An End-to-End Model of the Earth Radiation Budget Experiment (ERBE) Earth-Viewing Nonscanning Radiometric Channels

by

Kory J. Priestley

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:

J.R. Mahan
Dr. J. R. Mahan, Chairman

E.P. Scott
Dr. E. P. Scott

C.H. Stern
Dr. C. H. Stern

October, 1993
Blacksburg, Virginia

An End-to-End Model of the Earth Radiation Budget Experiment (ERBE) Earth-Viewing Nonscanning Radiometric Channels

by

Kory J. Priestley

Dr. J. R. Mahan, Chairman

Mechanical Engineering

(ABSTRACT)

The Earth Radiation Budget Experiment (ERBE) active-cavity radiometers are used to measure the incoming solar, reflected solar, and emitted longwave radiation from the Earth and its atmosphere. The radiometers are carried by the National Aeronautics and Space Administration's Earth Radiation Budget Satellite (ERBS) and the National Oceanic and Atmospheric Administration's NOAA-9 and NOAA-10 spacecraft. Four Earth-viewing nonscanning active-cavity radiometers are carried by each platform. Two of the radiometers are sensitive to radiation in the spectral range from 0.2 to 50 μm, while the other two radiometers are sensitive to radiation in the spectral range from 0.2 to 5.0 μm. Each set of radiometers comes in a wide-field-of-view (WFOV) and a medium-field-of-view (MFOV) configuration. The cavities of the shortwave (visible) radiometers are covered with a Suprasil® hemispherical dome to filter out the incoming longwave radiation.

Knowledge of the optical and physical properties of the radiometers allows

their responses to be predicted using a low-order physical model. A high-level, dynamic electrothermal end-to-end model which accurately predicts the radiometers dynamic output has also been completed. This latter model is used to numerically simulate the calibration procedures of the actual instruments. With calibration of the end-to-end model complete, a simulation of a phenomena referred to as the "solar blip" is conducted to investigate the instruments' responses to steep transient events. The solar blip event occurs when direct solar radiation is briefly incident to the active-cavity radiometric channels as the spacecraft passes into and out of the Earth's shadow.

Acknowledgments

I would like to thank my advisor, Professor J. R. Mahan, for his continuing guidance and support of this research. I would also like to thank Professors E. P. Scott and C.H. Stern for serving on my committee.

A special debt of gratitude is owed to the Radiation Sciences Branch of the NASA Langley Research Center which has made this research possible under NASA Grants NAG-1-1325 and NGT-51108. In Particular, I would like to thank Mr. Robert B. Lee III and Mr. Robert J. Keynton for their guidance and support over the past year and a half.

To my wife Anne I want to say thank you for your unwavering support, patience and understanding over the course of this research. Finally, to my family, thank you for the values and ideals which you have presented me, allowing me to go as far as I have.

Table of Contents

List of Tables

List of Figures

Nomenclature

A	Surface area (m^2)
A_F	Coefficient associated with the field-of-view limiter temperature (W/m^2K)
A_R	Coefficient associated with the reference cavity heater voltage (W/m^2V^2)
A_V	Coefficient associated with the active cavity heater voltage (W/m^2V^2)
B	Offset term which calibrates the instruments by means of internal calibration sources (W/m^2)
C_1	Constant used in Planck's radiation distribution function (5.9544 x 108 $W\text{-}\mu m^4/m^2$)
C_2	Constant used in Planck's radiation distribution function (1.4338×10^4 $\mu m\text{-}K$)
D_{ijk}	Monochromatic radiation distribution factor (-)
E	Irradiance at satellite altitude (W/m^2)
e	Emissive power (W)
F	Radiation view factor (-)
G	Volumetric heat generation (W/m^3)

N_{ik}	Number of energy bundles emitted by element i, in wavelength interval k (-)
N_{ijk}	Number of energy bundles emitted by element i which are absorbed by element j, in wavelength interval k (-)
n	Index of refraction for a participating medium (-); time step (-)
Q	Heat input (W)
R	Electrical resistance (Ω); Radius (m); Random number (-); Reflectivity ratio (-)
$T, \Delta T$	Temperature, temperature drop between the active and reference resistance temperature detectors (K)
$t, \Delta t$	Time, time increment (s)
$\mathbf{U}$	Unit vector (-)
V	Electrical potential (V)
V_b	Bias voltage (V)
V_0	Bridge supply voltage (V)
V_1	Bridge output voltage (V)
V_2	Voltage across the heater wire (V)
x,y,z	Cartesian coordinates (m)

Greek

α	Absorptivity (-); linear temperature coefficient of resistance (1/K)
δ	Material thickness (m)
ϵ	Emissivity (-)

θ	Zenith angle (rad); Cone angle (rad); Brewster angle (rad)
κ	Monochromatic absorption coefficient (1/m)
λ	Wavelength (μm)
ρ^d	Diffuse component of reflectivity (-)
ρ^s	Specular component of reflectivity (-)
σ	Stefan-Boltzmann constant (5.6696×10^{-8} W/m^{2}K)
τ	Transmissivity (-)
ϕ	Azimuthal angle (rad)

1.0 Introduction

1.1 The Earth Radiation Budget

The ability of man to influence his surroundings on a global scale is a phenomenon unique to the twentieth century. With the advent of industrialization has come the consequence of releasing vast quantities of pollutants into the atmosphere. As a result, atmospheric composition has become dynamic, with climatic consequences that are not yet well understood. Measurement of the radiative exchange between the planet Earth and its space environment represents one of the fundamental activities for understanding the driving mechanisms of our planet's weather and climate.

The Earth radiation budget, or the energy balance of the Earth-atmosphere-ocean system, is defined as the difference between the absorbed solar radiation and the radiation emitted from the Earth and its atmosphere. Determination of the Earth's radiation budget allows scientists to better understand the dynamic behavior of our planet's climatic system [1,2]. Contributions to the Earth's radiation budget are shown in Figure 1.

1.2 Earth Radiation Budget Measurements

The following historical overview has been obtained from Hunt et al. [3,4].

The first serious attempts at estimating the Earth radiation budget components on the basis of measurements were not made until well into the nineteenth century. Pouillet obtained an estimate of the solar constant of 1211 W/m^2 in 1837. However, it was not until the first decade of the twentieth century that the initial attempt to determine the components of the Earth radiation budget, on the basis of observational data, were made. These early efforts suffered from severe limitations in geographical sampling of climatic observations.

Simpson was the first to recognize, during the 1920's, the importance of the spectral distribution of water vapor absorption and to calculate the distribution of incoming and outgoing radiation as a function of latitude and season, making use of observed cloud distributions.

The 1940's and 50's marked a time of significant developments in atmospheric physics that led to increased knowledge of the Earth radiation budget.

Aircraft studies during this period provided much needed data on cloud physics.

With the onset of the 60's, man was about to enter the space-age, and with it the era of satellite meteorology. The first satellite dedicated to the determination of the Earth radiation budget was Explorer 7, launched on October 13, 1959. The radiometers on Explorer 7 consisted of black and white hemispheres containing thermistor bolometers attached to rectangular mirrors on the equator of the spacecraft. The reflected image of each hemisphere in the mirrors made the radiometers appear as "spheres" in space.

The first imaging system to be carried into space was launched on April 1, 1960, on board the TIROS-1 satellite, the first in a series of seven satellites. The Television Infrared Observation Satellites (TIROS) missions provided four years of Earth radiation budget data and included the first such satellite to provide a continuous record of data for one year, TIROS 7.

The most successful of the early observational systems were the Nimbus-6 and Nimbus-7 satellites. These satellites were launched in 1975 and 1978, respectively, and together provided more than a decade of continuous Earth radiation budget data [4].

Advancements which have influenced the evolution of Earth radiation budget instrumentation may be broken down into four broad categories: (1) spacecraft power budgets, on-board data storage, satellite stabilization, and attitude control; (2) viewing angles of nonscanning medium field-of-view and wide field-of-view radiometers and scanning medium and high-resolution radiometers; (3) spectral isolation of the shortwave and longwave components; and (4) on-board calibration standards for both longwave and shortwave detectors.

The current Earth Radiation Budget Experiment (ERBE) is the first program to incorporate all of these advancements.

1.3 The Earth Radiation Budget Experiment

The Earth Radiation Budget Experiment (ERBE) was begun by the National Aeronautics and Space Administration (NASA) in 1979 in order to quantify the radiative interaction between the Earth and its space environment. The radiation from the sun which is absorbed by the Earth and eventually re-emitted is the energy source which drives the motions of the Earth's atmosphere and oceans and which determines our weather and climate. The goal of ERBE is to monitor this interaction over a long time scale and over specific geographical regions [5].

The ERBE mission consists of a three-satellite platform. The first ERBE instruments were launched on October 5, 1984, aboard the National Aeronautics and Space Administration's Earth Radiation Budget Satellite (ERBS). The remaining radiometric instruments were launched on December 5, 1984, and September 17, 1986, aboard the National Oceanic and Atmospheric Administration's NOAA-9 and NOAA-10 satellites. Each radiometric package contains four Earth-viewing nonscanning active- cavity radiometers, three scanning thermistor bolometer radiometers, and a solar monitor. Inflight stability of all radiometric channels is monitored by internal calibration sources. Combined, these radiometers can quantitatively describe the spectral and spatial distribution of the Earth's radiative field.

The Earth-viewing nonscanning active cavity radiometers are described in detail in Chapter 2. The scanning thermistor bolometer radiometers and solar monitors are described elsewhere [6,7].

ERBE has established the benchmark by which the success of future Earth radiation budget measurement missions will be judged. Definitive answers to several important scientific questions have been obtained throughout the mission. Two such questions answered by ERBE have been the net cooling effect which clouds and volcanic activity produce on our climate system [8-11] and the correlation of solar activity to global-average atmospheric temperature change [12,13].

1.4 The Thermal Radiation Group

Under the direction of Professor J. R. Mahan, graduate students have been studying the dynamic electrothermal properties of instruments designed to measure the Earth radiation budget since the early 1970's. The Thermal Radiation Group originally studied the theoretical radiative characteristics of spherical balloon-type detectors [14-16]. However, with the development of more powerful computers in the early eighties the group turned their focus towards applied numerical modelling of more complex radiative enclosures. Eskin [17] was the first student in the group to utilize a Monte-Carlo-based ray-trace technique. Eskin's work modeled the radiative exchange inside a conical cavity such as those on the ERBE mission. Gardiner [18] studied the operation of ERBE-type active cavity radiometers at cryogenic temperatures. Tira [19] was the first to utilize the Monte-Carlo ray-trace technique in conjunction with a finite-element model of an active-cavity radiometer to determine the electrical response of the detector to varying radiative inputs.

Thus began the current focus of the Thermal Radiation Group: completion of end-to-end models for space-based radiometric channels. These models usually begin with a radiative and optical analysis of the instrument in question using a Monte-Carlo ray-trace technique. The results from the ray trace are then implemented into either a finite-element or finite-difference heat conduction code which characterizes the electrothermal behavior of the instrument to various radiative inputs. Once validated by successfully processing actual data product, these models may be used to investigate the sources of any anomalous behavior of the actual instruments.

Research concluded so far includes the work of Fanney [14], Rasnic [15], Passwaters [16], Eskin [17], Gardiner [18], Tira [19-22], Kowsary [23], Meekins

[24], Haeffelin [25-28], Bongiovi [29], Chapman [30], Villeneuve [31], and the current author.

1.5 Motivation and Goals

This thesis presents the effort involved in the development of a complete end-to-end model of the Earth-viewing nonscanning active-cavity radiometric channels developed for the Earth Radiation Budget Experiment. The first step toward completion of this goal was the work completed by Eskin [17] in which he characterized the transient thermal behavior of the ERBE active cavity. Tira [19,22] advanced Eskins work by incorporating a two-dimensional finite-element description of the electrothermal response of the active cavity. Haeffelin [26,28] characterized the optical front end of these channels, completing a transient model of heat conduction in the hemispherical filter dome.

The goal of the current effort is to integrate the optical and radiative front end model created by Haeffelin with the electrothermal model of the active cavity created by Tira. This goal is met by adding the active cavity geometry to the radiative model of the front end completed by Haeffelin. This results in a complete description of the radiative exchange within the instrument. This description can be used as input for radiative boundary conditions in both the finite-difference heat conduction code of the hemispherical filter dome completed by Haeffelin and the finite-element electrothermal characterization of the active cavity completed by Tira. The resulting high-level dynamic electrothermal model would then be formulated at a level which would permit the assessment of changes in the geometry, surface temperature distributions, or other thermophysical properties on the response of the instrument.

In order to demonstrate the accuracy of the resulting high-level end-to-end model it is calibrated numerically with procedures identical to those for the actual instrument. This simulated calibration will allow the model's response to a given radiative scene to be characterized in such a way that the performance of the actual radiometers may be assessed.

In addition, a low-order physical model of the radiometers is used to estimate the uncertainty in the instrument data product. This low-order physical model is also used to characterize the instruments through the use of first-principle physics. Thus three types of instrument calibration are completed: ground calibration of the actual instruments, a first-principle approach using a low-order physical model, and a numerically simulated ground calibration using the end-to-end model.

Further, a numerical simulation of the so-called "solar blip" phenomenon, an event in which an instrument is rapidly subjected to an excess of energy when the sun briefly enters its field-of-view during the satellite's entry and exit from the Earth's shadow, is conducted, allowing the responses of the actual radiometers to be studied.

2.0 The Earth-Viewing, Nonscanning, Active-Cavity Radiometer

2.1 Radiometer Description and Operation

Figures 2 and 3 show a cross-section and an assembly drawing of an ERBE Earth-viewing nonscanning active-cavity radiometer. Each radiometer consists of an optical front end and two silver conical cavities, one active and one passive. The optical front end consists of a truncated hemispherical field-of-view limiter and a base plate having a precision aperture at its center. The visible channels contain a Suprasil® hemispherical filter dome to remove the longwave component of the incident flux. Both cavities communicate thermally with a common heat sink through concentric cylindrical thermal impedances.

The field-of-view limiters and substrates of all channels have specularly reflecting coatings. These surfaces are covered by a highly emissive black coating($\alpha \cong 0.95$) for the visible channels and a low emissive vacuum deposited aluminum coating($\alpha \cong 0.04$) for the total channels [32]. The wide field-of-view (WFOV) channels view the entire Earth disk, while the medium field-of-view (MFOV) channels view the Earth with an Earth central angle of approximately 10 deg.

As shown in Figure 4, each cavity consists a 30-deg cone of length 14.94 mm bonded to a cylinder of diameter 8.00 mm and length 5.49 mm. The thermal impedance sleeves are cylinders 10.28 mm in diameter and 23.00 mm long. A coupling ring connects the thermal impedances to the cavities. The cavity components are constructed from 0.0635-mm (nominal) thick electrodeposited silver (99.99 percent pure).

The ends of the thermal impedances near the coupling ring are wound with platinum wire which acts as a resistance temperature detector (RTD). The windings, along with the external surface of the sleeve, are covered with an aluminized mylar insulation jacket.

The interiors of both cavities are coated with Chemglaze Z302, a specularly reflecting and highly emissive ($\alpha \cong 0.9$) black paint [32]. This special coating and the conical shape of the cavities assure a spectrally flat response over the 0.2 to 50 μm range.

The reference cavity views only a closed environment maintained at the heat sink temperature. During normal Earth viewing the reference cavity heater is turned off; thus, its temperature tracks that of the heat sink very closely. The active cavity views the scene (i.e., the Earth) and is maintained at a temperature of approximately 0.78°C [19] above that of the reference cavity by electrical substitution heating. The constant temperature difference is maintained via closed-loop feedback control of the active heater using inputs from the active and reference resistance temperature detectors. This thermal control is accomplished by balancing a resistance bridge circuit. The temperature differential is the result of a bias resistor selected for each detector to produce an imbalance in the bridge in the zero heater power condition. To maintain a balanced bridge the voltage to the active heater is varied such that its squared value varies inversely with the radiative power absorbed by the cavity. The channel output is a 13-bit digitized direct measurement of the active heater voltage, sampled every 0.8 s, ranging from 0 to 10 V. A block diagram and a simplified version of the circuit which implements this feedback function are shown, respectively, in Figures 5(a) and 5(b).

Complete dimensions and material properties for the Earth-viewing nonscanning radiometric channels are given in Tables 1 through 3.

2.2 Radiometer Characterization and Calibration

The ERBE instruments are the most completely characterized and most accurately calibrated instruments of their type ever to be flown [32]. The equations used to characterize the total (T) and shortwave (SW) radiant energy received by the radiometers during calibration are referred to as the ground count-conversion equations. The forms of these equations have been determined [33, 34] as

$$E_T' = A_V'V^2 + A_F'(T_F - \overline{T}_{F0}) + A_R'V_R^2 + B_T' \tag{2.1}$$

and

$$E_{SW}' = A_V'V^2 + A_F'(T_F - \overline{T}_{F0}) + A_R'V_R^2 + A_E'E_T + B_{SW}' , \tag{2.2}$$

where E' is the irradiance at the field-of-view limiter aperture (W/m^2), A_V' is a coefficient corresponding to the heater voltage term (W/m^2V^2), V is the active cavity heater voltage (V), A_F' is a coefficient corresponding to the field-of-view limiter temperature term (W/m^2K), T_F is the temperature of the field-of-view limiter (K), T_{F0} is the mean field-of-view limiter temperature during the calibration runs (K), A_R' is a coefficient corresponding to the reference cavity heater voltage (W/m^2V^2), V_R is the reference cavity heater voltage (V), A_E' is a coefficient meant to correct for the presence of the shortwave filter dome (-), and the offset terms $B'_{T,SW}$ (W/m^2) calibrate the instruments by means of internal calibration sources. For nominal Earth scenes V_R is set equal to zero.

To perform an absolute calibration, the instruments were allowed to view laboratory radiometric sources based on the International Practical Temperature Scale of 1968. Radiometric sources for the nonscanning channels consisted of a Master Reference Black Body (MRBB), an integrating sphere (ISP), and a solar simulator. Figure 6 shows the ERBE calibration chamber, which is an eight-foot diameter cylinder containing a Master Reference Black Body (MRBB) at one end and an integrating sphere (ISP) on the other end. The solar simulator can project a beam through a quartz window onto an instrument mounted on a carousel in the middle of the chamber. The chamber walls, chilled with liquid nitrogen, were used as a space reference source for scanner calibration.

Each radiometric source was viewed under a variety of conditions which simulated the radiative fields which the instruments would be subjected to in a space environment. Upon viewing each source, the instruments were allowed to reach a steady-state condition, at which time the voltage across the active cavity heater was recorded along with temperatures of the field-of-view limiter and various other instrument components. Additionally, the voltage in the reference cavity heater was set at one of three discrete levels while viewing the radiometric sources in order to increase the dynamic range of the instruments.

The voltage output and surface temperature data from ground calibration runs were used in a multiple regression analysis of Eqs. 2.1 and 2.2 to determine the ground count-conversion equation coefficients for each of the channels.

Inflight count-conversion equations are obtained by multiplying the ground count- conversion equations with an appropriate configuration factor F, which relates the radiative flux incident to the primary aperture to that at the field-of-view limiter aperture. These configuration factors are listed in Table 4. The resulting count conversion equations are

$$E_T = A_V V^2 + A_F T_F + A_R V_R^2 + B_T \qquad (2.3)$$

and

$$E_{SW} = A_V V^2 + A_F T_F + A_R V_R^2 + A_E E_T + B_{SW} , \qquad (2.4)$$

where

$$E_T = F E_T' , \qquad (2.5)$$

$$E_{SW} = F E_{SW}' , \qquad (2.6)$$

$$A_V = F A_V' , \qquad (2.7)$$

$$A_F = F A_F' , \qquad (2.8)$$

$$A_E = F A_E' , \qquad (2.9)$$

and

$$B_{T,SW} = F(B_{T,SW}' - A_F' \overline{T}_{F0}) . \qquad (2.10)$$

2.3 A Low-Order Physical Model of the Active-Cavity Radiometers

By employing a low-order physical model, theoretical values for the A_V and A_F coefficients can be represented by [35,36]

$$A_V = \frac{-1}{\tau_{sw} A_A R \alpha} \qquad (2.11)$$

and

$$A_F = -\frac{4\sigma\epsilon F_{CF} \tau_{LW} T_F^3}{F_{CS} \tau_{SW}} , \qquad (2.12)$$

where σ is the Stefan-Boltzman constant (5.670×10^{-8} W/m^2K^4), ϵ is the emissivity of the field-of-view limiter, τ_{LW} is the transmissivity of the filter dome to longwave radiation, τ_{SW} is the transmissivity of the filter dome to shortwave radiation, T_F is the temperature of the field-of-view limiter (K), F_{CF} is the view factor from the cavity to the field-of-view limiter, F_{CS} is the view factor from the cavity to the source of the incident thermal radiation, A_A is the area of the primary aperture (m^2), R is the electrical resistance of the active cavity heater element (Ω), and α is the effective absorptivity of the cavity. Equations 2.11 and 2.12 assume diffuse radiation and isothermal surfaces.

The offset terms B, are updated every two weeks by rotating the radiometers to look at onboard calibration sources. Values for the offset terms are obtained by rearranging Eqs. 2.3 and 2.4 to yield

$$B_T = F_{CS}\sigma T_{bb}^4 - A_V V_{bb}^2 - A_F T_F - A_R V_R^2 + B_{MRBB-IBB} \tag{2.13}$$

and

$$B_{SW} = -A_V V_{SWICS}^2 - A_F T_F - A_E E_T - A_R V_R^2 \ , \tag{2.14}$$

where T_{bb} is the temperature of the blackbody (K), V_{bb} is the active cavity heater voltage while looking at a blackbody (V), V_{SWICS} is the active cavity heater voltage when looking at a tungsten lamp (V), and $B_{MRBB-IBB}$ is the correction term which brings the internal blackbodies into agreement with the Master Reference Black Body (MRBB). Values for the $B_{MRBB-IBB}$ terms were determined by analyzing the differences in instrument output between the initial inflight Internal Black Body

(IBB) calibrations with previous ground calibrations which used the Master Reference Black Body (MRBB).

A low-order mathematical relationship which describes the interaction of the shortwave filter dome with the incident radiant energy (i.e., the A_E coefficient) was not determined by the ERBE Science Team; therefore, only an analysis of the A_V and A_F coefficients is reported for the visible channels.

Substitution of the physical and optical properties of the radiometers into Eqs. 2.11 and 2.12 results in theoretical values for the coefficients A_F and A_V. Calculation of the offset terms B requires an additional step. Inspection of Eq. 2.13 reveals that a $B_{MRBB\text{-}IBB}$ correction term is included. This correction term was developed using the values for A_V and A_F based on a statistical regression of calibration data and not the values used in the low-order physical model; therefore, it must be modified in order to be meaningfully introduced into the model.

The values obtained for the A_F and A_V coefficients using this low-order physical model appear in Table 5. Table 5 displays values for the count conversion equation coefficients for the low-order physical model of the radiometers as well as for the values the ERBE science team obtained for these coefficients through statistical regression of calibration data. Chapter 6 discusses how these coefficients were also obtained utilizing the high-level end-to-end numerical models.

The differences in flux obtained using the two sets of coefficients in Table 5 in the count conversion equation are shown in Figures 7 and 8 for the Earth Radiation Budget Satellite (ERBS) wide field-of-view total (WFOVT) and medium field-of-view total (MFOVT) channels, respectively. The fluxes correspond to data from approximately three orbits on April 17, 1985.

Figures 7 and 8 illustrate that the fluxes obtained with the low-order physical model are slightly lower than those corresponding to the statistical regression of ground calibration data. However, the fluxes based on the two sets of coefficients agree well for both channels. Differences of less than 0.5 percent for the wide field-of-view total channel and less than 1.25 percent for the medium field-of-view total channel result. These differences are based on nominal average total irradiances of 310 (W/m^2) for the wide field-of-view total channel and 150 (W/m^2) for the medium field-of-view total channel.

3.0 An Uncertainty Analysis for an ERBE Active-Cavity Radiometer

3.1 Uncertainty Analysis Methodology

A benefit of performing an uncertainty analysis is that it allows the experimenter to determine the individual contributions each independent variable makes to the total uncertainty in the system. Description of uncertainties in single-sample experiments is discussed by Kline and McClintock [37]. This statistically based methodology requires that the experimenter know, to the same stated probability, the uncertainty associated with each variable. Under this restriction if $R(p_1,p_2,\ldots,p_n)$ is a linear function of n independent variables p_i, the uncertainty of each of which is normally distributed, then the uncertainty in the result, ΔR, is related to the uncertainty in each of the variables, Δp_i, according to

$$\Delta R = \left[\left(\frac{\partial R}{\partial p_1} \Delta p_1 \right)^2 + \left(\frac{\partial R}{\partial p_2} \Delta p_2 \right)^2 + \ldots + \left(\frac{\partial R}{\partial p_n} \Delta p_n \right)^2 \right]^{1/2} . \tag{3.1}$$

It is emphasized that this result holds strictly only as long as Δp_1, Δp_2, ..., and Δp_n are all known to the same probability.

3.2 Application of Uncertainty Analysis Methodology

From Eq. 2.12 it is apparent that the coefficient A_F is a function of six independent variables,

$$A_F = A_F \left(F_{CF}, \tau_{LW}, \epsilon, T_F, F_{CS}, \tau_{SW} \right) \cdot \tag{3.2}$$

Applying the Kline and McClintock uncertainty analysis methodology to A_F yields

$$\frac{\partial A_F}{\partial F_{CF}} \Delta F_{CF} = \frac{4 \sigma \epsilon \tau_{LW} T_F^3}{F_{CS} \tau_{SW}} \Delta F_{CF} , \tag{3.3}$$

$$\frac{\partial A_F}{\partial \tau_{LW}} \Delta \tau_{LW} = \frac{4 \sigma \epsilon F_{CF} T_F^3}{F_{CS} \tau_{SW}} \Delta \tau_{LW} , \tag{3.4}$$

$$\frac{\partial A_F}{\partial \epsilon}\Delta\epsilon = \frac{4\sigma F_{CF}\,\tau_{LW} T_F{}^3}{F_{CS}\,\tau_{SW}}\Delta\epsilon \ , \tag{3.5}$$

$$\frac{\partial A_F}{\partial T_F}\Delta T_F = \frac{12\sigma\epsilon F_{CF}\,\tau_{LW} T_F{}^2}{F_{CS}\,\tau_{SW}}\Delta T_F \ , \tag{3.6}$$

$$\frac{\partial A_F}{\partial F_{CS}}\Delta F_{CS} = -\frac{4\sigma\epsilon F_{CF}\,\tau_{LW} T_F{}^3}{F_{CS}{}^2\,\tau_{SW}}\Delta F_{CS} \ , \tag{3.7}$$

and

$$\frac{\partial A_F}{\partial \tau_{SW}}\Delta\tau_{SW} = -\frac{4\sigma\epsilon F_{CF}\,\tau_{LW} T_F{}^3}{F_{CS}\,\tau_{SW}{}^2}\Delta\tau_{SW} \ . \tag{3.8}$$

For the total channels A_F is a function of only four independent variables because both transmissivity terms have a value of unity in the absence of a filter dome.

Similarly, from Eq. 2.11 it is apparent that A_V is a function of four independent variables,

$$A_V = A_V\left(\tau_{SW}, A_A, R, \alpha\right) \ . \tag{3.9}$$

Applying the Kline and McClintock uncertainty analysis methodology to the A_V term yields

$$\frac{\partial A_V}{\partial \tau_{SW}} \Delta\tau_{SW} = \frac{-1}{(\tau_{SW} A_A R\alpha)^2} A_A R\alpha\Delta\tau_{SW} \ , \tag{3.10}$$

$$\frac{\partial A_V}{\partial A_A} \Delta A_A = \frac{-1}{(\tau_{SW} A_A R\alpha)^2} \tau_{SW} R\alpha\Delta A_A \ , \tag{3.11}$$

$$\frac{\partial A_V}{\partial R} \Delta R = \frac{-1}{(\tau_{SW} A_A R\alpha)^2} \tau_{SW} A_A \alpha\Delta R \ , \tag{3.12}$$

and

$$\frac{\partial A_V}{\partial \alpha} \Delta\alpha = \frac{-1}{(\tau_{SW} A_A R\alpha)^2} \tau_{SW} A_A R\Delta\alpha \ . \tag{3.13}$$

For the total channels A_V is a function of only three independent variables because the transmissivity has a value of unity in the absence of a filter dome.

Likewise, from Eq. 2.13 it is apparent that the total channel offset term, B_T, is a function of six independent variables,

$$B_T = B_T \left(F_{CS}, T_{bb}, A_V, V_{bb}, A_F, T_F \right) \cdot \tag{3.14}$$

Applying the Kline and McClintock uncertainty analysis methodology to this term yields

$$\frac{\partial B_T}{\partial F_{CS}} \Delta F_{CS} = \sigma T_{bb}{}^4 \Delta F_{CS} \ , \tag{3.15}$$

$$\frac{\partial B_T}{\partial T_{bb}} \Delta T_{bb} = 4 F_{CS} \sigma T_{bb}{}^3 \Delta T_{bb} \ , \tag{3.16}$$

$$\frac{\partial B_T}{\partial A_V} = V_{bb}{}^2 \Delta A_V \ , \tag{3.17}$$

$$\frac{\partial B_T}{\partial V_{bb}} \Delta V_{bb} = 2 A_V V_{bb} \Delta V_{bb} \ , \tag{3.18}$$

$$\frac{\partial B_T}{\partial A_F}\Delta A_F = T_F \Delta A_F \ , \tag{3.19}$$

and

$$\frac{\partial B_T}{\partial T_F}\Delta T_F = A_F \Delta T_F \ . \tag{3.20}$$

The visible channel offset term, B_{SW}, is not treated here because a physical model of the A_E term is not available.

The uncertainties associated with the above coefficients and offset terms are obtained upon substituting the terms based on Eqs. 3.3 through 3.8, 3.10 through 3.13, and 3.15 through 3.20 into equations of the form of Eq. 3.1. Table 6 displays the uncertainties of the count conversion equation coefficients for each of the four ERBS nonscanning radiometric channels.

A benefit of the Kline and McClintock methodology of uncertainty analysis is that it allows the experimenter to determine the individual contributions each variable makes to the total uncertainty in the result, ΔR. Tables 7 and 8 display the contributions to the total uncertainties in A_F and A_V made by each independent variable in the theoretical (low-order model) relations, Eqs. 2.11 and 2.12, representing A_F and A_V, respectively. Results are displayed for each channel. Table 9 displays the contributions to uncertainty in B_T by each independent variable in Eq. 3.14. Discussion of the results in these tables is deferred until Chapter 6, where they are discussed in detail.

4.0 High-Order Radiative Analysis of the ERBE Nonscanning Channels

4.1 The Monochromatic Distribution Factor, D_{ijk}

The monochromatic distribution factor D_{ijk} is defined as the fraction of energy emitted from surface or volume element i in wavelength interval k which is absorbed by surface or volume element j. The distribution factor includes direct radiation from i to j as well as all possible diffuse and specular reflections and refraction through the filter dome.

Following this definition, the power absorbed by surface or volume element j, due to emission from surface element i, in the wavelength interval k, is given simply by

$$Q_{ijk} = \epsilon_i A_i e_{b,\Delta\lambda_k}(\Delta\lambda_k, T_i) D_{ijk} , \tag{4.1}$$

where ϵ_i, A_i, and T_i are the emissivity, surface area and temperature of element i, and $e_{b,\Delta\lambda}$ is the emissive power of element i in wavelength interval k.

Similarly, the power absorbed by surface or volume element j, due to diffuse emission from volume element i, in the wavelength interval k, is given as

$$Q_{ijk} = 4\kappa_{\Delta\lambda_k} V_i \, e_{b,\Delta\lambda_k}(\Delta\lambda_k, T_i) \, D_{ijk} \; , \tag{4.2}$$

where V_i and T_i are the volume and temperature of element i, $\kappa_{\Delta\lambda k}$ is the monochromatic absorption coefficient in wavelength interval k of element i, and $e_{b,\Delta\lambda}$ is the emissive power of volume element i in wavelength interval k. Equation 4.2 is independent of the element's optical depth, because this has already been considered in computing D_{ijk}.

The emissive power $e_{b,\Delta\lambda}$ of an element in a given wavelength interval is found by integrating Planck's blackbody radiation distribution function over the wavelength band of interest,

$$e_{b,\Delta\lambda_k} = \int_{\lambda_1}^{\lambda_2} \frac{2\pi C_1}{\lambda^5 (e^{C_2/\lambda T} - 1)} d\lambda \; , \tag{4.3}$$

where C_1 and C_2 are physical constants and T is the temperature of the element.

Finally, the total power absorbed by surface or volume element j due to emission from surface or volume element i is given by

$$Q_{ij} = \sum_k Q_{ijk} \, , \tag{4.4}$$

where the summation is over the wavelength bands of interest. Note that the analysis implies the need for a separate monochromatic distribution factor matrix for each wavelength interval k.

Three simple properties of monochromatic distribution factors exist which may be used to reduce the time and effort required for their calculation. They are

$$\sum_{i=1}^{n} D_{ijk} = 1.0, \quad j = 1,2,\ldots n \, , \tag{4.5}$$

$$\epsilon_i A_i D_{ijk} = \epsilon_j A_j D_{ijk}, \; i = 1,2,\ldots n, \; j = 1,2,\ldots n \, , \tag{4.6}$$

and

$$\sum_{i=1}^{n} \epsilon_i A_i D_{ijk} = \epsilon_j A_j, \quad j = 1,2,\ldots n \, . \tag{4.7}$$

Equation 4.5 is a statement of conservation of energy, while Eq. 4.6 expresses a reciprocity relationship. Equation 4.7 is a combination of Eqs. 4.5 and 4.6 obtained by summing Eq. 4.5 over i and applying Eq. 4.6 to the result. Equation 4.7 may be rearranged to represent the error in the distribution factors between each surface of the enclosure,

$$\text{percent error} = \left[1 - \frac{\sum_{i=1}^{n} \epsilon_i A_i D_{ijk}}{\epsilon_j A_j} \right] \times 100, \quad j = 1,2,\ldots n \; . \tag{4.8}$$

As the number of energy bundles emitted from each surface increases, the monochromatic distribution factors converge to their proper values, and the right hand side of Eq. 4.8 approaches zero.

4.2 Determination of Monochromatic Distribution Factors

The Monte-Carlo ray-trace method is used to compute the distribution of monochromatic radiation within the radiometer enclosure. The basic assumptions allowing the use of this method are that thermal radiation transport between two elements may be assumed to occur in discreet energy bundles, and that the laws of chance may be used to determine the disposition of each energy bundle as it interacts with an element.

The enclosure specified by the field-of-view limiter aperture and walls, the substrate, and the cavity (i.e., cone and cylinder) is subdivided into surface elements whose absorptivities, specularities, and temperatures are specified. In addition, the filter dome is subdivided into volume elements with known temperatures and transmissivities. Millions of energy bundles are then allowed to enter the enclosure through the field-of-view limiter aperture with directional and spatial distribution dependent upon the nature of the source field being simulated. Millions of energy bundles are also diffusely emitted from the field-of-view limiter walls, the substrate, the primary aperture, the cavity, and from the volume elements describing the filter dome. The path of each bundle is traced through the radiometer as it is reflected from, and transmitted through, the various surface and volume elements until it is eventually absorbed or exits through the field-of-view limiter aperture.

The procedure for determining the monochromatic distribution factors for axially symmetric enclosures is outlined below.

Step 1. Location and Direction of Emission

For diffuse emission, emission locations must be uniformly distributed over an element's surface or throughout its volume. For a hemispherical surface element, such as on the field-of-view limiter, the distribution of emission points along the z-direction is found from

$$z = r(1-\sqrt{R_\theta}) \;, \tag{4.9}$$

where z is measured from the origin of the hemisphere, r is the radius of the hemisphere and R_θ is a random number uniformly distributed between zero and one. For cylindrical surface elements, such as on the cylinder portion of the cavity, the z-coordinate of emission is found from

$$z = z_0 + R_z \Delta z \;, \tag{4.10}$$

where z_0 is the z-coordinate of the lowest edge of the element, Δz is the element height, and R_z is a random number uniformly distributed between zero and one. For an element of a horizontal disk, such as on the primary aperture and substrate, the z-coordinate is a constant value determined by the geometry. In this case uniform distribution of emission locations is a function of radius. The emission radius is found from

$$r = r_0 + \sqrt{R_r}\, \Delta r \;, \tag{4.11}$$

where r_0 is the inner radius of the element, R_r is a uniformly distributed random number between zero and one, and Δr is the difference between the inner and outer radii of the element. For elements of a conical surface, such as on the cavity, uniform distribution of emission points is also a function of radius, and the radial location of emission is found using Eq. 4.11. Once this radius is determined, the z-coordinate, measured from the vertex of the cone is determined by

$$z = \frac{\tan\theta}{r} , \tag{4.12}$$

where θ is the cone angle and r is the radius determined by Eq. 4.11. For hemispherical volume elements such as those in the filter dome, the radial emission location must first be determined using Eq. 4.11. The z-coordinate is then determined using Eq. 4.9.

Having found either the z-coordinate or radius, the azimuthal location of emission is found from

$$\phi = \Pi R_\phi , \tag{4.13}$$

where Π is the included azimuthal angle of the element in question, and R_ϕ is a uniformly distributed random number between zero and one.

Once the emission location has been determined, the emission direction is found. The direction is determined by the equations

$$\theta = \sin^{-1}\sqrt{R_\theta} \tag{4.14}$$

and

$$\phi = 2\pi R_{\phi} , \qquad (4.15)$$

where R_θ and R_ϕ are uniformly distributed random numbers between zero and one, θ is the angle between the local surface normal and the emission direction, and ϕ is the angle between the local surface tangent and the emission direction. These two angles (θ,ϕ) are then used to find the emission direction cosines. For volume elements an additional random number is used to determine into which of the two hemispheres the energy bundle is emitted.

Step 2. Determination of Intersection Points

With known direction cosines assigned to the energy bundle, the possible intersections it makes with the mathematical surfaces of the enclosure are easily determined. The equation which describes the path of the energy bundle is

$$\frac{x_2 - x_1}{l} = \frac{y_2 - y_1}{m} = \frac{z_2 - z_1}{n} , \qquad (4.16)$$

where (x_1,y_1,z_1) are the coordinates of the emission point; l, m and n are the emission direction cosines; and (x_2,y_2,z_2) is the point of intersection of the energy bundle's path with another surface of the enclosure. Possible intersections are found by solving Eq. 4.16 in conjunction with the mathematical relations describing the enclosure surfaces. Several possible intersection points are obtained for each curved surface, the number depending on the geometrical complexity of the surface. Only one possible intersection point exists for each planar surface of the enclosure. Those points which are physically impossible, either due to blockage, or due to limits of the actual geometry not reflected in the mathematical surface equations, are discarded. Of the remaining points, the nearest one to the emission location is chosen as the point of intersection.

Step 3. Absorption, Reflection, or Transmission?

In general, once an energy bundle intersects a surface, one of four events can occur: the bundle can be absorbed, it can be reflected specularly or diffusely, or it can be transmitted. These possible events may be seen in Figures 9 and 10. In the current work, for an intersection point not on the filter dome, the bundle will either be absorbed or reflected. In order to determine whether the bundle is absorbed, a uniformly distributed random number between zero and one is drawn and compared to the known surface absorptivity. If the random number is less than the surface absorptivity, the bundle is absorbed by the element and is recorded by a counter. At this stage the ray trace would be complete and one would return to Step 1 and emit a new bundle. If the random number is less than the known surface absorptivity, the bundle is reflected.

To determine the type of reflection, a uniformly distributed random number is obtained and compared to the reflectivity ratio, R, defined

$$R = \frac{\rho^s}{\rho^s + \rho^d} . \quad (4.17)$$

The reflectivity ratio may be thought of as the probability that a reflection will be specular. If the random number is less than R, the bundle undergoes a specular reflection as described in Step 4; otherwise a diffuse reflection occurs. Diffuse reflections are treated as diffuse emission described in Step 2.

If the intersection point occurs on the filter dome, then the possibility of transmission must also be considered. In the current effort, filtering in the dome is assumed to work by absorption; that is, reflections occur only when the angle between the incident energy bundle and the local surface normal are larger than the Brewster angle, θ_b. The Brewster angle is defined as

$$\theta_b = \tan^{-1}\frac{n_0}{n_1} , \quad (4.18)$$

where n_0 is the index of refraction of a vacuum and n_1 is the index of refraction of the filter dome material. Bundles whose incident angles are larger than the Brewster angle are assumed to be specularly reflected as described in Step 4.

For bundles that are not reflected, refraction in the filter dome is modeled according to Snell's law,

$$\frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{n_2}{n_1} , \tag{4.19}$$

where θ_1 and θ_2 are the angles incident and refracted beams make with the normal to the interface between two media having refractive indices of n_1 and n_2.

When an energy bundle in wavelength interval k enters a volume element in the filter dome, a uniformly distributed random number between zero and one is drawn and set equal to

$$P[\kappa_\lambda(\lambda)d] = 1.0 - \exp[-\kappa_\lambda(\lambda)d] , \tag{4.20}$$

where $P[\kappa_\lambda(\lambda)d]$ is the probability that the ray will be absorbed after travelling a distance d through a medium whose monochromatic absorption coefficient is $\kappa_\lambda(\lambda)$. Equation 4.20 is solved for d and this distance is compared to the path length l through the volume element that the ray would follow if it were not absorbed. If the distance d is greater than or equal to the path length l, the ray passes through the volume element without being absorbed. If on the other hand d is less than l, the ray will be absorbed in the dome volume element at a location determined by the value of d.

Step 4. Specular Reflection

The direction of specular reflection is determined by two criteria. The first states that the angles between the local surface normal and the paths of the incident and reflected energy bundle be equal. The second criteria requires that the incident path, the local surface normal, and the reflected path be planar. Mathematically these requirements may be stated as

$$\overline{U_2} = \overline{U_1} - 2(\overline{U_1}\bullet\overline{n})\overline{n} \, , \qquad (4.21)$$

where $\mathbf{U}_1$ is the vector representing the incident energy bundle, $\mathbf{U}_2$ is the vector representing the reflected energy bundle, and **n** is the local unit normal vector. At this point, one would return to Step 2 and continue through the steps until the bundle is absorbed.

These steps are repeated until a sufficiently large number of energy bundles N_i are emitted from each surface i, in each wavelength interval k, to correctly characterize the radiative exchange within the enclosure. Once enough energy bundles have been emitted, the monochromatic distribution factors are calculated as

$$D_{ijk} = \frac{N_{ijk}}{N_{ik}} \, , \qquad (4.22)$$

where N_{ijk} is the number of bundles emitted by element i which are absorbed by element j in wavelength interval k, and N_{ik} is the number of energy bundles emitted by element i, in wavelength interval k.

5.0 Model Formulation

5.1 Introduction

The development of equations which govern the transient electrical and radiative heating of the active-cavity radiometer in the completed end-to-end model are described in this chapter. These developments were originally presented by Tira [19]. The geometry of the end-to-end model is significantly more complex than that used by Tira, and as a consequence, equations describing radiative heating of the active cavity are correspondingly more complex. In addition, Tira's development had the resistors in the bridge circuit shown in Table 5(b) located on the wrong legs. Fortunately, this error did not affect his analysis, however, because he assumed that the reference cavity was maintained at the heat sink temperature.

5.2 Overview

The completed end-to-end model includes a transient two-dimensional finite-element characterization of the electrothermal behavior of the active cavity for all channels, along with a three-dimensional transient finite-difference model of heat conduction in the filter dome included for the visible channels. The radiative boundary conditions for both models are obtained using a Monte-Carlo ray-trace analysis, as described in Chapter 4. An assumption which was made in the analysis was that the values specified for the surface emissivities were constant and equivalent to the surface absorptivities over all wavelength intervals. That is, all surfaces are considered to be gray. While probably a good assumption for the black surfaces, and especially those in the cavity, this is of questionable validity for the polished aluminum surfaces for visible radiation.

The current model consists of 220 surface elements describing the internal surfaces of the instrument, 100 of which describe the surfaces of the cavity which can receive incident radiation, one surface element characterizing the plane of the field-of-view limiter aperture, and 60 volume elements which characterize the filter dome. A complete characterization of the surface and volume elements is given in Table 10.

This distribution of surface and volume elements was chosen with reference to the work done by Tira [19] and Haeffelin [39]. Tira showed that 100 surface elements was the optimal number to describe the electrothermal behavior of the cavity. This configuration includes ten azimuthal divisions, six axial divisions on the cone, and four on the cylinder. Haeffelin found that eight azimuthal divisions were sufficient to accurately represent the optical front end of the instrument; however, in order to maintain geometrical continuity, ten azimuthal divisions are used throughout the model.

Both the finite-element and finite-difference implementations are described in detail elsewhere [19,20,28]. The rest of this chapter deals with the numerical description of radiative power incident to the surfaces of the active cavity.

5.3 Radiative Heating of the Active Cavity

Radiation entering the active cavity emanates from either an instrument surface or from a body external to the instrument. While the spatial distribution of radiation emitted from instrument surfaces is constant, the spatial distribution of radiation from external bodies may vary with time. This requires that a new distribution factor matrix be computed each time the directional emissive properties of the external source change.

The net radiative input for the cavity, Q_{rad}, is the sum of the net radiative inputs, Q_{rad_j}, for each element j in the active cavity. The net radiative input for each element j, is equal to the difference between the power absorbed , $Q^{a}_{rad_j}$, and the power emitted, $Q^{e}_{rad_j}$, by that element, or

$$Q_{rad_j} = Q^{a}_{rad_j} - Q^{e}_{rad_j} \; . \tag{5.1}$$

The power absorbed by element j, $Q^{a}_{rad_j}$ includes radiation emitted from all elements in the model geometry which is absorbed by element j. For the total channels, $Q^{a}_{rad_j}$ is

$$Q^{a}_{rad_j} = Q_{scene_j} + Q_{fovl_j} + Q_{sub_j} + Q_{ring_j} + Q_{pa_j} + Q_{cav_j} \; , \tag{5.2}$$

and for the visible channels

$$Q^{a}_{rad_j} = Q_{scene_j} + Q_{fovl_j} + Q_{sub_j} + Q_{dome_j} + Q_{pa_j} + Q_{cav_j} \; , \tag{5.3}$$

where, Q_{scene_j} is the power absorbed from the scene, Q_{fovl_j} is the power absorbed from the field-of-view limiter, Q_{sub_j} is the power absorbed from the substrate, Q_{ring_j} is the power absorbed from the ring which links the primary aperture to the substrate, Q_{dome_j} is the power absorbed from the filter dome, Q_{pa_j} is the power absorbed from the primary aperture, and Q_{cav_j} is the power absorbed from all cavity elements.

These absorbed powers are calculated using

$$Q_j = \sum_{i=1}^{n} \sum_{k=1}^{m} \epsilon_i A_i e_{b,\Delta\lambda_k}(\Delta\lambda_k, T_i) D_{ijk} \tag{5.4}$$

for surface elements, and

$$Q_j = \sum_{i=1}^{n} \sum_{k=1}^{m} 4\kappa_{\Delta\lambda_k} V_i e_{b,\Delta\lambda_k}(\Delta\lambda_k, T_i) D_{ijk} \tag{5.5}$$

for volume elements. In Eqs. 5.4 and 5.5 the subscript k refers to wavelength interval $\Delta\lambda_k$, the subscript i refers to surface element i, ϵ_i is the emissivity of surface element i, A_i is the surface area of surface element i, V_i is the volume of volume element i, $e_{b,\Delta\lambda}$ is the emissive power of volume or surface element i in wavelength interval k, $\kappa_{\Delta\lambda k}$ is the monochromatic absorption coefficient of volume element i in wavelength interval k, and D_{ijk} is the monochromatic distribution factor from element i to element j in wavelength interval k.

Cavity element j emits radiation to its surroundings in wavelength interval $\Delta\lambda_k$ according to

$$Q^{e}_{rad_j} = \epsilon_j A_j e_{b,\Delta\lambda_k}(\Delta\lambda_k, T_j) \ , \tag{5.6}$$

where ϵ_j is the emissivity of surface element j, A_j is the surface area of surface element j, and $e_{b,\Delta\lambda}$ is the emissive power of element j in wavelength interval k.

The volummetric generation within each cavity element j, G_j, is obtained by substituting Eqs. 5.4, 5.5 and 5.6 into Eqs. 5.2 and 5.3 and dividing by the volume of cavity element j, $V_j = A_j\delta$, where δ is the element thickness. These substitutions yield

$$
\begin{aligned}
G_{rad_j} = {} & \frac{1}{A_j\delta}\left(\sum_{i=1}^{n}\sum_{k=!}^{n}\epsilon_i A_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{scene_j} + \frac{1}{A_j\delta}\left(\sum_{i=1}^{n}\sum_{k=1}^{k}\epsilon_i A_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{fovl_j} + \\
& \frac{1}{A_j\delta}\left(\sum_{i=1}^{n}\sum_{k=1}^{m}\epsilon_i A_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{sub_j} + \frac{1}{A_j\delta}\left(\sum_{i=1}^{n}\sum_{k=1}^{m}\epsilon_i A_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{ring_j} + \\
& \frac{1}{A_j\delta}\left(\sum_{i=1}^{n}\sum_{k=1}^{m}\epsilon_i A_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{pa_j} + \frac{1}{A_j\delta}\left(\sum_{i=1}^{n}\sum_{k=1}^{m}\epsilon_i A_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{cav_j} - \\
& \frac{1}{A_j\delta}\left(\epsilon_j A_j e_{b,\Delta\lambda_k}\right)_{cav_j}
\end{aligned}
\tag{5.7}
$$

for the total channels, and

$$
\begin{aligned}
G_{rad_j} = {} & \frac{1}{A_j\delta}\left(\sum_{i=1}^{n}\sum_{k=1}^{n}\epsilon_i A_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{scene_j} + \frac{1}{A_j\delta}\left(\sum_{i=1}^{n}\sum_{k=1}^{k}\epsilon_i A_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{fovl_j} + \\
& \frac{1}{A_j\delta}\left(\sum_{i=1}^{n}\sum_{k=1}^{m}\epsilon_i A_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{sub_j} + \frac{1}{A_j\delta}\left(\sum_{i=1}^{n}\sum_{k=1}^{m} 4\kappa_{\Delta\lambda_k} V_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{dome_j} +
\end{aligned}
$$

$$\frac{1}{A_j\,\delta}\left(\sum_{i=1}^{n}\sum_{k=1}^{m}\epsilon_i A_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{pa_j} + \frac{1}{A_j\,\delta}\left(\sum_{i=1}^{n}\sum_{k=1}^{m}\epsilon_i A_i e_{b,\Delta\lambda_k} D_{ijk}\right)_{cav_j} -$$

$$\frac{1}{A_j\,\delta}\left(\epsilon_j A_j e_{b,\Delta\lambda_k}\right)_{cav_j} \tag{5.8}$$

for the visible channels.

5.4 Electrical Heating of the Active Cavity

Fine wire electrical heaters are wound around the outside surfaces of both the active and reference cavities, as shown in Figure 4. It is assumed that heat generated by these wires is transferred entirely to the cavity walls. The resistance temperature detector (RTD), also a fine wire, is wound around the thermal impedance sleeve near its junction with the coupling ring, as shown in Figure 4. Two of the four arms of the deflection bridge shown in Figure 5(a) are fixed resistors (R_3 and R_4). A third arm is the RTD associated with the reference cavity (R_2). During normal Earth viewing, the voltage in the reference cavity is set equal to zero, and thus the temperature of the reference cavity and its RTD closely follow that of the heat sink. During ground, internal, and solar calibrations the voltage in the reference cavity is set at one of three discrete levels. This nonzero voltage raises the temperature of the reference cavity, and ultimately the resistance of this arm of the bridge.

The temperature of the fourth arm, which is that of the active cavity RTD (R_1), is not uniform around the thermal impedance since the cavity itself is not at a completely uniform temperature. However, this nonuniformity is on the order of something less than one degree kelvin. The temperature variation at the location of the RTD would be much smaller. Consequently, in the finite element analysis, the RTD is assumed to take the average temperature of the band of elements around the thermal impedance at that axial location.

The output voltage of the deflection bridge can be shown to be given by

$$V_1 = V_0\left[\frac{R_1}{R_1 + R_2} - \frac{R_3}{R_3 + R4}\right], \tag{5.9}$$

where V_0 is the bridge supply voltage, R_1 is the active cavity RTD resistance, R_2 is the reference cavity RTD resistance, and R_3 and R_4 are the two fixed resistances.

At the equilibrium condition,

$$R_1 = R + \Delta R_1 \tag{5.10}$$

and

$$R_2 = R + \Delta R_2 , \qquad (5.11)$$

where R (= $R_3 = R_4$) is the resistance of the active cavity RTD which would produce a zero bridge deflection. The quantity ΔR_1 is always greater than zero because the temperature of the active RTD is controlled to be higher than that of the reference cavity RTD. When the amount of radiation incident to the active cavity changes, the cavity adopts a new temperature distribution, and the average temperature of the RTD changes accordingly. This in turn causes the value of ΔR_1 to change. The other resistances remain constant at R. Substituting Eqs. 5.10 and 5.11 into Eq. 5.9 and taking into account the small values of ΔR_1 and ΔR_2 which permit the result to be linearized, the expression for the output voltage becomes

$$V_1 = \frac{V_0}{4}\left[\frac{\Delta R_1 - \Delta R_2}{R}\right] , \qquad (5.12)$$

or directly in terms of R_1 and R_2,

$$V_1 = \frac{V_0}{4}\left[\frac{R_1 - R_2}{R}\right] . \qquad (5.13)$$

The resistances of the RTD's are assumed to vary linearly with temperature; therefore, they can be expressed by

$$R_1 = R\left[1 + \alpha(T_1 - T_{hs})\right] \tag{5.14}$$

and

$$R_2 = R\left[1 + \alpha(T_2 - T_{hs})\right] , \tag{5.15}$$

where α is the resistance-temperature coefficient of the RTD and T_1, T_2 and T_{hs} are the temperatures of the active and reference RTD's and the heatsink, respectively. Substitution of Eqs. 5.14 and 5.15 into Eq. 5.13 gives the output voltage in terms of the temperatures,

$$V_1 = \frac{\alpha}{4} V_0\left[T_1 - T_2\right] . \tag{5.16}$$

Note that T_1 is always maintained higher than T_2, so that V_1 is never zero. The input voltage to the circuit integrator is the difference between the output voltage of the bridge V_1 and the bias voltage V_b. The integrator processes this voltage difference to give an output voltage V_2, which is the energizing voltage for the electric heater wrapped around the active cavity. The output voltage of the circuit as a function of time can be expressed by the differential equation

$$\frac{dV_2}{dt} = \frac{V_b - V_1}{\tau} , \tag{5.17}$$

where τ is the electrical time constant, $\tau = R_cC_c$. Upon substitution for V_1 from Eq. 5.16, Eq. 5.17 becomes

$$\frac{dV_2}{dt} = \frac{V_b}{\tau} - \frac{\alpha}{4\tau}V_0(T_1 - T_2) . \tag{5.18}$$

For steady-state conditions with a specified temperature drop ΔT between the active and reference RTD's, the right hand side of Eq. 5.18 is zero, which gives for the bias voltage

$$V_b = \frac{\alpha}{4}V_0\Delta T . \tag{5.19}$$

Using Eq. 5.19 to eliminate V_b from Eq. 5.18, yields

$$\frac{dV_2}{dt} = \frac{\alpha}{4\tau}V_0\left[\Delta T + (T_2 - T_1)\right] . \tag{5.20}$$

The values of V_2 at two consecutive time steps n and n+1 can then be estimated as

$$V_2^{n+1} = V_2^n + \Delta t \frac{\alpha}{4\tau} V_0 [\Delta T + (T_2 - T_1)] \ . \tag{5.21}$$

The heater power input Q_{elec} is given by

$$Q_{elec} = \frac{V_2^2}{R_{hw}} \ , \tag{5.22}$$

where R_{hw} is the resistance of the heater wire.

The volummetric heat generation in each surface element j covered by the heater wire is given by

$$G_{elec_j} = \frac{Q_{elec}}{A_{hw}\delta} \ , \tag{5.23}$$

where A_{hw} is the total area of the cavity covered by the heater wire and δ is the thickness of the cavity material. In terms of the heater voltage G_{elec_j} is expressed by

$$G_{elec_j} = \frac{V_2^2}{R_{hw}A_{hw}\delta} \ . \tag{5.24}$$

An expression for the volummetric heat generation at time step n+1 may be obtained by substituting Eq. 5.21 into Eq. 5.24, giving

$$G_{elec_j}^{n+1} = \frac{1}{R_{hw} A_{hw} \delta}\left(V_2^n + \Delta t \frac{\alpha}{4\tau} V_0 \left[\Delta T + (T_2 - T_1)\right]\right)^2 . \tag{5.25}$$

As expected, for steady-state conditions Eq. 5.25 reduces to

$$G_{elec_j} = \frac{V_2^2}{R_{hw} A_{hw} \delta} . \tag{5.26}$$

6.0 Results and Discussion

6.1 Calibration of the End-to-End Model

Calibration of the end-to-end model for the various channels was conducted by spectrally specifying radiative fluxes of sources located in the plane of the field-of-view limiter aperture. The magnitudes of these diffuse radiative source fields were specified along with their blackbody temperatures. Specification of the blackbody temperatures allows the spectral distribution of the radiative flux to be determined from Planck's blackbody radiation distribution function, as described in Chapter 4. Additionally, temperatures were specified for the field-of-view limiter, substrate, and primary aperture surface elements, and calculated for the cavity surface elements and filter dome volume elements. Knowledge of these temperatures allows the determination of both the magnitude and spectral distribution of power emitted from all instrument surfaces. For the visible channels, temperatures of the filter dome elements were obtained from Haeffelin's [28] thermal diffusion code. Finally, the reference cavity heater voltage was set at one of three discrete levels in order to increase the dynamic range of the sensors. The temperatures and radiative fluxes used in the simulated calibrations were obtained from actual ground calibrations run conducted by TRW [35]. Values are given in Tables 11 through 14.

For each simulation, the filter dome thermal diffusion code was first allowed to reach a steady-state condition. The resulting steady-state temperatures of the filter dome volume elements were then used in the electrothermal cavity model. Once the cavity model reached a steady-state condition, the calculated voltage in the active-cavity heater wire was recorded. These voltage values were used in conjunction with the corresponding field-of-view limiter temperatures to determine the ground count-conversion equation coefficients in Eqs. 2.1 and 2.2.

In performing these calibration simulations, two assumptions are made. The first is that the field-of-view limiter, substrate, and primary aperture are isothermal over the time period of the simulations. The second assumption is that the source fields may be modeled as a diffuse source located in the plane of the field-of-view limiter aperture.

During these simulations it was determined that the response of the detector was very sensitive to the thickness of the cavity walls. The manufacturing tolerance on the cavity thickness is quite large relative to its stated nominal value, that is 0.0635 ± 0.0254 mm. The cavity thickness used in the model was adjusted within this tolerance to fine tune the model's response. The cavity thickness was varied until the steady-state response of the end-to-end model agreed with that of the actual instrument. The thickness variations used are all on the order of one-half or less of the stated tolerance. Final values for cavity thicknesses used in the model are given in Table 15.

The voltage outputs of the actual instruments, the voltage outputs of the end-to-end models, and the differences between the two outputs for each calibration run are given in Tables 16 through 19. Figures 11 through 14 display the voltage output of the wide and medium, total and visible channel end-to-end models during simulated ground calibrations as a function of both the magnitude of the incident radiation and the reference cavity heater voltage. The voltage output of the visible channel models is further subdivided according to the spectral distribution of the specified incident radiative flux. Defining the overall gains of these models as the slopes of the best-fit lines correlating the data in Figures 9 through 12, or

$$\text{Gain} = \frac{\Delta V_{out}}{\Delta \text{Flux}_{in}} , \tag{6.1}$$

allows a direct comparison between the end-to-end models and the actual instruments. In certain cases there are only two data points, and so the best fit line is "perfect" in these cases. The calculated gains of the actual instruments and end-to-end models as a function of reference cavity heater voltage are shown in Table 20. Voltage outputs of the actual shortwave channel radiometers when subjected to longwave radiometric calibration sources (i.e. the MRBB) have not been obtained; therefore gains for the shortwave channel radiometers are not presented for longwave sources. Inspection of Table 20 shows that the differences between the instrument and model gains for normal Earth viewing (i.e. the voltage across the reference heater is set to zero) are small, 1.18 and 1.88 percent for the wide field-of-view total and visible channels, respectively, and 5.16 and 4.44 percent for the medium field-of-view total and visible channels, respectively.

Application of a multiple regression analysis to Eqs. 2.1 and 2.2 yields statistically based ground count-conversion coefficients. The results of such an analysis appear in Table 21, which displays the ground count conversion equations determined by the ERBE Science Team (EST) as well as those determined by the current end-to-end model (MODEL). The A_V', A_R', B_T' and B_{SW}' coefficients obtained with the end-to-end models agree well with the coefficients obtained during actual ground calibrations of the instruments. Tables 22 through 25 display the specified radiative fluxes emitted by the simulated radiometric calibration source, located in the plane of the field-of-view limiter aperture, during the calibration simulations, the radiative fluxes measured by the end-to-end models during the calibration simulations, and the differences between the specified and measured fluxes. The fluxes measured by the wide field-of-view models agree with the known fluxes within an average percent difference of -0.0025 and 0.0207 percent for the total and visible channels, respectively. The fluxes measured by the medium field-of-view models agree with the known fluxes within an average percent difference of -0.006 and 0.0085 percent for the total and visible channels, respectively. Agreement is better for the wide field-of-view total channel, as expected, since the medium field-of-view channels view a considerable portion of the field-of-view limiter walls, introducing more possibility for modelling error because of the isothermal field-of-view limiter assumption.

6.2 Comparison of Inflight Count-Conversion Equation Coefficients

Table 26 displays the inflight count conversion coefficients A_F and A_V determined by three methods: the low-order radiative model, the end-to-end model, and ground calibration by the ERBE science team. The three methods of calculating the A_V coefficient agree to within four percent for the wide field-of-view channels and seven percent for the medium field-of-view channels. Agreement for the A_F coefficient is not as good. An ideal application of a multiple regression analysis to Eqs. 2.1 and 2.2 would require two things. First, that each variable be varied over a sufficiently large range. Second, that the matrix which must be inverted to complete the analysis must not be ill conditioned. Neither of these criteria were met. The field-of-view limiter temperature was varied over less than 2.5 K for the actual ground calibration runs, which was not adequate, and the matrix which must be inverted was ill conditioned, containing values from 10^1 to 10^6. Not meeting these criteria results in values for some of the count-conversion equation coefficients which are not physically realistic; however, the coefficients are quite adequate to model the fluxes within the calibration window. Since the inflight field-of-view limiter temperature does not differ greatly (<1.5 K) from the ground calibration field-of-view limiter temperature range, errors introduced by not varying the field-of-view limiter temperature over a large enough range are quite small.

6.3 Sources of Uncertainty in the ERBE Active-Cavity Radiometers

A benefit of the Kline and McClintock methodology of uncertainty analysis is that it allows the experimenter to determine the individual contributions each variable makes to the total uncertainty in the result, ΔR. Table 6 displays the uncertainties of the count conversion equation coefficients for each of the four ERBS nonscanning radiometric channels. Tables 7 and 8 display the contributions to the total uncertainties in A_F and A_V made by each independent variable in the theoretical (low-order model) relations, Eqs. 2.11 and 2.12, representing A_F and A_V, respectively. Results are displayed for each channel. Table 9 displays the contributions to the uncertainty in B_T made by each independent variable in Eq. 3.14.

6.3.1 Uncertainty in the A_F coefficient

From Table 7 it is apparent that the uncertainty in the A_F coefficient for the total channels is largely due to uncertainty in the knowledge of the field-of-view limiter temperature. This uncertainty of the field-of-view limiter temperature is due mainly to the linearization of this temperature in the count conversion equation. For the visible channels, uncertainty in the knowledge of the transmissivity of the filter dome to longwave radiation is the major source of uncertainty in the A_F coefficient.

6.3.2 Uncertainty in the A_V coefficient

From Table 8 it is apparent that the uncertainty in the A_V coefficient for the total channels is largely due to the manufacturing tolerances on the area of the primary aperture. For the visible channels uncertainty in the knowledge of the transmissivity of the filter dome to longwave radiation is the major source of uncertainty.

6.3.3 Uncertainty in the B_T term

From Table 9 it is apparent that the uncertainty in the B_T term for the WFOVT channel is associated with how well the A_V coefficient is known. We have already seen that the uncertainty in the coefficient A_V is largely due to the manufacturing tolerances on the area of the primary aperture. For the MFOVT channel the largest contributor of uncertainty is that of the coefficient A_F. As we have seen, the uncertainty in the coefficient A_F is due to the uncertainty in the knowledge of the temperature of the field-of-view limiter.

6.3.4 Orbital variation of the uncertainties

The results of the uncertainty calculations may be seen in Figures 15 and 16, which display fluxes based on the low-order physical model for the WFOV and MFOV total channels, respectively. Uncertainty ranges are also shown on these graphs, which display a 90-min running average of the actual data product. This means that each plotted point is the average value of the data ranging 45 min on either side of the point. This is also referred to as an orbital average since one orbit is 90 min in duration. For the WFOV total channel (Figure 15) the uncertainty is approximately ±7.5 W/m^2, or roughly ±2.5 percent, and for the MFOV total channel (Figure 16) the uncertainty is approximately ±15 W/m^2, or roughly ±10 percent.

6.3.5 Uncertainty as a function of solar zenith angle

Although it is not readily apparent from Figures 15 and 16, the magnitude of the uncertainty is a weak function of solar zenith angle (i.e., a function of the spectral mix of the incident energy). Further investigation is needed to determine whether this dependence is simply a function of the intensity of the incident thermal radiation, or if it is a function of the spectral distribution of the incident radiation, or some combination of both. When the solar zenith angle has a low value (~0 to 50 deg) the incident radiation contains a large shortwave spectral component, and when the solar zenith angle is large (≥ ~120 deg) the incident thermal radiation has a spectral distribution that is almost entirely longwave.

The relationship between solar zenith angle and uncertainty in the data product is displayed in Figures 17 and 18. The labels "day" and "night" refer to the values of these uncertainties averaged over the time period when the solar zenith angle varies between 0 and 50 deg and 125 to 180 deg, respectively, while the satellite is looking at a typical Earth scene. The total uncertainty in the data product is the sum of these individual components. Figures 17 and 18 show that the uncertainty of the data product is slightly lower during typical "day" scenes than during typical "night" scenes, and that most of this difference is accounted for by the AV coefficient. These differences are shown to be statistically significant at P values of less than 0.0001.

6.3.6 Uncertainty Analysis Conclusions

The results of this investigation clearly show that the uncertainty associated with the WFOVT channel is much less than that of the MFOVT channel. The reasons for the much larger uncertainty in the MFOVT channel is the larger uncertainty in the $A_F T_F$ term of the count conversion equation for this channel. The uncertainty of this term for the MFOVT channel is inherently larger than in the case of the WFOVT channel because in the former case the field-of-view limiter fills a much larger portion of the cavity field of view than in the latter. However, this fact alone is not sufficient to explain the much larger values of the uncertainty for this channel. Rather, it is the combination of the relatively large uncertainty in the temperature of the field-of-view limiter and the inherently larger uncertainty in the $A_F T_F$ term for the MFOV count conversion equation which causes the observed uncertainty.

Analysis of the $A_F T_F$ terms for the visible channels shows that the uncertainty of this term is much smaller for these channels, as would be expected. The hemispherical filter dome absorbs essentially all of the longwave thermal radiation emitted from the field-of-view limiter, effectively eliminating the longwave component of any irradiance of the cavity by the field-of-view limiter.

6.4 Dynamic Simulation of the "Solar Blip" Event

The "solar blip" event is a phenomenon in which the wide field-of-view instruments are rapidly subjected to an excess of energy when the sun briefly enters the instrument field-of-view during the satellite's entry and exit from the Earth's shadow. Steep transients associated with this event saturate and introduce errors in the instrument data product, resulting in a loss of data. This event has been simulated utilizing the end-to-end model, thereby allowing the responses of the real instruments during this event to be investigated.

Two types of solar blip phenomena occur. Both the satellite's entry into the Earth's shadow and the satellite's exit from the Earth's shadow produce solar blip events. The solar blip event corresponding to the satellite's exit from the Earth's shadow has been simulated. This choice was based on the fact that the rise in radiative flux incident to the radiometers associated with the satellite's emergence from the Earth's shadow is steeper than for entrance, and therefore takes the instrument outside of its designed performance envelope more rapidly.

From April 17, 1985, solar blip data obtained from the Earth Radiation Budget Satellite (ERBS), the duration of a solar blip event was determined to be 120 s, where duration is defined as the length of time that the sun emits energy directly to the cavity. By fitting smooth curves to data during an April 17, 1985, solar blip event, it was determined that the magnitude of the radiative flux incident to the radiometers and the field-of-view limiter temperatures during an event are as shown in Figures 19 and 20, respectively. Additionally it was determined that the solar zenith angles ranged from 64 to 71 deg, increasing by 0.5 deg every 8 s.

During the simulated event it is assumed that the longwave component of radiation remains constant since the only longwave source is the unchanging Earth. The shortwave component is then the difference between the total incident radiation and the assumed constant longwave source. The magnitude of the constant longwave component for purposes of the simulation was set equal 180 W/m^2, which is representative of the average total channel night-time Earth measurements for April 17, 1985. The results of implementing this procedure are shown in Figure 19.

In general, energy emitted by the sun that reaches the instrument directly is modeled as collimated radiation, incident at the solar zenith angle, and energy emitted by the Earth that reaches the instrument is modelled as diffuse. That is, in an event such as the solar blip, the incident radiation contains both diffuse and collimated components.

The monochromatic distribution factors discussed in Chapter 4 are, in general, calculated for either collimated radiation at a given angle, or diffuse radiation. The relation

$$\sum_{i=1}^{n} D_{ijk} = 1.0 \ , \ j=1,2,...n \ , \qquad (6.4)$$

first seen in Eq. 4.5, makes it possible to calculate distribution factors for a mixed scene such as the solar blip event. The first step in modelling a source with both diffuse and collimated components is to calculate distribution factor matrices for both the collimated and diffuse sources. The collimated distribution factor matrix is then multiplied by a weighting factor W_1, such that

$$W_1 = \frac{\Phi_{col}}{\Phi_{col} + \Phi_{dif}} , \tag{6.5}$$

where Φ_{col} represents the magnitude of the collimated component of flux emitted by surface i, and Φ_{dif} the diffuse component. The diffuse distribution factor matrix is multiplied by W_2 such that

$$W_2 = 1.0 - W_1 . \tag{6.6}$$

The two resulting distribution factor matrices are then added together, creating an appropriately weighted matrix which obeys Eq. 6.4 while accurately characterizing the mixed scene. In the case of time-varying scenes this method implies the need for calculating this weighted distribution factor matrix for each time step. For the case of the simulated solar blip event this means calculating 120 weighted distribution factor matrices for each channel. However only 16 collimated and one diffuse distribution factor matrices need to be calculated using the ray-trace model since the incident angle varies only every eight seconds.

The incident flux predicted by the total and shortwave channel models, due to the specified input flux for the simulated solar blip event, is shown in Figures 21 and 23. The corresponding average temperatures of the resistance temperature detectors are shown in Figures 22 and 24. The predicted longwave component of flux is shown along with the assumed incident value in Figure 25.

The end-to-end model shows that the total and visible channel models overpredict the specified incident flux by 3.5 and 17.6 percent, respectively. Additionally, the peak flux values predicted by both the total and visible channels lag the specified input fluxes by 7 s. However, by the end of the simulated event, the predicted fluxes once again agree with the specified incident fluxes. Thus, although the period of the predicted fluxes matches that of the specified incident fluxes, the shape of the curve is skewed upward and to the right for both the total and visible channels.

The predicted longwave component shown in Figure 25 was found by subtracting the predicted visible component from the predicted total flux. Inspection of this figure reveals that the predicted longwave component is considerably less than the specified input value, reaching a minimum 8 s after both the predicted total and predicted shortwave fluxes peak. This trend is representative of flight data which also significantly underpredicts the longwave component.

