDE) then
          Call RECXYZ (19)
          Goto 21
        Endif

c       Intersection with the 9th element of group 4, S20 :
        Call TRISUR (p13, p28, p29, x2, z2, INSIDE)
        If (INSIDE) then
          Call RECXYZ (20)
          Goto 21
        Endif

      Endif
c     (2)
      Endif
c     (1)
 21   Continue

c-------------------------------------------------------------------
c     Find the point of intersection with group 5 {s21}:
c     (note : S21 has only one side)
c-------------------------------------------------------------------
c     A ray emitted or reflected on surface Ns can see the surfaces
c     in group 5 :
      if (see(ns,5)) then
c
c         Define the point of intersection between the ray and the plane
c         containing the surface S21 :
          Call PRSURF (p2, dc21)
c
c          check if the possible hit point is in front of the emitting
c          surface :
          Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),front)
          If (FRONT) then
c
c             Check if the point of intersection is inside S21 :
*             Call PRCHEC (x2, y2, 21, inside)
            Call PPXZ (y2, z2, p2(3), p11(3), 21, INSIDE)
            If (INSIDE) then
c
c               the point is inside the surface S21, so record the
c               coordinates of this point as a candidate for the point
c               of intersection :
              Call RECXYZ (21)
            Endif
          Endif
        Endif
```

```
c-------------------------------------------------------------------------------
c    Find the point of intersection with group 6 {s22, s31}
c    (group 6 is a 2 sided surface with different surface areas,
c     use logical combination 1 (front, face, inside) to determine the
c     the intersection)
c    (note : S22, s23 are the two sides of the same wall)
c-------------------------------------------------------------------------------
c    A ray emitted or reflected on surface Ns can see the surfaces
c    in group 6 :
      If (see(ns,6)) then
c      (1)
c        Define the point of intersection between the ray and the PLANE
c        containing the surface S22 :
         Call PRSURF (p1 , dc22)

c        Check if the possible hit point is in front of the
c        emitting surface :
         Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
         If (FRONT) then
c         (2)
c           Check if the first element of group 2 is facing the source
c           surface :
            Call SFACE (dc22, FACE)
            if (FACE) then
c            (3)
c              Check if the point of intersection is inside the
c              surface S22 :
               Call PRCHEC (x2, y2, 22, INSIDE)
               If (INSIDE) then
c                (4)
c                  The ray is inside S22, so record the coordinates of
c                  this point as candidate for the point of intersection.
                  Call RECXYZ (22)
                  Goto 22
               Endif
c              (4)
            Else
c           (3)
c              If the ray cannot see the top side (s22), then it must see
c              the bottom side (s31), so here I check if the hit point
c              is inside the surface s31:
               Call PRCHEC (x2, y2, 31, INSIDE)
               If (INSIDE) then
c                (5)
                  Call RECXYZ (31)
               Endif
c              (5)
            Endif
c           (3)
         Endif
c        (2)
      Endif
c     (1)
 22   Continue

c-------------------------------------------------------------------------------
c    Find the point of intersection with group 7 {23}:
```

```
c    (note : surfaces in this group have only one side)
c-----------------------------------------------------------------------
c    A ray emitted or reflected on surface Ns can see the surfaces
c    in group 7:
      If (see(ns,7)) then
c
c         Define the point of intersection between the ray and the plane
c         containing the surface S23:
         Call PRSURF (p11, dc23)
c
c         Check if the point of intersection is in front of the emitting
c         surface:
         Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
         If (FRONT) then
c
c            Check if the point of intersection is inside S23 :
           Call PRCHEC (x2, y2, 2, inside)
           If (INSIDE) then
c
c              the point is inside the surface S23, so record the
c              coordinates of this point as a candidate for the point
c              of intersection:
             Call RECXYZ (23)
           Endif
         Endif
      Endif

c-----------------------------------------------------------------------
c    Find the point of intersection with group 8 {24}:
c    (note : S24 has only one side)
c-----------------------------------------------------------------------
c    A ray emitted or reflected on surface Ns can see the surfaces
c    in group 8:
      If (see(ns,8)) then
c
c         Define the point of intersection between the ray and the PLANE
c         containing the surface S24:
         Call PRSURF (p14 , dc24)
c
c         Check if the ray can hit the top side of group 8 :
         Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),front)
         If (FRONT) then
c
c            Check if the point of intersection is inside
c            the surface S24 :
           Call PRCHEC (x2, y2, 24, INSIDE)
           If (INSIDE) then
c
c              The ray is inside S24, so record the coordinates
c              of this point as a candidate for the point of
c              intersection.
             Call RECXYZ (24)
           Endif
         Endif
      Endif

c-----------------------------------------------------------------------
c    Find the point of intersection with group 9 {s25: closeout plate}.
c    (note: S25 has only one side)
```

```
c-------------------------------------------------------------------------
c   A ray emitted or reflected on surface Ns can see the surfaces
c   in group 9:
     If (see(ns,9)) then
c
c        Define the point of intersection between the ray and the plane
c        containing the surface S25:
        Call PRSURF (p18 , dc25)
c
c        Check if the ray can hit the top side of group 9 :
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),front)
        If (FRONT) then
c
c          Check if the point of intersection is inside the surface S25
          Call PRCHEC (x2, y2, 25, INSIDE)
          If (INSIDE) then
c            The ray is inside S25, so record the coordinates of this
c            point as a candidate for the point of intersection.
            Call RECXYZ (25)
          Endif
        Endif
    Endif
c-------------------------------------------------------------------------
c   Find the point of intersection with group 10 {26}.
c   (note: s26 has only one side)
c-------------------------------------------------------------------------
c   A ray emitted or reflected on surface Ns can see the surfaces
c   in group 10:
     if (see(ns,10)) then
c
c        Define the point of intersection between the ray and the plane
c        containing the surface S26:
        Call PRSURF (p17 , dc26)
c
c        Check if the hit point is in front of the emitting surface.
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),front)
        If (FRONT) then
c
c          Check if the point of intersection is inside the surface S26.
          Call PRCHEC (x2, y2, 26, INSIDE)
          If (INSIDE) then
c
c            The ray is inside S26, so record the coordinates of this
c            point as a candidate for the point of intersection.
            Call RECXYZ (26)
          Endif
        Endif
     Endif
*-------------------------------------------------------------------------
*   Find the point of intersection with group 11 {27}.
c   (note: this group has only one side)
c-------------------------------------------------------------------------
c   A ray emitted or reflected on surface Ns can see the surfaces
c   in group 11:
     If (see(ns,11)) then
c
c        Define the point of intersection between the ray and the PLANE
c        containing the surface S27:
        Call PRSURF (p22 , dc27)
```

```
c
c        Check if the hit point is in front of the emitting surface.
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
        if (FRONT) then
c
c           check if the hit point is inside s27.
          Call PRCHEC (x2, y2, 27, INSIDE)
          If (INSIDE) then
c
c             The ray is inside S27, so record the coordinates of this
c             point as a candidate for the point of intersection.
            Call RECXYZ (27)
          Endif
        Endif
      Endif
c-----------------------------------------------------------------------
c  Find the point of intersection with group 12, this group consists of
c  the sensor aperture (S29) and the sensor wall (S28). The two surfaces
c  are in the same direction and lying on the same plane.
c-----------------------------------------------------------------------
c    A ray emitted or reflected on surface Ns can see the surfaces
c    in group 12:
      if (see(ns,12)) then
c      (1)
c        Define the hit point on the plane containing S28 :
        Call PRSURF (p24, dc28)
c
c         check if the hit point is in front of the emitting surface.
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
        if (FRONT) then
c         (2)
c           check if the hit point is inside the surface S28 :
          Call PRCHEC (x2, y2, 28, INSIDE)
          If (INSIDE) then
c           (3)
c             The ray is inside the sensor wall, then check again if the
c             hit point is inside the sensor aperture (s29) :
            Call SENSOR (x2, y2, z2, a2, rsa, APER)
            If (APER) then
c             (4)
c               the hit point is on the sensor aperture :
              Call RECXYZ (29)
            Else
c             (4)
c               The hit point is on the sensor wall :
              Call RECXYZ (28)
            Endif
c           (4)
          Endif
c         (3)
        Endif
c       (2)
      Endif
c     (1)
c-----------------------------------------------------------------------
c  Find the point of intersection with group 13 {s30: bottom wall}.
c-----------------------------------------------------------------------
c    A ray emitted or reflected on surface Ns can see the surfaces
```

```
c    in group 13:
      If (see(ns,13)) then
c
c        Define the point of intersection between the ray and the PLANE
c        containing the surface S30 :
        Call PRSURF (p24 , dc30)
c
c        check if the hit point is in front of the emitting surface.
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
        If (FRONT) then
c
c           Check if the point of intersection is inside the surface S30.
          Call PRCHEC (x2, y2, 30, INSIDE)
          If (INSIDE) then
c
c             The ray is inside S30, so record the coordinates of this
c             point as candidate for the point of intersection.
            Call RECXYZ (30)
          Endif
        Endif
      Endif
*-------------------------------------------------------------------------------
*    Find the point of intersection with group 14 (internal wall)
*   { S34: top side, S36: bottom side, S35: MAM aperture }
*-------------------------------------------------------------------------------
c    A ray emitted or reflected on surface Ns can see the surfaces
c    in group 14 :
      If (see(ns,14)) then
c     (1)
c        Define the point of intersection between the ray and the PLANE
c        containing the surface S34 :
        Call PRSURF (p10 , dc34)
c
c        Check if the possible hit point is in front of the emitting
c        surface Ns.
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
        If (FRONT) then
c        (2)
c           Check if the point of intersection is inside the internal
c           wall:
          Call PRCHEC (x2, y2, 34, inside)
          If (INSIDE) then
c           (3)
c              the ray is inside the wall, check again if it can pass
c              through the aperture :
            Call APERTU
            If (.not.APER) then
c             (4)
c                the ray is on the wall, check if it can hit the top
c                side S34:
              Call SFACE (dc34, face)
              If (FACE) then
c               (5)
                Call RECXYZ (34)
              Else
c               (5)
c                  the ray is on the wall and is not on the top, so
c                  it must hit the bottom (s36)
```

```
                Call RECXYZ (36)
               Endif
c               (5)
              Endif
c              (4)
            Endif
c            (3)
          Endif
c          (2)
        Endif
c       (1)
*-------------------------------------------------------------------------------
* Find the point of intersection with the diffuser plate, the surfaces
* of this plate are in group 15 which consists of 3 elements :
*          1- Top side of the plate .....: S32
*          2- Back side of the plate .....: S33
*          3- The 105 spherical surfaces are
*            considered as one unique element : S37
*-------------------------------------------------------------------------------
c   A ray emitted or reflected on surface (Ns) can see the surfaces
c   in group 15:
     if (see(ns,15)) then
c     (1)
c        Define the point of intersection between the ray and the plane
c        containing the surface S32:
        Call PRSURF (p30 , dc32)
c
c         Check if the possible point of intersection is in front of the
c         emitting surface:
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
        If (FRONT) then
c        (2)
c           Check if the hit point is inside the diffuser plate.
          Call PRCHEC (x2, y2, 32, inside)
          If (INSIDE) then
c           (3)
c              Check if the top of the diffuser plate (S32) is facing the
c              source surface :
             Call SFACE (dc32, FACE)
             If (FACE) then
c              (4)
c                The ray is on the top side of the plate, check again
c                if it can hit one of the spherical mirrors (s37):
               Call MIRCHE (x2, y2)
               If (INCIRC) then
c                (5)
c                   the ray hits a spherical mirror, determine the
c                   coordinates of the point of incidence on the
c                   hemispherical mirror and record this point.
                 Call HEMIS
                 Nmir = Nmir + 1
                 Call RECXYZ (37)
               Else
c                (5)
c                   the ray hits the top side of the diffuser plate,
c                   which is S32:
                  Call RECXYZ (32)
               Endif
```

```
c                (5)
                Else
c               (4)
c                  The ray is inside the plate but not on the top side,
c                  so it is on the back of the diffuser plate S33.
                  Call RECXYZ (33)
                Endif
c               (4)
              Endif
c             (3)
            Endif
c           (2)
          Endif
c         (1)
*-------------------------------------------------------------------------------
* Find the point of intersection with the reflector number 1, the
* surfaces of this reflector are in group number 16, which consists
* of 2 elements:
*               1- Front side: S38
*               2- Back side : S39
*    (intersection combination number 3: Front, Inside, Face)
*-------------------------------------------------------------------------------
c    A ray emitted or reflected on surface (Ns) can see the surfaces
c    in group 16:
      If (see(ns,16)) then
c     (1)
c        Define the point of intersection between the ray and the plane
c        containing the surface S38:
         Call PRSURF (p44 , dc38)
c
c         Check if the surfaces of group 16 are in front of the source
c         surface :
         Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
         If (FRONT) then
c        (2)
c           The hit point is inside the ring ?
           Call RING (x2, y2, 38, INSIDE)
           If (INSIDE) then
c           (3)
c              Is the hit point on the top side of the ring ?
              Call SFACE (dc38, FACE)
              If (FACE) then
c             (4)
                Call RECXYZ (38)
              Else
c             (4)
c                The ray is on the ring and it is not on the top
c                side, so it is on the back side !
                Call RECXYZ (39)
              Endif
c             (4)
            Endif
c           (3)
          Endif
c         (2)
        Endif
c       (1)
*-------------------------------------------------------------------------------
```

```
*  Find the point of intersection with the reflector number 2, the
*  surfaces of this reflector are in group number 17, which consists
*  of 2 elements :
*               1- Front side : S40
*               2- Back side  : S41
*     (intersection combination number 3 : Front, Inside, Face)
*-------------------------------------------------------------------------------
c    A ray emitted or reflected on surface (Ns) can see the surfaces
c    in group 17:
      If (see(ns,17)) then
c      (1)
c         Define the point of intersection between the ray and the PLANE
c         containing the surface S40 :
        Call PRSURF (p52 , dc40)
c
c         Check if the surfaces of group 17 are in front of the source
c         surface :
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
        If (FRONT) then
c         (2)
c            The hit point is inside the ring ?
          Call RING (x2, y2, 40, INSIDE)
          If (INSIDE) then
c           (3)
c              Is the hit point on the top side of the ring ?
            Call SFACE (dc40, FACE)
            If (FACE) then
c             (4)
                Call RECXYZ (40)
            Else
c             (4)
c                The ray is on the ring and it is not on the top
c                side, so it is on the back side !
              Call RECXYZ (41)
            Endif
c           (4)
          Endif
c         (3)
        Endif
c       (2)
      Endif
c     (1)

*-------------------------------------------------------------------------------
*  Find the point of intersection with the reflector number 3, the
*  surfaces of this reflector are in group number 18, which consists
*  of 2 elements :
*               1- Front side : S42
*               2- Back side  : S43
*     (intersection combination number 3 : Front, Inside, Face)
*-------------------------------------------------------------------------------
c    A ray emitted or reflected on surface (Ns) can see the surfaces
c    in group 18:
      If (see(ns,18)) then
c     (1)
c         Define the point of intersection between the ray and the plane
c         containing the surface S42:
        Call PRSURF (p60 , dc42)
c
```

```
c         Check if the surfaces of group 18 are in front of the source
c         surface :
          Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
          If (FRONT) then
c         (2)
c           The hit point is inside the ring ?
            Call RING (x2, y2, 42, INSIDE)
            If (INSIDE) then
c           (3)
c             Is the hit point on the top side of the ring ?
              Call SFACE (dc42, FACE)
              If (FACE) then
c             (4)
                Call RECXYZ (42)
              Else
c             (4)
c               The ray is on the ring and it is not on the top
c               side, so it is on the back side !
                Call RECXYZ (43)
              Endif
c             (4)
            Endif
c           (3)
          Endif
c         (2)
        Endif
c       (1)

*-------------------------------------------------------------------------------
* Find the point of intersection with the reflector number 4, the
* surfaces of this reflector are in group number 19, which consists
* of 2 elements :
*               1- Front side : S44
*               2- Back side  : S45
*    (intersection combination number 3 : Front, Inside, Face)
*-------------------------------------------------------------------------------
c     A ray emitted or reflected on surface (Ns) can see the surfaces
c     in group 19:
      If (see(ns,19)) then
c     (1)
c       Define the point of intersection between the ray and the PLANE
c       containing the surface S44 :
        Call PRSURF (p68 , dc44)
c
c       Check if the surfaces of group 19 are in front of the source
c       surface :
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
        If (FRONT) then
c       (2)
c         The hit point is inside the ring ?
          Call RING (x2, y2, 44, INSIDE)
          If (INSIDE) then
c           (3)
c             Is the hit point on the top side of the ring ?
            Call SFACE (dc44, FACE)
            If (FACE) then
c             (4)
```

```
                Call RECXYZ (44)
              Else
c             (4)
c                The ray is on the ring and it is not on the top
c                side, so it is on the back side !
                Call RECXYZ (45)
              Endif
c             (4)
            Endif
c           (3)
          Endif
c         (2)
        Endif
c       (1)

*-------------------------------------------------------------------------------
* Find the point of intersection with the reflector number 5, the
* surfaces of this reflector are in group number 20, which consists
* of 2 elements :
*              1- Front side : S46
*              2- Back side  : S47
*    (intersection combination number 3 : Front, Inside, Face)
*-------------------------------------------------------------------------------
c     A ray emitted or reflected on surface (Ns) can see the surfaces
c     in group 20:
      If (see(ns,20)) then
c       (1)
c         Define the point of intersection between the ray and the plane
c         containing the surface S46:
        Call PRSURF (p76 , dc46)
c
c         Check if the surfaces of group 20 are in front of the source
c         surface:
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
        If (FRONT) then
c         (2)
c           The hit point is inside the ring ?
          Call RING (x2, y2, 46, INSIDE)
          If (INSIDE) then
c           (3)
c               Is the hit point on the top side of the ring ?
            Call SFACE (dc46, FACE)
            If (FACE) then
c             (4)
              Call RECXYZ (46)
            Else
c             (4)
c               The ray is on the ring and it is not on the top
c               side, so it is on the back side !
              Call RECXYZ (47)
            Endif
c           (4)
          Endif
c         (3)
        Endif
c       (2)
```

```
      Endif
c     (1)

*-------------------------------------------------------------------------------
* Find the point of intersection with the reflector number 6, the
* surfaces of this reflector are in group number 21, which consists
* of 2 elements :
*               1- Front side : S48
*               2- Back side  : S49
*    (intersection combination number 3 : Front, Inside, Face)
*-------------------------------------------------------------------------------
c    A ray emitted or reflected on surface (Ns) can see the surfaces
c    in group 21:
      If (see(ns,21)) then
c     (1)
c        Define the point of intersection between the ray and the plane
c        containing the surface S48:
        Call PRSURF (p84 , dc48)
c
c        Check if the surfaces of group 21 are in front of the source
c        surface:
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
        If (FRONT) then
c       (2)
c          The hit point is inside the ring ?
          Call RING (x2, y2, 48, INSIDE)
          If (INSIDE) then
c           (3)
c             Is the hit point on the top side of the ring ?
            Call SFACE (dc48, FACE)
            If (FACE) then
c            (4)
              Call RECXYZ (48)
            Else
c             (4)
c               The ray is on the ring and it is not on the top
c               side, so it is on the back side !
              Call RECXYZ (49)
            Endif
c           (4)
          Endif
c         (3)
        Endif
c       (2)
      Endif
c     (1)

*-------------------------------------------------------------------------------
* Find the point of intersection with the reflector number 7, the
* surfaces of this reflector are in group number 22, which consists
* of 2 elements :
*               1- Front side : S50
*               2- Back side  : S51
*    (intersection combination number 3 : Front, Inside, Face)
*-------------------------------------------------------------------------------
c    A ray emitted or reflected on surface (Ns) can see the surfaces
c    in group 22:
```

```
      If (see(ns,22)) then
c     (1)
c       Define the point of intersection between the ray and the PLANE
c       containing the surface S50 :
        Call PRSURF (p92 , dc50)
c
c       Check if the surfaces of group 22 are in front of the source
c       surface :
        Call SFRONT (dcm(ns,1),dcm(ns,2),dcm(ns,3),FRONT)
        If (FRONT) then
c       (2)
c         The hit point is inside the ring ?
          Call RING (x2, y2, 50, INSIDE)
          If (INSIDE) then
c           (3)
c             Is the hit point on the top side of the ring ?
            Call SFACE (dc50, FACE)
            If (FACE) then
c             (4)
              Call RECXYZ (50)
            Else
c             (4)
c               The ray is on the ring and it is not on the top
c               side, so it is on the back side !
              Call RECXYZ (51)
            Endif
c           (4)
          Endif
c         (3)
        Endif
c       (2)
      Endif
c     (1)

*       If (i.eq.711207) then
*         Write (*,*)'end intersection'
*       Endif

*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
*             Find the shortest distance and                *
*       define the coordinates of THE point of incidence    *
*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
c
c  Find the shortest distance from the point of emission to all the
c  possible points of incidence, if there is at least one :
c
      If (HIT) then
c     (1)
        DisOld = Dis(1)
        Do 11 L = 1, Ninter
          If (Dis(L).LE.DisOld) then
c           (2)
            DisOld = Dis(L)
            Ns = Nsurf(L)
          Endif
c         (2)
```

```
 11    Continue

*              Write (6,*)'i,ns:'
*              Write (6,*)i,ns

c       Record the coordinates of THE point of intersection :
              x2 = Xcoor(ns)
              y2 = Ycoor(ns)
              z2 = Zcoor(ns)



*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
*              ABSORPTION / REFLECTION                                          *
*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
* The absorption is determined by a random number, Ra. If Ra is less
* than the emissivity of the surface of incidence, then the energy
* bundle is absorbed by this surface.

