RAT
A general-purpose computer program for RAdiative Transfer

Version 1.0
User Manual

John C. Chai
Nanyang Technological University
Singapore
At the invitation of Professor Michael F. Modest, I have started writing this version of the manual for a general-purpose computer program for radiative transfer problems. This version of the program is meant to serve as a starting point for you to further develop the finite-volume method for your own application.

This version of the computer code is the result of my work at the University of Minnesota. Special thanks go to Prof. Suhas V. Patankar for his valuable training, guidance and patience. His direct participation in the writing of this manual would definitely have made it more comprehensive and enlightening. Although he is not involved with the detail writing, his influence can be seen throughout this manual. I have followed the format of his latest book (Computation of Conduction and Duct Flow Heat Transfer, Taylor & Francis, 1991) in writing this manual.

Thanks also go to Dr. HaeOk S. Lee, who introduced me to the fascinating field of radiation heat transfer and for her guidance, especially during the initial part of my work in radiation heat transfer. I thank Prof. Roy. S. Amano for introducing me to computational fluid dynamics, which opened doors for me to pursue my knowledge in the field. I am indebted to Prof. Ephraim M. Sparrow. His support has resulted in this work on radiation heat transfer. I have benefited from numerous discussions with Dr. J. P. Moder over the years.

I would like to thank Mr. Prasenjit Rath for writing and typing the narration to some of the example problems. Lastly, but certain not least, I am grateful to my wife for her support and patience throughout the years.

John C. Chai
Singapore
December 2002
INTRODUCTION

1.1 Purpose of the Manual

This manual provides examples on how to use a two-dimensional general-purpose computer program for RAdiative Transfer; called RAT hereafter. Although RAT can be used to model a variety of radiative transfer problems in two-dimensional framework, three examples are shown in this version of the manual.

1.2 Capabilities and Limitations of RAT

This version of RAT is written to solve the steady-state form of the radiative transfer equation using the finite-volume method of Chai et al. (1994a) and Chai and Patankar (2000). It is designed for Cartesian coordinates. Irregular geometries with vertical or horizontal surfaces can be handled using the procedure proposed by Chai et al. (1994b). For this class of irregular geometries, the irregularities are captured exactly and no additional approximations are introduced by using the current version of RAT. Irregular geometries with inclined surfaces (this includes geometries with curved surfaces) can also be modeled using the procedure of Chai et al. (1994b). The inclined surfaces are however, approximated using staircase-like irregular geometries consist of vertical and horizontal surfaces. As a result, additional approximations are introduced in the modeling of inclined or curved surfaces. These types of irregular geometries can be modeled more accurately using a more advanced approach (Chai et. al., 1995). This will however, makes RAT more difficult to understand and use. As a result, a simpler version of RAT is included here.

Other than the above mentioned restriction, RAT is quite general. It can handle absorbing-emitting and scattering medium. Isotropic and anisotropic scattering can be modeled. Selected mie-scattering phase functions are incorporated into RAT. Black and diffusely reflecting walls can be modeled. Symmetry boundary condition is incorporated in RAT. Inhomogeneous medium and radiative equilibrium condition can be modeled. Both SI or English units can be used with RAT, as long as a consistent set of units is used. For ease of use, the current version of RAT sets the Stefan-Boltzmann constant, $\sigma$ to $W/m^2-K^4$. As a result, by default the length, mass and time must be in $m$, $kg$, and $sec$. Other units can be used by changing the value (and thus the units) of $\sigma$. 


1.3 Structure of RAT

There are two main modules in RAT. These are the invariant portion and the adaptation part. As the name implies, you should not have to change the invariant portion of the program for almost all of your problems which fall within the general capabilities of RAT. This part contains the solution procedure (using the FV method). The adaptation part of RAT is where you provide the problem-specific information; such as geometry, optical properties, boundary conditions, phase functions, output etc. This manual provides three example adaptations for you to get started on using RAT.

RAT is written using FORTRAN 77. Some newer features of FORTRAN are not exploited in this version of RAT. This is done intentionally so that RAT can be run using almost all compilers and computers without modifications.

1.4 How to run RAT

Four modules are needed to run RAT. These are PARAM.FOR, COMMON.FOR, RAT.FOR and ADAPT.FOR. In this nomenclature, RAT.FOR and ADAPT.FOR are the invariant part and the adaptation portion of the program. COMMON.FOR contains all the common block related variables. PARAM.FOR contains the parameters for the program.

For case-sensitive compilers and/or operating systems, the first two files, namely, PARAM.FOR and COMMON.FOR must be stored in upper-case. The other two modules can be in either upper or lower case. You must compile and link both RAT.FOR and ADAPT.FOR to create an executable file. The results can then be obtained by running the executable file. Note that since RAT.FOR does not change from problem-to-problem, you will need to compile it once. However, you should recompile ADAPT.FOR every time you make changes to it.