By integrating the actual and predicted fluxes over the length of the simulation and multiplying by the area of the field-of-view limiter aperture, a comparison of the specified and predicted energy incident to the instruments can be made. Table 27 presents the results of this comparison. Following this methodology, the end-to-end model overpredicts the amount of energy received by 2.17 percent for the total channel and by 15.83 percent for the visible channel.

7.0 Conclusions and Recommendations

7.1 Conclusions

The following conclusions can be drawn from the results presented in this thesis:

1. An end-to-end model has been completed for the Earth-viewing, nonscanning, radiometric channels of the Earth Radiation Budget Experiment. The model permits the sensitivity of the instrument output to variations in the thermophysical properties of the detector to be determined.
2. The comparison of simulated and actual calibration procedures demonstrates that the model is a valid tool with which to investigate

anomolous behavior in the archived data product.

3. The three methods used to calculate the A_V coefficient, the low-order physical model, the end-to-end model, and ground calibration by the ERBE science team, agree to within four percent for the wide field-of-view channels and seven percent for the medium field-of-view channels.
4. Uncertainty in the A_F coefficient is largely due to uncertainty in the knowledge of the field-of-view limiter temperature for the total channels while uncertainty in the knowledge of the transmissivity of the filter dome to longwave radiation is the major source of uncertainty in the A_F coefficient for the visible channels.
5. Uncertainty in the A_V coefficient is largely due to the manufacturing tolerances on the area of the primary aperture for the total channels while uncertainty in the knowledge of the transmissivity of the filter dome to longwave radiation is the major source of uncertainty in the A_V coefficient for the visible channels.
6. Uncertainty in the B_T term for the wide field-of-view channel is due largely to uncertainty in the A_V coefficient while uncertainty in the B_T term for the medium field-of-view channel is due largely to uncertainty in the A_F coefficient.
7. Simulation of a solar blip event demonstrates that the total channel model predicts the total energy arriving to the instrument during this

steep transient to within 2.25 percent. However, the visible channel model does not do as good a job of predicting the shortwave component of energy arriving at the instrument during these same transients, overpredicting the actual amount by nearly 16 percent.

6.2 Recommendations

The following recommendations are made as a result of this study:

1. The current end-to-end model should be coupled with a thermal diffusion model of the field-of-view limiter in order to assess the influence of field-of-view limiter temperature variations on the detectors' response.
2. A parametric analysis should be completed to assess the influence of perturbations of the thermophysical properties of the instrument on the detectors' response.

Table 1. Geometry for the Earth-viewing nonscanning radiometric channels located on the Earth Radiation Budget Satellite, ERBS. (all dimensions in mm)

Dimension	WFOVT[1]	MFOVT[2]	WFOVSW[3]	MFOVSW[4]
FOVL[5] radius	38.10	36.8554	38.10	36.8554
FOVL[5] height	12.70	28.829	12.70	28.829
Primary Aperture Diameter	6.35	6.35	6.35	6.35
Cavity Diameter	8.00	8.00	8.00	8.00
Cone Height	14.94	14.94	14.94	14.94
Cylinder Height	5.49	5.49	5.49	5.49
Thermal Impedance Diameter	10.28	10.28	10.28	10.28
Thermal Impedance Height	23.00	23.00	23.00	23.00
Cavity Thickness	0.0536	0.0498	0.0573	0.0659

[1] Wide Field-of-View Total channel
[2] Medium Field-of-View Total channel
[3] Wide Field-of-View Shortwave channel
[4] Medium Field-of-View Shortwave channel
[5] Field-of-View Limiter

Table 2. Thermophysical properties for the Earth-viewing nonscanning radiometric channels located on the Earth Radiation Budget Satellite, ERBS.

Property	WFOVT[1]	MFOVT[2]	WFOVSW[3]	MFOVSW[4]	Units
FOVL[5] absorptivity α	0.04	0.04	0.95	0.95	-
FOVL[5] Reflectivity Ratio R	0.90	0.90	0.90	0.90	-
Substrate absorptivity α	0.95	0.95	0.95	0.95	-
Substrate Reflectivity Ratio R	0.90	0.90	0.90	0.90	-
Primary Aperture absorptivity α	0.95	0.95	0.95	0.95	-
Cavity absorptivity α	0.90	0.90	0.90	0.90	-
Cavity Conductivity k	429.0	429.0	429.0	429.0	W/m-K
Specific Heat c	235.0	235.0	235.0	235.0	J/Kg-K
Density ρ	10500.0	10500.0	10500.0	10500.0	Kg/m^3
Electrical time constant τ	0.005	0.005	0.005	0.005	s

[1] Wide Field-of-View Total channel
[2] Medium Field-of-View Total channel
[3] Wide Field-of-View Shortwave channel
[4] Medium Field-of-View Shortwave channel
[5] Field-of-View Limiter

Table 3. Filter dome transmissivities for the shortwave Earth-viewing nonscanning radiometric channels located on the Earth Radiation Budget Satellite, ERBS.

Wavelength Interval (μm)	Transmissivity
0 - 0.15	0.0001
0.15 - 0.20	0.4350
0.20 - 0.30	0.8295
0.30 - 2.50	0.9275
2.50 - 3.00	0.9350
3.00 - 3.50	0.9225
3.50 - 4.00	0.8475
4.00 - 4.50	0.5850
4.50 - 5.00	0.1950
5.00 - ∞	0.0001

Table 4. Configuration factors between the primary aperture and the field-of-view limiter aperture for the Earth-viewing nonscanning radiometric channels located on the Earth Radiation Budget Satellite, ERBS.

Channel	Configuration Factor F
WFOVT[1]	0.8692
MFOVT[2]	0.4240
WFOVSW[3]	0.8797
MFOVSW[4]	0.4240

[1] Wide Field-of-View Total channel
[2] Medium Field-of-View Total channel
[3] Wide Field-of-View Shortwave channel
[4] Medium Field-of-View Shortwave channel

Table 5. Comparison of inflight count-conversion equation coefficients for April 17, 1985, determined by both a low order physical model and the utilization of multiple regression methodology by the ERBE Science Team (EST).

Channel	Coefficient	EST	Physical Model	Percent Difference[1]	Units
WFOVT[2]	A_F	-1.3968	-.2391	82.89	W/m^2K
	A_V	-22.7873	-22.5212	1.17	W/m^2V^2
	B_T	1703.58	1355.13	20.45	W/m^2
MFOVT[3]	A_F	-.9230	-2.1244	-130.16	W/m^2K
	A_V	-22.7093	-22.6721	0.16	W/m^2V^2
	B_T	1272.12	1625.32	-27.76	W/m^2
WFOVSW[4]	A_F	-0.6502	-0.0056	99.14	W/m^2K
	A_V	-25.7758	-24.8279	3.69	W/m^2V^2
MFOVSW[5]	A_F	1.2242	-0.05631	104.60	W/m^2K
	A_V	-25.5630	-24.5551	3.94	W/m^2V^2

[1] percent difference $= \frac{\text{EST} - \text{Model}}{\text{EST}} \times 100$

[2] Wide Field-of-View Total channel

[3] Medium Field-of-View Total channel

[4] Wide Field-of-View Shortwave channel

[5] Medium Field-of-View Shortwave channel

Table 6. Uncertainties in the count-conversion equation coefficients for the Earth Radiation Budget Satellite, ERBS, based on a low-order physical model.

Channel	Coefficient	Value	Uncertainty	Percent Uncertainty[1]	Units
WFOVT[2]	A_F	-0.2391	0.0030	1.25	W/m^2K
	A_V	-22.5212	0.0745	0.33	W/m^2V^2
	B_T	1355.13	3.51	0.26	W/m^2
MFOVT[3]	A_F	-2.1244	0.0242	1.14	W/m^2k
	A_V	-22.6721	0.0751	0.33	W/m^2V^2
	B_T	1625.32	7.66	0.47	W/m^2
WFOVSW[4]	A_F	-0.0056	0.0004	7.68	W/m^2K
	A_V	-24.8279	0.1586	0.64	W/m^2V^2
MFOVSW[5]	A_F	-0.05631	0.0042	7.47	W/m^2K
	A_V	-24.5551	0.1569	0.64	W/m^2V^2

[1] percent uncertainty $= \frac{\text{Uncertainty}}{\text{Value}} \times 100$

[2] Wide Field-of-View Total channel
[3] Medium Field-of-View Total channel
[4] Wide Field-of-View Shortwave channel
[5] Medium Field-of-View Shortwave channel

Table 7. Sources of uncertainty in the A_F coefficient for each radiometric channel based on a low-order physical model.

Variable	WFOVT[1]	MFOVT[2]	WFOVSW[3]	MFOVSW[4]	Units
τ_{LW}	0	0	0.000415	0.004140	-
τ_{SW}	0	0	0.000031	0.000162	-
ϵ	0.000048	0.004249	0.000011	0.000113	-
F_{CF}	0.000954	0.008483	0.000022	0.000201	-
F_{CS}	0.000951	0.008518	0.000022	0.000226	-
T_F	0.002619	0.020566	0.000084	0.000657	K
ΔA_F[5]	0.002983	0.024198	0.000425	0.004207	W/m^2K

[1] Wide Field-of-View Total channel
[2] Medium Field-of-View Total channel
[3] Wide Field-of-View Shortwave channel
[4] Medium Field-of-View Shortwave channel
[5] $\Delta A_F = \sqrt{\sum (\text{Variables})^2}$

Table 8. Sources of uncertainty in the A_V coefficient for each radiometric channel based on a low-order physical model.

Variable	WFOVT[1]	MFOVT[2]	WFOVSW[3]	MFOVSW[4]	Units
τ_{SW}	0	0	0.1356	0.1342	-
A_A	0.0676	0.0680	0.0745	0.0737	m^2
R	0.0161	0.0163	0.0179	0.0175	Ω
α	0.0270	0.0272	0.0298	0.0295	-
ΔA_V[5]	0.0745	0.0751	0.1586	0.1568	W/m^2V^2

[1] Wide Field-of-View Total channel

[2] Medium Field-of-View Total channel

[3] Wide Field-of-View Shortwave channel

[4] Medium Field-of-View Shortwave channel

[5] $\Delta A_V = \sqrt{\sum (\text{Variables})^2}$

Table 9. Sources of uncertainty in the B_T term for the wide and medium field-of-view total channels based on a low-order physical model.

Variable	WFOVT[1]	MFOVT[2]	Units
F_{CS}	1.463	0.711	-
T_F	0.012	0.106	K
T_{bb}	0.251	0.121	K
V_{bb}	0.351	0.333	V
A_F	0.876	7.110	W/m^2K
A_V	3.035	2.728	W/m^2V^2
ΔB_T[3]	3.51	7.66	W/m^2

[1] Wide Field-of-View Total channel

[2] Medium Field-of-View Total channel

[3] $\Delta B_T = \sqrt{\sum (\text{Variables})^2}$

Table 10. Distribution of Surface and Volume Elements for the end-to-end model.

	Azimuthal Divisions	Horizontal Divisions	Radial Divisions	Number of Elements
FOVL[1] Aperture	0	-	1	1
FOVL[1]	10	4	-	40
Substrate	10	-	4	40
Ring	10	1	-	10
Dome	10	6	-	60
Primary Aperture	10	-	3	30
Cavity Cylinder	10	4	-	40
Cavity Cone	10	6	-	60

[1] Field-of-View Limiter

Table 11. Wide field-of-view total (WFOVT) channel ground calibration data [35].

Run	Flux (W/m^2)	T_{source} (K)	T_{hs} (K)	T_{fovl} (K)	T_{pa} (K)	Bias Voltage (V)
1	116.94	213.11	306.82	291.29	306.83	0
2	116.94	213.11	306.82	291.32	306.82	4.33
3	197.75	243.02	306.82	292.00	306.84	0
4	197.72	243.01	306.82	292.06	306.83	4.33
5	197.75	243.02	306.82	292.10	306.82	6.14
6	249.07	257.45	306.82	292.52	306.85	0
7	249.11	257.46	306.82	292.58	306.84	4.33
8	315.43	273.11	306.82	292.80	306.85	0
9	315.43	273.11	306.82	292.92	306.84	4.33
10	315.43	273.11	306.82	293.03	306.84	6.14
11	364.64	283.19	306.82	292.91	306.86	0
12	364.64	283.19	306.82	293.17	306.84	4.33
13	418.08	293.04	306.82	293.46	306.82	0
14	418.08	293.04	306.82	293.70	306.85	4.33

Table 12. Medium field-of-view total (MFOVT) channel ground calibration data [35].

Run	Flux (W/m^2)	T_{source} (K)	T_{hs} (K)	T_{fovl} (K)	T_{pa} (K)	Bias Voltage (V)
1	116.94	213.11	306.82	291.15	306.69	0
2	116.94	213.11	306.82	291.30	306.68	4.33
3	197.72	243.01	306.82	291.93	306.70	0
4	197.72	243.01	306.82	292.01	306.69	4.33
5	197.75	243.02	306.82	292.09	306.70	6.14
6	249.11	257.46	306.82	292.43	306.71	0
7	249.11	257.46	306.82	292.49	306.70	4.33
8	315.48	273.12	306.82	292.03	306.70	0
9	315.48	273.12	306.82	292.29	306.70	4.33
10	315.43	273.11	306.82	292.46	306.70	6.14
11	364.64	283.19	306.82	2192.86	306.72	0
12	364.64	283.19	306.82	293.09	306.71	4.33
13	418.19	293.06	306.82	292.90	306.72	0
14	418.19	293.06	306.82	293.13	306.72	4.33
15	418.14	293.05	306.82	293.28	306.72	6.14

Table 13. Wide field-of-view shortwave (WFOVSW) channel ground calibration data [35].

Run	Flux (W/m^2)	T_{source} (K)	T_{hs} (K)	T_{fovl} (K)	T_{pa} (K)	Bias Voltage (V)
1	111.999	5780	306.82	292.06	306.84	4.33
2	165.157	5780	306.82	293.70	306.87	6.13
3	164.780	5780	306.82	293.62	306.87	4.33
4	163.969	5780	306.82	293.54	306.88	0
5	218.955	5780	306.82	293.72	306.87	4.33
6	218.267	5780	306.82	293.58	306.89	0
7	247.334	5780	306.82	294.04	306.88	6.13
8	247.097	5780	306.82	294.03	306.88	4.33
9	245.947	5780	306.82	293.89	306.89	0
10	111.999	211	306.82	292.06	306.84	4.33
11	165.157	232	306.82	293.70	306.87	6.13
12	164.780	232	306.82	293.62	306.87	4.33
13	163.969	232	306.82	293.54	306.88	0
14	218.955	249	306.82	293.72	306.87	4.33
15	218.267	249	306.82	293.58	306.89	0
16	247.334	257	306.82	294.04	306.88	6.13
17	247.097	257	306.82	294.03	306.88	4.33
18	245.947	257	306.82	293.89	306.89	0

Table 14. Medium field-of-view shortwave (MFOVSW) channel ground calibration data [35].

Run	Flux (W/m^2)	T_{source} (K)	T_{hs} (K)	T_{fovl} (K)	T_{pa} (K)	Bias Voltage (V)
1	122.843	5780	306.82	293.72	306.70	4.33
2	122.467	5780	306.82	293.61	306.71	0
3	181.488	5780	306.82	293.65	306.70	6.13
4	181.450	5780	306.82	293.42	306.69	4.33
5	180.895	5780	306.82	293.09	306.70	0
6	242.800	5780	306.82	293.88	306.71	4.33
7	242.251	5780	306.82	293.73	306.72	0
8	275.914	5780	306.82	294.18	306.71	6.13
9	275.778	5780	306.82	294.07	306.71	4.33
10	274.895	5780	306.82	293.95	306.72	0
11	122.843	216	306.82	293.72	306.70	4.33
12	122.467	216	306.82	293.61	306.71	0
13	181.488	238	306.82	293.65	306.70	6.13
14	181.450	238	306.82	293.42	306.69	4.33
15	180.895	238	306.82	293.09	306.70	0
16	242.800	256	306.82	293.88	306.71	4.33
17	242.251	256	306.82	293.73	306.72	0
18	275.914	264	306.82	294.18	306.71	6.13
19	275.778	264	306.82	294.07	306.71	4.33
20	274.895	264	306.82	293.95	306.72	0

Table 15. Cavity thicknesses used in end-to-end model simulations.

Channel	Nominal Thickness (mm)	Thickness Used (mm)	Difference (mm)	Difference as Percent of Tolerance[1]
WFOVT[2]	0.0635	0.05364	0.00986	38.82
MFOVT[3]	0.0635	0.04985	0.01365	53.74
WFOVSW[4]	0.0635	0.05725	0.00625	24.61
MFOVSW[5]	0.0635	0.06585	0.00235	9.25

[1] Difference as percent of tolerance $= \frac{\text{Difference}}{\text{Tolerance}} \times 100$

Tolerance = 0.001mm

[2] Wide Field-of-View Total channel

[3] Medium Field-of-View Total channel

[4] Wide Field-of-View Shortwave channel

[5] Medium Field-of-View Shortwave channel

Table 16. Wide field-of-view total (WFOVT) channel voltage outputs during calibration runs for the actual instrument [35] and the end-to-end model. (See also Table 11)

Run	Radiometric Source[1]	Instrument output (V)	Model output (V)	Difference (Inst - Model) (V)	Percent Difference[2]
1	MRBB	7.425224	7.429431	-0.004207	-0.0567
2	MRBB	8.753511	8.757126	-0.003615	-0.0413
3	MRBB	7.211574	7.213815	-0.002241	-0.0311
4	MRBB	8.572823	8.575021	-0.002197	-0.0256
5	MRBB	9.757051	9.758428	-0.001377	-0.0141
6	MRBB	7.072398	7.073414	-0.001016	-0.0143
7	MRBB	8.455622	8.457084	-0.001462	-0.0173
8	MRBB	6.88927	6.887942	0.001328	0.0193
9	MRBB	8.304239	8.302632	0.001606	0.0193
10	MRBB	9.521427	9.52005	0.001377	0.0145
11	MRBB	6.751313	6.747151	0.004162	0.0616
12	MRBB	8.189477	8.186142	0.003335	0.0407
13	MRBB	6.596264	6.590529	0.005736	0.0870
14	MRBB	8.061287	8.057542	0.003745	0.0465

[1] MRBB = Master Reference Black Body

[2] $$\text{percent difference} = \frac{\text{Instrument} - \text{Model}}{\text{Instrument}} \times 100$$

Average percent difference = 0.00632
Standard deviation = 0.0419

Table 17. Medium field-of-view total (MFOVT) channel voltage outputs during calibration runs for the actual instrument [35] and the end-to-end model. (See also Table 12)

Run	Radiometric Source[1]	Instrument Output (V)	Model Output (V)	Difference (Inst - Model) (V)	Percent Difference[2]
1	MRBB	6.657307	6.670171	-0.012864	-0.1932
2	MRBB	8.066171	8.075988	-0.009818	-0.1217
3	MRBB	6.540106	6.54761	-0.007504	-0.1147
4	MRBB	7.969724	7.97512	-0.005396	-0.0677
5	MRBB	9.188134	9.192275	-0.004141	-0.0451
6	MRBB	6.464413	6.468376	-0.003963	-0.0613
7	MRBB	7.90746	7.910227	-0.002767	-0.0350
8	MRBB	6.370407	6.366627	0.004142	0.0650
9	MRBB	7.830547	7.826951	0.003596	0.0459
10	MRBB	9.067269	9.063749	0.00352	0.0388
11	MRBB	6.294714	6.287961	0.006753	0.1073
12	MRBB	7.769504	7.763173	0.006331	0.0815
13	MRBB	6.214138	6.202924	0.011214	0.1805
14	MRBB	7.704799	7.694426	0.010373	0.1346
15	MRBB	8.958613	8.949531	0.009082	0.1014

[1] MRBB = Master Reference Black Body

[2] percent difference $= \frac{\text{Instrument} - \text{Model}}{\text{Instrument}} \times 100$

Average percent difference = 0.0078
Standard deviation = 0.1079

Table 18. Wide field-of-view shortwave (WFOVSW) channel voltage outputs during calibration runs for the actual instrument [35] and the end-to-end model. (See also Table 13)

Run	Radiometric Source[1]	Instrument Output (V)	Model Output (V)	Difference (Inst - Model) (V)	Percent Difference[2]
1	ISP	6.082285	6.0806768	0.001608	0.0264
2	ISP	7.509462	7.5075147	0.001947	0.0259
3	ISP	8.718105	8.7163506	0.001754	0.0201
4	ISP	6.162862	6.1628310	0.000031	0.0005
5	ISP	7.575388	7.5753361	0.000052	0.0007
6	ISP	6.317910	6.3207678	-0.002858	-0.0452
7	ISP	7.701135	7.7040700	-0.002935	-0.0381
8	ISP	8.882920	8.8860036	-0.003084	-0.0347
9	ISP	7.826883	7.8270771	-0.000194	-0.0025
10	MRBB	-	6.7511842	-	-
11	MRBB	-	8.0634469	-	-
12	MRBB	-	9.1999704	-	-
13	MRBB	-	6.7540802	-	-
14	MRBB	-	8.0659551	-	-
15	MRBB	-	6.7594729	-	-
16	MRBB	-	8.0704198	-	-
17	MRBB	-	9.2060693	-	-
18	MRBB	-	8.0748572	-	-

[1] MRBB = Master Reference Black Body
ISP = Integrating Sphere

[2] $\text{percent difference} = \frac{\text{Instrument} - \text{Model}}{\text{Instrument}} \times 100$

Average percent difference = -0.0052
Standard deviation = 0.0279

Table 19. Medium field-of-view shortwave (MFOVSW) channel voltage outputs during calibration runs for the actual instrument [35] and the end-to-end model. (See also Table 14)

Run	Radiometric Source	Instrument Output (V)	Model Output (V)	Difference (Inst - Model) (V)	Percent Difference
1	ISP	6.911244	6.9064264	0.004818	0.0697
2	ISP	8.310341	8.3037823	0.006559	0.0079
3	ISP	9.514101	9.5085776	0.005523	0.0581
4	ISP	6.947870	6.9474149	0.000455	0.0065
5	ISP	8.338420	8.3382201	0.000200	0.0024
6	ISP	7.023562	7.0239572	-0.000395	-0.0056
7	ISP	8.401905	8.4020398	-0.000135	-0.0016
8	ISP	9.595898	9.5945255	0.004455	0.0464
9	ISP	7.091930	7.0954097	-0.003480	-0.0491
10	ISP	8.461726	8.4620933	-0.000367	-0.0043
11	MRBB	-	7.231473	-	-
12	MRBB	-	8.577088	-	-
13	MRBB	-	9.748290	-	-
14	MRBB	-	7.233369	-	-
15	MRBB	-	8.578452	-	-
16	MRBB	-	7.236463	-	-
17	MRBB	-	8.581001	-	-
18	MRBB	-	9.751672	-	-
19	MRBB	-	7.238604	-	-
20	MRBB	-	8.582913	-	-

Average percent difference = 0.0130
Standard deviation = 0.0354

Table 20. Comparison of instrument and end-to-end model gains.

Channel	Reference Cavity Heater Voltage (V)	Radiometric Source[1]	End-to-End Model Gain (Vm_2/W)	Instrument Gain (Vm^2/W)	Percent Difference[2]
WFOVT[3]	0	MRBB	-0.002786	-0.002753	1.185
	4.33	MRBB	-0.002323	-0.002299	0.103
	6.14	MRBB	-0.002026	-0.002003	1.135
MFOVT[4]	0	MRBB	-0.001551	-0.001471	5.158
	4.33	MRBB	-0.001267	-0.001200	5.288
	6.14	MRBB	-0.001101	-0.001041	5.450
WFOVSW[5]	0	ISP	-0.002929	-0.002874	1.878
	4.33	ISP	-0.002365	-0.002350	0.634
	6.14	ISP	-0.002065	-0.002006	2.857
	0	MRBB	-0.000101	-	-
	4.33	MRBB	-0.000084	-	-
	6.14	MRBB	-0.000074	-	-
MFOVSW[6]	0	ISP	-0.001240	-0.001185	4.435
	4.33	ISP	-0.001035	-0.00099	4.348
	6.14	ISP	-0.00091	-0.00086	5.495
	0	MRBB	-0.000047	-	-
	4.33	MRBB	-0.000038	-	-
	6.14	MRBB	-0.000036	-	-

[1] MRBB = Master Reference Black Body
ISP = Integrating Sphere

[2] $\text{percent difference} = \frac{\text{Model} - \text{Instrument}}{\text{Model}} \times 100$

[3] Wide Field-of-View Total channel

[4] Medium Field-of-View Total channel

[5] Wide Field-of-View Shortwave channel

[6] Medium Field-of-View Shortwave channel

Table 21. Ground count-conversion equation coefficients as determined using multiple regression methodology by the ERBE Science Team (EST), and by using the end-to-end model.

Channel	Coefficient	EST	End-to-End Model	Percent Difference[1]	Units
WFOVT[2]	A_V'	-26.2164	-25.6298	2.24	W/m^2V^2
	A_F'	-1.6070	-0.1224	92.38	W/m^2K
	A_R'	30.0461	29.36538	2.27	W/m^2V^2
	B_T'	1560.32	1567.33	-0.45	W/m^2
MFOVT[3]	A_V'	-53.5596	-50.2028	6.27	W/m^2V^2
	A_F'	-2.1768	-0.5349	75.43	W/m^2K
	A_R'	59.2633	55.44413	6.44	W/m^2V^2
	B_T'	2487.64	2506.689	-0.77	W/m^2
WFOVSW[4]	A_V'	-28.2194	-28.5472	-1.16	W/m^2V^2
	A_F'	-0.7097	0.7394	204.18	W/m^2K
	A_R'	29.2817	29.6613	-1.30	W/m^2V^2
	A_E'	-0.02902	-0.04121	-42.01	-
	B_{SW}'	1234.19	1093.73	11.38	W/m^2
MFOVSW[5]	A_V'	-58.454	-59.7392	-2.20	W/m^2V^2
	A_F'	2.7990	-1.51031	153.96	W/m^2K
	A_R'	66.4883	67.92095	-2.15	W/m^2V^2
	A_E'	-0.03641	-0.03151	13.46	-
	B_{SW}'	2984.17	3576.073	-19.83	W/m^2

[1] percent difference $= \dfrac{\text{EST} - \text{Model}}{\text{EST}} \times 100$

[2] Wide Field-of-View Total channel

[3] Medium Field-of-View Total channel

[4] Wide Field-of-View Shortwave channel

[5] Medium Field-of-View Shortwave channel

Table 22. Specified and predicted fluxes for the wide field-of-view total (WFOVT) channel calibration runs. (See also Tables 11 and 16)

Run	Specified Flux (W/m^2)	Predicted Flux (W/m^2)	Difference (Pred - Spec)	Percent Difference[1]
1	116.9433	116.9991	0.055775	0.048
2	116.9433	116.7585	-0.1848	-0.158
3	197.7553	197.8338	0.078464	0.040
4	197.7228	197.5628	-0.16001	-0.081
5	197.7553	197.9896	023426	0.118
6	249.0759	249.1821	0.106192	0.043
7	249.1146	248.9822	-0.13239	-0.053
8	315.4354	315.5144	0.079031	0.025
9	315.4354	315.2848	-0.15057	-0.048
10	315.4354	315.6591	0.223706	0.071
11	364.6463	364.7025	0.056153	0.015
12	364.6463	364.4835	-0.16283	-0.045
13	418.0881	418.1754	0.087281	0.021
14	418.0881	417.9578	-0.13027	-0.031

[1] percent difference $= \frac{\text{Predicted} - \text{Specified}}{\text{Predicted}} \times 100$

Average percent difference = -0.0025
Standard deviation = 0.071

Table 23. Specified and predicted fluxes for the medium field-of-view total (MFOVT) channel calibration runs. (See also Tables 12 and 17)

Run	Specified Flux (W/m^2)	Predicted Flux (W/m^2)	Difference (Pred - Spec)	Percent Difference[1]
1	116.9433	117.3689	0.425642	0.364
2	116.9433	116.0798	-0.86347	-0.738
3	197.7228	198.2796	0.556773	0.282
4	197.7228	196.9811	-0.74173	-0.375
5	197.7553	198.6354	0.880057	0.445
6	249.1146	249.7866	0.672017	0.270
7	249.1146	248.4757	-0.63889	-0.256
8	315.4816	315.7945	0.312917	0.099
9	315.4816	314.3749	-1.10672	-0.351
10	315.4354	316.2325	0.797143	0.253
11	364.6463	365.0683	0.422046	0.116
12	364.6463	363.8638	-0.78246	-0.215
13	418.2022	418.3986	0.196354	0.047
14	418.2022	417.1915	-1.0107	-0.242
15	418.2022	419.0832	0.881014	0.211

[1] $\text{percent difference} = \frac{\text{Predicted} - \text{Specified}}{\text{Predicted}} \times 100$

Average percent difference = -0.006
Standard deviation = 0.3376

Table 24. Specified and predicted fluxes for the wide field-of-view shortwave (WFOVSW) channel calibration runs. (See also Tables 13 and 18)

Run	Specified LW Flux (W/m^2)	Specified SW Flux (W/m^2)	Predicted Flux (W/m^2)	Difference (Pred - Spec)	Percent Difference[1]
1	0	245.9479	245.3763	-0.57121	-0.232
2	0	247.097	248.0715	0.974496	0.394
3	0	247.334	246.6638	-0.67015	-0.271
4	0	218.2678	217.5738	-0.69398	-0.318
5	0	218.9558	219.7999	0.844129	0.386
6	0	163.9693	163.4977	-0.4716	-0.288
7	0	164.7808	165.8069	1.02607	0.623
8	0	165.1577	164.5487	-0.60897	-0.369
9	0	111.999	112.2908	0.291838	0.261
10	245.9479	0	-0.23976	-0.23976	-
11	247.097	0	0.955148	0.955148	-
12	247.334	0	-0.6891	-0.6891	-
13	218.2678	0	-0.44472	-0.44472	-
14	218.9558	0	0.730793	0.730793	-
15	163.9693	0	-0.3169	-0.3169	-
16	164.7808	0	0.832881	0.832881	-
17	165.1577	0	-0.75844	-0.75844	-
18	111.999	0	-0.19053	-0.19053	-

[1] $$\text{percent difference} = \frac{\text{Predicted} - \text{Specified}}{\text{Predicted}} \text{ X } 100$$

Average percent difference = 0.0207
Standard deviation = 0.3880

Table 25. Specified and predicted fluxes for the medium field-of-view shortwave channel calibration runs. (See also Tables 14 and 19)

Run	Specified LW Flux (W/m^2)	Specified SW Flux (W/m^2)	Predicted Flux (W/m^2)	Difference (Pred - Spec)	Percent Difference[1]
1	0	274.8957	273.9739	-0.92181	-0.335
2	0	275.7781	277.5075	1.729419	0.627
3	0	275.9143	274.1356	-1.77865	-0.645
4	0	242.2518	241.4119	-0.83986	-0.347
5	0	242.8004	244.5962	1.795792	0.740
6	0	180.8951	180.4267	-0.46844	-0.259
7	0	181.4509	183.4011	1.95017	1.063
8	0	181.4882	179.8272	-1.66096	-0.915
9	0	122.4678	121.2135	-1.25426	-1.024
10	0	122.8437	124.2937	1.450042	1.180
11	274.8957	0	-0.55559	-0.55559	-
12	275.7781	0	1.892456	1.892456	-
13	275.9143	0	-1.62673	-1.62673	-
14	242.2518	0	-0.83313	-0.83313	-
15	242.8004	0	1.820578	1.820578	-
16	180.8951	0	-0.60842	-0.60842	-
17	181.4509	0	1.835446	1.835446	-
18	181.4882	0	-1.79024	-1.79024	-
19	122.4678	0	-1.404	-1.404	-
20	122.8437	0	1.268186	1.268186	-

Average percent difference = 0.0085
Standard deviation = 0.8208

Table 26. Comparison of the Inflight count-conversion equation coefficients as determined by the ERBE Science Team (EST), the end-to-end model, and the low-order physical model.

Channel	Coefficient	EST	End-to-End Model	Physical Model	Units
WFOVT[1]	A_V	-22.7873	-22.277	-22.5212	W/m^2V^2
	A_F	-1.3968	-0.1064	-0.2390	W/m^2K
	A_R	26.1161	25.5244	-	W/m^2V^2
MFOVT[2]	A_V	-22.7093	-21.286	-22.6721	W/m^2V^2
	A_F	-0.9230	-0.2268	-2.1244	W/m^2K
	A_R	25.1276	23.5071	-	W/m^2V^2
WFOVSW[3]	A_V	-25.7758	-25.1129	-24.8229	W/m^2V^2
	A_F	-0.6502	0.6505	-0.0056	W/m^2K
	A_R	25.7591	26.0930	-	W/m^2V^2
	A_E	-0.0255	-0.3625	-	-
MFOVSW[4]	A_V	-25.5630	-25.3294	-24.5551	W/m^2V^2
	A_F	1.2242	-0.64037	-0.05631	W/m^2K
	A_R	28.1910	28.7985	-	W/m^2V^2
	A_E	-0.0154	-0.01336	-	-

[1] Wide Field-of-View Total channel
[2] Medium Field-of-View Total channel
[3] Wide Field-of-View Shortwave channel
[4] Medium Field-of-View Shortwave channel

Table 27. Comparison of incident and predicted energy during a solar blip.

Channel	Specified Energy (KJ)	Predicted Energy (KJ)	Percent Difference[1]
WFOVT[2]	43.38	44.34	2.17
WFOVSW[3]	21.96	26.09	15.83

[1] percent difference $= \frac{\text{Predicted} - \text{Specified}}{\text{Predicted}} \text{ X } 100$

[2] Wide Field-of-View Total channel

[3] Wide Field-of-View Shortwave Channel

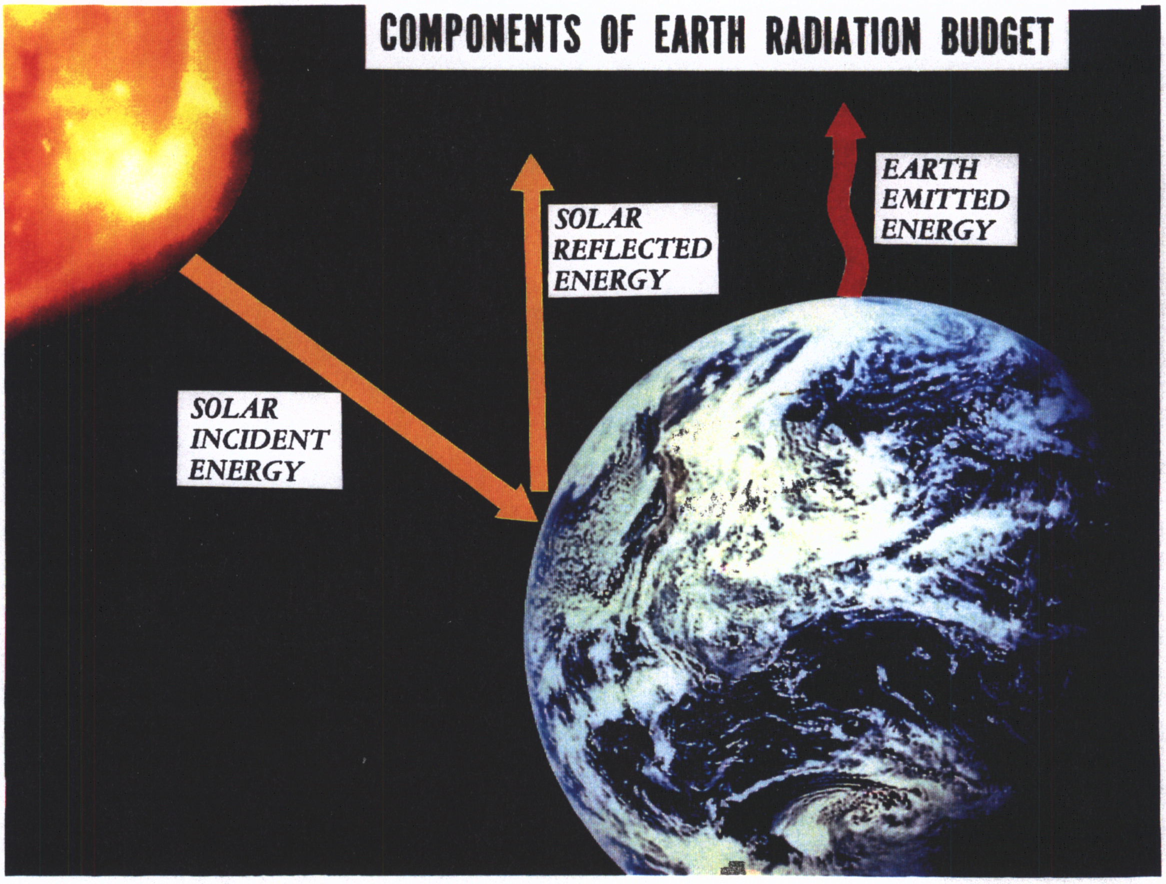

Fig. 1. Components of the Earth Radiation Budget.

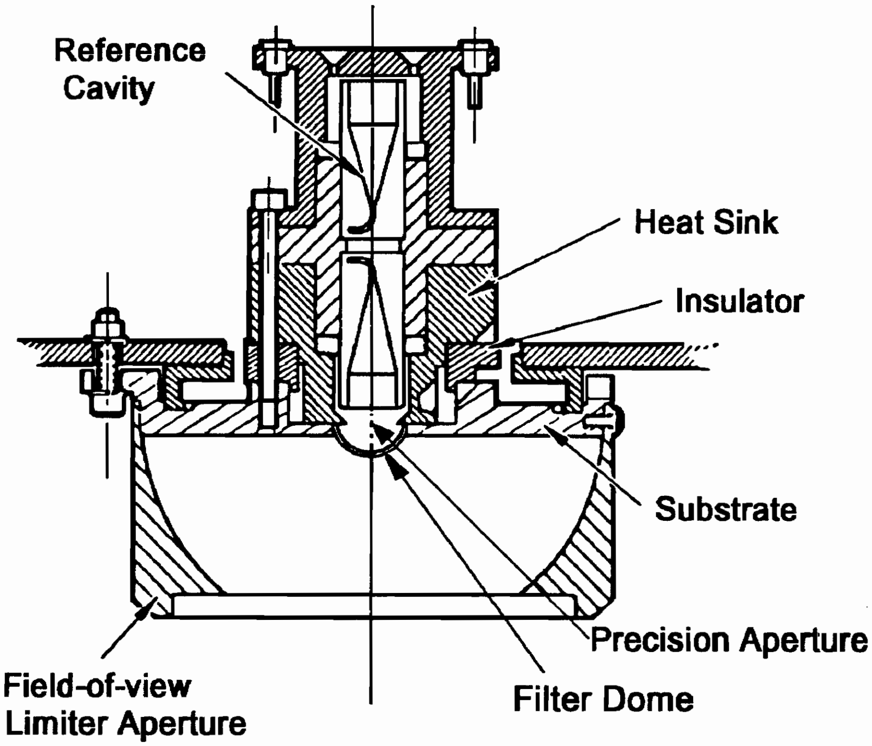

Fig. 2. An ERBE Earth-Viewing Nonscanning Active-Cavity Radiometer.

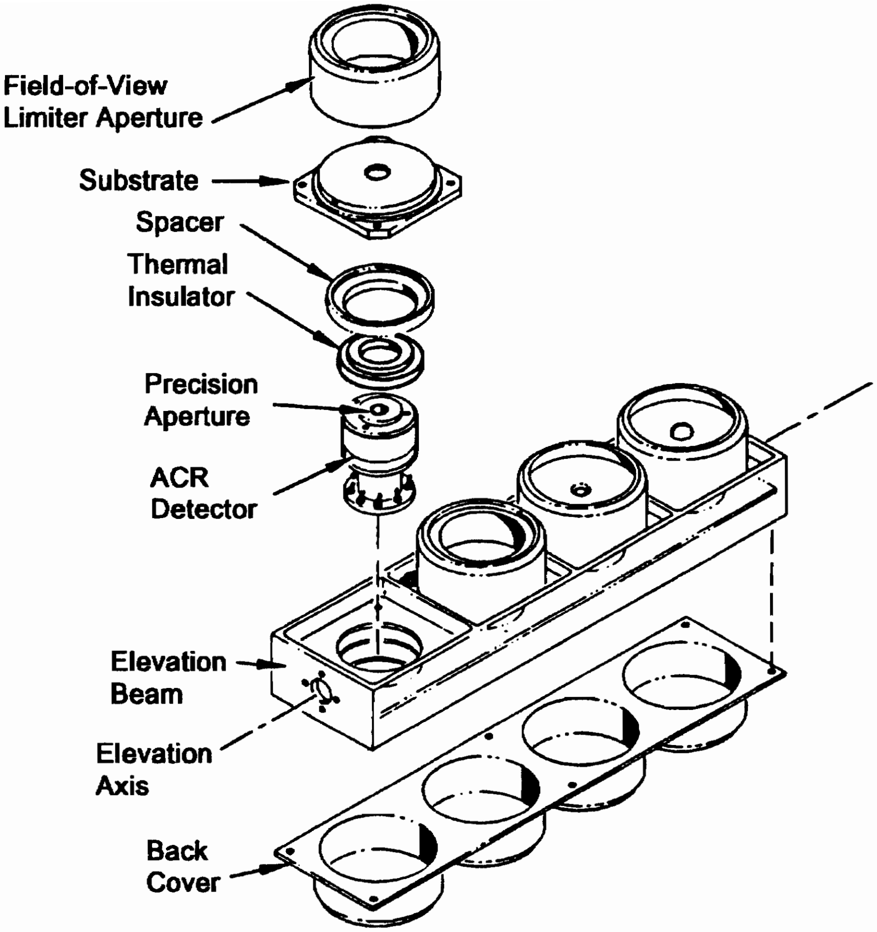

Fig. 3. Nonscanner Assembly Drawing.

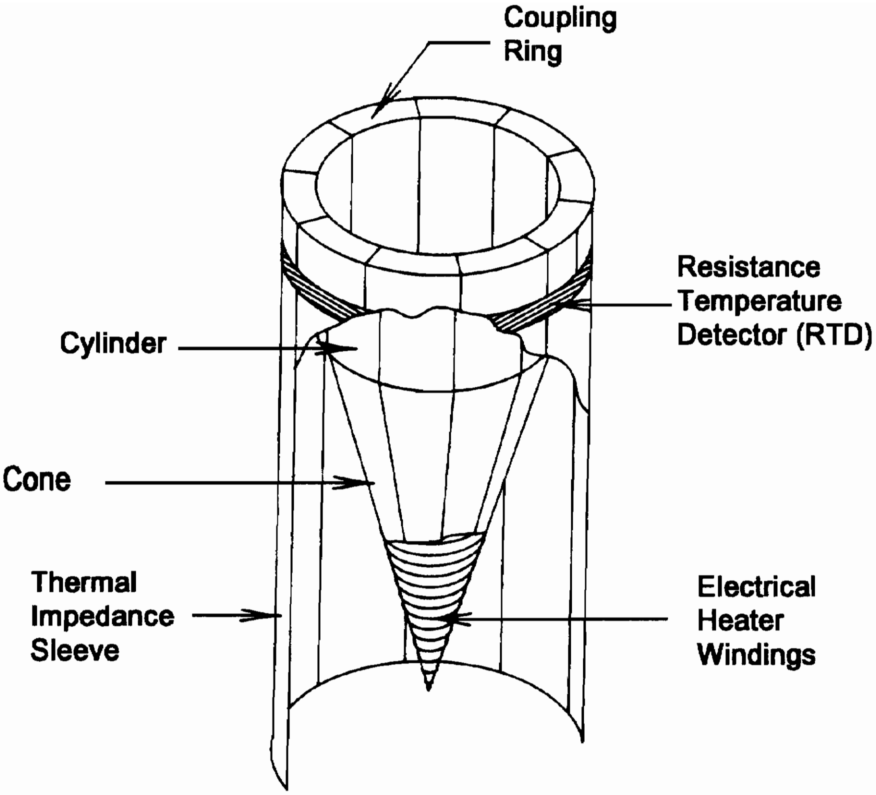

Fig. 4. Radiometer Sensing Element.

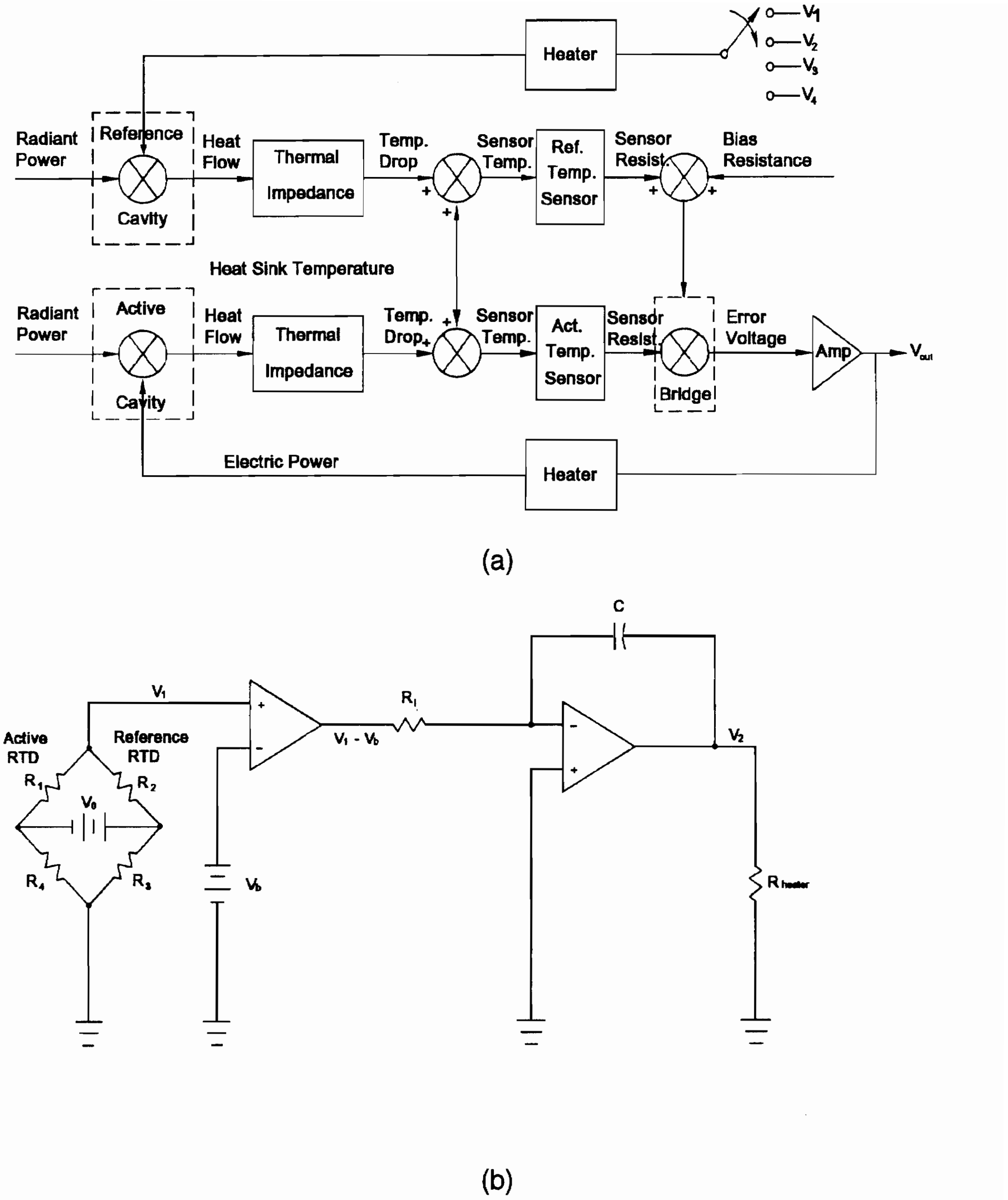

Fig. 5. Electrical Feedback Control (a) Block Diagram and (b) Simplified Electrical Schematic.

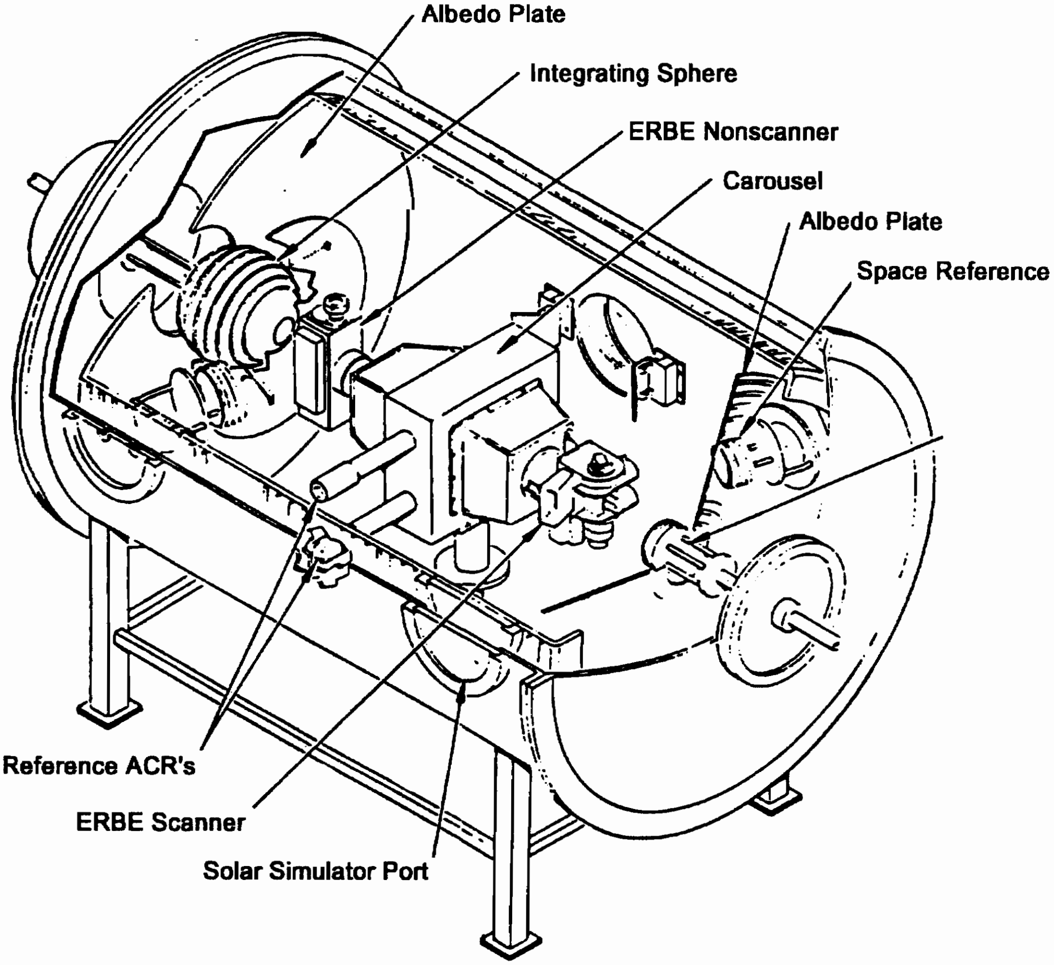

Fig. 6. ERBE Calibration Chamber [33].

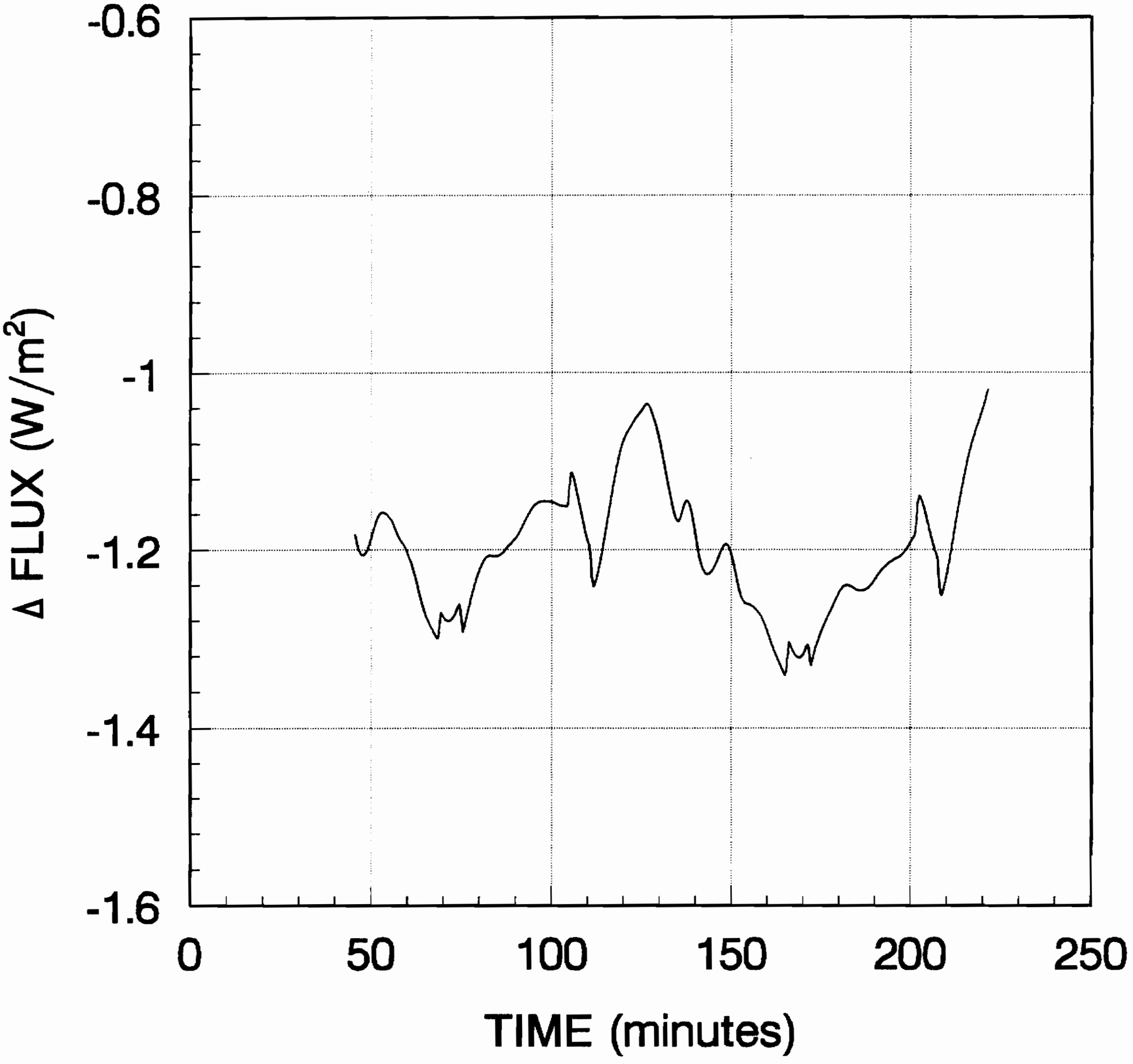

Fig. 7. Differences in the Earth Radiation Budget Satellite (ERBS) Wide Field-of-View Total (WFOVT) Channel Fluxes Obtained Utilizing Low-Order Physical Model Coefficients and Coefficients Based on the ERBE Science Team's Statistical Regression (Model - Regressed).

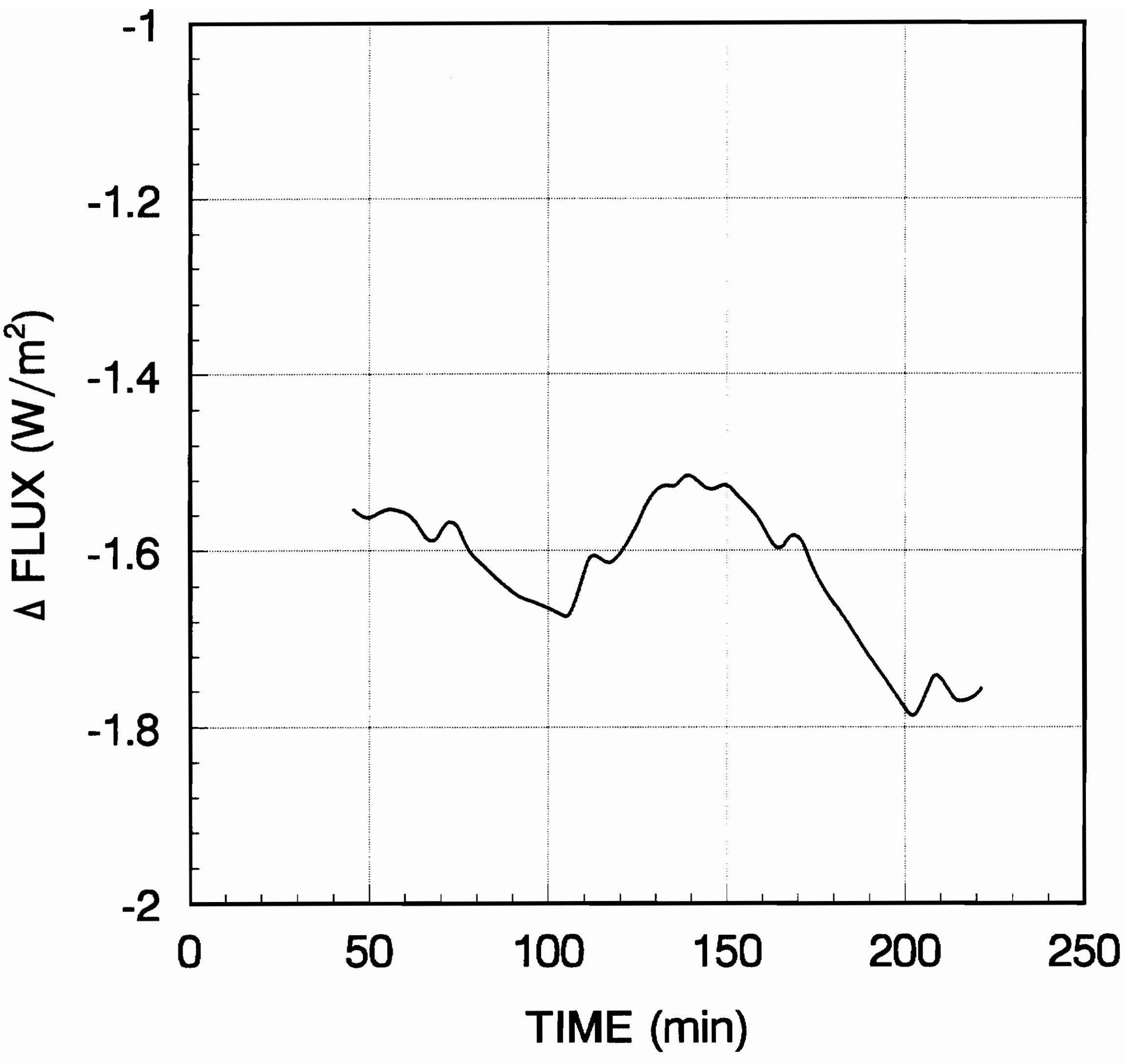

Fig. 8. Differences in the Earth Radiation Budget Satellite (ERBS) Medium Field-of-View Total (MFOVT) Channel Fluxes Obtained Utilizing Low-Order Physical Model Coefficients and Coeficients Based on the ERBE Science Team's Statistical Regression (Model - Regressed).

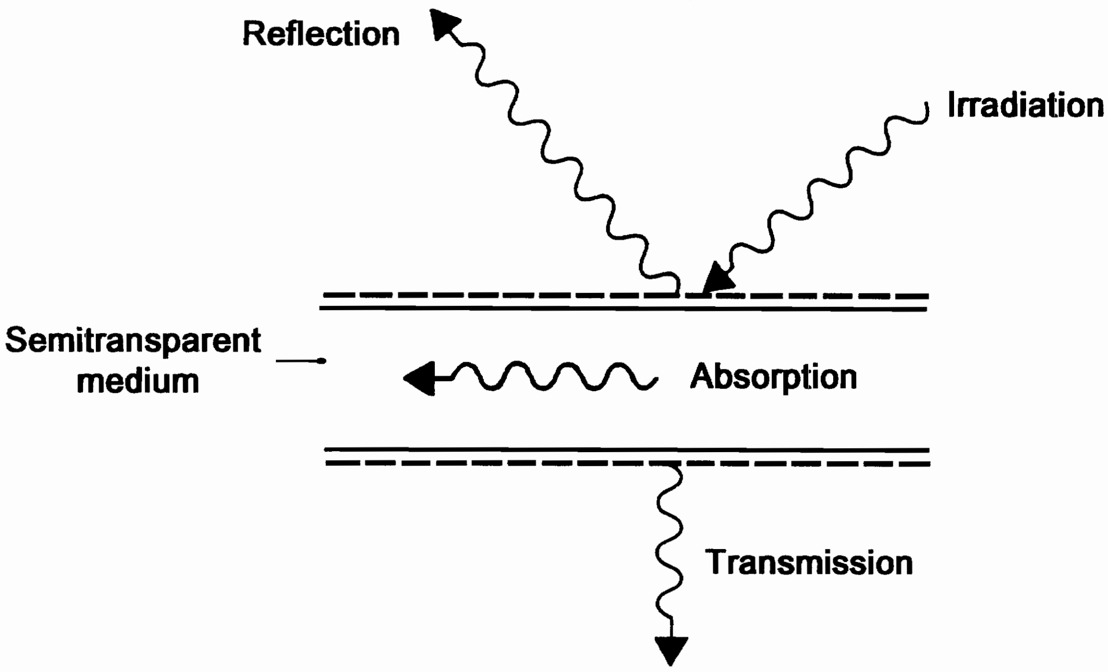

Fig. 9. Illustration of Absorption, Reflection and Transmission.

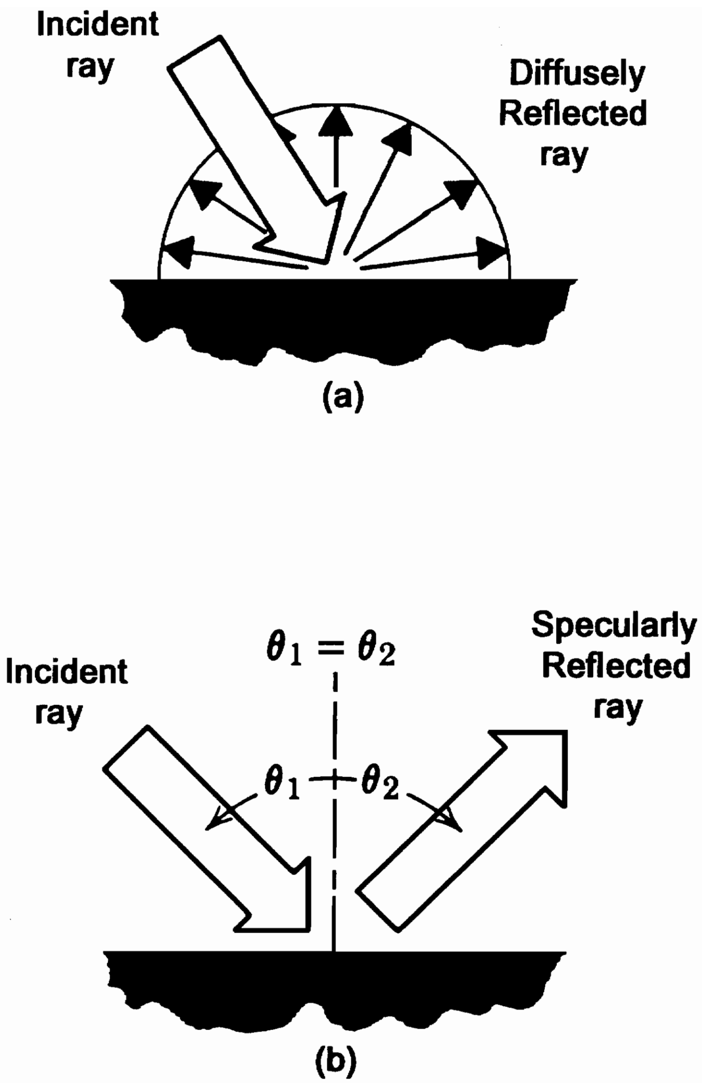

Fig. 10. Illustration of (a) Diffuse and (b) Specular Reflections.

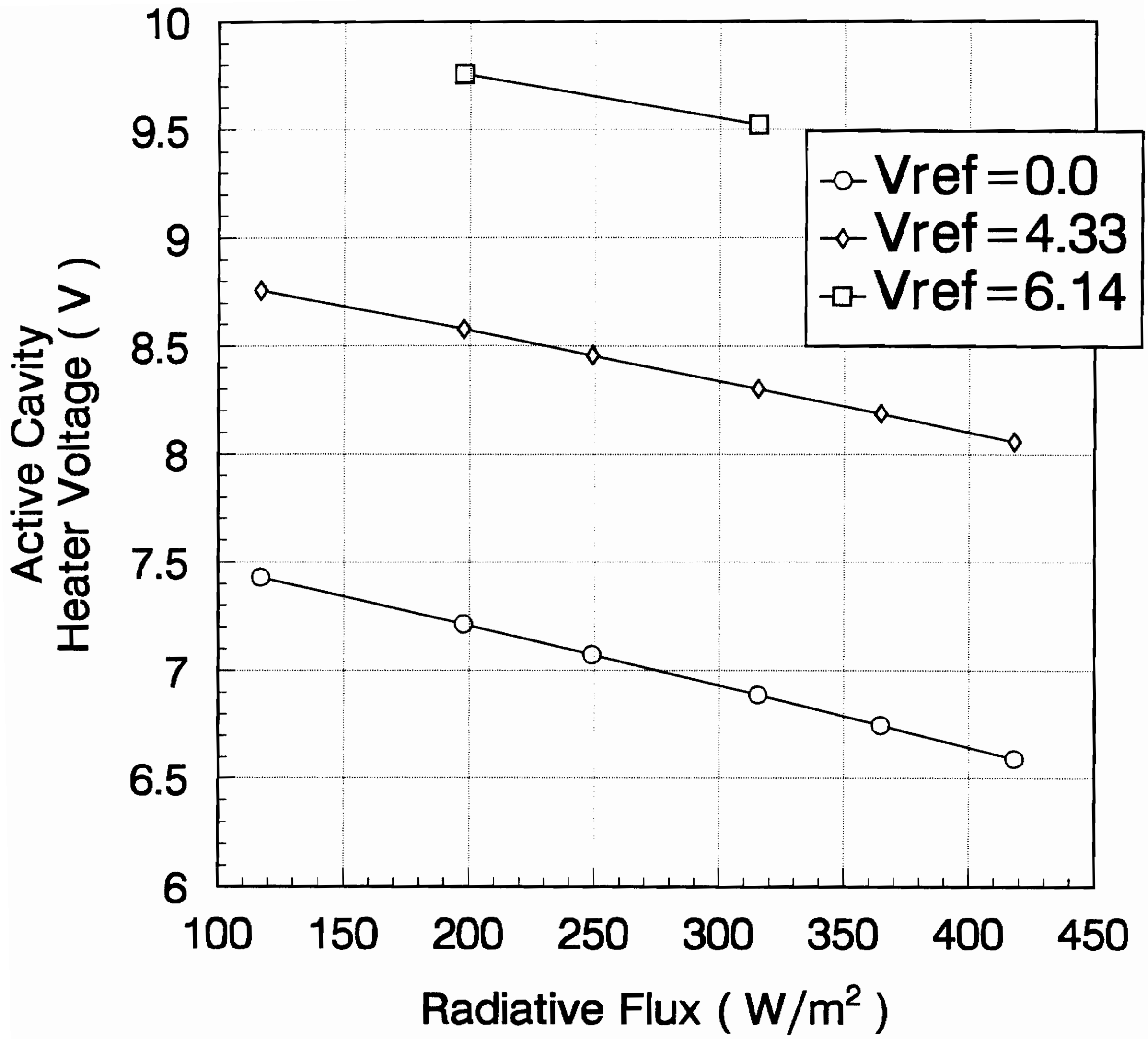

Fig. 11. Gains of the Wide Field-of-View (WFOVT) Channel End-to-End Model During Simulated Ground Calibration Runs as a Function of Reference Cavity Heater Voltage.

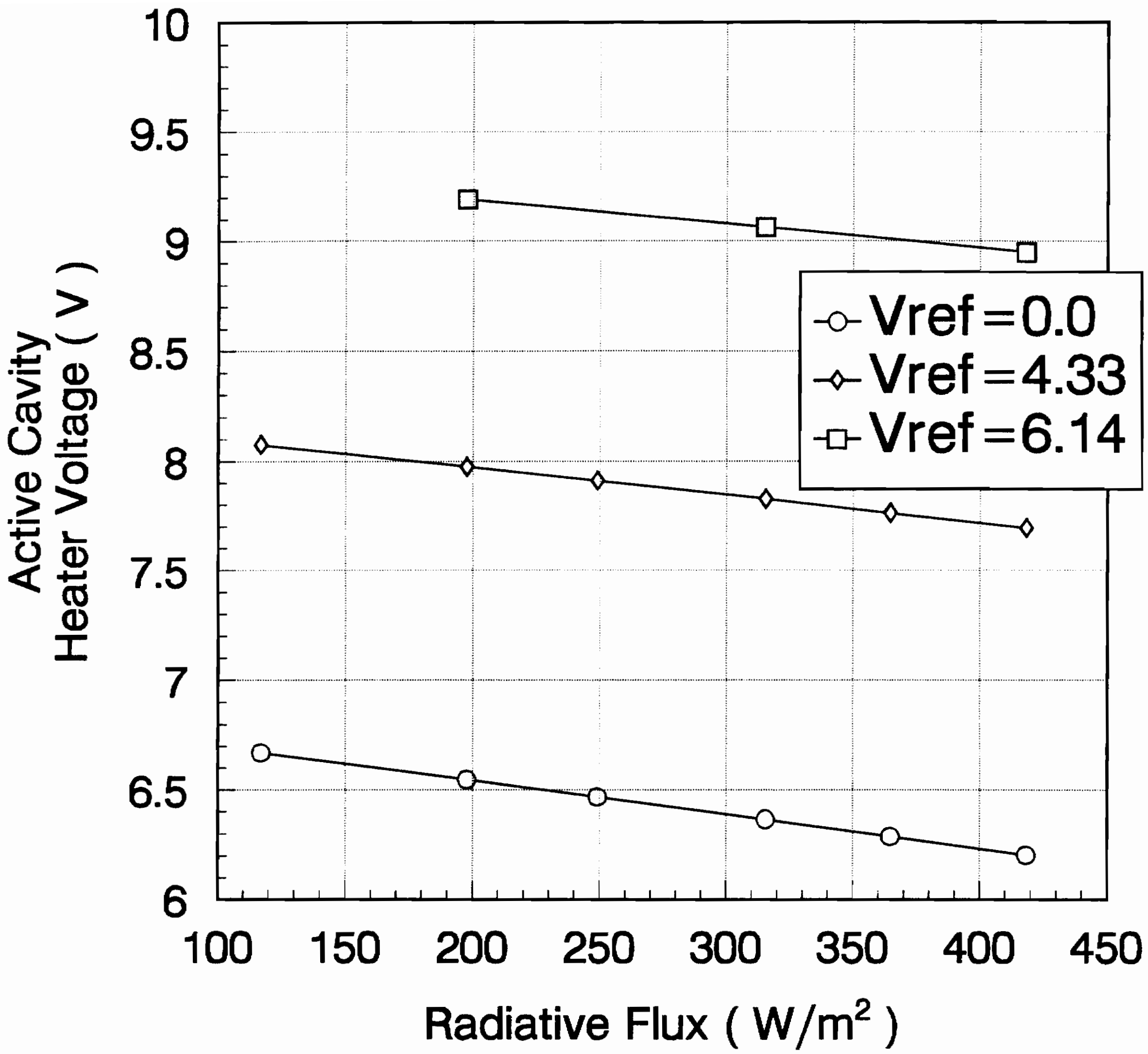

Fig. 12. Gains of the Medium Field-of-View Total (MFOVT) Channel End-to-End Model During Simulated Ground Calibration Runs as a Function of Reference Cavity Heater Voltage.