          Ra = RANDO ()

      If (Ra.LT.emis(ns)) then
c     (3)
c          c-------------------------------------------------------c
c          c              Absorption                               c
c          c-------------------------------------------------------c
c
c          On what area element is the point of incidence ?
c
        If (ns.EQ.2) then
c       (4)
*               Call NURAPE (x2,y2,cXmin(2),cXmax(2),cYmin(2),
*     *                   cYmax(2),idir(2),jdir(2),nsa)
          nsa = 1
        Else if (ns.EQ.3) then
          Call NURAPA (x2,z2,p5,p6,p10,idir(3),jdir(3),nsa)
        Else if (ns.EQ.4) then
          Call NUMTRA (x2,z2,p6,p2,p11,idir(4),jdir(4),nsa)
        Else if (ns.EQ.5) then
          Call NUMTRA (x2,z2,p5,p1,p10,idir(5),jdir(5),nsa)
        Else if (ns.EQ.6) then
          Call NUMQUA (x2,z2,p10,p14,p18,idir(6),nsa)
        Else if (ns.EQ.7) then
          Call NUMQUA (x2,z2,p17,p10,p18,idir(7),nsa)
        Else if (ns.EQ.8) then
          Call NUMQUA (x2,z2,p10,p17,p22,idir(8),nsa)
        Else if (ns.EQ.9) then
          Call NUMQUA (x2,z2,p10,p22,p23,idir(9),nsa)
        Else if (ns.EQ.10) then
          Call NUMQUA (x2,z2,p10,p23,p24,idir(10),nsa)
        Else if (ns.EQ.11) then
          Call NUMQUA (x2,z2,p10,p24,p25,idir(11),nsa)
        Else if (ns.EQ.12) then
          Call NUMTRA (x2,z2,p8,p3,p12,idir(12),jdir(12),nsa)
        Else if (ns.EQ.13) then
```

```
          Call NURAPA (x2,z2,p9,p8,p13,idir(13),jdir(13),nsa)
        Else if (ns.EQ.14) then
          Call NUMTRA (x2,z2,p9,p4,p13,idir(14),jdir(14),nsa)
        Else if (ns.EQ.15) then
          Call NUMQUA (x2,z2,p13,p16,p20,idir(15),nsa)
        Else if (ns.EQ.16) then
          Call NUMQUA (x2,z2,p21,p13,p20,idir(16),nsa)
        Else if (ns.EQ.17) then
          Call NUMQUA (x2,z2,p13,p21,p26,idir(17),nsa)
        Else if (ns.EQ.18) then
          Call NUMQUA (x2,z2,p13,p26,p27,idir(18),nsa)
        Else if (ns.EQ.19) then
          Call NUMQUA (x2,z2,p13,p27,p28,idir(19),nsa)
        Else if (ns.EQ.20) then
          Call NUMQUA (x2,z2,p13,p28,p29,idir(20),nsa)
        Else if (ns.EQ.21) then
*           Call NURAPE (x2,y2,cXmin(21),cXmax(21),cYmin(21),
*  *                  cYmax(21),idir(21),jdir(21),nsa)
          Call NURAP2 (x2,y2,p2(3),p11(3),cYmin(21),
     *                  cYmax(21),idir(21),jdir(21),nsa)
        Else if (ns.EQ.22) then
          Call NURAPE (x2,y2,cXmin(22),cXmax(22),cYmin(22),
     *                  cYmax(22),idir(22),jdir(22),nsa)
        Else if (ns.EQ.23) then
          Call NURAPE (x2,y2,cXmin(23),cXmax(23),cYmin(23),
     *                  cYmax(23),idir(23),jdir(23),nsa)
        Else if (ns.EQ.24) then
          Call NURAPE (x2,y2,cXmin(24),cXmax(24),cYmin(24),
     *                  cYmax(24),idir(24),jdir(24),nsa)
        Else if (ns.EQ.25) then
          Call NURAPE (x2,y2,cXmin(25),cXmax(25),cYmin(25),
     *                  cYmax(25),idir(25),jdir(25),nsa)
        Else if (ns.EQ.26) then
          Call NURAPE (x2,y2,cXmin(26),cXmax(26),cYmin(26),
     *                  cYmax(26),idir(26),jdir(26),nsa)
        Else if (ns.EQ.27) then
          Call NURAPE (x2,y2,cXmin(27),cXmax(27),cYmin(27),
     *                  cYmax(27),idir(27),jdir(27),nsa)
        Else if (ns.EQ.28) then
          Call NURAPE (x2,y2,cXmin(28),cXmax(28),cYmin(28),
     *                  cYmax(28),idir(28),jdir(28),nsa)
        Else if (ns.EQ.29) then
          Call NUMCIR (pi,x2,y2,z2,a2,Nr,Nphi,nsa)

c           if a ray is incident on the sensor aperture (S29),
c           I also increase the number of rays absorbed by the
c           sensor wall (S28) by 1, this allows me to have a
c           complete distribution of energy on the sensor wall.

          Call NURAPE (x2,y2,cXmin(28),cXmax(28),cYmin(28),
     *                  cYmax(28),idir(28),jdir(28),nsapr)
          n(28,nsapr) = n(28,nsapr) + 1
```

```
      Else if (ns.EQ.30) then
       Call NURAPE (x2,y2,cXmin(30) ,cXmax(30),cYmin(30),
     *              cYmax(30),idir(30),jdir(30),nsa)
      Else if (ns.EQ.31) then
       Call NURAPE (x2,y2,cXmin(31),cXmax(31),cYmin(31),
     *              cYmax(31),idir(31),jdir(31),nsa)
      Else if (ns.EQ.32) then
       Call NURAPE (x2,y2,cXmin(32),cXmax(32),cYmin(32),
     *              cYmax(32),idir(32),jdir(32),nsa)
      Else if (ns.EQ.33) then
       Call NURAPE (x2,y2,cXmin(33),cXmax(33),cYmin(33),
     *              cYmax(33),idir(33),jdir(33),nsa)
      Else if (ns.EQ.34) then
       Call NUMINT (x2,y2,z2,nsa)
c       (S35 is the MAM aperture)
      Else if (ns.EQ.36) then
       Call NUMINT (x2,y2,z2,nsa)
c       (S37 is the mirrors)
      Else if (ns.EQ.38) then
       Call NURING (x2,y2,38,idir(38),jdir(38),nsa)
      Else if (ns.EQ.39) then
       Call NURING (x2,y2,39,idir(39),jdir(39),nsa)
      Else if (ns.EQ.40) then
       Call NURING (x2,y2,40,idir(40),jdir(40),nsa)
      Else if (ns.EQ.41) then
       Call NURING (x2,y2,41,idir(41),jdir(41),nsa)
      Else if (ns.EQ.42) then
       Call NURING (x2,y2,42,idir(42),jdir(42),nsa)
      Else if (ns.EQ.43) then
       Call NURING (x2,y2,43,idir(43),jdir(43),nsa)
      Else if (ns.EQ.44) then
       Call NURING (x2,y2,44,idir(44),jdir(44),nsa)
      Else if (ns.EQ.45) then
       Call NURING (x2,y2,45,idir(45),jdir(45),nsa)
      Else if (ns.EQ.46) then
       Call NURING (x2,y2,46,idir(46),jdir(46),nsa)
      Else if (ns.EQ.47) then
       Call NURING (x2,y2,47,idir(47),jdir(47),nsa)
      Else if (ns.EQ.48) then
       Call NURING (x2,y2,48,idir(48),jdir(48),nsa)
      Else if (ns.EQ.49) then
       Call NURING (x2,y2,49,idir(49),jdir(49),nsa)
      Else if (ns.EQ.50) then
       Call NURING (x2,y2,50,idir(50),jdir(50),nsa)
      Else if (ns.EQ.51) then
       Call NURING (x2,y2,51,idir(51),jdir(51),nsa)
      Endif
c       (4)
c       Increment the number of rays absorbed by area element Nsa
c       of surface Ns :
        n(ns,nsa) = n(ns,nsa) + 1
```

```
c          Increment the total number of rays absorbed by the
c          enclosure :
           Ntot = Ntot + 1

      Else
c     (3)
c          The energy bundle is reflected,the point of incidence
c          becomes the point of emission:
           x = x2
           y = y2
           z = z2
c          *-------------------------------------------------------
c          *                  Reflection
c          *-------------------------------------------------------
c               How is the reflection ?
            Rrho = RANDO ( )
       If (Rrho.GE.Rhoref(ns)) then
c       (5)
c            *.....................................................
c            *            Diffuse reflection
c            *.....................................................
c            Define the direction of diffuse reflection:

           If (ns.GE.3.AND.ns.LE.20) then
c          (6)
             If (3.le.ns.and.ns.le.11) then
                dcy = 1.00d0
             Else
                dcy = -1.00d0
             Endif
             Call DIREC2 (dcy)
           Else if (ns.GE.21)  then
             Call DIREC1 (omega(ns))
           Endif
c          (6)

c          The energy bundle is diffusely reflected,
c          retrace it again.
           Goto 7

      Else
c     (5)
c          *.....................................................
c          *   Specular reflection ( Rrho < Rhoref(ns) )
c          *.....................................................
c          Define the direction of specular reflection from
c          the spherical mirrors:
           If (ns.eq.37) then
c          (7)
c            At this line, point 1 is already set
c            identical to point 2.
c
 24          Call REFLEC
c
c            The subroutine REFLEC uses the coordinates of
```

```
c          point 1 and the components of direction 1 to
c          determine the new direction 1.
c
           Call DISK (V)
c
c          The subroutine DISK uses the coordinates of
c          point 1 and the direction 1 to determine point 3,
c          if point 3 is outside the base of the hemispheri-
c          cal mirror (a circle of radius rd) then the reflec-
c          tion is within the mirror. Otherwise, the reflect-
c          ion is out to the enclosure.
c
           If (.not.V) then
c           (8)
c             Reflection within the mirror:
              Call MIRROR
c
c             The subroutine MIRROR uses the coordinates of
c             point 1 and the direction 1 to determine the
c             new coordinates of point 1.
c
              Goto 24
              Else
c              (8)
c                  Reflection out to the enclosure;
c                  Redefine the inward normal unit vector of the
c                  hemispherical mirror centered at xcs, ycs, zcs,
c                  the new site of emission is already
c                  redefined :
c
                  dc37(1) = (xcs-x) / rs
                  dc37(2) = (ycs-y) / rs
                  dc37(3) = (zcs-z) / rs
                  dcm(37,1) = dc37(1)
                  dcm(37,2) = dc37(2)
                  dcm(37,3) = dc37(3)

                  Goto 7
              Endif
c              (8)
           Else
c           (7)
              Call SPEREF (dcm(ns,1),dcm(ns,2),dcm(ns,3))
c
c             The subroutine SPEREF uses the direction 1 and
c             the normal unit vector of the incident surface
c             (ns) to redefine the direction 1.
c

c             The energy bundle is specularly reflected, retrace
c             it again.
              Goto 7
           Endif
c           (7)
        Endif
c       (5)
```

```
          Endif
c         (3)
      Else
c     (1)
c          c-------------------------------------------------------------
c          c              The energy bundle is lost
c          c-------------------------------------------------------------
c          Increment the number of rays that are lost by the ray-trace technique
c
          Nlost = Nlost + 1
*         Write (*,302) i
*         write (*,320) ns
*         stop
      Endif
c     (1)



c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c             End emitting energy bundles loop
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 6    Continue

c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c             End of emission from surface nse loop
c (place this end here to see the results of emission from
c  a group of surfaces)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 5    Continue



**********************************************************************
*                 Print out the results
**********************************************************************


c  Unit 7 :
c
      If (NF5.eq.1) then
        Write (7,303) nse-1, ntre
        Write (7,304) Ntot
        Write (7,305) Nlost
        Write (7,*)
        Write (7,*) '  Distribution of energy :'
c
c       Define the number of rays absorbed by each surface and the
c       distribution factor:
      Do 30 k1 = 2, Nstot

c       Define the total number of area elements for each surface:
c
       If (k1.eq.6.or.k1.eq.7.or.k1.eq.8.or.k1.eq.9.or.k1.eq.10.
     *       or.k1.eq.11.or.k1.eq.15.or.k1.eq.16.or.k1.eq.17.or.k1.
```

```
     *         eq.18.or.k1.eq.19.or.k1.eq.20) then

c           (big triangles)
          Lmax = jdir(k1)/2 * (jdir(k1)+1)

        Else if (k1.eq.4.or.k1.eq.5.or.k1.eq.12.or.k1.eq.14) then
c           (small triangles)
          Lmax = jdir(k1)/2 * ( 2*idir(k1) +
     *               (jdir(k1)-1) * (-idir(k1)/jdir(k1)) )

        Else if (k1.eq.34.or.k1.eq.36) then
c           (internal wall)
          Lmax = 8*nip + idir(k1)*js*2 +
     *               (jdir(k1)-2*js)*(is+idir(k1)-ie)
        Else if (k1.eq.35) then
          Lmax = 0

        Else if (k1.eq.29) then
c           (sensor aperture)
          Lmax = Nr*Nphi

        Else if (k1.ge.38.and.k1.le.51) then
c           (rings)
          Lmax = idir(k1)*jdir(k1)*4

        Else
c           (rectangular surfaces, or hemispherical shields)
          Lmax = idir(k1)*jdir(k1)
        Endif

c       Define the total number of rays absorbed by surface S(k1):
         ntrabs = 0
         Do 31 L = 1, Lmax
               ntrabs = ntrabs + N(k1,L)
 31      Continue
         dntrabs = dble( ntrabs )
         dntre   = dble( ntre )
         dfacs   = dntrabs / dntre
         Write (7,306) k1, ntrabs, nse, k1, dfacs
 30      Continue
c-------------------------------------------------------------------------
c    Distribution factor of the sensor-aperture:
c-------------------------------------------------------------------------
         Write (7,308)
c       Number of rays absorbed by the sensor-aperture:
         Nra29 = 0
         ddeno = dble ( Ntre )
         Do 33 m = 1, Nr*Nphi
               Nra29 = Nra29 + N(29,m)
               dnume = dble( N(29,m) )
               dfacs = dnume / ddeno
               Write (7,307) m, N(29,m), nse, m, dfacs
 33      Continue
```

```
      Endif
c
c  Unit 8 :
c
      Dnume = dble(Nra29)
      Ddeno = dble(Ntre)
      Df29  = Dnume/Ddeno



c-------------------------------------------------------------------
c     Calculate CPU run time:
c       (Ntma: hours , Ntmb: minutes , tmc: seconds)
c-------------------------------------------------------------------
      tmc=Dble( mclock() )/1.00d2/60.00d0/60.00d0
      Ntma=int(tmc)
      tmc=(tmc-dble(Ntma))*60.00d0
      Ntmb=int(tmc)
      tmc=(tmc-dble(Ntmb))*60.00d0
c-------------------------------------------------------------------
c     Determine the mean value of the distribution factor on the
c     sensor aperture (Narea : Number of area-elements of the sensor
c     aperture):
c-------------------------------------------------------------------
      Narea = Nr*Nphi
      Dnarea = dble (narea)
      DNtre = dble (Ntre)
      Sumdf = 0.00d0

      Do  m = 1 , Nr*Nphi
          Sumdf = sumdf + N ( 29,m )
      Enddo
      Dmean = sumdf / Dnarea / Dntre
c-------------------------------------------------------------------
c     Determine the standard deviation of the distribution factor:
c-------------------------------------------------------------------
      Sumdf2 = 0.00d0
      Do  m = 1,Narea
          Dnume = dble ( N(29,m))
          Dxi = Dnume / Dntre
          Sumdf2 = Sumdf2 + Dxi**2
      Enddo
      Sigma = ( Sumdf2 - Dnarea * Dmean **2 ) / Dnarea
      Sigma = dsqrt (sigma)
c.....
c               write the result to unit 8
c.....
      If (NF6.eq.1) then
        line1 = '-------------------------------------------------'
        Write (8,*)'Nmir='
        write (8,*) Nmir
        write (8,311) hname
        write (8,*) 'CPU run time:'
        write (8,310) Ntma, Ntmb, tmc
```

```
      Write (8,312) Ntre
      Write (8,313) Ntrt
      Write (8,314) Dmud, Dlad
      Write (8,*) 'Baffle-mirror-sensor configuration: '
      Write (8,317) a1deg, a2deg, b9
      Write (8,322) Tramir
      Write (8,309) Df29
      Write (8,315) Dmean
      Write (8,316) sigma
      Write (8,*) 'Output filenames:'
      write (8,321) name1, NF1
      Write (8,321) name2, NF2
      Write (8,321) name3, NF3
      Write (8,321) name4, NF4
      Write (8,321) name5, NF5
      Write (8,321) name6, NF6
    Endif
c-------------------------------------------------------------
c                    format
c-------------------------------------------------------------

 302  Format (2X,'Ray number ',i12,' is lost !!!')
 303  Format (2X,'Energy bundles emitted on surface S',i2,' :',
     *      i12)
 304  Format (2X,'Energy bundles absorbed by the enclosure  :',i12)
 305  Format (2x,'Energy bundles lost                :',i12)
 306  Format (2x,'Surface (',i3,')= ',i12,3x,'df(',i3,',',i4,')=',e10.4)
 307  Format (2X,'Area (',i3,')=',i12,3x,'df(',i3,',',i3,')=',e10.4)
 308  Format (2X,'Distribution of energy on the sensor aperture:')
 309  Format (2x,'Distribution Factor of the sensor aperture   :',
     *          E10.4)
 310  Format (2x,i5,' Hour(s) ',i2,' Minutes ',f4.1,' Seconds')

 311  Format (2x,'Host Machine: ',a20)
 312  Format (2x,i12,' rays emitted')
 313  Format (2x,i12,' rays traced')
 314  Format (2x,'Emission at Mu =',f7.3,' Lambda =',f7.3)
 317  Format (2x,'a1 =',f7.3,' a2 =',f7.3,' b9 =',f7.3)
 315  Format (2x,'Mean value of distribution factors on sensor aperture:
     *',e10.4)
 316  Format (2x,'Standard deviation: ',e10.4)
 320  Format (2x,'Last bounced on surface ',i3)
 321  Format (2x,a10,'  , ',i1)
 322  Format (2x,'Distance of the diffuser plate from the origin: ',
     *      f7.3)

*c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*c            End of emission from surface nse loop
*c  (place this end here to see the result of emission from individual
*c   surface.)
*c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* 5    Continue
```

```
c*************************************************************************
c                    End ray-trace technique
c*************************************************************************
c Write the coordinates of the points forming the geometry of the
c MAM enclosure to the file defined in unit 1 :
      If (NF1.eq.1) then
      Call PRINT1 (p1,1)
      Call PRINT1 (p2,2)
      Call PRINT1 (p3,3)
      Call PRINT1 (p4,4)
      Call PRINT1 (p5,5)
      Call PRINT1 (p6,6)
      Call PRINT1 (p7,7)
      Call PRINT1 (p8,8)
      Call PRINT1 (p9,9)
      Call PRINT1 (p10,10)
      Call PRINT1 (p11,11)
      Call PRINT1 (p12,12)
      Call PRINT1 (p13,13)
      Call PRINT1 (p14,14)
      Call PRINT1 (p15,15)
      Call PRINT1 (p16,16)
      Call PRINT1 (p17,17)
      Call PRINT1 (p18,18)
      Call PRINT1 (p19,19)
      Call PRINT1 (p20,20)
      Call PRINT1 (p21,21)
      Call PRINT1 (p22,22)
      Call PRINT1 (p23,23)
      Call PRINT1 (p24,24)
      Call PRINT1 (p25,25)
      Call PRINT1 (p26,26)
      Call PRINT1 (p27,27)
      Call PRINT1 (p28,28)
      Call PRINT1 (p29,29)
      Call PRINT1 (p30,30)
      Call PRINT1 (p31,31)
      Call PRINT1 (p32,32)
      Call PRINT1 (p33,33)
      Call PRINT1 (p34,34)
      Call PRINT1 (p35,35)
      Call PRINT1 (p36,36)
      Call PRINT1 (p37,37)
      Call PRINT1 (p38,38)
      Call PRINT1 (p39,39)
      Call PRINT1 (p40,40)
      Call PRINT1 (p41,41)
      Call PRINT1 (p42,42)
      Call PRINT1 (p43,43)
      Call PRINT1 (p44,44)
      Call PRINT1 (p45,45)
```

```
Call PRINT1 (p46,46)
Call PRINT1 (p47,47)
Call PRINT1 (p48,48)
Call PRINT1 (p49,49)
Call PRINT1 (p50,50)
Call PRINT1 (p51,51)
Call PRINT1 (p52,52)
Call PRINT1 (p53,53)
Call PRINT1 (p54,54)
Call PRINT1 (p55,55)
Call PRINT1 (p56,56)
Call PRINT1 (p57,57)
Call PRINT1 (p58,58)
Call PRINT1 (p59,59)
Call PRINT1 (p60,60)
Call PRINT1 (p61,61)
Call PRINT1 (p62,62)
Call PRINT1 (p63,63)
Call PRINT1 (p64,64)
Call PRINT1 (p65,65)
Call PRINT1 (p66,66)
Call PRINT1 (p67,67)
Call PRINT1 (p68,68)
Call PRINT1 (p69,69)
Call PRINT1 (p70,70)
Call PRINT1 (p71,71)
Call PRINT1 (p72,72)
Call PRINT1 (p73,73)
Call PRINT1 (p74,74)
Call PRINT1 (p75,75)
Call PRINT1 (p76,76)
Call PRINT1 (p77,77)
Call PRINT1 (p78,78)
Call PRINT1 (p79,79)
Call PRINT1 (p80,80)
Call PRINT1 (p81,81)
Call PRINT1 (p82,82)
Call PRINT1 (p83,83)
Call PRINT1 (p84,84)
Call PRINT1 (p85,85)
Call PRINT1 (p86,86)
Call PRINT1 (p87,87)
Call PRINT1 (p88,88)
Call PRINT1 (p89,89)
Call PRINT1 (p90,90)
Call PRINT1 (p91,91)
Call PRINT1 (p92,92)
Call PRINT1 (p93,93)
Call PRINT1 (p94,94)
Call PRINT1 (p95,95)
Call PRINT1 (p96,96)
```

```
      Call PRINT1 (p97,97)
      Call PRINT1 (p98,98)
      Call PRINT1 (p99,99)
c-----------------------------------------------------------------------------
c  Order of connection for drawing of the enclosure by TECPLOT
c-----------------------------------------------------------------------------
      Call Print4 (1,2,3,4)
      Call print4 (5,6,11,10)
      Call print4 (2,6,11,11)
      Call print4 (1,5,10,10)
      Call print4 (10,14,18,18)
      Call print4 (10,18,17,17)
      Call print4 (10,17,22,22)
      Call print4 (10,22,23,23)
      Call print4 (10,23,24,24)
      Call print4 (10,24,25,25)
      Call print4 (3,8,12,12)
      Call print4 (8,9,13,12)
      Call print4 (4,9,13,13)
      Call print4 (13,16,20,20)
      Call print4 (13,20,21,21)
      Call print4 (13,21,26,26)
      Call print4 (13,26,27,27)
      Call print4 (13,27,28,28)
      Call print4 (13,28,29,29)
      Call print4 (2,3,12,11)
      Call print4 (1,4,13,10)
      Call print4 (12,16,14,11)
      Call print4 (14,16,20,18)
      Call print4 (20,18,17,21)
      Call print4 (21,17,22,26)
      Call print4 (26,22,23,27)
      Call print4 (27,23,24,28)
      Call print4 (28,24,25,29)
      Call print4 (41,40,10,13)
      Call print4 (30,31,32,33)
      Call print4 (44,45,46,47)
      Call print4 (48,49,50,51)
      Call print4 (52,53,54,55)
      Call print4 (56,57,58,59)
      Call print4 (60,61,62,63)
      Call print4 (64,65,66,67)
      Call print4 (68,69,70,71)
      Call print4 (72,73,74,75)
      Call print4 (76,77,78,79)
      Call print4 (80,81,82,83)
      Call print4 (84,85,86,87)
      Call print4 (88,89,90,91)
      Call print4 (92,93,94,95)
      Call print4 (96,97,98,99)
      Endif
```

```
c-------------------------------------------------------------------------------
c    Subdivide the surfaces of the enclosure into area elements
c    then write the coordinates of the grid points and the normalized
c    distribution factors to a file for data visualization by TECPLOT.
c-------------------------------------------------------------------------------
*      title = '2-baf'
*      Call SUBREC (p1,p2,p3,dc(2),2,idir(2),jdir(2),iflag(2),title,2)
c Right side :
       If (NF3.eq.1) then
       title = '3re-baf'
       Call SUBREP (p5,p6,p10,3,idir(3),jdir(3),title,3)
       title = '4tr-baf'
       Call SUBTRA (p6,p2,p11,4,idir(4),jdir(4),title,3)
       title = '5tr-baf'
       Call SUBTRA (p5,p1,p10,5,idir(5),jdir(5),title,3)
       title = '6tr'
       Call SUBQUA (p10,p11,p18,idir(6),6,title,3)
       title = '7tr'
       Call SUBQUA (p17,p10,p18,idir(7),7,title,3)
       title = '8tr'
       Call SUBQUA (p10,p17,p22,idir(8),8,title,3)
       title = '9tr'
       Call SUBQUA (p10,p22,p23,idir(9),9,title,3)
       title = '10tr'
       Call SUBQUA (p10,p23,p24,idir(10),10,title,3)
       title = '11tr'
       Call SUBQUA (p10,p24,p25,idir(11),11,title,3)
       Endif
c Left side :
       If (NF2.eq.1) then
       title = '12tr-baf'
       Call SUBTRA (p8,p3,p12,12,idir(12),jdir(12),title,2)
       title = '13re-baf'
       Call SUBREP (p9,p8,p13,13,idir(13),jdir(13),title,2)
       title = '14tr-baf'
       Call SUBTRA (p9,p4,p13,14,idir(14),jdir(14),title,2)
       title = '15tr'
       Call SUBQUA (p13,p16,p20,idir(15),15,title,2)
       title = '16tr'
       Call SUBQUA (p21,p13,p20,idir(16),16,title,2)
       title = '17tr'
       Call SUBQUA (p13,p21,p26,idir(17),17,title,2)
       title = '18tr'
       Call SUBQUA (p13,p26,p27,idir(18),18,title,2)
       title = '19tr'
       Call SUBQUA (p13,p27,p28,idir(19),19,title,2)
       title = '20tr'
       Call SUBQUA (p13,p28,p29,idir(20),20,title,2)
c Top side:
       title = '21to-baf'
*      Call SUBREC (p2,p3,p11,dc21,21,
*      *            idir(21),jdir(21),iflag(21),title,2)
```

```
      Call SUBRE2 (p2(3),p11(3),p2,p3,p11,dc21,21,
     *          idir(21),jdir(21),iflag(21),title,2)
      title = '22bo-baf'
      Call SUBREC (p1,p4,p10,dc22,22,
     *          idir(22),jdir(22),iflag(22),title,2)
      title = '23le-enc'
      Call SUBREC (p11,p12,p14,dc23,23,
     *          idir(23),jdir(23),iflag(23),title,2)
      title = '24to-enc'
      Call SUBREC (p14,p16,p18,dc24,24,
     *          idir(24),jdir(24),iflag(24),title,2)
      title = '25close'
      Call SUBREC (p17,p18,p20,dc25,25,
     *          idir(25),jdir(25),iflag(25),title,2)
      title = '26bo-enc'
      Call SUBREC (p17,p21,p22,dc26,26,
     *          idir(26),jdir(26),iflag(26),title,2)
      title = '27ri-sap'
      Call SUBREC (p22,p23,p26,dc27,27,
     *          idir(27),jdir(27),iflag(27),title,2)
      title = '28-sw'
      Call SUBREC (p23,p24,p27,dc28,28,
     *          idir(28),jdir(28),iflag(28),title,2)
      title = '29minus'
      b9p = b9 - 0.15d0
      Call SUBCIR (b9p,rsa,a2,nr,nphi,29,title,2)
      title = '29plus'
      b9p = b9 + 0.15d0
      Call SUBCIR (b9p,rsa,a2,nr,nphi,29,title,2)
      title = '30bow'
      Call SUBREC (p24,p25,p28,dc30,30,
     *          idir(30),jdir(30),iflag(30),title,2)
      title = '31ba-ba'
      Call MOVET (p25,p25p,tau,0)
      Call MOVET (p10,p10p,tau,0)
      Call MOVET (p29,p29p,tau,0)
      Call SUBREC (p25p,p10p,p29p,dc31,31,
     *          idir(31),jdir(31),iflag(31),title,2)
      title = '32to-mi'
      Call SUBREC (p30,p31,p32,dc32,32,
     *          idir(32),jdir(32),iflag(32),title,2)
      title = '33ba-mi'
      Call MOVET (p30,p30p,0,tmir)
      Call MOVET (p31,p31p,0,tmir)
      Call MOVET (p33,p33p,0,tmir)
      Call SUBREC (p30p,p31p,p33p,dc33,33,
     *          idir(33),jdir(33),iflag(33),title,2)
      title = '34to-int'
      Call SUBINT (a6,p42(1),p42(3),34,rint,title,2)
c     (s35 is the circular hole on the internal wall)
      title = '36bo-int'
      p42p(1) = p42(1) + tau
```

```
      p42p(3) = p42(3) - tau
      Call SUBINT (a6,p42p(1),p42p(3),36,rint,title,2)
c     (s37 represents the spherical mirrors)
      title = '38t-r1'
      Call SUBRIN (p44,dc38,tau,2,a1,a3,cXmax(38),cXmin(38),rXmax(38),
     *          rXmin(38),cYmin(38),cYmax(38),rYmin(38),rYmax(38),
     *          idir(38),jdir(38),title,2,38)
      title = '39b-r1'
      Call SUBRIN (p44,dc39,tau,1,a1,a3,cXmax(39),cXmin(39),rXmax(39),
     *          rXmin(39),cYmin(39),cYmax(39),rYmin(39),rYmax(39),
     *          idir(39),jdir(39),title,2,39)
      title = '40t-r2'
      Call SUBRIN (p52,dc40,tau,2,a1,a3,cXmax(40),cXmin(40),rXmax(40),
     *          rXmin(40),cYmin(40),cYmax(40),rYmin(40),rYmax(40),
     *          idir(40),jdir(40),title,2,40)
      title = '41b-r2'
      Call SUBRIN (p52,dc41,tau,1,a1,a3,cXmax(41),cXmin(41),rXmax(41),
     *          rXmin(41),cYmin(41),cYmax(41),rYmin(41),rYmax(41),
     *          idir(41),jdir(41),title,2,41)
      title = '42t-r3'
      Call SUBRIN (p60,dc42,tau,2,a1,a3,cXmax(42),cXmin(42),rXmax(42),
     *          rXmin(42),cYmin(42),cYmax(42),rYmin(42),rYmax(42),
     *          idir(42),jdir(42),title,2,42)
      title = '43b-r3'
      Call SUBRIN (p60,dc43,tau,1,a1,a3,cXmax(43),cXmin(43),rXmax(43),
     *          rXmin(43),cYmin(43),cYmax(43),rYmin(43),rYmax(43),
     *          idir(43),jdir(43),title,2,43)
      title = '44t-r4'
      Call SUBRIN (p68,dc44,tau,2,a1,a3,cXmax(44),cXmin(44),rXmax(44),
     *          rXmin(44),cYmin(44),cYmax(44),rYmin(44),rYmax(44),
     *          idir(44),jdir(44),title,2,44)
      title = '45b-r4'
      Call SUBRIN (p68,dc45,tau,1,a1,a3,cXmax(45),cXmin(45),rXmax(45),
     *          rXmin(45),cYmin(45),cYmax(45),rYmin(45),rYmax(45),
     *          idir(45),jdir(45),title,2,45)
      title = '46t-r5'
      Call SUBRIN (p76,dc46,tau,2,a1,a3,cXmax(46),cXmin(46),rXmax(46),
     *          rXmin(46),cYmin(46),cYmax(46),rYmin(46),rYmax(46),
     *          idir(46),jdir(46),title,2,46)
      title = '47b-r5'
      Call SUBRIN (p76,dc47,tau,1,a1,a3,cXmax(47),cXmin(47),rXmax(47),
     *          rXmin(47),cYmin(47),cYmax(47),rYmin(47),rYmax(47),
     *          idir(47),jdir(47),title,2,47)
      title = '48t-r6'
      Call SUBRIN (p84,dc48,tau,2,a1,a3,cXmax(48),cXmin(48),rXmax(48),
     *          rXmin(48),cYmin(48),cYmax(48),rYmin(48),rYmax(48),
     *          idir(48),jdir(48),title,2,48)
      title = '49b-r6'
      Call SUBRIN (p84,dc49,tau,1,a1,a3,cXmax(49),cXmin(49),rXmax(49),
     *          rXmin(49),cYmin(49),cYmax(49),rYmin(49),rYmax(49),
     *          idir(49),jdir(49),title,2,49)
      title = '50t-r7'
```

```
      Call SUBRIN (p92,dc50,tau,2,a1,a3,cXmax(50),cXmin(50),rXmax(50),
     *          rXmin(50),cYmin(50),cYmax(50),rYmin(50),rYmax(50),
     *          idir(50),jdir(50),title,2,50)
      title = '51b-r7'
      Call SUBRIN (p92,dc51,tau,1,a1,a3,cXmax(51),cXmin(51),rXmax(51),
     *          rXmin(51),cYmin(51),cYmax(51),rYmin(51),rYmax(51),
     *          idir(51),jdir(51),title,2,51)
      Endif

      Stop
      End



c.....
c                Subroutine LOCGLO
c.....
c  This subroutine transforms the coordinates of a point from
c  P(x', y', z') in the local coordinate system {o', x', y', z'}
c  to P(x, y, z) in the global coordinate system {o, x, y, z}.
c  Frame {o, x, y, z} is the transformation of frame {o', x', y', z'}
c  by a translation T along the z'-direction and a rotation by ALPHA
c  about the y-axis, the angle ALPHA is considered positive counter-
c  clockwise from {o', x', y', z'} to {o, x, y, z}.
c.....
      Subroutine LOCGLO (Pp, p, T, alpha)
      Implicit double precision (a-h, o-z)
      Dimension Pp(3), p(3)

        p(1) =  Pp(1)*dcos(alpha) + (Pp(3) - T)*dsin(alpha)
        p(2) =  Pp(2)
        p(3) = -Pp(1)*dsin(alpha) + (Pp(3) - T)*dcos(alpha)

      Return
      end
c.....
c             Subroutine GLOLOC
c.....
c  This subroutine transforms the coordinates of a point P (x,y,z)
c  from the global coordinate system {o,x,y,z} to a local coordinate
c  system {o',x',y',z'}. The trans formation of {F} to {F'} is a
c  translation Tx, Tz from (o) to (o') and a rotation by alpha about
c  the y-axis.
c.....
      Subroutine GLOLOC (p, Pp, Tx, Tz, alpha)
      Implicit double precision (a-h, o-z)
      Dimension p(3), Pp(3)

      Pp(1) = (p(1)-Tx)*dcos(alpha) + (p(3)-Tz)*dsin(alpha)
      Pp(2) =  p(2)
      Pp(3) = -(p(1)-Tx)*dsin(alpha) + (p(3)-Tz)*dcos(alpha)

      Return
      End
```

```
c.....
c                Subroutine ANGLEZ
c.....
c  This suborutine defines the angle between the z-axis and the normal
c  of a surface perpendicular to the x,z-plane.
c          xn: x-component of the normal vector.
c          yn: y-component of the normal vector.
c.....
      Subroutine ANGLEZ (xn, yn, omega)
      Implicit double precision (a-h, o-z)
        pi = dacos(-1.00d0)
        omega = datan2 (xn, yn)
        If (omega.lt.0.00d0) omega = 2.00d0*pi + omega
      Return
      End


c.....
c                Function DIST
c.....
c    This function determines the distance between 2 points.
c.....
      Function DIST (p1, p2)
      Implicit double precision (a-h, o-z)
      Dimension p1(3), p2(3)

     Dist = dsqrt( (p1(1)-p2(1))**2 + (p1(2)-p2(2))**2
     *              + (p1(3)-p2(3))**2 )

      Return
      End

c.....
c              Subroutine XMIMA
c.....
c  This subroutine determines the Xmin and Xmax of a rectangular
c  surface perpendicular to the x,z-plane.
c     pa, pb : Vectors containing the coordinates of 2 corners
c          on the side parallel to the x,z-plane.
c     Ns     : The number of the surface.
c.....
      Subroutine XMIMA (pa, pb, ns)
      Implicit double precision (a-h, o-z)
      Dimension pa(3), pb(3), cXmin(55), cXmax(55), cYmin(55), cYmax(55)

c  Common variables :
      Common / bl11 / cXmin, cXmax, cYmin, cYmax

      If (pa(1).LE.pb(1)) then
        cXmin(ns) = pa(1)
        cXmax(ns) = pb(1)
      Else
        cXmin(ns) = pb(1)
        cXmax(ns) = pa(1)
      Endif

      Return
```

```
      End
c.....
c              Subroutine rXmima
c.....
c  This subroutine determines the Xmin and Xmax of the inner rectangu-
c  lar surface of the reflectors
c    pa, pb : Vectors containing the coordinates of 2 corners
c             of the surface
c    Ns     : The number of the surface.
c.....
      Subroutine rXmima (pa, pb, ns)
      Implicit double precision (a-h, o-z)
      Dimension pa(3), pb(3), rXmin(38:51), rXmax(38:51)
     *                      , rYmin(38:51), rYmax(38:51)

c  Put frequently used variables in one storage block :
      Common / bl12 / rXmin, rXmax, rYmin, rYmax

      If (pa(1).LE.pb(1)) then
         rXmin(ns) = pa(1)
         rXmax(ns) = pb(1)
      Else
         rXmin(ns) = pb(1)
         rXmax(ns) = pa(1)
      Endif

      Return
      End

c.....
c              Subroutine SFRONT
c.....
c    This subroutine determines the target surface is in front of
c    the source surface,
c
c       The dot product of Usij and Ni > 0.
c       (The ray is emitted in the proper direction / the target
C        surface is in front of the source surface)
c
c
c.....
      Subroutine SFRONT (dci1, dci2, dci3, front)
      Implicit double precision (a-h, o-z)
      Dimension dc(3)
      Logical front

c  Put frequently used variables in one unique block :
      Common / block7 / pi,x1,y1,z1,dl1,dm1,dn1,x2,y2,z2,dl2,dm2,dn2

     Dotiji = (x2-x1)*dci1 + (y2-y1)*dci2 + (z2-z1)*dci3

      If (dotiji.GE.0.00d0) then
         front = .true.
      Else
         front = .false.
      Endif
```

```
      Return
      End
c.....
c                Subroutine SFACE
c.....
c    This subroutine determines if the target surface is facing the
c    source surface,
c    The condition to sastisfy is :
c       (i : Emitting surface, j : Incident surface)
c
c     The dot product of Usij and Nj is less than or equal to 0.
c
c  Note : This subroutine is needed for intersection with 2 sided
c        surfaces, by coincidence, all the 2 sided surfaces in this
c        particular study are perpendicular to the x,z-plane, i.e.,
c        the y-component of the normal vector is zero.
c        Thus,  Dotijj = (x2-x)*dc(1) + (y2-y)*dc(2) + (z2-z)*dc(3)
c        becomes Dotijj = (x2-x)*dc(1) + (z2-z)*dc(3)
c
c.....
      Subroutine SFACE (dc, face)
      Implicit double precision (a-h, o-z)
      Dimension dc(3)
      Logical face

c  Put frequently used variables in one unique block :
      Common / block7 / pi,x1,y1,z1,dl1,dm1,dn1,x2,y2,z2,dl2,dm2,dn2

     Dotijj = (x2-x1)*dc(1) + (z2-z1)*dc(3)

      If (dotijj.LE.0.00d0) then
         face = .true.
      Else
         face = .false.
      Endif

      Return
      End

c.....
c                Subroutine PRSURF
c.....
c  This subroutine determines the coordinates of the point of incidence
c  on a surface PERPENDICULAR to the x,z-plane.
c    x1,y1,z1 : Point of emission.
c    dl1,dm1,dn1 : Direction of emission.
c    p(p1,p2,p3) : One point of the incident surface.
c    dc(dc1,dc2,dc3) : Normal uint vector of the incident surface.
c    x2,y2,z2 : Point of intersection.
c.....
      Subroutine PRSURF (p ,dc)
      Implicit double precision (a-h, o-z)
      Dimension p(3), dc(3)

c  Put frequently used variables in one unique block :
      Common /block7/pi,x1,y1,z1,dl1,dm1,dn1,x2,y2,z2,dl2,dm2,dn2
```

```
      T = ( dc(1)*(p(1)-x1) + dc(3)*(p(3)-z1) ) /
     *                ( dc(1)*dl1 + dc(3)*dn1 )
      x2 = T*dl1 + x1
      y2 = T*dm1 + y1
      z2 = T*dn1 + z1

      Return
      End


c.....
c              Subroutine PRCHEC
c.....
c   This subroutine checks if a point is inside a rectangular area
c   and perpendicular to the x,z-plane.
c   x2,y2,z2:point of intersection on the surface (from PRSURF).
c   Xmin, Xmax of the surface are from cXmin(ns) and cXmax(ns)
c   inside=true:the point is inside the area.
c.....
      Subroutine PRCHEC (x2, y2, ns, inside)
      Implicit double precision (a-h, o-z)
      Dimension cXmin(51), cXmax(51), cYmin(51), cYmax(51)
      Logical inside

c  Put frequently used variables in one block :
      Common / bl11 / cXmin, cXmax, cYmin, cYmax

      inside = .false.
      If ( (cYmin(ns).LE.y2).AND.(y2.LE.cYmax(ns)) ) then
c    (if the first condition is not true, there is no need to check
c    the second condition)
        If ( (cXmin(ns).LE.x2).AND.(x2.LE.cXmax(ns)) ) then
          inside = .true.
        Endif
      Endif
      Return
      End
c.....
c              Subroutine PPXZ
c.....
      Subroutine PPXZ (y2, x2, Zmin, Zmax, ns, inside)
      Implicit double precision (a-h, o-z)
      Dimension cXmin(51), cXmax(51), cYmin(51), cYmax(51)
      Logical inside

c  Put frequently used variables in common blocks:
      Common / bl11 / cXmin, cXmax, cYmin, cYmax

      inside = .false.
      If ( (cYmin(ns).LE.y2).AND.(y2.LE.cYmax(ns)) ) then
c      (if the first condition is not true, then there is no need to
c      check for the second condition)
```

```
          If ( (Zmin.LE.z2).AND.(z2.LE.Zmax) ) then
             inside = .true.
          Endif
        Endif
        Return
        End

c.....
c                Subroutine RING
c.....
c    This subroutine checks if a point is inside the surface of a
c    reflector of the baffle.
c    inside = true : the point is inside the area.
c.....
        Subroutine RING (x2, y2, ns, inside)
        Implicit double precision (a-h, o-z)
        Dimension cXmin(51), cXmax(51), cYmin(51), cYmax(51),
       *          rXmin(38:51), rXmax(38:51), rYmin(38:51), rYmax(38:51)
       Logical inside, outer, inner
c
c  Put frequently used variables in one block:
c
        Common / bl11 / cXmin, cXmax, cYmin, cYmax
        Common / bl12 / rXmin, rXmax, rYmin, rYmax
 c
c  Initialize necessary variables:
c
        inside = .false.
        outer  = .false.
       inner  = .true.
c
c  Check the intersection with the outer area:
c
        If ( (cYmin(ns).LE.y2).AND.(y2.LE.cYmax(ns)).AND.
       *     (cXmin(ns).LE.x2).AND.(x2.LE.cXmax(ns)) ) outer = .true.
c
c  Check the intersection with the inner area:
c
        If ( (rYmin(ns).LT.y2).AND.(y2.LT.rYmax(ns)).AND.
       *     (rXmin(ns).LT.x2).AND.(x2.LT.rXmax(ns)) ) inner = .false.
c
c  The point of intersection is on the reflector if this point is
c  inside the outer area and outside the inner area.
c
        If (outer.AND.inner) inside = .true.