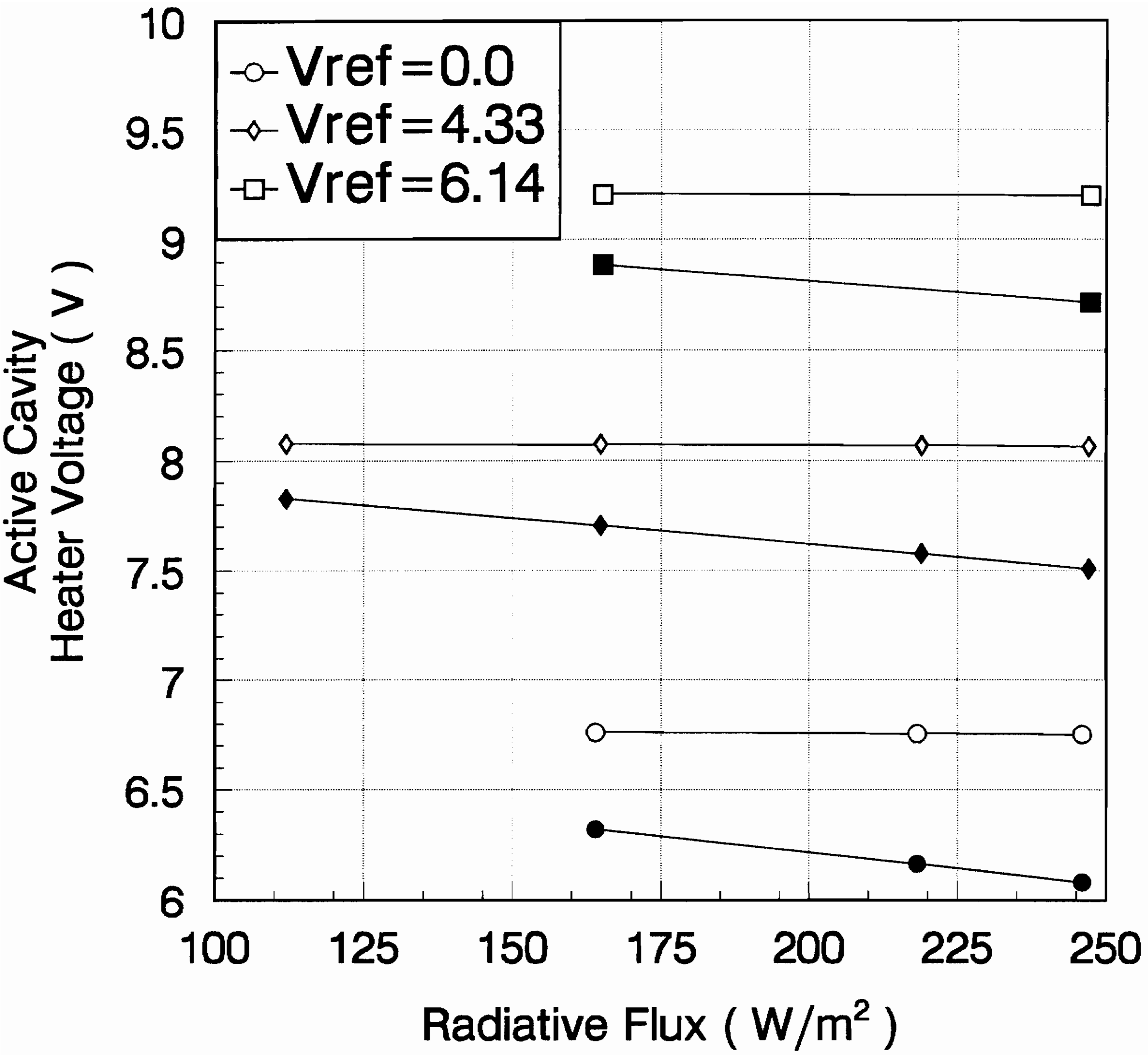

Fig. 13. Gains of the Wide Field-of-View Shortwave (WFOVSW) Channel End-to-End Model During Simulated Ground Calibration Runs as a Function of Reference Cavity Heater Voltage and Spectral Distribution of the Radiative Flux. (Solid Symbols = Shortwave Sources, Hollow Symbols = Longwave Sources)

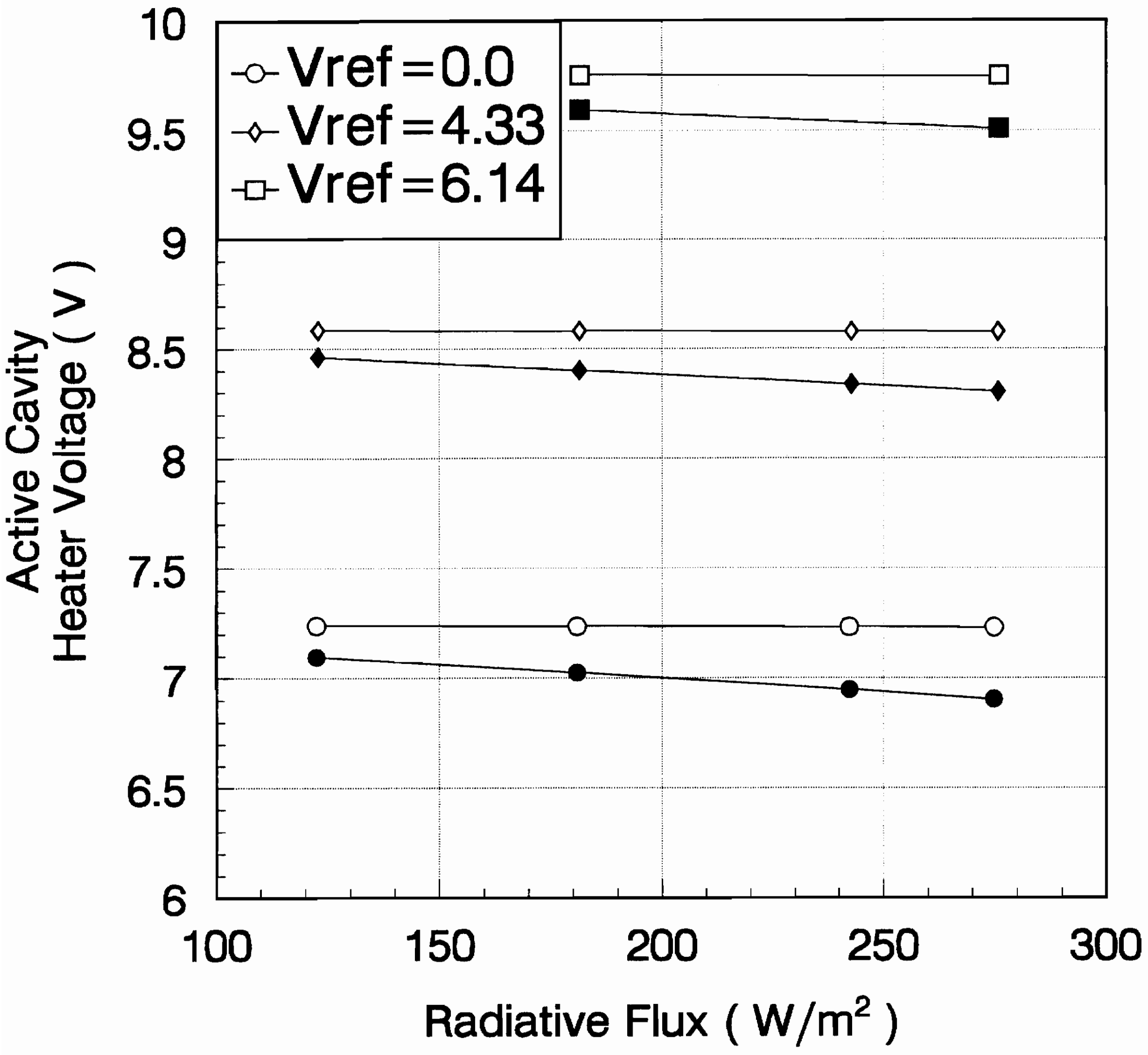

Fig. 14. Gains of the Medium Field-of-View Shortwave (MFOVSW) Channel End-to-End Model During Simulated Ground Calibration Runs as a Function of Reference Cavity Heater Voltage and Spectral Distribution of the Radiative Flux. (Solid Symbols = Shortwave Sources, Hollow Symbols = Longwave Sources)

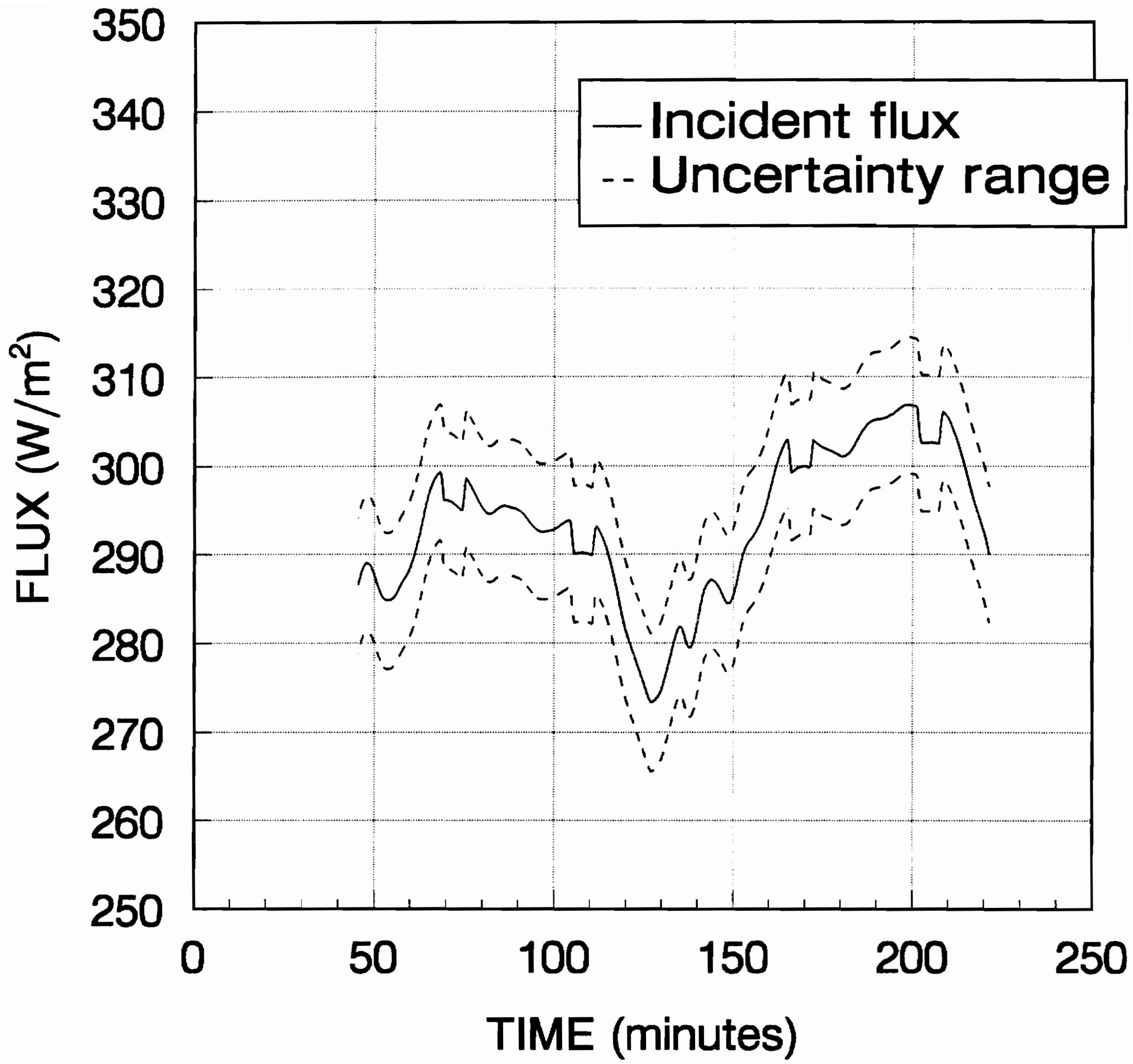

Fig. 15. Uncertainty in the Wide Field-of-View Total (WFOVT) Channel Data Product Derived Using a Low-Order Physical Model.

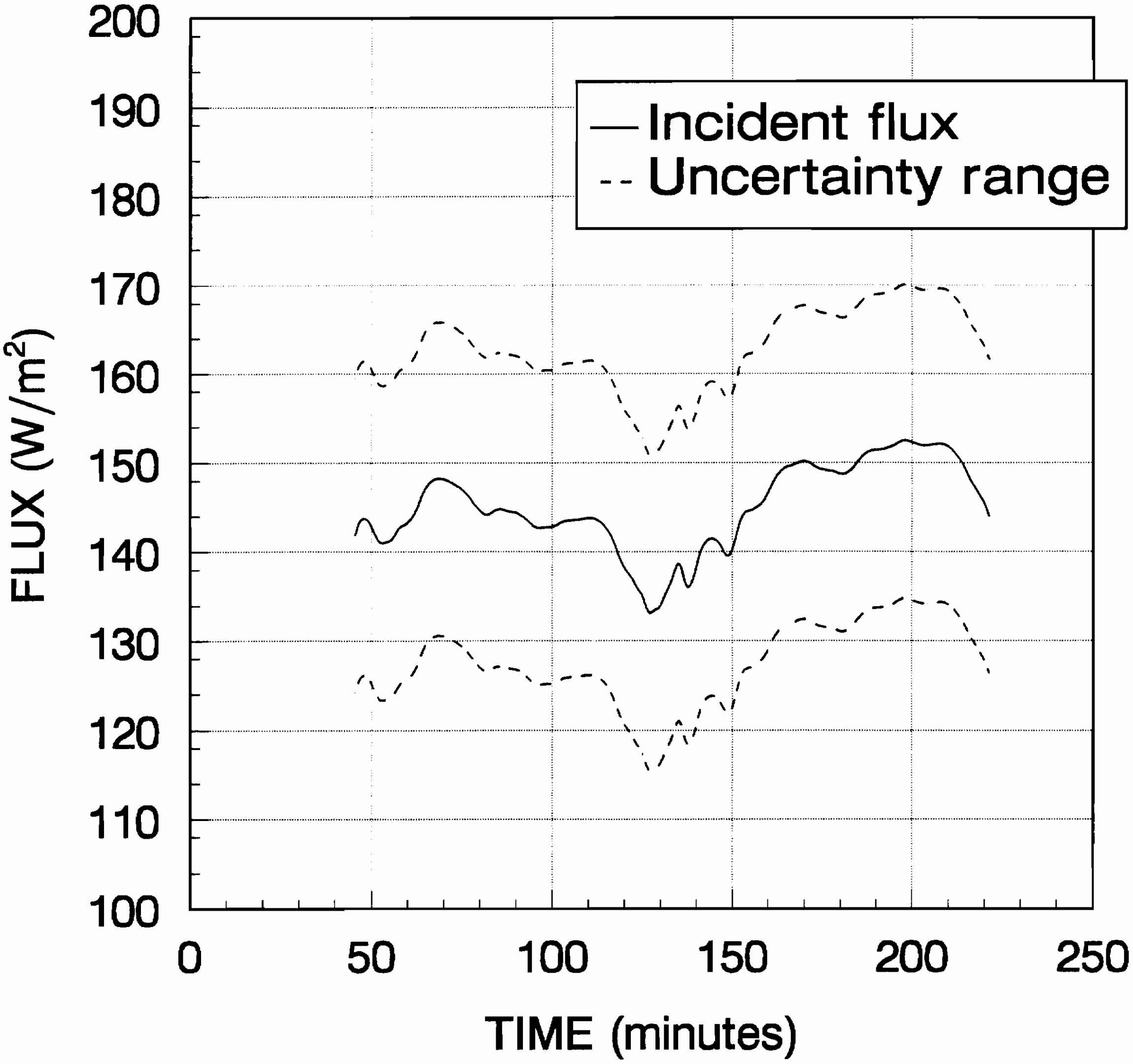

Fig. 16. Uncertainty in the Medium Field-of-View Total (MFOVT) Channel Data Product Derived Using a Low-Order Physical Model.

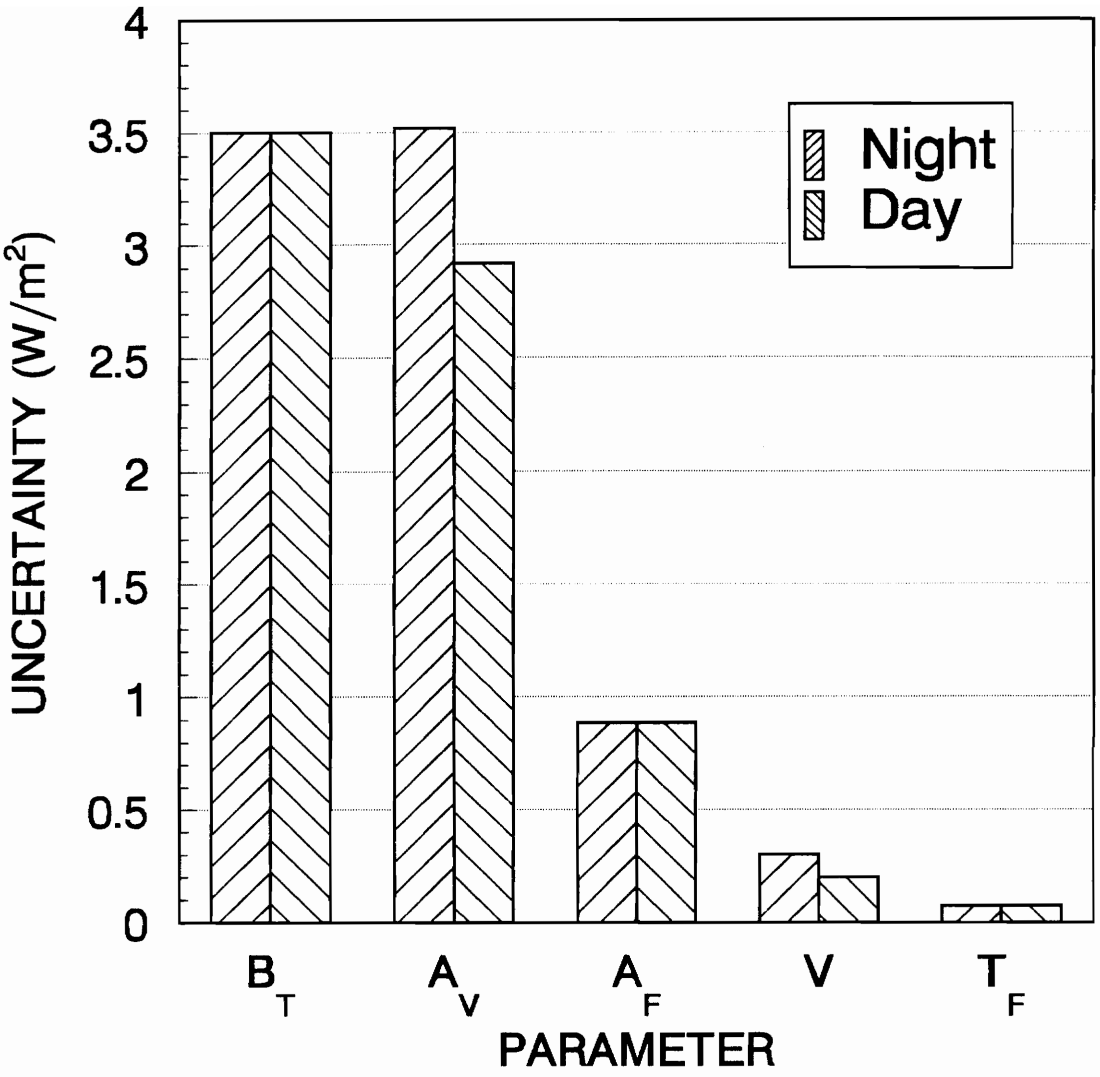

Fig. 17. Variation in the Wide Field-of-View Total (WFOVT) Channel Uncertainty as a Function of Solar Zenith Angle.

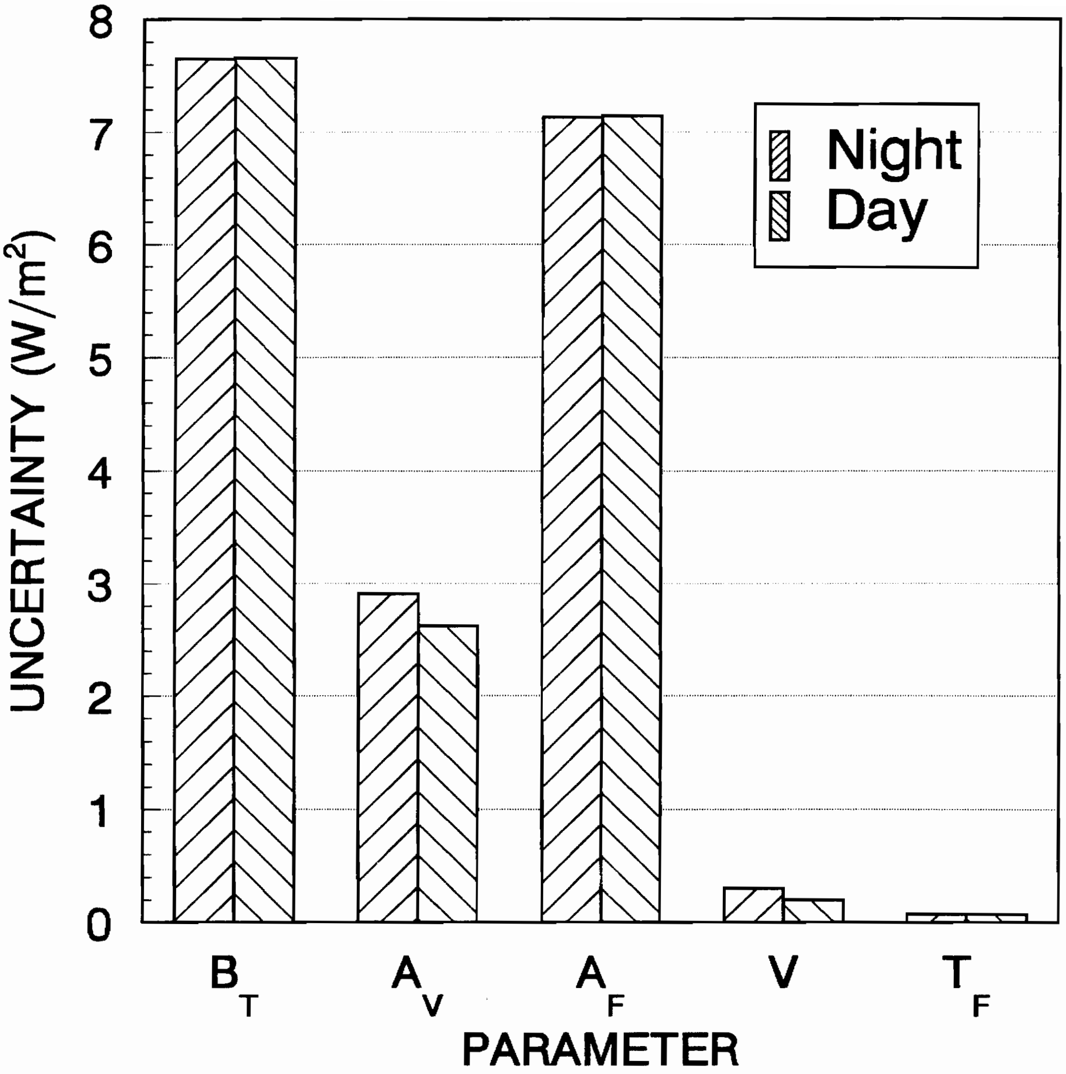

Fig. 18. Variation in the Medium Field-of-View Total (MFOVT) Channel Uncertainty as a Function of Solar Zenith Angle.

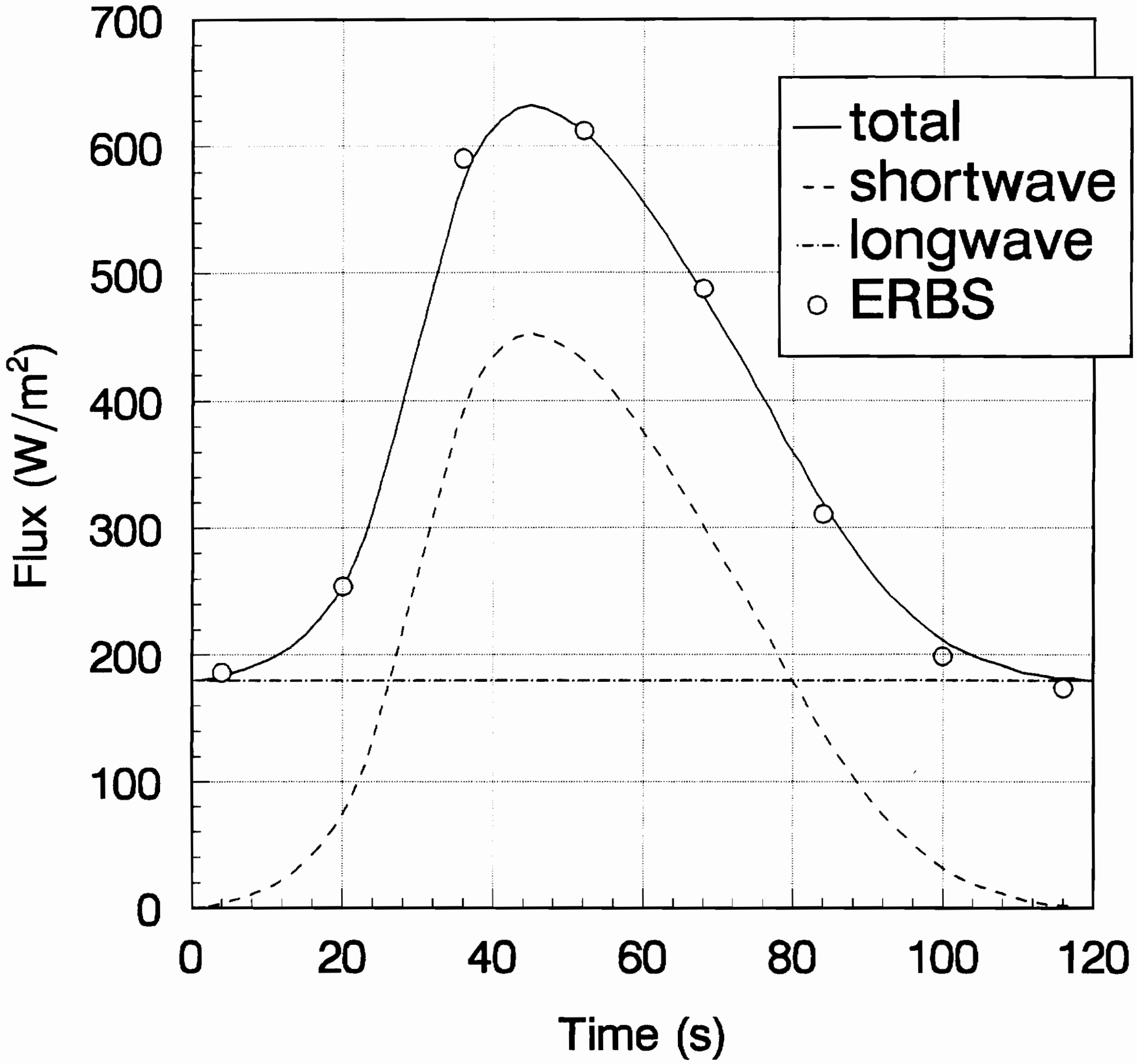

Fig. 19. Simulated Solar Blip Event Radiative Fluxes, Determined Using Data Recorded by the Earth Radiation Budget Satellite (ERBS) on April 17, 1985.

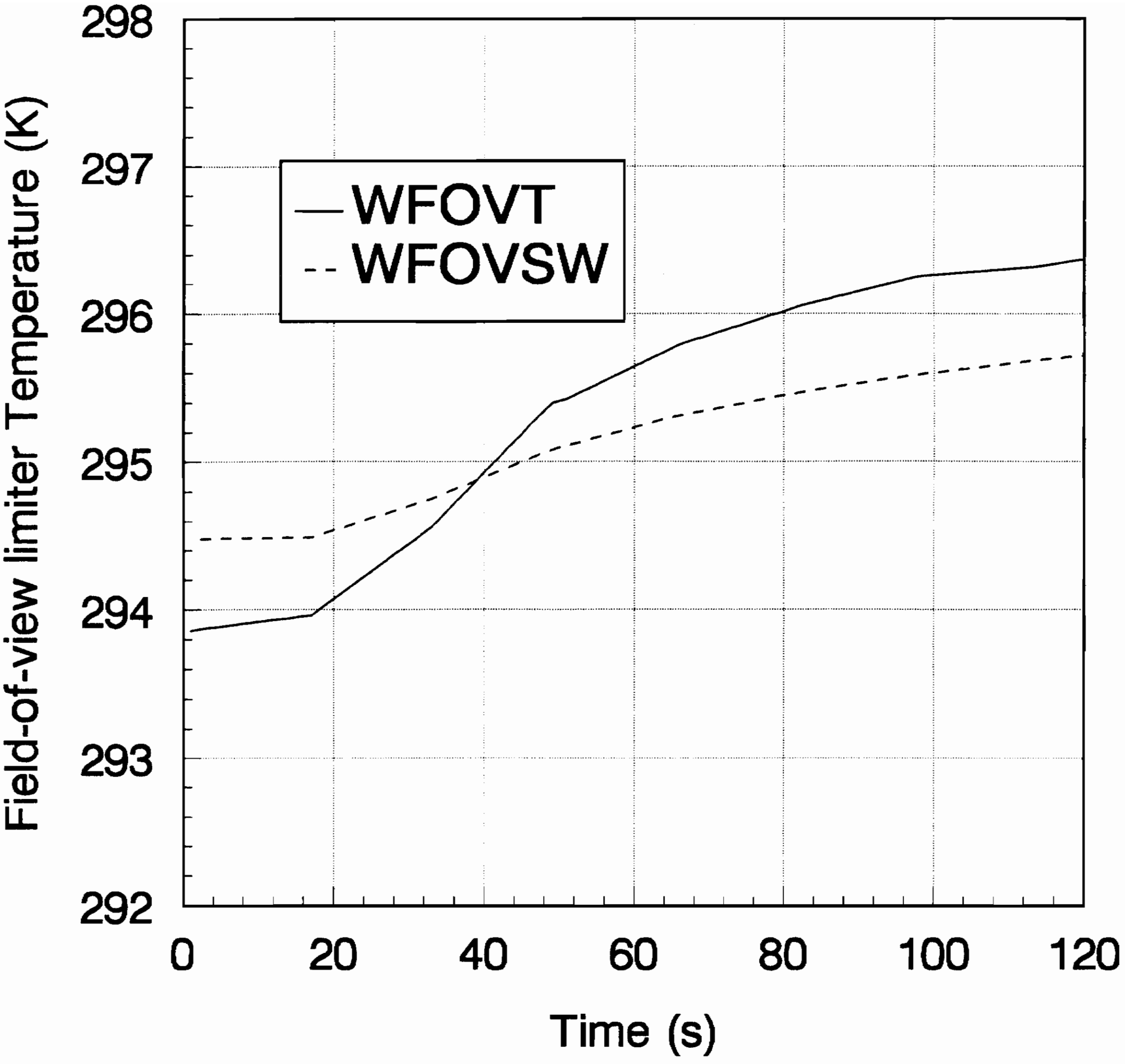

Fig. 20. Field-of-View Limiter Temperatures for the Simulated Solar Blip Event, Values were Obtained from Data Recorded by the Earth Radiation Budget Satellite (ERBS) on April 17, 1985.

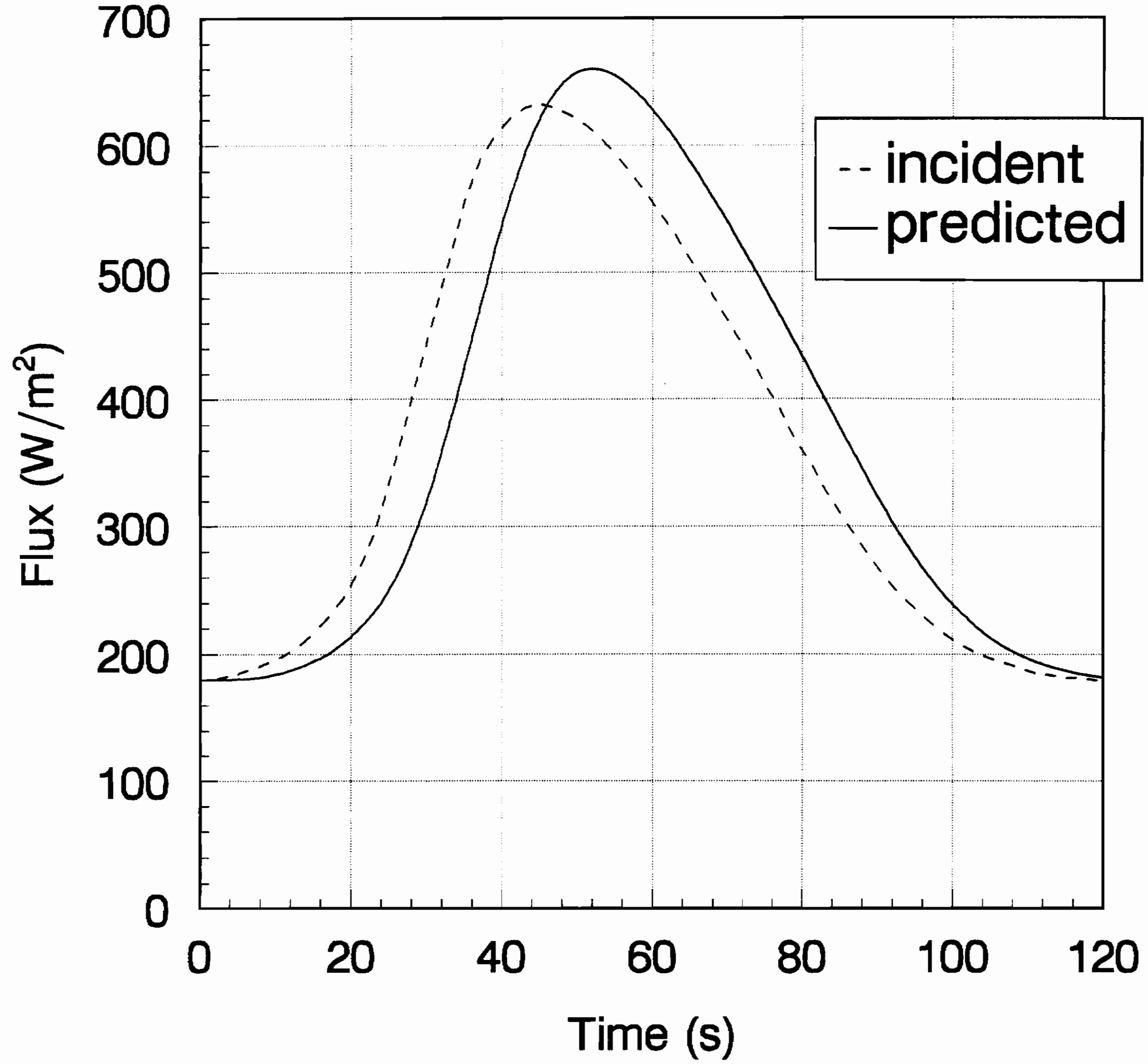

Fig. 21. Wide Field-of-View Total (WFOVT) Channel Simulated Solar Blip Event, Showing the Specified Incident Flux, and the Corresponding Flux Predicted by the End-to-End Model.

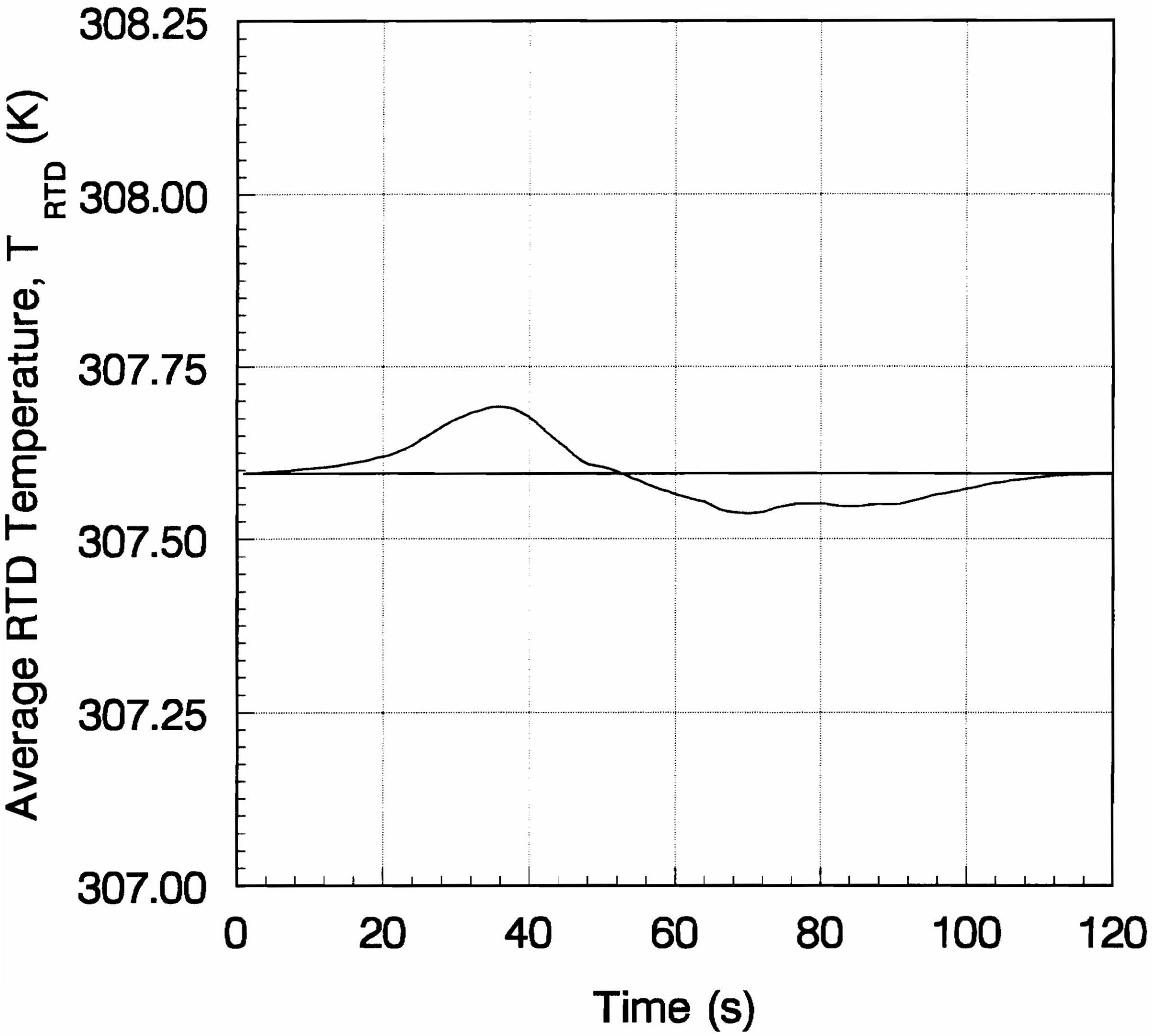

Fig. 22. Average Resistance Temperature Detector (RTD) Temperature Predicted for the Wide Field-of-View Total (WFOVT) Channel Solar Blip Event.

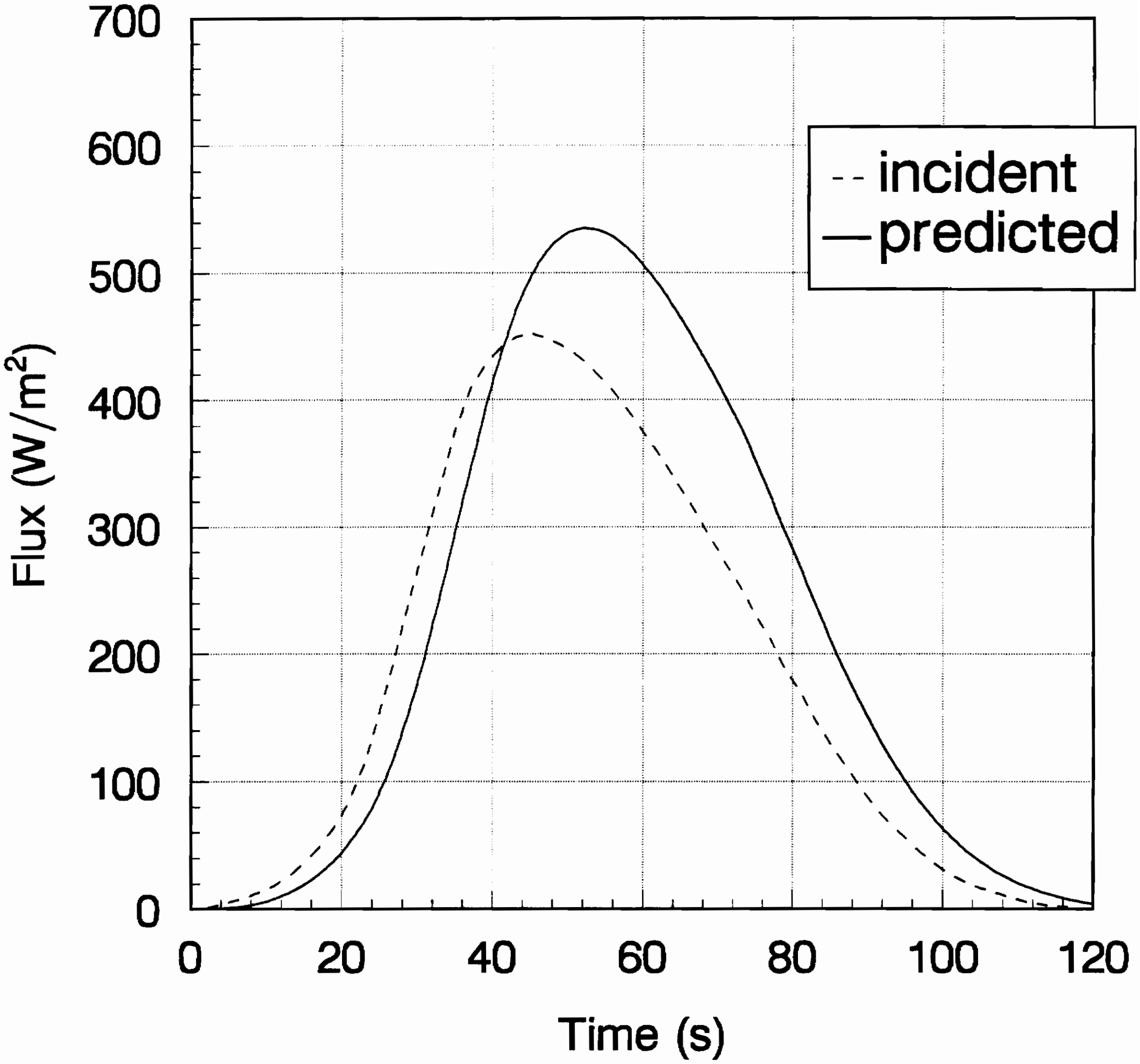

Fig. 23. Wide Field-of-View Shortwave (WFOVSW) Channel Simulated Solar Blip Event, Showing the Specified Incident Flux, and the Corresponding Flux Predicted by the End-to-End Model.

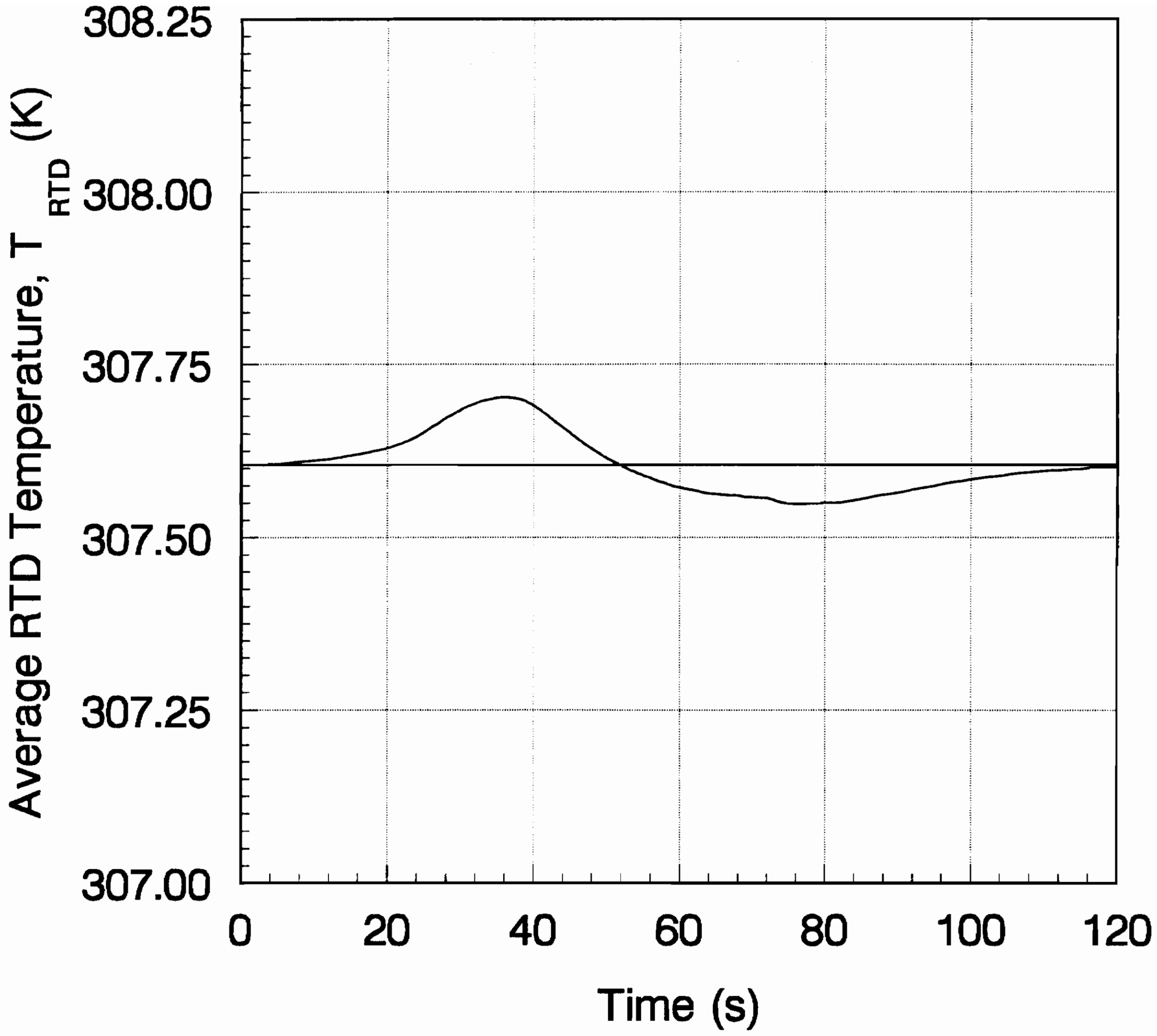

Fig. 24. Average Resistance Temperature Detector (RTD) Temperature Predicted for the Wide Field-of-View Shortwave (WFOVSW) Channel Solar Blip Event.

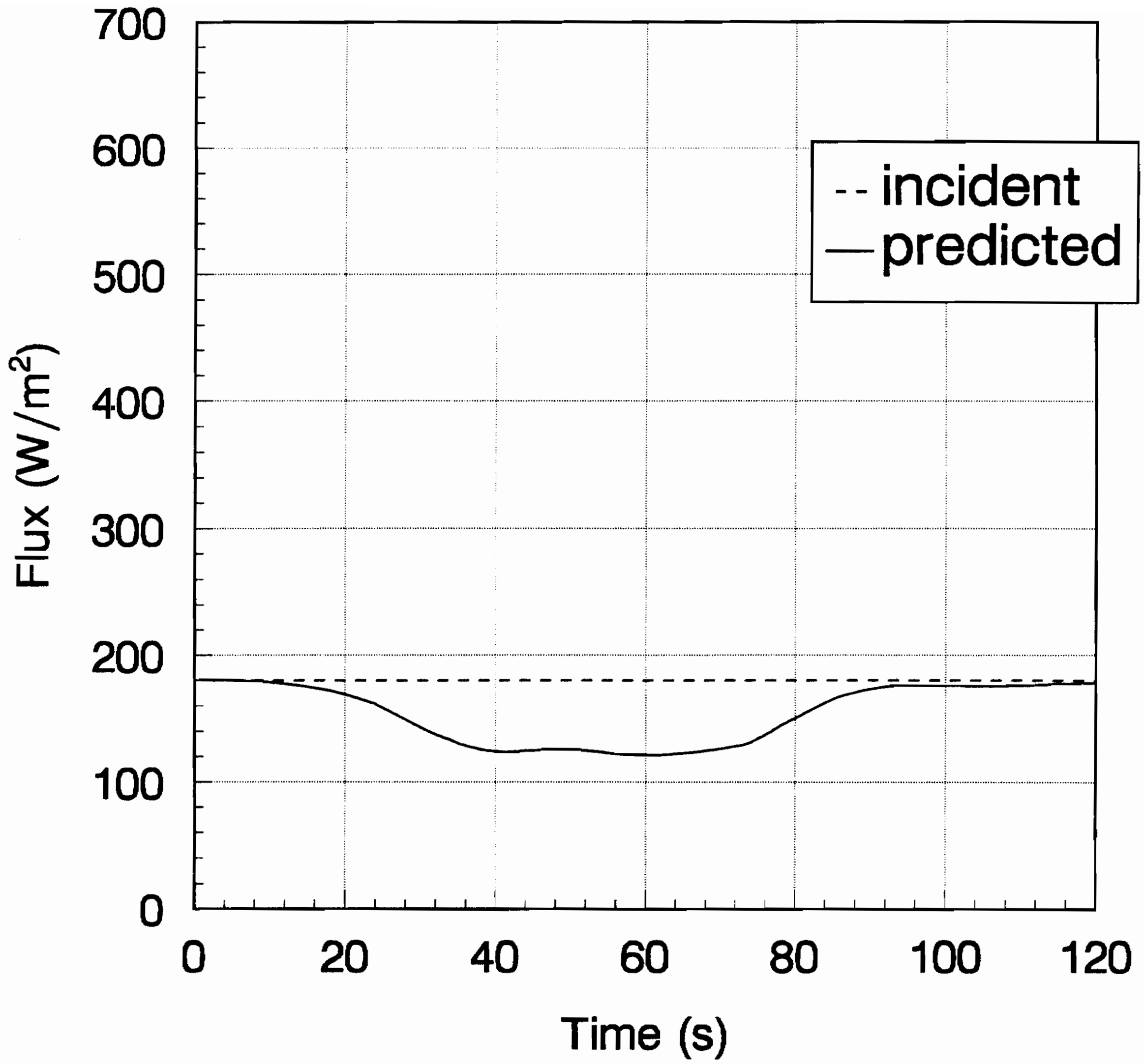

Fig. 25. Specified Incident and the Corresponding Predicted Longwave Flux During a Simulated Solar Blip Event. (Longwave Flux = Total Flux - Shortwave Flux)

References

1. Kann, D. M., Miller, A. J., and Yang, S., *Atmospheric Energetics and Earth Radiation Budget*, **Proceedings of SPIE-The International Society for Optical Engineering**, Vol. 1299, 1990, pp 40-49.

2. Hartman, D., V. Ramanathan, A. Berroir, and G. E. Hunt, *Earth Radiation Budget Data and Climate Research*, **Reviews of Geophysics**, Vol. 24, May 1986, pp 330-350.

3. 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, May 1986, pp 351-356.

4. 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.

5. Barkstrom, B., and G. Louis Smith, *The Earth Radiation Budget Experiment: Science and Implementation*, **Reviews of Geophysics**, Vol. 24, May 1986, pp 379-390.

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

7. Lee III, R. B.,B. R. Barkstrom, N. Halyo, M. A. Gibson, and L. M. Avis, *Characterizations of the Earth Radiation Budget Experiment (ERBE) Scanning Radiometers*, **Proceedings of SPIE-The International Society for Optical Engineering**, Vol. 1109, 1989, pp 186-194.

8. Ramanathan, V., R. D. Cess, E. F. Harrison, P. Minnis, B. R. Barkstrom, E. Ahmad, D. Hartmann, *Cloud-Radiative Forcing and Climate: Results from the Earth Radiation Budget Experiment*, **Science**, Vol. 243, January 1989, pp 57-63.

9. Ramanathan, V., B. R. Barkstrom, E. F. Harrison, *Climate and the Earth's Radiation Budget*, **Physics Today**, May 1989 pp 22-32.

10. Harrison, E. F., P. Minnis, B. R. Barkstrom, V. Ramanathan, R. D. Cess, and G. Gibson, *Seasonal Variation of Cloud Radiative Forcing Derived from the Earth Radiation Budget Experiment*, **Journal of Geophysical Research**, Vol. 95, October 1990, pp 18,687-18,703.

11. Minnis, P.,E. F. Harrison, L. L. Stowe, G. G. Gibson, F. M. Denn, D. R. Doelling, W. L. Smith, Jr., *Radiative Climate Forcing by the Mount Pinatubo Eruption*, **Science**, Vol. 259, March 5, 1993, pp 1369-1508.

12. Lee III, R. B., M. A. Gibson, N. Shivakumar, R. Wilson, H. L. Kyle, and A. T. Mecherikunnel, *Solar Irradiance Measurements: Minimum through Maximum Solar Activity*, **Metrologia**, Vol. 28, 1991, pp 265-268.

13. Lee III, R. B., *Implications of Solar Irradiance Variability Upon Long-Term Changes in the Earth's Atmospheric Temperatures*, **Journal of the National Technical Association**, Vol. 65, Spring 1992, pp 65-71.

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

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

17. 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).

18. 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).

19. Tira, N. E., **Dynamic Simulation of Solar Calibration of the Total, Earth-Viewing Channel of the Earth Radiation Budget Experiment (ERBE)**, M. S. Thesis, Mechanical Engineering DEpartment, Virginia Polytechnic Institute and State University, Blacksburg, VA (1987).

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

21. Tira, N. E., J. R. Mahan, R. B. Lee, III, and R. J. Keynton, *Linear-Array apertures for inflight Dynamic Solar Calibration of Radiometric Channels for Earth Radiation Budget Applications*, **Applied Optics**, in press.

22. Tira, N. E., J. R. Mahan, J. L. Meekins, R. B. Lee, III, and R. J. Keynton, *Optical Characterization of Scanning Thermistor Bolometer Radiometers Using Monte-Carlo Ray Trace Techniques*, **International Radiation Symposium**, Tallinn, Estonia, August 3-8, 1992.

23. Kowsary, F.,**Radiative Characteristics of Spherical Cavities Having Partially or Completely Specular Walls**,Ph. D. Dissertation, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1989).

24. Meekins, J. L., **Optical Analysis of 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. Haeffelin, M. P., **A Numerical Study of Equivalence in Scanning Thermistor Bolometer Radiometers for Earth Radiation Budget Applications**, M. S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1993).

26. Haeffelin, M. P.,J. R. Mahan, R. B. Lee, III, R. J. Keynton, *A Dynamic Model of the ERBE Nonscanning Channels,* Proceedings of SPIE, in press.

27. Haeffelin, M. P.,J. R. Mahan, N. E. Tira, J. L. Meekins, R. B. Lee, III, R. J. Keynton, *A High-Level End-to-End Model for the ERBE Scanning*

Thermistor Bolometer Radiometers, Proceedings of SPIE, in press.

28. Haeffelin, M. P., K. J. Priestley, J. R. Mahan, R. B. Lee, III, R. J. Keynton, N. E. Tira, *Transient Radiation-Conduction Analysis of a Filtered Active-Cavity Radiometer for Earth Radiation Budget Studies*, 6th AIAA/ASME Thermophysics and Heat Transfer Conference, Session on Radiation Heat Transfer, Under Review.

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

30. Chapman, D.,**Monte-Carlo Based Simulation of Jet Exhaust Nozzle and Thermal Radiative Signatures**,M. S. Thesis, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA (1992).

31. Villeneuve, P.,D. Chapman,J. R. Mahan, *Use of the Monte-Carlo Ray-Trace Method as a Design Tool for Jet Engine Infra-red Visibility Supression*, 6th AIAA/ASME Thermophysics and Heat Transfer Conference, Session on Radiation in Energy Systems, Under Review.

32. Luther, M. R., J. E. Cooper, and G. R. Taylor, *The Earth Radiation Budget Experiment Nonscanner Instrument*, **Reviews of Geophysics**, Vol. 24, No. 2, May 1986, pp 391-399.

33. Halyo, N., D. K. Pandey, and D. B. Taylor, **Modeling and Characterization of the Earth Radiation Budget Experiment (ERBE) Nonscanner and Scanner Sensors**, NASA Contractor Report 181818, March 1989.

34. Paden, J., D. K. Pandey, R. S. Wilson, S. Thomas, M. A. Gibson, and R. B. Lee III, *Ground and Inflight Calibrations for the Earth Radiation Budget Experiment (ERBE) Nonscanning Radiometers*, **SPIE Proceedings**, 1300, 1990, pp 190-201.

35. Fellner, R., T. Evert and R. Hesser, *Final Calibration Report on ERBE Flight Model One Nonscanner Instrument*, TRW Document No. D06904, 1984.

36. Priestley, K. J., J. R. Mahan, R. B. Lee, III, R.J. Keynton, *An Uncertainty Analysis for the ERBE Active Cavity Radiometers*, **SPIE Proceedings**, in press.

37. Kline, S. J. and F. A. McClintock,*Describing Uncertainties in Single-Sample Experiments*, **Mechanical Engineering**, Vol. 75, January 1953, pp. 3-8.

38. Paden, J., D. K. Pandey, N.D. Shivakumar, J. C. Stassi, R. Wilson, W. Bolden, S. Thomas and M. A. Gibson, **Radiometer Offsets and Count Conversion Coefficients for the Earth Radiation Budget Experiment (ERBE) Spacecraft for the Years 1984, 1985, and 1986**, NASA Contractor Report 187589, NASA Langley Research Center, May 1991.