        Return
        End

c.....
c                Subroutine TRISUR
c.....
c    Intersection with a triangular area parallel to the x,z-plane
c    pa(1),...,pc(3):coordinates of 3 points making up the surface
c    x2,y2,z2: hit point on the surface
c    inside=true:the hit point is inside the area.
c.....
```

```
      Subroutine TRISUR (pa, pb, pc, x2, z2, inside)
      Implicit double precision (a-h, o-z)
      Dimension pa(3), pb(3), pc(3)
      Logical inside

      dnu = (pb(1)-pa(1))*(pc(3)-pa(3)) - (pc(1)-pa(1))*(pb(3)-pa(3))

c  Coordinate along the AB-direction, q :

      q = (x2-pa(1))*(pc(3)-pa(3)) - (z2-pa(3))*(pc(1)-pa(1))
      q = q/dnu

c  Coordinate along the AC-direction, r :

      r = (pb(1)-pa(1))*(z2-pa(3)) - (pb(3)-pa(3))*(x2-pa(1))
      r = r/dnu

c  The point is inside the triangular area (a,b,c) if q and r are both
c  positive and q+r is less than unity :

      If ((q.GE.0.00d0).AND.(r.GE.0.00d0)) then
         If ((q+r).LE.1.00d0) then
            inside = .true.
         Endif
      Else
         inside = .false.
      Endif

      Return
      End

c.....
c              Subroutine RECXYZ
c.....
c     Record the point of intersection and the distance from the site
c     emission to the site of intersection.
c.....
      Subroutine RECXYZ (j)
      Implicit double precision (a-h, o-z)
      Dimension Nsurf(51), dis(51), xcoor(51), ycoor(51), zcoor(51)
      Logical hit

c  Put frequently used variables in common blocks :

      Common / block1 / Ninter,Nsurf
      Common / block3 / Dis,xcoor,ycoor,zcoor
      Common / block4 / Hit
      Common / block7 / pi,x1,y1,z1,dl1,dm1,dn1,x2,y2,z2,dl2,dm2,dn2

      Hit = .true.
      Ninter = Ninter + 1
      Nsurf(Ninter) = j
      Dis(Ninter) = (x2-x1)**2 + (y2-y1)**2 + (z2-z1)**2
```

```
c  Note that Dis(Ninter) represents the distances between the point of
c  emission and the possible points of intersection, the point of inter-
c  section is the one that has the shortest distance. But the radical is
c  always a positive number, so I do not have to take the square root
c  operation to perform this comparison.

      xcoor(j) = x2
      ycoor(j) = y2
      zcoor(j) = z2

      Return
      End

c.....
c              Subroutine XZPLAN
c.....
c  This subroutine determines the point of intersection between a line
c  passing through (x1, y1, z1) in the direction l1, m1, n1 and a sur-
c  face parallel to the x,z-plane at y = py.
c.....
      Subroutine XZPLAN (py)
      Implicit double precision (a-h, o-z)
      Common / block7 /pi,x1,y1,z1,dl1,dm1,dn1,x2,y2,z2,dl2,dm2,dn2

      y2 = py
      x2 = (y2-y1)*dl1/dm1 + x1
      z2 = (y2-y1)*dn1/dm1 + z1

      Return
      End

c.....
c              Subroutine IOAF
c.....
c  This subroutine determines if a point is inside a rectangular area
c  PARALLEL to the x,z-plane and this area is defined by 4 points :
c       (top-left:1,top-right:2,bottom-left:3,bottom-right:4).
c.....
      Subroutine IOAF (x, z, p1, p2, p3, p4, inside)
      Implicit double precision (a-h, o-z)
      Dimension p1(3), p2(3), p3(3), p4(3)
      Logical inside


      c1 = z + (p1(1)-x)*(p1(3)-p2(3))/(p1(1)-p2(1))-p1(3)
      c2 = z + (p3(1)-x)*(p3(3)-p4(3))/(p3(1)-p4(1))-p3(3)
      c3 = z + (p3(1)-x)*(p3(3)-p1(3))/(p3(1)-p1(1))-p3(3)
      c4 = z + (p4(1)-x)*(p4(3)-p2(3))/(p4(1)-p2(1))-p4(3)

      If ((c1.LE.0.00d0).AND.(c2.GE.0.00d0).AND
     *  .(c3.GE.0.00d0).AND.(c4.LE.0.00d0)) then
         inside = .true.
      Else
         inside = .false.
      Endif
```

```
      Return
      End


c.....
c                Subroutine MIRCHE
c.....
c  Check if the point of intersection on the top side of the diffuser plate
c  if inside a circle of the MAM.  If yes, what are the coordinates of the
c  center of this circle.
c.....
      Subroutine MIRCHE (x2, y2)
      Implicit double precision (a-h, o-z)

      Dimension xc(5), yc(5)
      Logical Incirc

c  Frequently used variables :
      Common /block2/ dmX, dmY, dbX, dbY, dsX, dsY, rd, e1, e2, e3, e4
      Common / bl13 / xcs, ycs
      Common / bl14 / xc, yc, Ncase
      Common / bl15 / rs, incirc

c  Determine on which cell is the point of incidence, and the coordi-
c  nates of the centers of the 5 possible mirrors :
      incirc = .false.
      Call MIRS29 (x2, y2)

      If (Ncase.EQ.3) then
c     The hit point is on an inner cell of the difuser plate, is this
c     point inside one of the five mirrors of the cell? If yes, what are
c     the coordinates of the center of that mirror?

       Do 3 i = 1, 5
            c = (x2-xc(i))**2 + (y2-yc(i))**2
            If (c.LT.rd**2) then
               incirc = .true.
               xcs = xc(i)
               ycs = yc(i)
              Goto 4
           Endif
  3    Continue
  4    Continue

      Else if (Ncase.EQ.2) then
c     The hit point is at the border of the difuser plate, is this point
c     inside one of the two mirrors of that cell? If yes, what are the
c     coordinates of the center of the incident mirror?

       Do 1 i = 1, 2
            c = (x2-xc(i))**2 + (y2-yc(i))**2
            If (c.LT.rd**2) then
               incirc = .true.
```

```
                xcs = xc(i)
                ycs = yc(i)
                goto 2
              Endif
 1    Continue
 2    Continue

      Else if (Ncase.EQ.1) then
c     The hit point is at a corner, is the point indisde the mirror of
c     this cell? if yes, what are the coordinates of the center of this
c     mirror?

        If ( ((x2-xc(1))**2 + (y2-yc(1))**2).LT.rd**2) then
          incirc = .true.
          xcs = xc(1)
          ycs = yc(1)
        End if

      Endif
      Return
      End
c.....
c                Subroutine MIRS29
c.....
c     Subdivide the MAM into parts (corners, borders, inners) in order
c     to find the coordinates of the center of the spherical mirror.
c.....
c      Note  : This suborutine would run faster than MIRS__ because
c              the centers xc(i) and yc(i) are not to be redefined
c              everytime this subroutine is called.
c.....
      Subroutine MIRS29 (x2, y2)
      Implicit double precision (a-h, o-z)
      Dimension xc(5),yc(5),xcin(5,5,9),ycin(5,5,9),xcto(2,9),ycto(2,9),
     *          xcbo(2,9),ycbo(2,9),xcle(2,9),ycle(2,9),
     *          xcri(2,9),ycri(2,9)

c  Frequently used variables :
      Common / block2 / dmX,dmY,dbX,dbY,dsX,dsY,rd,e1,e2,e3,e4
      common / bl14 / xc, yc, Ncase
      Common / bl26 / xcin,ycin,xcto,ycto,xcbo,ycbo,xcle,ycle,xcri,ycri,
     *                xctl,yctl,xctr,yctr,xcbl,ycbl,xcbr,ycbr

c  Coordinates of the centers of the five circles on the inner cells :
      If ((y2.le.e4).and.(y2.ge.e3).and.(x2.ge.e1).and.(x2.le.e2)) then
        m = int( (x2-e1)/dsX + 1 )
        n = int( (y2-e3)/dsY + 1 )
        Do i = 1, 5
          xc(i) = xcin(i,m,n)
          yc(i) = ycin(i,m,n)
        Enddo
        Ncase = 3
        Return
      Endif
```

```
c Coordinates of the centers of the two circles on the border cells:
c top:
      If ((x2.lt.e1).and.(y2.le.e4).and.(y2.ge.e3)) then
        n = int( (y2-e3)/dsY + 1 )
        Do i = 1, 2
           xc(i) = xcto(i,n)
           yc(i) = ycto(i,n)
         Enddo
         Ncase = 2
         Return
       Endif
c bottom:
      If ((x2.gt.e2).and.(y2.le.e4).and.(y2.ge.e3)) then
         n = int ( (y2-e3)/dsY + 1)
         Do i = 1, 2
            xc(i) = xcbo(i,n)
            yc(i) = ycbo(i,n)
         Enddo
         Ncase = 2
         Return
       Endif
c Left:
       If ((y2.gt.e4).and.(x2.ge.e1).and.(x2.le.e2)) then
          m = int( (x2-e1)/dsX + 1)
          Do i = 1, 2
             xc(i) = xcle(i,m)
             yc(i) = ycle(i,m)
           Enddo
           Ncase = 2
           Return
        Endif
c right:
      If ((y2.lt.e3).and.(x2.ge.e1).and.(x2.le.e2)) then
         m = int( (x2-e1)/dsX + 1 )
        do i = 1, 2
           xc(i) = xcri(i,m)
           yc(i) = ycri(i,m)
        enddo
        Ncase = 2
       Return
      Endif

c Coordinates of the center of the mirror at one of the 4 corners :
c top-left:
      If ((y2.gt.e4).and.(x2.lt.e1)) then
         xc(1) = xctl
         yc(1) = yctl
         Ncase = 1
        return
      endif
```

```
c top-right:
      If ((y2.lt.e3).and.(x2.lt.e1)) then
         xc(1) = xctr
         yc(1) = yctr
         Ncase = 1
         Return
      Endif
c bottom-left:
      If ((y2.gt.e4).and.(x2.gt.e2)) then
         xc(1) = xcbl
         yc(1) = ycbl
         Ncase=1
         Return
      Endif
c bottom-right:
      If ((y2.lt.e3).and.(x2.gt.e2)) then
         xc(1) = xcbr
         yc(1) = ycbr
         Ncase=1
      Endif

      Return
      End


c.....
c                Subroutine HEMIS
c.....
c Find the point of intersection between a line in direction 1, pas-
c sing through point 1  and a hemisperical surface. The point of in-
c tersection is renamed point 2.
c.....
      Subroutine HEMIS
      Implicit double precision (a-h, o-z)

c Frequently used variables :
      Common/block7/pi,x1,y1,z1,dl1,dm1,dn1,x2,y2,z2,dl2,dm2,dn2
      Common/ bl13 / xcs, ycs, zcs
      Common/ bl15 / rs, incirc

c Constants :
      a1 = dl1**2 + dm1**2
      b1 = dn1*( dl1*(x1-xcs) + dm1*(y1-ycs) )

c Coefficients of the quadratic equation :
c    (a = 1.00d0)
      b = b1 - z1*a1
      c = a1*z1**2 - 2.00d0*z1*b1 +
     *    dn1**2*((x1-xcs)**2 + (y1-ycs)**2 - rs**2)
      sdelta = dsqrt (b**2 - c)


c First possible solution :
```

```
      z2p = -b + sdelta
      x2p = (z2p-z1)*dl1/dn1 + x1
      y2p = (z2p-z1)*dm1/dn1 + y1
      disp = (x2p-x1)**2 + (y2p-y1)**2 + (z2p-z1)**2

c Second possible solution :
      z2dp = -b - sdelta
      x2dp = (z2dp-z1)*dl1/dn1 + x1
      y2dp = (z2dp-z1)*dm1/dn1 + y1
      disdp = (x2dp-x1)**2 + (y2dp-y1)**2 + (z2dp-z1)**2

c The point of incidence is the one that has the longest distance
c from the point of emission :
      If (disp.GT.disdp) then
         x2 = x2p
         y2 = y2p
         z2 = z2p
      Else
         x2 = x2dp
         y2 = y2dp
         z2 = z2dp
      Endif

      Return
      End

c.....
c              Subroutine MIRROR
c.....
c Find the point of intersection between a line in direction 1, pas-
c sing through point 1  and a hemisperical surface. The point of in-
c tersection is then renamed point 1.
c.....
      Subroutine MIRROR
      Implicit double precision (a-h, o-z)

c Frequently used variables :
      Common/block7/pi,x1,y1,z1,dl1,dm1,dn1,x2,y2,z2,dl2,dm2,dn2
      Common/ bl13 / xcs, ycs, zcs
      Common/ bl15 / rs, incirc

c Constants :
      a1 = dl1**2 + dm1**2
      b1 = dn1*( dl1*(x1-xcs) + dm1*(y1-ycs) )

c Coefficients of the quadratic equation :
c    (a = 1.00d0)
      b = b1 - z1*a1
      c = a1*z1**2 - 2.00d0*z1*b1 +
     *    dn1**2*((x1-xcs)**2 + (y1-ycs)**2 - rs**2)
      sdelta = dsqrt (b**2 - c)

c First possible solution :
      z2p = -b + sdelta
```

```
      x2p = (z2p-z1)*dl1/dn1 + x1
      y2p = (z2p-z1)*dm1/dn1 + y1
      disp = (x2p-x1)**2 + (y2p-y1)**2 + (z2p-z1)**2

c Second possible solution :
      z2dp = -b - sdelta
      x2dp = (z2dp-z1)*dl1/dn1 + x1
      y2dp = (z2dp-z1)*dm1/dn1 + y1
      disdp = (x2dp-x1)**2 + (y2dp-y1)**2 + (z2dp-z1)**2

c The point of incidence is the one that has the longest distance
c from the point of emission :
      If (disp.GT.disdp) then
         x1 = x2p
         y1 = y2p
         z1 = z2p
      Else
         x1 = x2dp
         y1 = y2dp
         z1 = z2dp
      Endif

      Return
      End
c.....
c                Subroutine REFLEC
c.....
c This subroutine defines the direction 1 of specular reflection by a
c hemispherical surface at point 1 and incident direction 1.
c.....
      Subroutine REFLEC
      Implicit double precision (a-h,o-z)

c Frequently used variables :
      Common/block7/pi,x1,y1,z1,dL1,dm1,dn1,x2,y2,z2,dL2,dm2,dn2
      Common / bl13 / xcs, ycs, zcs
      Common / bl15 / rs, incirc

      a = (2.00d0/Rs**2) * (dL1*(x1-xcs) + dm1*(y1-ycs) + dn1*z1)
      dL2p = dL1 - a*(x1-xcs)
      dm2p = dm1 - a*(y1-ycs)
      dn2p = dn1 - a*z1

c Rescale the direction cosines:

      Scale = dsqrt(dL2p**2 + dm2p**2 + dn2p**2)
      dL1 = dL2p/scale
      dm1 = dm2p/scale
      dn1 = dn2p/scale

      Return
      End
```

```
c.....
c                Subroutikne DISK
c.....
c This subroutine checks if the energy bundle reflected by the hemi-
c spherical mirror at point 1 and in the direction 1 can hit the base
c of the mirror at point 3.
c.....
      Subroutine DISK (V)
      Implicit double precision (a-h,o-z)
      Logical V

c Put frequently used variables in common blocks :
      Common/block7/pi,x1,y1,z1,dL1,dm1,dn1,x2,y2,z2,dL2,dm2,dn2
      Common / bl13 / xcs, ycs, zcs
      common / bl16 / rd, rmh

c Define the coordinates of the point of intersection on the disk (the
c base of the spherical mirror) :
      z3 = rmh
      x3 = (z3-z1)*dL1/dn1 + x1
      y3 = (z3-z1)*dm1/dn1 + y1

c Check if this point is inside the disk of radius rd:
      c = (x3-xcs)**2 + (y3-ycs)**2
      If (c.LT.rd**2) then
        V = .true.
      Else
        V = .false.
      Endif
      Return
      End

c.....
c                Function RAD
c.....
c              Convert Deg to Rad.
c.....
      Function RAD (x)
      Double precision Rad, x, pi
        pi = Dacos(-1.00d0)
        Rad = x*pi/1.80d+2
      Return
      End
c.....
c              Random Number Generator
c.....
c
c This is the initialization routine for the random number generator
c RANDO ().
c   Note: The seed variables can have values between:
c         0 <= ij <= 31328
c         0 <= kl <= 30081
c The random number sequences created by these two seeds are of suff-
c icient length to complete an entire calculation with.For example,
c if several different groups are working on different parts of the
```

```
c  same calculation, each group could be assigned its own IJ seed.
c  That is to say, this random number generator can create 900 million
c  different subsequences - with each subsequence having a length
c  of approximatively 10^30.
c
c  Use IJ=1802 & KL=9373 to test the random number generator. The
c  subroutine RANDO should be used to generate 20000 random numbers.
c  Then display the next 6 six random numbers generated multiplied
c  by 4096*4096. If the random number generator is working properly,
c  the random numbers should be:
c        6533892.0   14220222.0   7275067.0
c        6172232.0   8354498.0    10633180.0
c
c
      Subroutine Randin (ij,kl)
      Implicit double precision (a-h,o-z)
      Dimension u(97)
      Logical Test
      Common/raset1/u,c,cd,cm,i97,j97,test

      Test=.False.
c
*     if (ij.lt.0 .or. ij.gt.31328 .or.
*     1    kl.lt.0 .or. kl.gt.30081 ) then
*        Write (*,*) 'The first random number seed must have a'
*        write (*,*) 'value between 0 and 31328'
*        write (*,*)
*        write (*,*) 'The second seed must have a value between'
*        write (*,*) '0 and 30081'
*        stop
*     endif
c
      i= mod(ij/177,177) + 2
      j= mod(ij    ,177) + 2
      k= mod(kl/169,178) + 1
      l= mod(kl    ,169)
C
      Do 2 ii=1,97
      s = 0.00d0
      t = 0.50d0
      Do 3 jj = 1,24
      m = mod(mod(i*j,179)*k,179)
      i = j
      j = k
      k = m
      l = mod(53*L+1,169)
      if (mod(L*m,64) .ge. 32) then
        s = s + t
      endif
      t = 0.50d0 * t
 3      Continue
      u(ii) = s
 2    Continue
C
```

```
      c = 362436.00d0 / 16777216.00d0
      cd = 7654321.00d0 / 16777216.00d0
      cm = 16777213.00d0 / 16777216.00d0
C
      i97 = 97
      j97 = 33
C
      test = .true.
C
      return
      end
c.....
c                Function RANDO
c.....
c    This is the random number generator proposed by George Marsaglia
c    in Florida State University Report: FSU-SCRI-87-50
c.....
      Function RANDO ()
      implicit double precision (a-h,o-z)
      dimension u(97)
      logical test
      common/raset1/u,c,cd,cm,i97,j97,test
C
*     if (.not.test) then
*       write (*,*) 'Call the initialization routine RANDOIN before'
*       write (*,*) 'calling RANDO !!!'
*       stop
*     endif
c
      uni = u(i97) - u(j97)
      if ( uni.lt.0.0) uni = uni +1.0
      u(i97) = uni
      i97 = i97 - 1
      if (i97 .eq. 0 ) i97 = 97
      j97 = j97 - 1
      if (j97 .eq. 0 ) j97 = 97
      c = c - cd
      if ( c .lt. 0.0 ) c = c + cm
      uni = uni - c
      if (uni .lt. 0.0 ) uni = uni + 1.0
C
      rando = uni
C
      return
      end

c.....
c  Define the Xmin, Xmax, Ymin, Ymax from the coordinates of 3 points .
c.....
      Subroutine MIMAXY (p1, p2, p3)
      Implicit double precision (a-h, o-z)
      Dimension p1(3), p2(3), p3(3), w(2,3)
c  Put frequently used variables in common block:
```

```
      Common / block5 / Xmin, Xmax, Ymin, Ymax

      Do i = 1, 2
        w(i,1) = p1(i)
        w(i,2) = p2(i)
        w(i,3) = p3(i)
      Enddo
      Xmin = p1(1)
      Xmax = Xmin
      Ymin = p1(2)
      Ymax = Ymin
      Do i = 2, 3
        if (w(1,i).le.Xmin) Xmin = w(1,i)
        if (w(1,i).ge.Xmax) Xmax = w(1,i)
        if (w(2,i).le.Ymin) Ymin = w(2,i)
        if (w(2,i).ge.Ymax) Ymax = w(2,i)
      Enddo
      Return
      End
c.....
c                Subroutine EMIT1
c.....
c   This subroutine emits energy bundles in random direction from
c   random location on a rectangular surface PERPENDICULAR to the
c   x,z-plane.
c.....
      Subroutine EMIT1 (p, dc, ns, omega)
      Implicit double precision (a-h, o-z)
      Dimension p(3),dc(3),cXmin(51),cXmax(51),cYmin(51),cYmax(51)

c Put frequently used variables in one storage block :
      Common / bl11 / cXmin, cXmax, cYmin, cYmax
      Common / block7/ pi,x,y,z,dL,dm,dn,x2,y2,z2,dL2,dm2,dn2

c Define random site of emiision in the global coordinate system:
      x = (cXmax(ns)-cXmin(ns))*RANDO () + cXmin(ns)
      y = (cYmax(ns)-cYmin(ns))*RANDO () + cYmin(ns)
      z = p(3) - (x-p(1)) * dc(1) / dc(3)

c Direction of emission:
      Call DIREC1 (dc, omega)

      Return
      End

c.....
c                Subroutine DIREC1
c.....
c   This subroutine defines the direction of emission from a point on
c   a surface area PERPENDICULAR to the x,z-plane.
c   Omega(i) is the angle between the x'-direction of the surface
c   and the x-direction of the global coordinate sytem.
c.....
      Subroutine DIREC1 (omega)
```

```
      Implicit double precision (a-h, o-z)
c
c  Put frequently used variables in common blocks:
c
      Common / block7 / pi,x,y,z,dl,dm,dn,x2,y2,z2,dl2,dm2,dn2
c
c  Define the circumferential angle PHI and the cone angle THETA:
c
        rt = dsqrt( RANDO () )
        theta = dasin(rt)
        phi = 2.00d0 * pi * RANDO ()
c
c  Define the direction of emission in the local coordinate system:
c
        dnp = dcos(theta)
        dmp = dcos(phi) * dsqrt( 1.00d0 - dnp**2 )
        If ( (phi.GE.0.00d0).AND.(phi.LE.pi)) then
          dlp = - dsqrt( 1.00d0 - dmp**2 - dnp**2 )
        Else
          dlp =   dsqrt( 1.00d0 - dmp**2 - dnp**2 )
        Endif
c
c  Direction of emission in the global coordinate system:
c
        dl = dlp*dcos(omega) + dnp*dsin(omega)
        dm = dmp
        dn = -dlp*dsin(omega) + dnp*dcos(omega)

      Return
      End
c.....
c                 Subroutine EMIT2
c.....
c    This subroutine emits energy bundles,in random direction and from
c    random location,on a rectangular surface area PARALLEL to the
c    x,z-plane.
c.....
      Subroutine EMIT2 (pa, pb, pc, dcy)
      Implicit double precision (a-h, o-z)
      Dimension pa(3), pb(3), pc(3)
c
c  Put frequently ysed variables in one storage blocks:
c
      Common / block7 / pi,x,y,z,dl,dm,dn,x2,y2,z2,dl2,dm2,dn2
c
c  Define the random location of emission in the local coord. syst.:
c
      rc = RANDO ()
      rb = RANDO ()
c
c  Random location in the global coordinate system:
c
      x = pa(1) + rc*(pc(1)-pa(1)) + rb*(pb(1)-pa(1))
      y = pa(2)
      z = pa(3) + rc*(pc(3)-pa(3)) + rb*(pb(3)-pa(3))
c
c  Random direction in the global coordinate system:
```

```
c
      Call DIREC2 (dcy)

      Return
      End
c.....
c                  Subroutine EMIT3
c.....
c    This subroutine emits energy bundles,in random direction and from
c    random location on a TRIANGULAR surface area PARALLEL to the
c    x,z-plane. The emitting surface is defined by 3 points a, b, and
c    c forming the system {a, AB, AC}.
c.....
      Subroutine EMIT3 (pa, pb, pc, dcy)
      Implicit double precision (a-h, o-z)
      Dimension pa(3), pb(3), pc(3)
c
c  Put frequently used varaiables in one unique block:
c
      Common / block7 / pi,x,y,z,dl,dm,dn,x2,y2,z2,dl2,dm2,dn2
c
c  Define the random location in the local coordinate system:
c
      rb = RANDO ()
      rc = RANDO ()
      If ((rb+rc).GT.1.00d0) then
c
c          if the point is on the right of the area (on the right of
c          the line BC), then flip the point over to the left side
c          using a symmetry that would be an orthogonal symmetry with
c          respect to the line BC if {a, AB, CA} was an orthonormed system:
c
           rb = 1.00d0 - rb
           rc = 1.00d0 - rc
           ra = rb
           rb = rc
           rc = ra
c
c          this is aonter way to swap rB and rC without using the intermediate
c          variable rA :
c          rB = rB + rC
c          rC = rB - rC
c          rB = rB - rC
      Endif
c
c  Define the random location of emission in the global coordinate system:
c
      x = pa(1) + rb*(pb(1)-pa(1)) + rc*(pc(1)-pa(1))
      y = pa(2)
      z = pa(3) + rb*(pb(3)-pa(3)) + rc*(pc(3)-pa(3))
c
c  Define the random direction of emission in the global coord. system :
c
      Call DIREC2 (dcy)

      Return
      End
c.....
```

```
c                Subroutine DIREC2
c.....
c    This subroutine defines the random direction of emission from a
c    surface area parallel to the x,z-plane.
c    dcy : the y-component of the surface normal vector
c.....
      Subroutine DIREC2 (dcy)
      Implicit double precision (a-h, o-z)
c
c  Put frequently used variables in one storage block:
c
      Common / block7 / pi,x,y,z,dl,dm,dn,x2,y2,z2,dl2,dm2,dn2
c
c  Randomly define the azimuth angle, Theta, and the circumferential
c  angle, phi:
c
      rt = dsqrt(RANDO ())
      theta = dasin (rt)
      phi = 2.00d0 * pi * RANDO ()
c
c  Direction of emission in the global coordinate system:
c
      dm = dcos(theta) * dcy
      If (pi/2.00d0.lt.phi.and.phi.lt.1.50d0*pi) then
         dl = - dcos(phi) * dsqrt ( 1.00d0 - dm**2 )
      Else
         dl =   dcos(phi) * dsqrt ( 1.00d0 - dm**2)
      Endif
      If ((phi.GE.0.00d0).AND.(phi.LE.pi)) then
         dn = + dsqrt ( 1.00d0 - dl**2 - dm**2 )
      Else
         dn = - dsqrt ( 1.00d0 - dl**2 - dm**2)
      Endif

      Return
      End


c.....
c                Subroutine EMIT4
c.....
c    This subroutine defines the random site, and the direction
c    cosines of solar emission.
c.....
      Subroutine EMIT4 (p, dc, ns)
      Implicit double precision (a-h, o-z)
      Dimension p(3),dc(3),cXmin(51),cXmax(51),cYmin(51),cYmax(51)
c
c  Put frequently used variables in common blocks:
c
      Common / bl11 / cXmin, cXmax, cYmin, cYmax
      Common / block7/ pi,x1,y1,z1,dL1,dm1,dn1,x2,y2,z2,dL2,dm2,dn2
      Common / bl24 / dlsun, dmsun, dnsun
c
c  Define the random site of emission in the global coordinate system:
c
     x1 = (cXmax(ns)-cXmin(ns))* RANDO () + cXmin(ns)
```

```
      y1 = (cYmax(ns)-cYmin(ns))* RANDO () + cYmin(ns)
      z1 = p(3) - (x1-p(1))*dc(1)/dc(3)
c
c  Redefine the direction of emission in the global coordinate system:
c
      dl1 = dlsun
      dm1 = dmsun
      dn1 = dnsun

      Return
      End

c.....
c                Subroutine EMIT5
c.....
c  This subroutine defines the random site and direction of emission
c  for a rectangular surface perpendicular to the x,z-plane and having
c  a circular hole of radius Rint, centered at xo, zo and yo = 0.00
c.....
      Subroutine EMIT5 (p, dc, ns, omega)
      Implicit double precision (a-h, o-z)
      Dimension p(3),dc(3),cXmin(51),cXmax(51),cYmin(51),cYmax(51)
     *        , po(3)
c
c  Put frequently used variables in one storage block :
c
      Common / bl11 / cXmin, cXmax, cYmin, cYmax
      Common / block7/ pi,x,y,z,dL,dm,dn,x2,y2,z2,dL2,dm2,dn2
      Common / bl20 / a6, po, rint
c
c  Define random site of emiision in the global coordinate system:
c
  1   Continue
      x = (cXmax(ns)-cXmin(ns))*RANDO () + cXmin(ns)
      y = (cYmax(ns)-cYmin(ns))*RANDO () + cYmin(ns)
      z = p(3) - (x-p(1)) * dc(1) / dc(3)
c
c  Transform the coordinate of the point to the local coordinate system:
c
      xa = (x2-po(1))*dcos(a6) + (z2-po(3))*dsin(a6)
c
c  If the site is inside the circular hole, redefine another:
c
      c = xa**2 + y**2
      If (c.lt.rint**2) goto 1
c
c     Direction of emission:
c
      Call DIREC1 (dc, omega)
c
      Return
      End

c.....
c                Subroutine EMIT6
c.....
c     This subroutine emits energy bundles in random direction from
```

```
c    random location from a reflector of the baffle.
c.....
      Subroutine EMIT6 (p, dc, omega, nre, p1, p2, p3, p4, p5, p6)
      Implicit double precision (a-h, o-z)
      Dimension p(3),dc(3),p1(3),p2(3),p3(3),p4(3),p5(3),p6(3)
     *         ,nresar(4)
c
c Put frequently used variables in common blocks:
c
      Common / block7/ pi,x,y,z,dL,dm,dn,x2,y2,z2,dL2,dm2,dn2
      Common / bl27 / Nresar
c
c Define the sub-surface of emission:
c
      If (nre.le.nresar(1)) then
c
c        emission from the bottom sub-surface:
c
        x = (p1(1)-p2(1))*RANDO () + p2(1)
        y = (p4(2)-p1(2))*RANDO () + p1(2)
        z = p(3) - (x-p(1)) * dc(1) / dc(3)
      Else if (nre.le.nresar(2)) then
c
c        emission from the top sub-surface:
c
        x = (p1(1)-p2(1))*RANDO () + p2(1)
        y = (p3(2)-p6(2))*RANDO () + p6(2)
        z = p(3) - (x-p(1)) * dc(1) / dc(3)
      Else if (nre.le.nresar(3)) then
c
c        emissiom from the left sub-surface:
c
        x = (p5(1)-p2(1))*RANDO () + p2(1)
        y = (p6(2)-p5(2))*RANDO () + p5(2)
        z = p(3) - (x-p(1)) * dc(1) / dc(3)
      Else
c
c        emission from the right sub-surface:
c
        x = (p1(1)-p4(1))*RANDO () + p4(1)
        y = (p6(2)-p5(2))*RANDO () + p5(2)
        z = p(3) - (x-p(1)) * dc(1) / dc(3)
      Endif
c
c Define the direction of emission:
c
      Call DIREC1 (dc, omega)
      Return
      End


c.....
c                Subroutine SENSOR
c.....
c    This subroutine checks if the point of incidence is inside the
c    sensor-aperture.
```

```
c.....
      Subroutine SENSOR
      Implicit double precision (a-h, o-z)
      Logical APER
c
c  Put frequently used variables in common blocks:
      Common / bl18 / a2, rsa
      Common / bl19 / aper
      Common / block7 / pi,x,y,z,dl,dm,dn,x2,y2,z2,dl2,dm2,dn2
c
c  Coordinates in the local coordinate system:
c  (Note that I omit the translation along the z'-direction, because
c  I am interested only in the projection in the x',y'-plane)
c
      xa = x2*dcos(a2) + z2*dsin(a2)
*     ya = y2
c
c  Check if the point is inside the circle of radius Rsa, centered at
c  the origin of the local coordinate system :
c
      If ( (xa**2 + y2**2).LE.(rsa**2) ) then
          APER = .true.
      Else
          APER = .false.
      Endif
      Return
    End
c.....
c              Subroutine APERTU
c.....
c  This subroutine checks if a point is inside the MAM aperture
c.....
      Subroutine APERTU
      Implicit double precision (a-h, o-z)
      Dimension p(3)
      Logical aper
c
c  Put frequently used variables in common blocks:
      Common / block7 / pi,x,y,z,dl,dm,dn,x2,y2,z2,dl2,dm2,dn2
      Common / bl19 / aper
      Common / bl20 / a6, p, rint
c
c  Transform the coordinate of the point to the local coordinate system :
      xa = (x2-p(1))*dcos(a6) + (z2-p(3))*dsin(a6)
*     ya = y2
c  ( I ignored the determination of za because it should be za = 0.00.
c    Also, note the difference from SENSOR because the translation is
c    not along the z'-direction)
c
c  Check if the point is inside the circle of radius Rint and centered
c  at the origin of the local coordinates system:
      If ((xa**2 + y2**2).LE.Rint**2) then
          aper = .true.
      Else
          aper = .false.
      Endif
      Return
```

```
      End


c.....
c              Subroutine SWAP12
c.....
c              Point 1 = point 2
c.....
      Subroutine SWAP12
      Implicit double precision (a-h, o-z)
      Common / block7 /pi,x,y,z,dl,dm,dn,x2,y2,z2,dl2,dm2,dn2

        x = x2
        y = y2
        z = z2
        dl = dl2
        dm = dm2
        dn = dn2
      Return
      End
c.....
c              Subroutine SPEREF
c.....
c    This subroutine defines the direction cosines of specular
c    reflection. dcx, dcy, dcz are the 3 components of the normal
c    unit vector of the incident surface.
c.....
      Subroutine SPEREF (dcx,dcy,dcz)
      Implicit double precision (a-h, o-z)
c
c  Put frequently used variables in common block:
      Common / block7 / pi,x1,y1,z1,dl1,dm1,dn1,x2,y2,z2,dl2,dm2,dn2
c
c  direction cosines of the specular reflection :
      a   = dl1*dcx + dm1*dcy + dn1*dcz
      dl2 = dl1 - 2.00d0*dcx*a
      dm2 = dm1 - 2.00d0*dcy*a
      dn2 = dn1 - 2.00d0*dcz*a
c
c  Normalize these direction cosines to guarantee that
c  dl**2 + dm**2 + dn**2 = 1.00    :
c
      b = dsqrt(dl2**2 + dm2**2 + dn2**2)
      dl1 = dl2 / b
      dm1 = dm2 / b
      dn1 = dn2 / b
      Return
      End


c.....
c    Write the coordinate of a vector to unit 1
c.....
      Subroutine PRINT1 (p, n)
      Implicit double precision (a-h, o-z)
      Dimension p(3)
```

```
        Write (1,10) p(1), p(2), p(3), n
 10     format (3(2x, f7.3), 2x, i3)
      Return
      End
c.....
c    define how nodes in unit 1 are connected
c.....
      Subroutine PRINT4 (n1, n2, n3, n4)
      Implicit double precision (a-h, o-z)
        Write (1,10) n1, n2, n3, n4
 10     Format (4(2x,i3))
      Return
      End
c.....
c Define the Xmin, Xmax, Ymin, Ymax from the coordinates of 3 points .
c.....
      Subroutine XYMIMA (p1, p2, p3)
      Implicit double precision (a-h, o-z)
      Dimension p1(3), p2(3), p3(3), w(2,3)
c Put frequently used variables in common block:
      Common / bl1 / Xmin, Xmax, Ymin, Ymax

      Do i = 1, 2
        w(i,1) = p1(i)
        w(i,2) = p2(i)
        w(i,3) = p3(i)
      Enddo
      Xmin = p1(1)
      Xmax = Xmin
      Ymin = p1(2)
      Ymax = Ymin
      Do i = 2, 3
        If (w(1,i).le.Xmin) Xmin = w(1,i)
        If (w(1,i).ge.Xmax) Xmax = w(1,i)
        If (w(2,i).le.Ymin) Ymin = w(2,i)
        If (w(2,i).ge.Ymax) Ymax = w(2,i)
      Enddo
      Return
      End
```

```
c.....
c                Subroutine SUBREC
c.....
c This subroutine subdivides a rectangular surface perpendicular to
c the x,z-plane into iN*jM equal-rectangular-area elements and deter-
c mines the coordinates of the 4 corners of each area element, then
c the distribution factor of the element is normalized by the surface
```

```
c area of the element.
c The above data are then written to unit NF , which is used by
c TECPLOT to visualize the distribution of energy on the surface.
c.....
      Subroutine SUBREC (p1,p2,p3,dc,ns,iN,jM,iflag,title,nf)
      Implicit double precision (a-h, o-z)
      Character*8 title
      Dimension p1(3), p2(3), p3(3), dc(3), c(800,4,3), Nra(51,800)
     *        , dfac(51,800)

c Put frequently used variables in common blocks :
      Common / bl1 / Xmin, Xmax, Ymin, Ymax
      Common / bl9 / Nra, dfac, Ntre


c Define the maxima and the minima of the surface:
      Call XYMIMA (P1,P2,P3)
      Zmax = dc(1)/dc(3)*(p1(1)-Xmax)+p1(3)
      Zmin = dc(1)/dc(3)*(p1(1)-Xmin)+p1(3)

c Surface area of each area element:
      dNtre = dble (Ntre)
      Dde = dNtre
     *      * (Ymax-Ymin) * dsqrt((Xmax-Xmin)**2+(Zmax-Zmin)**2)
     *      / (iN*jM)


c Set the number of area elements to be represented according to
c the value of iflag :
      If (iflag.EQ.1) then
c       the surface is completely represented :
        jmin = 1
        jmax = jM
      Else if (iflag.EQ.2) then
c       only the left side of the surface is represented :
        jmin = jM/2
        jmax = jM
      Else if (iflag.EQ.3) then
c       only the right side of the surface is represented :
        jmin = 1
        jmax = jM/2
      Endif
c
c Define zone for TECPLOT :
c
c   Number of quadrilateral elements :
      numele = iN*(jmax-jmin+1)
c   Number of points :
      numpoi = 4 * numele
c   write the title :
      Write (nf,30) title, numpoi, numele
 30   format ('Zone T ="',a8,'", i = ',i3,', j=',i3,', f = fepoint')

c Define the coordinates of the 4 corners , and the normalized
c distribution factor of the area element:
```

```
      Do 1 j = jmin, jmax
       Do 2 i = 1, iN
c          nae : number of the area element
         nae = (j-1)*iN + i

c          Coordinates of corner 1 :
         c(nae,1,1) = (i-1)*(Xmax-Xmin)/iN + Xmin
         c(nae,1,2) = (j-1)*(Ymax-Ymin)/jM + Ymin
         c(nae,1,3) = dc(1)/dc(3)*(p1(1)-c(nae,1,1)) + p1(3)

c          Coordinates of corner 2 :
         c(nae,2,1) = i*(Xmax-Xmin)/iN + Xmin
         c(nae,2,2) = c(nae,1,2)
         c(nae,2,3) = dc(1)/dc(3)*(p1(1)-c(nae,2,1)) + p1(3)

c          Coordinates of corner 3 :
         c(nae,3,1) = c(nae,2,1)
         c(nae,3,2) = j*(Ymax-Ymin)/jM + Ymin
         c(nae,3,3) = c(nae,2,3)

c          Coordinates of corner 4 :
         c(nae,4,1) = c(nae,1,1)
         c(nae,4,2) = c(nae,3,2)
         c(nae,4,3) = c(nae,1,3)

c  Normalize the distribution factor for each area:
         dnu = dble (nra(ns,nae))
         dfac(ns,nae) = dnu / dde

c  Write the coordinates of the 4 corners and the normalized
c  distribution factor of the area element to unit nf :
         Write (nf,10) c(nae,1,1),c(nae,1,2),c(nae,1,3),dfac(ns,nae)
         Write (nf,10) c(nae,2,1),c(nae,2,2),c(nae,2,3),dfac(ns,nae)
         Write (nf,10) c(nae,3,1),c(nae,3,2),c(nae,3,3),dfac(ns,nae)
         Write (nf,10) c(nae,4,1),c(nae,4,2),c(nae,4,3),dfac(ns,nae)
   10    Format (3(2x,f7.3),2x,e10.4)