39. Haeffelin, M. P. *Personal Communication.*

Appendix A

Program ERBERT

```
**********************************************************************
  ******************************************************************
    **************************************************************
      **********************************************************
        ******************************************************
       **********************************************************
     **************************************************************
  *******************************************************************
***********************************************************************
******                                                           ******
****                  PROGRAM ERBERT                               ****
****                                                               ****
***********************************************************************
*                                                                     *
*                                                                     *
*  THIS CODE IS A MONTE-CARLO-BASED RAY-TRACE ANALYSIS OF THE ERBE    *
*  EARTH-VIEWING NONSCANNING ACTIVE CAVITY RADIOMETERS.               *
*                                                                     *
*  IT COMPUTES THE DISTRIBUTION FACTORS BETWEEN THE VARIOUS           *
*  PARTS OF THE INSTRUMENT.                                           *
*                                                                     *
*  THE DISTRIBUTION FACTORS CAN THEN BE USED TO COMPUTE A THERMAL     *
*  DIFFUSION ANALYSIS IN THE FILTER DOME OR TO OBTAIN THE             *
*  ELECTROTHERMAL RESPONSE OF THE ACTIVE CAVITY.                      *
*                                                                     *
*                                                                     *
*  THIS CODE WAS DEVELOPED BY MARTIAL HAEFFELIN & KORY PRIESTLEY      *
*  UNDER NASA FUNDING.                                                *
*                                                                     *
*                                                                     *
***********************************************************************
*                                                                     *
      PROGRAM ERBERT
      IMPLICIT NONE
C
C VARIABLES FOR THE SPECTRAL DISTRIBUTION OF ENERGY BUNDLES
C
      REAL*8 FREMPO,TS
      REAL*8 LBDA,LBDAMX
C
C VARIABLES FOR GEOMETRY AND PHYSICAL PROPERTIES OF THE MODULE
C
      REAL*8 RFOV,TETAFM,RPA,REXT,RINT,DENIV,PI,ZMAX,CRADAN
      REAL*8 ALPHA1,ALPHA2,ALPHA9,ALPHA11,ALPHA13,ALPHA14
      REAL*8 ROSPEC1,ROSPEC2,ROSPEC3,ROSPEC9,ROSPEC11,ROSPEC13,ROSPEC14
      REAL*8 RODIFF1,RODIFF2,RODIFF3,RODIFF9,RODIFF11,RODIFF13,RODIFF14
      REAL*8 RADF,PHIF,TTAF,SAREA,RCEN
      REAL*8 RCAV,LBAR,LCON,CAVANG,DLCON
      REAL*8 VOLUM,IVAC,IDOM
C
C VARIABLES FOR ABSORPTION/TRANSMISSION IN THE FILTER DOME
C
      REAL*8 DELZ,ALPHAD,LEN,RDMLEN,DOABCO,DOMTHC,RDM
C
C VARIABLES FOR LOCATION AND DIRECTION OF THE TRACED RAY
C
      REAL*8 X0,Y0,Z0,X1,Y1,Z1,X2,Y2,Z2,XX,YY,ZZ,XPTSO,YPTSO
      REAL*8 XT,YT,ZT,XTT,YTT,ZTT
      REAL*8 L,M,N,INORM,ANORM,MAGNV,COSOLD,COSNEW
      REAL*8 ANORM1,ANORM2,ANORM3
      REAL*8 ANORM4,ANORM5,ANORM6
      REAL*8 ANORM7,ANORM8,ANORM9
      REAL*8 XNOD,YNOD,ZNOD
      REAL*8 XFAC,YFAC,ZFAC
C
```

```
C  List of functions
C
      REAL*8 NORME,DOTP,FRAKEMI,F,SUM
C
C OTHER VARIABLES
C
      REAL*8 DT,DIST,MAGMAX,MAGMIN
      REAL*8 COSDOM,carre
      REAL*8 DFAC,DFACTT,DFACSW,DFACLW,CHECKDF,CHECK1,checkj
      REAL*8 CNTREF,DUMSUM,NBBUSW,NBBULW
      REAL*8 XDUM,YDUM,TH
C
C VARIABLES TO SET UP THE SURFACE AND VOLUME ELEMENTS
C

      INTEGER NBLAT,NBLONG,NBWASH,NBPS,NBWLSU,NBLSU,NBSH,NBCO,NBPSD
      INTEGER NBPSB,NBPSC,NBHSB,NBHSC,NBPSCS,NBHSR,NBPSR
      INTEGER NB1,NB2,NB3,NB4,NB5,NB6,NB7,NB9,NB10,NB11,NB12,NB13
      INTEGER NB14,NB15,NB16,NB8,NB17,NB18
      INTEGER NBBUAB,NBVE
      INTEGER EMISUR,NBSURF,NSURI1,NSURI2,NSURI3
      INTEGER CHANEL,CH
      INTEGER EXCAV
C
C VARIOUS COUNTERS
C
      INTEGER CNTBDL,CNTCCL,CNTRDM,CNTBRW,CNTRUN,CNTINT
      INTEGER CNTIND,CNTIN1(100),CNTIN2(100),CNTEMI
C
C VARIOUS INDICES
C
      INTEGER I,J,K,JMAX,SURFAC,BOOL,IN,SIZE,SURF
      INTEGER MAXBDL,ANSWER,FT,WIN,WINIT,IJ,KL
      INTEGER NBWLIN,NBDLIN,NBBUEA
      INTEGER MAXELT,MAXMAX,MAXWLI
      INTEGER SEED1,MENU3
C
C BOOLEANS NECESSARY FOR TESTS
C
      LOGICAL ABSOR,REFL,TRANS,HITTEN,ENTER,EXIT,INDOME,PTSOUR,DIFUS
      LOGICAL EXITCAV
      PARAMETER(MAXELT=700,MAXMAX=750,SIZE=15,MAXWLI=15)
C
C ARRAYS
C
      DIMENSION FREMPO(MAXWLI)
      DIMENSION DT(SIZE),FT(SIZE)
      DIMENSION DFAC(MAXMAX,MAXMAX,MAXWLI)
      DIMENSION X2(SIZE),Y2(SIZE),Z2(SIZE),MAGNV(SIZE)
      DIMENSION COSNEW(SIZE),HITTEN(SIZE),BOOL(SIZE),ANORM(SIZE)
      DIMENSION INORM(SIZE,3),DIST(MAXELT)
C
C  External functions
C
      EXTERNAL NORME,DOTP,FRAKEMI,F,SUM
C
C  Common variables
C
      COMMON /RIING/ NBHSR,NBPSR
      COMMON /CHANNEL/ CHANEL
      COMMON /CAVSUR/ NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS
      COMMON /RAND/ CNTRDM
      COMMON /REAB/ ALPHA1,ALPHA2,ALPHA9,ALPHA11,ALPHA13,ALPHA14,
     &              ROSPEC1,ROSPEC2,ROSPEC9,ROSPEC11,ROSPEC13,ROSPEC14,
     &              RODIFF1,RODIFF2,RODIFF9,RODIFF11,RODIFF13,RODIFF14
```

```
      COMMON /RETR/ ROSPEC3,RODIFF3
      COMMON /TRAN/ IVAC,IDOM
      COMMON /NB14/ NB1,NB2,NB3,NB4,NB9,NB11,NB13,NB14
      COMMON /NB57/ NB5,NB6,NB7,NB15,NB16,NB3,NB17,NB18
      COMMON /SER1/ NBBUAB(MAXELT) /SER2/ NBLAT,NBLONG
      COMMON /SER3/ NBWASH,NBPS,NBLSU,NBWLSU /SER4/ NBSH,NBCO,NBPSD
      COMMON /FOVL/ RFOV,TETAFM /SUBS/ RPA,SURFAC
      COMMON /DOME/ REXT,RINT,DENIV
      COMMON /NSUR/ NSURI1,NSURI2,NSURI3
      COMMON /COUN/ CNTBRW
      COMMON /PTSO/ XPTSO,YPTSO
      COMMON /SETU/ RADF(MAXELT),PHIF(MAXELT),ttaf(maxelt),
     &              SAREA(MAXELT),RCEN
      COMMON /NODE/ XNOD(MAXELT),YNOD(MAXELT),ZNOD(MAXELT)
      COMMON /WLS1/ NBDLIN(MAXWLI),NBBUEA(MAXWLI),NBWLIN
      COMMON /WLS2/ DOABCO(MAXWLI),LBDA(MAXWLI)
      COMMON /CHAN/ CH
      COMMON /FACE/ XFAC(MAXELT),YFAC(MAXELT),ZFAC(MAXELT)
      COMMON /TECP/ DFACTT(1,MAXELT),DFACSW(1,MAXELT),DFACLW(1,MAXELT)
      COMMON /CAVITY/ RCAV,LBAR,LCON,CAVANG
      COMMON /CONSET/ DLCON(20)
C
      OPEN(21,FILE='ttchanel.input',status='old')
      OPEN(22,FILE='swchanel.input',status='old')
      OPEN(12,FILE='earthflu.input',status='old')
      OPEN(03,FILE='nsradan1.dat')
      OPEN(07,FILE='dfacmtx.dat')
      OPEN(08,FILE='tecplot1.dat')
      OPEN(09,FILE='tecplot2.dat')
      OPEN(10,FILE='curfile.dat10')
      OPEN(16,FILE='curfile.dat16')
C
C MENU 1, choose which channel to simulate and specify the number of
C energy bundles to be emitted from each element
C
10    WRITE(06,520)'==================================================='
      WRITE(06,520)' MONTE-CARLO-BASED RAY-TRACE METHOD FOR            '
      WRITE(06,520)' SIMULATION OF RADIATION HEAT TRANSFER IN THE      '
      WRITE(06,520)' FRONT END OF THE ERBE NONSCANNING RADIOMETER.     '
      WRITE(06,520)'      FOR TOTAL CHANNEL SIMULATION        (1)      '
      WRITE(06,520)'      FOR SHORTWAVE CHANNEL SIMULATION    (2)      '
      WRITE(06,520)'==================================================='
      READ(05,*)CHANEL
      CH=CHANEL+20
      WRITE(06,520)'==================================================='
      WRITE(06,520)' SPECIFY THE NUMBER OF BLUNDLES TO BE EMITTED:     '
      WRITE(06,520)'==================================================='
      READ(05,*)MAXBDL
      WRITE(06,520)'==================================================='
      WRITE(06,520)' SPECIFY THE INTERVAL AT WHICH YOU WANT A RESPONSE '
      WRITE(06,520)'==================================================='
      READ(05,*)WINIT
c
c  Initialization for random number generator. Get two seeds from user
c  for initialization.
c
      WRITE(06,520)'==================================================='
      WRITE(06,520)' GIVE AN INTEGER BETWEEN 0 AND 2,147,483,647:      '
      WRITE(06,520)'==================================================='
      READ(05,*)SEED1
      CALL RNSET(SEED1)
C
C GET GEOMETRY OF THE PROBLEM FROM INPUT FILE
C
```

```
      READ(CH,*)
      READ(CH,*)
      READ(CH,*)RFOV
      READ(CH,*)RCAV
      READ(CH,*)LCON
      READ(CH,*)LBAR
      READ(CH,*)TETAFM
      READ(CH,*)RPA
      READ(CH,*)REXT
      READ(CH,*)RINT
      READ(CH,*)DENIV
      READ(CH,*)ALPHA1
      READ(CH,*)ALPHA2
      READ(CH,*)ALPHA9
      READ(CH,*)ALPHA11
      READ(CH,*)ALPHA13
      READ(CH,*)ALPHA14
      READ(CH,*)ROSPEC1
      READ(CH,*)ROSPEC2
      READ(CH,*)ROSPEC3
      READ(CH,*)ROSPEC9
      READ(CH,*)ROSPEC11
      READ(CH,*)ROSPEC13
      READ(CH,*)ROSPEC14
      READ(CH,*)RODIFF1
      READ(CH,*)RODIFF2
      READ(CH,*)RODIFF3
      READ(CH,*)RODIFF9
      READ(CH,*)RODIFF11
      READ(CH,*)RODIFF13
      READ(CH,*)RODIFF14
      READ(CH,*)IVAC
      READ(CH,*)IDOM
      READ(CH,*)NBLAT
      READ(CH,*)NBLONG
      READ(CH,*)NBLSU
      READ(CH,*)NBWLSU
      READ(CH,*)NBWASH
      READ(CH,*)NBPS
      READ(CH,*)NBSH
      READ(CH,*)NBCO
      READ(CH,*)NBPSD
      READ(CH,*)NBPSB
      READ(CH,*)NBPSC
      READ(CH,*)NBPSR
      READ(CH,*)NBHSB
      READ(CH,*)NBHSC
      READ(CH,*)NBPSCS
      READ(CH,*)NBHSR
      READ(CH,*)TS
C
C Calculate the cone angle
C
      CAVANG=DATAN(RCAV/LCON)
      WRITE(06,*)'CAVANG',CAVANG
C
C INITIALIZE EXITCAV COUNTER
C
      EXCAV=0
C
C Read wavelength of the wl intervals.
C LBDA(I) is the upper limit of the wl interval I
C
      LBDAMX=50.0D0
      READ(CH,*)NBWLIN
```

```
      DO 15 I=1,NBWLIN-1
          READ(CH,*)LBDA(I)
15    CONTINUE
      LBDA(NBWLIN)=LBDAMX
C
C  Read the absorbtivity for each wavelength interval
C
      READ(CH,*)DOMTHC
      DO 16 I=1,NBWLIN
          READ(CH,*)DOABCO(I)
          DOABCO(I)=DLOG(1.D0/DOABCO(I))/DOMTHC
16    CONTINUE
C
C Go back to top of input file
C
      REWIND CH
      WRITE(06,*)' FILE 1 READ, NO PB'
C
C MENU 2, start execution or check the input file
C
      WRITE(06,520)'==================================================='
      WRITE(06,520)' INPUT FILE HAS BEEN READ. SPECIFY NEXT OPERATION  '
      WRITE(06,520)' TO EXECUTE:                                       '
      WRITE(06,520)'    "1": START EXECUTION OF THE PROGRAM DIRECTLY,  '
      WRITE(06,520)'    "2": VIEW INPUT FILE ON SCREEN.                '
      WRITE(06,520)'==================================================='
      READ(05,*)ANSWER
C
      IF (ANSWER.EQ.2) THEN
C
C SHOW THE USER WHAT THE GEOMETRY IS
C
      WRITE(06,520)'==================================================='
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,522)'RADIUS OF FOV LIMITER (MM)',RFOV
      WRITE(6,522)'OPENING OF FOV LIMITER (RAD)',TETAFM
      WRITE(6,522)'RADIUS OF PRECISION APERTURE',RPA
      WRITE(6,522)'EXTERNAL RADIUS OF DOME (MM)',REXT
      WRITE(6,522)'INTERNAL RADIUS OF DOME (MM)',RINT
      WRITE(6,522)'DEPTH AT WHICH DOME IS BURIED IN SUBSTR. (MM)',DENIV
      WRITE(6,522)'RADIUS OF CAVITY',RCAV
      WRITE(6,522)'LENGTH OF BARREL',LBAR
      WRITE(6,522)'LENGTH OF CONE',LCON
      WRITE(6,522)'HALF ANGLE OF CONE',CAVANG
      WRITE(6,522)'COEF. OF ABSORBTION OF FOV LIMITER',ALPHA1
      WRITE(6,522)'COEF. OF ABSORBTION OF SUBSTRATE',ALPHA2
      WRITE(6,522)'COEF. OF ABSORPTION OF BARREL',ALPHA9
      WRITE(6,522)'COEF. OF ABSORPTION OF CONE',ALPHA11
      WRITE(6,522)'COEF. OF ABSORPTION OF CAV SUB',ALPHA13
      WRITE(6,522)'COEF. OF ABSORPTION OF RING',ALPHA14
      WRITE(6,522)'SPECULARITY OF FOV LIMITER',ROSPEC1
      WRITE(6,522)'SPECULARITY OF SUBSTRATE',ROSPEC2
      WRITE(6,522)'SPECULARITY OF DOME',ROSPEC3
      WRITE(6,522)'SPECULARITY OF BARREL',ROSPEC9
      WRITE(6,522)'SPECULARITY OF CONE',ROSPEC11
      WRITE(6,522)'SPECULARITY OF CAV SUB',ROSPEC13
      WRITE(6,522)'SPECULARITY OF RING',ROSPEC14
      WRITE(6,522)'DIFFUSIVITY OF FOV LIMITER',RODIFF1
      WRITE(6,522)'DIFFUSIVITY OF SUBSTRATE',RODIFF2
      WRITE(6,522)'DIFFUSIVITY OF DOME',RODIFF3
      WRITE(6,522)'DIFFUSIVITY OF BARREL',RODIFF9
      WRITE(6,522)'DIFFUSIVITY OF CONE',RODIFF11
      WRITE(6,522)'DIFFUSIVITY OF CAV SUB',RODIFF13
      WRITE(6,522)'DIFFUSIVITY OF RING',RODIFF14
```

```
      WRITE(6,522)'INDEX OF REFRACTION IN VACCUM',IVAC
      WRITE(6,522)'INDEX OF REFRACTION OF THE DOME',IDOM
      WRITE(6,523)'NBLAT',NBLAT
      WRITE(6,523)'NBLONG',NBLONG
      WRITE(6,523)'NBLSU',NBLSU
      WRITE(6,523)'NBWLSU',NBWLSU
      WRITE(6,523)'NBWASH',NBWASH
      WRITE(6,523)'NBPS',NBPS
      WRITE(6,523)'NBSH',NBSH
      WRITE(6,523)'NBCO',NBCO
      WRITE(6,523)'NBPSD',NBPSD
      WRITE(6,523)'NBPSB',NBPSB
      WRITE(6,523)'NBPSC',NBPSC
      WRITE(6,523)'NBHSB',NBHSB
      WRITE(6,523)'NBHSC',NBHSC
      WRITE(6,523)'NBHSR',NBHSR
      WRITE(6,523)'NBPSCS',NBPSCS
      WRITE(6,523)'NBPSR',NBPSR
      WRITE(6,522)'SURROUNDING TEMP',TS
      WRITE(06,523)'NBER OF WL INTERVALS',NBWLIN
      DO 19 I=1,NBWLIN
          WRITE(06,522)'WAVELENGTH',LBDA(I)
19    CONTINUE
      DO 22 I=1,NBWLIN
          WRITE(06,522)'ABSORPTIVITY OF DOME',DOABCO(I)
22    CONTINUE
      WRITE(06,520)'==================================================='
      WRITE(06,520)' TO START PROGRAM: ENTER 1; TO EXIT: ENTER 0.      '
      WRITE(06,520)'==================================================='
      READ*,ANSWER
      IF (ANSWER.EQ.0) GO TO 270
      END IF
C
C Initialization of arrays and variables for each run of program
C
      PI=DACOS(-1.D0)
      CNTRUN=0
      CNTRDM=0
      NB1 = NBLAT*NBLONG
      NB2 = NB1 + NBLSU*NBWLSU
      NB3 = NB2 + NBWASH*NBPS
      NB4 = NB3 + NBSH*NBCO*NBPSD
      NB11 = NB4 + 1 + NBPSC*NBHSC
      NB9 = NB11 + NBPSB*NBHSB
      NB13 = NB9 + NBPSCS
      NB14 = NB13 + NBPSR*NBHSR
      NB5 = (NBLAT + 1)*NBLONG
      NB6 = NB5 + (NBWLSU+1)*NBLSU
      NB7 = NB6 + (NBWASH + 1)*NBPS
      NB15 = NB8 + (NBHSB + 1)*NBPSB
      NB16 = NB15 + (NBHSC + 1)*NBPSC
      NB17 = NB16 + 2*NBPSCS
      NB18 = NB17 + (NBHSR + 1)*NBPSR
      ZMAX=RFOV*DCOS(TETAFM)+DENIV
      WRITE(*,*)'NB1,NB2,NB3,NB4,NB9,NB11',NB1,NB2,NB3,NB4,NB9,NB11
C
C Show indices range for each part of the optical front end
C
      WRITE(03,520)'==================================================='
      WRITE(03,520)' ELEMENT INDEX FOR PARTS OF THE OPTICAL FRONT-END  '
      WRITE(03,520)'==================================================='
      WRITE(03,530)' OPENING OF THE FOV LIMITER: ',NB4+1,' - ',NB4+1
      WRITE(03,530)' FOV LIMITER WALLS         : ',1,' - ',NB1
      WRITE(03,530)' LOWER SUBSTRATE           : ',NB1+1,' - ',NB2
      WRITE(03,530)' UPPER SUBSTRATE           : ',NB2+1,' - ',NB3
```

```
      WRITE(03,530)' FILTER DOME                 : ',NB3+1,' - ',NB4
      WRITE(03,530)' BARREL                      : ',NB11+1,' - ',NB9
      WRITE(03,530)' CONE                        : ',NB4+2,' - ',NB11
      WRITE(03,530)' SUBST. VISIBLE TO CAVITY    : ',NB9+1,' - ',NB13
      WRITE(03,530)' RING                        : ',NB13+1,' - ',NB14
      WRITE(03,520)'===================================================='
530   FORMAT(1X,A30,I4,A3,I4)
      WRITE(03,*)'NB5,NB6,NB7',NB5,NB6,NB7
C
C SET UP THE DOME IN VOLUME ELEMENTS
C
      CALL DOMSETUP(DIST)
C
C SET UP THE CONE IN SURFACE ELEMENTS (FIND HEIGHTS OF RINGS FOR
C EQUAL AREA ELEMENTS)
C
      CALL CONSETUP(DLCON)
C
C MENU 3, choose between complete DFAC generation procedure and emission
C from one element only
C
899   WRITE(06,520)'==================================================='
      WRITE(06,520)' FOR COMPLETE DFAC GENERATION, ENTER (1)           '
      WRITE(06,520)' FOR EMISSION FROM A GIVEN ELEMENT, ENTER (2)      '
      WRITE(06,520)'==================================================='
      READ(05,*)MENU3
      CNTIND = 1
      IF (MENU3.EQ.1) THEN
C
C Procedure for computation of distribution factors
C Two indices are used. CNTIN1 contains the value of EMISUR (1,2,3,4,9,11)
C and CNTIN2 contains the value of NBSURF (1-235). The emission elements
C are in the order:
C                    the aperture of the FOV Limiter
C                    the FOV limiter walls
C                    the lower substrate
C                    the upper substrate
C                    the filter dome
C                    the barrel
C                    the cone
C                    the cavsub
C                    the ring
C
C CNTIND is a counter which keeps track of the number of emitting surfaces.
C CNTEMI is a counter which keeps track of how many surfaces have emitted
C so far. When CNTEMI reaches CNTIND the DFAC computation is completed.
C
C
C  FOV APERTURE
C
      CNTIN1(1) = 0
      CNTIN2(1) = NB4 + 1
C
C  FOVL
C
      DO 900 I = 1,NBLAT
        CNTIND = CNTIND + 1
        CNTIN1(CNTIND) = 1
        CNTIN2(CNTIND) = (I-1)*NBLONG + 1
900   CONTINUE
C
C  LSUB
C
      DO 901 I=1,NBWLSU
        CNTIND = CNTIND + 1
```

```
        CNTIN1(CNTIND) = 2
        CNTIN2(CNTIND) = NB1 + (I-1)*NBLSU + 1
901   CONTINUE
C
C  USUB
C
      DO 902 I=1,NBWASH
        CNTIND = CNTIND + 1
        CNTIN1(CNTIND) = 3
        CNTIN2(CNTIND) = (I-1)*NBPS + 1 + NB2
902   CONTINUE
C
C  DOME
C
      IF (CH.EQ.22) THEN
        DO 903 I=1,NBSH*NBCO
          CNTIND = CNTIND + 1
          CNTIN1(CNTIND) = 4
          CNTIN2(CNTIND) = (I-1)*NBPSD + 1 + NB3
903     CONTINUE
      END IF
C
C  CONE
C
      DO 905 I=1,NBHSC
        CNTIND = CNTIND + 1
        CNTIN1(CNTIND) = 11
        CNTIN2(CNTIND) = (I-1)*NBPSC + 1 + NB9
905   CONTINUE
C
C  BARREL
C
      DO 904 I=1,NBHSB
        CNTIND = CNTIND + 1
        CNTIN1(CNTIND) = 9
        CNTIN2(CNTIND) = (I-1)*NBPSB + 1 + NB4 +1
904   CONTINUE
C
C  CAVSUB
C
        CNTIND = CNTIND +1
        CNTIN1(CNTIND) = 13
        CNTIN2(CNTIND) = NB11 + 1
C
C  RING
C
      IF (CH.EQ. 21) THEN
        DO 907 I=1,NBHSR
          CNTIND = CNTIND + 1
          CNTIN1(CNTIND) = 14
          CNTIN2(CNTIND) = NB13+1+(I-1)*NBPSR
907     CONTINUE
      END IF
      WRITE(03,520)'==================================================='
      WRITE(03,520)' LIST OF EMITTING ELEMENTS FOR DFAC GENERATION     '
      WRITE(03,520)'==================================================='
      DO 908 I=1,CNTIND
        IF (CNTIN1(I).EQ.0) WRITE(03,531)'FIELD  ',CNTIN1(I),CNTIN2(I)
        IF (CNTIN1(I).EQ.1) WRITE(03,531)'FOV_L  ',CNTIN1(I),CNTIN2(I)
        IF (CNTIN1(I).EQ.2) WRITE(03,531)'L_SUB  ',CNTIN1(I),CNTIN2(I)
        IF (CNTIN1(I).EQ.3) WRITE(03,531)'U_SUB  ',CNTIN1(I),CNTIN2(I)
        IF (CNTIN1(I).EQ.4) WRITE(03,531)'DOME   ',CNTIN1(I),CNTIN2(I)
        IF (CNTIN1(I).EQ.9) WRITE(03,531)'BARREL ',CNTIN1(I),CNTIN2(I)
        IF (CNTIN1(I).EQ.11) WRITE(03,531)'CONE   ',CNTIN1(I),CNTIN2(I)
        IF (CNTIN1(I).EQ.13) WRITE(03,531)'CAVSUB ',CNTIN1(I),CNTIN2(I)
```

```
        IF (CNTIN1(I).EQ.14) WRITE(03,531)'RING   ',CNTIN1(I),CNTIN2(I)
908   CONTINUE
      WRITE(04,520)'==================================================='
531   FORMAT(1X,A10,2I10)
C
      ELSE IF (MENU3.EQ.2) THEN
C
C MENU 4, choose the emission element
C
      WRITE(06,520)'==================================================='
      WRITE(06,520)' FROM WHICH PART OF THE NONSCANNER MODULE DO YOU   '
      WRITE(06,520)' WISH TO SEND THE ENERGY BUNDLES?                 '
      WRITE(06,520)'    "0"  : THE APERTURE OF THE FOV LIMITER,        '
      WRITE(06,520)'    "1"  : THE FOV LIMITER ITSELF,                 '
      WRITE(06,520)'    "2"  : THE LOWER SUBSTRATE,                    '
      WRITE(06,520)'    "3"  : THE UPPER SUBSTRATE,                    '
      WRITE(06,520)'    "4"  : THE FILTER DOME.                        '
      WRITE(06,520)'    "9"  : THE BARREL,                             '
      WRITE(06,520)'    "11" : THE CONE.                               '
      WRITE(06,520)'    "13" : THE SUBST. VISIBLE TO THE CAVITY        '
      WRITE(06,520)'    "14" : THE RING                                '
      WRITE(06,520)'==================================================='
      READ(05,*)CNTIN1(1)
C
C For emission from the secondary aperture specify the emission type
C
      IF (CNTIN1(1).EQ.0) THEN
      CNTIN2(1)=NB4+1
      WRITE(06,520)'==================================================='
      WRITE(06,520)' THE APERTURE OF THE FOV LIMITER IS NOT DIVIDED    '
      WRITE(06,520)' INTO SURFACE ELEMENTS.                            '
      WRITE(06,520)' PLEASE SPECIFY WHETHER EMISSION FROM THIS SURFACE '
      WRITE(06,520)' SHOULD BE:                                        '
      WRITE(06,520)'             DIFFUSE                           (1) '
      WRITE(06,520)'             COLLIMATED                        (2) '
      WRITE(06,520)'             POINT SOURCE COLLIMATED           (3) '
      WRITE(06,520)'==================================================='
      READ(05,*)ANSWER
C
C FOR COLLIMATED RADIATION THE INCIDENT ANGLE MUST BE SPECIFIED
C
      IF (ANSWER.EQ.2.OR.ANSWER.EQ.3) THEN
      WRITE(06,520)'==================================================='
       WRITE(06,520)' PLEASE SPECIFY THE ANGLE OF THE COLLIMATED       '
       WRITE(06,520)' RADIATION (BETWEEN 0 AND 90 DEGREES)             '
      WRITE(06,520)'==================================================='
       READ(05,*)CRADAN
       CRADAN=CRADAN*PI/180.D0
      ELSE
       CRADAN=-1.D0
      END IF
C
C FOR A POINT SOURCE THE COORD. OF THE EMITTING POINT MUST BE SPECIFIED
C
      IF (ANSWER.EQ.3) THEN
      WRITE(06,520)'==================================================='
       WRITE(06,520)' PLEASE SPECIFY THE X,Y COORDINATES OF THE POINT  '
       WRITE(06,520)' FROM WHICH THE POINT SOURCE COMES.'
       WRITE(06,521)-RFOV*DSIN(TETAFM),' <X AND Y< ',RFOV*DSIN(TETAFM)
      WRITE(06,520)'==================================================='
       READ(05,*)XPTSO,YPTSO
       PTSOUR=.TRUE.
      ELSE
       PTSOUR=.FALSE.
      END IF
```

```
C
      ELSE
C
C For emission from all other surfaces specify emission element.
C
      WRITE(06,520)'==================================================='
      WRITE(06,520)' SPECIFY THE NUMBER OF THE ELEMENT FROM            '
      WRITE(06,520)' WHICH THE BUNDLES ARE GOING TO BE EMITED:         '
      WRITE(06,*)
C
C  FOVL
C
      IF (CNTIN1(1).EQ.1) THEN
      WRITE(06,520)' THE FIRST ELEMENT ON THE FOV LIMITER IS LOCATED AT'
      WRITE(06,520)' THE LOWEST LATITUDE AND NOON LONGITUDE.           '
      WRITE(06,526)' FOR EACH OF THE',NBLAT,'LATITUDES THERE ARE',NBLONG
      WRITE(06,520)' LONGITUDES'
      WRITE(06,525)' THE NUMBERS RUN BETWEEN',1,'AND',NB1
      WRITE(06,520)'==================================================='
       READ(05,*)CNTIN2(1)
      WRITE(06,520)'==================================================='
      END IF
C
C  LSUB
C
      IF (CNTIN1(1).EQ.2) THEN
      WRITE(06,520)' THE FIRST ELEMENT ON THE LOWER SUBSTRATE IS THE   '
      WRITE(06,520)' PRIMARY APERTURE, THE FOLLOWING ELEMENTS SURROUND '
      WRITE(06,520)' THE APERTURE, SPIRALING OUTWARDS.                 '
      WRITE(06,526)' FOR EACH OF THE',NBWLSU,'RINGS THERE ARE',NBLSU
      WRITE(06,520)' PIE-SECTIONS'
      WRITE(06,525)' THE NUMBERS RUN BETWEEN',NB1+1,'AND',NB2
      WRITE(06,520)'==================================================='
       READ(05,*)CNTIN2(1)
      WRITE(06,520)'==================================================='
      END IF
C
C  USUB
C
      IF (CNTIN1(1).EQ.3) THEN
      WRITE(06,520)' THE FIRST ELEMENT ON THE SUBSTRATE IS ON THE RING '
      WRITE(06,520)' CLOSEST TO THE CENTER.                            '
      WRITE(06,526)' FOR EACH OF THE',NBWASH,'RINGS THERE ARE',NBPS
      WRITE(06,520)' PIE-SECTIONS'
      WRITE(06,525)' THE NUMBERS RUN BETWEEN',NB2+1,'AND',NB3
      WRITE(06,520)'==================================================='
       READ(05,*)CNTIN2(1)
      WRITE(06,520)'==================================================='
      END IF
C
C  DOME
C
      IF (CNTIN1(1).EQ.4) THEN
      WRITE(06,520)' THE FIRST ELEMENT IN THE DOME IS LOCATED ON THE   '
      WRITE(06,520)' OF THE DOME.                                      '
      WRITE(06,526)' FOR EACH OF THE',NBSH,'SHELLS THERE ARE',NBCO
      WRITE(06,526)' CONES, THERE ARE',NBPSD,'PIE-SECTIONS PER CONE'
      WRITE(06,525)' THE NUMBERS RUN BETWEEN',NB3+1,'AND',NB4
      WRITE(06,520)'==================================================='
       READ(05,*)CNTIN2(1)
      WRITE(06,520)'==================================================='
      END IF
C
C  BARREL
C
```

```
      IF (CNTIN1(1).EQ.9) THEN
      WRITE(06,520)' THE FIRST ELEMENT ON THE BARREL IS LOCATED ON THE '
      WRITE(06,520)' LOWEST RING AT NOON LONGITUDE.                  '
      WRITE(06,526)' FOR EACH OF THE',NBHSB,'RINGS THERE ARE ',NBPSB
      WRITE(06,520)' AZIMUTHAL DIVISIONS.'
      WRITE(06,525)' THE NUMBERS RUN BETWEEN',NB4+2,'AND',NB9
      WRITE(06,520)'==================================================='
       READ(05,*)CNTIN2(1)
      WRITE(06,520)'==================================================='
      END IF
C
C  CONE
C
      IF (CNTIN1(1).EQ.11) THEN
      WRITE(06,520)' THE FIRST ELEMENT ON THE CONE IS LOCATED ON THE   '
      WRITE(06,520)' LOWEST RING AT NOON LONGITUDE.                  '
      WRITE(06,526)' FOR EACH OF THE',NBHSC,'RINGS THERE ARE',NBPSC
      WRITE(06,520)' AZIMUTHAL DIVISIONS.'
      WRITE(06,525)' THE NUMBERS RUN BETWEEN',NB9+1,'AND',NB11
      WRITE(06,520)'==================================================='
       READ(05,*)CNTIN2(1)
      WRITE(06,520)'==================================================='
      END IF
C
C  CAVSUB
C
      IF(CNTIN1(1).EQ.13) THEN
      WRITE(06,520)' THE FIRST ELEMENT ON THE CAV SUB RING IS LOCATED  '
      WRITE(06,520)' AT NOON LONGITUDE.                                '
      WRITE(06,520)' THERE ARE',NBPSCS,'AZIMUTHAL DIVISIONS.'
      WRITE(06,525)' THE NUMBERS RUN BETWEEN',NB9+1,'AND',NB13
      WRITE(06,520)'==================================================='
       READ(05,*)CNTIN2(1)
      WRITE(06,520)'==================================================='
      END IF
C
C  RING
C
      IF (CNTIN1(1).EQ.14) THEN
      WRITE(06,520)' THE FIRST ELEMENT ON THE RING IS LOCATED ON THE '
      WRITE(06,520)' HIGHEST LATITUDE AT NOON LONGITUDE.             '
      WRITE(06,526)' FOR EACH OF THE',NBHSR,'RINGS THERE ARE ',NBPSR
      WRITE(06,520)' AZIMUTHAL DIVISIONS.'
      WRITE(06,525)' THE NUMBERS RUN BETWEEN',NB13+1,'AND',NB14
      WRITE(06,520)'==================================================='
       READ(05,*)CNTIN2(1)
      WRITE(06,520)'==================================================='
      END IF
C
      END IF
C
      ELSE
       GO TO 899
C
C End of MENU 3
C
      END IF
C
C Zero elements have emitted so far.
C
      CNTEMI = 0
C
C Start loop for each emitting element
C
922   CNTINT=0
```

```
      CNTCCL=0
      CNTBDL=0
      CNTBRW=0
      CNTRUN=CNTRUN+1
      CNTREF=-1.0
      WIN = WINIT
      DO 20 I=1,SIZE
          FT(I)=0
          DT(I)=0.D0
20    CONTINUE
      DO 21 I=1,MAXELT
          NBBUAB(I)=0
21    CONTINUE
      CALL INIT(X2)
      CALL INIT(Y2)
      CALL INIT(Z2)
      CALL INIT(MAGNV)
      CALL INIT(COSNEW)
      CALL INIT(ANORM)
C
C Retrieve the i,j,k indices from EMISUR & NBSURF
C
      CNTEMI = CNTEMI + 1
      EMISUR = CNTIN1(CNTEMI)
      NBSURF = CNTIN2(CNTEMI)
      IF (EMISUR.EQ.0.AND.MENU3.EQ.1) THEN
        CRADAN = -1.D0
        PTSOUR = .FALSE.
      ELSE IF (EMISUR.EQ.1) THEN
        NSURI1=INT((NBSURF-1)/NBLONG)+1
        NSURI2=NBSURF-(NSURI1-1)*NBLONG
        NSURI3=0
      ELSE IF (EMISUR.EQ.2) THEN
        NSURI1=0
        NSURI2=INT((NBSURF-1-NB1)/NBLSU) + 1
        NSURI3=NBSURF-NB1-(NSURI2-1)*NBLSU
      ELSE IF (EMISUR.EQ.3) THEN
        NSURI1=INT((NBSURF - 1 - NB2) / NBPS) + 1
        NSURI2=NBSURF - NB2 - (NSURI1 - 1) * NBPS
        NSURI3=0
      ELSE IF (EMISUR.EQ.4) THEN
        NSURI1 = INT((NBSURF - NB3 - 1)/(NBPSD*NBCO)) + 2
        NSURI2 = INT((REAL(NBSURF) - REAL(NB3) - (REAL(NSURI1) - 2.0)*
     &           REAL(NBCO)*REAL(NBPSD) - 1.0)/(REAL(NBPSD))) + 2
        NSURI3 = NBSURF - NB3 - (NSURI1 - 2)*NBCO*NBPSD - (NSURI2 - 2)
     &          *NBPSD
      ELSE IF (EMISUR.EQ.9) THEN
        NSURI1=INT((NBSURF-1-NB4-1)/NBPSB)+1
        NSURI2=(NBSURF-NB4-1)-((NSURI1-1)*NBPSB)
        NSURI3=0
      ELSE IF (EMISUR.EQ.11) THEN
        NSURI1=INT((NBSURF-1-NB9)/NBPSC)+1
        NSURI2=(NBSURF-NB9)-((NSURI1-1)*NBPSC)
        NSURI3=0
      ELSE IF (EMISUR.EQ.13) THEN
        NSURI1=INT((NBSURF-1-NB11)/NBPSCS)+1
        NSURI2=0
        NSURI3=0
      ELSE IF (EMISUR.EQ.14) THEN
        NSURI1=INT((NBSURF-1-NB13)/NBPSR)+1
        NSURI2=(NBSURF-NB13)-((NSURI1-1)*NBPSR)
        NSURI3=0
      END IF
C
C Set up the spectral distribution of the energy bundles
```

```
C
      CALL WLSET(MAXBDL,EMISUR)
C
C BEGINNING OF LOOP FOR EACH WAVELENGTH INTERVAL
C
29    CNTINT=CNTINT+1
      ALPHAD=DOABCO(CNTINT)
      DO 28 J=1,NB14
          NBBUAB(J)=0
28    CONTINUE
C
C BEGINNING OF LOOP FOR EACH EMITTED BUNDLE
C
30    CNTBDL=CNTBDL+1
      CNTREF=CNTREF+1.0
      SURFAC=0
      INDOME=.FALSE.
      ABSOR=.FALSE.
      ENTER=.FALSE.
      EXIT=.FALSE.
      REFL=.FALSE.
      DIFUS=.FALSE.
      TRANS=.FALSE.
      DO 40 I=1,15
          HITTEN(I)=.FALSE.
40    CONTINUE
C
C GIVE A MESSAGE TO THE USER THAT THE PROGRAM IS RUNNING
C
      IF (CNTBDL.EQ.WIN) THEN
          WIN = WIN + WINIT
          WRITE(06,41)'RUNNING FINE; SHOT MORE THAN ',CNTBDL,' BUNDLES'
41        FORMAT(1X,A30,I8,A10)
      END IF
C
C RANDOM EMISSION FROM RANDOM LOCATION ON DEFINED SURFACE ELEMENT
C
      IF (EMISUR.EQ.0) THEN
          CALL EMIT0(X1,Y1,Z1,L,M,N,CRADAN,PTSOUR)
      ELSE IF (EMISUR.EQ.1) THEN
          CALL EMIT1(X1,Y1,Z1,L,M,N)
      ELSE IF (EMISUR.EQ.2) THEN
          CALL EMIT2(X1,Y1,Z1,L,M,N,INDOME)
      ELSE IF (EMISUR.EQ.3) THEN
          CALL EMIT3(X1,Y1,Z1,L,M,N)
      ELSE IF (EMISUR.EQ.4) THEN
          CALL EMIT4(X1,Y1,Z1,L,M,N,anorm)
          INDOME = .TRUE.
      ELSE IF (EMISUR.EQ.9) THEN
          CALL EMIT9(X1,Y1,Z1,L,M,N,ANORM1,ANORM2,ANORM3)
      ELSE IF (EMISUR.EQ.11) THEN
          CALL EMIT11(X1,Y1,Z1,L,M,N)
      ELSE IF (EMISUR.EQ.13) THEN
          CALL EMIT13(X1,Y1,Z1,L,M,N,ANORM4,ANORM5,ANORM6)
      ELSE IF (EMISUR.EQ.14) THEN
          CALL EMIT14(X1,Y1,Z1,L,M,N,ANORM7,ANORM8,ANORM9)
      END IF
      XT=X1
      YT=Y1
      ZT=Z1
C
C GET COORD. OF INTERSECTION OF THE FIRST EMITTED BUNDLE WITH EACH
C SURFACE INSIDE THE FOV LIMITER
C
      CALL FOVLIM(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN,EXIT)
```

```
      CALL SUBSTR(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN,ENTER)
      IF (CH.EQ.22) THEN
        CALL SPHERE(X1,Y1,Z1,L,M,N,X2,Y2,Z2,3,MAGNV,INORM,HITTEN)

        CALL SPHERE(X1,Y1,Z1,L,M,N,X2,Y2,Z2,4,MAGNV,INORM,HITTEN)
      ELSE
        CALL RING(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN)
      END IF
      CALL BARREL(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN)
      CALL CONE(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN)
      CALL CAVSUB(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN,
     &             EXITCAV,EXCAV)
c
c some special cases:
C
C FOR THE FIRST SHOT THE BUNDLE CANNOT LEAVE THE MODULE IF IT WAS
C SHOT FROM THE APERTURE OF THE FOV LIMITER.
C
      IF (EMISUR.EQ.0) THEN
          IF (Z2(1).GT.ZMAX) THEN
              HITTEN(1)=.FALSE.
              EXIT=.FALSE.
          END IF
          IF (Z2(5).GT.ZMAX) THEN
              HITTEN(5)=.FALSE.
              EXIT=.FALSE.
          END IF
C
C FOR THE FIRST SHOT, THE BUNDLE CANNOT HIT THE CAVITY OR UPPER
C SUBSTRATE IF IT WAS SHOT FROM THE LOWER SUBSTRATE.
C
      ELSE IF (EMISUR.EQ.2) THEN
           HITTEN(2)=.FALSE.
           HITTEN(9)=.FALSE.
           HITTEN(10)=.FALSE.
           HITTEN(11)=.FALSE.
           HITTEN(12)=.FALSE.
           HITTEN(13)=.FALSE.
C
C FOR THE FIRST SHOT, THE BUNDLE CAN NOT GO BELOW Z1 IF IT WAS SHOT
C FROM THE SUBSTRATE
C
      ELSE IF (EMISUR.EQ.3) THEN
           IF (Z2(3).LT.Z1) HITTEN(3)=.FALSE.
           IF (Z2(4).LT.Z1) HITTEN(4)=.FALSE.
           IF (Z2(6).LT.Z1) HITTEN(6)=.FALSE.
           IF (Z2(7).LT.Z1) HITTEN(7)=.FALSE.
           IF (Z2(8).LT.Z1) HITTEN(8)=.FALSE.
           IF (Z2(9).LT.Z1) HITTEN(9)=.FALSE.
           IF (Z2(10).LT.Z1) HITTEN(10)=.FALSE.
           IF (Z2(11).LT.Z1) HITTEN(11)=.FALSE.
           IF (Z2(12).LT.Z1) HITTEN(12)=.FALSE.
           HITTEN(13)=.FALSE.
           HITTEN(14)=.FALSE.
           HITTEN(15)=.FALSE.
C
C FOR THE FIRST SHOT, THE BUNDLE CANNOT GO ABOVE Z1 IF IT WAS SHOT
C FROM THE PORTION OF THE SUBSTRATE WHICH FACES THE CAVITY
C
      ELSE IF (EMISUR.EQ.13) THEN
           HITTEN(1)=.FALSE.
           HITTEN(2)=.FALSE.
           HITTEN(6)=.FALSE.
           HITTEN(3)=.FALSE.
           HITTEN(4)=.FALSE.
```

```
            HITTEN(5)=.FALSE.
            HITTEN(7)=.FALSE.
            HITTEN(8)=.FALSE.
            HITTEN(13)=.FALSE.
            HITTEN(14)=.FALSE.
            HITTEN(15)=.FALSE.
            DO 53 I=1,15
               IF (.NOT.HITTEN(I)) GOTO 53
                  COSDOM = DOTP(X2(I)-X1,Y2(I)-Y1,Z2(I)-Z1,
     &                     ANORM4,ANORM5,ANORM6)
                  IF (COSDOM.LE.0.D0) HITTEN(I)=.FALSE.
53       CONTINUE
C
C FOR THE FIRST SHOT THE BUNDLE CANNOT HIT THE LOWER OR UPPER SUBSTRATE,
C IF IT WAS EMITTED FROM THE CONE
C
         ELSE IF (EMISUR.EQ.11) THEN
            HITTEN(2)=.FALSE.
            HITTEN(6)=.FALSE.
C
C  FOR ENERGY BUNDLES EMITTED FROM THE BARREL CHOOSE WHICH SURFACES
C  CANNOT BE HIT ACCORDING TO THE 2-PI SPHERE INTO WHICH THE
C  BUNDLE IS EMITTED.
C
         ELSE IF (EMISUR.EQ.9) THEN
            HITTEN(2)=.FALSE.
            HITTEN(6)=.FALSE.
            DO 54 I=1,15
               IF (.NOT.HITTEN(I)) GOTO 54
                  COSDOM = DOTP(X2(I)-X1,Y2(I)-Y1,Z2(I)-Z1,
     &                     ANORM1,ANORM2,ANORM3)
                  IF (COSDOM.LE.0.D0) HITTEN(I)=.FALSE.
54          CONTINUE
C
C  FOR ENERGY BUNDLES EMITTED FROM THE RING CHOOSE WHICH SURFACES
C  CANNOT BE HIT ACCORDING TO THE 2-PI SPHERE INTO WHICH THE
C  BUNDLE IS EMITTED.
C
         ELSE IF (EMISUR.EQ.14) THEN
            HITTEN(2)=.FALSE.
            HITTEN(3)=.FALSE.
            HITTEN(4)=.FALSE.
            HITTEN(7)=.FALSE.
            HITTEN(8)=.FALSE.
            HITTEN(13)=.FALSE.
            DO 57 I=1,15
               IF (.NOT.HITTEN(I)) GOTO 57
               IF (I.EQ.6) GOTO 57
                  COSDOM = DOTP(X2(I)-X1,Y2(I)-Y1,Z2(I)-Z1,
     &                     ANORM7,ANORM8,ANORM9)
                  IF (COSDOM.LE.0.D0) THEN
                    HITTEN(I)=.FALSE.
                  END IF
57          CONTINUE
c
c  For an energy bundle emitted by the dome choose which surfaces
c  cannot be hit according to the 2pi-sphere in which the bundle
c  was emitted.
c
         ELSE IF (EMISUR.EQ.4) THEN
            CALL DRNUN(1,RDM)
            CNTRDM = CNTRDM + 1
            IF (RDM.GT.0.5D0) THEN
               DO 55 I=1,15
                  COSDOM = DOTP(X2(I)-X1,Y2(I)-Y1,Z2(I)-Z1,
```

```
     &                        ANORM(1),ANORM(2),ANORM(3))
               IF (COSDOM.GE.0.D0) HITTEN(I)=.FALSE.
55          CONTINUE
         ELSE
            DO 56 I=1,15
               COSDOM = DOTP(X2(I)-X1,Y2(I)-Y1,Z2(I)-Z1,
     &                        ANORM(1),ANORM(2),ANORM(3))
               IF (COSDOM.LT.0.D0) HITTEN(I)=.FALSE.
56          CONTINUE
         END IF
      END IF
C
C THE SURFACE HIT IS THE CLOSEST ONE TO THE EMISSION POINT
C
      SURFAC=0
      MAGMIN=10.D0*RFOV
      DO 60 I=1,15
            IF (HITTEN(I))THEN
                IF (MAGNV(I) .LT. MAGMIN) THEN
                   MAGMIN=MAGNV(I)
                   SURFAC=I
                END IF
****************
      IF (MAGNV(I).EQ.0.D0) THEN
          WRITE(16,*)'ATTENTION USER THE BUNDLE IS TRAPPED'
          WRITE(16,*)'CNTREF',CNTREF
          WRITE(06,*)'ERROR!!!!!!!!!!!!'
          WRITE(16,*)'MGNV ET SURFAC 11111',MAGNV(I),I
c       CALL SHOW (X1,Y1,Z1,I)
      END IF
****************
            END IF
60    CONTINUE
C
C IF THE BUNDLE IS EMITTED ON THE BORDER OF THE IMAGINARY TOP SURFACE
C IT IS CONSIDERED TO BE ABSORBED BY THE FOV LIMITER
C
      IF (SURFAC.EQ.0) THEN
          ABSOR=.TRUE.
          EXIT=.FALSE.
          SURFAC=1
          CNTCCL=CNTCCL+1
C          WRITE(03,*)CNTBDL,CNTCCL
      END IF
C
C STORE THE COORDINATES OF THE POINT HIT
C
      XX=X2(SURFAC)
      YY=Y2(SURFAC)
      ZZ=Z2(SURFAC)
      XTT=XX
      YTT=YY
      ZTT=ZZ
C
C  If the bundle was emitted within the dome check if it was extinguished
C  before it actually reached the other surface
C
      IF (INDOME) THEN
C
C COMPUTE THE DISTANCE P1P2, GET A RANDOM NUMBER, AND COMPUTE THE MAXI
C MUM DISTANCE A BUNDLE CAN TRAVEL BASED ON THE ABSORPTIVITY
C
          LEN=NORME(XX-X1,YY-Y1,ZZ-Z1)
          CALL DRNUN(1,RDM)
          CNTRDM=CNTRDM+1
```

```
          RDMLEN=DLOG(1.D0/(1.D0-RDM))/ALPHAD
C
C IF THE DISTANCE P1P2 IS GREATER THAN THE MAXIMUM DISTANCE RDMLEN THEN
C THE ENERGY BUNDLE WILL BE ABSORBED IN THE DOME AT A DISTANCE "RDMLEN"
C BETWEEN P1(X1,Y1,Z1) AND P2(X2,Y2,Z2). LEAVE THE LOOP INCREASE COUNTER
C
          IF (LEN.GT.RDMLEN) THEN
              ABSOR=.TRUE.
              SURFAC=4
              DELZ=RDMLEN/DSQRT((L/N)*(L/N)+(M/N)*(M/N)+1.D0)*
     &             DABS(ZZ-Z1)/(ZZ-Z1)
              XX=(L/N)*DELZ+X1
              YY=(M/N)*DELZ+Y1
              ZZ=DELZ+Z1
          ELSE IF (SURFAC.EQ.6) THEN
              ABSOR=.TRUE.
          END IF
      END IF

C
C IF THE ENERGY BUNDLE LEAVES THE MODULE, LEAVE THE LOOP AND INCREASE
C THE CORRESPONDING ABSORPTION COUNTER
C
      IF (EXIT.AND.(SURFAC.EQ.1.OR.SURFAC.EQ.5).OR.ABSOR) GO TO 222
C
C AFTER EMISSION THE PROGRAM GOES AROUND THE FOLLOWING LOOP UNTIL
C THE BUNDLE IS ABSORBED
C
C ******************* LOOOOOOOOOOOOOOOOP ************************
C
100   CNTREF=CNTREF+1.0
      ABSOR=.FALSE.
      REFL=.FALSE.
      TRANS=.FALSE.
      DIFUS=.FALSE.
C
CCCCCCCCCCCC
      IF (MAGNV(SURFAC).EQ.0.D0) THEN
          WRITE(16,*)'ATTENTION USER THE BUNDLE IS TRAPPED'
          WRITE(16,*)'CNTREF',CNTREF,JMAX
          WRITE(06,*)'ERROR !!!!!'
          WRITE(16,*)'MGNV ET SURFAC 2222222',MAGNV(SURFAC),SURFAC
      END IF
CCCCCCCCCCCC
C
C
C COMPUTE THE COSINE OF THE ANGLE BETWEEN THE LINE P2P1 AND THE VECTOR
C NORMAL TO THE SURFACE AT P2.
C
      ANORM(1)=INORM(SURFAC,1)
      ANORM(2)=INORM(SURFAC,2)
      ANORM(3)=INORM(SURFAC,3)
      COSOLD=DOTP(XX-X1,YY-Y1,ZZ-Z1,ANORM(1),ANORM(2),ANORM(3))/
     &       MAGNV(SURFAC)
C
C IF THE BUNDLE HITS THE FOV LIMITER, SUBSTRATE, BARREL, CONE, RING,
C OR CAV SUB, IT CAN BE EITHER REFLECTED OR ABSORBED
C
      IF ((SURFAC.EQ.1).OR.(SURFAC.EQ.5).OR.
     &    (SURFAC.EQ.2).OR.(SURFAC.EQ.6).OR.
     &    (SURFAC.EQ.9).OR.(SURFAC.EQ.10).OR.
     &    (SURFAC.EQ.11).OR.(SURFAC.EQ.12).OR.
     &    (SURFAC.EQ.13).OR.
     &    (SURFAC.EQ.14).OR.(SURFAC.EQ.15)) THEN
```

```
        CALL REFOAB(L,M,N,XX,YY,ZZ,INORM,SURFAC,ABSOR,REFL,DIFUS)
        TRANS=.FALSE.
C
C IF THE BUNDLE HIT THE DOME, IT CAN BE TRANSMITTED, REFLECTED OR
C ABSORBED
C
      ELSE IF ((SURFAC.EQ.3).OR.(SURFAC.EQ.4).OR.(SURFAC.EQ.7).OR.
     &         (SURFAC.EQ.8)) THEN
C
C FOR THE OUTER SURFACE OF THE DOME,IF THE Z-COORDINATE IS BELOW ZERO,
C THE RAY CAN NOT GO THRU THE INTERFACE. IT HAS TO BE REFLECTED SPECU-
C LARLY. IT CAN NOT BE TRANSMITTED!!!!!
C
       IF ((SURFAC.EQ.3.OR.SURFAC.EQ.7).AND.Z2(SURFAC).LE.0.D0) THEN
          TRANS=.FALSE.
          CALL REFLEX(L,M,N,INORM,SURFAC)
          REFL=.TRUE.
C
C OTHERWISE THE BUNDLE TRY TRANSMISSION
C
       ELSE
        CALL REFOTR(L,M,N,INORM,SURFAC,REFL,TRANS,INDOME)
C
       END IF
      END IF
C
C IF THE ENERGY BUNDLE WAS ABSORBED BY THE FOV LIMITER OR BY THE SUB
C STRATE, KEEP P1 AND P0 AS THEY WERE GO TO THE END OF THE FOLLOWING
C "IF" AND CHECK FOR POSSIBLE EXTINCTION IN THE DOME. IF NOT ABSORBED
C EXECUTE THE FOLLOWING STEPS:
C
      IF (.NOT.ABSOR) THEN
C
C MEMORIZE P0 AND SET THE FORMER POINT P0 TO BE P1
C
      X0=X1
      Y0=Y1
      Z0=Z1
      X1=XX
      Y1=YY
      Z1=ZZ
C
C FIND THE POSSIBLE INTERSECTIONS WITH EACH SURFACE OF THE MODULE
C
      CALL FOVLIM(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN,EXIT)
      CALL SUBSTR(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN,ENTER)
      IF (CH.EQ.22) THEN
        CALL SPHERE(X1,Y1,Z1,L,M,N,X2,Y2,Z2,3,MAGNV,INORM,HITTEN)
        CALL SPHERE(X1,Y1,Z1,L,M,N,X2,Y2,Z2,4,MAGNV,INORM,HITTEN)
      ELSE
        CALL RING(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN)
      END IF
      CALL BARREL(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN)
      CALL CONE(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN)
      CALL CAVSUB(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN,
     &              EXITCAV,EXCAV)
C
C IF A REFLEXION OCCURED LAST, A SURFACE CAN ONLY BE HIT IF THE COSINE
C OF THE ANGLE OF REFLEXION IS THE OPPOSITE TO THAT OF INCIDENCE
C
      IF (REFL) THEN
          J=0
          DO 150 I=1,15
           IF (HITTEN(I)) THEN
C
```

```
CCCCCCCCCC
      IF (MAGNV(I).EQ.0.D0) THEN
          WRITE(16,*)'ATTENTION USER THE BUNDLE IS TRAPPED'
          WRITE(16,*)'CNTREF',CNTREF,JMAX
          WRITE(06,*)'ERROR !!!'
          WRITE(16,*)'MGNV ET SURFAC 333333',MAGNV(I),I
c         CALL SHOW (X1,Y1,Z1,I)
      END IF
CCCCCCCCCCCC
C
              COSNEW(I)=DOTP(X2(I)-X1,Y2(I)-Y1,Z2(I)-Z1,ANORM(1),ANORM
     &        (2),ANORM(3))/MAGNV(I)
              IF (-COSOLD*COSNEW(I) .GT. 0.D0) THEN
                  J=J+1
                  BOOL(J)=I
              END IF
           END IF
150       CONTINUE
        JMAX=J
C
C IF NO SURFAC IS HIT, THE USER IS WARNED SINCE THE PROGRAM CAN NOT
C PERSUE EXECUTION
CCCCCCCCCCCCCC
      IF (JMAX.EQ.0) THEN
          WRITE(16,*)'NO JMAX REFL',JMAX
          DO 165 I=1,15
           IF (HITTEN(I)) THEN
              WRITE(16,*)'COSTTA PRECEDENT ',-COSOLD
              WRITE(16,*)'COSTTA ACTUEL ',COSNEW(I),I
           END IF
165       CONTINUE
          GO TO 222
      END IF
CCCCCCCCCCCC
C
C
C BETWEEN THE ELIGIBLE SURFACES CHOOSE THE CLOSEST ONE
C
          IF (JMAX .GT. 1) THEN
              MAGMAX=MAGNV(BOOL(1))
              SURFAC=BOOL(1)
              DO 170 I=2,JMAX
                  IF (MAGNV(BOOL(I)).LT.MAGMAX) THEN
                   MAGMAX=MAGNV(BOOL(I))
                   SURFAC=BOOL(I)
                  END IF
170           CONTINUE
          ELSE
              SURFAC=BOOL(JMAX)
          END IF
C
C IF A TRANSMISSION OCCURED LAST, A SURFACE CAN ONLY BE HIT IF THE
C COSINES OF THE ANGLES ARE EQUAL.
C
      ELSE IF (TRANS) THEN
          J=0
          DO 190 I=1,15
           IF (HITTEN(I)) THEN
C
CCCCCCCCCCCC
      IF (MAGNV(I).EQ.0.D0) THEN
          WRITE(16,*)'ATTENTION USER THE BUNDLE IS TRAPPED'
          WRITE(16,*)'CNTREF AND JMAX',CNTREF,JMAX
          WRITE(06,*)'ERROR!!!!! '
          WRITE(16,*)'MGNV ET SURFAC',MAGNV(I),I
```

```
          CALL SHOW (X1,Y1,Z1,I)
      END IF
CCCCCCCCCCCC
C
              COSNEW(I)=DOTP(X2(I)-X1,Y2(I)-Y1,Z2(I)-Z1,ANORM(1),ANORM
     &        (2),ANORM(3))/MAGNV(I)
              IF (COSOLD*COSNEW(I) .GT. 0.D0) THEN
                  J=J+1
                  BOOL(J)=I
              END IF
           END IF
190       CONTINUE
          JMAX=J
C
C IF NO SURFAC IS HIT, THE USER IS WARNED SINCE THE PROGRAM CAN NOT
C PERSUE EXECUTION
CCCCCCCCCCCC
      IF (JMAX.EQ.0) THEN
          WRITE(16,*)' JMAX TRANS',JMAX
          CALL SHOW(X0,Y0,Z0,25)
          CALL SHOW(X1,Y1,Z1,SURFAC)
          DO 200 I=1,15
           IF (HITTEN(I)) THEN
              WRITE(16,*)'COSTTA PRECEDENT ',-COSOLD
              WRITE(16,*)'COSTTA ACTUEL ',COSNEW(I),I
              CALL SHOW(X2(I),Y2(I),Z2(I),I)
           END IF
200       CONTINUE
          GO TO 222
      END IF
CCCCCCCCCCCC
C
C
C BETWEEN THE ELIGIBLE SURFACES CHOOSE THE CLOSEST ONE
C
          IF (JMAX .GT. 1) THEN
              MAGMAX=MAGNV(BOOL(1))
              SURFAC=BOOL(1)
              DO 205 I=2,JMAX
                  IF (MAGNV(BOOL(I)).LT.MAGMAX) THEN
                   MAGMAX=MAGNV(BOOL(I))
                   SURFAC=BOOL(I)
                  END IF
205           CONTINUE
          ELSE
              SURFAC=BOOL(JMAX)
          END IF
C  CALL SHOW(X2(SURFAC),Y2(SURFAC),Z2(SURFAC),SURFAC)
C
C IF THE BUNDLE WAS NEITHER REFLECTED NOR TRANSMITTED: PROBLEM!
C
C IF DIFFUSE REFLECTION OCCURED TREAT THE BUNDLE AS IN DIFFUSE EMISSION
C
      ELSE IF (DIFUS) THEN
          IF (SURFAC.EQ.2) THEN
              IF (Z2(3).LT.0.0D0) HITTEN(3)=.FALSE.
              IF (Z2(4).LT.0.0D0) HITTEN(4)=.FALSE.
              IF (Z2(6).LT.0.0D0) HITTEN(6)=.FALSE.
              IF (Z2(7).LT.0.0D0) HITTEN(7)=.FALSE.
              IF (Z2(8).LT.0.0D0) HITTEN(8)=.FALSE.
              IF (Z2(9).LT.0.0D0) HITTEN(9)=.FALSE.
              IF (Z2(10).LT.0.0D0) HITTEN(10)=.FALSE.
              IF (Z2(11).LT.0.0D0) HITTEN(11)=.FALSE.
              IF (Z2(12).LT.0.0D0) HITTEN(12)=.FALSE.
              IF (Z2(13).LT.0.0D0) HITTEN(13)=.FALSE.
```

```
              IF (Z2(14).LT.0.0D0) HITTEN(14)=.FALSE.
              IF (Z2(15).LT.0.0D0) HITTEN(15)=.FALSE.
          END IF
C
C THE SURFACE HIT IS THE CLOSEST ONE TO THE EMISSION POINT
C
      MAGMIN=10.D0*RFOV
      J=0
      DO 210 I=1,15
        IF (HITTEN(I))THEN
          IF (MAGNV(I) .LT. MAGMIN) THEN
              MAGMIN=MAGNV(I)
              SURFAC=I
              J=J+1
          END IF
*
****************
      IF (MAGNV(I).EQ.0.D0) THEN
          WRITE(16,*)'ATTENTION USER THE BUNDLE IS TRAPPED'
          WRITE(16,*)'CNTREF',CNTREF
          WRITE(06,*)'ERROR!!!!!!!!!!!!'
          WRITE(16,*)'MGNV ET SURFAC 11111',MAGNV(I),I
          CALL SHOW (X1,Y1,Z1,I)
      END IF
****************
*
        END IF
210     CONTINUE
        IF (J.EQ.0) THEN
          WRITE(16,*)'GROS CACA DANS IF DIFUS'
          GO TO 222
        END IF
      ELSE
          WRITE(06,*)'CA MERDE'
          GO TO 260
      END IF
C
C STORE THE COORDINATES OF THE POINT HIT
C
      XX=X2(SURFAC)
      YY=Y2(SURFAC)
      ZZ=Z2(SURFAC)
C
C IF THE ENERGY BDLE LEAVES THE MODULE LEAVE
C THE LOOP AND INCREASE THE CORRESPONDING ABSORPTION COUNTER
C
      IF (EXIT.AND.(SURFAC.EQ.1.OR.SURFAC.EQ.5)) GOTO 222
C
C END OF THE (NOT ABSOR) IF
C
      END IF
C
CCCCCCCCCCCCC
      IF (ABS(Y2(SURFAC)).GT.RFOV.OR.ABS(X2(SURFAC)).GT.RFOV) THEN
          CALL SHOW(X2(SURFAC),Y2(SURFAC),Z2(SURFAC),SURFAC)
          CALL SHOW(X1,Y1,Z1,SURFAC)
          DO 211 I=1,15
              WRITE(06,*)HITTEN(I)
211       CONTINUE
          WRITE(06,*)'ERREUR QUELQUE PART',CNTBDL
          GO TO 260
      END IF
CCCCCCCCCCCCC
C
```

```
C
C IF THE BUNDLE IS IN THE DOME CHECK IF IT WAS EXTINGUISHED BEFORE
C IT ACTUALLY REACHED THE OTHER SURFACE
C
      IF (INDOME) THEN
C
C COMPUTE THE DISTANCE P1P2, GET A RANDOM NUMBER, AND COMPUTE THE MAXI
C MUM DISTANCE A BUNDLE CAN TRAVEL BASED ON THE ABSORPTIVITY
C
          LEN=NORME(XX-X1,YY-Y1,ZZ-Z1)
          CALL DRNUN(1,RDM)
          CNTRDM=CNTRDM+1
          RDMLEN=DLOG(1.D0/(1.D0-RDM))/ALPHAD
C
C IF THE DISTANCE P1P2 IS GREATER THAN THE MAXIMUM DISTANCE RDMLEN THEN
C THE ENERGY BUNDLE WILL BE ABSORBED IN THE DOME AT A DISTANCE "RDMLEN"
C BETWEEN P1(X1,Y1,Z1) AND P2(X2,Y2,Z2). LEAVE THE LOOP INCREASE COUNTER
C
          IF (LEN.GT.RDMLEN) THEN
              ABSOR=.TRUE.
              SURFAC=4
              DELZ=RDMLEN/DSQRT((L/N)*(L/N)+(M/N)*(M/N)+1.D0)*
     &             DABS(ZZ-Z1)/(ZZ-Z1)
              XX=(L/N)*DELZ+X1
              YY=(M/N)*DELZ+Y1
              ZZ=DELZ+Z1
          ELSE IF (SURFAC.EQ.6) THEN
              ABSOR=.TRUE.
          END IF
      END IF
C
C IF THE BUNDLE WAS NOT ABSORBED THIS TIME GO BACK TO TOP OF LOOP
C
220   IF (.NOT.ABSOR) GO TO 100
C
C TEST SECTION TO INCREASE THE COUNTERS
C
222   IF((SURFAC.EQ.1.OR.SURFAC.EQ.5).AND.(ABSOR.AND..NOT.EXIT)) THEN
          FT(1)=FT(1)+1
          CALL SEARCH(XX,YY,ZZ,SURFAC,CNTBDL,DIST)
      ELSE IF (SURFAC.EQ.2.AND.ABSOR) THEN
          FT(12)=FT(12)+1
          CALL SEARCH(XX,YY,ZZ,SURFAC,CNTBDL,DIST)
      ELSE IF ((SURFAC.EQ.6).AND.(ABSOR.AND..NOT.ENTER)) THEN
          FT(2)=FT(2)+1
          CALL SEARCH(XX,YY,ZZ,SURFAC,CNTBDL,DIST)
      ELSE IF ((SURFAC.EQ.3.OR.SURFAC.EQ.7).AND.ABSOR) THEN
          FT(3)=FT(3)+1
          CALL SEARCH(XX,YY,ZZ,SURFAC,CNTBDL,DIST)
      ELSE IF ((SURFAC.EQ.4.OR.SURFAC.EQ.8).AND.ABSOR) THEN
          FT(4)=FT(4)+1
          CALL SEARCH(XX,YY,ZZ,SURFAC,CNTBDL,DIST)
      ELSE IF ((SURFAC.EQ.1.OR.SURFAC.EQ.5).AND.EXIT) THEN
          FT(6)=FT(6)+1
          NBBUAB(NB4+1)=NBBUAB(NB4+1)+1
          ABSOR=.TRUE.
      ELSE IF ((SURFAC.EQ.9.OR.SURFAC.EQ.10).AND.ABSOR) THEN
          FT(7)=FT(7)+1
          CALL SEARCH(XX,YY,ZZ,SURFAC,CNTBDL,DIST)
      ELSE IF ((SURFAC.EQ.11.OR.SURFAC.EQ.12).AND.ABSOR) THEN
          FT(8)=FT(8)+1
          CALL SEARCH(XX,YY,ZZ,SURFAC,CNTBDL,DIST)
      ELSE IF (SURFAC.EQ.13.AND.ABSOR) THEN
          FT(9)=FT(9)+1
          CALL SEARCH(XX,YY,ZZ,SURFAC,CNTBDL,DIST)
```

```
      ELSE IF ((SURFAC.EQ.14.OR.SURFAC.EQ.15).AND.ABSOR) THEN
          FT(10)=FT(10)+1
          CALL SEARCH(XX,YY,ZZ,SURFAC,CNTBDL,DIST)
      ELSE
      END IF
C
C INCREMENT COUNTER FOR BUNDLES ENTERING CAVITY
C
      IF (EMISUR.EQ.4.AND.ENTER) THEN
          FT(5)=FT(5)+1
      END IF
C
C INCREMENT COUNTER FOR BUNDLES LEAVING CAVITY
C
      IF ((EMISUR.EQ.9.OR.EMISUR.EQ.11).AND.EXITCAV) THEN
         FT(11)=FT(11)+1
      END IF
C
C CHECK IF ALL THE BUNDLES HAVE BEEN EMITTED FOR THE CURRENT WL INTERVAL
C
      IF (CNTBDL.LT.NBDLIN(CNTINT)) GO TO 30
C
C COMPUTE THE DISTRIBUTION FACTORS:
C  DFAC(EMITTING ELMT,ABSORBING ELMT,SPECIFIC WAVELENGTH INTERVAL),
C DEFINED AS THE RATIO OF THE NUMBER OF BUNDLES EMITTED BY ELMT "I"
C IN WL INTERVAL "K" THAT IS ABSORBED BY ELMT "J" TO THE NUMBER OF
C BUNDLES EMITTED BY ELMT "I" IN WL INTERVAL "K"
C
      DO 300 J=1,NB14
          DFAC(NBSURF,J,CNTINT)=DBLE(NBBUAB(J))/DBLE(NBBUEA(CNTINT))
300   CONTINUE
C
C CHECK IF ALL THE BUNDLES HAVE BEEN EMITTED FOR THE ENTIRE SPECTRUM
C
      IF (CNTBDL.LT.MAXBDL) GO TO 29
C
C If all energy bundles have been emitted write all the necessary
C OUTPUT FILES.
C
C If energy bundles were emitted from one element only,
C
      IF (MENU3.EQ.2) THEN
      DT(1)=DBLE(FT(1))/DBLE(MAXBDL)
      DT(2)=DBLE(FT(2))/DBLE(MAXBDL)
      DT(3)=DBLE(FT(3))/DBLE(MAXBDL)
      DT(4)=DBLE(FT(4))/DBLE(MAXBDL)
      DT(5)=DBLE(FT(5))/DBLE(MAXBDL)
      DT(6)=DBLE(FT(6))/DBLE(MAXBDL)
      DT(7)=DBLE(FT(7))/DBLE(MAXBDL)
      DT(8)=DBLE(FT(8))/DBLE(MAXBDL)
      DT(9)=DBLE(FT(9))/DBLE(MAXBDL)
      DT(10)=DBLE(FT(10))/DBLE(MAXBDL)
      DT(11)=DBLE(FT(11))/DBLE(MAXBDL)
      DT(12)=DBLE(FT(12))/DBLE(MAXBDL)
      WRITE(03,231)'================================================'
      WRITE(03,232)'NUMBER OF ENERGY BUNDLES EMITTED :',CNTBDL
      WRITE(03,232)'NUMBER OF RANDOM NUMBER USED     :',CNTRDM
      WRITE(03,231)'================================================'
      WRITE(03,230)'DFAC TO FOV LIMITER (DT1)            ',DT(1)
      WRITE(03,230)'DFAC TO UPPER SUBSTRATE (DT12)       ',DT(12)
      WRITE(03,230)'DFAC TO LOWER SUBSTRATE (DT2)        ',DT(2)
      WRITE(03,230)'DFAC TO DOME FROM BELOW (DT3)        ',DT(3)
      WRITE(03,230)'DFAC TO DOME FROM ABOVE (DT4)        ',DT(4)
      WRITE(03,230)'DFAC TO FOV-LIMITER APERTURE (DT6) ',DT(6)
      WRITE(03,230)'DFAC TO BARREL (DT7)                 ',DT(7)
```

```
      WRITE(03,230)'DFAC TO CONE (DT8)                  ',DT(8)
      WRITE(03,230)'DFAC TO CAVITY SUBSTRATE (DT9)      ',DT(9)
      WRITE(03,230)'DFAC TO RING (DT10)                 ',DT(10)
      WRITE(03,*)  '                                    =========='
      WRITE(03,230)'SUM OF DFACS                        ',DT(1)+DT(2)
     &              +DT(3)+DT(4)+DT(6)+DT(7)+DT(8)+DT(9)+DT(10)+DT(12)
      WRITE(03,231)'=========================================='
      WRITE(03,230)'POSSIBLE BUNDLES ENTERING CAVITY   ',DT(6)
      WRITE(03,230)'BUNDLES LEAVING CAVITY              ',DT(11)
      WRITE(03,230)'AVERAGE # OF REFLECTIONS PER BUNDLE',CNTREF/CNTBDL
      WRITE(03,232)'NUMBER OF BUNDLES REFLECTED ON DOME',CNTBRW
C
230   FORMAT(2X,A36,F10.5)
231   FORMAT(1X,A48)
232   FORMAT(2X,A34,I10)
240   FORMAT(1X,2I10,F15.9)
C
C Compute the DFAC for shortwave, longwave and total spectrum
C  and output the DFAC to file for tecplot postprocessing
C
      I = 1
      DO 310 J=1,NB14
          DFACSW(I,J)=0.0D0
          DFACLW(I,J)=0.0D0
          DFACTT(I,J)=0.0D0
310   CONTINUE
      DO 311 J=1,NB14
      DO 311 K=1,NBWLIN-1
          DFACSW(I,J)=(DFACSW(I,J)+DFAC(NBSURF,J,K))
          DFACTT(I,J)=(DFACTT(I,J)+DFAC(NBSURF,J,K))
311   CONTINUE
      checkdf=0.0D0
      K=NBWLIN
      DO 312 J=1,NB14
          DFACLW(I,J)=(DFACLW(I,J)+DFAC(NBSURF,J,K))
          DFACTT(I,J)=(DFACTT(I,J)+DFAC(NBSURF,J,K))
          write(18,*)'DFACTT TO ELMENT',J,'=',DFACTT(I,J)
          checkdf=checkdf+dfactt(i,j)
312   CONTINUE
      write(18,*)'checkdf=',checkdf
      CALL TECPLOT
      END IF
C
C Output of the distribution factors to file "dfacmtx.dat"
C
      IF (EMISUR.EQ.0) WRITE(07,770)'FIELD',NBSURF
      IF (EMISUR.EQ.1) WRITE(07,770)'FOV_L',NBSURF
      IF (EMISUR.EQ.2) WRITE(07,770)'L_SUB',NBSURF
      IF (EMISUR.EQ.3) WRITE(07,770)'U_SUB',NBSURF
      IF (EMISUR.EQ.4) WRITE(07,770)'DOME ',NBSURF
      IF (EMISUR.EQ.9) WRITE(07,770)'BARREL',NBSURF
      IF (EMISUR.EQ.11) WRITE(07,770)'CONE',NBSURF
      IF (EMISUR.EQ.13) WRITE(07,770)'CAVSUB',NBSURF
      IF (EMISUR.EQ.14) WRITE(07,770)'RING',NBSURF
      DO 350 K=1,NBWLIN
      DO 350 J=1,NB14
          WRITE(07,780)J,K,DFAC(NBSURF,J,K)
350   CONTINUE
770   FORMAT(1X,A6,I4)
780   FORMAT(1X,2I6,G30.20)
790   FORMAT(1X,I6,G30.20)
C
C
C END-OF-PROGRAM MENU
C
```

```
260    IF (CNTEMI.LT.CNTIND) GO TO 922
       WRITE(03,*)CNTRDM,' RANDOM NUMBER USED FOR DFAC COMPUTATION'
C
270    WRITE(06,520)'====================================================='
       WRITE(06,520)' TO QUIT,                                ENTER (0) '
       WRITE(06,520)' IF YOU WANT TO RUN THIS PROGRAM AGAIN,  ENTER (1) '
       WRITE(06,520)'====================================================='
       READ(05,*)ANSWER
       IF (ANSWER.EQ.1) THEN
         GO TO 10
       ELSE
         GO TO 261
       END IF
C
C FORMATS OF ALL WRITE AND READ STATEMENTS
C
520    FORMAT(1X,A52)
521    FORMAT(1X,F10.3,A11,F10.3)
522    FORMAT(1X,A30,F10.6)
523    FORMAT(1X,A10,I5)
525    FORMAT(1X,A25,I4,A6,I4)
526    format(1x,a16,i4,a22,i4)
700    FORMAT(1X,F9.3,1X,F9.3,1X,F9.3,1X,F15.9)
710    FORMAT(1X,4I8)
261    END

*****************************************************************
*****************END OF MAIN PROGRAMME***************************
*****************************************************************

       SUBROUTINE EMIT0 (X1,Y1,Z1,L,M,N,CRADAN,PTSOUR)
       IMPLICIT NONE
       REAL*8 X1,Y1,Z1,PHI1,TETA1,TETA,PI,RDM,RFOV,TETAFM
       REAL*8 R,L,M,N,LP,MP,NP,RADIUS,CRADAN
       REAL*8 XPTSO,YPTSO
       REAL*8 REXT,RINT,DENIV
C
       INTEGER CNTRDM
C
       LOGICAL PTSOUR
C
       COMMON /RAND/ CNTRDM
       COMMON /FOVL/ RFOV,TETAFM
       COMMON /PTSO/ XPTSO,YPTSO
       COMMON /DOME/ REXT,RINT,DENIV
C
       PI=DACOS(-1.D0)
C
C WHEN EMITTING FROM THE APERTURE OF THE FOV LIMITER Z1 IS GIVEN
C
       Z1=RFOV*DCOS(TETAFM)+DENIV

C
C IF POINT SOURCE OPTION IS SET X1=XPTSO AND Y1=YPTSO
C
       IF (PTSOUR) THEN
           X1=XPTSO
           Y1=YPTSO
       ELSE
C
C GET RANDOM POSITION ON EMISSION DISC *
C
       CALL DRNUN(1,RDM)
       CNTRDM=CNTRDM+1
       R=DSQRT(RDM)*RFOV*DSIN(TETAFM)
```

```
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TETA=RDM*2.D0*PI
C
C OBTAIN X Y COORDINATES *
C
      X1=R*DCOS(TETA)
      Y1=R*DSIN(TETA)
      RADIUS=DSQRT(X1*X1+Y1*Y1)
      END IF
C
C IF THE ANGLE CRADAN IS >0, THE RADIATION IS EMITTED WITH CST DIRECTION
C
      IF (CRADAN.GE.0.D0) THEN
          L=0.D0
          M=DSIN(CRADAN)
          N=DCOS(CRADAN)
      ELSE
C
C GET RANDOM EMISSION DIRECTION IN THE LOCAL COORDINATE SYSTEM
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      PHI1=DASIN(DSQRT(RDM))
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TETA1=2.D0*PI*RDM
C
C COMPUTE THE DIRECTION COSINES IN THE LOCAL COORDINATE SYSTEM
C
      LP=DSIN(PHI1)*DCOS(TETA1)
      MP=DSIN(PHI1)*DSIN(TETA1)
      NP=DCOS(PHI1)
C
C TRANSFORMATION OF THE DIRECTION COSINES INTO GLOBAL COORD. SYSTEM
C
      L=LP
      M=MP
      N=NP
      END IF
      END

************************************************************************
*                                                                      *
*   THIS SUBROUTINE EMITS BUNDLES IN RANDOM DIRECTION FROM A RANDOM    *
*   LOCATION OF A SURFACE ELEMENT OF THE FOV LIMITER.                  *
*                                                                      *
************************************************************************

      SUBROUTINE EMIT1 (X1,Y1,Z1,L,M,N)

      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,PHI,TETA,L,M,N,LP,MP,NP,ZED,DZED,DTETA
      REAL*8 RFOV,TETAFM,ZMAX,PI,RDM,PHILOC,TTALOC
      REAL*8 REXT,RINT,DENIV
C
      INTEGER NBLAT,NBLONG,NSURI1,NSURI2,NSURI3
      INTEGER CNTRDM
C
      COMMON /RAND/ CNTRDM
      COMMON /FOVL/ RFOV,TETAFM
      COMMON /SER2/ NBLAT,NBLONG
      COMMON /NSUR/ NSURI1,NSURI2,NSURI3
      COMMON /DOME/ REXT,RINT,DENIV
```

```
      PI=DACOS(-1.D0)
      ZMAX=RFOV*DCOS(TETAFM)+DENIV

      DZED=ZMAX/NBLAT
      DTETA=2.D0*PI/NBLONG
C
C GET RANDOM POSITION ON EMISSION SURFAC
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      ZED=(DBLE(NSURI1)-1.D0+DSQRT(RDM))*DZED-DENIV

      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TETA=(DBLE(NSURI2)-1.D0+RDM)*DTETA
      PHI=DACOS(ZED/RFOV)
C
C OBTAIN X Y Z COORDINATES *
C
      X1=RFOV*DSIN(PHI)*DCOS(TETA)
      Y1=RFOV*DSIN(PHI)*DSIN(TETA)
      Z1=RFOV*DCOS(PHI)+DENIV

C
C RANDOM EMISSION DIRECTION IN THE LOCAL COORD. SYSTEM
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      PHILOC=DASIN(DSQRT(RDM))
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TTALOC=2.D0*PI*RDM
C
C COMPUTE THE SLOPE OF THE EMITTED RAY IN THE LOCAL COORD. SYSTEM
C
      LP=DSIN(PHILOC)*DCOS(TTALOC)
      MP=DSIN(PHILOC)*DSIN(TTALOC)
      NP=DCOS(PHILOC)
C
C TRANSFORMATION OF THE DIRECTION COSINES INTO GLOBAL COORD. SYSTEM
C
      L=-(X1*Z1/(RFOV*DSQRT(RFOV*RFOV-Z1*Z1))*LP
     &    +Y1/DSQRT(RFOV*RFOV-Z1*Z1)*MP
     &    +X1/RFOV*NP)
      M=-(Y1*Z1/(RFOV*DSQRT(RFOV*RFOV-Z1*Z1))*LP
     &    -X1/DSQRT(RFOV*RFOV-Z1*Z1)*MP
     &    +Y1/RFOV*NP)
      N=DSQRT(RFOV*RFOV-Z1*Z1)/RFOV*LP-Z1/RFOV*NP
      END

***********************************************************************
*                                                                     *
*   THIS SUBROUTINE EMITS BUNDLES IN RANDOM DIRECTION FROM A RANDOM   *
*   LOCATION OF A SURFACE ELEMENT OF THE LOWER SUBSTRATE.             *
*                                                                     *
***********************************************************************

      SUBROUTINE EMIT2 (X1,Y1,Z1,L,M,N,INDOME)


      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,TETA,L,M,N,LP,MP,NP,DTETA
      REAL*8 RFOV,TETAFM,PI,RDM,PHILOC,TTALOC
      REAL*8 RADIUS,REXT,RINT,RPA,DENIV
```

```
C
      INTEGER NBWASH,NBPS,NBLSU,NBWLSU,NSURI1,NSURI2,NSURI3,SURFAC
      INTEGER CNTRDM
C
      LOGICAL INDOME
C
      COMMON /RAND/ CNTRDM
      COMMON /FOVL/ RFOV,TETAFM /SUBS/ RPA,SURFAC
      COMMON /SER3/ NBWASH,NBPS,NBLSU,NBWLSU
      COMMON /DOME/ REXT,RINT,DENIV
      COMMON /NSUR/ NSURI1,NSURI2,NSURI3
C
      PI=DACOS(-1.D0)
      INDOME=.FALSE.
C
C GET RANDOM POSITION ON EMISSION SURFACE (LOWER SUBSTRATE)
C
      IF (NSURI1.EQ.1) THEN
          DTETA=2.D0*PI
          CALL DRNUN(1,RDM)
          CNTRDM=CNTRDM+1
          RADIUS=DSQRT(RDM)*RPA
          CALL DRNUN(1,RDM)
          CNTRDM=CNTRDM+1
          TETA=(DBLE(NSURI2)-1.D0+RDM)*DTETA
      ELSE
          DTETA=2.D0*PI/DBLE(NBLSU)
          CALL DRNUN(1,RDM)
          CNTRDM=CNTRDM+1
          IF (NSURI2.EQ.1) THEN
             RADIUS=DSQRT(RDM*(RINT*RINT-RPA*RPA)+RPA*RPA)
          ELSE
             RADIUS=DSQRT(RDM*(REXT*REXT-RINT*RINT)+RINT*RINT)
             INDOME=.TRUE.
          END IF
          CALL DRNUN(1,RDM)
          CNTRDM=CNTRDM+1
          TETA=(DBLE(NSURI3)-1.D0+RDM)*DTETA
      END IF
C
C OBTAIN X Y Z COORDINATES *
C
      X1=RADIUS*DCOS(TETA)
      Y1=RADIUS*DSIN(TETA)
      Z1=DENIV
C
C RANDOM EMISSION DIRECTION IN THE LOCAL COORD. SYSTEM
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      PHILOC=DASIN(DSQRT(RDM))
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TTALOC=2.D0*PI*RDM
C
C COMPUTE THE SLOPE OF THE EMITTED RAY IN THE LOCAL COORD. SYSTEM
C
      LP=DSIN(PHILOC)*DCOS(TTALOC)
      MP=DSIN(PHILOC)*DSIN(TTALOC)
      NP=DCOS(PHILOC)
C
C TRANSFORMATION OF THE DIRECTION COSINES INTO GLOBAL COORD. SYSTEM
C
      L=(X1*LP-Y1*MP)/RADIUS
      M=(Y1*LP+X1*MP)/RADIUS
```

```
      N=NP
      END
***********************************************************************
*                                                                     *
*   THIS SUBROUTINE EMITS BUNDLES IN RANDOM DIRECTION FROM A RANDOM   *
*   LOCATION OF A SURFACE ELEMENT OF THE UPPER SUBSTRATE.             *
*                                                                     *
***********************************************************************

      SUBROUTINE EMIT3 (X1,Y1,Z1,L,M,N)

      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,TETA,L,M,N,LP,MP,NP,DTETA
      REAL*8 RFOV,TETAFM,PI,RDM,PHILOC,TTALOC
      REAL*8 RADIUS,RADIUS1,RADIUS2,REXT,RINT,DENIV,RAD1,RAD2
C
      INTEGER NBWASH,NBPS,NBLSU,NBWLSU,NSURI1,NSURI2,NSURI3
      INTEGER CNTRDM
C
      COMMON /RAND/ CNTRDM
      COMMON /FOVL/ RFOV,TETAFM
      COMMON /SER3/ NBWASH,NBPS,NBLSU,NBWLSU
      COMMON /NSUR/ NSURI1,NSURI2,NSURI3
      COMMON /DOME/ REXT,RINT,DENIV
C
      PI=DACOS(-1.D0)
      DTETA=2.D0*PI/NBPS
      RAD1 = DSQRT(REXT*REXT - DENIV*DENIV)
      RAD2 = DSQRT(RFOV*RFOV - DENIV*DENIV)
      RADIUS1=DSQRT(DBLE(NSURI1-1)/DBLE(NBWASH)*(RAD2*RAD2-
     &  RAD1*RAD1) + RAD1*RAD1)
      RADIUS2=DSQRT(DBLE(NSURI1)/DBLE(NBWASH)*(RAD2*RAD2-
     &  RAD1*RAD1) + RAD1*RAD1)
C
C GET RANDOM POSITION ON EMISSION SURFAC
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      RADIUS=DSQRT(RDM*(RADIUS2*RADIUS2-RADIUS1*RADIUS1)+RADIUS1
     &        *RADIUS1)
      IF (RADIUS.EQ.0.D0) THEN
          WRITE(16,*)'RADIUS1,RADIUS2,RADIUS,NSURI1,NSURI2'
          WRITE(16,*)RADIUS1,RADIUS2,RADIUS,NSURI1,NSURI2
      END IF
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TETA=(DBLE(NSURI2)-1.D0+RDM)*DTETA
C
C OBTAIN X Y Z COORDINATES
C
      X1=RADIUS*DCOS(TETA)
      Y1=RADIUS*DSIN(TETA)
      Z1=0.D0
C
C RANDOM EMISSION DIRECTION IN THE LOCAL COORD. SYSTEM
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      PHILOC=DASIN(DSQRT(RDM))
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TTALOC=2.D0*PI*RDM
C
C COMPUTE THE SLOPE OF THE EMITTED RAY IN THE LOCAL COORD. SYSTEM
C
```

```
      LP=DSIN(PHILOC)*DCOS(TTALOC)
      MP=DSIN(PHILOC)*DSIN(TTALOC)
      NP=DCOS(PHILOC)
C
C TRANSFORMATION OF THE DIRECTION COSINES INTO GLOBAL COORD. SYSTEM
C
      L=(X1*LP-Y1*MP)/RADIUS
      M=(Y1*LP+X1*MP)/RADIUS
      N=NP
      END
***********************************************************************
*                                                                     *
*     This subroutine emits bundles in a random direction from the    *
*     centroid of a volume element in the DOME                        *
*                                                                     *
***********************************************************************

      SUBROUTINE EMIT4 (X1,Y1,Z1,L,M,N,ANORM)

      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,L,M,N,LP,MP,NP
      REAL*8 DENIV,REXT,RINT
      REAL*8 PHI,TETA,PHILOC,TTALOC,RDM,PI,DISTTOC
      REAL*8 RADF,PHIF,TTAF,SAREA,RCEN
      REAL*8 RADP1,PHIP1,TTAP1
      REAL*8 ANORM(15)
C
      INTEGER NBSH,NBCO,NBPSD,NSURI1,NSURI2,NSURI3
      INTEGER CNTRDM,NBVELT,MAXELT
C
      PARAMETER (MAXELT=700)
C
      COMMON /SER4/ NBSH,NBCO,NBPSD
      COMMON /NSUR/ NSURI1,NSURI2,NSURI3
      COMMON /DOME/ REXT,RINT,DENIV
      COMMON /SETU/ RADF(MAXELT),PHIF(MAXELT),TTAF(MAXELT),SAREA(MAXELT),RCEN
      COMMON /RAND/ CNTRDM
C
      PI = DACOS(-1.D0)
C
C  Get emission location. point P1(X1,Y1,Z1)
C
      CALL DRNUN(1,RDM)
      CNTRDM = CNTRDM + 1
      RADP1 = (((RADF(NSURI1))**3)*(1.D0-RDM) +
     &         ((RADF(NSURI1-1))**3)*RDM)**(1.D0/3.d0)
      CALL DRNUN(1,RDM)
      CNTRDM = CNTRDM + 1
      PHIP1 = DACOS(DCOS(PHIF(NSURI2))*(1.D0-RDM) +
     &               DCOS(PHIF(NSURI2-1))*RDM)
      CALL DRNUN(1,RDM)
      CNTRDM = CNTRDM + 1
      TTAP1 = TTAF(NSURI3)*(1.D0-RDM) + TTAF(NSURI3+1)*RDM
         X1 = RADP1*DSIN(PHIP1)*DCOS(TTAP1)
      Y1 = RADP1*DSIN(PHIP1)*DSIN(TTAP1)
      Z1 = RADP1*DCOS(PHIP1) + DENIv
      DISTTOC = DSQRT(X1*X1 + Y1*Y1 + (Z1-DENIV)*(Z1-DENIV))
C
C  Compute the vector normal to the shell of point P1
C
      ANORM(1) = -X1/RADP1
      ANORM(2) = -Y1/RADP1
      ANORM(3) = -(Z1-DENIV)/RADP1
C
C  Random emission direction in the local coordinate system
```

```
C
      CALL DRNUN(1,RDM)
      CNTRDM = CNTRDM + 1
      PHILOC = DASIN(DSQRT(RDM))
      CALL DRNUN(1,RDM)
      CNTRDM = CNTRDM + 1
      TTALOC = 2.D0*PI*RDM
C
C  Compute the slope of the emitted ray in the local coordinate system
C
      LP = DSIN(PHILOC)*DCOS(TTALOC)
      MP = DSIN(PHILOC)*DSIN(TTALOC)
      NP = DCOS(PHILOC)
C
C  Transformation of the direction cosines into the global coordinate
C  system
C
      L = -(X1*(Z1-DENIV)/(DISTTOC*DSQRT(DISTTOC*DISTTOC - (Z1-DENIV)*
     1    (Z1-DENIV)))*LP + Y1/DSQRT(DISTTOC*DISTTOC - (Z1-DENIV)*
     2    (Z1-DENIV))*MP + X1/DISTTOC*NP)
      M = -(Y1*(Z1-DENIV)/(DISTTOC*DSQRT(DISTTOC*DISTTOC - (Z1-DENIV)*
     1    (Z1-DENIV)))*LP - X1/DSQRT(DISTTOC*DISTTOC - (Z1-DENIV)*
     2    (Z1-DENIV))*MP + Y1/DISTTOC*NP)
      N = DSQRT(DISTTOC*DISTTOC - (Z1-DENIV)*(Z1-DENIV))/DISTTOC*LP -
     1    (Z1-DENIV)/DISTTOC*NP
C
       END
***********************************************************************
*                                                                     *
*            This subroutine emits energy bundles from a random           *
*            location on a given surface element of the barrel            *
*                                                                     *
***********************************************************************

      SUBROUTINE EMIT9(X1,Y1,Z1,L,M,N,ANORM1,ANORM2,ANORM3)
      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,L,M,N,LP,MP,NP
      REAL*8 RCAV,LBAR,DLBAR,HEIGHT,DENIV,LCON,CAVANG
      REAL*8 TETA,DTETA,PI,PHILOC,TTALOC,RDM,REXT,RINT
      REAL*8 ANORM1,ANORM2,ANORM3
C
      INTEGER NBHSB,NBHSC,NBPSB,NBPSC,NSURI1,NSURI2,NSURI3,SURFAC
      INTEGER CNTRDM,NBPSCS
C
      COMMON /RAND/ CNTRDM
      COMMON /NSUR/NSURI1,NSURI2,NSURI3
      COMMON /CAVITY/ RCAV,LBAR,LCON,CAVANG
      COMMON /DOME/ REXT,RINT,DENIV
      COMMON /CAVSUR/ NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS
C
      DLBAR=(LBAR/DBLE(NBHSB))
      PI=DACOS(-1.D0)
      DTETA=2.D0*PI/NBPSB
C
C GET RANDOM POSITION ON EMISSION SURFACE
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      HEIGHT=LBAR-DLBAR*(RDM+DBLE(NSURI1)-1.D0)
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TETA=(DBLE(NSURI2)-1.D0+RDM)*DTETA
C
C OBTAIN X Y Z COORDINATES
C
```

```
      X1=RCAV*DCOS(TETA)
      Y1=RCAV*DSIN(TETA)
      Z1=DENIV-HEIGHT
C
C OBTAIN NORMAL VECTOR FROM EMISSION POINT
C
      ANORM1=-1.D0*X1/RCAV
      ANORM2=-1.D0*Y1/RCAV
      ANORM3=0.0D0

C
C RANDOM EMISSION DIRECTION IN THE LOCAL COORDINATE SYSTEM
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      PHILOC=DASIN(DSQRT(RDM))
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TTALOC=2.D0*PI*RDM
C
C COMPUTE THE SLOPE OF THE EMITTED RAY IN THE LOCAL COORDINATE SYSTEM
C
      LP=DSIN(PHILOC)*DCOS(TTALOC)
      MP=DSIN(PHILOC)*DSIN(TTALOC)
      NP=DCOS(PHILOC)
C
C TRANSFORMATION OF THE DIRECTION COSINES INTO GLOBAL COORDINATE SYSTEM
C
      L=-MP*DSIN(TETA)-NP*DCOS(TETA)
      M= MP*DCOS(TETA)-NP*DSIN(TETA)
      N=LP

      END


************************************************************************
*                                                                      *
*       This subroutine emits energy bundles from a random             *
*       location on a given surface element of the cone                *
*                                                                      *
************************************************************************

      SUBROUTINE EMIT11(X1,Y1,Z1,L,M,N)

      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,L,M,N,LP,MP,NP
      REAL*8 RCAV,LCON,LBAR,DLCON,DENIV,HEIGHT,RPRIME,CAVANG
      REAL*8 TETA,DTETA,PI,PHILOC,TTALOC,RDM,REXT,RINT
      REAL*8 DSNCAV,DCSCAV,LCONP
      REAL*8 RIN,ROUT,RADIUS,LCAV,TANCAV
C
      INTEGER NBHSB,NBHSC,NBPSB,NBPSC,NSURI1,NSURI2,NSURI3,NBPSCS
      INTEGER CNTRDM
C
      COMMON /RAND/ CNTRDM
      COMMON /NSUR/ NSURI1,NSURI2,NSURI3
      COMMON /CAVITY/ RCAV,LBAR,LCON,CAVANG
      COMMON /DOME/ REXT,RINT,DENIV
      COMMON /CAVSUR/ NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS
      COMMON /CONSET/ DLCON(20)
C
      PI=DACOS(-1.D0)
      LCAV=LCON+LBAR-DENIV
      DTETA=2.D0*PI/NBPSC
      DSNCAV=RCAV/(DSQRT(RCAV*RCAV+LCON*LCON))
```

```
      DCSCAV=LCON/(DSQRT(RCAV*RCAV+LCON*LCON))
      TANCAV=RCAV/LCON
C
      ROUT=(DLCON((NBHSC+1)-NSURI1)+LCAV)*TANCAV
      RIN=(DLCON((NBHSC+1)-(NSURI1-1))+LCAV)*TANCAV
C
C GET RANDOM POSITION ON EMISSION SURFACE
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      RADIUS=DSQRT(RDM*(ROUT*ROUT-RIN*RIN)+RIN*RIN)
      HEIGHT=(RADIUS/TANCAV)-LCAV
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TETA=(DBLE(NSURI2)-1.D0+RDM)*DTETA
C
C OBTAIN X Y Z COORDINATES
C
      LCONP=LCON+HEIGHT-DENIV+LBAR
      RPRIME=RCAV*(LCONP/LCON)
      X1=RPRIME*DCOS(TETA)
      Y1=RPRIME*DSIN(TETA)
      Z1=HEIGHT
C
C RANDOM EMISSION DIRECTION IN THE LOCAL COORDINATE SYSTEM
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      PHILOC=DASIN(DSQRT(RDM))
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TTALOC=2.D0*PI*RDM
C
C COMPUTE THE SLOPE OF THE EMITTED RAY IN THE LOCAL COORDINATE SYSTEM
C
      LP=DSIN(PHILOC)*DCOS(TTALOC)
      MP=DSIN(PHILOC)*DSIN(TTALOC)
      NP=DCOS(PHILOC)
C
C TRANSFORMATION OF THE DIRECTION COSINES INTO GLOBAL COORDINATE SYSTEM
C
      L= LP*DCOS(TETA)*DSNCAV-MP*DSIN(TETA)-
     &   NP*DCOS(TETA)*DCSCAV
      M= LP*DSIN(TETA)*DSNCAV+MP*DCOS(TETA)-
     &   NP*DSIN(TETA)*DCSCAV
      N= LP*DCSCAV+NP*DSNCAV

      END

**********************************************************************
*                                                                    *
* THIS SUBROUTINE EMITS BUNDLES IN A RANDOM DIRECTION FROM A RANDOM  *
* LOCATION OF A SURFACE ELEMENT OF THE PORTION OF THE LOWER          *
* SUBSTRATE WHICH FACES THE CAVITY.                                  *
*                                                                    *
**********************************************************************

      SUBROUTINE EMIT13(X1,Y1,Z1,L,M,N,ANORM4,ANORM5,ANORM6)


      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,TETA,L,M,N,LP,MP,NP,DTETA
      REAL*8 RCAV,RPA,PI,RDM,PHILOC,TTALOC,RADIUS
      REAL*8 LBAR,LCON,CAVANG,SURFAC
      REAL*8 TETAFM,RFOV
```

```
      REAL*8 ANORM4,ANORM5,ANORM6
      REAL*8 REXT,RINT,DENIV
C
      INTEGER NSURI1,NSURI2,NSURI3
      INTEGER NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS
      INTEGER CNTRDM
C
      COMMON /RAND/ CNTRDM
      COMMON /CAVSUR/ NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS
      COMMON /CAVITY/ RCAV,LBAR,LCON,CAVANG
      COMMON /FOVL/ RFOV,TETAFM /SUBS/ RPA,SURFAC
      COMMON /DOME/ REXT,RINT,DENIV
C
      PI=DACOS(-1.D0)
C
C GET RANDOM POSITION ON EMISSION SURFACE
C
      DTETA=2.D0*PI/DBLE(NBPSCS)
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      RADIUS=DSQRT(RDM*(RCAV*RCAV-RPA*RPA)+RPA*RPA)
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TETA=(DBLE(NSURI1)-1.D0+RDM)*DTETA
C
C OBTAIN X Y Z COORDINATES
C
      X1=RADIUS*DCOS(TETA)
      Y1=RADIUS*DSIN(TETA)
      Z1=DENIV
C
C OBTAIN NORMAL VECTOR FROM EMISSION POINT
C
      ANORM4=0.0D0
      ANORM5=0.0D0
      ANORM6=-1.0D0
C
C RANDOM EMISSION DIRECTION IN THE LOCAL COORDINATE SYSTEM
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      PHILOC=DASIN(DSQRT(RDM))
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TTALOC=2.D0*PI*RDM
C
C COMPUTE THE SLOPE OF THE EMITTED RAY IN THE LOCAL COORDINATE SYSTEM
C
      LP=DSIN(PHILOC)*DCOS(TTALOC)
      MP=DSIN(PHILOC)*DSIN(TTALOC)
      NP=DCOS(PHILOC)
C
C TRANSFORMATION OF THE DIRECTION COSINES INTO GLOBAL COORD. SYSTEM
C
      L=(Y1*LP+X1*MP)/RADIUS
      M=(X1*LP-Y1*MP)/RADIUS
      N=NP
      END

***********************************************************************
*                                                                     *
*       This subroutine emits energy bundles from a random            *
*       location on a given surface element of the ring               *
*                                                                     *
***********************************************************************
```

```
      SUBROUTINE EMIT14(X1,Y1,Z1,L,M,N,ANORM7,ANORM8,ANORM9)
      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,L,M,N,LP,MP,NP
      REAL*8 DLRING,HEIGHT,DENIV
      REAL*8 TETA,DTETA,PI,PHILOC,TTALOC,RDM,REXT,RINT
      REAL*8 ANORM7,ANORM8,ANORM9
      REAL*8 PHI,DISTTOC,ZED
C
      INTEGER NSURI1,NSURI2,NSURI3,SURFAC,CNTRDM
      INTEGER NBHSR,NBPSR
C
      COMMON /RIING/ NBHSR,NBPSR
      COMMON /RAND/ CNTRDM
      COMMON /NSUR/NSURI1,NSURI2,NSURI3
      COMMON /DOME/ REXT,RINT,DENIV
C
      DLRING=DABS(DENIV/DBLE(NBHSR))
      PI=DACOS(-1.D0)
      DTETA=2.D0*PI/NBPSR
C
C GET RANDOM POSITION ON EMISSION SURFACE
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      ZED=DSQRT(RDM)+DENIV
      PHI=DACOS(-ZED/REXT)
      IF ((PHI.LT.1.412658D0).OR.(PHI.GT.(PI/2.D0))) THEN
        WRITE(6,*)'PHI out of range for RING'
      END IF
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TETA=(DBLE(NSURI2)-1.D0+RDM)*DTETA
C
C OBTAIN X Y Z COORDINATES
C
      X1=REXT*DCOS(TETA)*DSIN(PHI)
      Y1=REXT*DSIN(TETA)*DSIN(PHI)
      Z1=REXT*DCOS(PHI)+DENIV
      DISTTOC=DSQRT(X1*X1+Y1*Y1+(Z1-DENIV)*(Z1-DENIV))
C
C OBTAIN NORMAL VECTOR FROM EMISSION POINT
C
      ANORM7=-X1/REXT
      ANORM8=-Y1/REXT
      ANORM9=-(Z1-DENIV)/REXT
C
C RANDOM EMISSION DIRECTION IN THE LOCAL COORDINATE SYSTEM
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      PHILOC=DASIN(DSQRT(RDM))
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TTALOC=2.D0*PI*RDM
C
C COMPUTE THE SLOPE OF THE EMITTED RAY IN THE LOCAL COORDINATE SYSTEM
C
      LP=DSIN(PHILOC)*DCOS(TTALOC)
      MP=DSIN(PHILOC)*DSIN(TTALOC)
      NP=DCOS(PHILOC)
C
C TRANSFORMATION OF THE DIRECTION COSINES INTO GLOBAL COORDINATE SYSTEM
C
      L = -(X1*(Z1-DENIV)/(DISTTOC*DSQRT(DISTTOC*DISTTOC - (Z1-DENIV)*
     1     (Z1-DENIV)))*LP + Y1/DSQRT(DISTTOC*DISTTOC - (Z1-DENIV)*
```

```
     2    (Z1-DENIV))*MP + X1/DISTTOC*NP)
      M = -(Y1*(Z1-DENIV)/(DISTTOC*DSQRT(DISTTOC*DISTTOC - (Z1-DENIV)*
     1    (Z1-DENIV)))*LP - X1/DSQRT(DISTTOC*DISTTOC - (Z1-DENIV)*
     2    (Z1-DENIV))*MP + Y1/DISTTOC*NP)
      N = DSQRT(DISTTOC*DISTTOC - (Z1-DENIV)*(Z1-DENIV))/DISTTOC*LP -
     1    (Z1-DENIV)/DISTTOC*NP

      END

************************************************************************
*                                                                      *
*  This function computes the norme of a vector                       *
*                                                                      *
************************************************************************

      REAL*8 FUNCTION NORME(A,B,C)
      IMPLICIT NONE
      REAL*8 A,B,C
      NORME=SQRT(A*A+B*B+C*C)
      END

************************************************************************
*                                                                      *
*  This function computes the dot product between two vectors         *
*                                                                      *
************************************************************************

      REAL*8 FUNCTION DOTP(A1,B1,C1,A2,B2,C2)
      IMPLICIT NONE
      REAL*8 A1,B1,C1,A2,B2,C2
      DOTP=A1*A2+B1*B2+C1*C2
      END

*************************************************************************
*                                                                       *
* THIS SUBROUTINE FINDS THE ROOTS OF A SECOND ORDER POLYNOMIAL. THIS    *
* IS USED TO FIND THE Z-COORDINATE OF A POSSIBLE NEW POINT HIT BY A     *
* BUNDLE. THE SOLUTION CAN NOT BE THE PREVIOUS POINT HIT BY THE BUNDLE *
* AND NOT BE OUT OF THE FOV LIMITER IF THE BUNDLE WAS SHOT FROM THE     *
* IMAGINARY TOP SURFACE.                                                *
*                                                                       *
*************************************************************************

      SUBROUTINE POL2 (R,P,Q,SOLPLU,SOLMIN,ROOTS)

      IMPLICIT NONE
      REAL*8 R,P,Q,DELTA,SOLPLU,SOLMIN
      LOGICAL ROOTS
      ROOTS=.TRUE.
      DELTA=P*P-4.D0*R*Q
      IF (DELTA .GT. 0.D0) THEN
          SOLPLU=(-P+DSQRT(DELTA))/(2.D0*R)
          SOLMIN=(-P-DSQRT(DELTA))/(2.D0*R)
      ELSE IF (DELTA .EQ. 0.D0) THEN
          SOLPLU=(-P+DSQRT(DELTA))/(2.D0*R)
          SOLMIN=SOLPLU
      ELSE
          ROOTS=.FALSE.
          SOLPLU=-1.D0
          SOLMIN=-1.D0
      END IF
      END

*************************************************************************
```

```
*                                                                      *
* THIS SUBROUTINE FINDS THE POSSIBLE POINTS THAT WOULD BE AT THE INTER-*
* SECTION OF THE FOV LIMITER SURFACE AND THE LINE REPRESENTING THE DI- *
* RECTION OF THE BUNDLE. THE NUMBER OF POINTS CAN BE 0, 1, OR 2.       *
*                                                                      *
************************************************************************

      SUBROUTINE FOVLIM (X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,
     &                   HITTEN,EXIT)


      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,L,M,N,X2,Y2,Z2,R,P,Q,MAGNV,PI,TETAFM
      REAL*8 INORM,RFOV,ZMAX,NORME,SOLPLU,SOLMIN
      REAL*8 REXT,RINT,DENIV
      LOGICAL HITTEN,EXIT,ROOTS
      DIMENSION MAGNV(15),HITTEN(15),X2(15),Y2(15),Z2(15),INORM(15,3)
      EXTERNAL NORME
      COMMON /FOVL/ RFOV,TETAFM
      COMMON /DOME/ REXT,RINT,DENIV
C
      PI=DACOS(-1.D0)
      ZMAX=RFOV*DCOS(TETAFM)+DENIV
      HITTEN(1)=.FALSE.
      HITTEN(5)=.FALSE.
      EXIT=.FALSE.
C
      R=L*L+M*M+N*N
      P=2.0D0*(L*X1+M*Y1+N*Z1-N*DENIV)
      Q=X1*X1+Y1*Y1+Z1*Z1-RFOV*RFOV-2*Z1*DENIV+DENIV*DENIV
      CALL POL2(R,P,Q,SOLPLU,SOLMIN,ROOTS)
C
C CHECK IF THE FOV LIMITER WAS HIT AT ALL
C
      IF (.NOT.ROOTS) RETURN
C
C ELSE COMPUTE THE POSSIBLE POINTS OF INTERSECTION
C
      Z2(1)=SOLPLU*N+Z1
      Z2(5)=SOLMIN*N+Z1
C
C Z2(1) IS CONSIDERED ONLY IF GREATER THAN 0.0
C
      IF (Z2(1).GT.0.D0) THEN
          X2(1)=SOLPLU*L+X1
          Y2(1)=SOLPLU*M+Y1
C
C THE POINT P2(X2,Y2,Z2) IS ACCEPTED IF NOT EQUAL TO Z1
C
       IF (.NOT.(DABS(X2(1)-X1).LT.1.0D-06.AND.DABS(Y2(1)-Y1).LT.1.0D-06
     &      .AND.DABS(Z2(1)-Z1).LT.1.0D-06)) THEN
              HITTEN(1)=.TRUE.
              INORM(1,1)=-X2(1)/RFOV
              INORM(1,2)=-Y2(1)/RFOV
              INORM(1,3)=-(Z2(1)-DENIV)/RFOV
              MAGNV(1)=NORME(X2(1)-X1,Y2(1)-Y1,Z2(1)-Z1)
              IF (Z2(1).GT.ZMAX) EXIT=.TRUE.
       END IF
      END IF
C
C Z2(5) IS CONSIDERED ONLY IF GREATER THAN 0.0
C
      IF (Z2(5).GT.0.D0) THEN
          X2(5)=SOLMIN*L+X1
          Y2(5)=SOLMIN*M+Y1
```

```
C
C THE POINT P2(X2,Y2,Z2) IS ACCEPTED IF NOT EQUAL TO Z1
C
       IF (.NOT.(DABS(X2(5)-X1).LT.1.0D-06.AND.DABS(Y2(5)-Y1).LT.1.0D-06
     &      .AND.DABS(Z2(5)-Z1).LT.1.0D-06)) THEN
              HITTEN(5)=.TRUE.
              INORM(5,1)=-X2(5)/RFOV
              INORM(5,2)=-Y2(5)/RFOV
              INORM(5,3)=-(Z2(5)-DENIV)/RFOV
              MAGNV(5)=NORME(X2(5)-X1,Y2(5)-Y1,Z2(5)-Z1)
              IF (Z2(5).GT.ZMAX) EXIT=.TRUE.
       END IF
      END IF
      END

************************************************************************
*                                                                      *
* THIS SUBROUTINE FINDS THE POSSIBLE POINT THAT WOULD BE AT THE INTER- *
* SECTION OF THE SUBSTRATE'S SURFACE AND THE LINE REPRESENTING THE DI- *
* RECTION OF THE BUNDLE. THE SUBSTRATE IS DIVIDED INTO TWO SURFACES,   *
* (NB 2 AND 6) ONE BEING THE UPPER SUBSTRATE ON WHICH LIES THE FOV     *
* LIMITER, AND THE OTHER BEING THE LOWER SUBSTRATE ON WHICH THE DOME   *
* RESTS. THE NUMBER OF POINTS HIT CAN BE 0, 1,OR 2.                    *
*                                                                      *
************************************************************************

      SUBROUTINE SUBSTR(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,
     &                  HITTEN,ENTER)


      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,L,M,N,X2,Y2,Z2,CARRE,MAGNV,INORM
      REAL*8 RFOV,RPA,NORME,PI,TETAFM,REXT,RINT,DENIV
      INTEGER SURFAC
      LOGICAL HITTEN,ENTER
      DIMENSION MAGNV(15),HITTEN(15),X2(15),Y2(15),Z2(15),INORM(15,3)
      EXTERNAL NORME
      COMMON /FOVL/ RFOV,TETAFM
      COMMON /SUBS/ RPA,SURFAC
      COMMON /DOME/ REXT,RINT,DENIV
C
C INITIALIZE CONSTANTS AND VARIABLES
C
      PI=DACOS(-1.D0)
      HITTEN(2)=.FALSE.
      HITTEN(6)=.FALSE.
      ENTER=.FALSE.
C
C COORD. OF THE POINT THAT WOULD HIT THE UPPER PART OF THE SUBSTRATE
C
      Z2(2)=0.0D0
      Y2(2)=(Z2(2)-Z1)*M/N+Y1
      X2(2)=(Z2(2)-Z1)*L/N+X1
      CARRE=X2(2)*X2(2)+Y2(2)*Y2(2)
C
C THE SOLUTION IS ACCEPTED IF ON DEFINED DISC (R1<=R<=R2)
C
      IF (CARRE.GT.(REXT*REXT-DENIV*DENIV).AND.CARRE.LT.(RFOV*RFOV-
     &    DENIV*DENIV)) THEN
C
C THE SOLUTION IS ACCEPTED IF NOT EQUAL TO P1
C
       IF (.NOT.(DABS(X2(2)-X1).LT.1.0D-06.AND.DABS(Y2(2)-Y1).LT.1.0D-06
     &    .AND.DABS(Z2(2)-Z1).LT.1.0D-06)) THEN
C
```

```
C IF THE BUNDLE DOES HIT THE UPPER PART OF THE SUBSTRATE
C
        HITTEN(2)=.TRUE.
        MAGNV(2)=NORME(X2(2)-X1,Y2(2)-Y1,Z2(2)-Z1)
        INORM(2,1)=0.0D0
        INORM(2,2)=0.0D0
        INORM(2,3)=1.0D0
       END IF
      END IF
C
C COORD. OF THE POINT THAT WOULD HIT THE LOWER PART OF THE SUBSTRATE
C
      Z2(6)=DENIV
      Y2(6)=(Z2(6)-Z1)*M/N+Y1
      X2(6)=(Z2(6)-Z1)*L/N+X1
      CARRE=X2(6)*X2(6)+Y2(6)*Y2(6)
C
C THE SOLUTION IS ACCEPTED IF ON DEFINED DISC (RPA<=R<=R1)
C
      IF (CARRE.GT.(RPA*RPA).AND.CARRE.LT.(REXT*REXT)) THEN
C
C THE SOLUTION IS NOT ACCEPTED IF EQUAL TO P1
C
       IF (.NOT.(DABS(X2(6)-X1).LT.1.0D-06.AND.DABS(Y2(6)-Y1).LT.1.0D-06
     &    .AND.DABS(Z2(6)-Z1).LT.1.0D-06)) THEN
C
C IF THE BUNDLE DOES HIT THE LOWER PART OF THE SUBSTRATE
C
        HITTEN(6)=.TRUE.
        MAGNV(6)=NORME(X2(6)-X1,Y2(6)-Y1,Z2(6)-Z1)
        INORM(6,1)=0.0D0
        INORM(6,2)=0.0D0
        INORM(6,3)=1.0D0
       END IF
      END IF

*     WRITE(16,*)'IF THE SURFACE IS CHOSEN THE BUNDLE ENTERS THE CAVITY'
C
C ALSO CHECK IF THE BUNDLE ENTERS THE CAVITY
C
      IF (CARRE.LE.(RPA*RPA)) THEN
         HITTEN(6)=.FALSE.
      END IF
      END

************************************************************************
*                                                                      *
* THIS SUBROUTINE FINDS THE POSSIBLE POINTS THAT WOULD BE AT THE INTER-*
* SECTION OF ONE SURFACE OF THE DOME AND THE LINE REPRESENTING THE     *
* DIRECTION OF THE BUNDLE. THE NUMBER OF POINTS CAN BE 0, 1, OR 2.     *
*                                                                      *
************************************************************************

      SUBROUTINE SPHERE(X1,Y1,Z1,L,M,N,X2,Y2,Z2,NO,MAGNV,INORM,
     &                  HITTEN)


      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,L,M,N,X2,Y2,Z2,A,B,C,RAD,R,P,Q,MAGNV
      REAL*8 INORM,NORME,REXT,RINT,DENIV,SOLPLU,SOLMIN
      LOGICAL HITTEN,ROOTS
      DIMENSION MAGNV(15),HITTEN(15),RAD(15),X2(15),Y2(15)
      DIMENSION Z2(15),A(15),B(15),C(15),INORM(15,3)
      INTEGER NO
      EXTERNAL NORME
```

```
      COMMON /DOME/ REXT,RINT,DENIV
      DATA A,B,C /45*0.0D0/
      RAD(3)=REXT
      RAD(4)=RINT
      C(3)=DENIV
      C(4)=DENIV
      HITTEN(NO)=.FALSE.
      HITTEN(NO+4)=.FALSE.
C
      R=1.D0+(L/N)*(L/N)+(M/N)*(M/N)
      P=2.D0*((X1-A(NO))*L/N+(Y1-B(NO))*M/N-Z1*((L/N)*(L/N)
     &+(M/N)*(M/N))-C(NO))
      Q=(Z1*L/N-X1)*(Z1*L/N-X1)+(Z1*M/N-Y1)*(Z1*M/N-Y1)
     &+2.D0*A(NO)*(Z1*L/N-X1)+2.D0*B(NO)*(Z1*M/N-Y1)
     &+A(NO)*A(NO)+B(NO)*B(NO)+C(NO)*C(NO)-RAD(NO)*RAD(NO)
C
      CALL POL2(R,P,Q,SOLPLU,SOLMIN,ROOTS)
C
C CHECK IF THE DOME WAS HIT AT ALL
C
      IF (.NOT.ROOTS) RETURN
C
C ELSE COMPUTE THE POSSIBLE POINTS OF INTERSECTION
C
      Z2(NO)=SOLPLU
      Z2(NO+4)=SOLMIN
C
C IF THE Z-COORDINATE Z2(NO) IS GREATER THAN DENIV X & Y ARE COMPUTED
C
      IF (Z2(NO).GT.DENIV) THEN
          X2(NO)=(Z2(NO)-Z1)*L/N+X1
          Y2(NO)=(Z2(NO)-Z1)*M/N+Y1
C
C POINT P2(X2,Y2,Z2) IS ACCEPTED IF NOT EQUAL TO P1(X1,Y1,Z1)
C
      IF (.NOT.(DABS(X2(NO)-X1).LT.1.0D-06.AND.DABS(Y2(NO)-Y1).LT.
     &     1.0D-06.AND.DABS(Z2(NO)-Z1).LT.1.0D-06)) THEN
         HITTEN(NO)=.TRUE.
C
C IF NOT REJECTED THE VECTOR NORMAL TO THE SURFACE AT P2, AND THE
C DISTANCE P1P2 ARE COMPUTED
C
         INORM(NO,1)=-(X2(NO)-A(NO))/RAD(NO)
         INORM(NO,2)=-(Y2(NO)-B(NO))/RAD(NO)
         INORM(NO,3)=-(Z2(NO)-C(NO))/RAD(NO)
         MAGNV(NO)=NORME(X2(NO)-X1,Y2(NO)-Y1,Z2(NO)-Z1)
      END IF
      END IF
C
C IF THE Z-COORDINATE Z2(NO+4) IS GREATER THAN DENIV X & Y ARE COMPUTED
C
      IF (Z2(NO+4).GT.DENIV) THEN
          X2(NO+4)=(Z2(NO+4)-Z1)*L/N+X1
          Y2(NO+4)=(Z2(NO+4)-Z1)*M/N+Y1
C
C POINT P2(X2,Y2,Z2) IS ACCEPTED IF NOT EQUAL TO P1(X1,Y1,Z1)
C
      IF (.NOT.(DABS(X2(NO+4)-X1).LT.1.0D-06.AND.DABS(Y2(NO+4)-Y1).LT.
     &     1.0D-06.AND.DABS(Z2(NO+4)-Z1).LT.1.0D-06)) THEN
           HITTEN(NO+4)=.TRUE.
C
C IF NOT REJECTED THE VECTOR NORMAL TO THE SURFACE AT P2, AND THE
C DISTANCE P1P2 ARE COMPUTED
C
           INORM(NO+4,1)=-(X2(NO+4)-A(NO))/RAD(NO)
```

```
           INORM(NO+4,2)=-(Y2(NO+4)-B(NO))/RAD(NO)
           INORM(NO+4,3)=-(Z2(NO+4)-C(NO))/RAD(NO)
           MAGNV(NO+4)=NORME(X2(NO+4)-X1,Y2(NO+4)-Y1,Z2(NO+4)-Z1)
       END IF
      END IF

      END
***********************************************************************
*                                                                     *
*  This subroutine finds the possible points that would be at the    *
*  intersection of the barrel and the line representing the          *
*  direction of the bundle.  The number of point can be 0, 1, or 2.  *
*                                                                     *
***********************************************************************

      SUBROUTINE BARREL (X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN)

      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,L,M,N,X2,Y2,Z2,R,P,Q,MAGNV,NORME
      REAL*8 INORM,RCAV,LBAR,SOLPLU,SOLMIN,LCAV,LCON,CAVANG
      REAL*8 REXT,RINT,DENIV
      LOGICAL HITTEN,ROOTS
      DIMENSION MAGNV(15),HITTEN(15),X2(15),Y2(15),Z2(15),INORM(15,3)
      EXTERNAL NORME
      COMMON/CAVITY/RCAV,LBAR,LCON,CAVANG
      COMMON /DOME/REXT,RINT,DENIV

      HITTEN(9)=.FALSE.
      HITTEN(10)=.FALSE.


      R=L*L+M*M
      P=2.D0*(X1*L+Y1*M)
      Q=X1*X1+Y1*Y1-RCAV*RCAV

      CALL POL2(R,P,Q,SOLPLU,SOLMIN,ROOTS)
C
C CHECK IF THE BARREL WAS HIT AT ALL
C
      IF(.NOT.ROOTS) RETURN
C
C ELSE COMPUTE THE POINTS OF INTERSECTION
C
      Z2(9)=SOLPLU*N+Z1

      Z2(10)=SOLMIN*N+Z1
C
C Z2(9) IS CONSIDERED ONLY IF BETWEEN -(LBAR + DENIV) AND -DENIV
C
      IF ((DENIV-LBAR).LE.Z2(9).AND.Z2(9).LE.DENIV) THEN
            X2(9)=SOLPLU*L+X1
            Y2(9)=SOLPLU*M+Y1
C
C THE POINT P2(X2,Y2,Z2) IS ACCEPTED IF NOT EQUAL TO P1
C
      IF (.NOT.(DABS(X2(9)-X1).LT.1.0D-06.AND.DABS(Y2(9)-Y1).LT.1.0D-06
     &     .AND.DABS(Z2(9)-Z1).LT.1.0D-06)) THEN
             HITTEN(9)=.TRUE.
             INORM(9,1)=-1.D0*X2(9)/RCAV
             INORM(9,2)=-1.D0*Y2(9)/RCAV
             INORM(9,3)=0.0D0
             MAGNV(9)=NORME(X2(9)-X1,Y2(9)-Y1,Z2(9)-Z1)
        END IF
      END IF
C
```

```
C Z2(10) IS ONLY CONSIDERED IF BETWEEN -(LBAR + DENIV) AND -DENIV
C
      IF ((DENIV-LBAR).LE.Z2(10).AND.Z2(10).LE.DENIV) THEN
            X2(10)=SOLMIN*L+X1
            Y2(10)=SOLMIN*M+Y1
C
C THE POINT P2(X2,Y2,Z2) IS ACCEPTED IF NOT EQUAL TO P1
C
      IF (.NOT.(DABS(X2(10)-X1).LT.1.0D-06.AND.DABS(Y2(10)-Y1).LT.
     &    1.0D-06.AND.DABS(Z2(10)-Z1).LT.1.0D-06)) THEN
             HITTEN(10)=.TRUE.
             INORM(10,1)=-1.D0*X2(10)/RCAV
             INORM(10,2)=-1.D0*Y2(10)/RCAV
             INORM(10,3)=0.0D0
             MAGNV(10)=NORME(X2(10)-X1,Y2(10)-Y1,Z2(10)-Z1)
        END IF
      END IF
      END
***********************************************************************
*                                                                     *
*  This subroutine finds the possible points that would be at the     *
*  intersection of the cone and the line representing the direction   *
*  of the bundle.  The number of points can be 0, 1, or 2.            *
*                                                                     *
***********************************************************************

      SUBROUTINE CONE(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN)

      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM
      REAL*8 RCAV,LCON,CAVANG,C,P,Q,R,SOLPLU,SOLMIN,NORME
      REAL*8 LCAV,LBAR,DENIV,REXT,RINT,TANCAV

      LOGICAL HITTEN,ROOTS
      DIMENSION MAGNV(15),HITTEN(15),X2(15),Y2(15),Z2(15),INORM(15,3)
      EXTERNAL NORME
      COMMON /CAVITY/RCAV,LBAR,LCON,CAVANG
      COMMON /DOME/REXT,RINT,DENIV

      HITTEN(11)=.FALSE.
      HITTEN(12)=.FALSE.

      C=DENIV-LBAR
      LCAV=-1.D0*(LBAR+LCON-DENIV)

      TANCAV=RCAV/LCON


      R=L*L+M*M-N*N*TANCAV*TANCAV
      P=2.D0*(L*X1+M*Y1+N*TANCAV*TANCAV*(LCAV-Z1))
      Q=(X1*X1+Y1*Y1+TANCAV*TANCAV*(2.D0*LCAV*Z1-LCAV*LCAV-Z1*Z1))

      CALL POL2(R,P,Q,SOLPLU,SOLMIN,ROOTS)
C
C CHECK TO SEE IF THE CONE WAS HIT AT ALL
C
      IF (.NOT. ROOTS) RETURN
C
C ELSE COMPUTE THE POSSIBLE POINTS OF INTERSECTION
C
      Z2(11)=SOLPLU*N+Z1
      Z2(12)=SOLMIN*N+Z1
C
C Z2(11) IS CONSIDERED ONLY IF BETWEEN -(LCON+LBAR-DENIV) AND
C       -(LBAR+DENIV)
```

```
C
      IF ((LCAV).LE.Z2(11).AND.Z2(11).LT.C) THEN
           X2(11)=SOLPLU*L+X1
           Y2(11)=SOLPLU*M+Y1
C
C THE POINT P2(X2,Y2,Z2) IS ACCEPTED IF NOT EQUAL TO P1
C
      IF (.NOT.(DABS(X2(11)-X1).LT.1.0D-06.AND.DABS(Y2(11)-Y1).LT.
     &    1.0D-06.AND.DABS(Z2(11)-Z1).LT.1.0D-06)) THEN
             HITTEN(11)=.TRUE.
             INORM(11,1)=-1.D0*(X2(11)/(DSQRT(X2(11)*X2(11)+
     &                    Y2(11)*Y2(11))))*DCOS(CAVANG)

             INORM(11,2)=-1.D0*(Y2(11)/(DSQRT(X2(11)*X2(11)+
     &                    Y2(11)*Y2(11))))*DCOS(CAVANG)
             INORM(11,3)=DSIN(CAVANG)
             MAGNV(11)=NORME(X2(11)-X1,Y2(11)-Y1,Z2(11)-Z1)
        END IF
      END IF
C
C Z2(12) IS CONSIDERED ONLY IF BETWEEN -(LCON+LBAR-DENIV) AND
C     -(LBAR+DENIV)
C
      IF((LCAV).LE.Z2(12).AND.Z2(12).LT.C) THEN
            X2(12)=SOLMIN*L+X1
            Y2(12)=SOLMIN*M+Y1
C
C THE POINT P2(X2,Y2,Z2) IS ACCEPTED IF NOT EQUAL TO P1
C
      IF (.NOT.(DABS(X2(12)-X1).LT.1.0D-06.AND.DABS(Y2(12)-Y1).LT.
     &    1.0d-06.AND.DABS(Z2(12)-Z1).LT.1.0D-06)) THEN
             HITTEN(12)=.TRUE.
             INORM(12,1)=-1.D0*(X2(12)/(DSQRT(X2(12)*X2(12)+
     &                    Y2(12)*Y2(12))))*DCOS(CAVANG)
             INORM(12,2)=-1.D0*(Y2(12)/(DSQRT(X2(12)*X2(12)+
     &                    Y2(12)*Y2(12))))*DCOS(CAVANG)
             INORM(12,3)=DSIN(CAVANG)
             MAGNV(12)=NORME(X2(12)-X1,Y2(12)-Y1,Z2(12)-Z1)
        END IF
      END IF
      END

************************************************************************
*                                                                      *
* THIS SUBROUTINE FINDS THE POSSIBLE POINT THAT WOULD BE AT THE        *
* INTERSECTION OF THE PORTION OF THE SUBSTRATE WHICH IS VISIBLE TO     *
* THE CAVITY AND THE LINE REPRESENTING THE PATH OF THE BUNDLE          *
*                                                                      *
************************************************************************

      SUBROUTINE CAVSUB(X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,
     &                  HITTEN,EXITCAV,EXCAV)

      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,CARRE
      REAL*8 RFOV,RPA,NORME,PI,TETAFM,REXT,RINT,DENIV,RCAV
      REAL*8 LBAR,LCON,CAVANG
      INTEGER SURFAC,EXCAV
      LOGICAL HITTEN,ENTER,EXITCAV
      DIMENSION MAGNV(15),HITTEN(15),X2(15),Y2(15),Z2(15),INORM(15,3)
      EXTERNAL NORME
      COMMON /FOVL/ RFOV,TETAFM
      COMMON /SUBS/ RPA, SURFAC
      COMMON /DOME/ REXT,RINT,DENIV
      COMMON /CAVITY/ RCAV,LBAR,LCON,CAVANG
```

```
C
C INITIALIZE CONSTANTS AND VARIABLES
C
      PI=DACOS(-1.D0)
      HITTEN(13)=.FALSE.
      EXITCAV=.FALSE.
C
C COORD. OF THE POINT THAT WOULD HIT THE DISC VISIBLE TO THE CAVITY
C
      Z2(13)=DENIV
      Y2(13)=(Z2(13)-Z1)*M/N+Y1
      X2(13)=(Z2(13)-Z1)*L/N+X1
      CARRE=DSQRT(X2(13)*X2(13)+Y2(13)*Y2(13))
C
C THE SOLUTION IS ACCEPTED IF ON DEFINED DISC (RPA<=R<=RCAV)
C
       IF (CARRE.GT.(RPA).AND.CARRE.LE.(RCAV)) THEN
C
C THE SOLUTION IS NOT ACCEPTED IF EQUAL TO P1
C
         IF (.NOT.(DABS(X2(13)-X1).LT.1.0D-06.AND.DABS(Y2(13)-Y1)
     &       .LT.1.0D-06.AND.DABS(Z2(13)-Z1).LT.1.0D-06)) THEN
C
C IF THE BUNDLE DOES HIT THE DISC FACING THE CAVITY
C
            HITTEN(13)=.TRUE.
            MAGNV(13)=NORME(X2(13)-X1,Y2(13)-Y1,Z2(13)-Z1)
            INORM(13,1)=0.0D0
            INORM(13,2)=0.0D0
            INORM(13,3)=-1.0D0
         END IF
      END IF
C
C ALSO CHECK IF THE BUNDLE LEAVES THE CAVITY
C
      IF (CARRE.LE.(RPA)) THEN
         HITTEN(13)=.FALSE.
         EXITCAV=.TRUE.
         EXCAV=EXCAV+1
      END IF
      END

*************************************************************************
*                                                                       *
*  This subroutine finds the possible points that would be at the      *
*  intersection of the ring and the line representing the              *
*  direction of the bundle.  The number of point can be 0, 1, or 2.    *
*                                                                       *
*************************************************************************

      SUBROUTINE RING (X1,Y1,Z1,L,M,N,X2,Y2,Z2,MAGNV,INORM,HITTEN)

      IMPLICIT NONE
      REAL*8 X1,Y1,Z1,L,M,N,X2,Y2,Z2,R,P,Q,MAGNV,NORME
      REAL*8 INORM,SOLPLU,SOLMIN
      REAL*8 REXT,RINT,DENIV
      LOGICAL HITTEN,ROOTS
      DIMENSION MAGNV(15),HITTEN(15),X2(15),Y2(15),Z2(15),INORM(15,3)
      EXTERNAL NORME
      COMMON /DOME/REXT,RINT,DENIV

      HITTEN(14)=.FALSE.
      HITTEN(15)=.FALSE.

      R=L*L+M*M+N*N
```

```
      P=2.D0*(X1*L+Y1*M+N*Z1-N*DENIV)
      Q=X1*X1+Y1*Y1+Z1*Z1-REXT*REXT-2*Z1*DENIV+DENIV*DENIV

      CALL POL2(R,P,Q,SOLPLU,SOLMIN,ROOTS)

C
C CHECK IF THE RING WAS HIT AT ALL
C
      IF (.NOT.ROOTS) THEN
        RETURN
      END IF
C
C ELSE COMPUTE THE POINTS OF INTERSECTION
C
      Z2(14)=SOLPLU*N+Z1
      Z2(15)=SOLMIN*N+Z1
C
C Z2(14) IS CONSIDERED ONLY IF BETWEEN 0.0D0 AND DENIV
C
      IF ((DENIV).LE.Z2(14).AND.Z2(14).LE.0.0D0) THEN
            X2(14)=SOLPLU*L+X1
            Y2(14)=SOLPLU*M+Y1
C
C THE POINT P2(X2,Y2,Z2) IS ACCEPTED IF NOT EQUAL TO P1
C
      IF (.NOT.(DABS(X2(14)-X1).LT.1.0D-06.AND.DABS(Y2(14)-Y1).LT.
     &    1.0D-06.AND.DABS(Z2(14)-Z1).LT.1.0D-06)) THEN
             HITTEN(14)=.TRUE.
             INORM(14,1)=-X2(14)/REXT
             INORM(14,2)=-Y2(14)/REXT
             INORM(14,3)=-(Z2(14)-DENIV)/REXT
             MAGNV(14)=NORME(X2(14)-X1,Y2(14)-Y1,Z2(14)-Z1)
        END IF
      END IF
C
C Z2(15) IS ONLY CONSIDERED IF BETWEEN 0.0D0 AND DENIV
C
      IF ((DENIV).LE.Z2(15).AND.Z2(15).LE.0.0D0) THEN
            X2(15)=SOLMIN*L+X1
            Y2(15)=SOLMIN*M+Y1
C
C THE POINT P2(X2,Y2,Z2) IS ACCEPTED IF NOT EQUAL TO P1
C
      IF (.NOT.(DABS(X2(15)-X1).LT.1.0D-06.AND.DABS(Y2(15)-Y1).LT.
     &    1.0D-06.AND.DABS(Z2(15)-Z1).LT.1.0D-06)) THEN
             HITTEN(15)=.TRUE.
             INORM(15,1)=-X2(15)/REXT
             INORM(15,2)=-Y2(15)/REXT
             INORM(15,3)=(-Z2(15)+DENIV)/REXT
             MAGNV(15)=NORME(X2(15)-X1,Y2(15)-Y1,Z2(15)-Z1)
        END IF
      END IF
      END

************************************************************************
*                                                                      *
* THIS SUBROUTINE DETERMINES WHETHER A BUNDLE IS ABSORBED OR REFLECTED *
* WHEN IT IS FOUND TO HAVE HIT THE FOV LIMITER OR THE SUBSTRATE        *
*                                                                      *
************************************************************************

      SUBROUTINE REFOAB(L,M,N,X,Y,Z,INORM,SURFAC,ABSOR,REFL,DIFUS)

      IMPLICIT NONE
      REAL*8 L,M,N,INORM,PI,RS,X,Y,Z,RDM,ALPHA,ROSPEC,RODIFF
```

```
      REAL*8 ALPHA1,ALPHA2,ALPHA9,ALPHA11,ALPHA13,ALPHA14
      REAL*8 ROSPEC1,ROSPEC2,ROSPEC9,ROSPEC11,ROSPEC13,ROSPEC14
      REAL*8 RODIFF1,RODIFF2,RODIFF9,RODIFF11,RODIFF13,RODIFF14
C
      DIMENSION INORM(15,3)
C
      INTEGER SURFAC
      INTEGER CNTRDM
C
      LOGICAL ABSOR,REFL,DIFUS
C
      COMMON /RAND/ CNTRDM
      COMMON /REAB/ ALPHA1,ALPHA2,ALPHA9,ALPHA11,ALPHA13,ALPHA14,
     &              ROSPEC1,ROSPEC2,ROSPEC9,ROSPEC11,ROSPEC13,ROSPEC14,
     &              RODIFF1,RODIFF2,RODIFF9,RODIFF11,RODIFF13,RODIFF14

      PI=DACOS(-1.D0)
C
C SET THE FLAGS TO FALSE
C
      ABSOR=.FALSE.
      REFL=.FALSE.
      DIFUS=.FALSE.
C
C IF CURRENT SURFACE IS THE FOV LIMITER SET ITS PROPERTIES TO BE CURRENT
C
      IF (SURFAC.EQ.1.OR.SURFAC.EQ.5) THEN
          ALPHA=ALPHA1
          ROSPEC=ROSPEC1
          RODIFF=RODIFF1
C
C IF CURRENT SURFACE IS THE SUBSTRATE SET ITS PROPERTIES TO BE CURRENT
C
      ELSE IF (SURFAC.EQ.2.OR.SURFAC.EQ.6) THEN
          ALPHA=ALPHA2
          ROSPEC=ROSPEC2
          RODIFF=RODIFF2
C
C  IF CURRENT SURFACE IS THE BARREL SET ITS PROPERTIES TO BE CURRENT
C
      ELSE IF (SURFAC.EQ.9.OR.SURFAC.EQ.10) THEN
          ALPHA=ALPHA9
          ROSPEC=ROSPEC9
          RODIFF=RODIFF9
C
C  IF CURRENT SURFACE IS THE CONE SET ITS PROPERTIES TO BE CURRENT
C
      ELSE IF (SURFAC.EQ.11.OR.SURFAC.EQ.12) THEN
          ALPHA=ALPHA11
          ROSPEC=ROSPEC11
          RODIFF=RODIFF11
C
C IF CURRENT SURFACE IS CAVSUB SET ITS PROPERTIES TO BE CURRENT
C
      ELSE IF (SURFAC.EQ.13) THEN
          ALPHA=ALPHA13
          ROSPEC=ROSPEC13
          RODIFF=RODIFF13
C
C IF CURRENT SURFACE IS RING SET ITS PROPERTIES TO BE CURRENT
C
      ELSE IF (SURFAC.EQ.14.OR.SURFAC.EQ.15) THEN
          ALPHA=ALPHA14
          ROSPEC=ROSPEC14
          RODIFF=RODIFF14
```

```
      ELSE
          WRITE(06,*)'CASE IGNORED IN REFOAB!!!!!!!!!!'
          STOP
      END IF
      RS=ROSPEC/(ROSPEC+RODIFF)
C
C GET A RANDOM NUMBER TO SELLECT BETWEEN ABSORPTION AND REFLEXION
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
C
C IF THE RANDOM NUMBER IS LESS THAN ALPHA, THE BUNDLE IS ABSORBED
C
      IF (RDM .LE. ALPHA) THEN
          ABSOR=.TRUE.
C
C IF THE RANDOM NUMBER IS GREATER THAN ALPHA, THE BUNDLE IS REFLECTED
C
      ELSE
C
C GET A RANDOM NUMBER TO SELLECT BETWEEN SPECULAR AND DIFFUSE REFLEXION
C
          CALL DRNUN(1,RDM)
          CNTRDM=CNTRDM+1
C
C IF THE RDM NUMBER IS LESS THAN RS, THE BUNDLE IS SPECULARLY REFLECTED
C
          IF (RDM.LE.RS) THEN
              CALL REFLEX(L,M,N,INORM,SURFAC)
              REFL=.TRUE.
C
C IF THE RDM NBER IS GREATER THAN RS, THE BUNDLE IS DIFFUSELY REFLECTED
C
          ELSE
              CALL DIFEMI(L,M,N,X,Y,Z,SURFAC)
              DIFUS=.TRUE.
         END IF
      END IF
      END
************************************************************************
*                                                                      *
* THIS SUBROUTINE DETERMINES WHETHER A BUNDLE IS TRANSMITTED OR
* REFLECTED WHEN IT IS FOUND TO HAVE HIT THE DOME                      *
*                                                                      *
************************************************************************
      SUBROUTINE REFOTR(L,M,N,INORM,SURFAC,REFL,TRANS,INDOME)
      IMPLICIT NONE
      REAL*8 L,M,N,INORM
      REAL*8 R
      DIMENSION INORM(15,3)
      INTEGER SURFAC,CNTBRW,CHANEL
      LOGICAL ABSOR,REFL,TRANS,OK,INDOME
      COMMON /COUN/ CNTBRW
      COMMON /CHANNEL/ CHANEL
C
C SET FLAGS TO FALSE
C
      OK=.FALSE.
      ABSOR=.FALSE.
      REFL=.FALSE.
      TRANS=.FALSE.
C
C CALL ROUTINE WHICH WILL FIND THE NEW DIR. COSINES AFTER TRANSMISSION
C
      CALL TRANSM(L,M,N,INORM,SURFAC,INDOME,OK)
```

```
C
C IF THE ANGLE OF INCIDENCE IS SUCH THAT THE RAY CAN NOT GO THRU THE
C INTERFACE DOME/VACUUM, THE RAY IS REFLECTED SPECULARLY.
C
      IF (.NOT.OK) THEN
          CNTBRW=CNTBRW+1
          CALL REFLEX(L,M,N,INORM,SURFAC)
          REFL=.TRUE.
C
C If the ray was transmitted, change the status of INDOME
C
      ELSE
        TRANS=.TRUE.
        IF (INDOME) THEN
          INDOME=.FALSE.
        ELSE
          INDOME=.TRUE.
        END IF
      END IF

      END

*****************************************************************
*                                                               *
*  THIS SUBROUTINE CALCULATES THE NEW DIRECTION COSINES AFTER   *
*  A REFLEXION OCCURED ON THE FOV LIMITER OR SUBSTRATE          *
*                                                               *
*****************************************************************
      SUBROUTINE REFLEX(L,M,N,INORM,SURFAC)

      IMPLICIT NONE
      REAL*8 L,M,N,INORM,PI,N1,N2,N3,LP,MP,NP,KA
      INTEGER SURFAC
      DIMENSION INORM(15,3)

      PI=DACOS(-1.0D0)
      N1=INORM(SURFAC,1)
      N2=INORM(SURFAC,2)
      N3=INORM(SURFAC,3)
      KA=L*N1+M*N2+N*N3
      LP=L-2.D0*KA*N1
      MP=M-2.D0*KA*N2
      NP=N-2.D0*KA*N3
      L=LP
      M=MP
      N=NP
      END
*******************************************************************
*                                                                 *
*  THIS SUBROUTINE CALCULATES THE NEW DIRCTION COSINES WHEN       *
*  TRANSMISSION OCCURS THROUGH THE FILTER DOME                    *
*                                                                 *
*******************************************************************
      SUBROUTINE TRANSM(L,M,N,INORM,SURFAC,INDOME,OK)

      IMPLICIT NONE
      REAL*8 L,M,N,INORM,RIND1,RIND2,N1,N2,N3,KA,KB,KC,KD,LP,MP,NP
      REAL*8 TETA1,TETA2,IVAC,IDOM,TAMPON
      INTEGER IS,SURFAC,CHANEL
      PARAMETER (IS=15)
      DIMENSION INORM(15,3)
      LOGICAL OK,INDOME
      COMMON /TRAN/ IVAC,IDOM
      COMMON /CHANNEL/ CHANEL
```

```
      OK=.FALSE.
C
C Assign the correct indeces of refraction according to whether
C the bundle is inside or outside the dome
C
      IF (INDOME) THEN
        RIND1 = IDOM
        RIND2 = IVAC
      ELSE
        RIND1 = IVAC
        RIND2 = IDOM
      END IF
C
C Compute the angle of transmission in the transmission plan
C
      N1=INORM(SURFAC,1)
      N2=INORM(SURFAC,2)
      N3=INORM(SURFAC,3)
      KA=L*N1+M*N2+N*N3
      IF (DABS(KA).GT.1.0D0) THEN
          IF (DABS(DABS(KA)-1.0D0).LT.1.0D-04) THEN
              KA=1.0D0
          ELSE
              WRITE(16,*)'PB AVEC KA DANS TRANS'
              WRITE(16,*)L,M,N,N1,N2,N3
          END IF
      END IF
      KC=RIND1/RIND2
      TETA1=DACOS(ABS(KA))
      KD=KC*DSIN(TETA1)
C
C CHECK IF TETA1 IS LARGER THAN THE BREWSTER ANGLE
C
      IF (KD.LE.1.0D0) THEN
          OK=.TRUE.
          TETA2=DASIN(KD)
          KB=DCOS(TETA2)/DCOS(TETA1)-RIND1/RIND2
          LP=RIND1/RIND2*L+KA*N1*KB
          MP=RIND1/RIND2*M+KA*N2*KB
          NP=RIND1/RIND2*N+KA*N3*KB
          L=LP
          M=MP
          N=NP
      END IF
      END

******************************************************************
*                                                                *
*  THIS SUBROUTINE GENERATES A RANDOM DIRECTION WHEN A DIFFUSE   *
*  REFLEXION OCCURS ON THE FOV LIMITER OR THE SUBSTRATE          *
*                                                                *
******************************************************************
      SUBROUTINE DIFEMI(L,M,N,X,Y,Z,SURFAC)

      IMPLICIT NONE
      REAL*8 L,M,N,LP,MP,NP,X,Y,Z,PHILOC,TTALOC,PI,RDM
      REAL*8 RFOV,TETAFM,RADIUS
      REAL*8 REXT,RINT,DENIV,DISTTOC
C
      INTEGER SURFAC
      INTEGER CNTRDM
C
      COMMON /RAND/ CNTRDM
      COMMON /FOVL/ RFOV,TETAFM
      COMMON /DOME/ REXT,RINT,DENIV
```

```
      PI=DACOS(-1.D0)
C
C RANDOM EMISSION DIRECTION IN THE LOCAL COORD. SYSTEM
C
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      PHILOC=DASIN(DSQRT(RDM))
      CALL DRNUN(1,RDM)
      CNTRDM=CNTRDM+1
      TTALOC=2.D0*PI*RDM
C
C COMPUTE THE SLOPE OF THE EMITTED RAY IN THE LOCAL COORD. SYSTEM
C
      LP=DSIN(PHILOC)*DCOS(TTALOC)
      MP=DSIN(PHILOC)*DSIN(TTALOC)
      NP=DCOS(PHILOC)
C
C IF EMISION FROM THE FOV LIMITER
C
      IF (SURFAC.EQ.1.OR.SURFAC.EQ.5) THEN
C
C TRANSFORMATION OF THE DIRECTION COSINES INTO GLOBAL COORD. SYSTEM
C
      L=-(X*Z/(RFOV*DSQRT(RFOV*RFOV-Z*Z))*LP
     &    +Y/DSQRT(RFOV*RFOV-Z*Z)*MP
     &    +X/RFOV*NP)
      M=-(Y*Z/(RFOV*DSQRT(RFOV*RFOV-Z*Z))*LP
     &    -X/DSQRT(RFOV*RFOV-Z*Z)*MP
     &    +Y/RFOV*NP)
      N=DSQRT(RFOV*RFOV-Z*Z)/RFOV*LP-Z/RFOV*NP
C
C IF EMISSION FROM SUBSTRATE
C
      ELSE IF (SURFAC.EQ.2.OR.SURFAC.EQ.6) THEN
C
C TRANSFORMATION OF THE DIRECTION COSINES INTO GLOBAL COORD. SYSTEM
C
      RADIUS=DSQRT(X*X+Y*Y)
      L=(X*LP-Y*MP)/RADIUS
      M=(Y*LP+X*MP)/RADIUS
      N=NP
C
C IF EMISSION FROM RING
C
      ELSE IF (SURFAC.EQ.14.OR.SURFAC.EQ.15) THEN
C
C TRANSFORMATION OF THE DIRECTION COSINES INTO GLOBAL COORD.SYSTEM
C
      DISTTOC=DSQRT(X*X+Y*Y+(Z-DENIV)*(Z-DENIV))
      L = -(X*(Z-DENIV)/(DISTTOC*DSQRT(DISTTOC*DISTTOC - (Z-DENIV)*
     1    (Z-DENIV)))*LP + Y/DSQRT(DISTTOC*DISTTOC - (Z-DENIV)*
     2    (Z-DENIV))*MP + X/DISTTOC*NP)
      M = -(Y*(Z-DENIV)/(DISTTOC*DSQRT(DISTTOC*DISTTOC - (Z-DENIV)*
     1    (Z-DENIV)))*LP - X/DSQRT(DISTTOC*DISTTOC - (Z-DENIV)*
     2    (Z-DENIV))*MP + Y/DISTTOC*NP)
      N = DSQRT(DISTTOC*DISTTOC - (Z-DENIV)*(Z-DENIV))/DISTTOC*LP -
     1    (Z-DENIV)/DISTTOC*NP
      END IF

      END
**************************************************************
*                                                            *
*  THIS SUBROUTINE INITIALIZES AN ARRAY OF SIZE 13           *
```

```
*                                                             *
***************************************************************
      SUBROUTINE INIT(X)

      IMPLICIT NONE
      REAL*8 X
      INTEGER I
      DIMENSION X(15)

      DO 284 I=1,15
          X(I)=0.D0
284   CONTINUE

      END
***************************************************************
*                                                             *
*  THIS SUBROUTINE SETS UP THE GEOMETRY FOR ALL SUBELEMENTS *
*  OF THE OPTICAL FRONT-END OF THE NONSCANNER.  IT BRAKES    *
*  UP THE DIFFERENT PARTS INTO SURFACE AND VOLUME ELEMENTS   *
*  IT ALSO COMPUTES THE SURFACE AREAS AND VOLUMES OF THE     *
*  DIFFERENT ELEMENTS.                                        *
*                                                             *
***************************************************************
      SUBROUTINE DOMSETUP(DIST)

      IMPLICIT NONE
      REAL*8 PI,DIST
      REAL*8 CE,CI,REXT,RINT,DENIV,RCEN
      REAL*8 RADF,RADP,DRAD
      REAL*8 PHIDEN,PHIF,PHIP,DPHI,DZPHI,ZPHIF,ZPHIP
      REAL*8 TTAF,TTAP,DTTA
      REAL*8 XNOD,YNOD,ZNOD,XFAC,YFAC,ZFAC
      REAL*8 VOLUM,REALSUM,SAREA,SURFAC
      REAL*8 ZMAX,Z,DZ,RFOV,TETAFM,FI,TH,RAD1,RAD2,RPA
      REAL*8 LBAR,RCAV,LCON,CAVANG
      REAL*8 DR,R,COSRNG
      REAL*8 AREA,TANCAV,H,INDEX,ZZ
C
      INTEGER I,J,K,MAXELT
      INTEGER NBSH,NBCO,NBCO1,NBPSD,NBLAT,NBLONG,NBPS,NBWASH,NBLSU
      INTEGER NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS,NBWLSU
      INTEGER NB1,NB2,NB3,NB4,NB9,NB11,NBVE,NB5,NB6,NB7,NB13
      INTEGER NB14,NB15,NB16,NB8,NB17,NB18
      INTEGER NBHSR,NBPSR
C
      PARAMETER(MAXELT=700)
C
      DIMENSION RADP(MAXELT),DRAD(MAXELT)
      DIMENSION PHIP(MAXELT),DPHI(MAXELT)
      DIMENSION TTAP(MAXELT)
      DIMENSION DIST(MAXELT)
      DIMENSION VOLUM(MAXELT)
C
      COMMON /RIING/ NBHSR,NBPSR
      COMMON /SER2/ NBLAT,NBLONG /SER3/ NBWASH,NBPS,NBLSU,NBWLSU
      COMMON /SER4/ NBSH,NBCO,NBPSD
      COMMON /FOVL/ RFOV,TETAFM
      COMMON /DOME/ REXT,RINT,DENIV
      COMMON /NB14/ NB1,NB2,NB3,NB4,NB9,NB11,NB13,NB14
      COMMON /NB57/ NB5,NB6,NB7,NB15,NB16,NB8,NB17,NB18
      COMMON /SETU/ RADF(MAXELT),PHIF(MAXELT),ttaf(maxelt),
     &              SAREA(MAXELT),RCEN
      COMMON /NODE/ XNOD(MAXELT),YNOD(MAXELT),ZNOD(MAXELT)
      COMMON /FACE/ XFAC(MAXELT),YFAC(MAXELT),ZFAC(MAXELT)
      COMMON /CAVITY/ RCAV,LBAR,LCON,CAVANG
```

```
      COMMON /CAVSUR/ NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS
      COMMON /SUBS/ RPA,SURFAC

      PI=DACOS(-1.D0)
C
C DOME DOME DOME DOME MESH DEFINITION DOME DOME DOME DOME DOME DOME
C
C DEFINITION OF THE RADII OF THE CV FACES AND THE CV NODES:
C THE 1ST CV FACE COINCIDES WITH THE OUTER SURFACE OF THE DOME.
C THE 1ST CV NODE HAS A RADIUS = REXT.
C THE INDEX OF THE CV NODES ARE THE SAME AS THE FACE FOLLOWING THEM
C IF YOU MOVE IN THE "DECREASING-RAD" DIRECTION.
C THE LAST CV FACE COINCIDES WITH THE INNER SURFACE OF THE DOME.
C THERE IS AN EXTRA SET OF NODES AT RADIUS = RINT FOR B.C. PURPOSES.
C
      RADF(1)=REXT
      RADP(1)=REXT
      DO 10 I=2,NBSH+1
       RADF(I)=((RADF(I-1))**3 - (REXT**3 - RINT**3)/NBSH)**(1.D0/3.D0)
       RADP(I)=(0.5D0*((RADF(I))**3 + (RADF(I-1))**3))**(1.D0/3.D0)
       DRAD(I-1)=RADP(I-1) - RADP(I)
10    CONTINUE
      RADP(NBSH+2)=RINT
      DRAD(NBSH+1)=RADP(NBSH+1) - RADP(NBSH+2)
C
C DEFINITION OF THE ZENITH ANGLE OF THE CV FACES AND CV NODES:
C PHI=0 AT NADIR; (FIRST CV FACE)
C PHIDEN IS THE ANGLE AT WHICH THE DOME STARTS TO BE BURIED IN THE
C SUBSTRATE.
C THE INDEX OF THE CV NODES ARE THE SAME AS THE FACE FOLLOWING THEM
C IF YOU MOVE IN THE "INCREASING-PHI" DIRECTION.
C SINCE THERE IS NO CV NODE BEFORE THE 1ST CV FACE, THE 1ST CV NODE IS
C INDEXED BY # 2.
C
      RCEN=(0.5D0*(REXT*REXT*REXT+RINT*RINT*RINT))**(1.D0/3.D0)
      DZPHI=(RCEN+DENIV)/(NBCO-1)
      ZPHIF=RCEN
      ZPHIP=RCEN+DZPHI/2.D0
      PHIF(1)=0.0D0
      PHIP(1)=0.0D0
      DO 20 J=2,NBCO
          ZPHIF=ZPHIF-DZPHI
          ZPHIP=ZPHIP-DZPHI
          PHIF(J)=DACOS(ZPHIF/RCEN)
          PHIP(J)=DACOS(ZPHIP/RCEN)
          DPHI(J-1)=PHIP(J) - PHIP(J-1)
20    CONTINUE
      PHIF(NBCO+1)=PI/2.D0
      PHIP(NBCO+1)=DACOS((-DENIV/2.D0)/RCEN)
      DPHI(NBCO)=PHIP(NBCO+1) - PHIP(NBCO)
      PHIP(NBCO+2)=PHIF(NBCO+1)
      DPHI(NBCO+1)=PHIP(NBCO+2) - PHIP(NBCO+1)
C
C DEFINITION OF THE AZIMUTHAL ANGLE OF THE CV FACES AND CV NODES
C THE 1ST CV FACE IS AT THETA = 0.
C !!!!NOTE: THE CV NODE HAS THE SAME # AS THE CV FACE PRECCEEDING IT!!!!
C THE CV FACES ARE CALCULATED TO NBPSD+1 JUST SO THAT ALL THE CV NODES
C CAN BE CALCULATED WITH THE SAME FORMULA.
C
      DTTA=2.D0*PI/NBPSD
      TTAF(1)=0.0D0
      DO 40 K=2,NBPSD+1
          TTAF(K)=TTAF(K-1) + DTTA
40    CONTINUE
      DO 41 K=1,NBPSD
```

```
          TTAP(K)=0.5D0*(TTAF(K+1) + TTAF(K))
41    CONTINUE
C
C SAVE RADF(I) AS DIST(I)
C
      DO 60 I=1,NBSH+1
          DIST(I)=RADF(I)
60    CONTINUE
C
C COMPUTE THE COORDINATES OF THE CENTROID OF EACH VOLUME ELEMENT
C
      DO 80 I=2,NBSH+1
      DO 80 J=2,NBCO+1
      DO 80 K=1,NBPSD
          NBVE=(I-2)*NBCO*NBPSD + (J-2)*NBPSD + K + NB3
          XNOD(NBVE)=RADP(I)*DSIN(PHIP(J))*DCOS(TTAP(K))
          YNOD(NBVE)=RADP(I)*DSIN(PHIP(J))*DSIN(TTAP(K))
          ZNOD(NBVE)=RADP(I)*DCOS(PHIP(J))+deniv
80    CONTINUE
C
C COMPUTE THE COORDINATES OF THE POINTS AT THE INTERSECTIONS OF THE
C CONTROL VOLUME FACES
C
      DO 90 I=1,NBSH+1
          NBVE=(I-1)*(NBCO*NBPSD + 1) + 1 + NB7
          XFAC(NBVE)=0.0D0
          YFAC(NBVE)=0.0D0
          ZFAC(NBVE)=RADF(I)
      DO 90 J=2,NBCO+1
      DO 90 K=1,NBPSD
          NBVE=(I-1)*(NBCO*NBPSD + 1) + (J-2)*NBPSD + K + 1 + NB7
          XFAC(NBVE)=RADF(I)*DSIN(PHIF(J))*DCOS(TTAF(K))
          YFAC(NBVE)=RADF(I)*DSIN(PHIF(J))*DSIN(TTAF(K))
          ZFAC(NBVE)=RADF(I)*DCOS(PHIF(J))+DENIV
          NB8=NBVE
90    CONTINUE
C
C COMPUTE THE VOLUME OF THE VOLUME ELEMENTS BRAKING UP THE DOME
C
      REALSUM=0.0D0
      DO 100 I=2,NBSH+1
      DO 100 J=2,NBCO+1
      DO 100 K=1,NBPSD
          NBVE=(I-2)*NBCO*NBPSD + (J-2)*NBPSD + K + NB3
          VOLUM(NBVE)=(DCOS(PHIF(J-1)) - DCOS(PHIF(J)))*
     1                 (RADF(I-1)*RADF(I-1)*RADF(I-1) - RADF(I)*
     2                 RADF(I)*RADF(I))/3.D0*DTTA
          SAREA(NBVE)=(DCOS(PHIF(J-1)) - DCOS(PHIF(J)))*
     1                 RADF(I-1)*RADF(I-1)*DTTA
          REALSUM=REALSUM+VOLUM(NBVE)
      WRITE(10,*)NBVE,SAREA(NBVE),VOLUM(NBVE)
100   CONTINUE
C
C COMPUTE THE COORDINATES OF THE POINTS AT THE INTERSECTION OF THE
C CONTROL SURFACE FACES FOR THE FOV LIMITER
C
      ZMAX=RFOV*DCOS(TETAFM)+DENIV
      Z=0.0D0
      DZ=ZMAX/DBLE(NBLAT)
      DO 120 I=1,NBLAT+1
          TH=0.0D0
          FI=DACOS(Z/RFOV)
      DO 121 J=1,NBLONG
          NBVE=(I-1)*NBLONG + J
          XFAC(NBVE)=RFOV*DSIN(FI)*DCOS(TH)
```

```
          YFAC(NBVE)=RFOV*DSIN(FI)*DSIN(TH)
          ZFAC(NBVE)=RFOV*DCOS(FI)+DENIV
          TH=TH + 2.D0*PI/DBLE(NBLONG)
121   CONTINUE
          Z=Z+DZ
120   CONTINUE
      DO 122 I=1,NBLAT
      DO 122 J=1,NBLONG
          NBVE=(I-1)*NBLONG + J
          SAREA(NBVE)=DCOS(TETAFM)*2.D0*PI*RFOV*RFOV/NB1
          WRITE(10,*)NBVE,SAREA(NBVE)
122   CONTINUE
C
C COMPUTE THE COORDINATES OF THE POINTS AT THE INTERSECTION OF THE
C CONTROL SURFACE FACES FOR THE LOWER SUBSTRATE
C
      DO 130 J=1,NBLSU
          TH=DBLE(J-1)*2.D0*PI/DBLE(NBLSU)
          NBVE=NB5+J
          XFAC(NBVE)=RPA*DCOS(TH)
          YFAC(NBVE)=RPA*DSIN(TH)
          ZFAC(NBVE)=DENIV
          SAREA(J+NB1)=PI*(RINT*RINT-RPA*RPA)/NBLSU
          WRITE(10,*)J+NB1,SAREA(J+NB1)
130   CONTINUE
      DO 131 J=1,NBLSU
          TH=DBLE(J-1)*2.D0*PI/DBLE(NBLSU)
          NBVE=NB5+NBLSU+J
          XFAC(NBVE)=RINT*DCOS(TH)
          YFAC(NBVE)=RINT*DSIN(TH)
          ZFAC(NBVE)=DENIV
131   CONTINUE
      DO 132 J=1,NBLSU
          TH=DBLE(J-1)*2.D0*PI/DBLE(NBLSU)
          NBVE=NB5+J+2*NBLSU
          YFAC(NBVE)=REXT*DCOS(TH)
          XFAC(NBVE)=REXT*DSIN(TH)
          ZFAC(NBVE)=DENIV
          SAREA(J+NB1+NBLSU)=PI*(REXT*REXT-RINT*RINT)/NBLSU
          WRITE(10,*)J+NB1+NBLSU,SAREA(J+NB1+NBLSU)
132   CONTINUE
C
C COMPUTE THE COORDINATES OF THE POINTS AT THE INTERSECTION OF THE
C CONTROL SURFACE FACES FOR THE SUBSTRATE
C
      RAD1=DSQRT(REXT*REXT-DENIV*DENIV)
      RAD2=RFOV
      DO 140 I=1,NBWASH+1
          R=DSQRT(DBLE(I-1)/NBWASH*(RAD2*RAD2-RAD1*RAD1) + RAD1*RAD1)
          TH=0
      DO 140 J=1,NBPS
          NBVE=(I-1)*NBPS + J + NB6
          XFAC(NBVE)=R*DCOS(TH)
          YFAC(NBVE)=R*DSIN(TH)
          ZFAC(NBVE)=0.0D0
          TH=TH + 2*PI/NBPS
140   CONTINUE
      DO 145 I=1,NBWASH*NBPS
          NBVE=NB2+I
          SAREA(NBVE)=PI*(RFOV*RFOV-RAD1*RAD1)/(NBWASH*NBPS)
          WRITE(10,*)NBVE,SAREA(NBVE)
145   CONTINUE
C
C Compute the surface area of the FOV limiting aperture
C
```

```
      SAREA(NB4+1)=PI*RFOV*DSIN(TETAFM)*RFOV*DSIN(TETAFM)
C
C COMPUTE THE COORDINATES OF THE POINTS AT THE INTERSECTION OF THE
C CONTROL SURFACE FACES FOR THE BARREL
C
      NBVE=NB8+1
      Z=DENIV
      DZ=LBAR/DBLE(NBHSB)
      DO 150 I=1,NBHSB+1
         DO 151 J=1,NBPSB
            TH=(J-1)*2.0D0*PI/DBLE(NBPSB)
            XFAC(NBVE)=RCAV*DCOS(TH)
            YFAC(NBVE)=RCAV*DSIN(TH)
            ZFAC(NBVE)=Z
            NBVE=NBVE+1
151      CONTINUE
         Z=Z-DZ
150   CONTINUE
      NBVE=NB8+1
      DO 152 I=1,(NBHSB*NBPSB)
         SAREA(NBVE)=PI*2.0D0*RCAV*DZ/NBPSB
         NBVE=NBVE+1
152   CONTINUE
C
C COMPUTE THE COORDINATES OF THE POINTS AT THE INTERSECTION OF THE
C CONTROL SURFACE FACES FOR THE CONE
C
      NBVE=NB15+1
      ZZ=DENIV-LBAR
      AREA=PI*RCAV*DSQRT(RCAV**2+LCON**2)
      TANCAV=RCAV/LCON
      H=DSQRT(1.D0+TANCAV**2)
      DO 160 I=1,NBHSC+1
         INDEX=(DBLE(NBHSC)+1.D0-DBLE(I))/DBLE(NBHSC)
         DZ=LCON-DSQRT((AREA*INDEX)/(PI*H*TANCAV))
         Z=ZZ-DZ
         R=(LBAR+LCON-DENIV+Z)*RCAV/LCON
         write(6,*)z
         DO 161 J=1,NBPSC
            TH=(J-1)*2.0D0*PI/DBLE(NBPSC)
            ZFAC(NBVE)=Z
            XFAC(NBVE)=R*DCOS(TH)
            YFAC(NBVE)=R*DSIN(TH)
            NBVE=NBVE+1
161      CONTINUE
160   CONTINUE
      NBVE=NB9+1
      DO 162 I=1,(NBHSC*NBPSC)
          SAREA(NBVE)=AREA/DBLE(NBHSC*NBPSC)
         NBVE=NBVE+1
162   CONTINUE
C
C COMPUTE THE COORDINATES OF THE POINTS AT THE INTERSECTION OF THE
C CONTROL SURFACE FACES FOR THE CAVSUB
C
      NBVE=NB16+1
      DO 170 J=1,NBPSCS
         TH=(J-1)*2.0D0*PI/DBLE(NBPSCS)
         XFAC(NBVE)=RPA*DCOS(TH)
         YFAC(NBVE)=RPA*DSIN(TH)
         ZFAC(NBVE)=-1.D0
         NBVE=NBVE+1
170   CONTINUE
      NBVE=NBVE+NBPSCS+1
      DO 171 J=1,NBPSCS
```

```
         TH=(J-1)*2.0D0*PI/DBLE(NBPSCS)
         XFAC(NBVE)=RCAV*DCOS(TH)
         YFAC(NBVE)=RCAV*DSIN(TH)
         ZFAC(NBVE)=-1.D0
         NBVE=NBVE+1
171   CONTINUE
      NBVE=NB11+1
      DO 172 I=1,NBPSCS
         SAREA(NBVE)=PI*(REXT**2-RPA**2)/DBLE(NBPSCS)
         NBVE=NBVE+1
172   CONTINUE
C
C COMPUTE THE COORDINATES OF THE POINTS AT THE INTERSECTION OF THE
C CONTROL SURFACE FACES FOR THE RING
C
      NBVE=NB17+1
      Z=0.0D0
      DZ=DABS(DENIV/DBLE(NBHSR))
      COSRNG=DSQRT(1.0D0-DENIV*DENIV/REXT*REXT)
      DO 180 I=1,NBHSR+1
         DO 181 J=1,NBPSR
            TH=(J-1)*2.0D0*PI/DBLE(NBPSR)
            XFAC(NBVE)=REXT*DCOS(TH)*COSRNG
            YFAC(NBVE)=REXT*DSIN(TH)*COSRNG
            ZFAC(NBVE)=Z
            NBVE=NBVE+1
181      CONTINUE
         Z=Z-DZ
         COSRNG=1.0D0
180   CONTINUE
      NBVE=NB13+1
      DO 182 I=1,(NBHSR*NBPSR)
         SAREA(NBVE)=PI*2.0D0*REXT*DZ/NBPSR
         NBVE=NBVE+1
182   CONTINUE

      END

*********************************************************************
*                                                                   *
*  THIS SUBROUTINE RETREIVE THE SURFACE OR VOLUME ELEMENT THAT WAS  *
*  HIT FROM X,Y,Z COORDINATES.                                      *
*                                                                   *
*********************************************************************
      SUBROUTINE SEARCH(X,Y,Z,SURFAC,CNTBDL,DIST)

      IMPLICIT NONE
C
      REAL*8 PI,DZED,ZMAX,RFOV,TETAFM,RADIUS,X,Y,Z
      REAL*8 CE,CI,REXT,RINT,DIST,DTETA,DPHI,VOLUM
      REAL*8 RAD1,RAD2,SINTTA,COSTTA,RPA
      REAL*8 DITOCD,ANGPHI,ANGTTA,DENIV,PHIDEN
      REAL*8 RADF,PHIF,TTAF,SAREA,RCEN
      REAL*8 RCAV,LBAR,LCON,CAVANG
      REAL*8 ZBAR,ZCON
      REAL*8 DLCON
      REAL*8 ZRING
C
      INTEGER I,J,K,W,N,SURFAC,MAXELT,NBBUAB,SS,CNTBDL
      INTEGER NB1,NB2,NB3,NB4,NB9,NB11,NB13,NB14
      INTEGER NBLAT,NBLONG,NBWASH,NBPS,NBLSU,NBSH,NBCO,NBPSD
      INTEGER NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS,NBWLSU
      INTEGER NBHSR,NBPSR,SURFC
C
      PARAMETER(MAXELT=700)
```

```
C
      DIMENSION DIST(MAXELT)
C
      COMMON /RIING/ NBHSR,NBPSR
      COMMON /SER1/ NBBUAB(MAXELT) /SER2/ NBLAT,NBLONG
      COMMON /SER3/ NBWASH,NBPS,NBLSU,NBWLSU /SER4/ NBSH,NBCO,NBPSD
      COMMON /FOVL/ RFOV,TETAFM
      COMMON /DOME/ REXT,RINT,DENIV
      COMMON /NB14/ NB1,NB2,NB3,NB4,NB9,NB11,NB13,NB14
      COMMON /SETU/ RADF(MAXELT),PHIF(MAXELT),TTAF(MAXELT),
     &              SAREA(MAXELT),RCEN
      COMMON /CAVITY/ RCAV,LBAR,LCON,CAVANG
      COMMON /CAVSUR/ NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS
      COMMON /CONSET/ DLCON(20)
      COMMON /SUBS/ RPA,SURFC
      PI=DACOS(-1.D0)
      ZMAX=RFOV*DCOS(TETAFM)+DENIV
      CE=DENIV
      CI=DENIV
      RAD1=DSQRT(REXT*REXT-DENIV*DENIV)
      RAD2=RFOV
C
C FIND THE ELEMEMT THAT WAS HIT, ACCORDING TO THE GEOM. PART INVOLVED
C FOR THE FOV LIMITER:
C
      IF (SURFAC.EQ.1.OR.SURFAC.EQ.5) THEN
      DZED=ZMAX/DBLE(NBLAT)
      DTETA=2.D0*PI/DBLE(NBLONG)
        ANGPHI=DACOS(Z/RFOV)
        SINTTA=(Y/RFOV)/DSIN(ANGPHI)
        COSTTA=(X/RFOV)/DSIN(ANGPHI)
        ANGTTA=PI+(DABS(SINTTA)/SINTTA)*(DACOS(COSTTA)-PI)
        I=INT(Z/DZED)+1
        J=INT(ANGTTA/DTETA)+1
        SS=(I-1)*NBLONG+J
C
C FOR THE UPPER SUBSTRATE
C
      ELSE IF (SURFAC.EQ.2) THEN
         DTETA=2.D0*PI/DBLE(NBPS)
         RADIUS=DSQRT(X**2+Y**2)
         SINTTA=Y/RADIUS
         COSTTA=X/RADIUS
         ANGTTA=PI+(DABS(SINTTA)/SINTTA)*(DACOS(COSTTA)-PI)
         I=INT(DBLE(NBWASH)*((RADIUS*RADIUS-RAD1*RAD1)/(RAD2*RAD2-
     &         RAD1*RAD1)))+1
         J=INT(ANGTTA/DTETA)+1
         SS=(NB2)+(I-1)*NBPS+J
C
C FOR THE LOWER SUBSTRATE
C
      ELSE IF (SURFAC.EQ.6) THEN
         DTETA=2.D0*PI/DBLE(NBLSU)
         RADIUS=DSQRT(X**2+Y**2)
         SINTTA=Y/RADIUS
         COSTTA=X/RADIUS
         ANGTTA=PI+(DABS(SINTTA)/SINTTA)*(DACOS(COSTTA)-PI)
         J=INT(ANGTTA/DTETA)+1
         IF (RADIUS.GT.RINT) THEN
            I=2
         ELSE
            I=1
         END IF
         SS=NB1+(I-1)*NBLSU+J
```

```
C
C FOR THE BARREL
C
      ELSE IF (SURFAC.EQ.9.OR.SURFAC.EQ.10) THEN
         ZBAR=Z+LBAR-DENIV
         DTETA=2.D0*PI/DBLE(NBPSB)
         DZED=LBAR/DBLE(NBHSB)
         RADIUS=DSQRT(X**2+Y**2)
         SINTTA=Y/RADIUS
         COSTTA=X/RADIUS
         ANGTTA=PI+(DABS(SINTTA)/SINTTA)*(DACOS(COSTTA)-PI)
         I=(INT(ZBAR/DZED)+1)
         J=(INT(ANGTTA/DTETA)+1)
         SS=NB11+(I-1)*NBPSB+J
C
C FOR THE CONE
C
      ELSE IF (SURFAC.EQ.11.OR.SURFAC.EQ.12) THEN
         DTETA=2.D0*PI/DBLE(NBPSC)
         RADIUS=DSQRT(X**2+Y**2)
         J=0
         DO 5 K=NBHSC+1,1,-1
           J=J+1
           IF (Z.LT.DLCON(K)) THEN
           I=J-1
           GOTO 6
           END IF
5        CONTINUE
6        SINTTA=Y/RADIUS
         COSTTA=X/RADIUS
         ANGTTA=PI+(DABS(SINTTA)/SINTTA)*(DACOS(COSTTA)-PI)
         J=INT(ANGTTA/DTETA)+1
         SS=NB4+1+(I-1)*NBPSC+J
C
C FOR THE SUBSTRATE VISIBLE TO THE CAVITY
C
      ELSE IF (SURFAC.EQ.13) THEN
         DTETA=2.D0*PI/DBLE(NBPSCS)
         RADIUS=DSQRT(X**2+Y**2)
         SINTTA=Y/RADIUS
         COSTTA=X/RADIUS
         ANGTTA=PI+(DABS(SINTTA)/SINTTA)*(DACOS(COSTTA)-PI)
         I=INT(ANGTTA/DTETA)+1
         SS=NB11+I
C
C FOR THE RING
C
      ELSE IF (SURFAC.EQ.14.OR.SURFAC.EQ.15) THEN
         ZRING=-1.D0*Z
         DTETA=2.D0*PI/DBLE(NBPSR)
         DZED=-1.0D0*DENIV/DBLE(NBHSR)
         ANGPHI=DACOS(Z/DSQRT(X*X+Y*Y+Z*Z))
         SINTTA=(Y/(X*X+Y*Y+Z*Z))/DSIN(ANGPHI)
         COSTTA=(X/(X*X+Y*Y+Z*Z))/DSIN(ANGPHI)
         ANGTTA=PI+(DABS(SINTTA)/SINTTA)*(DACOS(COSTTA)-PI)
         I=(INT(ZRING/DZED)+1)
         J=INT(ANGTTA/DTETA)+1
         SS=NB13+(I-1)*NBPSR+J
C
C FOR THE DOME
C
      ELSE
        DTETA=2*PI/DBLE(NBPSD)

C
```

```
C DISTANCE FROM BASE OF DOME TO POINT P (X,Y,Z)
C
        DITOCD=DSQRT(X*X+Y*Y+(Z-DENIV)*(Z-DENIV))
C
C ZENITH ANGLE FROM BASE OF DOME
C
        ANGPHI=DACOS((Z-DENIV)/DITOCD)
C
C AZIMUTHAL ANGLE CORRESPONDING TO X,Y,Z
C
        SINTTA=(Y/DITOCD)/DSIN(ANGPHI)
        COSTTA=(X/DITOCD)/DSIN(ANGPHI)
        ANGTTA=PI+(DABS(SINTTA)/SINTTA)*(DACOS(COSTTA)-PI)
C
C RETRIEVE CONTROL VOLUME INDECES FROM DITOCD, PHI, AND THETA
C
        K=INT(ANGTTA/DTETA)+1
        DO 10 J=2,NBCO+1
            IF (ANGPHI.LT.PHIF(J)) GO TO 1
10      CONTINUE
1       DO 20 I=2,NBSH+1
            IF (DITOCD.GT.DIST(I)) GO TO 2
20      CONTINUE
2       SS=(I-2)*(NBCO*NBPSD)+(J-2)*NBPSD+K+NB3
      END IF
C
C UPDATE THE BUNDLE COUNTER FOR THE ELEMENT WHICH ABSORBED THE LAST
C BUNDLE
C
      NBBUAB(SS)=NBBUAB(SS)+1
      END
****************************************************************
*                                                              *
*  THIS FUNCTION COMPUTES THE FRACTION OF EMISSIVE POWER FOR   *
*  A GIVEN WAVELENGTH INTERVAL USING PLANK'S BLACKBODY         *
*  DISTRIBUTION                                                *
*                                                              *
****************************************************************
      REAL*8 FUNCTION FRAKEMI(LBDA1,LBDA2,T)

      IMPLICIT NONE
      REAL*8 LBDA1,LBDA2,INT,INT1,INT2,T,SUM,X1,X2,X,W
      INTEGER IS,N,I

      PARAMETER (IS=1000)
      DIMENSION X(IS),W(IS)
      EXTERNAL SUM

      N=100
      INT=0.D0
      INT1=0.D0
      INT2=0.D0
C
C COMPUTE THE INTEGRAL FROM 0 TO LBDA1                 *
C
      IF (LBDA1.LE.0.D0) GO TO 7
      DO 10 I = 1,IS
        X(I) = 0.0D0
        W(I) = 0.0D0
10    CONTINUE
      X1=0.D0
      X2=LBDA1*T
      CALL GAULEG(IS,X1,X2,X,W,N)
      INT1=SUM(IS,X,W,N)
C
```

```
C COMPUTE THE INTEGRAL FROM 0 TO LBDA2*
C
7     IF (LBDA2.LE.0.D0) THEN
          WRITE(06,*)'ERROR IN FRAKEMI'
          RETURN
      END IF
      DO 20 I = 1,IS
        X(I) = 0.0D0
        W(I) = 0.0D0
20    CONTINUE
      X1=0.D0
      X2=LBDA2*T
      CALL GAULEG(IS,X1,X2,X,W,N)
      INT2=SUM(IS,X,W,N)
C
C COMPUTE THE INTEGRAL FROM LBDA1 TO LBDA2             *
C
      FRAKEMI=INT2-INT1
      RETURN
      END
**************************************************************
*                                                            *
*  THIS FUNCTION COMPUTES THE POINTS AND WEIGHTS NECESSARY   *
*  TO COMPUTE A NUMERICAL INTEGRAL USING A GAUSS-LEGENDRE    *
*  POLYNOMIAL                                                *
*  COURTESY OF DR. J.R. THOMAS OF THE MECHANICAL ENGINEERING *
*  DEPT. OF VIRGINIA TECH                                    *
*                                                            *
**************************************************************
      SUBROUTINE GAULEG(IS,X1,X2,X,W,N)

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(IS),W(IS)
      PARAMETER (EPS=1.D-15)

      PI=3.141592653589793D0
      M=(N+1)/2
      XM=0.5D0*(X2+X1)
      XL=0.5D0*(X2-X1)
      DO 12 I=1,M
        Z=DCOS(PI*(DBLE(I)-.25D0)/(DBLE(N)+.5D0))
1       CONTINUE
          P1=1.D0
          P2=0.D0
          DO 11 J=1,N
            DJ=DBLE(J)
            P3=P2
            P2=P1
            P1=((DBLE(2.D0*J)-1.D0)*Z*P2-(DJ-1.D0)*P3)/DJ
11        CONTINUE
          PP=DBLE(N)*(Z*P1-P2)/(Z*Z-1.D0)
          Z1=Z
          Z=Z1-P1/PP
        IF(ABS(Z-Z1).GT.EPS)GO TO 1
        X(I)=XM-XL*Z
        X(N+1-I)=XM+XL*Z
        W(I)=2.D0*XL/((1.D0-Z*Z)*PP*PP)
        W(N+1-I)=W(I)
12    CONTINUE
      RETURN
      END

***************************************************************
*                                                             *
*  THIS FUNCTION IS USED BY THE FUNCTION FRAKEMI TO COMPUTE   *
```

```
*  THE FOLLOWING EXPRESSION                                    *
*                                                              *
****************************************************************
      REAL*8 FUNCTION F(X)

      IMPLICIT NONE
      REAL*8 PI,C1,C2,X,SIGMA

      PI=DACOS(-1.0D0)
      C1=0.59544D8
      C2=14388D0
      SIGMA=5.6696D-8
      F=2.d0*PI*C1*DEXP(-C2/X)/(SIGMA*(X**5)*(1.D0-DEXP(-C2/X)))
      RETURN

      END
****************************************************************
*                                                              *
*  THIS FUNCTION IS USED BY THE FUNCTION FRAKEMI TO COMPUTE    *
*  THE FOLLOWING SUM                                           *
*                                                              *
****************************************************************
      REAL*8 FUNCTION SUM(IS,X,W,N)


      IMPLICIT NONE
      REAL*8 X,W,F
      INTEGER IS,N,I
      DIMENSION X(IS),W(IS)
      EXTERNAL F
      SUM=0.D0
      DO 200 I=1,N
          SUM=SUM+F(X(I))*W(I)
200   CONTINUE
      RETURN
      END

****************************************************************
*                                                              *
*  THIS SUBROUTINE OBTAINS A VALUE BY LINEAR INTERPOLATION     *
*                                                              *
****************************************************************
      SUBROUTINE LINREG(LBDA1,LBDAI,LBDA2,SFEP1,SFEPI,SFEP2)

      IMPLICIT NONE
      REAL*8 SFEP1,SFEP2,SFEPI,LBDA1,LBDA2,LBDAI
      REAL*8 A,B

      A=(SFEP2-SFEP1)/(LBDA2-LBDA1)
      B=(SFEP1*LBDA2-SFEP2*LBDA1)/(LBDA2-LBDA1)
      SFEPI=A*LBDAI+B
      RETURN
      END

*************************************************************************
*                                                                       *
* This subroutine distributes the energy bundles over a given spectrum *
* The spectrum can be that of a typical day light Earth scene, or that *
* of a grey surface at a given temperature or just a uniform spectrum  *
* between 0 and 50.0 microns                                           *
* In addition the user can choose to emit in the shortwave band of the *
* previously defined spectrum or over the entire spectrum.             *
*                                                                       *
*************************************************************************
```

```
      SUBROUTINE WLSET(MAXBDL,EMISUR)

      IMPLICIT NONE
C
      REAL*8 LBDAMX,LBDA,LBDA1,LBDA2
      REAL*8 FEP,FEP1,FEP2,SFEP,SFEP1,SFEP2
      REAL*8 TEMP,FRAKEMI,FRAK
      REAL*8 DOABCO,DOMTHC
C
      INTEGER I,J,ANSWER,CH
      INTEGER NBWLIN,MAXBDL,NBDLIN,NBBUEA
      INTEGER MAXWLI,EMISUR
C
      PARAMETER (MAXWLI=15)
      DIMENSION SFEP(MAXWLI)
      EXTERNAL FRAKEMI
C
      COMMON /WLS1/ NBDLIN(MAXWLI),NBBUEA(MAXWLI),NBWLIN
      COMMON /WLS2/ DOABCO(MAXWLI),LBDA(MAXWLI)
      COMMON /CHAN/ CH
C
C
C  Use the spectrum of a surface at temperature T
C
      NBBUEA(1)  = IDNINT(DBLE(MAXBDL)/DBLE(NBWLIN))
      NBDLIN(1)  = NBBUEA(1)
      DO 45 I=2,NBWLIN
       NBBUEA(I)  = IDNINT(DBLE(MAXBDL)/DBLE(NBWLIN))
       NBDLIN(I)  = NBDLIN(I-1) + NBBUEA(I)
   45 CONTINUE
C
C  Write the spectral distribution of energy bundles to the sceen
C
      WRITE(03,520)'=================================================='
      WRITE(03,520)'  SPECTRAL DISTRIBUTION OF EMITTED ENERGY BUNDLES  '
      WRITE(03,520)'                                                  '
      WRITE(03,520)' WAVELENGTH |    PARTIAL   | CUMULATIVE | ABSORPTION '
      WRITE(03,520)'  INTERVAL  | NUMBER OF ENERGY BUNDLES| COEFFICIENT'
      WRITE(03,520)'--------------------------------------------------'
      WRITE(03,521)0.0,LBDA(1),NBBUEA(1),NBDLIN(1),DOABCO(1)
      DO 90 I=2,NBWLIN
        WRITE(03,521)LBDA(I-1),LBDA(I),NBBUEA(I),NBDLIN(I),DOABCO(I)
90    CONTINUE
      WRITE(03,520)'=================================================='
C
520   FORMAT(1X,A52)
521   FORMAT(1X,F5.2,'-',F5.2,'  |',I10,'  |',I10,'  |',G10.3,3X)
      RETURN
      END

*****************************************************
*                                                   *
*            SUBROUTINE RNSET                       *
*                                                   *
* This subroutine must be run before RNUN to set    *
* the seed for the random number generator.  The    *
* seed must be an integer in the range              *
* (0,2147483647).                                   *
*                                                   *
*****************************************************
*
      SUBROUTINE RNSET(ISEED)
C
C Set up variables.
C
```

```
      REAL*8 X1,A,P
      INTEGER ISEED
C
C Create a common storage block for the random number data.
C
      COMMON /RANDOM/ X1,A,P
C
C Set up constants used in RNUN.
C
      A = 16807.0D0
      P = 2147483647.0D0
C
C Set up seed for first random number.
C
      X1 = DBLE(ISEED)

      RETURN
      END
*****************************************************
*                                                   *
*             SUBROUTINE DRNUN                      *
*                                                   *
* This subroutine returns the same pseudo-random    *
* numbers as the IMSL routine of the same name,     *
* provided it is set with the same seed value.      *
* Developed by Rob Bongiovi and Martial Haeffelin   *
* at VPI&SU, April 1993.                            *
*                                                   *
*****************************************************
*
      SUBROUTINE DRNUN(NR, R)
C
C Set up variables.
C
      INTEGER NR
      REAL*8 X1, X2, A, P, R(NR)
C
C Use common block created by RNSET
C
      COMMON /RANDOM/ X1,A,P
C
C Calculate NR random numbers.
C
      DO 1 I=1,NR
        X2 = DMOD(X1*A,P)
        R(I) = X2/P
        X1 = X2
1     CONTINUE

      RETURN
      END
*
****************************************************************
*                                                              *
*       THIS SUBROUTINE OUTPUTS DATA FOR TECPLOT               *
*       USES WRITE STATEMENTS TO FILE EXTENSION 8,9,10...      *
*                                                              *
****************************************************************
*
      SUBROUTINE TECPLOT

      REAL*8 XFAC,YFAC,ZFAC
      REAL*8 RADF,PHIF,TTAF,SAREA,RCEN
      REAL*8 REXT,RINT,DENIV
      REAL*8 DUMSUM
```

```
      REAL*8 RCAV,LBAR,LCON,CAVANG
      REAL*4 TH,XDUM,YDUM
      REAL*4 DFACTT,DFACSW,DFACLW
C
      INTEGER NBLAT,NBLONG,NBLSU,NBWASH,NBPS,NBSH,NBCO,NBPSD
      INTEGER NB1,NB2,NB3,NB4,NB5,NB6,NB7,NB9,NB11,NB13
      INTEGER NB14,NB15,NB16,NB8
      INTEGER I,J,K,MAXELT
      INTEGER NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS,NBWLSU
C
      PARAMETER (MAXELT=700)
C
      COMMON /TECP/ DFACTT(1,MAXELT),DFACSW(1,MAXELT),DFACLW(1,MAXELT)
      COMMON /SER2/ NBLAT,NBLONG
      COMMON /SER3/ NBWASH,NBPS,NBLSU,NBWLSU
      COMMON /SER4/ NBSH,NBCO,NBPSD
      COMMON /NB14/ NB1,NB2,NB3,NB4,NB9,NB11,NB13,NB14
      COMMON /NB57/ NB5,NB6,NB7,NB15,NB16,NB8,NB17,NB18
      COMMON /FACE/ XFAC(MAXELT),YFAC(MAXELT),ZFAC(MAXELT)
      COMMON /DOME/ REXT,RINT,DENIV
      COMMON /SETU/ RADF(MAXELT),PHIF(MAXELT),ttaf(maxelt),
     &              SAREA(MAXELT),RCEN
      COMMON /CAVITY/ RCAV,LBAR,LCON,CAVANG
      COMMON /CAVSUR/ NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS
C
C OUTPUT OF TOTAL DFAC TO FOV LIMITER
C
      WRITE(09,*)'TITLE = "EMISSION FROM EARTH SCENE; IMAGE ON DOME"'
      WRITE(09,*)'VARIABLES = X, Y, Z, DFAC'
      WRITE(09,*)'ZONE T="FOV",I=25, J=7, F=POINT'
      I=1
      J=1
      DUMSUM=(DFACTT(I,NBLONG)+DFACTT(I,J))/2.D0
      WRITE(09,700)XFAC(J),YFAC(J),ZFAC(J),DUMSUM
      DO 400 J=2,NBLONG
          DUMSUM=(DFACTT(I,J-1)+DFACTT(I,J))/2.D0
          WRITE(09,700)XFAC(J),YFAC(J),ZFAC(J),DUMSUM
400   CONTINUE
      J=1
      DUMSUM=(DFACTT(I,NBLONG)+DFACTT(I,J))/2.D0
      WRITE(09,700)XFAC(J),YFAC(J),ZFAC(J),DUMSUM
      DO 401 K=2,NBLAT
          J=(K-1)*NBLONG+1
          DUMSUM=(DFACTT(I,J+NBLONG-1)+
     1                  DFACTT(I,J)+
     2                  DFACTT(I,J-NBLONG)+
     3                  DFACTT(I,J-1))/4.0
          WRITE(09,700)XFAC(J),YFAC(J),ZFAC(J),DUMSUM
      DO 402 J=(K-1)*NBLONG+2,K*NBLONG
          DUMSUM=(DFACTT(I,J-1)+
     1                  DFACTT(I,J)+
     2                  DFACTT(I,J-NBLONG)+
     3                  DFACTT(I,J-NBLONG-1))/4
          WRITE(09,700)XFAC(J),YFAC(J),ZFAC(J),DUMSUM
402   CONTINUE
          J=(K-1)*NBLONG+1
          DUMSUM=(DFACTT(I,J+NBLONG-1)+
     1                  DFACTT(I,J)+
     2                  DFACTT(I,J-NBLONG)+
     3                  DFACTT(I,J-1))/4.0
          WRITE(09,700)XFAC(J),YFAC(J),ZFAC(J),DUMSUM
401   CONTINUE
      J=NBLAT*NBLONG+1
      DUMSUM=(DFACTT(I,J-1)+DFACTT(I,J-NBLONG))/2.D0
      WRITE(09,700)XFAC(J),YFAC(J),ZFAC(J),DUMSUM
```

```
      DO 403 J=NBLAT*NBLONG+2,(NBLAT+1)*NBLONG
          DUMSUM=(DFACTT(I,J-NBLONG-1)+DFACTT(I,J-NBLONG))/2.D0
          WRITE(09,700)XFAC(J),YFAC(J),ZFAC(J),DUMSUM
403   CONTINUE
      J=NBLAT*NBLONG+1
      DUMSUM=(DFACTT(I,J-1)+DFACTT(I,J-NBLONG))/2.D0
      WRITE(09,700)XFAC(J),YFAC(J),ZFAC(J),DUMSUM
C
C OUTPUT OF TOTAL DFAC TO THE LOWER SUBSTRATE
C
      NB5=(NBLAT+1)*NBLONG
      WRITE(09,*)'ZONE T="SUBLOW",I=25, J=3, F=POINT'
      I=1
      J=1
      DUMSUM=(DFACTT(I,NB1)+
     1        DFACTT(I,NB1+J)+
     2        DFACTT(I,NB1+NBLSU))/3.D0
          WRITE(09,700)XFAC(NB5+J),YFAC(NB5+J),ZFAC(NB5+J),DUMSUM
      DO 404 J=2,NBLSU+1
          DUMSUM=(DFACTT(I,NB1)+
     1            DFACTT(I,NB1+J)+
     2            DFACTT(I,NB1+J-1))/3.D0
          WRITE(09,700)XFAC(NB5+J),YFAC(NB5+J),ZFAC(NB5+J),DUMSUM
404   CONTINUE
      J=1
      DUMSUM=(DFACTT(I,NB1)+
     1        DFACTT(I,NB1+J)+
     2        DFACTT(I,NB1+NBLSU))/3.D0
          WRITE(09,700)XFAC(NB5+J),YFAC(NB5+J),ZFAC(NB5+J),DUMSUM
      J=NBLSU+1
      DUMSUM=(DFACTT(I,NB1+J-1)+DFACTT(I,NB1+J-NBLSU))/2.D0
      WRITE(09,700)XFAC(NB5+J),YFAC(NB5+J),ZFAC(NB5+J),DUMSUM
      DO 405 J=NBLSU+2,2*NBLSU
          DUMSUM=(DFACTT(I,NB1+J-NBLSU)+DFACTT(I,NB1+J-NBLSU))/2.D0
          WRITE(09,700)XFAC(NB5+J),YFAC(NB5+J),ZFAC(NB5+J),DUMSUM
405   CONTINUE
      J=NBLSU+1
      DUMSUM=(DFACTT(I,NB1+J-1)+DFACTT(I,NB1+J-NBLSU))/2.D0
      WRITE(09,700)XFAC(NB5+J),YFAC(NB5+J),ZFAC(NB5+J),DUMSUM
C
C OUTPUT OF TOTAL DFAC TO THE UPPER SUBSTRATE
C
      NB6=NB5 + (NBWLSU+1)*NBLSU
      WRITE(09,*)'ZONE T="SUBUPP",I=25, J=7, F=POINT'
      I=1
      J=1
      DUMSUM=(DFACTT(I,NB2+NBPS)+DFACTT(I,NB2+J))/2.D0
      WRITE(09,700)XFAC(NB6+J),YFAC(NB6+J),ZFAC(NB6+J),DUMSUM
      DO 406 J=2,NBPS
          DUMSUM=(DFACTT(I,NB2+J-1)+DFACTT(I,NB2+J))/2.D0
          WRITE(09,700)XFAC(NB6+J),YFAC(NB6+J),ZFAC(NB6+J),DUMSUM
406   CONTINUE
      J=1
      DUMSUM=(DFACTT(I,NB2+NBPS)+DFACTT(I,NB2+J))/2.D0
      WRITE(09,700)XFAC(NB6+J),YFAC(NB6+J),ZFAC(NB6+J),DUMSUM
      DO 407 K=2,NBWASH
          J=(K-1)*NBPS+1
          DUMSUM=(DFACTT(I,NB2+J-1)+
     1            DFACTT(I,NB2+J-NBPS)+
     2            DFACTT(I,NB2+J)+
     3            DFACTT(I,NB2+J+NBPS-1))/4.0
          WRITE(09,700)XFAC(NB6+J),YFAC(NB6+J),ZFAC(NB6+J),DUMSUM
      DO 408 J=(K-1)*NBPS+2,K*NBPS
          DUMSUM=(DFACTT(I,NB2+J-1-NBPS)+
     1            DFACTT(I,NB2+J-NBPS)+
```

```
     2            DFACTT(I,NB2+J)+
     3            DFACTT(I,NB2+J-1))/4
          WRITE(09,700)XFAC(NB6+J),YFAC(NB6+J),ZFAC(NB6+J),DUMSUM
408   CONTINUE
          J=(K-1)*NBPS+1
          DUMSUM=(DFACTT(I,NB2+J-1)+
     1            DFACTT(I,NB2+J-NBPS)+
     2            DFACTT(I,NB2+J)+
     3            DFACTT(I,NB2+J+NBPS-1))/4.0
          WRITE(09,700)XFAC(NB6+J),YFAC(NB6+J),ZFAC(NB6+J),DUMSUM
407   CONTINUE
      J=NBWASH*NBPS+1
      DUMSUM=(DFACTT(I,NB2+J-1)+DFACTT(I,NB2+J-NBPS))/2.D0
      WRITE(09,700)XFAC(NB6+J),YFAC(NB6+J),ZFAC(NB6+J),DUMSUM
      DO 409 J=NBWASH*NBPS+2,(NBWASH+1)*NBPS
          DUMSUM=(DFACTT(I,NB2+J-NBPS-1)+DFACTT(I,NB2+J-NBPS))/2.D0
          WRITE(09,700)XFAC(NB6+J),YFAC(NB6+J),ZFAC(NB6+J),DUMSUM
409   CONTINUE
      J=NBWASH*NBPS+1
      DUMSUM=(DFACTT(I,NB2+J-1)+DFACTT(I,NB2+J-NBPS))/2.D0
      WRITE(09,700)XFAC(NB6+J),YFAC(NB6+J),ZFAC(NB6+J),DUMSUM
C
C OUTPUT OF SHORTWAVE DFAC TO DOME
C
      NB7=NB6+(NBWASH+1)*NBPS
      WRITE(09,*)'ZONE T="SW",I=25, J=10, F=POINT'
      I=1
      DUMSUM=0.0
      DO 410 J=1,NBPSD
          DUMSUM=DUMSUM+DFACSW(I,NB3+J)
410   CONTINUE
      DUMSUM=DUMSUM/REAL(NBPSD)
      DO 411 J=1,NBPSD+1
          TH=REAL(J-1)*2.0*PI/REAL(NBPSD)
          XDUM=0.001*COS(TH)
          YDUM=0.001*SIN(TH)
          WRITE(09,700)XDUM,YDUM,RADF(I)+DENIV,DUMSUM
411   CONTINUE
      DO 412 K=1,NBCO
          J=(K-1)*NBPSD+2
          DUMSUM=(DFACSW(I,NB3+J-1)+
     1            DFACSW(I,NB3+J+NBPSD-1)+
     2            DFACSW(I,NB3+J+NBPSD-2)+
     3            DFACSW(I,NB3+J+2*NBPSD-2))/4.0
          WRITE(09,700)XFAC(NB7+J),YFAC(NB7+J),ZFAC(NB7+J),DUMSUM
      DO 413 J=(K-1)*NBPSD+3,K*NBPSD+1
          DUMSUM=(DFACSW(I,NB3+J-1)+
     1            DFACSW(I,NB3+J-2)+
     2            DFACSW(I,NB3+J+NBPSD-1)+
     3            DFACSW(I,NB3+J+NBPSD-2))/4
          WRITE(09,700)XFAC(NB7+J),YFAC(NB7+J),ZFAC(NB7+J),DUMSUM
413   CONTINUE
          J=(K-1)*NBPSD+2
          DUMSUM=(DFACSW(I,NB3+J-1)+
     1            DFACSW(I,NB3+J+NBPSD-1)+
     2            DFACSW(I,NB3+J+NBPSD-2)+
     3            DFACSW(I,NB3+J+2*NBPSD-2))/4.0
          WRITE(09,700)XFAC(NB7+J),YFAC(NB7+J),ZFAC(NB7+J),DUMSUM
412   CONTINUE
C
C OUTPUT OF LONGWAVE DFAC TO DOME
C
      WRITE(09,*)'ZONE T="LW",I=25, J=10, F=POINT'
      I=1
      DUMSUM=0.0
```

```
      DO 420 J=1,NBPSD
          DUMSUM=DUMSUM+DFACLW(I,NB3+J)
420   CONTINUE
      DUMSUM=DUMSUM/REAL(NBPSD)
      DO 421 J=1,NBPSD+1
          TH=REAL(J-1)*2.0*PI/REAL(NBPSD)
          XDUM=0.001*COS(TH)
          YDUM=0.001*SIN(TH)
          WRITE(09,700)XDUM,YDUM,RADF(I)+DENIV,DUMSUM
421   CONTINUE
      DO 422 K=1,NBCO
          J=(K-1)*NBPSD+2
          DUMSUM=(DFACLW(I,NB3+J-1)+
     1                  DFACLW(I,NB3+J+NBPSD-1)+
     2                  DFACLW(I,NB3+J+NBPSD-2)+
     3                  DFACLW(I,NB3+J+2*NBPSD-2))/4.0
          WRITE(09,700)XFAC(NB7+J),YFAC(NB7+J),ZFAC(NB7+J),DUMSUM
      DO 423 J=(K-1)*NBPSD+3,K*NBPSD+1
          DUMSUM=(DFACLW(I,NB3+J-1)+
     1                  DFACLW(I,NB3+J-2)+
     2                  DFACLW(I,NB3+J+NBPSD-1)+
     3                  DFACLW(I,NB3+J+NBPSD-2))/4
          WRITE(09,700)XFAC(NB7+J),YFAC(NB7+J),ZFAC(NB7+J),DUMSUM
423   CONTINUE
          J=(K-1)*NBPSD+2
          DUMSUM=(DFACLW(I,NB3+J-1)+
     1                  DFACLW(I,NB3+J+NBPSD-1)+
     2                  DFACLW(I,NB3+J+NBPSD-2)+
     3                  DFACLW(I,NB3+J+2*NBPSD-2))/4.0
          WRITE(09,700)XFAC(NB7+J),YFAC(NB7+J),ZFAC(NB7+J),DUMSUM
422   CONTINUE
C
C OUTPUT OF TOTAL DFAC OF FIRST SHELL OF THE DOME
C
      WRITE(09,*)'ZONE T="TT2",I=25, J=10, F=POINT'
      DUMSUM=0.0
      I=1
      DO 440 J=1,NBPSD
          NBVE= J + (I-1)*NBCO*NBPSD + NB3
          DUMSUM=DUMSUM+DFACTT(1,NBVE)
440   CONTINUE
      DUMSUM=DUMSUM/REAL(NBPSD)
      DO 441 J=1,NBPSD+1
          TH=REAL(J-1)*2.0*PI/REAL(NBPSD)
          XDUM=0.001*COS(TH)
          YDUM=0.001*SIN(TH)
          WRITE(09,700)XDUM,YDUM,RADF(I)+DENIV,DUMSUM
441   CONTINUE
      DO 442 K=1,NBCO
          J=(K-1)*NBPSD + 2 + (I-1)*(NBCO*NBPSD+1)
          DUMSUM=(DFACTT(I,NB3+J-1)+
     1                  DFACTT(I,NB3+J+NBPSD-1)+
     2                  DFACTT(I,NB3+J+NBPSD-2)+
     3                  DFACTT(I,NB3+J+2*NBPSD-2))/4.0
          WRITE(09,700)XFAC(NB7+J),YFAC(NB7+J),ZFAC(NB7+J),DUMSUM
      DO 443 J=(K-1)*NBPSD + 3 + (I-1)*(NBCO*NBPSD+1),
     &          K*NBPSD + 1 + (I-1)*(NBCO*NBPSD+1)
          DUMSUM=(DFACTT(I,NB3+J-1)+
     1                  DFACTT(I,NB3+J-2)+
     2                  DFACTT(I,NB3+J+NBPSD-1)+
     3                  DFACTT(I,NB3+J+NBPSD-2))/4
          WRITE(09,700)XFAC(NB7+J),YFAC(NB7+J),ZFAC(NB7+J),DUMSUM
443   CONTINUE
          J=(K-1)*NBPSD + 2 + (I-1)*(NBCO*NBPSD+1)
          DUMSUM=(DFACTT(I,NB3+J-1)+
```

```
     1          DFACTT(I,NB3+J+NBPSD-1)+
     2          DFACTT(I,NB3+J+NBPSD-2)+
     3          DFACTT(I,NB3+J+2*NBPSD-2))/4.0
          WRITE(09,700)XFAC(NB7+J),YFAC(NB7+J),ZFAC(NB7+J),DUMSUM
442   CONTINUE
C
C OUTPUT OF TOTAL DFAC TO BARREL
C
      WRITE(09,*)'ZONE T="BARREL",I=25, J=5, F=POINT'
      I=1
      J=1
      DUMSUM=(DFACTT(I,NB4+1+NBPSB)+DFACTT(I,NB4+1+J))/2.D0
      WRITE(09,700)XFAC(NB8+J),YFAC(NB8+J),ZFAC(NB8+J),DUMSUM
      DO 460 J=2,NBPSB
          DUMSUM=(DFACTT(I,NB4+1+J-1)+DFACTT(I,NB4+1+J))/2.D0
          WRITE(09,700)XFAC(NB8+J),YFAC(NB8+J),ZFAC(NB8+J),DUMSUM
460   CONTINUE
      J=1
      DUMSUM=(DFACTT(I,NB4+1+NBPSB)+DFACTT(I,NB4+1+J))/2.D0
      WRITE(09,700)XFAC(NB8+J),YFAC(NB8+J),ZFAC(NB8+J),DUMSUM
      DO 461 K=2,NBHSB
          J=(K-1)*NBPSB+1
          DUMSUM=(DFACTT(I,NB4+1+J+NBPSB-1)+
     1          DFACTT(I,NB4+1+J)+
     2          DFACTT(I,NB4+1+J-NBPSB)+
     3          DFACTT(I,NB4+1+J-1))/4.0
          WRITE(09,700)XFAC(NB8+J),YFAC(NB8+J),ZFAC(NB8+J),DUMSUM
      DO 462 J=(K-1)*NBPSB+2,K*NBPSB
          DUMSUM=(DFACTT(I,NB4+1+J-1)+
     1          DFACTT(I,NB4+1+J)+
     2          DFACTT(I,NB4+1+J-NBPSB)+
     3          DFACTT(I,NB4+1+J-NBPSB-1))/4
          WRITE(09,700)XFAC(NB8+J),YFAC(NB8+J),ZFAC(NB8+J),DUMSUM
462   CONTINUE
          J=(K-1)*NBPSB+1
          DUMSUM=(DFACTT(I,NB4+1+J+NBPSB-1)+
     1          DFACTT(I,NB4+1+J)+
     2          DFACTT(I,NB4+1+J-NBPSB)+
     3          DFACTT(I,NB4+1+J-1))/4.0
          WRITE(09,700)XFAC(NB8+J),YFAC(NB8+J),ZFAC(NB8+J),DUMSUM
461   CONTINUE
      J=NBHSB*NBPSB+1
      DUMSUM=(DFACTT(I,NB4+1+J-1)+DFACTT(I,NB4+1+J-NBPSB))/2.D0
      WRITE(09,700)XFAC(NB8+J),YFAC(NB8+J),ZFAC(NB8+J),DUMSUM
      DO 463 J=NBHSB*NBPSB+2,(NBHSB+1)*NBPSB
         DUMSUM=(DFACTT(I,NB4+1+J-NBPSB-1)+DFACTT(I,NB4+1+J-NBPSB))/2.D0
         WRITE(09,700)XFAC(NB8+J),YFAC(NB8+J),ZFAC(NB8+J),DUMSUM
463   CONTINUE
      J=NBHSB*NBPSB+1
      DUMSUM=(DFACTT(I,NB4+1+J-1)+DFACTT(I,NB4+1+J-NBPSB))/2.D0
      WRITE(09,700)XFAC(NB8+J),YFAC(NB8+J),ZFAC(NB8+J),DUMSUM
C
C OUTPUT OF TOTAL DFAC TO CONE
C
      WRITE(09,*)'ZONE T="CONE",I=25, J=7, F=POINT'
      I=1
      J=1
      DUMSUM=(DFACTT(I,NB9+NBPSC)+DFACTT(I,NB9+J))/2.D0
      WRITE(09,700)XFAC(NB15+J),YFAC(NB15+J),ZFAC(NB15+J),DUMSUM
      DO 464,J=2,NBPSC
          DUMSUM=(DFACTT(I,NB9+J-1)+DFACTT(I,NB9+J))/2.D0
          WRITE(09,700)XFAC(NB15+J),YFAC(NB15+J),ZFAC(NB15+J),DUMSUM
464   CONTINUE
      J=1
      DUMSUM=(DFACTT(I,NB9+NBPSC)+DFACTT(I,NB9+J))/2.D0
```

```
      WRITE(09,700)XFAC(NB15+J),YFAC(NB15+J),ZFAC(NB15+J),DUMSUM
      DO 465 K=2,NBHSC
          J=(K-1)*NBPSC+1
          DUMSUM=(DFACTT(I,NB9+J+NBPSC-1)+
     1            DFACTT(I,NB9+J)+
     2            DFACTT(I,NB9+J-NBPSC)+
     3            DFACTT(I,NB9+J-1))/4.0
          WRITE(09,700)XFAC(NB15+J),YFAC(NB15+J),ZFAC(NB15+J),DUMSUM
      DO 466 J=(K-1)*NBPSC+2,K*NBPSC
          DUMSUM=(DFACTT(I,NB9+J-1)+
     1            DFACTT(I,NB9+J)+
     2            DFACTT(I,NB9+J-NBPSC)+
     3            DFACTT(I,NB9+J-NBPSC-1))/4
          WRITE(09,700)XFAC(NB15+J),YFAC(NB15+J),ZFAC(NB15+J),DUMSUM
466   CONTINUE
          J=(K-1)*NBPSC+1
          DUMSUM=(DFACTT(I,NB9+J+NBPSC-1)+
     1            DFACTT(I,NB9+J)+
     2            DFACTT(I,NB9+J-NBPSC)+
     3            DFACTT(I,NB9+J-1))/4.0
          WRITE(09,700)XFAC(NB15+J),YFAC(NB15+J),ZFAC(NB15+J),DUMSUM
465   CONTINUE
      J=NBHSC*NBPSC+1
      DUMSUM=(DFACTT(I,NB9+J-1)+DFACTT(I,NB9+J-NBPSC))/2.D0
      WRITE(09,700)XFAC(NB15+J),YFAC(NB15+J),ZFAC(NB15+J),DUMSUM
      DO 467 J=NBHSC*NBPSC+2,(NBHSC+1)*NBPSC
          DUMSUM=(DFACTT(I,NB9+J-NBPSC-1)+DFACTT(I,NB9+J-NBPSC))/2.D0
          WRITE(09,700)XFAC(NB15+J),YFAC(NB15+J),ZFAC(NB15+J),DUMSUM
467   CONTINUE
      J=NBHSC*NBPSC+1
      DUMSUM=(DFACTT(I,NB9+J-1)+DFACTT(I,NB9+J-NBPSC))/2.D0
      WRITE(09,700)XFAC(NB15+J),YFAC(NB15+J),ZFAC(NB15+J),DUMSUM
C
C OUTPUT OF TOTAL DFAC TO THE PORTION OF THE SUBSTRATE WHICH
C IS VISIBLE TO THE CAVITY (CAVSUB)
C
      WRITE(09,*)'ZONE T="CAVSUB",I=25, J=2, F=POINT'
      I=1
      J=1
      DUMSUM=(DFACTT(I,NB11+NBPSCS)+DFACTT(I,NB11+J))/2.D0
      WRITE(09,700)XFAC(NB16+J),YFAC(NB16+J),ZFAC(NB16+J),DUMSUM
      DO 468 J=2,NBPSCS
          DUMSUM=(DFACTT(I,NB11+J-1)+DFACTT(I,NB11+J))/2.D0
          WRITE(09,700)XFAC(NB16+J),YFAC(NB16+J),ZFAC(NB16+J),DUMSUM
468   CONTINUE
      J=NBPSCS+1
      DUMSUM=(DFACTT(I,NB11+J-1)+DFACTT(I,NB11+J-NBPSCS))/2.0D0
      WRITE(09,700)XFAC(NB16+J),YFAC(NB16+J),ZFAC(NB16+J),DUMSUM
      DO 469 J=NBPSCS+1,2*NBPSCS
         DUMSUM=(DFACTT(I,NB11+J-1)+DFACTT(I,NB11+J-NBPSCS))/2.0D0
         WRITE(09,700)XFAC(NB16+J),YFAC(NB16+J),ZFAC(NB16+J),DUMSUM
469   CONTINUE
      J=NBPSCS+1
      DUMSUM=(DFACTT(I,NB11+J-1)+DFACTT(I,NB11+J-NBPSCS))/2.0D0
      WRITE(09,700)XFAC(NB16+J),YFAC(NB16+J),ZFAC(NB16+J),DUMSUM
C
C FOR THE TOTAL CHANNEL ADD A LITTLE RING
C
      WRITE(09,*)'ZONE T="RING",I=25, J=1, F=POINT'
      I=1
      J=1
      DUMSUM=(DFACTT(I,NB13+NBSPR)+DFACTT(I,NB13+J))/2.D0
      WRITE(09,700)XFAC(NB17+J),YFAC(NB17+J),ZFAC(NB17+J),DUMSUM
      DO 450 J=2,NBPSR
          DUMSUM=(DFACTT(I,NB13+J-1)+DFACTT(I,NB13+J))/2.D0
```

```
          WRITE(09,700)XFAC(NB17+J),YFAC(NB17+J),ZFAC(NB17+J),DUMSUM
450   CONTINUE
      J=1
      DUMSUM=(DFACTT(I,NB13+NBPSR)+DFACTT(I,NB13+J))/2.D0
      WRITE(09,700)XFAC(NB17+J),YFAC(NB17+J),ZFAC(NB17+J),DUMSUM
      DO 451 K=2,NBHSR
          J=(K-1)*NBPSR+1
          DUMSUM=(DFACTT(I,NB13+J+NBPSR-1)+
     1              DFACTT(I,NB13+J)+
     2              DFACTT(I,NB13+J-NBPSR)+
     3              DFACTT(I,NB13+J-1))/4.0
          WRITE(09,700)XFAC(NB17+J),YFAC(NB17+J),ZFAC(NB17+J),DUMSUM
      DO 452 J=(K-1)*NBPSR+2,K*NBPSR
          DUMSUM=(DFACTT(I,NB13+J-1)+
     1              DFACTT(I,NB13+J)+
     2              DFACTT(I,NB13+J-NBPSR)+
     3              DFACTT(I,NB13+J-NBPSR-1))/4
          WRITE(09,700)XFAC(NB17+J),YFAC(NB17+J),ZFAC(NB17+J),DUMSUM
452   CONTINUE
          J=(K-1)*NBPSR+1
          DUMSUM=(DFACTT(I,NB13+J+NBPSR-1)+
     1              DFACTT(I,NB13+J)+
     2              DFACTT(I,NB13+J-NBPSR)+
     3              DFACTT(I,NB13+J-1))/4.0
          WRITE(09,700)XFAC(NB17+J),YFAC(NB17+J),ZFAC(NB17+J),DUMSUM
451   CONTINUE
      J=NBHSR*NBPSR+1
      DUMSUM=(DFACTT(I,NB13+J-1)+DFACTT(I,NB13+J-NBPSR))/2.D0
      WRITE(09,700)XFAC(NB17+J),YFAC(NB17+J),ZFAC(NB17+J),DUMSUM
      DO 453 J=NBHSR*NBPSR+2,(NBHSR+1)*NBPSR
         DUMSUM=(DFACTT(I,NB13+J-NBPSR-1)+DFACTT(I,NB13+J-NBPSR))/2.D0
         WRITE(09,700)XFAC(NB17+J),YFAC(NB17+J),ZFAC(NB17+J),DUMSUM
453   CONTINUE
      J=NBHSR*NBPSR+1
      DUMSUM=(DFACTT(I,NB13+J-1)+DFACTT(I,NB13+J-NBPSR))/2.D0
      WRITE(09,700)XFAC(NB17+J),YFAC(NB17+J),ZFAC(NB17+J),DUMSUM
C
700   FORMAT(1X,F9.3,1X,F9.3,1X,F9.3,1X,G20.9)
C
      END

****************************************************************************
*                                                                          *
*     THIS SUBROUTINE DIVIDES THE CONE HORIZONTALLY SO THAT THE            *
*     SURFACE ELEMENTS ARE ALL EQUAL IN SIZE.                              *
*                                                                          *
****************************************************************************

      SUBROUTINE CONSETUP(DLCON)

      IMPLICIT NONE
      REAL*8 AREA,PI,DZ,Z,ZZ,H,TANCAV,INDEX
      REAL*8 RCAV,LBAR,LCON,CAVANG
      REAL*8 REXT,RINT,DENIV
      REAL*8 DLCON

      INTEGER I
      INTEGER NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS


      DIMENSION DLCON(20)

      COMMON /CAVSUR/ NBHSB,NBHSC,NBPSB,NBPSC,NBPSCS
      COMMON /CAVITY/ RCAV,LBAR,LCON,CAVANG
      COMMON /DOME/ REXT,RINT,DENIV
```

```
      PI=DACOS(-1.D0)
      AREA=PI*RCAV*DSQRT(RCAV**2+LCON**2)
      TANCAV=RCAV/LCON
      H=DSQRT(1.D0+TANCAV**2)
      ZZ=DENIV-LBAR
      DO 100 I=1,NBHSC+1
        INDEX=(DBLE(NBHSC)+1.D0-DBLE(I))/DBLE(NBHSC)
        DZ=LCON-DSQRT((AREA*INDEX)/(PI*H*TANCAV))
        DLCON(I)=ZZ-DZ
100   CONTINUE
      RETURN
      END
```