   2     Continue
   1   Continue

c  Define how points in this file are connected:
      Do j = 1, 4*iN*(jmax-jmin+1), 4
         Write (nf,20) j, j+1, j+2, j+3
      Enddo
   20  Format (4(2x,i4))

      Return
      End
c.....
c                Subroutine SUBRE2
c.....
c  This subroutine subdivides a rectangular surface perpendicular to
c  the x,z-plane into iN*jM equal-rectangular-area elements and deter-
c  mines the coordinates of the 4 corners of each area element, then
```

```
c the distribution factor of the element is normalized by the surface
c area of the element.
c The above data are then written to unit NF , which is used by
c TECPLOT to visualize the distribution of energy on the surface.
c.....
      Subroutine SUBREC (Zmin,Zmax,p1,p2,p3,dc,ns,iN,jM,iflag,title,nf)
      Implicit double precision (a-h, o-z)
      Character*8 title
      Dimension p1(3), p2(3), p3(3), dc(3), c(800,4,3), Nra(51,800)
     *        , dfac(51,800)

c Put frequently used variables in common blocks :
      Common / bl1 / Xmin, Xmax, Ymin, Ymax
      Common / bl9 / Nra, dfac, Ntre

c Surface area of each area element:
      dNtre = dble (Ntre)
      Dde = dNtre
     *      * (Ymax-Ymin) * dsqrt((Xmax-Xmin)**2+(Zmax-Zmin)**2)
     *      / (iN*jM)


c Set the number of area elements to be represented according to
c the value of iflag :
      If (iflag.EQ.1) then
c       the surface is completely represented :
        jmin = 1
        jmax = jM
      Else if (iflag.EQ.2) then
c       only the left side of the surface is represented :
        jmin = jM/2
        jmax = jM
      Else if (iflag.EQ.3) then
c       only the right side of the surface is represented :
        jmin = 1
        jmax = jM/2
      Endif
c
c Define zone for TECPLOT :
c
c   Number of quadrilateral elements :
      numele = iN*(jmax-jmin+1)
c   Number of points :
      numpoi = 4 * numele
c   write the title :
      Write (nf,30) title, numpoi, numele
 30   Format ('Zone T ="',a8,'", i = ',i3,', j=',i3,', f = fepoint')

c Define the coordinates of the 4 corners , and the normalized
c distribution factor of the area element:
      Do 1 j = jmin, jmax
        Do 2 i = 1, iN
c         nae : number of the area element
          nae = (j-1)*iN + i
```

```
c          Coordinates of corner 1 :
           c(nae,1,1) = (i-1)*(Zmax-Zmin)/iN + Zmin
           c(nae,1,2) = (j-1)*(Ymax-Ymin)/jM + Ymin
           c(nae,1,3) = dc(3)/dc(1)*(p1(3)-c(nae,1,3)) + p1(1)

c          Coordinates of corner 2 :
           c(nae,2,1) = i*(Zmax-Zmin)/iN + Zmin
           c(nae,2,2) = c(nae,1,2)
           c(nae,2,3) = dc(3)/dc(1)*(p1(3)-c(nae,2,3)) + p1(1)

c          Coordinates of corner 3 :
           c(nae,3,1) = c(nae,2,1)
           c(nae,3,2) = j*(Ymax-Ymin)/jM + Ymin
           c(nae,3,3) = c(nae,2,3)

c          Coordinates of corner 4 :
           c(nae,4,1) = c(nae,1,1)
           c(nae,4,2) = c(nae,3,2)
           c(nae,4,3) = c(nae,1,3)

c Normalize the distribution factor for each area:
           dnu = dble (nra(ns,nae))
           dfac(ns,nae) = dnu / dde

c Write the coordinates of the 4 corners and the normalized
c distribution factor of the area element to unit nf :
           Write (nf,10) c(nae,1,1),c(nae,1,2),c(nae,1,3),dfac(ns,nae)
           Write (nf,10) c(nae,2,1),c(nae,2,2),c(nae,2,3),dfac(ns,nae)
           Write (nf,10) c(nae,3,1),c(nae,3,2),c(nae,3,3),dfac(ns,nae)
           Write (nf,10) c(nae,4,1),c(nae,4,2),c(nae,4,3),dfac(ns,nae)
 10        Format (3(2x,f7.3),2x,e10.4)

 2         Continue
 1       Continue

c Define how points in this file are connected:
         Do j = 1, 4*iN*(jmax-jmin+1), 4
           Write (nf,20) j, j+1, j+2, j+3
         Enddo
 20    Format (4(2x,i4))

         Return
         End


c.....
c                    Subroutine SUBREP
c.....
c This subroutine subdivides a rectangular surface PARALLEL to the
c x,z-plane into iN*jM equal-area elements and normalizes the distri-
c bution factor of each area element by the surface area of that ele-
c ment. The above data are then written to a file for visual represen-
c tation by TECPLOT.
c.....
         Subroutine SUBREP (pa, pb, pc, ns, iN, jM, title,nf)
```

```
      Implicit double precision (a-h, o-z)
      Character*8 title
      Dimension pa(3),pb(3),pc(3),cl(800,4,3),cg(800,4,3),Nra(51,800),
     *          dfac(51,800)
c
c  Put frequently used variables in common blocks :
c
      Common / bl9 / Nra, dfac, Ntre
c
c  Define the angle of rotation alpha that makes the LCS coincide
c  with the GCS, and the dimension of the surface :
c
      alpha = datan2 ( (pc(3)-pa(3)), (pc(1)-pa(1)) )
      ac = ( dsqrt ( (pc(1)-pa(1))**2 + (pc(3)-pa(3))**2 ) )/iN
      ab = ( dsqrt ( (pb(1)-pa(1))**2 + (pb(3)-pa(3))**2 ) )/jM
c
c  Surface area of each element:
c
      dntre = dble (ntre)
      dde = dntre * ab * ac
c
c  Define zone for TECPLOT :
c
c     Number of quadrilateral elements :
      numele = iN*jM
c     Number of points :
      numpoi = 4 * numele
c     write the title :
      Write (nf,30) title, numpoi, numele
 30   Format ('Zone T ="',a8,'", i = ',i3,', j=',i3,', f = fepoint')
c
c  Define the coordinates of the corners in the LCS:
c      (nae : number of the area element)
      Do 1 j = 1, jM
       Do 2 i = 1, iN
        nae = (j-1)*iN + i
        Do 3 L = 1, 4
          cl(nae,L,2) = 0.00d0
 3        Continue
        cl(nae,1,1) = (i-1)*ac
        cl(nae,1,3) = (j-1)*ab
        cl(nae,2,1) = i*ac
        cl(nae,2,3) = cl(nae,1,3)
        cl(nae,3,1) = cl(nae,2,1)
        cl(nae,3,3) = j*ab
        cl(nae,4,1) = cl(nae,1,1)
        cl(nae,4,3) = cl(nae,3,3)
c
c  Define the coordinates of the corners in the global coordinate
c  system :
        Do 6 L = 1, 4
          cg(nae,L,1) = cl(nae,L,1)*dcos(alpha) -
     *                  cl(nae,L,3)*dsin(alpha) + pa(1)
          cg(nae,L,2) = pa(2)
          cg(nae,L,3) = cl(nae,L,1)*dsin(alpha) +
```

```
     *                    cl(nae,L,3)*dcos(alpha) + pa(3)
 6          Continue
c
c  Define the distribution factor and normalize it by the surface area :
c
       dnu = dble (nra(ns,nae))
       dfac(ns,nae) = dnu/dde
c
c  Write the coordinates of the corners to file defined in unit nf:
c
       Do 4 L = 1, 4
        Write (nf,10) cg(nae,L,1), cg(nae,L,2),
     *                    cg(nae,L,3), dfac(ns,nae)
 4          Continue
 2        Continue
 1      Continue
c
c  Define how points in this file are connected:
c
       Do 5 j = 1, 4*iN*jM, 4
         Write (nf,20) j, j+1, j+2, j+3
 5          Continue
 10   Format (3(2x,f7.3),2x,e10.4)
 20   Format (4(2x,i4))

      Return
      End



c.....
c                 Subroutine SUBCIR
c.....
c  This subroutine subdivides a circular surface, of radius RSA and
c  centered at a distance B9 away from the origin O, into Nr*Nphi
c  equal-area elements,where Nr is the number of radial subdivision
c  and Nphi is the number of circumferential subdivision. Particularly,
c  the circular surface is lying on the x',y-plane which is at an angle.
c  A2 (counter-clockwise) from the x,y-plane.
c.....
      Subroutine SUBCIR (b9,rsa,a2,Nr,Nphi,ns,title,nf)
      Implicit double precision (a-h, o-z)
      Character*8 title
      Dimension rsap(6),cl(800,4,3),cg(800,4,3),Nra(51,800),dfac(51,800)
c
c  Put frequently used variables in commom blocks :
c
      Common / bl9  / Nra,dfac,ntre
      Common / bl22 / rsap
c
      pi = dacos(-1.00d0)

c
c  Define zone for TECPLOT :
c
c    Number of quadrilateral elements :
      numele = Nr*Nphi
```

```
c    Number of points :
      numpoi = 4 * numele
c    write the title :
      Write (nf,30) title, numpoi, numele
  30  Format ('Zone T ="',a8,'", i = ',i3,', j=',i3,', f = fepoint')
c
c  Surface area of each element :
c
      dntre = dble (ntre)
      nsub = Nr*Nphi
      dnsub = dble (nsub)
      dde = dntre *pi*rsa**2 / dnsub
c
c  Define the angle Phi of circumferential subdivision :
c
      dnphi = dble (Nphi)
      Phi = 2.00d0 * pi / dnphi
c
c  Define the coordinates of the corners of the area element :
c
      Do 1 j = 1, Nr
       Do 2 i = 1, Nphi
c        (nae : number of the area element)
     nae = (j-1) * Nphi + i
c
c        In the local coordinate system :
c
         Do 3 L = 1, 4
           cl(nae,L,3) = 0.00d0
   3     Continue
         cl(nae,1,1) = rsap(j) * dcos( phi*(i-1) )
         cl(nae,1,2) = rsap(j) * dsin( phi*(i-1) )
         cl(nae,2,1) = rsap(j+1) * dcos( phi*(i-1) )
         cl(nae,2,2) = rsap(j+1) * dsin( phi*(i-1) )
         cl(nae,3,1) = rsap(j+1) * dcos( phi*i )
         cl(nae,3,2) = rsap(j+1) * dsin( phi*i )
         cl(nae,4,1) = rsap(j) * dcos( phi*i )
         cl(nae,4,2) = rsap(j) * dsin( phi*i )
c
c        in the global coordinate system :
c
         Do 6 L = 1, 4
           cg(nae,L,1) = cl(nae,L,1)*dcos(a2) + b9*dsin(a2)
           cg(nae,L,2) = cl(nae,L,2)
           cg(nae,L,3) = cl(nae,L,1)*dsin(a2) - b9*dcos(a2)
   6     Continue
c
c        Define the distribution factor and normalize it by the
c        surface area of the element :
c
         dnu = dble( nra(ns,nae) )
         dfac(ns,nae) = dnu / dde
c
c        Write the coordinates of the corners and the normalized
c        distribution factor of the element to unit nf :
c
         Do 4 L = 1, 4
```

```
            Write (nf,10) cg(nae,L,1),cg(nae,L,2),cg(nae,L,3),
     *                    dfac(ns,nae)
  4      Continue
  2     Continue
  1    Continue
c
c  Define how nodes in unit nf are connected:
c
      Do 5 i = 1, 4*Nr*Nphi, 4
         Write (nf,20) i, i+1, i+2, i+3
  5    Continue
  10   Format (3(2x,f7.3),2x,e10.4)
  20   Format (4(2x,i4))

      Return
      End

c.....
c                Subroutine SUBQUA
c.....
c  This subroutine subdivides a triangular surface, parallel to the x,z-
c  plane and defined by 3 points A, B, and C, into quadrilateral and
c  triangular area elements (the latters are adjacent to the BC-side),
c  then defines the coordinates of the corners,the surface area and the.
c  distribution factor normalized by the surface area for each element.
c  The results will be stored in unit NF for data visualisation by
c  TECPLOT.
c  Note : The subdivision along AB is the same as that along AC
c.....
      Subroutine SUBQUA (a, b, c, n, ns, title, nf)
      Implicit double precision (a-h, o-z)
      Dimension a(3), b(3), c(3), cl(800,4,3), cg(800,4,3),
     *          Nra(51,800), dfac(51,800), sae(800)
c
c  Put frequently used variables in common block :
c
      Common / bl9 / Nra, dfac, Ntre
c
c  Define the characteristics of the surface :
c
      ac = (dsqrt( (c(3)-a(3))**2 + (c(1)-a(1))**2 )) / N
      ab = (dsqrt( (b(3)-a(3))**2 + (b(1)-a(1))**2 )) / N
      alpha = datan2( (c(3)-a(3)) , (c(1)-a(1)) )
      beta  = datan2( (b(3)-a(3)) , (b(1)-a(1)) )
c
c  Define the total number of area elements:
c
      Nae = N / 2 * (N + 1)
c
c  Define zone for TECPLOT :
c
c     Number of quadrilateral elements :
      numele = Nae
c     Number of points :
      numpoi = 4 * numele
c     write the title :
      Write (nf,30) title, numpoi, numele
  30  Format ('Zone T ="',a8,'", i = ',i3,', j=',i3,', f = fepoint')
```

```
c
c Define the coordinates of the corners in the local coordinate
c system :
c      ( Ne : Number of the area element )
       Nemax = 0
       Do 1 j = 1, N
        Nemin = Nemax + 1
        Nemax = Nemax + N - j + 1
        Do 3 Ne = Nemin, Nemax
c         (The area element is parallel to the x,z-plane, so the
c          y-coordinate is the same for the 4 corners)
          CL(Ne,1,2) = A(2)
          CL(Ne,2,2) = A(2)
          CL(Ne,3,2) = A(2)
          CL(Ne,4,2) = A(2)
          If (Ne.NE.Nemax) then
c           (The area elements are quadrilateral)
c            Corner 1 :
            CL(Ne,1,1) = (Ne-Nemin)*AC
            CL(Ne,1,3) = (j-1)*AB
c            Corner 2 :
            CL(Ne,2,1) = (Ne-Nemin+1)*AC
            CL(Ne,2,3) = CL(Ne,1,3)
c            Corner 3:
            CL(Ne,3,1) = CL(Ne,2,1)
            CL(Ne,3,3) = j*AB
c            Corner 4:
            CL(Ne,4,1) = CL(Ne,1,1)
            CL(Ne,4,3) = CL(Ne,3,3)
c
c            Surface Area of the element:
c
            SAE(Ne) = AB*AC

          Else
c           (Ne=Nemax.i.e.,The area element is triangular)
c            Corner 1:
            CL(Ne,1,1) = (Ne-Nemin)*AC
            CL(Ne,1,3) = (j-1)*AB
c            Corner 2:
            CL(Ne,2,1) = (Ne-Nemin+1)*AC
            CL(Ne,2,3) = CL(Ne,1,3)
c            Corner 3:
            CL(Ne,3,1) = CL(Ne,1,1)
            CL(Ne,3,3) = j*AB
c            Corner 4:
            CL(Ne,4,1) = CL(Ne,3,1)
            CL(Ne,4,3) = CL(Ne,3,3)
c
c            Surface area of the element:
c
            SAE(Ne) = AB*AC/2.00d0

          Endif
c    Transform the coordinates of the corners to the GCS:
```

```
      Do 4 L = 1, 4
         CG(Ne,L,1) = CL(Ne,L,1)*dcos(alpha) +
     *                CL(Ne,L,3)*dcos(beta) + A(1)
         CG(Ne,L,2) = CL(Ne,L,2)
         CG(Ne,L,3) = CL(Ne,L,1)*dsin(alpha) +
     *                CL(Ne,L,3)*dsin(beta) + A(3)
  4       Continue
  3      Continue
  1    Continue
c
c  Total number of rays emitted in double precision:
c
       Dntre=DBLE(Ntre)
c
c  Write to file in unit nf:
c
      Do 6 j = 1, Nae
       Dnu = Dble(NRA(ns,j))
       Dde = Dntre*SAE(j)
       Dfac(ns,j) = Dnu/Dde
       Do 7 L = 1, 4
         Write (NF,10) CG(j,L,1),CG(j,L,2),CG(j,L,3),Dfac(ns,j)
  7    Continue
  6   Continue
  10   Format (3(2x,f7.3),2x,e10.4)
c
c  Define the order of connection:
c
      Do 8 k = 1, nae*4, 4
       Write (NF,20) k, k+1, k+2, k+3
  8   Continue
  20   Format (4(2x,i3))
      Return
      End
```

```
c.....
c              Subroutine SUBTRA
c.....
c    (the length of one side of the triangle is considerably less
c     than that of the two other sides)
c  This subroutine subdivides a triangular surface, parallel to the x,z-
c  plane and defined by 3 points A, B, and C, into quadrilateral, tra-
c  pezoidal and triangular area elements (the latters are adjacent to
c  the BC-side), then defines the coordinates of the corners,the surf-
c  ace area and the distribution factor normalized by the surface area
c  for each element.The results will be stored in unitt NF for data
c  visualization by TECPLOT.
c.....
      Subroutine SUBTRA (a,b,c,ns,iN,jM,title,nf)
      Implicit double precision (a-h, o-z)
      Character*8 title
```

```
      Dimension a(3),b(3),c(3),cl(800,4,3),cg(800,4,3),
     *          Nra(51,800),dfac(51,800),sae(800)
c
c  Put frequently used variables in common block :
c
      Common / bl9 / nra,dfac,ntre
c
c  Define the characteristics of the surface :
c
      ac = ( dsqrt( (c(3)-a(3))**2 + (c(1)-a(1))**2) ) /iN
      ab = ( dsqrt( (b(3)-a(3))**2 + (b(1)-a(1))**2) ) /jM
      alpha = datan2 ( (c(3)-a(3)) , (c(1)-a(1)) )
      beta  = datan2 ( (b(3)-a(3)) , (b(1)-a(1)) )
c
c  Define the total number of area elements:
c
      nd = - iN / jM
      la = iN + (jM - 1) * nd
      Ntae = jM * (iN + la) / 2
      DiN = dble (iN)
      DjM = dble (jM)
c
c  Define zone for TECPLOT :
c
c    Number of quadrilateral elements :
      numele = Ntae
c    Number of points :
      numpoi = 4 * numele
c    write the title :
      Write (nf,30) title, numpoi, numele
  30  Format ('Zone T ="',a8,'", i = ',i3,', j=',i3,', f = fepoint')
c
c  Define the coordinates of the corners in the local coordinate
c  system :
c       ( Nae : Number of the area element )
      Naemax=0
      Do 1 j = 1, jM
        Nrow = iN + (j-1) * nd
        Naemin = Naemax + 1
        Naemax = Naemax + Nrow
        Med = Naemin + (Nrow + nd) - 1
        Do 3 Nae = Naemin, Naemax
c         (The area element is parallel to the x,z-plane, so the
c         y-coordinate is the same for the 4 corners)
            cl(nae,1,2) = a(2)
            cl(nae,2,2) = a(2)
            cl(nae,3,2) = a(2)
            cl(nae,4,2) = a(2)

            If (Nae.LT.(med)) then
c           (The area elements are quadrilateral)
c           Corner 1 :
              cl(nae,1,1) = (nae-Naemin)*ac
              cl(nae,1,3) = (j-1)*ab
c           Corner 2 :
              cl(nae,2,1) = (nae-Naemin+1)*ac
```

```
                cl(nae,2,3) = cl(nae,1,3)
c           Corner 3 :
                cl(nae,3,1) = cl(nae,2,1)
                cl(nae,3,3) = j*ab
c           Corner 4 :
                cl(nae,4,1) = cl(nae,1,1)
                cl(nae,4,3) = cl(nae,3,3)
c           Surface Area of the element:
                SAE(Nae) = AB*AC