Appendix B

Program ERBER1

```
***********************************************************************
*                         ===============                             *
*                        | ERBER1 FORTRAN |                           *
*                         ===============                             *
*    This FORTRAN 77 program prompts the user for mesh geometry data  *
*    and then writes the data to input files for the mesh generator   *
*    and thermal analysis programs.  The mesh geometry data is also   *
*    tabulated for user reference.                                    *
*                                                                     *
*    Written by Nour E. Tira, and modified by Kory J. Priestley       *
*_____________________________________________________________________*
*                                                                     *
*                            NOMENCLATURE                             *
*_____________________________________________________________________*
*                                                                     *
*    ATIP ....... Area of the truncated tip of the cone.              *
*    DIM(I) ..... Radiometer part dimension.                          *
*    DM(I) ...... The element dimensions along dimension I if         *
*                 NUNIF(I) is given as zero.                          *
*    HO ......... Height of the cone tip.                             *
*    MDIV(I) .... Number of element divisions along dimension I.      *
*    NUNIF(I) ... Mesh uniformity flag for dimension I.  If its       *
*                 value is zero, then elements along dimension I      *
*                 are spaced unequally and the element dimensions     *
*                 must be input by the user.  If its value is one,    *
*                 then elements along dimension I are equally         *
*                 spaced and further mesh computation is done by      *
*                 the mesh generator.                                 *
*    N1 ......... The number of elements around the circumference     *
*                 of the radiometer cavity.                           *
*    RI ......... The inside radius of the ring (cavity radius).      *
*    RO ......... The outside radius of the ring.                     *
*    RT ......... The cone tip radius.                                *
*                                                                     *
***********************************************************************
C
C  Request double precision real variables.
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C  Dimension the arrays and specify the lengths of the character
C  strings.
C
      DIMENSION NUNIF(4),DIM(4),MDIV(4),CH(4),DM(60)
      CHARACTER*6 CHAR
      CHARACTER*1 CH
      CHARACTER*3 UNIF
C
C  Open all input and output files.
C
      OPEN(1,FILE='msh.dat')
      OPEN(3,FILE='rundat.dat')

C
C  Set the logical unit numbers for I/O data files and set the constants
C  in the program.
C
      CHAR = 'ABCDEF'
      PI = 3.141592654D0
C
C  Set the mesh uniformity flags for the cone and the cylinder.
C  The cone mesh will always be non-uniform and the cylinder mesh
C  will always be uniform to allow the elements in each of these
C  sections of the cavity to have equal areas.
C
```

```
      NUNIF(1) = 0
      NUNIF(2) = 1
      NUNIF(3) = 1
      NUNIF(4) = 1
C
C  Read the radiometer dimensions.
C
      WRITE(6,1)
      WRITE(6,2)
      READ(6,*) DIM(1)
      WRITE(6,3)
      READ(6,*) DIM(2)
      WRITE(6,4)
      READ(6,*) RI
      WRITE(6,5)
      READ(6,*) RO
      DIM(3) = RO-RI
      WRITE(6,6)
      READ(6,*) DIM(4)
      WRITE(6,8)
      READ(6,*) RT
      HO = RT*DIM(1)/(RI-RT)
C
C  Read the number of horizontal slices across each dimension.
C
      WRITE(6,9)
      WRITE(6,10)
      READ(6,*) MDIV(1)
      WRITE(6,11)
      READ(6,*) MDIV(2)
      WRITE(6,12)
      READ(6,*) MDIV(3)
      WRITE(6,14)
      READ(6,*) MDIV(4)
C
C  Read the number of elements (pie slices) around the circumference
C  of the cavity.
C
      WRITE(6,19)
      READ(6,*) N1
C
C  write the mesh geometry data file.
C
      WRITE(1,21) DIM(1),DIM(2),RI,RO,DIM(4),RT,HO
      WRITE(1,22) N1,(MDIV(I),I=1,4)
      WRITE(1,22) (NUNIF(I),I=1,4)
C
C  Since the cone mesh is always non-uniform, (i.e. delta z is not constant)
C  calculate the element dimensions along the cone height so that all the
C  elements of the cone have the same area.
C
      BETA = DATAN(RI/(DIM(1)+HO))
      ATIP = PI*RT*DSQRT(RT**2+HO**2)
      AREA = PI*RI*DSQRT(RI**2+(DIM(1)+HO)**2) - ATIP
      C = (AREA/MDIV(1))*DCOS(BETA)/(PI*DTAN(BETA))
      H = HO
      DO 50 I = 2,MDIV(1)*2,2
         HI =DSQRT(C + H**2)
         DM(I) = (HI - H)/2.0D0
         DM(I-1) = DM(I)
         H = HI
   50 CONTINUE
C
C  Write these dimensions to the mesh geometry data file.
C
```

```
      WRITE(1,21) (DM(I),I=1,2*MDIV(1))
C
C  Write the mesh geometry data to the run data record.
C
      WRITE(3,26)
      WRITE(3,27)
      WRITE(3,28)
      WRITE(3,29)
      DO 90 I=1,4
      UNIF = 'YES'
      IF(NUNIF(I).EQ.0) UNIF = 'NO '
      CH(I) = CHAR(I:I)
      WRITE(3,30) CH(I),DIM(I),MDIV(I),UNIF
   90 CONTINUE
      WRITE(3,26)
      WRITE(3,31) RI
      WRITE(3,32) RT
      WRITE(3,33) N1
C
      STOP
C
C  The format statements follow:
C
    1 FORMAT(//,10X,41('*'),/,10X,'*',7X,'RADIOMETER GEOMETRY INPUT',
     &        7X,'*',/,10X,41('*'),///,1X,'THE RADIOMETER DIMENSIONS ',
     &        'FOLLOW',/,1X,32('='),/)
    2 FORMAT(1X,'Cone height?  (mm)')
    3 FORMAT(1X,'Cylinder height?  (mm)')
    4 FORMAT(1X,'Inside radius of the ring?  (mm)')
    5 FORMAT(1X,'Outside radius of the ring?  (mm)')
    6 FORMAT(1X,'Length of the thermal impedance?  (mm)')
    8 FORMAT(1X,'Cone tip raduis? (mm)')
    9 FORMAT(//,1X,'THE RADIOMETER MESH SPECIFICATIONS FOLLOW',/,
     &          1X,41('='),//)
   10 FORMAT(1X,'Number of horizontal slices along cone height?')
   11 FORMAT(1X,'Number of horizontal slices along cylinder height?')
   12 FORMAT(1X,'Number of washers along ring radius?')
   14 FORMAT(1X,'Number of horizontal slices along thermal impedance?')
   19 FORMAT(/,1X,'Number of azimuthal divisions around cavity?',
     &       /,3X,'(this must be an even number greater than 8)')
   21 FORMAT(6D12.5)
   22 FORMAT(14I5)
   23 FORMAT(/,1X,'Enter the subdivisional dimensions for',
     &         ' measurement "',A1,'"')
   24 FORMAT(1X,I2,' values are required over length ',F5.2,/,
     &'  (separate each by a space)')
   26 FORMAT(12X,50('-'))
   27 FORMAT(11X,'│           │ MEASUREMENT │  NO. OF   │            │')
   28 FORMAT(11X,'│ DIMENSION │    (mm)     │ DIVISIONS │ UNIFORMITY │')
   29 FORMAT(11X,'│',50('='),'│')
   30 FORMAT(11X,'│',5X,A1,5X,'│',4X,F5.2,4X,'│',4X,I2,5X,'│',4X,A3,
     &5X,'│')
   31 FORMAT(/,10X,'Radius of cavity aperature ..........',
     &         F4.2,'  (mm)')
   32 FORMAT(10X,'Cone tip radius..................... ',F6.4,'  (mm)')
   33 FORMAT(10X,'Number of elements around cavity .... ',I3)
      END
```

Appendix C

Program ERBER2

```
************************************************************************
*                                                                      *
*                         ERBER2 FORTRAN                               *
*                                                                      *
*    This FORTRAN 77 program prompts the user for thermal analysis     *
*    data and then writes this data to input files for the thermal     *
*    analysis program, 'ERBETA', and 'ERBE EXEC'.  A complete record   *
*    of all run variables, including the mesh geometry data, is also   *
*    created by this program.                                          *
*                                                                      *
*    Written by Nour E. Tira, and modified by Kory J. Priestley        *
*                                                                      *
************************************************************************
*                                                                      *
*                           NOMENCLATURE                               *
*                                                                      *
*    ABSP ......... Absorptivity of the radiometer material.           *
*    ALPHA ........ Resistance-temperature constant.                   *
*    APJ(I) ....... Illuminated area at angle I.                       *
*    COND ......... Conductivity of the radiometer material.           *
*    CP ........... Specific heat of the radiometer material.          *
*    DELT ......... Temperature drop between resistors and heat sink.  *
*    DENS ......... Density of the radiometer material.                *
*    DT ........... Time increment.                                    *
*    EM ........... Emissivity of the radiometer material.             *
*    E0 ........... Bridge voltage for the feedback circuit.           *
*    E2 ........... The steady-state or initial voltage across the     *
*                   electical substitution heater wire.                *
*    FILE1 ........ The name of the disk file which contains the       *
*                   non-uniform initial conditions.                    *
*    FILE2 ........ The name of the disk file which contains the       *
*                   cavity distribution factors (DFC's).               *
*    FILE3 ........ The name of the disk file which contains the       *
*                   aperature distribution factors (DFA's).            *
*    FILE4 ........ The name of the disk file which contains the       *
*                   distribution factors for a collimated beam (DFS's)*
*    FMAG(I) ...... The magnitude of the source field for each         *
*                   time step.  Dimensioned NDT.                       *
*    FMAX  ........ The maximum magnitude of the source field          *
*    HWIRE ........ Height to which the heater wire is wound above     *
*                   the cone tip.                                      *
*    INIT ......... The indicator for transient analysis.              *
*                   INIT = 0   steady-state analysis                   *
*                   INIT = 1   transient, uniform initial conditions   *
*                   INIT = 2   transient, non-uniform i.c.             *
*    IPDF(I) ...... The array of node numbers with specified temper-   *
*                   ature boundary conditions.  Dimensioned NPDF.      *
*    LNOD(I) ...... Array of nodes located at the base of the          *
*                   thermal impedance.Dimensioned nnb.                 *
*    LNRES(I) ..... Array of nodes along the thermal impedance.        *
*                   Dimensioned NNRES.                                 *
*    NAME ......... The name of the radiometer material.               *
*    NAP .......... Imaginary aperature element used to describe       *
*                   the position of the entering irradiance vector.    *
*    NCAV ......... The cavity element used to describe the position   *
*                   of the entering irradiance vector.                 *
*    NDT .......... Number of time steps in the analysis--equal to     *
*                   one for a steady-state analysis.                   *
*    NDTS ......... Number of time steps corresponding to a complete   *
*                   span of the collimated beam.                       *
*    NF ........... Indicator for the type of source field.            *
*                   NF = 1   constant FMAG                             *
*                   NF = 2   sinusoidal FMAG                           *
*    NHEAT ........ Indicator for the electrical cavity heater.        *
```

```
*                   NHEAT = 0   heater not utilized in the analysis    *
*                   NHEAT = 1   heater utilized in the analysis        *
*    NNB .......... The number of nodes at the thermal impedance base.*
*    NNRES ........ The number of node levels along the thermal        *
*                   impedance, used to select the position of the     *
*                   resistance thermometer.                            *
*    NPDF ......... The number of specified temperature boundary       *
*                   conditions.                                        *
*    NRAD ......... Flag indicating thermal radiative analysis.        *
*                   NRAD = 0   radiation analysis not performed        *
*                   NRAD = 1   radiative analysis performed            *
*    NVEC ......... Flag indicating the presence of an incoming        *
*                   irradiance vector.                                 *
*                   NVEC = 0   no vector entering cavity               *
*                   NVEC = 1   vector entering cavity                  *
*    NRES ........  The first node on the thermal impedance which      *
*                   is covered by the resistance thermometer.          *
*    PHIM ........  Maximum span angle of the collimated beam.         *
*    REFR ......... Reflectivity ratio of the radiometer material.     *
*    RI ........... Inside radius of the cavity.                       *
*    SMAG(I) ...... The magnitude of the collimated beam for each      *
*                   time step.  Dimensioned NDT.                       *
*    SMAX  ........ The maximum magnitude of the collimated beam.      *
*    TAU .......... Time constant in the analysis of the voltage       *
*                   across the heater wire.                            *
*    TFON ......... Time during which FMAG is on.                      *
*    THETA ........ Time approximation parameter.                      *
*                   Theta = 0     forward-difference scheme            *
*                   Theta = 1/2   Crank-Nicolson scheme                *
*                   Theta = 2/3   Galerkin scheme                      *
*                   Theta = 1     backward-difference scheme           *
*    THCK ......... Thickness of the radiometer material.              *
*    THS .......... Temperature of the heat sink.                      *
*    TMAX ......... Upper bound on time for a transient analysis.      *
*    TINIT  ....... Radiometer predominant initial temperature.        *
*    TSON ......... Time during which SMAG is on.                      *
*    TVON ......... Time during which VMAG is on.                      *
*    VMAG(I) ...... The irradiance vector magnitude for each time      *
*                   step.  Dimensioned NDT.                            *
*    VMAX  ........ The maximum magnitude of the irradiance vector.    *
*    VPDF(I) ...... Nodal temperatures specified as boundary condi-    *
*                   tions.  Dimensioned NPDF.                          *
*                                                                      *
************************************************************************
C
C  Request double precision real variables.
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C  Dimension the arrays and specify the lengths of the character
C  strings.
C
      INTEGER CHANEL,CH

      DIMENSION IPDF(60),VPDF(60),LNOD(60)
      DIMENSION LINE(20),LRES(30),LNRES(20)
      DIMENSION FMAG(250)
      CHARACTER*1 CHAR
      CHARACTER*20 NAME,FILE1,FILE2,FILE3,FILE4
C
C  Open all necessary input and output files.
C
      OPEN(2,FILE='rundat.dat',status='old')
      OPEN(4,FILE='mesh.dat',status='old')
      OPEN(5,FILE='nodes.dat',status='old')
```

```
      OPEN(3,FILE='source.dat')
      OPEN(8,FILE='flags.dat')
      OPEN(9,FILE='props.dat')
      OPEN(10,FILE='time.dat')
      OPEN(11,FILE='bndtmp.dat')
      OPEN(12,FILE='emiss.dat')
      OPEN(13,FILE='heater.dat')
      OPEN(15,FILE='temp.dat')
      OPEN(21,FILE='ttchanel.input')
      OPEN(22,FILE='swchanel.input')
      OPEN(42,FILE='run.dat')
      OPEN(43,FILE='init.dat')
C
C  Set the logical unit numbers for I/O data files, and initialize the
C  program flags.
C
      DATA LUN3,LUN4,LUN5,LUN7,LUN8,LUN22,LUN6/3,4,5,7,8,22,99/
      DATA NDT,NDTF,NDTV,NDTS/1,1,1,1/
      DATA CP,DENS,EM,REFR,ABSP/0.0D0,0.0D0,0.0D0,0.0D0,0.0D0/
      PI = 3.141592654D0
      READ(4,*) NEM,NNM
      READ(4,*) DM1,DM2,DM3,DM4,RI
C
C  Input whether the total or visible channel is being considered.
C
      WRITE(6,*)'Which channel is being modeled?'
      WRITE(6,*)'Enter 1 for the total channel, or'
      WRITE(6,*)'Enter 2 for the visible channel.'
      READ(6,*)CHANEL
      CH=CHANEL+20
C
C  Input if the analysis is steady-state or transient.
C
   10 WRITE(6,301)
      READ(6,201) CHAR
      IF(CHAR.NE.'S' .AND. CHAR.NE.'T') GO TO 10
      IF(CHAR.EQ.'S') INIT = 0
C
C  If the analysis is transient, input the type of initial conditions
C  present in the model.
C
      IF(CHAR.EQ.'T') THEN
   20     WRITE(6,302)
          READ(6,201) CHAR
          IF(CHAR.NE.'U' .AND. CHAR.NE.'N') GO TO 20
          IF(CHAR.EQ.'U') INIT = 1
          IF(CHAR.EQ.'N') INIT = 2
      ENDIF
      WRITE(43,*)INIT

C
C  Input the material properties.
C
      WRITE(6,305)
      READ(6,202) NAME
      WRITE(6,306)
      READ(6,*) COND
      IF(INIT.GT.0) THEN
          WRITE(6,307)
          READ(6,*) CP
          WRITE(6,308)
          READ(6,*) DENS
          DENS = DENS/(1000.0D0**2)
      ENDIF
      DO 11 I=1,11
```

```
          READ(CH,*)
   11  CONTINUE
       READ(CH,*)EMFOVL
       READ(CH,*)EMUSUB
       EMLSUB=EMUSUB
       READ(CH,*)EMBAR
       READ(CH,*)EMCONE
       READ(CH,*)EMCAVSUB
       READ(CH,*)EMRING

       WRITE(6,312)
       READ(6,*) THCK
C
C  If the analysis is transient, input transient analysis parameters.
C
       IF(INIT.GT.0) THEN
           WRITE(6,313)
           READ(6,*) TMAX
           WRITE(6,314)
           READ(6,*) DT
           WRITE(6,315)
           READ(6,*) THETA
           WRITE(6,316)
           NDT = NINT(TMAX/DT)
       ENDIF
C
C  For steady state,

C  If the initial conditions are uniform for each zone(i.e. fovl,dome.etc.)
C  or semi-uniform, input the predominant initial temperature.
C
       IF(INIT.EQ.0) THEN
           WRITE(6,507)
           READ(6,*) TFOVL
           WRITE(6,508)
           READ(6,*) TUSUB
           WRITE(6,509)
           READ(6,*) TLSUB
           WRITE(6,510)
           READ(6,*) TDOME
           WRITE(6,512)
           READ(6,*) TCAVSUB
           WRITE(6,513)
           READ(6,*) TRING
       ENDIF
C
C  If the initial conditions are non-uniform, input the disk file where
C  the initial conditions are located.
C
       IF(INIT.EQ.2) THEN
           WRITE(6,323)
           READ(6,202) FILE1
       ENDIF
C
C  Read the node numbers located at the base of the T.I , and input the
C  temperature of the heat sink.  The heat sink temperature will be
C  applied to these nodes as a boundary condition after other
C  possible boundary conditions are input.
C
       READ(5,*) NNB
       DO 50 I=1,NNB
   50  READ(5,*) LNOD(I)
       WRITE(6,324)
       READ(6,*) THS
C
```

```
C  Input the number of applied temperature boundary conditions not
C  including the boundary conditions located at the thermal impedance
C  base. The nodes with applied temperature boundary conditions
C  and their temperatures are then input.
C
       WRITE(6,325)
       READ(6,*) NPDF
       IF(NPDF.NE.0) THEN
           WRITE(6,326)
           WRITE(6,320)
           READ(6,*)  (IPDF(I),I=1,NPDF)
           WRITE(6,321)
           WRITE(6,322)
           READ(6,*)  (VPDF(I),I=1,NPDF)
       ENDIF
C
C
C  Since thermal radiation is considered in the analysis, input the data

C  needed for the thermal analysis.
C
C  Initialize the active radiative input arrays
      DO 60 I = 1,NDT+1
         FMAG(I) = 0.0D0
   60 CONTINUE
C
C  Input the maximum magnitude of the source field and
C  specify whether it is constant or sinusoidal.
C
c     WRITE(6,330)
c     READ(6,*) FMAX
c     IF (INIT.EQ.0) THEN
c       FMAG(1) = FMAX

c     ELSE
c       WRITE(6,345)
c       READ(6,*) NF
C
C   Input time during which FMAG is on, and calculate the corresponding
C   number of time steps.
C
        WRITE(6,346)
        READ(6,*) TFON
        NDTF = NINT(TFON/DT)

C
C  Calculate FMAG for each time step.
C
c       T = 0.0D0

c       DO 65 I = 1,NDTF

c         IF (NF.EQ.1) THEN
c            FMAG(I) = FMAX
c         ELSE
c            T = T + DT
c            FMAG(I) = FMAX*DSIN(PI*T/TFON)
c            IF (FMAG(I).LE.0.1E-08) FMAG(I) = 0.D0
c         ENDIF
c  65   CONTINUE
c     END IF
C
C  Since the electrical substitution heater is being utilized in the

C  analysis, read the heater voltage, resistance and wire height.
```

```
C
      WRITE(6,336)
      READ(6,*) E2
      WRITE(6,337)
      READ(6,*) RH
      WRITE(6,338)
      READ(6,*) HWIRE
C
C  If the analysis is transient, read the bridge voltage, circuit time
C  constant, resistance-temperature constant and the temperature drop.
C
      IF(INIT.GT.0) THEN
          WRITE(6,339)
          READ(6,*) ALPHA
          WRITE(6,340)
          READ(6,*) E0
          WRITE(6,341)
          READ(6,*) TAU
          WRITE(6,342)
          READ(6,*) DELT
      ENDIF
C
C  Read the node numbers along the edge of leg 1 which are used to selec
C  the position of the resistance thermometers on the legs, input the
C  node on leg 1 describing the position of the resistance thermometers,
C  then determine the nodes covered by the thermometers.
C
      READ(5,204) NNRES
      DO 70 I = 1,NNRES
   70 READ(5,204) LNRES(I)
      WRITE(6,343) (LNRES(I),I=1,NNRES)
      READ(6,*) NRES
      DO 80 I = 1,NNB
      LRES(I) = NRES + I - 1
   80 CONTINUE
C
C  Write the thermal analysis data necessary for the thermal analysis
C  program 'ERBETA'.  First, assign the temperature of the heat sink
C  as an applied temperature boundary condition to the nodes at the
C  ends of the legs.
C
      K1 = 1 + NPDF
      K2 = NNB  + NPDF
      K = 0
      DO 90 I = K1,K2
      K = K+1
      IPDF(I) = LNOD(K)
      VPDF(I) = THS
   90 CONTINUE
      NPDF = NPDF + NNB
C
C  Write the program flags,
C
      WRITE(8,*)CHANEL
      WRITE(8,204) INIT,NPDF,NRES
C
C  the material properties,
C
      WRITE(9,206) COND,THCK,CP,DENS
C
C  the applied temperature boundary condition data,
C
      IF(NPDF.NE.0) THEN
          WRITE(11,204) (IPDF(I),I=1,NPDF)
          WRITE(11,206) (VPDF(I),I=1,NPDF)
```

```
      ENDIF
C
C
C  For a transient analysis, write the time increment, the time limit,
C  time approximation parameter, and the initial conditions if they are
C  uniform or semi-uniform.
C
      IF (INIT.NE.0) WRITE(10,206) DT,TMAX,THETA,TSON
      IF (INIT.NE.0) WRITE(10,204) NDT,NDTF,NDTV,NDTS
      IF(INIT.EQ.1) WRITE(10,206) TINIT
C
C  Write the source field magnitude for each timestep.
C
c     WRITE(3,206) (FMAG(I),I=1,NDT)
C
C  Write the parameters needed for the analysis of the heater circuit.

C
      WRITE(13,206) E2,RH,HWIRE
C     WRITE(13,204) NRES
      IF (INIT.GT.0) WRITE(13,206) E0,TAU,ALPHA,DELT
C
C  Write the surface radiative properties to a file to be read by
C  the thermal analysis code.
C
      WRITE(12,*)EMFOVL
      WRITE(12,*)EMUSUB
      WRITE(12,*)EMLSUB
      WRITE(12,*)EMDOME
      WRITE(12,*)EMBAR
      WRITE(12,*)EMCONE
      WRITE(12,*)EMCAVSUB
      WRITE(12,*)EMRING
C
C  Write the zonal surface temperatures for uniform and constant
C  conditions.
C
      WRITE(15,*)TFOVL
      WRITE(15,*)TUSUB
      WRITE(15,*)TLSUB
      WRITE(15,*)TDOME
      WRITE(15,*)TCAV
      WRITE(15,*)TCAVSUB
      WRITE(15,*)TRING
C
C  Read the mesh geometry data chart provided by 'ERBER1', and write
C  the chart into the run data record.
C
      DO 120 I=1,12
      READ(2,205) LINE
  120 WRITE(42,205) LINE
C
C  Read the number of elements and nodes in the mesh and write them to
C  the data record along with the name of the radiometer material.
C
      WRITE(42,401) NEM,NNM
      WRITE(42,402) NAME
C
C  Write all the data needed for the thermal analysis to the run data
C  record in the same manner as previously described.  First, write
C  the material properties.
C
      WRITE(42,403) COND
      IF(INIT.GT.0) THEN
          WRITE(42,404) CP
```

```
          WRITE(42,405) DENS
      ENDIF
      WRITE(42,406) EM
      WRITE(42,408) REFR
      WRITE(42,409) THCK
C
C  Write the type of analysis.
C
      IF (INIT.EQ.0) THEN
         WRITE(42,410)
C
C  Write the data necessary for a transient analysis including the
C  initial conditions.
C
      ELSE
         WRITE(42,411) TMAX,DT,THETA
         WRITE(42,412)
         IF(INIT.EQ.1) WRITE(42,413) TINIT
         IF(INIT.EQ.2) WRITE(42,416) FILE1
      ENDIF
C
C  Write the boundary conditions.
C
      IF (NPDF.NE.0) THEN
          WRITE(42,417)
          DO 140 I=1,NPDF
  140     WRITE(42,415) IPDF(I),VPDF(I)
      ENDIF
C
C  Write the position and the magnitudes of the sourde irradiance
C
         WRITE(42,420)
         WRITE(42,422)
         DO 160 I = 1,NDT
  160    WRITE(42,423) I,FMAG(I)
C
C  Write the data for the electrical substitution heater.
C
      WRITE(42,424) LRES(1),LRES(NNB)
      WRITE(42,426)
      WRITE(42,427) E2,RH
      IF(INIT.GT.0) WRITE(42,428) E0,TAU,ALPHA,DELT
      WRITE(42,429) HWIRE
      STOP
C
C  The format statements for data input follow:
C
C
  201 FORMAT(A1)
  202 FORMAT(20A)
  203 FORMAT(3F5.2,5I5,2F10.5)
  204 FORMAT(16I5)
  205 FORMAT(20A4)
  206 FORMAT(6D12.5)
  207 FORMAT(I5,D14.5)
  208 FORMAT(30A)
C
C
C  The format statements to prompt the user for input follow:
C
C
  301 FORMAT(//,10X,36('*'),/,10X,'*',6X,'THERMAL ANALYSIS INPUT',
     &       6X,'*',/,10X,36('*'),//,1X,'Transient or steady-state ',
     &       'analysis?  (T/S)')
  302 FORMAT(1X,'Zonally uniform or non-unif. initial conditions?(U/N)')
```

```
305 FORMAT(1X,'Radiometer material?  (20 character limit)')
306 FORMAT(//,1X,'Material properties follow:',//,5X,
   &        'Conductivity?  (W/m-K)')
307 FORMAT(5X,'Specific heat?  (J/kg-K)')
308 FORMAT(5X,'Density?  (kg/cu meter)')
312 FORMAT(5X,'Cavity thickness?  (mm)')
313 FORMAT(//,1X,'Transient analysis parameters follow:',//,
   &           5X,'The maximum time limit?  (sec)')
314 FORMAT(5X,'Time increment?  s')
315 FORMAT(5X,'Time approximation parameter (THETA)?',/,
   &        10X,'THETA = 0.00   Forward-difference scheme',/,
   &        10X,'THETA = 0.50   Crank-Nicolson scheme',/,
   &        10X,'THETA = 0.66   Galerkin scheme',/,
   &        10X,'THETA = 1.00   Backward-difference scheme')
316 FORMAT(//,1X,'Initial conditions follow:',/)
507 FORMAT(5X,'Predominant initial FOVL temperature?  (K)')
508 FORMAT(5X,'Predominant initial usub temperature?  (K)')
509 FORMAT(5X,'Predominant initial lsub temperature?  (K)')
510 FORMAT(5X,'Predominant initial dome temperature?  (K)')
512 FORMAT(5X,'Predominant initial cavsub temperature?  (K)')
513 FORMAT(5X,'Predominant initial ring temperature?  (K)')

320 FORMAT(10X,'(separate each with a space)')
321 FORMAT(5X,'Temperatures of these nodes?  (K)')
322 FORMAT(10X,'(give in same order and separate each with a space)')
323 FORMAT(5X,'File where initial conditions are located?',/,
   &        6X,'<fn>  <ft>  <fm>')
324 FORMAT(1X,'Boundary conditions follow:',//,5X,'Temperature of ',
   &        'the heat sink?  (K)')
325 FORMAT(5X,'How many nodes with an applied temperature boundary',
   &        ' condition (32 Max)?',/,10X,'(Do not include nodes at',
   &        ' thermal impedance base)')
326 FORMAT(5X,'Node numbers with applied temperature?')
330 FORMAT(//,1X,'Radiative environment input follows:',//,
   &           5X,'Maximum magnitude of the source field ? (mW)')
351 FORMAT(5X,'Maximum magnitude of the irradiance vector ? (mW)')
345 FORMAT(5X,'Inter 1 if FMAG is constant, and 2 if it is sinusoidal.
    &')
346 FORMAT(5X,'Time during which FMAG is on? (sec)')
353 FORMAT(5X,'Time during which VMAG is on? (sec)')
555 FORMAT(5X,'Angle of incidence w/r to aperature normal? (Deg.)')
331 FORMAT(5X,'File where the DFC values are located?',/,5X,
   &           '<fn>  <ft>  <fm>')
332 FORMAT(5X,'Is there an irradiance vector entering the ',
   &           'cavity?  (Y/N)')
333 FORMAT(5X,'Aperature and cavity elements describing its',
   &           ' position?')
334 FORMAT(5X,'Magnitude of the irradiance vector for each time',
   &           ' step?  mW',/,5X,I3,' values required.')
335 FORMAT(5X,'File where the DFAC values from the FOVL aperture
   &            are located?',/,5X,
   &           '<fn>  <ft>  <fm>')
336 FORMAT(//,1X,'Electrical substitution heater data input follows:',
   &        //,5X,'Initial/Steady-state heater voltage?  (V)')
337 FORMAT(5X,'The resistance of the heater wire?  (ohms)')
338 FORMAT(5X,'The height of heater wire on the cavity?  (mm)')
339 FORMAT(5X,'The temperature-resistance constant?  (1/K)')
340 FORMAT(5X,'The bridge voltage?  (V)')
341 FORMAT(5X,'The circuit time constant?  (sec)')
342 FORMAT(5X,'The desired temperature drop between the resistor',
   &           ' and the heat sink?  (K)')
343 FORMAT(5X,'The lowest node number on the thermal impedance',
   &           ' covered by the resistor?',//,5X,
   &           ' the following nodes are valid choices:',//,5X,12I5,/)
```

```
C
C  The format statements for the run data record follow:
C
  401 FORMAT(10X,'Number of elements in mesh ........... ',I3,/,
     &       10X,'Number of nodes in mesh .............. ',I3,//)
  402 FORMAT(10X,'Radiometer material:  ',20A)
  403 FORMAT(15X,'Conductivity .......... ',F10.5,'  mW/mm-K')
  404 FORMAT(15X,'Specific heat ......... ',F10.5,'  mW-s/g-K')
  405 FORMAT(15X,'Density ............... ',F10.5,'  g/mm**3')
  406 FORMAT(15X,'Emissivity ............ ',F10.5)
  407 FORMAT(15X,'Absorptivity .......... ',F10.5)
  408 FORMAT(15X,'Reflectivity ratio .... ',F10.5)
  409 FORMAT(15X,'Thickness ............. ',F10.5,'  mm')
  410 FORMAT(//,10X,'Analysis:  Steady-state')
  411 FORMAT(//,10X,'Analysis:  Transient',/,
     &          15X,'Time limit ............ ',F10.5,'  s',/,
     &          15X,'Time increment ........ ',F10.5,'  s',/,
     &          15X,'Time parameter ........ ',F7.2)
  412 FORMAT(//,10X,'Initial conditions:',/)
  413 FORMAT(15X,'Predominant initial temperature ... ',F6.2,' K',/)
  415 FORMAT(18X,I3,13X,F6.2)
  416 FORMAT(15X,'Non-uniform initial temperatures from file:  ',20A)
  417 FORMAT(//,10X,'Boundary conditions:',//,
     &       18X,'Node',10X,'Temperature  K')
  419 FORMAT(//,10X,'Thermal radiation neglected')
  420 FORMAT(//,10X,'Radiative environment:',/)
  421 FORMAT(15X,'Irradiance vector position:  ',I2,' to ',I2,/)
  430 FORMAT(15X,'Collimated beam angle: ',F6.2,' Deg.',/)
  418 FORMAT(15X,'Collimated beam angle span: from ',F6.2,' to ',F6.2,
     &' Deg.',/)
  422 FORMAT(15X,'Time',6X,'Field',9X,'Vector',9X,/,
     &       15X,'step',4X,'magnitude',6X,'magnitude',/,26X,' mW',11X,
     &       ' mW ',/)
  722 FORMAT(15X,'Magnitude of the collimated beam: ',F10.5,' mW/mm**2'
     &,/)
  423 FORMAT(14X,I3,5X,E10.5,5X,E10.5,5X,E10.5)
  424 FORMAT(/,15X,'Thermometer location',/,
     &         15X,'Thermal Impedance Nodes.......... ',I3,' TO ',I3)
  425 FORMAT(//,10X,'Electrical substitution heater not utilized')
  426 FORMAT(//,10X,'Electrical substitution heater:')
  427 FORMAT(/,15X,'Heater voltage .......... ',F7.2,'    volts  ',/,
     &         15X,'Heater resistance ....... ',F7.2,'    ohms   ')
  428 FORMAT(/,15X,'Bridge voltage .......... ',F7.2,'    volts  ',/,
     &         15X,'Time constant ........... ',F7.2,'      s    ',/,
     &         15X,'Resistance-temperature',/,
     &         15X,'     constant ........... ',F7.5,'     1/K   ',/,
     &         15X,'Temperature drop ........ ',F7.5,'     K   ')
  429 FORMAT(/,15X,'Heater wire height ...... ',F7.2,'     mm   ')
  453 FORMAT(5X,'Time during which SMAG is on? (sec)')
  454 FORMAT(5X,'Maximum angle w/r to aperature normal? (Deg.)')
  455 FORMAT(5X,'Solar port Radius ? (mm)')
  456 FORMAT(5X,'Distance between Solar port and aperture planes? (mm)')
      END
```

Appendix D

Program ERBER3

```
***********************************************************************
*                                                                     *
*                            ERBER3 FORTRAN                           *
*                                                                     *
*   This FORTRAN 77 program computes the monochromatic distribution   *
*   factor matrices for the thermal analysis program, 'ERBETA'.       *
*   The Monochromatic distribution factor matrix calculated by        *
*   'ERBERT' is the necessary input file.                             *
*                                                                     *
*   Written by Kory J. Priestley                                      *
*                                                                     *
***********************************************************************
*                                                                     *
C    PROGRAM ERBER3
C
C    This program modifies the output of erbert.f into the
C    form needed by erbeta.f. (DFAC's).  It also creates the
C    file numb.dat
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER CH,CHANEL

      DIMENSION DFBAR(200,100,15),DFCONE(200,100,15),DFRING(200,100,15)
      DIMENSION DFSCN(1,100,15),DFOPT(200,100,15)
      DIMENSION DFDOME(200,100,15),DFCSUB(200,100,15)
      DIMENSION DFUSUB(200,100,15),DFLSUB(200,100,15)

      OPEN(7,FILE='dfacmtx.dat',status='old')
      OPEN(11,FILE='ttchanel.input',status='old')
      OPEN(12,FILE='swchanel.input',status='old')
      OPEN(16,FILE='numb.dat')
      OPEN(17,FILE='dfac.dat')
      OPEN(20,FILE='dfopt.dat')
      OPEN(21,FILE='dfscn.dat')
      OPEN(22,FILE='dfcav.dat')
      OPEN(23,FILE='dfdome.dat')
      OPEN(24,FILE='dfring.dat')
      OPEN(65,FILE='dfcsub.dat')
      OPEN(66,FILE='dfusub.dat')
      OPEN(67,FILE='dflsub.dat')
      OPEN(68,FILE='dfbar.dat')
      OPEN(69,FILE='dfcone.dat')

1     WRITE(6,*)'Which channel do you wish to model?'
      WRITE(6,*)'For total channel enter 1.'
      write(6,*)'For visible channel enter 2.'
      READ(6,*) CHANEL
      IF (CHANEL.NE.1.AND.CHANEL.NE.2) GO TO 1
      CH=CHANEL+10
      DO 10 I=1,33
        READ(CH,*)
   10 CONTINUE
      READ(CH,*)NBLAT
      READ(CH,*)NBLONG
      READ(CH,*)NBLSU
      READ(CH,*)NBWLSU
      READ(CH,*)NBWASH
      READ(CH,*)NBPS
      READ(CH,*)NBSH
      READ(CH,*)NBCO
      READ(CH,*)NBPSD
      READ(CH,*)NBPSB
      READ(CH,*)NBPSC
      READ(CH,*)NBPSR
      READ(CH,*)NBHSB
```

```
      READ(CH,*)NBHSC
      READ(CH,*)NBPSCS
      READ(CH,*)NBHSR
      READ(CH,*)
      READ(CH,*)NBWLIN
c     write(6,*)'nbwlin =',nbwlin
C
C  CREATE DATA FOR FILE NUMB.DAT
C
      NBFOVL=NBPS*NBLAT
      NBUSUB=NBPS*NBWASH
      NBLSUB=NBPS*NBWLSU
      NBCAVSUB=NBPS
      NBRING=NBPS*NBHSR
      NBDOME=NBPS*NBSH*NBCO
      NBSCENE=1
      NBBAR=NBPS*NBHSB
      NBCONE=NBPS*NBHSC
      NBCAV=NBBAR+NBCONE
      NBDUM=NBFOVL+NBUSUB+NBLSUB+NBDOME+NBSCENE
      NBDUM2=NBRING+NBCAVSUB
      write(6,*)'nbdum =',nbdum
      write(6,*)'nbdum2 =',nbdum2
      write(6,*)'nbcav =',nbcav
      write(6,*)'nbfovl =',nbfovl
      write(6,*)'nbusub =',nbusub
      write(6,*)'nblsub =',nblsub
      write(6,*)'nbdome =',nbdome
      write(6,*)'nbscene =',nbscene
C
C  INITIALIZE ALL DFACS TO ZERO
C
      CALL ZERO(DFSCN)
      CALL ZERO(DFDOME)
      CALL ZERO(DFOPT)
      CALL ZERO(DFRING)
C
C  SET UP DFSCN
C
      nn=0
      DO 20 I=1,NBSCENE
        READ(7,*)
        nn=nn+1
        DO 22 K=1,NBWLIN
          DO 30 L=1,NBDUM
            READ(7,*)
            nn=nn+1
   30     CONTINUE
          DO 24 J=1,NBCAV
            READ(7,*)M,N,DFSCN(I,J,K)
            nn=nn+1
   24     CONTINUE
          DO 32 L=1,NBDUM2
            READ(7,*)
            nn=nn+1
   32     CONTINUE
   22   CONTINUE
   20 CONTINUE
      write(6,*)'nn',nn
C
C  SET UP DFOPT
C
      DO 40 I=1,NBFOVL,NBPS
        READ(7,*)
        nn=nn+1
```

```
          DO 42 K=1,NBWLIN
            DO 50 L=1,NBDUM
              nn=nn+1
              READ(7,*)
   50       CONTINUE
            DO 44 J=1,NBCAV
              nn=nn+1
              READ(7,*)M,N,DFOPT(I,J,K)
   44       CONTINUE
            DO 52 L=1,NBDUM2
              nn=nn+1
              READ(7,*)
   52       CONTINUE
   42     CONTINUE
        write(6,*)i
   40 CONTINUE
        CALL DFCALC(NBWLIN,(NBFOVL/NBPS),NBPS,DFOPT)
        write(6,*)'nn',nn
C
C   SET UP DFLSUB
C
        DO 60 I=1,NBLSUB,NBPS
          READ(7,*)
          nn=nn+1
          DO 62 K=1,NBWLIN
            DO 70 L=1,NBDUM
              nn=nn+1
              READ(7,*)
   70       CONTINUE
            DO 64 J=1,NBCAV
              nn=nn+1
              READ(7,*)M,N,DFLSUB(I,J,K)
   64       CONTINUE
            DO 72 L=1,NBDUM2
              nn=nn+1
              READ(7,*)
   72       CONTINUE
   62     CONTINUE
   60 CONTINUE
        CALL DFCALC(NBWLIN,(NBLSUB/NBPS),NBPS,DFLSUB)
        write(6,*)'nn',nn
C
C   SET UP DFUSUB
C
        DO 80 I=1,NBUSUB,NBPS
          READ(7,*)
          nn=nn+1
          DO 82 K=1,NBWLIN
            DO 90 L=1,NBDUM
              nn=nn+1
              READ(7,*)
   90       CONTINUE
            DO 84 J=1,NBCAV
              nn=nn+1
              READ(7,*)M,N,DFUSUB(I,J,K)
   84       CONTINUE
            DO 92 L=1,NBDUM2
              nn=nn+1
              READ(7,*)
   92       CONTINUE
   82     CONTINUE
   80 CONTINUE
        CALL DFCALC(NBWLIN,(NBUSUB/NBPS),NBPS,DFUSUB)
        write(6,*)'nn',nn
```

```
C
C  SET UP DFDOME
C
      IF (CHANEL.EQ.2) THEN
        DO 100 I=1,NBDOME,NBPS
          nn=nn+1
          READ(7,*)
          DO 102 K=1,NBWLIN
            DO 110 L=1,NBDUM
              nn=nn+1
              READ(7,*)
  110       CONTINUE
            DO 104 J=1,NBCAV
              nn=nn+1
              READ(7,*)M,N,DFDOME(I,J,K)
  104       CONTINUE
            DO 112 L=1,NBDUM2
              nn=nn+1
              READ(7,*)
  112       CONTINUE
  102     CONTINUE
  100   CONTINUE
      ELSE
        DO 114 I=1,NBDOME,NBPS
        DO 114 K=1,NBWLIN
        DO 114 J=1,NBCAV
          DFDOME(I,J,K)=0.0D0
  114   CONTINUE
      END IF
      CALL DFCALC(NBWLIN,(NBDOME/NBPS),NBPS,DFDOME)
      write(6,*)'nn',nn
C
C  SET UP DFBAR
C
      DO 120 I=1,NBBAR,NBPS
        nn=nn+1
        READ(7,*)
        DO 122 K=1,NBWLIN
          DO 130 L=1,NBDUM
            nn=nn+1
            READ(7,*)
  130     CONTINUE
          DO 124 J=1,NBCAV
            nn=nn+1
            READ(7,*)M,N,DFBAR(I,J,K)
  124     CONTINUE
          DO 132 L=1,NBDUM2
            nn=nn+1
            READ(7,*)
  132     CONTINUE
  122   CONTINUE
  120 CONTINUE
      CALL DFCALC(NBWLIN,(NBBAR/NBPS),NBPS,DFBAR)
      write(6,*)'nn',nn
C
C  SET UP DFCONE
C
      DO 125 I=1,NBCONE,NBPS
        write(6,*)'i=',I
        nn=nn+1
        READ(7,*)
        DO 126 K=1,NBWLIN
          DO 128 L=1,NBDUM
            nn=nn+1
            READ(7,*)
```

```
 128      CONTINUE
          DO 127 J=1,NBCAV
            nn=nn+1
            READ(7,*)M,N,DFCONE(I,J,K)
 127      CONTINUE
          DO 129 L=1,NBDUM2
            nn=nn+1
            READ(7,*)
 129      CONTINUE
 126    CONTINUE
 125  CONTINUE
      NNN=6
      CALL DFCALC(NBWLIN,NNN,NBPS,DFCONE)
C
C   SET UP DFCSUB
C
      DO 140 I=1,NBCAVSUB,NBPS
        nn=nn+1
        READ(7,*)
        DO 142 K=1,NBWLIN
          DO 150 L=1,NBDUM
            nn=nn+1
            READ(7,*)
 150      CONTINUE
          DO 144 J=1,NBCAV
            nn=nn+1
            READ(7,*)M,N,DFCSUB(I,J,K)
 144      CONTINUE
          DO 152 L=1,NBDUM2
            nn=nn+1
            READ(7,*)
 152      CONTINUE
 142    CONTINUE
 140  CONTINUE
      CALL DFCALC(NBWLIN,(NBCAVSUB/NBPS),NBPS,DFCSUB)
      write(6,*)'nn',nn
C
C   SET UP DFRING
C
      IF (CHANEL.EQ.1) THEN
        DO 160 I=1,NBRING,NBPS
          nn=nn+1
          READ(7,*)
          DO 162 K=1,NBWLIN
            DO 170 L=1,NBDUM
              nn=nn+1
              READ(7,*)
 170        CONTINUE
            DO 164 J=1,NBCAV
              nn=nn+1
              READ(7,*)M,N,DFRING(I,J,K)
 164        CONTINUE
            DO 172 L=1,NBDUM2
              nn=nn+1
              READ(7,*)
 172        CONTINUE
 162      CONTINUE
 160    CONTINUE
      ELSE
        DO 174 I=1,NBRING,NBPS
        DO 174 K=1,NBWLIN
        DO 174 J=1,NBCAV
          DFRING(I,J,K)=0.0D0
 174    CONTINUE
      END IF
```

```
      CALL DFCALC(NBWLIN,(NBRING/NBPS),NBPS,DFRING)
      write(6,*)'nn',nn
C
C  WRITE OUTPUT TO PROPER FILES FOR ERBETA.F
C

C
C  FIRST CREATE NUMB.DAT
C
      WRITE(16,*)NBFOVL
      WRITE(16,*)NBUSUB
      WRITE(16,*)NBLSUB
      WRITE(16,*)NBCAVSUB
      WRITE(16,*)NBRING
      WRITE(16,*)NBDOME
      WRITE(16,*)NBSCENE
      WRITE(16,*)NBBAR
      WRITE(16,*)NBCONE
C
C  CREATE DFSCN.DAT
C
      DO 180 I=1,NBSCENE
      DO 180 K=1,NBWLIN
      DO 180 J=1,NBCAV
         WRITE(21,*)DFSCN(I,J,K)
  180 CONTINUE
C
C  CREATE DFOPT.DAT
C
      DO 190 I=1,NBFOVL
      DO 190 K=1,NBWLIN
      DO 190 J=1,NBCAV
         WRITE(20,*)DFOPT(I,J,K)
  190 CONTINUE
C
C  CREATE DFDOME.DAT
C
      DO 200 I=1,NBDOME
      DO 200 K=1,NBWLIN
      DO 200 J=1,NBCAV
         WRITE(23,*)DFDOME(I,J,K)
  200 CONTINUE
C
C  CREATE DFRING.DAT
C
      DO 210 I=1,NBRING
      DO 210 K=1,NBWLIN
      DO 210 J=1,NBCAV
         WRITE(24,*)DFRING(I,J,K)
  210 CONTINUE
C
C  CREATE DFBAR.DAT
C
      DO 220 I=1,NBBAR
      DO 220 K=1,NBWLIN
      DO 220 J=1,NBCAV
         WRITE(68,*)DFBAR(I,J,K)
  220 CONTINUE
C
C  CREATE DFCONE.DAT
C
      DO 225 I=1,NBCONE
      DO 225 K=1,NBWLIN
      DO 225 J=1,NBCAV
         WRITE(69,*)DFCONE(I,J,K)
```

```
  225 CONTINUE

C
C  CREATE DFCSUB.DAT
C
      DO 230 I=1,NBCAVSUB
      DO 230 K=1,NBWLIN
      DO 230 J=1,NBCAV
        WRITE(65,*)DFCSUB(I,J,K)
  230 CONTINUE
C
C  CREATE DFUSUB.DAT
C
      DO 240 I=1,NBUSUB
      DO 240 K=1,NBWLIN
      DO 240 J=1,NBCAV
        WRITE(66,*)DFUSUB(I,J,K)
  240 CONTINUE
C
C  CREATE DFLSUB.DAT
C
      DO 250 I=1,NBLSUB
      DO 250 K=1,NBWLIN
      DO 250 J=1,NBCAV
        WRITE(67,*)DFLSUB(I,J,K)
  250 CONTINUE
      END
**********************************************
*                                            *
*                                            *
*                                            *
**********************************************
      SUBROUTINE ZERO(DF)

      REAL*8 DF
      INTEGER I,J,K
      DIMENSION DF(200,100,15)

      DO 10 K=1,15
      DO 10 I=1,200
      DO 10 J=1,100
        DF(I,J,K)=0.0D0
   10 CONTINUE
      END
**********************************************
*                                            *
*                                            *
**********************************************
      SUBROUTINE DFCALC(NBWLIN,NBHS,NBPS,DFAC)

      REAL*8 DFAC
      INTEGER NBWLIN,NBHS,NBPS
      INTEGER I,J,K,I1,I2,I3,LL
      DIMENSION DFAC(200,100,15)
      write(17,*)nbhs
C
C  Create three dimensional DFAC matrix
C
      DO 250 K=1,NBWLIN
        DO 120 I1=1,NBHS
          write(17,*)i1
          DO 130 J=1,NBPS
          DO 130 I2=1,10
            DFAC((I1-1)*NBPS+J,(I2-1)*NBPS+J,K) =
     &          DFAC((I1-1)*NBPS+1,(I2-1)*NBPS+1,K)
```

```
  130       CONTINUE
            DO 140 LL=1,NBPS-1
              DO 150 J=1,NBPS-LL
              DO 150 I2=1,10
                DFAC((I1-1)*NBPS+J,(I2-1)*NBPS+J+LL,K) =
     &              DFAC((I1-1)*NBPS+1,(I2-1)*NBPS+1+LL,K)
  150         CONTINUE
              I3=0
              DO 160 J=NBPS-LL+1,NBPS
                I3=I3+1
              DO 160 I2=1,10
                DFAC((I1-1)*NBPS+J,(I2-1)*NBPS+I3,K) =
     &              DFAC((I1-1)*NBPS+1,(I2-1)*NBPS+1+LL,K)
  160         CONTINUE
  140       CONTINUE
  120     CONTINUE
  250   CONTINUE
c       DO 1 I=1,NBHS*NBPS
c       DO 1 K=1,NBWLIN
c       DO 1 J=1,100
c         WRITE(17,*)'I, J, K, DFAC(I,J,K)',I,J,K,DFAC(I,J,K)
c     1 CONTINUE
        END
```

Appendix E

Program ERBER4

```
*************************************************************************
*                                                                       *
*                            ERBER4 FORTRAN                             *
*                                                                       *
*                                                                       *
*    This FORTRAN 77 program sets up the temperature distribution       *
*    files needed by 'ERBETA'. The user is prompted for all necessary   *
*    information.                                                       *
*                                                                       *
*    Written by Kory J. Priestley                                       *
*                                                                       *
*************************************************************************
      IMPLICIT DOUBLE PRECISION(A-L,O-Z)

      OPEN(29,file='tfovl.dat')
      OPEN(30,FILE='tusub.dat')
      OPEN(31,FILE='tlsub.dat')
      OPEN(32,FILE='tdome.dat')
      OPEN(33,FILE='tring.dat')
      OPEN(34,FILE='tcavsub.dat')
      OPEN(35,FILE='tcav.dat')
      OPEN(36,FILE='tscn.dat')
      OPEN(3,FILE='source.dat')

      WRITE(6,*)'Number of rows needed in files?'
      READ(6,*)NROW
      NROW=60
      WRITE(6,*)'Flux from shortwave source?'
      READ(6,*)SWFLUX
      SWFLUX=0.0D0
      WRITE(6,*)'Temperature of shortwave source?'
      READ(6,*)SWTEMP
      SWTEMP=5780.D0
      WRITE(6,*)'Flux from longwave source?'
      READ(6,*)LWFLUX
      WRITE(6,*)'Temperature of longwave source?'
      READ(6,*)LWTEMP
      WRITE(6,*)'Temperature of the FOVL?'
      READ(6,*)TFOVL
      WRITE(6,*)'Temperature of the USUB?'
      READ(6,*)TUSUB
      TUSUB=TFOVL
      WRITE(6,*)'Temperature of the LSUB?'
      READ(6,*)TLSUB
      WRITE(6,*)'Temperature of the primary aperture?'
      READ(6,*)TAPER
      TLSUB=TAPER
      TCAVSUB=TAPER
      WRITE(6,*)'Temperature of the CAVSUB?'
      READ(6,*)TCAVSUB
      WRITE(6,*)'Temperature of the RING?'
      READ(6,*)TRING
      TRING=TFOVL
      WRITE(6,*)'Temperature of the DOME?'
      READ(6,*)TDOME
      TDOME=0.0D0
      WRITE(6,*)'Temperature of the cavity?'
      READ(6,*)TCAV
      TCAV=308.0D0

      DO 10 N=1,NROW
        WRITE(3,*)SWFLUX,LWFLUX
        WRITE(29,*)TFOVL
        WRITE(30,*)TUSUB
        WRITE(31,*)TLSUB
```

```
      WRITE(32,*)TDOME
      WRITE(33,*)TRING
      WRITE(34,*)TCAVSUB
      WRITE(35,*)TCAV
      WRITE(36,*)SWTEMP,LWTEMP
 10   CONTINUE
      END
```

Appendix F

Program ERBEMESH

```
************************************************************************
*                                                                      *
*                          ERBEMSH FORTRAN                             *
*                                                                      *
*       This FORTRAN 77 program generates three-dimensional mesh       *
*   geometry data required for 'MOVIE.BYU' and two-dimensional mesh    *
*   geometry data required for 'ERBETA FORTRAN'.  The global node      *
*   coordinates, connectivity array and parts array for a three-       *
*   dimensional mesh are determined for 'MOVIE.BYU', and the local     *
*   node coordinates and connectivity array for a two-dimensional      *
*   mesh are determined for 'ERBETA FORTRAN'.                          *
*                                                                      *
*                                                                      *
*   Written by Nour E. Tira, and modified by Kory J. Priestley         *
*                                                                      *
************************************************************************
*                                                                      *
*                           Nomenclature                               *
*                                                                      *
*                                                                      *
*    ATIP ....... Area of the truncated tip of the cone.               *
*    DIM(I) ..... Radiometer part dimension.  Dimensioned 4.           *
*                 I = 1   cone height                                  *
*                 I = 2   cylinder height                              *
*                 I = 3   radial width of ring (RO - RI)               *
*                 I = 4   thermal impedance length.                    *
*    DM(J) ...... The element dimensions along DIM(I), for             *
*                 0 < I < 4.  Dimensioned two times the sum            *
*                 of MDIV(I), for I = 1 to 4.                          *
*    DN(I) ...... The element dimensions over the leg width.           *
*                 Dimentioned two times MDIV(4).                       *
*    H0 ......... The cone tip height.                                 *
*    JP(I) ...... Connectivity array of the three-dimensional mesh     *
*                 for 'MOVIE.BYU'.  Dimensioned NEDGE.                 *
*    K1 ......... The last node level in the cone mesh.                *
*    K2 ......... The last node level in the cavity mesh.              *
*    K3 ......... The last node level in the ring mesh.                *
*    K4 ......... Number of node levels in the entire model.           *
*    MDIV(I) .... Number of element divisions along DIM(I).            *
*                 Dimentioned 4.                                       *
*    NECN ....... Number of elements in the cone mesh.                 *
*    NECY ....... Number of elements in the cylinder mesh.             *
*    NEDGE ...... Number of element edges in the three-dimensional     *
*                 mesh.                                                *
*    NEM ........ Total number of elements in the mesh.                *
*    NERG ....... Number of elements in the ring mesh.                 *
*    NETI ....... Number of elements in the thermal impedance.         *
*    NNB ........ Number of nodes at the thermal impedance base,       *
*                 used as heat sink boundary conditions nodes          *
*    NNM ........ Total number of global nodes in the mesh.            *
*    NRT ........ First node of each node level on the thermal         *
*                 impedance, used to select the position of the        *
*                 resistance thermometer.                              *
*    NOD(N,J) ... Global node number corresponding to local node J     *
*                 of element N (connectivity array) of the             *
*                 two-dimensional mesh.  Dimensioned NEM by 9.         *
*    NP ......... Number of parts in the three-dimensional mesh.       *
*    NPL(I,J) ... The lower (I = 1) and upper (I = 2) limits on the    *
*                 element numbers in radiometer part J.  Dimensioned   *
*                 2 by NP.                                             *
*                 J = 1   cone                                         *
*                 J = 2   cylinder                                     *
*                 J = 3   ring                                         *
*                 J = 4   Thermal impedance                            *
*    NUNIF(I) ... Mesh uniformity flag for DIM(I).  If its             *
```

```
*                value is zero, then elements along dimension I      *
*                are spaced unequally and the element dimensions     *
*                must be input by the user.  If its value is one,    *
*                then elements along dimension I are equally         *
*                spaced and the element dimensions are calculated    *
*                by the mesh generator.  Dimensioned 6.              *
*    N1 ......... The number of elements around the circumference    *
*                of the radiometer cavity.                           *
*    RI ......... The inside radius of the ring (cavity radius).     *
*    RO ......... The outside radius of the ring.                    *
*    RT ......... The cone tip radius.                               *
*    X(J) ....... X-coordinate of global node J.  Dimensioned NNM.   *
*    XT(N,J) .... X-coordinate of local node J of element N of the   *
*                two-dimensional finite element mesh.  Dimensioned   *
*                NEM by 9.                                           *
*    Y(J) ....... Y-coordinate of global node J.  Dimensioned NNM.   *
*    YT(N,J) .... Y-coordinate of local node J of element N of the   *
*                two-dimensional finite element mesh.  Dimensioned   *
*                NEM by 9.                                           *
*    Z(J) ....... Z-coordinate of global node J.  Dimensioned NNM.   *
*                                                                    *
**********************************************************************
C
C  Request double precision real variables.
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C  Dimension the arrays and place variables used by the subroutines
C  in common storage blocks.
C
      COMMON /BLOCK1/ DIM(4),X(2000),Y(2000),Z(2000),
     &               DM(100),RI,RO,RT
      COMMON /BLOCK2/ N1,MDIV(4),K1,K2,K3,K4,NOD(900,9),
     &               NECN,NECY,NERG,NETI
      DIMENSION XT(400,9), YT(400,9)
      DIMENSION NUNIF(4), JP(5000), NPL(3,4)
C
C  Open all necessary input and output files.
C
      OPEN(1,FILE='msh.dat',status='old')
      OPEN(4,FILE='mesh.dat')
      OPEN(5,FILE='nodes.dat')
C
C  Set the logical unit numbers for I/O files and set the value of pi.
C
      PI = 3.141592654D0
C
C  Read the radiometer dimensions, mesh discretization data and the
C  mesh uniformity flags,
C
      READ(1,3) DIM(1),DIM(2),RI,RO,DIM(4),RT,H0
      READ(1,1) N1,(MDIV(I),I=1,4)
      READ(1,1) (NUNIF(I),I=1,4)
C
C  then determine the radial width of the ring.
C
      DIM(3) = RO - RI
C
C  Examine each uniformity flag to determine if any part of the mesh
C  is non-uniform.  Run a DO-loop over the number of dimensions.
C
      M = 0
      DO 40 I = 1,4
C
C  Determine the lower and upper bounds on the subscripts of DM(J)
```

```
C  for the dimension in question.
C
      N = M + 1
      M = M + (2*MDIV(I))
C
C  If the mesh is non-uniform along dimension I, read the element
C  dimensions for all DIM(I) except the leg width.
C
      IF(NUNIF(I).EQ.0) THEN
          READ(1,3) (DM(J),J=N,M)
C
C  If the mesh is uniform along dimension I, then determine the element
C  dimensions.
C
      ELSE
              DO 30 J = N,M
   30         DM(J) = DIM(I)/(M - N + 1)
      ENDIF
   40 CONTINUE
C
C  Generate the three-dimensional nodal coordinates for 'MOVIE.BYU'.
C  First, determine the number of node levels in the cavity and ring
C  meshes.
C
      K1 =  1 + (MDIV(1)*2)
      K2 = K1 + (MDIV(2)*2)
      K3 = K2 + (MDIV(3)*2)
      K4 = K3 + (MDIV(4)*2)
C  Find the last node number, i.e. The total number of elements.
      NNM = 2*N1*K4
C
C  Calculate the three-dimentional nodal coordinates of the model.
C
      CALL MSHCOR (X1,Y1,Z1)
C
C
C  Determine the number of elements in each part of the radiometer mesh
C  and the total number of elements in the mesh.
C
      NECN  = MDIV(1)*N1
      NECY  = MDIV(2)*N1
      NERG  = MDIV(3)*N1
      NETI = MDIV(4)*N1
      NEM = NECN + NECY + NERG + NETI
C
C
      CALL CUNECT (LGNODE,KK,JJ)
C
C  Determine the connectivity array for the three-dimensional mesh
C  needed for 'MOVIE.BYU' from the two-dimensional connectivity array.
C
      K = 0
      DO 80 I = 1,NEM
      N = 8
      DO 70 J = 1,N
      K = K + 1
      JP(K) = NOD(I,J)
      IF(J.EQ.N) JP(K) = -NOD(I,J)
   70 CONTINUE
   80 CONTINUE
C
C  Determine the parts array for 'MOVIE.BYU'.  There are four parts in
C  the model.
C
      NP = 4
```

```
      NPL(1,1) = 1
      NPL(2,1) = NECN
      NPL(1,2) = NECN + 1
      NPL(2,2) = NECN + NECY
      NPL(1,3) = NECN + NECY + 1
      NPL(2,3) = NECN + NECY + NERG
      NPL(1,4) = NECN + NECY + NERG + 1
      NPL(2,4) = NECN + NECY + NERG + NETI
C
C  Determine the number of edges in the model for 'MOVIE.BYU'.
C  Each quadrilateral element has eight edges.
C
      NEDGE = 0
      DO 90 I = 1,NEM
   90 NEDGE = NEDGE + 8
C
C  Calculate the two-dimensional local node coordinates for the finite
C  element mesh.
C
      CALL TWODIM (XT,YT)
C
C  Determine the nodes at the base of the thermal impedance,
C  and write them to the data file.
C
      NNB = 2*N1
      WRITE(5,1) NNB
      DO 100 I = 1,NNB
      NBTI = NNM - NNB + I
  100 WRITE(5,1) NBTI
C
C  Determine the nodes along the thermal impedance, used to select
C  the position of the resistance thermometer, and write them to
C  the data file.
C
      N = MDIV(4)*2
      WRITE(5,1) N
      NRT   = NNM - 2*N1 + 1
      DO 110 I = 1,N
      NRT   = NRT   - 2*N1
  110 WRITE(5,1) NRT
C
C  Write the mesh geometry data file for 'ERBETA FORTRAN'.
C
      WRITE(4,1) NEM,NNM,(MDIV(I),I=1,4),N1
      WRITE(4,3) (DIM(I),I = 1,4),RI,RO,RT,H0
      WRITE(4,3) (DM(I),I=1,(MDIV(1)+MDIV(2)+MDIV(3)+MDIV(4))*2)
      DO 120 I=1,NEM
  120 WRITE(4,1) (NOD(I,J),J=1,9)
      DO 130 I=1,NEM
      WRITE(4,3) (XT(I,J),YT(I,J),J=1,9)
  130 CONTINUE
      STOP
C
C  The format statements follow:
C
    1 FORMAT(16I5)
    2 FORMAT(6E12.5)
    3 FORMAT(6D12.5)
    4 FORMAT(1I5,1X,6D12.5)
      END
************************************************************************
*                                                                      *
*                         SUBROUTINE MSHCOR                            *
*                                                                      *
*       This subroutine is called by the main program to determine     *
```

```
*   the three-dimensional coordinates of the nodes of the entire      *
*   mesh.                                                             *
*                                                                     *
***********************************************************************
*                                                                     *
*                           Nomenclature                              *
*                                                                     *
*                                                                     *
*    A .......... Depth of the node level below the aperature.        *
*    DIM(I) ..... Radiometer part dimension.  Dimensioned 6.          *
*                 I = 1   cone height                                 *
*                 I = 2   cylinder height                             *
*                 I = 3   radial width of ring (RO - RI)              *
*                 I = 4   thermal impedance length.                   *
*    DM(J) ...... The element dimensions along DIM(I), for            *
*                 I = 1 to 4.  Dimensioned two times the sum          *
*                 of MDIV(I), for I = 1 to 4.                         *
*    DTHETA ..... The angle between nodes on a common level.          *
*    I .......... The node level.                                     *
*    K1 ......... Number of node levels in the cone mesh.             *
*    K2 ......... Number of node levels in the cavity mesh.           *
*    K3 ......... Number of node levels in the cavity plus ring mesh. *
*    K4 ......... Number of node levels in the entire mesh.           *
*    M .......... Upper limit on the node numbers of node level I.    *
*    N .......... Lower limit on the node numbers of node level I.    *
*    N1 ......... The number of elements around the circumference     *
*                 of the radiometer cavity.                           *
*    PI ......... Mathematical constant.                              *
*    R .......... Radius of the node level.                           *
*    RI ......... The inside radius of the ring (cavity radius).      *
*    RT ......... The cone tip radius.                                *
*    THETA ...... The nodal angle measured counter-clockwise from     *
*                 the positive x-axis.                                *
*    X(J) ....... X-coordinate of global node J.  Dimensioned NNM.    *
*    Y(J) ....... Y-coordinate of global node J.  Dimensioned NNM.    *
*    Z(J) ....... Z-coordinate of global node J.  Dimensioned NNM.    *
*                                                                     *
***********************************************************************
      SUBROUTINE MSHCOR(X1,Y1,Z1)
C
C  Request double precision real variables and place variables used by
C  the subroutine in common storage blocks.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /BLOCK1/ DIM(4),X(2000),Y(2000),Z(2000),
     &              DM(100),RI,RO,RT
      COMMON /BLOCK2/ N1,MDIV(4),K1,K2,K3,K4,NOD(900,9),
     &              NECN,NECY,NERG,NETI
C
C  Define the mathematical constant pi, and initialize the depth of the
C  cavity.
C
      PI = 3.141592654D0
      A = DIM(1) + DIM(2)
      RR = DSQRT(RI**2+DIM(1)**2)
C
C  Run a DO-loop over the number of node levels in the cavity.  The
C  lower and upper limits on the node numbers of the node level in
C  question are determined on each loop.
C
      M = 0
      DO 20 I = 1,K4
      N = M + 1
      M = N + (2*N1) - 1
C
```

```
C  Determine the radius of each node level on the cone.
C  At the tip, the radius is RT.
C
      IF(I.EQ.1) R = RT
C
      IF(I.LT.K1.AND.I.GT.1) R = R + ((RI-RT)*DM(I-1)/DIM(1))
C
C  For all node levels in the cylinder, ther radius is RI.
C
      IF(I.GE.K1.AND.I.LE.K2) R = RI
C
C  Determine the radius of each node level on the ring.
C
      IF(I.GT.K2.AND.I.LT.K3) R = R + DM(I)
C
C  For all node levels on the thermal impedance, the radius is RO.
C
      IF(I.GE.K3) R = RO
C  Determine the angle between two consecutive nodes, and initialize
C  the angle of the first node.
C
      DTHETA = PI*2.0D0/(M - N + 1)
      THETA =  PI + DTHETA
C
C  Run a DO-loop over the node numbers of the level in question.
C
      DO 10 J = N,M
C
C  Calculate the coordinates of node J on node level I, then increment
C  the node angle.
C
      X(J)  = R*DCOS(THETA)
      Y(J)  = R*DSIN(THETA)
      Z(J)  = A
      THETA = THETA - DTHETA
   10 CONTINUE
C
C  Increment the depth of the node level. "A" at cone tip, zero on the
C  upper edge of the cylinder, and DIM(5) at bottom of the thermal
C  impedance.
C
      IF(I.LT.K2) A = A - DM(I)
      IF(I.GE.K2.AND.I.LT.K3) A = 0.0D0
      IF(I.GE.K3) A = A + DM(I)
   20 CONTINUE
C
C  Return to the main program.
C
      RETURN
      END
C
C
************************************************************************
*                                                                      *
*                      SUBROUTINE CUNECT                               *
*                                                                      *
*        THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM TO CALCULATE    *
*    THE CONNECTIVITY ARRAY FOR THE FINITE ELMENT MESH.                *
*                                                                      *
************************************************************************
*                                                                      *
*                           NOMENCLATURE                               *
*                                                                      *
*    I ......... NODE WHICH DEFINES ONE ENDPOINT OF THE LINE NEEDED    *
*                to determine the coordinates of node NC.              *
```

```
*    K .......... Element level.                                      *
*    NB ......... Node which defines one endpoint of the line needed  *
*                 to determine the coordinates of node NC.            *
*    NC ......... Node which lies at the intersection of a circle of  *
*                 radius R and the line defined by nodes NA and NB.   *
*    NERG ....... Number of elements in the ring mesh.                *
*    PI ......... Mathematical constant.                              *
*    R1 ......... Radius of the I-2 node level.                       *
*    R .......... Radius of the I-th node level.                      *
*    RI ......... The inside radius of the ring (cavity radius).      *
*    RO ......... The outside radius of the ring.                     *
*    RR ......... Radius of the I-1 node level.                       *
*    THETA ...... The nodal angle measured counter-clockwise from     *
*                 the positive x-axis.                                *
*    X(J) ....... X-coordinate of global node J.  Dimensioned NNM.    *
*    Y(J) ....... Y-coordinate of global node J.  Dimensioned NNM.    *
*    Z(J) ....... Z-coordinate of global node J.  Dimensioned NNM.    *
*                                                                     *
***********************************************************************
C
      SUBROUTINE CUNECT (LGNODE,KK,JJ)
C
C  Request double precision real variables and place variables used
C  by the subroutine in common storage blocks.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /BLOCK1/ DIM(4),X(2000),Y(2000),Z(2000),
     &              DM(100),RI,RO,RT
      COMMON /BLOCK2/ N1,MDIV(4),K1,K2,K3,K4,NOD(900,9),
     &              NECN,NECY,NERG,NETI
C
C
C  Determine the terms of the connectivity array for the model.
C
      K = MDIV(1) + MDIV(2) + MDIV(3) + MDIV(4)
      DO 30 N = 1,K
        DO 20 I = ((N1*(N-1)+1)),(N*N1)
        NOD(I,1) = 2*I-1
        IF(N.EQ.1) GO TO 39
        NOD(I,1) = NOD(I-N1,7)
 39     NOD(I,2) = NOD(I,1) + 1
        NOD(I,3) = NOD(I,1) + 2
        NOD(I,8) = NOD(I,1) + 2*N1
        NOD(I,9) = NOD(I,8) + 1
        NOD(I,4) = NOD(I,8) + 2
        NOD(I,7) = NOD(I,1) + 4*N1
        NOD(I,6) = NOD(I,7) + 1
 20     NOD(I,5) = NOD(I,7) + 2
      NOD(I-1,3) = NOD((N1*(N-1))+1,1)
      NOD(I-1,4) = NOD((N1*(N-1))+1,8)
      NOD(I-1,5) = NOD((N1*(N-1))+1,7)
 30   CONTINUE
      RETURN
      END
***********************************************************************
*                                                                     *
*                       SUBROUTINE TWODIM                             *
*                                                                     *
*      This subroutine is called by the main program to calculate    *
*   the two-dimensional local node coordinates of the finite element *
*   mesh.  Only the relative positions of the nodes of an element    *
*   need be calculated as defined by 'ERBETA'.                       *
*      The cone and cylinder are each cut between the first and      *
*   last elements on the same element level and laid flat in an      *
*   xy-plane to produce two-dimensional surfaces.  The local node    *
```

```
*   coordinates of the first element on each level of the cavity can   *
*   then be calculated and repeated for each element of that level      *
*   due to the radial symmetry of the cavity.                           *
*                                                                      *
************************************************************************
*                                                                      *
*                           NOMENCLATURE                               *
*                                                                      *
*                                                                      *
*    DIM(I) ..... Radiometer part dimension.  Dimensioned 6.           *
*                 I = 1   cone height                                  *
*                 I = 2   cylinder height                              *
*                 I = 3   radial width of ring (RO - RI)               *
*                 I = 4   cylinder leg length                          *
*    DTHETA ..... The angle between nodes on a common level.           *
*    DR(I) ...... The radius of the I-th cone node level.              *
*                 Dimensioned 2*MDIV(1).                               *
*    ELX(I,J) ... The x-coordinate of local node I of the first        *
*                 element (J) of a level.  Dimensioned                 *
*                 9 by MDIV(1)+MDIV(2).                                *
*    ELY(I,J) ... The y-coordinate of local node I of the first        *
*                 element (J) of a level.  Dimensioned                 *
*                 9 by MDIV(1)+MDIV(2).                                *
*    MDIV(I) .... Number of element divisions along DIM(I).            *
*                 DIMENSIONED 4.                                       *
*    NECN ....... Number of elements in the cone mesh.                 *
*    NECY ....... Number of elements in the cylinder mesh.             *
*    NI ......... Global node number.                                  *
*    NOD(N,J) ... Global node number corresponding to local node J     *
*                 of element N (connectivity array) of the             *
*                 two-dimensional mesh.  Dimensioned NEM by 9.         *
*    NPE ........ Number of local nodes in an element.                 *
*    N1 ......... The number of elements around the circumference      *
*                 of the radiometer cavity.                            *
*    PI ......... Mathematical constant.                               *
*    RI ......... The inside radius of the ring (cavity radius).       *
*    RT ......... The cone tip raduis.                                 *
*    THETA ...... The nodal angle measured counter-clockwise from      *
*                 the positive x-axis.                                 *
*    X(J) ....... X-coordinate of global node J.  Dimensioned NNM.     *
*    XT(N,J) .... X-coordinate of local node J of element N of the     *
*                 two-dimensional finite element mesh.  Dimensioned    *
*                 NEM by 9.                                            *
*    Y(J) ....... Y-coordinate of global node J.  Dimensioned NNM.     *
*    YT(N,J) .... Y-coordinate of local node J of element N of the     *
*                 two-dimensional finite element mesh.  Dimensioned    *
*                 NEM by 9.                                            *
*    Z(J) ....... Z-coordinate of global node J.  Dimensioned NNM.     *
*                                                                      *
************************************************************************
      SUBROUTINE TWODIM (XT,YT)
C
C  Request double precision real variables, place variables used
C  by the subroutine in common storage blocks and dimension the arrays.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /BLOCK1/ DIM(4),X(2000),Y(2000),Z(2000),
     &              DM(100),RI,RO,RT
      COMMON /BLOCK2/ N1,MDIV(4),K1,K2,K3,K4,NOD(900,9),
     &              NECN,NECY,NERG,NETI
      DIMENSION XT(400,9),YT(400,9)
      DIMENSION ELX(9,20),ELY(9,20),DR(20)
C
C  Define the mathematical constant pi, and determine the radius of the
C  sector of a circle created by the two-dimensional cone surface.
```

```
C
      PI = 3.141592654D0
      RR = DSQRT((RI**2) + (DIM(1)+H0)**2)
C
C  Determine the angle between the nodes on the second node level and
C  the radial distance between node levels.
C
      DTHETA = (RI*PI)/(RR*N1)
      SUM = 0.0D0
      DO 10 I = 1,2*MDIV(1)
      SUM = SUM + DM(I)
   10 DR(I) = SUM*RR/DIM(1)
C
C  INITIALIZE THE NODE ANGLE AND THE COORDINATES OF NODE 1.
C  If there is more than one division along the cone height, then
C  determine the local coordinates of the nodes of the first
C  quadrilateral element on each remaining level of the cone.
C  A DO-loop is run over the  number of remaining element levels
C  of the cone, and the angle and radius of the first local node
C  are defined on each loop.
C
          DO 60 J = 1,MDIV(1)
          THETA = PI/2.0D0
          IF(J.EQ.1) GO TO 11
          RAD = DR(2*J-2)
          GO TO 12
  11      RAD = RT*(RR/RI)
C
C  Determine the coordinates of the first three local nodes,
C
  12      DO 30 I = 1,3
          ELX(I,J) = RAD*DCOS(THETA)
          ELY(I,J) = RAD*DSIN(THETA)
   30     THETA = THETA - DTHETA
C
C  the fourth local node,
C
          RAD = DR(2*J-1)
          THETA = (PI/2.0D0) - 2.0D0*DTHETA
          ELX(4,J) = RAD*DCOS(THETA)
          ELY(4,J) = RAD*DSIN(THETA)
C
C  the fifth through seventh local nodes,
C
          RAD = DR(2*J)
          DO 40 I = 5,7
          ELX(I,J) = RAD*DCOS(THETA)
          ELY(I,J) = RAD*DSIN(THETA)
   40     THETA = THETA + DTHETA
C
C  and the eighth and ninth local nodes.
C
          THETA = PI/2.0D0
          RAD = DR(2*J-1)
          DO 50 I = 8,9
          ELX(I,J) = RAD*DCOS(THETA)
          ELY(I,J) = RAD*DSIN(THETA)
   50     THETA = THETA - DTHETA
   60     CONTINUE
C
C  The coordinates of the local nodes of the first element on each
C  level of the cylinder are calculated by running a DO-loop over the
C  number of element levels in the cylinder.  The distance between
C  nodes on a common level is determined first.
C
```

```
      DC1 = (RI*PI)/N1
      L1 = MDIV(1) + 1
      DO 70 J = L1,MDIV(2)+MDIV(1)
C
C  Calculate the coordinates of all nine nodes of the element.
C
      ELX(1,J) = 0.0D0
      ELX(2,J) = DC1
      ELX(3,J) = 2.0D0*DC1
      ELX(4,J) = 2.0D0*DC1
      ELX(5,J) = 2.0D0*DC1
      ELX(6,J) = DC1
      ELX(7,J) = 0.0D0
      ELX(8,J) = 0.0D0
      ELX(9,J) = DC1
      ELY(1,J) = 0.0D0
      ELY(2,J) = 0.0D0
      ELY(3,J) = 0.0D0
      ELY(4,J) = DM(2*MDIV(1)+1)
      ELY(5,J) = 2.0D0*ELY(4,J)
      ELY(6,J) = ELY(5,J)
      ELY(7,J) = ELY(5,J)
      ELY(8,J) = ELY(4,J)
      ELY(9,J) = ELY(4,J)
   70 CONTINUE
C
      DC2 = (RO*PI)/N1
      L2 = MDIV(1) + MDIV(2) + MDIV(3)+ 1
      DO 71 J = L2,(L2 + MDIV(4)-1)
         ELX(1,J) = 0.0D0
         ELX(2,J) = DC2
         ELX(3,J) = 2.0D0*DC2
         ELX(4,J) = ELX(3,J)
         ELX(5,J) = ELX(3,J)
         ELX(6,J) = DC2
         ELX(7,J) = 0.0D0
         ELX(8,J) = 0.0D0
         ELX(9,J) = DC2
         ELY(1,J) = 0.0D0
         ELY(2,J) = 0.0D0
         ELY(3,J) = 0.0D0
         ELY(4,J) = DM(2*L2-1)
         ELY(5,J) = 2.0D0*ELY(4,J)
         ELY(6,J) = ELY(5,J)
         ELY(7,J) = ELY(5,J)
         ELY(8,J) = ELY(4,J)
         ELY(9,J) = ELY(4,J)
   71 CONTINUE
C
C  Now that the local node coordinates of the first element on each
C  level of the cavity are known, determine the local coordinates
C  of all the element nodes in the cavity.  DO-loops are run over the
C  number of element levels and the element numbers of the level.
C
      N = 1
      M = N1
      DO 100 J = 1,MDIV(1)+MDIV(2)
      DO 90 I = N,M
C
C  Determine the number of nodes in the element, then loop over each of
C  these nodes to determine the nodal coordinates of the element.
C
      NPE = 9
      DO 80 K = 1,NPE
      XT(I,K) = ELX(K,J)
```

```
         YT(I,K) = ELY(K,J)
   80 CONTINUE
   90 CONTINUE
C
C  Increment the element numbers for the next level.
C
      N = M + 1
      M = M + N1
  100 CONTINUE
C
C  The local coordinates of the nodes of the ring and thermal impedance
C  can be found by using the three-dimensional coordinates and the
C  connectivity array.
C  Initialize the element limits as the element numbers in the ring,
C  and then run a DO-loop over the two pieces.
C
      L3 = L2 + MDIV(4) - 1
      N = (L2-1)*N1 + 1
      M = N + N1 - 1
      DO 101 J = L2,L3
        KI = 1
        DC = 0
        DO 91 I = N,M
          DO 81 K = 1,9
          XT(I,K) = ELX(K,J)+ 2*DC
          YT(I,K) = ELY(K,J)
   81     CONTINUE
          DC = KI*DC2
          KI = KI + 1
   91   CONTINUE
          DO 82 K = 1,9
   82     ELY(K,J+1) = YT(I-1,K) +2*DM(2*L2-1)
C
C  Increment the element number for the next level.
C
        N = M +1
        M = M + N1
  101 CONTINUE
      N = NECN + NECY + 1
      M = N - 1 + NERG
      DO 120 I = N,M
         DO 110 J = 1,9
           NI = NOD(I,J)
           XT(I,J) = X(NI) + RI
           YT(I,J) = Y(NI) + RI
  110    CONTINUE
  120 CONTINUE
      RETURN
      END
```