               Else if (nae.NE.naemax) then
c           (i.e.,The area element is trapezoidal)
c           corner 1 :
                cl(nae,1,1) = (nae-Naemin)*ac
                cl(nae,1,3) = (j-1)*ab
c           Corner 2 :
                cl(nae,2,1) = (nae-Naemin+1)*ac
                cl(nae,2,3) = cl(nae,1,3)
c           Corner 3 :
                cl(nae,3,1) = cl(nae,2,1)
                cl(nae,3,3) = ab*DjM/DiN*(Naemax-nae) + (j-1)*ab
c           Corner 4 :
                cl(nae,4,1) = cl(nae,1,1)
                cl(nae,4,3) = ab*DjM/DiN*(Naemax-nae+1) + (j-1)*ab
c           Surface area of the element :
                SAE(Nae) = ac*ab - (2*(Nae-Med)-1)*ab*DjM/DiN*ac/2.00d0
               Else
c           (The area element is triangular)
c           Corner 1 :
                cl(nae,1,1) = (nae-Naemin)*ac
                cl(nae,1,3) = (j-1)*ab
c           Corner 2:
                cl(nae,2,1) = (nae-naemin+1)*ac
                cl(nae,2,3) = cl(nae,1,3)
c           Corner 3:
                cl(nae,3,1) = cl(nae,1,1)
                cl(nae,3,3) = ab*DjM/DiN + (j-1)*ab
c           Corner 4:
                cl(nae,4,1) = cl(nae,3,1)
                cl(nae,4,3) = cl(nae,3,3)
c           Surface area of the element:
                SAE(Nae) = (ac/2.00d0)*(ab*DjM/DiN)
               Endif
c
c Transform the coordinates of the corners to the GCS:
c
          Do 4 L=1,4
             cg(nae,L,1) = cl(nae,L,1)*dcos(alpha) +
     *                   cl(nae,L,3)*dcos(beta) + A(1)
             cg(nae,L,2) = cl(nae,L,2)
             cg(nae,L,3) = cl(nae,L,1)*dsin(alpha) +
     *                   cl(nae,L,3)*dsin(beta) + A(3)
 4        Continue
 3       Continue
```

```
  1   Continue
c
c  Total number of rays emitted in double precision:
c
      Dntre=DBLE(Ntre)
c
c  Write to file in unit nf:
c
     Do 6 j = 1, Ntae
       Dnu = Dble(nra(ns,j))
       Dde = Dntre*sae(j)
       Dfac(ns,j) = Dnu / Dde
       Do 7 L = 1, 4
         Write (nf,10) cg(j,L,1),cg(j,L,2),cg(j,L,3),dfac(ns,j)
  7    Continue
  6   Continue
  10  Format (3(2x,f7.3),2x,e10.4)
c
c  Define the order of connection:
c
     Do 8 k = 1, ntae*4, 4
        Write (NF,20) k, k+1, k+2, k+3
  8   Continue
  20  Format (4(2x,i3))
     Return
     End

c.....
c              Subroutine SUBINT
c.....
c  This subroutine subdivides the internal wall into area elements
c.....
     Subroutine SUBINT (a, xo, zo, ns, rint, title, nf)
     Implicit double precision (a-h, o-z)
     Dimension pap(3),pbp(3),pcp(3),cl(800,4,3),cg(800,4,3)
    *       ,sae(800),dfac(51,800),Nra(51,800),atri(50)
c
c  Put frequently used variables in common blocks :
c
     Common / bl9 /  Nra, dfac, Ntre
     Common / bl23 / phi1,phi2,phi3,phi4,disab,disac,
    *            is,ie,js,je,in,jm,Nip
     Common / bl25 / pap, pbp, pcp
c
c  Characteristics of the surface :
c
     v = (pcp(1)-pap(1)) / iN
     u = (pbp(2)-pap(2)) / jM
     dex = ie*v + pap(1)
     dey = je*u + pap(2)
     dexm = dabs( is*v + pap(1) )
c
c  Define the total number of area elements:
c
     ntae = 8*nip + iN*js*2 + (je-js)*is + (je-js)*(iN-ie)
c
```

```
c  Define zone for TECPLOT :
c
c    Number of quadrilateral elements :
      numele = Ntae
c    Number of points :
      numpoi = 4 * numele
c    write the title :
      Write (nf,55) title, numpoi, numele
  55  Format ('Zone T ="',a8,'", i = ',i3,', j=',i3,', f = fepoint')
c
c  Define the coordinates of the corners forming the area elements
c  in the local coordinate system, and the surface area of these
c  elements :
c-------------------------------------------------------------------
c          Area elements around the circular hole
c-------------------------------------------------------------------
c
c  First "slice" :
c
      alpha = phi1 / Nip
      Do 1 i = 1, Nip
        cl(i,1,1) = Rint * dcos (alpha*(i-1))
        cl(i,1,2) = Rint * dsin (alpha*(i-1))
        cl(i,1,3) = 0.00d0
        cl(i,2,1) = dex
        cl(i,2,2) = dex * dtan (alpha*(i-1))
        cl(i,2,3) = 0.00d0
        cl(i,3,1) = dex
        cl(i,3,2) = dex * dtan (alpha*i)
        cl(i,3,3) = 0.00d0
        cl(i,4,1) = Rint * dcos (alpha*i)
        cl(i,4,2) = Rint * dsin (alpha*i)
        cl(i,4,3) = 0.00d0
c       surface area of elements in this slice :
        asec = 0.50d0 * Rint**2 * alpha
        If (i.eq.1) then
          atri(i) = 0.50d0 * dex**2 * dtan (alpha)
        Else
          atri(i) = 0.50d0 * dex**2 * ( dtan (alpha*i) -
     *                                  dtan (alpha*(i-1)) )
        Endif
        sae(i) = atri(i) - asec
  1   Continue
c
c  Second "slice" :
c
      alpha = phi2 / Nip
      Do 2 i = 1, Nip
        cl(nip+i,1,1) = Rint * dcos( alpha*(i-1) + phi1 )
        cl(nip+i,1,2) = Rint * dsin( alpha*(i-1) + phi1 )
        cl(nip+i,1,3) = 0.0d0
        cl(nip+i,2,1) = dey * dtan( phi2 - alpha*(i-1) )
        cl(nip+i,2,2) = dey
        cl(nip+i,2,3) = 0.00d0
        cl(nip+i,3,1) = dey * dtan( phi2 - alpha*i )
```

```
        cl(nip+i,3,2) = dey
        cl(nip+i,3,3) = 0.00d0
        cl(nip+i,4,1) = Rint * dcos( alpha*i + phi1 )
        cl(nip+i,4,2) = Rint * dsin( alpha*i + phi1 )
        cl(nip+i,4,3) = 0.00d0
c       surface area of elements in this slice :
        j = -i + nip + 1
        asec = 0.50d0 * Rint**2 * alpha
        If (j.eq.nip) then
          atri(nip+j) = 0.50d0 * dey**2 * dtan( alpha )
        Else
          atri(nip+j) = 0.50d0 * dey**2 * ( dtan( alpha*j ) -
     *                                dtan( alpha*(j-1) ) )
        Endif
        sae(nip+j) = atri(nip+j) - asec
  2   Continue
c
c Third "slice" :
c
      alpha = phi3 / nip
      Do 3 i = 1, nip
        cl(2*nip+i,1,1) = - Rint * dsin( alpha*(i-1) )
        cl(2*nip+i,1,2) =   Rint * dcos( alpha*(i-1) )
        cl(2*nip+i,1,3) =   0.00d0
        cl(2*nip+i,2,1) = - dey * dtan( alpha*(i-1) )
        cl(2*nip+i,2,2) =   dey
        cl(2*nip+i,2,3) =   0.00d0
        cl(2*nip+i,3,1) = - dey * dtan( alpha*i )
        cl(2*nip+i,3,2) =   dey
        cl(2*nip+i,3,3) =   0.00d0
        cl(2*nip+i,4,1) = - Rint * dsin( alpha*i )
        cl(2*nip+i,4,2) =   Rint * dcos( alpha*i )
        cl(2*nip+i,4,3) =   0.00d0
c
c       Surface area of the element :
c
        asec = 0.50d0 * Rint**2 * alpha
        If (i.eq.1) then
          atri(2*nip+i) = 0.50d0 * dey**2 * dtan( alpha )
        Else
          atri(2*nip+i) = 0.50d0 * dey**2 * (dtan( alpha*i ) -
     *                                dtan( alpha*(i-1) ) )
        Endif
        sae(2*nip+i) = atri(2*nip+i) - asec
  3   Continue
c
c Fourth "slice" :
c
      alpha = phi4 / nip
      Do 4 i = 1, nip
        cl(3*nip+i,1,1) = - Rint * dsin( phi3 + alpha*(i-1) )
        cl(3*nip+i,1,2) =   Rint * dcos( phi3 + alpha*(i-1) )
        cl(3*nip+i,1,3) =   0.00d0
```

```
        cl(3*nip+i,2,1) = - dexm
        cl(3*nip+i,2,2) =   dexm / dtan( phi3 + alpha*(i-1) )
        cl(3*nip+i,2,3) =   0.00d0
        cl(3*nip+i,3,1) = - dexm
        cl(3*nip+i,3,2) =   dexm / dtan( phi3 + alpha*i )
        cl(3*nip+i,3,3) =   0.00d0
        cl(3*nip+i,4,1) = - Rint * dsin( phi3 + alpha*i )
        cl(3*nip+i,4,2) =   Rint * dcos( phi3 + alpha*i )
        cl(3*nip+i,4,3) =   0.00d0
c
c       surface area of the element :
c
        asec = 0.50d0 * Rint**2 * alpha
        j = -i + nip + 1
        If (j.eq.nip) then
          atri(3*nip+i) = 0.50d0 * dexm**2 * dtan( alpha )
        Else
          atri(3*nip+i) = 0.50d0 * dexm**2 * (dtan( alpha*j ) -
     *                                dtan( alpha*(j-1)) )
        Endif
        sae(3*nip+i) = atri(3*nip+i) - asec
  4   Continue
c
c     Use the symmetry with respect to the x'-axis to define the coordinates
c     and the surface area of area elements on the lower side :
c
      Do 5 i = 1, 4*nip
c    (elements)
        inew = 4*nip + i
        iold = 4*nip - i + 1
        sae(inew) = sae(iold)
        Do 6 j = 1, 4
c           (4 corners)
c           (3 coordinates)
            cl(inew,j,1) =  cl(iold,j,1)
            cl(inew,j,2) = -cl(iold,j,2)
            cl(inew,j,3) =  cl(iold,j,3)
  6     Continue
  5   Continue
c-------------------------------------------------------------------
c          Area elements at the bottom
c-------------------------------------------------------------------
      Do 8 j = 1, js
        Do 9 i = 1, iN
c
c           number of the element :
c
            nae = (j-1)*iN +i + 8*nip
c
c           coordinates of the corners :
c
            cl(nae,1,1) = (i-1) * v + pap(1)
            cl(nae,1,2) = (j-1) * u + pap(2)
            cl(nae,1,3) = 0.00d0
            cl(nae,2,1) = i * v + pap(1)
```

```
            cl(nae,2,2) = cl(nae,1,2)
            cl(nae,2,3) = 0.00d0
            cl(nae,3,1) = cl(nae,2,1)
            cl(nae,3,2) = j * u + pap(2)
            cl(nae,3,3) = 0.00d0
            cl(nae,4,1) = cl(nae,1,1)
            cl(nae,4,2) = cl(nae,3,2)
            cl(nae,4,3) = 0.00d0
c
c           surface area :
c
            sae(nae) = u * v
 c
  9      Continue
  8    Continue
c-------------------------------------------------------------------
c           Area elements at the top
c-------------------------------------------------------------------
      Nctrj = 0
      Do 10 j = je + 1, jM
        Nctrj = Nctrj + 1
        Do 11 i = 1, iN
c
c           number of the element :
c
            nae = (Nctrj-1)*iN +i + 8*nip + iN*js
c
c           coordinates of the corners :
c
            cl(nae,1,1) = (i-1) * v + pap(1)
            cl(nae,1,2) = (j-1) * u + pap(2)
            cl(nae,1,3) = 0.00d0
            cl(nae,2,1) = i * v + pap(1)
            cl(nae,2,2) = cl(nae,1,2)
            cl(nae,2,3) = 0.00d0
            cl(nae,3,1) = cl(nae,2,1)
            cl(nae,3,2) = j * u + pap(2)
            cl(nae,3,3) = 0.00d0
            cl(nae,4,1) = cl(nae,1,1)
            cl(nae,4,2) = cl(nae,3,2)
            cl(nae,4,3) = 0.00d0
c
c           surface area :
c
            sae(nae) = u * v
 c
  11     Continue
  10   Continue
c-------------------------------------------------------------------
c           Area elements on the left
c-------------------------------------------------------------------
      Nctrj = 0
      Do 12 j = js + 1, je
        Nctrj = Nctrj + 1
```

```
      Do 13 i = 1, is
c
c          number of the element :
c
           nae = (Nctrj-1)*is + i + 8*nip + 2*(iN*js)
c
c          coordinates of the corners :
c
           cl(nae,1,1) = (i-1) * v + pap(1)
           cl(nae,1,2) = (j-1) * u + pap(2)
           cl(nae,1,3) = 0.00d0
           cl(nae,2,1) = i * v + pap(1)
           cl(nae,2,2) = cl(nae,1,2)
           cl(nae,2,3) = 0.00d0
           cl(nae,3,1) = cl(nae,2,1)
           cl(nae,3,2) = j * u + pap(2)
           cl(nae,3,3) = 0.00d0
           cl(nae,4,1) = cl(nae,1,1)
           cl(nae,4,2) = cl(nae,3,2)
           cl(nae,4,3) = 0.00d0
c
c        surface area :
c
           sae(nae) = u * v
c
  13     Continue
  12   Continue
c-------------------------------------------------------------------
c          Area elements on the right
c-------------------------------------------------------------------
      Nctrj = 0
      Do 14 j = js + 1, je
        Nctrj = Nctrj + 1
        Nctri = 0
        Do 15 i = ie + 1, iN
          Nctri = Nctri + 1
c
c          number of the element :
c
           nae = (Nctrj-1)*(iN-ie) + Nctri + 8*nip + 2*(iN*js) +
     *           (je-js)*is
c
c          coordinates of the corners :
c
           cl(nae,1,1) = (i-1) * v + pap(1)
           cl(nae,1,2) = (j-1) * u + pap(2)
           cl(nae,1,3) = 0.00d0
           cl(nae,2,1) = i * v + pap(1)
           cl(nae,2,2) = cl(nae,1,2)
           cl(nae,2,3) = 0.00d0
           cl(nae,3,1) = cl(nae,2,1)
           cl(nae,3,2) = j * u + pap(2)
           cl(nae,3,3) = 0.00d0
           cl(nae,4,1) = cl(nae,1,1)
           cl(nae,4,2) = cl(nae,3,2)
```

```
          cl(nae,4,3) = 0.00d0
c
c          surface area :
c
          sae(nae) = u * v
 c
  15     Continue
  14   Continue
c
c Define the coordinates of the points forming the area elements
c in the global coordinate system :
c
      Do 16 nae = 1, ntae
        Do 17 k = 1, 4
          cg(nae,k,1) = cl(nae,k,1)*dcos(a)-cl(nae,k,3)*dsin(a)+xo
          cg(nae,k,2) = cl(nae,k,2)
          cg(nae,k,3) = cl(nae,k,1)*dsin(a)+cl(nae,k,3)*dcos(a)+zo
  17     Continue
  16   Continue
c
c Normalize the distribution factor by the surface area of the element :
c
      Do 18 nae = 1, ntae
        dde = dble(ntre) * sae(nae)
        dnu = dble(nra(ns,nae))
        dfac(ns,nae) = dnu / dde
  18   Continue
c
c      Wirte the normalized distribution factor and the coordinates of the
c      element to unit nf :
c
      Do 19 j = 1, ntae
         Do 20 k = 1, 4
            Write (nf,56) cg(j,k,1),cg(j,k,2),cg(j,k,3),dfac(ns,j)
  20     Continue
  19   Continue
  56   Format (3(2x,f7.3),2x,e10.4)
c
c      Define the order of connection:
c
      Do 21 k = 1, ntae*4, 4
         Write (NF,57) k, k+1, k+2, k+3
  21   Continue
  57   Format (4(2x,i3))
 c
      Return
      End

c.....
c                  Subroutine SUBRIN
c.....
c This subroutine subdivides the rings on the baffle into area
c elements.
c    tau      : wall thickness
c    Ntype = 2 : top side
c    Ntype = 1 : bottom side
c.....
```

```
      Subroutine SUBRIN (p, dc, tau, Ntype, a1, a3,
     *            cXmax,cXmin,rXmax,rXmin,cYmin,cYmax,rYmin,rYmax,
     *            in, jm, title, nf, ns)
      Implicit double precision (a-h, o-z)
      Dimension p(3),dc(3),cl(800,4,3)
     *       ,sae(800),dfac(51,800),Nra(51,800)
c
c  Put frequently used variables in common blocks :
c
      Common / bl9 /  Nra, dfac, Ntre
c
c  Characteristics of the surface :
c
       tau = 0.20d0
       if (ntype.eq.1) then
c         (move this surface up by tau, i.e., redefine cXmin,cXmax,
c          rXmin,rXmax for the representation of the "back" of the
c          reflector)
          cXmin = tau * dsin(a1+a3) + cXmin
          rXmin = tau * dsin(a1+a3) + rXmin
          cXmax = tau * dsin(a1-a3) + cXmax
          rXmax = tau * dsin(a1-a3) + rXmax
       Endif
c
c  Define the total number of area elements:
c
      ntae = iN*jM*4
c
c  Define zone for TECPLOT :
c
c    Number of quadrilateral elements :
       numele = Ntae
c    Number of points :
       numpoi = 4 * numele
c    write the title :
      Write (nf,55) title, numpoi, numele
  55  format ('Zone T ="',a8,'", i = ',i3,', j=',i3,', f = fepoint')
c
c  Define the coordinates of the corners forming the area elements
c  in the global coordinate system, and the surface area of these
c  elements :
c-----------------------------------------------------------------
c           Area elements at the bottom
c-----------------------------------------------------------------
      v = (cXmax - cXmin) / iN
      u = (rYmin - cYmin) / jM

      Do 1 j = 1, jM
        Do 2 i = 1, iN
c
c           number of the element :
c
          nae = (j-1)*iN +i
c
c           coordinates of the corners :
c
          cl(nae,1,1) = (i-1) * v + cXmin
```

```
          cl(nae,1,2) = (j-1) * u + cYmin
          cl(nae,1,3) = -(cl(nae,1,1)-p(1))*dc(1)/dc(3) + p(3)
          cl(nae,2,1) = i * v + cXmin
          cl(nae,2,2) = cl(nae,1,2)
          cl(nae,2,3) = -(cl(nae,2,1)-p(1))*dc(1)/dc(3) + p(3)
          cl(nae,3,1) = cl(nae,2,1)
          cl(nae,3,2) = j * u + cYmin
          cl(nae,3,3) = -(cl(nae,3,1)-p(1))*dc(1)/dc(3) + p(3)
          cl(nae,4,1) = cl(nae,1,1)
          cl(nae,4,2) = cl(nae,3,2)
          cl(nae,4,3) = -(cl(nae,4,1)-p(1))*dc(1)/dc(3) + p(3)
c
c          surface area :
c
          sae(nae) = u * v
c
 2      Continue
 1    Continue
c-------------------------------------------------------------------
c           Area elements at the top
c-------------------------------------------------------------------
      v = (cXmax - cXmin) / iN
      u = (cYmax - rYmax) / jM

      Do 3 j = 1, jM
        Do 4 i = 1, iN
c
c          number of the element :
c
          nae = (j-1)*iN +i + iN*jM
c
c          coordinates of the corners :
c
          cl(nae,1,1) = (i-1) * v + cXmin
          cl(nae,1,2) = (j-1) * u + rYmax
          cl(nae,1,3) = -(cl(nae,1,1)-p(1))*dc(1)/dc(3) + p(3)
          cl(nae,2,1) = i * v + cXmin
          cl(nae,2,2) = cl(nae,1,2)
          cl(nae,2,3) = -(cl(nae,2,1)-p(1))*dc(1)/dc(3) + p(3)
          cl(nae,3,1) = cl(nae,2,1)
          cl(nae,3,2) = j * u + rYmax
          cl(nae,3,3) = -(cl(nae,3,1)-p(1))*dc(1)/dc(3) + p(3)
          cl(nae,4,1) = cl(nae,1,1)
          cl(nae,4,2) = cl(nae,3,2)
          cl(nae,4,3) = -(cl(nae,4,1)-p(1))*dc(1)/dc(3) + p(3)
c
c          surface area :
c
          sae(nae) = u * v
c
 4      Continue
 3    Continue
c-------------------------------------------------------------------
c           Area elements on the left
c-------------------------------------------------------------------
```

```
      v = (rXmin - cXmin) / jM
      u = (rYmax - rYmin) / iN

      Do 5 j = 1, iN
        Do 6 i = 1, jM
c
c           number of the element :
c
            nae = (j-1)*jM +i + 2*iN*jM
c
c           coordinates of the corners :
c
            cl(nae,1,1) = (i-1) * v + cXmin
            cl(nae,1,2) = (j-1) * u + rYmin
            cl(nae,1,3) = -(cl(nae,1,1)-p(1))*dc(1)/dc(3) + p(3)
            cl(nae,2,1) = i * v + cXmin
            cl(nae,2,2) = cl(nae,1,2)
            cl(nae,2,3) = -(cl(nae,2,1)-p(1))*dc(1)/dc(3) + p(3)
            cl(nae,3,1) = cl(nae,2,1)
            cl(nae,3,2) = j * u + rYmin
            cl(nae,3,3) = -(cl(nae,3,1)-p(1))*dc(1)/dc(3) + p(3)
            cl(nae,4,1) = cl(nae,1,1)
            cl(nae,4,2) = cl(nae,3,2)
            cl(nae,4,3) = -(cl(nae,4,1)-p(1))*dc(1)/dc(3) + p(3)
c
c           surface area :
c
           sae(nae) = u * v
c
 6      Continue
 5    Continue

c-------------------------------------------------------------------
c           Area elements on the right
c-------------------------------------------------------------------
      v = (cXmax - rXmax) / jM
      u = (rYmax - rYmin) / iN