Appendix G

Program ERBETA

```
***********************************************************************
*                                                                     *
*                          ERBETA  FORTRAN                            *
*                                                                     *
***********************************************************************
*                                                                     *
*      This program performs a steady-state or transient thermal      *
*      analysis on an active cavity radiometer using a finite         *
*      element method which uses quadratic quadrilateral elements.    *
*      The analysis considers thermal conduction and radiation.       *
*                                                                     *
*      The utilization of program flags allows the user several       *
*      options in the type of analysis performed.  The three major    *
*      flags are as follows:                                          *
*                                                                     *
*           1. Steady-state or transient analysis                     *
*           2. Conduction with or without radiative effects           *
*           3. Electrical substitution heater on or off.              *
*                                                                     *
*      The analysis considers radiative interaction between the       *
*      source field and the inside surface of the cavity with the     *
*      option of having one irradiance vector entering the cavity.    *
*      Radiative interaction between the outside of the cavity or     *
*      any other part of the radiometer with its surroundings will    *
*      not be considered in this analysis.                            *
*                                                                     *
*      Written by Nour E. Tira, and modified by Kory J. Priestley     *
*                                                                     *
***********************************************************************
*                                                                     *
*                           NOMENCLATURE                              *
*                                                                     *
*    ALPHA ........ Resistance-temperature constant.                  *
*    ATIP ......... Area of the truncated tip of the cone.            *
*    COND ......... Conductivity of the radiometer material.          *
*    CP ........... Specific heat of the radiometer material.         *
*    DELT ......... Temperature drop between the resistance           *
*                   thermometers and the heat sink.                   *
*    DENS ......... Density of the radiometer material.               *
*    DIM(I) ....... Radiometer part dimensions.  Dimensioned 6        *
*                   I = 1  cone height                                *
*                   I = 2  cylinder height                            *
*                   I = 3  radial width of the ring (RO - RI)         *
*                   I = 4  thermal impedance length.                  *
*    DM(I) ........ The element dimensions along the cavity height.   *
*                   Dimensioned 2*(MDIV(1) + MDIV(1)).                *
*    DT ........... Time increment.                                   *
*    EM ........... Emissivity of the radiometer material.            *
*    ERROR ........ Convergence error of the solution.                *
*    EO ........... Bridge voltage for the feedback circuit.          *
*    E2 ........... Voltage across the heater wire.                   *
*    GEN(N,K) ..... Volumetric heat generation in element N.          *
*                   Dimensioned NEM by 2.                             *
*    GTA(I,J) ..... Assembled global conductivity matrix              *
*                   Dimensioned NNM by NHBW.                          *
*    GTF(I) ....... Assembled global flux vector.  Dimensioned NNM.   *
*    HWIRE ........ Height to which the heater wire is wound above    *
*                   the cone tip.                                     *
*    INIT ......... The indicator for transient analysis.             *
*                   INIT = 0   steady-state analysis                  *
*                   INIT = 1   transient, uniform initial conditions  *
*                   INIT = 2   transient, non-uniform i.c.            *
*    IPDF(I) ...... The array of node numbers with specified temper-  *
*                   ature boundary conditions.  Dimensioned NPDF.     *
*    MDIV(I) ...... Number of element divisions along DIM(I).         *
```

```
*                   Dimensioned 6.                                    *
*    NC ........... Column of the global conductivity matrix.         *
*    NCMAX ........ Column-dimension of GTA(I,J).                     *
*    NDT .......... Number of time steps in the analysis--equal to    *
*                   one for a steady-state analysis.                  *
*    NEC .......... Number of elements in the cavity mesh.            *
*    NEM .......... Number of elements in the mesh.                   *
*    NHBW ......... Half-band width of the system of global equations.*
*    NNM .......... Number of nodes in the mesh                       *
*    NOD(N,J) ..... Global node number corresponding to the J-th node *
*                   of element N (connectivity array) of the mesh.    *
*                   The local nodes are defined counter-clockwise     *
*                   from any elemnt corner with the central node      *
*                   of the quadrilateral elements given last          *
*                   Dimensioned NEM by 9.                             *
*    NPDF ......... The number of specified temperature boundary      *
*                   conditions.                                       *
*    NR ........... Row of the global conductivity matrix and         *
*                   flux vector.                                      *
*    NRMAX ........ Row-dimension of GTA(I,J) and GTF(I).             *
*    NSTEP ........ Time step number.                                 *
*    NRES ......... The first node on the thermal impedance which     *
*                   is covered by the resistance thermometer.         *
*    N1 ........... The number of elements around the circumference   *
*                   of the cavity.                                    *
*    PHI .......... The collimated beam angle of incidence w/r to the *
*                   aperature normal. Dimensioned NPHI.               *
*    RH ........... Resistance of the heater wire.                    *
*    RI ........... Inside radius of the ring (cavity radius).        *
*    RIP .......... Precision aperture radius.                        *
*    RO ........... Outside radius of the ring.                       *
*    TA(I,J) ...... Element conductivity matrix.  Dimensioned 9 by 9. *
*    TARES ........ Average temperature of the two resistance         *
*                   thermometers.                                     *
*    TAU .......... Time constant in the analysis of the heater       *
*                   circuit.                                          *
*    TAVE(I) ...... Average temperature of cavity element I.          *
*                   Dimensioned NEC.                                  *
*    TF(I) ........ Element flux vector.  Dimensioned 9.              *
*    THETA ........ Time approximation parameter.                     *
*                   Theta = 0     forward-difference scheme           *
*                   Theta = 1/2   Crank-Nicolson scheme               *
*                   Theta = 2/3   Galerkin scheme                     *
*                   Theta = 1     backward-difference scheme          *
*    THCK ......... Thickness of the radiometer material.             *
*    THS .......... Temperature of the heat sink.                     *
*    TIME ......... The running time for a transient analysis.        *
*    TMAX ......... Upper bound on time for a transient analysis.     *
*    TNEW(I) ...... Global node temperatures calculated for the       *
*                   current time step.  Dimensioned NNM.              *
*    TOLD(I) ...... Column of local node temperatures from the        *
*                   previous time step                                *
*    TINIT  ....... Radiometer predominant initial temperature.       *
*    VPDF(I) ...... Nodal temperatures specified as boundary condi-   *
*                   tions.  Dimensioned NPDF.                         *
*    X(N,J) ....... X-coordinate of local node J of element N of the  *
*                   mesh.  Dimensioned NEM by 9.                      *
*    Y(N,J) ....... Y-coordinate of local node J of element N of the  *
*                   mesh.  Dimensioned NEM by 9.                      *
*                                                                     *
***********************************************************************
C
C  Request double precision real variables.
C
      IMPLICIT REAL*8 (A-H,O-Z)
```

```
      REAL*8 LBDA
C
C  Dimension the arrays, and specify variables used by the subroutines
C  in common storage blocks.
C
      DIMENSION GTA(800,120),GTF(800),GEN(200,2),TA(9,9),TF(9)
      DIMENSION IPDF(60),VPDF(60),FMAG(256)
      DIMENSION NOD(200,9),MDIV(4),DM(40),PHI(35)
      DIMENSION TNEW(800),TAVE(150),TOLD(9),X(200,9),Y(200,9)


      DIMENSION DFOPT(200,100,15),DFSCN(200,100,15),DFBAR(200,100,15)
      DIMENSION DFDOME(200,100,15),DFRING(200,100,15),DFCSUB(200,100,15)
      DIMENSION DFUSUB(200,100,15),DFLSUB(200,100,15),DFCONE(200,100,15)
      DIMENSION DOABCO(25),LBDA(20)

      COMMON/ RADI1/ AREA1,AREA2,PI,RI,THCK,DIM(4),NVEC,NSUN,NS

      COMMON/ RADI3/ AFOVL,AUSUB,ALSUBI,ALSUBO,ARING,VOL1,VOL2,
     &               ACAVSUB,AFOVAP,ABAR,ACON
      COMMON/ HEAT1/ HEATEL(15),EO,ALPHA,TAU,THS,DELT,NRES

      COMMON/ ELEM1/ DT,THETA,COND,CP,DENS
      COMMON/ NB/ NBFOVL,NBUSUB,NBLSUB,NBCAVSUB,NBRING,
     &            NBDOME,NBSCENE,NBBAR,NBCON,NBCAV
      COMMON/ EMISS/ EMFOVL,EMUSUB,EMLSUB,EMDOME,EMBAR,EMCON,
     &               EMCAVSUB,EMRING
      COMMON/ DFACS/
DFOPT,DFSCN,DFBAR,DFCONE,DFDOME,DFRING,DFCSUB,
     &               DFUSUB,DFLSUB
      COMMON/ DOME/ DOMTHC,DOABCO,NBWLIN
      COMMON/ CHAN/ CHANEL
      COMMON/ LMBDA/LBDA
      COMMON/ FLUX/ SCNFLUX
C
C  Open input files
C
      OPEN(3,FILE='source.dat',status='old')
      OPEN(4,FILE='mesh.dat',status='old')
      OPEN(8,FILE='flags.dat',status='old')
      OPEN(9,FILE='props.dat',status='old')
      OPEN(10,FILE='time.dat',status='old')
      OPEN(11,FILE='bndtmp.dat',status='old')
      OPEN(12,FILE='emiss.dat',status='old')
      OPEN(13,FILE='heater.dat',status='old')
      OPEN(15,FILE='temp.dat',status='old')
      OPEN(16,FILE='numb.dat',status='old')
      OPEN(17,FILE='er.dat')
      OPEN(20,FILE='dfopt.dat',status='old')
      OPEN(21,FILE='dfscn.dat',status='old')
      OPEN(22,FILE='dfcav.dat',status='old')
      OPEN(23,FILE='dfdome.dat',status='old')
      OPEN(24,FILE='dfring.dat',status='old')
      OPEN(25,FILE='dome.dat',status='old')
      OPEN(26,FILE='out.dat')
      OPEN(27,FILE='tmpdst.dat')
      OPEN(28,FILE='htout.dat')
      OPEN(29,FILE='tfovl.dat',status='old')
      OPEN(30,FILE='tusub.dat',status='old')
      OPEN(31,FILE='tlsub.dat',status='old')
      OPEN(32,FILE='tdome.dat',status='old')
      OPEN(33,FILE='tring.dat',status='old')
      OPEN(34,FILE='tcavsub.dat',status='old')
      OPEN(35,FILE='tcav.dat',status='old')
      OPEN(36,FILE='tscn.dat',status='old')
```

```
      OPEN(40,FILE='area.dat',status='old')
      OPEN(50,FILE='lbda.dat',status='old')
      OPEN(65,FILE='dfcsub.dat',status='old')
      OPEN(66,FILE='dfusub.dat',status='old')
      OPEN(67,FILE='dflsub.dat',status='old')
      OPEN(68,FILE='dfbar.dat',status='old')
      OPEN(70,FILE='dfcone.dat',status='old')

      OPEN(99,FILE='wswblip.dfc',status='old')
C
C  Define program constants.
C
      WRITE(26,*)'      TIME          TARES          VOLTAGE
     &    POWER          SCNPOW          POW+SCN'
      DATA NRMAX,NCMAX/800,120/
      DATA MCOUNT,E2,RH,SMAX/20,0.D0,1.D0,0.D0/
      DATA NDT,NDTS,NDTSH/1,1,1/
      PI = 3.141592654D0
      ERRMAX = 0.50D-05
***********************************************************************
*                                                                     *
*                       PREPROCESSOR UNIT                             *
*                                                                     *
*                   Reads and generates data                          *
*                                                                     *
***********************************************************************
C
C  Read the mesh geometry data and determine the number of elements
C  in the cavity.
C
      READ(4,300) NEM,NNM,(MDIV(I),I=1,4),N1
      READ(4,310) (DIM(I),I=1,4),RI,RO,RT,HO
      READ(4,310) (DM(I),I=1,(MDIV(1)+MDIV(2)+MDIV(3)+MDIV(4))*2)
      NEC = (MDIV(1) + MDIV(2))*N1
      write(17,*)'NEM, NNM, N1',NEM,NNM,N1
      write(17,*)'RI,RO,RT,HO',RI,RO,RT,HO
      write(17,*)'NEC',NEC
      write(17,*)'MDIV(I)',(MDIV(I),I=1,4)

C
C  Read the areas for each surface of the radiometer
C
      READ(40,*)AFOVL
      READ(40,*)AUSUB
      READ(40,*)ALSUBI
      READ(40,*)ALSUBO
      READ(40,*)ARING
      READ(40,*)VOL1
      READ(40,*)VOL2
      READ(40,*)ACAVSUB
      READ(40,*)AFOVAP
      READ(40,*)ABAR
      READ(40,*)ACON
      WRITE(17,*)'AFOVL',AFOVL
      WRITE(17,*)'AUSUB',AUSUB
      WRITE(17,*)'ALSUBI',ALSUBI
      WRITE(17,*)'ALSUBO',ALSUBO
      WRITE(17,*)'ARING',ARING
      WRITE(17,*)'VOL1',VOL1
      WRITE(17,*)'VOL2',VOL2
      WRITE(17,*)'ACAVSUB',ACAVSUB
      WRITE(17,*)'AFOVAP',AFOVAP
      WRITE(17,*)'ABAR',ABAR
      WRITE(17,*)'ACON',ACON
C
```

```
C  Calculate the areas of the cut-off cone tip, the element on
C  the cone and those on the cylinder.
      ATIP = PI*RT*DSQRT(RT**2+HO**2)
      AREA1 = (PI*RI*DSQRT(RI**2 + (DIM(1)+HO)**2)-ATIP)/(N1*MDIV(1))
      AREA2 = (PI*RI*2.0D0*DIM(2))/(N1*MDIV(2))
C
C  Read the connectivity array.
C
      DO 10 I = 1,NEM
      READ(4,300) (NOD(I,J),J=1,9)
   10 CONTINUE
C
C  Read the local node coordinates of each element in the mesh.
C
      DO 20 I = 1,NEM
      READ(4,310) (X(I,J),Y(I,J),J=1,9)
   20 CONTINUE
C
C  Read the program flags.
C
      READ(8,*)CHANEL

      READ(8,300) INIT,NPDF,NRES
C
C  Read the cavity material properties.
C
      READ(9,310) COND,THCK,CP,DENS
C
C  If the number of specified temperature boundary conditions is
C  greater than zero, then read the nodes and their specified
C  temperatures.
C
      IF(NPDF.GT.0) THEN
          READ(11,300) (IPDF(I),I=1,NPDF)
          READ(11,310) (VPDF(I),I=1,NPDF)
          THS = VPDF(NPDF)
      ENDIF
C
C
C  If a transient thermal analysis is desired, then read the time
C  increment, the time limit, and the time approximation parameter.
C
      IF (INIT.GT.0) THEN
        READ(10,*) DT,TMAX,THETA,TSON
        READ(10,*) NDT,NDTF,NDTV,NDTS
      ENDIF
      READ(35,*) TCAV
      DO 32 I=1,NNM
        TNEW(I) = TCAV
   32 CONTINUE
C
C
C
C  If the initial conditions are non-uniform, i.e. different
C  zones have varying temperatures, read the initial temperature
C  distribution for each zone.
C
      IF (INIT.EQ.2) READ(9,305) (TNEW(I), I = 1,NNM)
C
C  Read the radiative surface properties for the instrument
C
      READ(12,*) EMFOVL
      READ(12,*) EMUSUB
      READ(12,*) EMLSUB
      READ(12,*) EMDOME
```

```
      READ(12,*) EMBAR
      READ(12,*) EMCON
      READ(12,*) EMCAVSUB
      READ(12,*) EMRING
C
C  Read the number of elements on each surface of the instrument
C
      READ(16,*) NBFOVL
      READ(16,*) NBUSUB
      READ(16,*) NBLSUB
      READ(16,*) NBCAVSUB
      READ(16,*) NBRING
      READ(16,*) NBDOME
      READ(16,*) NBSCENE
      READ(16,*) NBBAR
      READ(16,*) NBCON

      NBCAV = NBBAR+NBCON
C
C  Read the distribution of wavelength intervals.
C
      READ(25,*)NBWLIN
      write(6,*)nbwlin
      DO 400 I=1,NBWLIN
        READ(50,*)LBDA(I)
  400 CONTINUE
C
C  Read the distribution factors for the front end
C
      WRITE(6,*)NBWLIN
      DO 50 I = 1,NBFOVL
      DO 50 K=1,NBWLIN
      DO 50 J = 1,NEC
        READ(20,*)DFOPT(I,J,K)
   50 CONTINUE
C
C  For the upper substrate
C
      DO 52 I=1,NBUSUB
      DO 52 K=1,NBWLIN
      DO 52 J=1,NEC
        READ(66,*)DFUSUB(I,J,K)
   52 CONTINUE
C
C For the lower substrate
C
      DO 58 I=1,NBLSUB
      DO 58 K=1,NBWLIN
      DO 58 J=1,NEC
        READ(67,*)DFLSUB(I,J,K)
   58 CONTINUE
C
C  For the barrel
C
      DO 70 I=1,NBBAR
      DO 70 K=1,NBWLIN
      DO 70 J=1,NEC
        READ(68,*)DFBAR(I,J,K)
   70 CONTINUE
C
C  For the cone
C
      DO 75 I=1,NBCON
      DO 75 K=1,NBWLIN
      DO 75 J=1,NEC
```

```
         READ(70,*)DFCONE(I,J,K)
   75 CONTINUE
C
C  For the dome
C
      READ(25,*)DOMTHC
      DO 81 K=1,NBWLIN
        READ(25,*)DOABCO(K)
        write(6,*)doabco(k)
   81 CONTINUE
      DO 80 I=1,NBDOME
      DO 80 K=1,NBWLIN
      DO 80 J=1,NEC
        READ(23,*)DFDOME(I,J,K)
   80 CONTINUE
C
C  For the ring
C
      DO 82 I=1,NBRING
      DO 82 K=1,NBWLIN
      DO 82 J=1,NEC
        READ(24,*)DFRING(I,J,K)
   82 CONTINUE
C
C  For the cavsub
C
      DO 83 I=1,NBCAVSUB
      DO 83 K=1,NBWLIN
      DO 83 J=1,NEC
        READ(65,*)DFCSUB(I,J,K)
   83 CONTINUE
C
C  NOTE: DFACS for both the dome and ring are read for both
C        channels although the dome is not a part of the total
C        channel and the ring is not a part of the visible channel.
C        Values of 0.0D0 are read in for these DFACS for the
C        corresponding channels, and then considered in the
C        subroutine radiate.
C
C
C  For the electrical substitution heater to be utilized in the
C  analysis, read the appropriate data.
C
      READ(13,310) E2,RH,HWIRE
      IF (INIT.GT.0) READ(13,310) EO,TAU,ALPHA,DELT
C
C  Determine the highest element ring which is completely covered
C  by the heater wire, and calculate the cavity surface area which
C  is covered by the heater wire.
C
        HROW = -0.01D0
        WAREA = 0.0D0
        AREA = AREA1
        DO 90 I = 1,MDIV(1)+MDIV(2)
           HROW = HROW + DM(2*I) + DM(2*I - 1)
           IF (HWIRE.GE.HROW) HEATEL(I) = 1.0D0
           IF (HWIRE.LT.HROW) HEATEL(I) = 0.0D0
           IF (I.GT.MDIV(1)) AREA = AREA2
           WAREA = WAREA + HEATEL(I)*N1*AREA
   90   CONTINUE
        DO 100 I=1,MDIV(1)+MDIV(2)
        HEATEL(I) = HEATEL(I)/(WAREA*THCK)
  100   CONTINUE
C
```

```
C  If a steady-state analysis with radiation is desired, an iterative
C  solution will be necessary.  An initial guess at the solution is
C  determined by assigning the heat sink temperature to all nodes in
C  the mesh.
C
      IF (INIT.EQ.0) THEN
         DO 110 I = 1,NNM
           TNEW(I) = THS
  110    CONTINUE
      ENDIF
C
C  Determine the half bandwidth of the system by calculating the
C  maximum difference between two related nodes.
C
      NHBW = 0
      DO 120 N = 1,NEM
      DO 120 I = 1,9
      DO 120 J = 1,9
        NW = (IABS(NOD(N,I) - NOD(N,J)) + 1)

        IF (NHBW.LT.NW) NHBW = NW
  120 CONTINUE
***********************************************************************
*                                                                     *
*                         PROCESSOR UNIT                              *
*                                                                     *
*       Performs the following functions:                             *
*                                                                     *
*       1.   generates the element matrices and vectors               *
*       2.   assembles the element matrices and vectors into the      *
*            global form                                              *
*       3.   imposes the boundary conditions                          *
*       4.   solves the equations.                                    *
*                                                                     *
***********************************************************************
      DO 125 I=1,NEM
  125 GEN(I,1) = 0.0D0

C
C  Initialize the iteration counter, the number of time steps and
C  the time.
C
      TIME = 0.0D0
C
C  Run a DO-loop over the number of time steps while incrementing the
C  time on each loop.
C
      DO 250 NSTEP = 1,NDT
        NCOUNT = 0

        IF (INIT.EQ.0) THEN

          NS = 1

        ELSE

          TIME = TIME + DT
        END IF
C
C  Read the DFAC's for the scene
C
      IF (NSTEP.LE.120.OR.NSTEP.GT.240) THEN
         DO 60 I=1,NBSCENE
         DO 60 K=1,NBWLIN
         DO 60 J=1,NEC
```

```
          READ(21,*)DFSCN(I,J,K)
   60   CONTINUE
        REWIND 21
       ELSE
        DO 65 I=1,NBSCENE
        DO 65 K=1,NBWLIN
        DO 65 J=1,NEC
          READ(99,*)DFSCN(I,J,K)
   65   CONTINUE
      END IF
C
C  The iteration loop begins for a steady-state analysis with radiation.
C  The iteration counter is incremented on each loop.
C
  130 CONTINUE
      NCOUNT = NCOUNT + 1
C
C  Initialize the generation vector, global conductivity matrix
C  and flux vector to zero.
C
      DO 140 I = 1,NEM
  140 GEN(I,2) = 0.0D0
      DO 150 I = 1,NNM
      GTF(I) = 0.0D0
      DO 150 J = 1,NHBW
  150 GTA(I,J) = 0.0D0
C
C  Since thermal radiation is present in the model,
C  determine the average temperature of each element in the cavity
C  and the effective generation in each element of the cavity
C  due to thermal diffuse-specular radiation.
C
      CALL TEMPAV (NEC,NNM,NOD,X,Y,TNEW,TAVE)
      CALL RADIAT (NEC,N1,MDIV,NSTEP,NDTS,NDTSH,TAVE,GEN,EM)
C
C  Determine the generation in each element of the cavity due to
C  the electric heating wires wound around the cavity.
C
      CALL HEATER (INIT,N1,MDIV,TNEW,GEN,RH,E2,DT)
C
C  For the first time step of a transient analysis, set the volumetric
C  heat generation in each element from the previous time step equal to
C  that of the present time step.
C
      IF ((INIT.GT.0) .AND. (NSTEP.EQ.1)) THEN
         DO 160 I = 1,NEM
         GEN(I,1) = GEN(I,2)
  160    CONTINUE
      ENDIF
C
C  Run a DO-loop over the number of elements in the mesh.  The
C  matrices of each element are determined in this loop and added
C  to the global matrices.
C
      DO 190 N = 1,NEM
C
C  If a transient solution is desired, determine the temperatures
C  of each node in the element from the previous solution.  For the
C  first time step the initial conditions are used as the previous
C  solution.
C
      IF (INIT.GT.0) THEN
         DO 170 I = 1,9
  170    TOLD(I) = TNEW(NOD(N,I))
```

```
      ENDIF
C
C  Determine the element conductivity matrix and flux vector.
C
      CALL ELEM (N,INIT,X,Y,TOLD,GEN,TA,TF,EM)
C
C  Assemble the element matrices into the global matrices in
C  banded symmetric form.
C
      DO 180 I = 1,9
      NR = NOD(N,I)
C
C  Add the element flux vector to the global flux vector.
C
      GTF(NR) = GTF(NR) + TF(I)
      DO 180 J = 1,9
      NC = NOD(N,J) - NOD(N,I) + 1
C
C  Add the element conductivity matrix to the global conductivity
C  matrix.
C
      IF(NC.GT.0) GTA(NR,NC) = GTA(NR,NC) + TA(I,J)
  180 CONTINUE
  190 CONTINUE
C
C
C  Impose the specified temperature boundary conditions on the
C  global matrices.
C
      CALL BNDY (NRMAX,NCMAX,NNM,NHBW,GTA,GTF,NPDF,IPDF,VPDF)

C
C  Solve the equations of the banded symmetric system.  The solution
C  is returned by subroutine SOLVE as the flux vector {GTF}.
C
      WRITE(6,*) 'SOLVING...'
      CALL SOLVE (NRMAX,NCMAX,NNM,NHBW,GTA,GTF)
c     DO 1 I=1,NNM
c       WRITE(27,*)'I, GTF(I)',I,GTF(I)
c   1 CONTINUE
C
C  If a steady-state solution is desired and thermal radiation
C  exists in the model, then determine if the solution has
C  converged within the specified limit (ERRMAX or MCOUNT).
C
      IF (INIT.EQ.0) THEN
         ERR = 0.0D0
         DNORM = 0.0D0
         DO 210 I = 1,NNM
         DNORM = DNORM + GTF(I)**2
  210    ERR = ERR + (GTF(I) - TNEW(I))**2
         ERROR = DSQRT(ERR/DNORM)
      ENDIF
C
C  Assign the new temperature values.
C
      DO 220 I = 1,NNM
  220 TNEW(I) = GTF(I)
      IF (INIT.EQ.0) THEN
         WRITE(6,325) NCOUNT,ERROR
         IF((ERROR.GT.ERRMAX).AND.(NCOUNT.LT.MCOUNT)) GO TO 130

      ENDIF
C
C  Calculate the average temperature of  of the resistance thermometer.
```

```
C
      N = 2*N1
      TARE = 0.0D0
      DO 230 I = 1,N
      TARE = TARE + TNEW(NRES+I-1)
  230 CONTINUE
      TARES = TARE/N
      write(6,*)'n',n
C
C  Determine the heat transferred to the heat sink by conduction,
C  and the heat emitted out through the aperture.
C
      TCOND = 0.0D0
      DO 235 I = NNM-4*N1+1,NNM-2*N1
 235     TCOND = TNEW(I) + TCOND
      TCOND = TCOND/(2*N1)
      ACOND = 2.0D0*PI*RO*THCK
      K = (MDIV(1)+MDIV(2)+MDIV(3)+MDIV(4))*2
      QCOND = COND*ACOND*(TCOND-THS)/DM(K)
C
C  Often it is desirable to obtain a specified temperature drop between
C  the resistance thermometers and the heat sink in a steady-state
C  analysis based upon the radiative and/or the electrical input.
C  Write the radiative input, resistor temperature and electrical
C  input values, then,
C
      IF (INIT.EQ.0) THEN
C
C  Since the heater is utilized in the analysis and radiation is considered

C  ask for a new heater voltage.  A negative number terminates program
C  execution.
C
        WRITE(6,*)'NEW E2? A negative number will stop execution.'
        READ(6,*) EE2
        IF (EE2.GT.0.0D0) THEN
        E2  = EE2
        GOTO 130
        ENDIF
      ENDIF
C
C  If the analysis is steady-state, write the temperature of the
C  resistance thermometer, and the radiative and electrical power input.
C
      PE2 = (1000.0D0/RH)*E2**2
      IF (INIT.EQ.0) WRITE(24,330) TARES,FMAG(1),SM,PE2
C
C  If the analysis is transient, write the average temperature of the
C  resistance thermometers, the heater power and the radiative input
C  power for each time step to the terminal screen and a disk file.
C
      IF (INIT.GT.0) THEN
C
         WRITE(26,330) TIME,TARES,E2,PE2,SCNFLUX,PE2+SCNFLUX
      ENDIF
C
C  Write the temperature distribution for the current time step to a
C  disk file.
C
CCCC  WRITE(27,*) 'TIME = ',TIME,'SECONDS'
CCCC  WRITE(27,305) (TNEW(I),I=1,NNM)
C
C  Assign the heat generation from present time step to that of the
C  previous time step, and then, for a transient analysis, proceed to
C  the next time step.
```

```
C
      DO 240 I = 1,NEM
        GEN(I,1) = GEN(I,2)

  240 CONTINUE
  250 CONTINUE
      STOP
C
C                  * * * FORMAT STATEMENTS * * *
C
  300 FORMAT(16I5)
  305 FORMAT(6E12.5)
  310 FORMAT(6D12.5)
  320 FORMAT(15X,D14.5)
  325 FORMAT(1X,'ITERATION NO. = ',I2,3X,'%ERROR = ',F10.6)
  330 FORMAT(6(3X,F12.7))
  335 FORMAT(1X,'VMAG = ',F10.5,3X,'FMAG = ',F10.5,3X,'SMAG = ',F10.5,
     &3X,'TARES = ',F10.5,3X,'E2 = ',F8.5)
  340 FORMAT(20X,D14.5)
  345 FORMAT(1X,'Heat conducted to the heat sink = ',F13.5,' (mW)')
  350 FORMAT(1X,'Energy emitted from the cavity = ',F13.5,' (mW)',/)
      END

************************************************************************
*                                                                      *
*                        SUBROUTINE TEMPAV                             *
*                                                                      *
*          This subroutine calculates, by linear interpolation, the    *
*     average temperature of each element in the radiometer cavity     *
*                                                                      *
************************************************************************
      SUBROUTINE TEMPAV(NEC,NNM,NOD,X,Y,TNEW,TAVE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NOD(200,9)
      DIMENSION TNEW(800),TAVE(100),X(200,9),Y(200,9),TOLD(9)
C
C  run a do-loop over the number of elements.
C
      DO 30 I=1,NEC
C
C  Initialize the area and tempoerature of this element to zero
C
      AREA=0.0D0
      TEMP=0.0D0
C
C  Determine the element local node temperatures.
C
      NPE=9
      DO 10 J=1,9
      TOLD(J)=TNEW(NOD(I,J))
   10 CONTINUE
C
C  The area is calculated by subcividing the element into
C  8 triangles for quadrilateral elements or 4 triangles for
C  triangular elements.  The total area is then found by summing
C  the areas of these small triangles.  A do-loop is run over
C  the number of triangles in the element.
C
      L=8
      DO 20 J=1,L
C
C  Determine the three vertices of the triangular area
C
      PT1X=X(I,J)
      PT1Y=Y(I,J)
```

```
      PT2X=X(I,J+1)
      PT2Y=Y(I,J+1)
      PT3X=X(I,(8/L)+L)
      PT3Y=Y(I,(8/L)+L)
      IF (J.EQ.8) THEN
           PT2X=X(I,1)
           PT2Y=Y(I,1)
      END IF
C
C CALCULATE THE LENGTHS OF SIDES 1 AND 3 OF THE TRIANGLE
C
      SIDE1=DSQRT((PT1X-PT2X)**2+(PT1Y-PT2Y)**2)
      SIDE3=DSQRT((PT3X-PT1X)**2+(PT3Y-PT1Y)**2)
C
C TAKE THE DOT PRODUCT OF THE UNIT VECTORS ALONG SIDES 1 AND 3
C TO DETERMINE THE ANGLE BETWEEN THESE TWO SIDES.
C
      AI=(PT2X-PT1X)/SIDE1
      AJ=(PT2Y-PT1Y)/SIDE1
      BI=(PT3X-PT1X)/SIDE3
      BJ=(PT3Y-PT1Y)/SIDE3
      THETA=DACOS((AI*BI)+(AJ*BJ))
C
C DEFINTE THE BASE AS SIDE 1 AND THE HEIGHT AS THE LENGTH OF THE
C SIDE OPPOSITE THE ANGLE BETWEEN THESE TWO SIDES
C
      BASE=SIDE1
      HEIGHT=SIDE3*DSIN(THETA)
C
C CALCULATE THE AREA OF THE TRIANGLE, THEN ADD THIS TO THE
RUNNING
C SUM TO DETERMINE THE ELEMENTAL AREA
C
      TAREA=(0.5D0*BASE*HEIGHT)
      AREA=AREA+TAREA
C
C DETERMINE THE TEMPERATURES OF THE THREE VERTICES OF THE
TRIANGLE.
C
      TA=TOLD(J)
      TB=TOLD(J+1)
      TC=TOLD(L+8/L)
      IF(J.EQ.8) TB=TOLD(1)
C
C DETERMINE THE TEMPERATURE AT THE MIODPOINT OF EACH SIDE OF THE
C TRIANGLE
C
      TP=(TA+TB)/2.0D0
      TQ=(TB+TC)/2.0D0
      TR=(TC+TA)/2.0D0
C
C ALSO, THE LENGTHS OF THE THREE BISECTORS OF THE TRIANGLE, THE
C COORDINATES OF THE CENTROID, AND THE LENGTH FROM EACH VERTES
C TO THE CENTROID ARE DETERMINED.
C
      AQ =DSQRT((PT1X-(PT2X+PT3X)/2.0)**2+
     &          (PT1Y-(PT2Y+PT3Y)/2.0)**2)
      BR =DSQRT((PT2X-(PT1X+PT3X)/2.0)**2+
     &          (PT2Y-(PT1Y+PT3Y)/2.0)**2)
      CPP=DSQRT((PT3X-(PT1X+PT2X)/2.0)**2+
     &          (PT3Y-(PT1Y+PT2Y)/2.0)**2)
      YZ=HEIGHT/3.0D0
      SLOPE=(PT1Y-(PT2Y+PT3Y)/2.0D0)/(PT1X-(PT2X+PT3X)/2.0D0)
      YINT=PT1Y-SLOPE*PT1X
      XZ=(YZ-YINT)/SLOPE
```

```
      AZ=DSQRT((PT1X-XZ)**2+(PT1Y-YZ)**2)
      BZ=DSQRT((PT2X-XZ)**2+(PT2Y-YZ)**2)
      CZ=DSQRT((PT3X-XZ)**2+(PT3Y-YZ)**2)
C
C NOW, DETERMINE THE TEMPERATURE AT THE CENTROID OF THE
TRIANGLE,
C MULTIPLY THIS BY THE AREA OF THE TRIANGLE, AND ADD THIS VALUE
C TO THE RUNNING TOTAL.
C
      TZ1=(AZ/AQ)*TA+(1-AZ/AQ)*TQ
      TZ2=(BZ/BR)*TB+(1-BZ/BR)*TR
      TZ3=(CZ/CPP)*TC+(1-CZ/CPP)*TP
      TZ=(TZ1+TZ2+TZ3)/3.0D0
      TEMP=TEMP+TAREA*TZ
c     WRITE(27,*)'TEMP, TAREA, TZ',TEMP,TAREA,TZ
c     WRITE(27,*)'TA, TB, TZ',TA,TB,TZ
c     WRITE(27,*)'AZ, BZ,CZ',AZ,BZ,CZ
c     WRITE(27,*)'AQ, BR, CPP',AQ,BR,CPP
   20 CONTINUE
C
C DETERMINE THE ELEMENT TEMPERATURE
C
      TAVE(I)=TEMP/AREA
c     WRITE(27,*)'I, TAVE(I)',I,TAVE(I)
   30 CONTINUE
      RETURN
      END
*****************************************************************
*                                                               *
*                       SUBROUTINE RADIAT                       *
*                                                               *
*      This subroutine calculates the effective generation in  *
*      each cavity element due to thermal radiation.            *
*                                                               *
*****************************************************************

      SUBROUTINE RADIAT(NEC,N1,MDIV,NSTEP,NDTS,NDTSH,TAVE,GEN,EM)
C
C Request double precision and real variables, place variables used
C by the subroutine in common storage blocks and dimension the arrays.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION GEN(200,2),TAVE(200),IS(20),IT(20),MDIV(4)
      DIMENSION DFOPT(200,100,15),DFSCN(200,100,15)
      DIMENSION DFDOME(200,100,15),DFRING(200,100,15),DFCSUB(200,100,15)
      DIMENSION DFUSUB(200,100,15),DFLSUB(200,100,15)
      DIMENSION DFCONE(200,100,15),DFBAR(200,100,15)
      DIMENSION DOABCO(25),KAPPA(25)
      DIMENSION EMIPOW(600,20),genn(100)
      DIMENSION GENOPT(100),GENSCN(100),GENRING(100)
      DIMENSION GENEMI(100),GENDOME(100),GENCSUB(100)
      DIMENSION GENUSUB(100),GENLSUB(100),GENCON(100),GENBAR(100)
      DIMENSION GENNBAR(100),GENNCON(100)

      COMMON /RADI1/ AREA1,AREA2,PI,RI,THCK,DIM(4),NVEC,NSUN,NS
      COMMON /RADI3/ AFOVL,AUSUB,ALSUBI,ALSUBO,ARING,VOL1,VOL2,
     &               ACAVSUB,AFOVAP,ABAR,ACON
      COMMON /NB/ NBFOVL,NBUSUB,NBLSUB,NBCAVSUB,NBRING,
     &            NBDOME,NBSCENE,NBBAR,NBCON,NBCAV
      COMMON /EMISS/
EMFOVL,EMUSUB,EMLSUB,EMDOME,EMBAR,EMCON,EMCAVSUB,
     &               EMRING
      COMMON /DFACS/
DFOPT,DFSCN,DFBAR,DFCONE,DFDOME,DFRING,DFCSUB,
     &               DFUSUB,DFLSUB
```

```
      COMMON /CHAN/ CHANEL
      COMMON /DOME/ DOMTHC,DOABCO,NBWLIN
      COMMON /FLUX/ SCNFLUX
C
C Define the Stefan-Boltzman constant in the correct units.
C
      SIGMA=5.6696D-8
      N=MDIV(1)+MDIV(2)
C
C Determine the emissive power from each zone as a function of the
C wavelength interval
C
      CALL EMPOW(EMIPOW,TAVE,NBWLIN)
C
C  The effective thermal generation in each element of the cavity is
C determined from the total flux incident upon the element from all
C the other elements of the instrument including itself and the FOVL aperture,
C and then subtracting the energy emitted by the element.
C
C Loop over the number of elements in the cavity, initializing the
C total flux incident upon the element to zero.
C
      DO 10 J=1,NEC
        GEN(J,2)=0.0D0
   10 CONTINUE
C
C Since the cavity mesh is radially symmetric, the distribution factors
C are calculated for the first element for each level.  The distribution
C factors are calculated using programs provided by HAEFFELIN and PRIESTLEY.
C The distribution factor for each cavity element
C to all other cavity elements, including itself and the optical front end
C is determined from the calculated values.
C
C BARREL TO CAVITY
C Loop over the number of elements (I) in the cavity to determine the
C total flux incident upon cavity element J.
C
      SUM=0.0D0
      DO 19 I=1,100
      genbar(i)=0.0d0
   19 CONTINUE

      WRITE(17,*)'HELLO'
      DO 20 K=1,NBWLIN
      DO 20 I=1,NBBAR
      DO 20 J=1,NEC
        genbar(j)=genbar(j)+embar*abar*emipow(i+66,k)*dfbar(i,j,k)
   20 CONTINUE
      do 25 i=1,nec
        if (i.le.60) then
          area=acon*1.0D6
        else
          area=abar*1.0D6
        end if
        SUM=SUM+GENBAR(I)
        GENNBAR(I)=GENBAR(I)
        genBAR(i)=genBAR(i)/(area*thck)
   25 continue
C
C  Add the energy from the cone
C
      SUM=0.0D0
      DO 26 I=1,100
        GENCON(I)=0.0D0
   26 CONTINUE
```

```
      DO 27 K=1,NBWLIN
      DO 27 I=1,NBCON
      DO 27 J=1,NEC
        GENCON(J)=GENCON(J)+EMCON*ACON*EMIPOW(I+6,K)*DFCONE(I,J,K)
   27 CONTINUE
      DO 28 I=1,NEC
        IF (I.LE.60) THEN
          AREA=ACON*1.0D6
        ELSE
          AREA=ABAR*1.0D6
        END IF
        SUM=SUM+GENCON(I)
        GENNCON(I)=GENCON(I)
        GENCON(I)=GENCON(I)/(AREA*THCK)
   28 CONTINUE
C
C Subtract the amount of energy which each element
C in the cavity emits to surfaces other than the cavity.
C
      sum=0.0d0
      DO 29 I=1,100
      genemi(i)=0.0d0
   29 CONTINUE

      DO 30 I=1,NEC
      DO 30 K=1,NBWLIN
      IF (I.GE.1.AND.I.LE.NBCON) THEN
        EM=EMCON
        AREA=ACON
      ELSE
        EM=EMBAR
        AREA=ABAR
      END IF
      genemi(i)=genemi(i)+em*area*EMIPOW(I+6,K)
   30 CONTINUE
      do 35 i=1,nec
        if (i.le.60) then
          area=acon*1.0D6
        else
          area=abar*1.0D6
        end if
        sum=sum+genemi(i)
        genemi(i)=genemi(i)/(area*thck)
   35 continue
      write(17,*)'sum of surfac genemi',sum

C
C Add the energy incident upon element I from the dome for the visible
C channels, or for the ring for the total channel.
C
      sum=0.0d0
      DO 39 I=1,100
      gendome(i)=0.0d0
      genring(i)=0.0d0
   39 CONTINUE

      IF (CHANEL.EQ.2) THEN
        DO 40 K=1,NBWLIN
          KAPPA(K)=DLOG(1.0D0/DOABCO(K))/DOMTHC
        DO 40 I=1,NBDOME
        DO 40 J=1,NEC
          IF (I.GE.1.AND.I.LE.(NBDOME-N1)) THEN
            VOL=VOL1
          ELSE IF (I.GT.(NBDOME-N1).AND.I.LE.NBDOME) THEN
            VOL=VOL2
```

```
          END IF
            GEN(J,2)=GEN(J,2)+4.0D0*KAPPA(K)*VOL*EMIPOW(I+106,K)
     &                 *DFDOME(I,J,K)
            gendome(j)=gendome(j)+4.0d0*kappa(k)*vol*emipow(I+106,k)
     &                *dfdome(i,j,k)
      IF (KAPPA(K).LT.0.0D0) THEN
         WRITE(17,*)'K, KAPPA(K)',K,KAPPA(K)
      ELSE IF(EMIPOW(I+106,K).LT.0.0D0) THEN
         WRITE(17,*)'I+106, K, EMIPOW(I+106,K)',I+106,K,EMIPOW(I+106,K)
      ELSE IF(DFDOME(I,J,K).LT.0.0D0) THEN
         WRITE(17,*)'I, J, K, DFDOME(I,J,K)',I,J,K,DFDOME(I,J,K)
      ELSE IF(GENDOME(J).LT.0.0D0) THEN
         WRITE(17,*)'I, J, K, GENDOME(J)',I,J,K,GENDOME(J)
      END IF
   40   CONTINUE
      do 45 i=1,nec
        if (i.le.60) then
          area=acon*1.0D6
        else
          area=abar*1.0D6
        end if
        sum=sum+gendome(i)
        gendome(i)=gendome(i)/(area*thck)
   45 continue
      ELSE
        DO 50 K=1,NBWLIN
        DO 50 I=1,NBRING
        DO 50 J=1,NEC
              EM=EMRING
              AREA=ARING
            GEN(J,2)=GEN(J,2)+EM*AREA*EMIPOW(5,K)*DFRING(I,J,K)
            genring(j)=genring(j)+em*area*emipow(5,k)*dfring(i,j,k)
   50   CONTINUE
      do 55 i=1,nec
        if (i.le.60) then
          area=acon*1.0D6
        else
          area=abar*1.0D6
        end if
        sum=sum+genring(i)
        genring(i)=genring(i)/(area*thck)
   55 continue
      END IF

C
C Add the energy incident upon element I from the field of view limiter
C
      sum=0.0d0
      DO 59 I=1,100
      genopt(i)=0.0d0
   59 CONTINUE

      DO 60 K=1,NBWLIN
      DO 60 I=1,NBFOVL
      DO 60 J=1,NEC
          GENOPT(J)=GENOPT(J)+EMFOVL*AFOVL*EMIPOW(2,K)*DFOPT(I,J,K)
   60 CONTINUE
      do 65 i=1,nec
        if (i.le.60) then
          area=acon*1.0D6
        else
          area=abar*1.0D6
        end if
        sum=sum+genopt(i)
        genopt(i)=genopt(i)/(area*thck)
```

```
   65 continue
C
C Add the energy incident upon element I from the upper substrate
C
      sum=0.0d0
      DO 61 I=1,100
      genusub(i)=0.0d0
   61 CONTINUE
      DO 62 K=1,NBWLIN
      DO 62 I=1,NBUSUB
      DO 62 J=1,NEC
          GENUSUB(J)=GENUSUB(J)+EMUSUB*AUSUB*EMIPOW(3,K)*DFUSUB(I,J,K)
   62 CONTINUE
      do 63 i=1,nec
        if (i.le.60) then
          area=acon*1.0D6
        else
          area=abar*1.0D6
        end if
        sum=sum+genusub(i)
        genusub(i)=genusub(i)/(area*thck)
   63 continue
C
C Add the energy incident upon element I from the lower substrate
C
      sum=0.0d0
      DO 66 I=1,100
      genLSUB(i)=0.0d0
   66 CONTINUE

      DO 67 K=1,NBWLIN
      DO 67 I=1,NBLSUB
      DO 67 J=1,NEC
          IF (I.LE.10) THEN
            AREA=ALSUBI
          ELSE
            AREA=ALSUBO
          END IF
          GENLSUB(J)=GENLSUB(J)+EMLSUB*AREA*EMIPOW(4,K)*DFLSUB(I,J,K)
   67 CONTINUE
      do 68 i=1,nec
        if (i.lt.60) then
          area=acon*1.0D6
        else
          area=abar*1.0D6
        end if
        sum=sum+genlsub(i)
        genlsub(i)=genlsub(i)/(area*thck)
   68 continue

C
C Add the energy incident upon element I from the aperture.
C
      SCNFLUX=0.0d0
      sum=0.0d0
      DO 69 I=1,100
      genscn(i)=0.0d0
   69 CONTINUE
      I=1
      DO 70 K=1,NBWLIN
      DO 70 J=1,NEC
        genscn(j)=genscn(j)+afovap*emipow(1,k)*dfscn(i,j,k)
   70 CONTINUE
      do 75 i=1,nec
        if (i.lt.60) then
```

```
          area=acon*1.0D6
        else
          area=abar*1.0D6
        end if
        sum=sum+genscn(i)
        genscn(i)=genscn(i)/(area*thck)
   75 continue
        SCNFLUX=SUM*1000.0D0
C
C Add the energy incident upon element j from the cavsub
C
      SUM=0.0D0
      DO 79 I=1,100
        GENCSUB(I)=0.0D0
   79 CONTINUE
      DO 80 I=1,NBCAVSUB
      DO 80 K=1,NBWLIN
      DO 80 J=1,NEC

GENCSUB(J)=GENCSUB(J)+EMCAVSUB*ACAVSUB*EMIPOW(6,K)*DFCSUB(I,J,K)
   80 CONTINUE
      DO 85 I=1,NEC
        IF (I.LE.60) THEN
          AREA=ACON*1.0D6
        ELSE
          AREA=ABAR*1.0D6
        END IF
        SUM=SUM+GENCSUB(I)
        GENCSUB(I)=GENCSUB(I)/(AREA*THCK)
   85 CONTINUE
C
C Determine the effective volumetric generation in element I.
C
      DO 90 J=1,NEC
        GEN(J,2)=1000.0D0*(GENSCN(J)+GENOPT(J)+GENDOME(J)+GENBAR(J)
     &             +GENRING(J)-GENEMI(J)+GENCSUB(J)+GENUSUB(J)
     &             +GENLSUB(J)+GENCON(J))
   90 CONTINUE
C
C Return to the main program
C
      RETURN
      END
************************************************************************
*                                                                      *
*                         SUBROUTINE HEATER                            *
*                                                                      *
*     This subroutine calculates the voltage across the electrical     *
*     substitution heater wire, and then determines the                *
*     volumetric heat generation in each cavity element due to this    *
*     voltage.  The subroutine is called by the main program.          *
*                                                                      *
************************************************************************
      SUBROUTINE HEATER(INIT,N1,MDIV,TNEW,GEN,RH,E2,DT)
C
C Request double precision real variables and dimension the arrays.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION MDIV(4),TNEW(800),GEN(200,2)
      COMMON /HEAT1/ HEATEL(15),EO,ALPHA,TAU,THS,DELT,NRES
C
C If the analysis is transient, determine the average temperature of
C the two resistance thermometers,
C
      IF (INIT.GT.0) THEN
```

```
         N=2*N1
         TARE=0.0D0
         DO 10 I=1,N
   10    TARE=TARE+TNEW(NRES+I-1)
         TARES=TARE/N
C
C and the corresponding voltage across the heater wire.
C
         E2=E2+(DT*ALPHA*EO/(4.0D0*TAU))*(DELT+(THS-TARES))
      END IF
C
C Determine the total power output of the heater and write
C it to a file.
C
      Q=1000.0D0*(E2**2)/RH
      WRITE(28,*)'E2, Q',E2,Q
C
C Calculate the volumetric heat generation in each cavity element due
C to the heater, and add this to the volumetric heat generation due to
C thermal radiation.
C
      DO 70 N=1,N1*(MDIV(1)+MDIV(2))
      I=1+INT((N-1)/N1)
      GEN(N,2)=GEN(N,2)+Q*HEATEL(I)
   70 CONTINUE
C
C Return to the main program.
C
      RETURN
      END

**********************************************************************
*                                                                    *
*                      SUBROUTINE ELEM                               *
*                                                                    *
*     This subroutine generates the element matrices for quadratic   *
*     quadrilateral elements for either a steady-state or transient  *
*     analysis using the two-dimensional heat conduction equation.   *
*     This subroutine is called by the main program and by subroutine *
*     INTER.                                                         *
*                                                                    *
**********************************************************************

      SUBROUTINE ELEM(N,INIT,X,Y,TOLD,GEN,TA,TF,EM)
C
C Request double precision real variables, dimension the arrays, and
C provide Gauss quadratic data.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION GAUSS(3,2),WEIGH(3,2),A(9,9),B(9,9),F(9,2),C(9,9),D(9)
      DIMENSION SF(9),DSF(9,2),TF(9),TA(9,9),GINV(2,2)
      DIMENSION TOLD(9),GEN(200,2),X(200,9),Y(200,9)
      COMMON /ELEM1/ DT,THETA,COND,CP,DENS
      DATA GAUSS/-0.77459667D0,0.00000000D0,0.77459667D0,
     &            0.50000000D0,0.50000000D0,0.00000000D0/
      DATA WEIGH/0.55555555D0,0.88888888D0,0.55555555D0,
     &           0.33333333D0,0.33333333D0,0.33333333D0/
C
C Determine the number of Gauss points in the element.
C
      NQP1=3
      NQP2=3
C
C Initialize the element submatrices to zero.
C
```

```
      DO 10 I=1,9
      F(I,1)=0.0D0
      F(I,2)=0.0D0
      D(I)=0.0D0
      DO 10 J=1,9
      A(I,J)=0.0D0
      B(I,J)=0.0D0
   10 CONTINUE
C
C Begin the gauss-quadrature to determine the element submatrices
C by running DO-loops over the number of gauss points.
C
      DO 30 LL=1,NQP1
      DO 30 KK=1,NQP2
C
C If the element is quadrilateral, determine the coordinates of the
C Gauss point, then determine the values of the interpolation functions
C inverse jacobian matrix and the jacobian.
C
      XI=GAUSS(LL,1)
      ETA=GAUSS(KK,1)
      CALL INTER (N,XI,ETA,X,Y,SF,DSF,GINV,DET)
      CONST=DET*WEIGH(LL,1)*WEIGH(KK,1)
C
C
C Evaluate the element submatrices for the gauss points using the
C equations given in the analysis.  DO-loops are run over the
C number of nodes in the element.
C
      DO 20 I=1,9
      F(I,1)=F(I,1)+CONST*GEN(N,1)*SF(I)
      F(I,2)=F(I,2)+CONST*GEN(N,2)*SF(I)
      DO 20 J=1,9
      IF (INIT.GT.0) A(I,J)=A(I,J)+CONST*SF(I)*SF(J)*CP*DENS
      B1=(GINV(1,1)*DSF(I,1)+GINV(1,2)*DSF(I,2))
      B2=(GINV(1,1)*DSF(J,1)+GINV(1,2)*DSF(J,2))
      B3=(GINV(2,1)*DSF(I,1)+GINV(2,2)*DSF(I,2))
      B4=(GINV(2,1)*DSF(J,1)+GINV(2,2)*DSF(J,2))
      B(I,J)=B(I,J)+CONST*COND*((B1*B2)+(B3*B4))
   20 CONTINUE
   30 CONTINUE
C
C If the analysis is steady state, the element matrices are the
C same as the corresponding submatrices calculated above.  If the
C analysis is transient, the element conductivity matrix can be
C found directly from the element submatrices, but the element
C flux vector must be determined after an additional submatrix is
C found.
C
      DO 40 I=1,9
      IF (INIT.EQ.0) TF(I)=F(I,2)
      DO 40 J=1,9
      IF (INIT.EQ.0) THEN
         TA(I,J)=B(I,J)
      ELSE
         TA(I,J)=A(I,J)+THETA*DT*B(I,J)
         C(I,J)=A(I,J)-(1-THETA)*DT*B(I,J)
      END IF
   40 CONTINUE
C
C If the analysis is steady-state, return to the main program
C
      IF (INIT.EQ.0) RETURN
C
C If the analysis is transient, determine the final submatrix
```

```
C needed to determine the element flux vector.
C
      DO 50 I=1,9
      DO 50 J=1,9
      D(I)=D(I)+C(I,J)*TOLD(J)
   50 CONTINUE
C
C The flux vector can now be found for the transient analysis.
C
      DO 60 I=1,9
      TF(I)=D(I)+THETA*DT*F(I,2)+(1.0D0-THETA)*DT*F(I,1)
   60 CONTINUE
      RETURN
      END

*************************************************************************
*                                                                       *
*                       SUBROUTINE INTER                                *
*                                                                       *
*     This subroutine determines the interpolation functions            *
*     and their derivatives, the jacobian and the inverse of            *
*     the jacobian matrix for the gauss points of a quadratic           *
*     quadrilateral element.  This subroutine is called by              *
*     subroutine ELEM.                                                  *
*                                                                       *
*************************************************************************

      SUBROUTINE INTER(N,XI,ETA,X,Y,SF,DSF,GINV,DET)
C
C Request double precision real variables, and dimension the arrays.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SF(9),DSF(9,2),TF(9),TA(9,9),GINV(2,2),GJ(2,2)
      DIMENSION X(200,9),Y(200,9)
C
C The interpolation functions and their derivatives are calculated at
C the Gauss point for a quadrilateral element.
C
         SF(1)=0.25D0*ETA*XI*(1.0D0-XI)*(1.0D0-ETA)
         SF(2)=-0.5D0*ETA*(1.0D0-XI**2)*(1.0D0-ETA)
         SF(3)=-0.25D0*ETA*XI*(1.0D0+XI)*(1.0D0-ETA)
         SF(8)=-0.5D0*XI*(1.0D0-ETA**2)*(1.0D0-XI)
         SF(9)=(1.0D0-XI**2)*(1.0D0-ETA**2)
         SF(4)=0.5D0*XI*(1.0D0-ETA**2)*(1.0D0+XI)
         SF(7)=-0.25D0*ETA*XI*(1.0D0-XI)*(1.0D0+ETA)
         SF(6)=0.5D0*ETA*(1.0D0-XI**2)*(1.0D0+ETA)
         SF(5)=0.25D0*ETA*XI*(1.0D0+XI)*(1.0D0+ETA)
         DSF(1,1)=0.25D0*ETA*(1.0D0-ETA)*(1.0D0-2.0D0*XI)
         DSF(1,2)=0.25D0*XI*(1.0D0-XI)*(1.0D0-2.D0*ETA)
         DSF(2,1)=ETA*XI*(1.0D0-ETA)
         DSF(2,2)=-0.5D0*(1.0D0-XI**2)*(1.0D0-2.0D0*ETA)
         DSF(3,1)=-0.25D0*ETA*(1.0D0-ETA)*(1.0D0+2.0D0*XI)
         DSF(3,2)=-0.25D0*XI*(1.0D0+XI)*(1.0D0-2.0D0*ETA)
         DSF(4,1)=0.5D0*(1.0D0-ETA**2)*(1.0D0+2.0D0*XI)
         DSF(4,2)=-ETA*XI*(1.0D0+XI)
         DSF(5,1)=0.25D0*ETA*(1.0D0+ETA)*(1.0D0+2.0D0*XI)
         DSF(5,2)=0.25D0*XI*(1.0D0+XI)*(1.0D0+2.0D0*ETA)
         DSF(6,1)=-ETA*XI*(1.0D0+ETA)
         DSF(6,2)=0.5D0*(1.0D0-XI**2)*(1.0D0+2.0D0*ETA)
         DSF(7,1)=-0.25D0*ETA*(1.0D0+ETA)*(1.0D0-2.0D0*XI)
         DSF(7,2)=-0.25D0*XI*(1.0D0-XI)*(1.0D0+2.0D0*ETA)
         DSF(8,1)=-0.5D0*(1.0D0-ETA**2)*(1.0D0-2.0D0*XI)
         DSF(8,2)=ETA*XI*(1.0D0-XI)
         DSF(9,1)=-2.0D0*XI*(1.0D0-ETA**2)
         DSF(9,2)=-2.0D0*ETA*(1.0D0-XI**2)
```

```
C
C Initialize the jacobian matrix
C
      DO 10 II=1,2
      DO 10 JJ=1,2
      GJ(II,JJ)=0.0D0
   10 CONTINUE
C
C Determine the jacobian matrix and its determinant (the jacobian)
C
      DO 20 K=1,9
      GJ(1,1)=GJ(1,1)+DSF(K,1)*X(N,K)
      GJ(1,2)=GJ(1,2)+DSF(K,1)*Y(N,K)
      GJ(2,1)=GJ(2,1)+DSF(K,2)*X(N,K)
      GJ(2,2)=GJ(2,2)+DSF(K,2)*Y(N,K)
   20 CONTINUE
      DET=GJ(1,1)*GJ(2,2)-GJ(2,1)*GJ(1,2)
C
C Determine the inverse of the jacobian matrix
C
      GINV(1,1)=GJ(2,2)/DET
      GINV(1,2)=-GJ(1,2)/DET
      GINV(2,1)=-GJ(2,1)/DET
      GINV(2,2)=GJ(1,1)/DET
      RETURN
      END

************************************************************************
*                                                                      *
*                        SUBROUTINE BNDY                               *
*                                                                      *
*     This subroutine imposes the temperature boundary conditions      *
*     on the banded symmetric system of equations.                     *
*     Provided by J.N. Reddy                                           *
*                                                                      *
************************************************************************

      SUBROUTINE BNDY(NRMAX,NCMAX,NNM,NHBW,GTA,GTF,NPDF,IPDF,VPDF)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION GTA(NRMAX,NCMAX),GTF(NRMAX)
      DIMENSION IPDF(NPDF),VPDF(NPDF)
C
C Run a DO-loop over the number of specified temperatur
C boundary conditions.
C
      DO 30 NB=1,NPDF
      IE=IPDF(NB)
      SVAL=VPDF(NB)
      IT=NHBW-1
      I=IE-NHBW
      DO 10 II=1,IT
      I=I+1
      IF (I.GE.1) THEN
         J=IE-I+1
         GTF(I)=GTF(I)-GTA(I,J)*SVAL
         GTA(I,J)=0.0D0
      ENDIF
   10 CONTINUE
      GTA(IE,1)=1.0D0
      GTF(IE)=SVAL
      I=IE
      DO 20 II=2,NHBW
      I=I+1
      IF (I.LE.NNM) THEN
         GTF(I)=GTF(I)-GTA(IE,II)*SVAL
```

```
         GTA(IE,II)=0.0D0
      ENDIF
   20 CONTINUE
   30 CONTINUE
      RETURN
      END
***********************************************************************
*                                                                     *
*                         SUBROUTINE SOLVE                            *
*                                                                     *
*     Solves a banded symmetric system of equations.                  *
*     Provided by J.N. Reddy                                          *
*                                                                     *
***********************************************************************
      SUBROUTINE SOLVE(NRM,NCM,NEQNS,NBW,BAND,RHS)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BAND(NRM,NCM),RHS(NRM)
      MEQNS=NEQNS-1
      DO 30 NPIV=1,MEQNS
      NPIVOT=NPIV+1
      LSTSUB=NPIV+NBW-1
      IF(LSTSUB.GT.NEQNS) LSTSUB=NEQNS
      DO 20 NROW = NPIVOT,LSTSUB
C
C Invert rows and columns for row factor
C
      NCOL=NROW-NPIV+1
      FACTOR=BAND(NPIV,NCOL)/BAND(NPIV,1)
      DO 10 NCOL=NROW,LSTSUB
      ICOL=NCOL-NROW+1
      JCOL=NCOL-NPIV+1
      BAND(NROW,ICOL)=BAND(NROW,ICOL)-FACTOR*BAND(NPIV,JCOL)
   10 CONTINUE
      RHS(NROW)=RHS(NROW)-FACTOR*RHS(NPIV)
   20 CONTINUE
   30 CONTINUE
C
C Back substitution.
C
      DO 70 IJK=2,NEQNS
      NPIV=NEQNS-IJK+2
      RHS(NPIV)=RHS(NPIV)/BAND(NPIV,1)
      LSTSUB=NPIV-NBW+1
      IF (LSTSUB.LT.1) LSTSUB=1
      NPIVOT=NPIV-1
      DO 60 JKI=LSTSUB,NPIVOT
      NROW=NPIVOT-JKI+LSTSUB
      NCOL=NPIV-NROW+1
      FACTOR=BAND(NROW,NCOL)
   60 RHS(NROW)=RHS(NROW)-FACTOR*RHS(NPIV)
   70 CONTINUE
      RHS(1)=RHS(1)/BAND(1,1)
      RETURN
      END
***********************************************************************
*                                                                     *
*                         SUBROUTINE EMMIT                            *
*                                                                     *
***********************************************************************

      SUBROUTINE EMMIT(NEC,N1,MDIV,RI,EM,TAVE,DFC,DIM,HO,ATIP,SUM2)
C
C Request double precision real variables, place variables used
C by the subroutine in common storage blocks and dimension the arrays.
C
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION tAVE(200),DFC(15,151)
      DIMENSION IS(20),IT(20),MDIV(4),DIM(4)
C
C Define the Stefan-Boltzman constant in the correct units ane the
C constant pi.
      SIGMA=5.6696D-8
      PI=3.141592654D0
        IF (MDIV(1).EQ.0) THEN
        AREA1=0.0D0
        GO TO 47
        END IF
        AREA1=(PI*RI*DSQRT(RI**2+(DIM(1)+HO)**2)-ATIP)/(N1*MDIV(1))
47      IF (MDIV(2).EQ.0) THEN
        AREA2=0.0D0
        GO TO 48
        END IF
        AREA2=(PI*RI*2.0D0*DIM(2))/(N1*MDIV(2))
C
C Determine the number of element levels in the cavity, and
C initialize the distribution factor subscript.
C
   48 N=MDIV(1)+MDIV(2)
      LL=0
      DO 10 I=1,N1
   10 IS(I)=NEC-I+1
      NECC=NEC+1
      SUM2=0.0D0
      LL=LL+1
      IF(LL.GT.N1) LL=1
C
C then determine the distribution factor subscripts for the element
C from those of the previously considered element.
C
      DO 20 L=1,N1
      IF (L.EQ.1) IT(L)=IS(N1)
   20 IF (L.NE.1) IT(L)=IS(L-1)
      DO 30 L=1,N1
   30 IS(L)=IT(L)
C
C Loop over the number of elements in the cavity to determine the
C total flux incident upon element I from the radiometer cavity.
C
      I1=0
      DO 40 J=1,N
      DO 40 K=1,N1
C
C Determine the subscript of the distribution factor corresponding
C to element I1.
C
      jj=is(k)
      if(j.ne.1) jj=jj-n1*(j-1)
      IF(LL.EQ.N1) JJ=NEC-I1
      I1=I1+1
C
C Determine energy escaped from the cavity.
C
      AREA=AREA1
      IF (J.GT.MDIV(1)) AREA=AREA2
      JL=N-J+1
      SUM2=SUM2+EM*SIGMA*AREA*(TAVE(I1)**4)*DFC(JL,NECC)
40    CONTINUE
      RETURN
      END
```

```
***********************************************************************
*
*  The following subroutine computes the emissive power of all the
*  elements of the optical front end either from known surface or
*  volum temperatures or from a given power value.
*
***********************************************************************
      SUBROUTINE EMPOW(EMIPOW,TAVE,NBWLIN)

      IMPLICIT NONE
C
      REAL*8 LBDA,DLBDA,INT1,INT2,X1,X2,X,W,EMIPOW
      REAL*8 TPELMT,SWRADFLU,LWRADFLU
      REAL*8 C1,C2,SIGMA,SWTSCN,LWTSCN
      REAL*8 PI
      REAL*8 DOMTHC,DOABCO
      REAL*8 TFOVL,TUSUB,TLSUB,TDOME,TRING,TCAVSUB,TAVE
C
      INTEGER IS,NMAX,INDX,MAXWLI,MAXELT
      INTEGER ELT,ELT1,ELT2,WVL,I,II
      INTEGER NBWLIN,K
C
      PARAMETER (IS=1000,MAXWLI=20,MAXELT=600)
C
      DIMENSION X(IS),W(IS),EMIPOW(MAXELT,MAXWLI)
      DIMENSION LBDA(20),TAVE(100),TPELMT(200)
      DIMENSION TDOME(60)
C
c     COMMON /DOME/DOMTHC,DOABCO,NBWLIN
      COMMON /LMBDA/LBDA
      OPEN(98,FILE='surfac.tmp')
C
C Read the distribution of wavelength intervals and calculate dlbda
C
      NMAX=20
      PI=DACOS(-1.0D0)
      C1=0.59544D8
      C2=14388.0D0
      SIGMA=5.6696D-8
      X1=0.D0
C
C Read the radiative flux incident to the instrument, as well as
C the temperatures of the zones, including the scene temp.
C
      READ(03,*)SWRADFLU,LWRADFLU
      WRITE(6,*)SWRADFLU+LWRADFLU
      READ(36,*)SWTSCN,LWTSCN
      READ(29,*)TFOVL
      READ(30,*)TUSUB
      READ(31,*)TLSUB
      READ(32,*)
      DO 10 I=1,60
        READ(32,*)II,TDOME(I)
c       READ(32,*)TDOME(I)
C       TDOME(I)=0.0D0
   10 CONTINUE
      REWIND 32
      READ(33,*)TRING
      READ(34,*)TCAVSUB
C
      TPELMT(1)=SWTSCN
      TPELMT(2)=TFOVL
      TPELMT(3)=TUSUB
```

```
      TPELMT(4)=TLSUB
      TPELMT(5)=TRING
      TPELMT(6)=TCAVSUB
      DO 20 I=1,100
        TPELMT(I+6)=TAVE(I)
   20 CONTINUE
      DO 30 I=1,60
        TPELMT(I+106)=TDOME(I)
   30 CONTINUE
C
C Compute the spectral emissive power of each element.
C
      DO 100 ELT=2,166

      DO 101 WVL=1,NBWLIN-1
C
C Compute the integral from 0 to lbda(k)
C
      INT1 = 0.0D0
      IF (LBDA(WVL).LE.0.D0) GO TO 99
      X2 = LBDA(WVL)*TPELMT(ELT)
      CALL GAULEG(IS,X1,X2,X,W,NMAX)
      DO 110 INDX=1,NMAX
          INT1=INT1 + 2.D0*PI*C1*DEXP(-C2/X(INDX))/(SIGMA*(X(INDX)**5)*
     &             (1.D0-DEXP(-C2/X(INDX))))*W(INDX)
110   CONTINUE
C
C Compute the integral from 0 to lbda(k+1)
C
99    INT2 = 0.0D0
      X2=LBDA(WVL+1)*TPELMT(ELT)
      CALL GAULEG(IS,X1,X2,X,W,NMAX)
      DO 120 INDX=1,NMAX
          INT2=INT2 + 2.D0*PI*C1*DEXP(-C2/X(INDX))/(SIGMA*(X(INDX)**5)*
     &            (1.D0-DEXP(-C2/X(INDX))))*W(INDX)
120   CONTINUE
C
C Compute the emissive power in the wl interval lbda(k)-lbda(k+1)
C
      EMIPOW(ELT,WVL)=(INT2-INT1)*SIGMA*TPELMT(ELT)*TPELMT(ELT)*
     &      TPELMT(ELT)*TPELMT(ELT)
101   CONTINUE
      WVL = NBWLIN
      INT1 = 0.0D0
      INT2 = 0.0D0
      X2 = LBDA(WVL)*TPELMT(ELT)
      CALL GAULEG(IS,X1,X2,X,W,NMAX)
      DO 130 INDX=1,NMAX
          INT2=INT2 + 2.D0*PI*C1*DEXP(-C2/X(INDX))/(SIGMA*(X(INDX)**5)*
     &             (1.D0-DEXP(-C2/X(INDX))))*W(INDX)
130   CONTINUE
      EMIPOW(ELT,WVL)=(1.D0 - INT2)*SIGMA*TPELMT(ELT)*TPELMT(ELT)*
     &      TPELMT(ELT)*TPELMT(ELT)
100   CONTINUE
C
C Emissive power from the Scene
C
      ELT=1
C
C Shortwave power:
C
      DO 200 WVL=1,NBWLIN-1
C
C Compute the integral from 0 to lbda(k)
C
```

```
      INT1 = 0.0D0
      IF (LBDA(WVL).LE.0.D0) GO TO 98
      X2 = LBDA(WVL)*SWTSCN
      CALL GAULEG(IS,X1,X2,X,W,NMAX)
      DO 210 INDX=1,NMAX
          INT1=INT1 + 2.D0*PI*C1*DEXP(-C2/X(INDX))/(SIGMA*(X(INDX)**5)*
     &            (1.D0-DEXP(-C2/X(INDX))))*W(INDX)
210   CONTINUE
C
C Compute the integral from 0 to lbda(k+1)

C
98    INT2 = 0.0D0
      X2=LBDA(WVL+1)*SWTSCN
      CALL GAULEG(IS,X1,X2,X,W,NMAX)
      DO 220 INDX=1,NMAX
          INT2=INT2 + 2.D0*PI*C1*DEXP(-C2/X(INDX))/(SIGMA*(X(INDX)**5)*
     &           (1.D0-DEXP(-C2/X(INDX))))*W(INDX)
220   CONTINUE
C
C Compute the emissive power in the wl interval lbda(k)-lbda(k+1)
C
      EMIPOW(ELT,WVL)=(INT2-INT1)*SWRADFLU
200   CONTINUE
      WVL = NBWLIN
      INT1 = 0.0D0
      INT2 = 0.0D0
      X2 = LBDA(WVL)*SWTSCN
      CALL GAULEG(IS,X1,X2,X,W,NMAX)
      DO 230 INDX=1,NMAX
          INT2=INT2 + 2.D0*PI*C1*DEXP(-C2/X(INDX))/(SIGMA*(X(INDX)**5)*
     &            (1.D0-DEXP(-C2/X(INDX))))*W(INDX)
230   CONTINUE
      EMIPOW(ELT,WVL)=(1.D0 - INT2)*SWRADFLU
C
C Longwave Power (added to the shortwave power)
C
      DO 250 WVL=1,NBWLIN-1
C
C Compute the integral from 0 to lbda(k)
C
      INT1 = 0.0D0
      IF (LBDA(WVL).LE.0.D0) GO TO 97
      X2 = LBDA(WVL)*LWTSCN
      CALL GAULEG(IS,X1,X2,X,W,NMAX)
      DO 260 INDX=1,NMAX
          INT1=INT1 + 2.D0*PI*C1*DEXP(-C2/X(INDX))/(SIGMA*(X(INDX)**5)*
     &            (1.D0-DEXP(-C2/X(INDX))))*W(INDX)
260   CONTINUE
C
C Compute the integral from 0 to lbda(k+1)
C
97    INT2 = 0.0D0
      X2=LBDA(WVL+1)*LWTSCN
      CALL GAULEG(IS,X1,X2,X,W,NMAX)
      DO 270 INDX=1,NMAX
          INT2=INT2 + 2.D0*PI*C1*DEXP(-C2/X(INDX))/(SIGMA*(X(INDX)**5)*
     &           (1.D0-DEXP(-C2/X(INDX))))*W(INDX)
270   CONTINUE
C
C Compute the emissive power in the wl interval lbda(k)-lbda(k+1)
C
      EMIPOW(ELT,WVL) = EMIPOW(ELT,WVL) + (INT2-INT1)*LWRADFLU
250   CONTINUE
      WVL = NBWLIN
```

```
      INT1 = 0.0D0
      INT2 = 0.0D0
      X2 = LBDA(WVL)*LWTSCN
      CALL GAULEG(IS,X1,X2,X,W,NMAX)
      DO 280 INDX=1,NMAX
          INT2=INT2 + 2.D0*PI*C1*DEXP(-C2/X(INDX))/(SIGMA*(X(INDX)**5)*
     &           (1.D0-DEXP(-C2/X(INDX))))*W(INDX)
280   CONTINUE
      EMIPOW(ELT,WVL) = EMIPOW(ELT,WVL) + (1.D0 - INT2)*LWRADFLU
C
      RETURN
      END
************************************************************************
*
*   THE FOLLOWING SUBROUTINE PERFORMS A GAUSS-LEGENDRE
*   QUADRATURE TO
*   EXECUTE A NUMERICAL INTEGRATION.  THE X'S ARE THE ROOTS OF THE
*   LEGENDRE POLYNOMIAL AND THE W'S ARE THEIR CORRESPONDING
*   WEIGHTS.
*   (COURTESY OF DR. J.R. THOMAS, ME DEPARTMENT AT VIRGINIA TECH)
*
************************************************************************

      SUBROUTINE GAULEG(IS,X1,X2,X,W,N)

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(IS),W(IS)
      PARAMETER (EPS=1.D-15)

      PI=DACOS(-1.D0)
      M=(N+1)/2
      XM=0.5D0*(X2+X1)
      XL=0.5D0*(X2-X1)
      DO 12 I=1,M
        Z=DCOS(PI*(DBLE(I)-.25D0)/(DBLE(N)+.5D0))
1       CONTINUE
          P1=1.D0
          P2=0.D0
          DO 11 J=1,N
            DJ=DBLE(J)
            P3=P2
            P2=P1

            P1=((DBLE(2.D0*J)-1.D0)*Z*P2-(DJ-1.D0)*P3)/DJ

11        CONTINUE
          PP=DBLE(N)*(Z*P1-P2)/(Z*Z-1.D0)
          Z1=Z
          Z=Z1-P1/PP
        IF(ABS(Z-Z1).GT.EPS)GO TO 1
        X(I)=XM-XL*Z
        X(N+1-I)=XM+XL*Z
        W(I)=2.D0*XL/((1.D0-Z*Z)*PP*PP)
        W(N+1-I)=W(I)
12    CONTINUE
      RETURN
      END
```

Vita

Kory J. Priestley was born on April 14, 1969 in Panorama City, California. He grew up in the cities of Ft. Lauderdale, FL, San Juan, Puerto Rico, Oklahoma City, OK, Memphis, TN, and Lancaster, CA. In May, 1987 he graduated from Paraclete High School in Lancaster, CA.

In September, 1987 Kory began his undergraduate studies in Mechanical Engineering at Antelope Valley College in Lancaster, CA. In September, 1989 he transferred to California Polytechnic State University at San Luis Obispo, where he received his Bachelor of Science Degree in March, 1992. He spent the summer of that same year working as a Langley Research Summer Scholar at NASA's Langley Research Center.

In August of 1992, Kory began his graduate studies at Virginia Polytechnic Institute & State University and earned a Master of Science Degree from the Mechanical Engineering Department in October, 1993. He served as a Graduate Research Assistant until June of 1993 when he was awarded a Fellowship from NASA's Graduate Student Researcher's Program.

Kory J. Priestley