      Do 7 j = 1, iN
        Do 8 i = 1, jM
c
c           number of the element :
c
            nae = (j-1)*jM +i + 3*iN*jM
c
c           coordinates of the corners :
c
            cl(nae,1,1) = (i-1) * v + rXmax
            cl(nae,1,2) = (j-1) * u + rYmin
            cl(nae,1,3) = -(cl(nae,1,1)-p(1))*dc(1)/dc(3) + p(3)
            cl(nae,2,1) = i * v + rXmax
            cl(nae,2,2) = cl(nae,1,2)
            cl(nae,2,3) = -(cl(nae,2,1)-p(1))*dc(1)/dc(3) + p(3)
            cl(nae,3,1) = cl(nae,2,1)
            cl(nae,3,2) = j * u + rYmin
```

```
          cl(nae,3,3) = -(cl(nae,3,1)-p(1))*dc(1)/dc(3) + p(3)
          cl(nae,4,1) = cl(nae,1,1)
          cl(nae,4,2) = cl(nae,3,2)
          cl(nae,4,3) = -(cl(nae,4,1)-p(1))*dc(1)/dc(3) + p(3)
c
c           surface area :
c
          sae(nae) = u * v
 c
  8      Continue
  7    Continue
c
c  Normalize the distribution factor by the surface area of the element :
c
       Do 9 nae = 1, ntae
         dde = dble(ntre) * sae(nae)
         dnu = dble(nra(ns,nae))
         dfac(ns,nae) = dnu / dde
  9    Continue
c
c  Wirte the normalized distribution factor and the coordinates of the
c  element to unit nf :
c
       Do 10 j = 1, ntae
         Do 11 k = 1, 4
           Write (nf,56) cl(j,k,1),cl(j,k,2),cl(j,k,3),dfac(ns,j)
  11     Continue
  10   Continue
  56   Format (3(2x,f7.3),2x,e10.4)
c
c  Define the order of connection:
c
       Do 12 k = 1, ntae*4, 4
           Write (NF,57) k, k+1, k+2, k+3
  12   Continue
  57   Format (4(2x,i3))
c
      Return
      End

c.....
c                Suborutine SUBSHI
c.....
c  This subroutine subdivides a hemispherical surface into equal-area
c  elements and defines the corners forming the area element to a file
c.....
      Subroutine SUBSHI (rsh, ns, Nz, Ncir, title, nf)
      Implicit double precision (a-h, o-z)
      Character*8 title
      Dimension r(11), theta(10),cl(800,4,3),cg(800,4,3),Nra(51,800),
     *          dfac(51,800)
c
c  Put frequently used variables in common blocks :
c
      Common / bl9 / Nra, dfac, Ntre
```

```
      pi = dacos(-1.00d0)



c
c  Surface area of each element:
c
        dntre = dble (ntre)
        dde = dntre*2.00d0*pi*rsh**2/(Nz*Ncir)
c
c  Define zone for TECPLOT :
c
c    Number of quadrilateral elements :
         numele =  Nz*Ncir/2
c    Number of points :
         numpoi = 4 * numele
c    write the title :
         Write (nf,30) title, numpoi, numele
  30      format ('Zone T ="',a8,'", i = ',i3,', j=',i3,', f = fepoint')
c
c  Define the coordinates of the corners in the LCS:
c
       phi = 2.00d0*pi/Ncir
       r(1) = rsh
       Do 1 j = 1, Nz
         Theta(j) = dasin(dble(j)/dble(Nz))
          r(j+1) = rsh * dcos (theta(j))
          Do 2 i = 1, Ncir/2
c              nae: number of the area element.
              nae = (j-1)*Ncir + i
c           Corner 1:
              cl(nae,1,1) = r(j) * dcos ( (i-1)*phi )
              cl(nae,1,2) = r(j) * dsin ( (i-1)*phi )
              cl(nae,1,3) = dble(j-1) *rsh/dble(Nz)
c           Corner 2:
              cl(nae,2,1) = r(j) * dcos ( i*phi )
              cl(nae,2,2) = r(j) * dsin ( i*phi )
              cl(nae,2,3) = cl(nae,1,3)
c           corner 3:
             cl(nae,3,1) = r(j+1) * dcos ( i*phi )
             cl(nae,3,2) = r(j+1) * dsin ( i*phi )
             cl(nae,3,3) =dble(j)* rsh/dble(Nz)
c           Corner 4:
             cl(nae,4,1) = r(j+1) * dcos ( (i-1)*phi )
             cl(nae,4,2) = r(j+1) * dsin ( (i-1)*phi )
             cl(nae,4,3) = cl(nae,3,3)
c
c             Coordinates of the area element in the global coordinate system:
c
             Do 3 L = 1, 4
                cg(nae,L,1) = - cl(nae,L,1)
                cg(nae,L,2) =   cl(nae,L,2)
                cg(nae,L,3) = - cl(nae,L,3) + rmh
   3         Continue
c
c             Define the normalized distribution factor:
```

```
c
          dnu = dble( Nra(ns,nae) )
          dfac(ns,nae) = dnu/dde
c
c         Write the coordinates of the area element and its normalized
c         distribution factor to unit nf:
c
          Do 4 L = 1, 4
            Write (nf,10) cg(nae,L,1),cg(nae,L,2),cg(nae,L,3),
     *                   dfac(ns,nae)
 4        Continue
 2      Continue
 1    Continue
c
c Define how nodes in unit nf are connected:
c
      Do 5 i = 1, 4 *Nz*Ncir /2, 4
       Write (nf,20) i, i+1, i+2, i+3
 5    Continue
 10   format (3(2x,f7.3),2x,e10.4)
 20   format (4(2x,i4))

      Return
      End
```

```
c.....
c                Subroutine DETQR
c.....
c    Determination of the coordinates q,r in a 2-dimensional local
c    coordinate system defined by 3 points A,B,C of a point whose
c    coordinates in the global coordinate system are x2,y2,z2.
c.....
      Subroutine DETQR (x2, z2, pa, pb, pc)
      Implicit double precision (a-h, o-z)
      Dimension pa(3), pb(3), pc(3)
c
c Put frequently used variables in common blocks :
c
      Common / bl21 / q, r
c
c Define q and r such that AP = q*AB + r*AC :
c
      det = (pb(1)-pa(1))*(pc(3)-pa(3))
     *     - (pb(3)-pa(3))*(pc(1)-pa(1))
      q = ( (x2-pa(1))*(pc(3)-pa(3))
     *    - (z2-pa(3))*(pc(1)-pa(1)) ) /det
      r = ( (pb(1)-pa(1))*(z2-pa(3))
     *    - (pb(3)-pa(3))*(x2-pa(1)) ) /det

      Return
      End
```

```
c.....
c                Subroutine NURAPE
c.....
c    Numbering of a rectangular-area element perpendicular
c    to the x,z-plane, using the x- and y-coordinates.
c.....
      Subroutine NURAPE (x2, y2, cXmin, cXmax, cYmin, cYmax,
     *                   in, jm, Nsa)
      Implicit double precision (a-h, o-z)

c  Determine the number of the area element :
      Nsa = int ( (x2-cXmin)/(cXmax-cXmin)*iN + 1 )
     *      + iN*int ( (y2-cYmin)/(cYmax-cYmin)*jM )

      Return
      End

c.....
c                Subroutine NURAP2
c.....
c    Numbering of a rectangular-area element perpendicular
c    to the x,z-plane, using the z- and y-coordinates.
c.....
      Subroutine NURAP2 (z2, y2, cZmin, cZmax, cYmin, cYmax,
     *                   in, jm, Nsa)
      Implicit double precision (a-h, o-z)

c  Determine the number of the area element :
      Nsa = int ( (z2-cZmin)/(cZmax-cZmin)*iN + 1 )
     *      + iN*int ( (y2-cYmin)/(cYmax-cYmin)*jM )

      Return
      End

c.....
c                Subroutine NURAPA
c.....
c    Numbering of a rectangular-area element parallel to
c    the x,z-plane.
c.....
      Subroutine NURAPA (x2, z2, pa, pb, pc, in, jm, Nsa)
      Implicit double precision (a-h, o-z)
      Dimension pa(3), pb(3), pc(3)

c  Put frequently used variables in common block :
      Common / bl21 / q, r

c  Define the coordinates of the point P(x2,y2,z2) in the 2-dimensional
c  coordinate system defined by {a,AB,AC}, such that AP = q*AB + r*AC :
      Call DETQR (x2, z2, pa, pb, pc)

c  Number of the area element:
      Nsa = int ( r*in +1 ) + in*int ( q*jm )

      Return
      End
```

```
c.....
c                Subroutine NUMCIR
c.....
c  Numbering of an area element on the sensor-aperture.
c  pi.....: 3.1416...
c  x2, y2, z2 : Coordinates of the point of incidence in the global
c            coordinate system.
c  a2.....: Direction of the sensor aperture.
c  Nphi.....: Number of circumferential subdivision.
c  Nr.....: Number of radial subdivision.
c  Nsa.....: Number of the area element.
c.....
      Subroutine NUMCIR (pi, x2, y2, z2, a2, Nr, Nphi, Nsa)
      Implicit double precision (a-h, o-z)
      Dimension Rsap(6)

c  Put frequently used variables in common block :
      Common / bl22 / rsap

c  Coordinates in the local coordinate system:
      x2p = x2*dcos(a2) + z2*dsin(a2)
*     y2p = y2

c  Define the radius and angle in the local coordinate sysytem:
      r = dsqrt(x2p**2 + y2**2)
      gamma = datan2( y2, x2p)
      if (gamma.lt.0.00d0) gamma = gamma + 2.00d0*pi

c  Number of the area element :
      Do 2 i = 1, Nr
         if (r.GE.rsap(i).AND.r.LE.rsap(i+1)) goto 3
 2    Continue
 3    Continue

      Nsa = (i-1)*Nphi + int( gamma/(2.00d0*pi)*Nphi + 1 )

      Return
      End


c.....
c                Subroutine NUMQUA
c.....
c     Numbering of area elements defined by SUBQUA.
c     x2, y2, z2 : Coordinates of the point of incidence in the
c                  global coordinate system.
c     A, B, C    : The 3 corners of the triangular surface.
c     N          : The nummber of equal intervals along any side of
c                  the surface.
c     Nsa        : The number of the area element on which is the
c                  point of incidence.
c.....
      Subroutine NUMQUA (x2,z2,A,B,C,N,NSA)
      Implicit double precision (A-H,O-Z)
      Dimension A(3),B(3),C(3)
```

```
c Put frequently used variables in common block :
      Common / bl21 / q, r

c Define the coordinates of the point of incidence  in the
c local coordinate system, such that AP = q*AB + r*AC :
     Call DETQR (x2,z2,A,B,C)

c Number of the area element :
      m2 = int(q*N)
      Nsa = m2 * ( 2*N - (m2 - 1) ) / 2 + int(r*N + 1)
      Return
      End

c.....
c                Subroutine NUMTRA
c.....
c     Numbering of area elements defined by SUBTRA.
c     x2, y2, z2 : Coordinates of the point of incidence in the
c                  global coordinate system.
c     A, B, C    : The 3 corners of the triangular surface.
c     iN, jM     : The nummbers of equal intervals along two sides of
c                  the surface.
c     Nsa        : The number of the area element on which is the
c                  point of incidence.
c.....
      Subroutine NUMTRA (x2,z2,A,B,C,iN,jM,NSA)
      Implicit double precision (A-H,O-Z)
      Dimension A(3),B(3),C(3)
c
c Put frequently used variables in common block:
c
      Common / bl21 / q, r
c
c Define the coordinates of the point of incidence  in the
c local coordinate system, such that AP = q*AB + r*AC :
     Call DETQR (x2,z2,A,B,C)
c
c Number of the area element:
c
     m2 = int(q*jM)
     Nsa = m2 * (2*iN -(m2 -1)*iN/jM) / 2 + int(r*iN + 1)

      Return
      End

c.....
c                Subroutine NUMINT
c.....
c Numbering of an area element of a rectangular surface with a
c circular hole on one of median lines (e.g., internal wall s34).
c.....
      Subroutine NUMINT (x2,y2,z2,nsa)
      implicit double precision (a-h, o-z)
      dimension p42(3), p(3), pp(3), pap(3), pbp(3), pcp(3)
c
c Put frequently used variables in common blocks:
c
      Common / bl20 / a6, p42, rint
```

```
      Common / bl23 / phi1,phi2,phi3,phi4,disab,disac,
     *                is,ie,js,je,in,jm,Nip
      Common / bl25 / pap, pbp, pcp
      pi = dacos(-1.00d0)
c
c  Define the coordinates of the point of incidence in the local coordi-
c  nate system:
c
      p(1) = x2
      p(2) = y2
      p(3) = z2
      Call GLOLOC (p, pp, p42(1), p42(3), a6)
c
c  Define the indices of the area element:
c
      ig = int( (pp(1)-pap(1))/disac*in + 1)
      jg = int( (pp(2)-pap(2))/disab*jm + 1)
c
c  Number of the area element :
c
      If (ig.GT.is.AND.ig.LE.ie.AND.jg.GT.js.AND.jg.LE.je) then
c     (1)
c        the point of incidence is on an area element around the circle,
c        define the circumferential angle:
         phi = datan2 (pp(2), pp(1))
         If (phi.LT.0.00d0) phi = 2.00d0*pi + phi
c        (2)
           If (phi.LE.phi1) then
             nsa = int( phi/phi1*NiP + 1)
           Else if (phi.le.pi/2.00d0) then
             nsa = NiP + int( (phi-phi1)/phi2*Nip +1 )
           Else if (phi.LE.(pi/2.00d0+phi3)) then
            nsa = 2*Nip + int( (phi-pi/2.00d0)/phi3*NiP + 1)
           Else if (phi.LE.pi) then
            nsa = 3*Nip+int( (phi-pi/2.00d0-phi3)/phi4*NiP+1)
           Else if (phi.LE.(pi+phi4)) then
            nsa = 4*Nip+int((phi-pi)/phi4*NiP+1)
           Else if (phi.LE.1.50d0*pi) then
            nsa = 5*NiP+int((phi-pi-phi4)/phi3*NiP+1)
           Else if (phi.LE.(1.50d0*pi+phi2)) then
            nsa = 6*NiP+int((phi-1.50d0*pi)/phi2*NiP+1)
           Else
            nsa = 7*NiP+int((phi-1.50d0*pi-phi2)/phi1*NiP+1)
         Endif
c        (2)
      Else if (jg.LE.js) then
c        the point is on an area element along the bottom:
         nsa = (8*NiP) + ig +iN*(jg-1)
      Else if (jg.GT.je) then
c        the point is on an area element along the top
         nsa = (8*NiP + iN*js) + ig + iN*(jg-je-1)
      Else if (ig.LE.is) then
c        the point is on an area element on the left:
         nsa = (8*NiP + iN*js*2) + ig + is*(jg-js-1)
      Else if (ig.GT.ie) then
```

```
c          the point is on an area element on the right:
           nsa = (8*NiP + iN*js*2 + (jM-2*js)*is)
     *           + (ig-ie) + (iN-ie)*(jg-js-1)
           Else
             Write (*,*) 'the numbering of area elements on the'
             Write (*,*) 'internal wall is wrong!'
           Endif
c     (1)
           Return
           End

c.....
c                  Subroutine NURING
c.....
c  This subroutine defines the numbering of a rectangular-area element
c  on a ring. The ring is divided into 4 rectangular zones : bottom,
c  top, left, right. Each zone is then sudivided into iN*jM equal-area
c  elements, where iN is the number of intervals on the longest side
c  and jM is the number of intervals on the remaining side.
c   The zoning is in the order : Bottom, top, left, right. So, the
c  number in each zone is :
c     Bottom : from          1  to  iN*jM
c     Top    : from   iN*jM + 1  to  2*iN*jM
c     Left   : from 2*iN*jM + 1  to  3*iN*jM
c     Right  : from 3*iN*jM + 1  to  4*iN*jM
c.....
           Subroutine NURING (x2, y2, ns, in, jm, nsa)
           Implicit double precision (a-h, o-z)
           Dimension cXmin(51),cXmax(51),cYmin(51),cYmax(51)
     *           ,rXmin(38:51),rXmax(38:51),rYmin(38:51),rYmax(38:51)
c
c  Put frequently used variables in common blocks:
c
           Common/ bl11 /cXmin,cXmax,cYmin,cYmax
           Common/ bl12 /rXmin,rXmax,rYmin,rYmax

           If (y2.LE.rYmin(ns)) then
c
c            The area element is along the bottom:
             nsa = int( (x2-cXmin(ns))/(cXmax(ns)-cXmin(ns))*iN + 1 )
     *           + iN*int( (y2-cYmin(ns))/(rYmin(ns)-cYmin(ns))*jM )

           Else if (y2.GE.rYmax(ns)) then
c
c            the area element is along the top:
             nsa = jM*iN + int( (x2-cXmin(ns)) /
     *                        (cXmax(ns)-cXmin(ns))*iN+1)
     *           + iN*int( (y2-rYmax(ns))/(cYmax(ns)-rYmax(ns))*jM )

           Else if (x2.LE.rXmin(ns)) then
c
c            the area element is on the left:
             nsa = 2*jM*iN + int( (x2-cXmin(ns)) /
     *                          (rXmin(ns)-cXmin(ns))*jM + 1 )
     *           + jM*int( (y2-rYmin(ns))/(rYmax(ns)-rYmin(ns))*iN )
```

```
      Else
c
c       the area element is on the right:
        nsa = 3*jM*iN + int( (x2-rXmax(ns)) /
     *                    (cXmax(ns)-rXmax(ns))*jM + 1 )
     *          + jM*int( (y2-rYmin(ns))/(rYmax(ns)-rYmin(ns))*iN )
      Endif

      Return
      End
c.....
c               Subroutine MOVET
c.....
c  Move a point by Tx and Tz, for visualization by TECPLOT.
c.....
      Subroutine MOVET (p, pp, Tx, Tz)
      Implicit double precision (a-h, o-z)
      Dimension p(3), pp(3)
        pp(1) = p(1) + Tx
        pp(2) = p(2)
        pp(3) = p(3) + Tz
      Return
      End



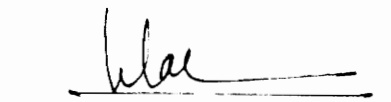
c.....
c               Subroutine SURTRI
c.....
c  This subroutine defines the surface area of a triangular surface
c  parallel to the x,z-plane.
c.....
      Subroutine SURTRI (a, b, c, sa)
      Implicit double precision (a-h, o-z)
      Dimension a(3), b(3), c(3), h(3)
        cm1 = (c(3)-a(3))/(c(1)-a(1))
        cb1 = a(3) - cm1*a(1)
        cm2 = -1.00d0 / cm1
        cb2 = b(3) - cm2*b(1)
        h(1) = (cb2-cb1) / (cm1-cm2)
        h(2) = a(2)
        h(3) = cm1*h(1) + cb1
        sa = dist(a, c) * dist(b, h) / 2.00d0
      Return
      End
c.....
c               Subroutine SURREF
c.....
c  This subroutine defines the surface area of a reflector and its
c  sub-surfaces.
c.....
      Subroutine SURREF (p1,p2,p3,p4,p5,p6,
     *                   ssat,ssab,ssal,ssar,sa)
      Implicit double precision (a-h, o-z)
      Dimension p1(3),p2(3),p3(3),p4(3),p5(3),p6(3)
```

```
    sa = dist(p1, p2) * dist(p2, p3) - dist(p4, p5) * dist(p5, p6)
    sl = dist(p5, p6)
    gl = dist(p2, p3)
    width = (gl - sl) / 2.00d0
    ssab = dist(p1, p2) * width
    ssat = ssab
    ssal = dist(p5, p6) * width
    ssar = ssal
  Return
  End
```

Vita

Tai K. Nguyen is a native of Saigon, Vietnam. He left his country of origin on April 30, 1979, exactly four years after the end of the Vietnam War to go to France. In June 1983, he received the Baccalaureat Diploma with honor from the Lyceé Technique Nationalisé Mixte de Douai, France. In September 1983, he started a five-year program in Mechanical Engineering at the Institut National des Sciences Appliqueés (INSA) de Lyon, France. During this program, Tai participated in the Cooperative Education Program as an engineer-trainee at the Renault Automobiles Research Center in Paris, France, from January to June, 1987. In June 1988, Tai received his Diplôme d'Ingénieur in Mechanical Engineering from INSA. One week after graduation, Tai represented INSA at a national meeting to compete for the best engineering project of the year from twelve French engineering schools in Paris, France.

In January 1989, Tai moved to America to live with his family in Roanoke, Virginia. In June 1992, he started his graduate study at Virginia Polytechnic Institute and State University (VPI&SU) where he became a member of the Thermal Radiation Group. In December 1994, Tai received his Master of Science Degree in Mechanical Engineering. Upon completion of his master's degree, Tai continues to work toward a doctor of philosophy degree also at VPI&SU.



Tai K. Nguyen