It is important that the parameters in PARAM.FOR are set properly. The meanings of the parameters are given in the nomenclature (Appendix B).
2.1 Black, Square Enclosure with Absorbing and Isothermal Medium (Example 1)

2.1-1 Problem Description

The problem under consideration is a steady-state radiation in participating hot medium surrounded by a black enclosure of square shape as shown in Fig. 2.1.1. The hot medium (at $T_g$) is assumed absorbing, emitting but non-scattering. The boundaries are at a prescribed temperature $T_w$. The medium has an uniform absorptivity $\kappa$. For the present problem the following values are used.

\[
T_w = 0 \text{ K, } \quad \varepsilon_w = 1, \quad \kappa = 10 \text{ m}^{-1}, \quad T_g = \left( \frac{1}{\sigma} \right)^{\frac{1}{4}} \quad (2.1.1)
\]

![Fig. 2.1.1 Radiation in an absorbing and isothermal medium.](image)
where $\sigma$ is Stefan-Boltzmann constant. Our aim is to calculate the total irradiation and boundary radiative heat flux distribution.

2.1-2 Design of ADAPT

GRID. The title of the field printout is set to ‘G’ through TITLE (1). The output file (PROB1.DAT) is then specified via OPEN. By default, the angular domains are $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$. These are specified in DEFLT using the variables TL and PL respectively. Two control angles represented by NCVLT = 2 are taken in the $\theta$-direction and 4 control angles represented by NCVLP = 4 are taken in the $\phi$-direction respectively. The default values of POWERT = 1 and POWERP = 1 are used to generate angular grids with uniform $\Delta\theta$ and $\Delta\phi$. The boundaries of the control angles are calculated by calling QUAD. The spatial domains are $0 \leq x \leq 1$ and $0 \leq y \leq 1$ which are specified through XL and YL respectively. Ten control volumes are used in the $x$ and $y$ directions which are represented by NCVLX = 10 and NCVLY = 10 respectively. The default values of POWERX = 1 and POWERY = 1 are used. As a result, an uniform spatial grid is created by calling EZGRID.

START. Numerical values of all boundary conditions as given in Eq. (2.1.1) are set here. The maximum number of iterations for the present problem are set as LAST = 20. The value of absorption coefficient is taken as ALPHA = 10. Then we fill $T(I,J)$ array by TEM, which serves as the temperature of hot gases at all interior control volumes. Boundary temperature is kept at the default value as all boundaries are at absolute zero temperature for the present problem.

LC. Inhomogeneous medium is set here. The present problem is homogeneous as absorption coefficient of the medium is constant here. The homogeneous absorption coefficient was specified in START.

OUTPUT. For each iteration (ITER) the value of an actual intensity, $F(I,J,L,M)$ at the center of the enclosure is printed and maximum difference in magnitude of the intensity from previous iteration, DMAX (which is printed as DIFF-MAX) over all control volumes and control angles is printed for verifying the convergence of the solution. The default value of EROR = 1.E-6 is used in this example for satisfying the convergence criteria. When the maximum number of iterations has been reached (ITER = LAST), or when the maximum change of the radiative intensities between two successive iterations falls below the prescribed value (KSTOP = 1), the boundary heat fluxes and incident radiation energy are calculated by calling HFLUX. Then incident radiation energy, $G(I,J)$ is nondimensionalised at all control volume nodes by dividing it by 4. All grid related variables and non-dimensional incident radiation energy at control volume nodes are printed by calling PRINT. QTOP represents the net radiative heat flux at the top boundary and QBOT represents the net radiative heat flux at the bottom boundary. $X(I)$ represents the value of $X$ at grid location $I$. Magnitude of QTOP and magnitude of QBOT are printed here for different $X(I)$. QLEFT represents the net radiative heat flux at the left boundary and QRITE represents the net radiative heat flux at the right boundary. $Y(J)$ represents the value of $Y$ at grid location $J$. Magnitude of QLEFT and magnitude of QRITE are printed here for different $Y(J)$.
GAMSOR. Irregular geometries and radiative equilibrium conditions are set here. Since in our present problem geometry of the enclosure is not irregular and it is in non-radiative equilibrium condition, hence no operation is performed here.

2.1-3 Additional Fortran Names

DMAX maximum of \( \left( \frac{I - I_{OLD}}{I} \right) \) calculated over all control volumes and control angles

LAST maximum number of iterations

ALPHA absorption coefficient

STFAN Stefan-Boltzmann constant

\( G(I,J) \) incident radiation

TEM non-dimensional gas temperature inside the enclosure

QTOP net radiative heat flux at top boundary

Q80T net radiative heat flux at bottom boundary

QLEFT net radiative heat flux at left boundary

QRITE net radiative heat flux at right boundary

2.1-4 Listing of ADAPT for Example 1

```fortran
C*******************************************************************
SUBROUTINE ADAPT
C*******************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C*******************************************************************
C  PROBLEM 1: BLACK, SQUARE ENCLOSURE WITH ABSORBING
C                     AND ISOTHERMAL MEDIUM
C*******************************************************************
ENTRY GRID
C
TITLE(1)=' G '
OPEN(7,FILE='PROB1.DAT')
C
NCVLVP=4
NCVLT=2
C
CALL QUAD
C
NCVLX=10
NCVLY=10
C
XL=1.
YL=1.
C
CALL EZGRID
C
RETURN
C*******************************************************************
ENTRY START
C
LAST=20
ALPHA=10.
C
TEM=(1./STFAN)**(1./4.)
```
C     DO 110 J=2,M2
     DO 111 I=2,L2
     T(I,J)=TEM
 111    CONTINUE
 110   CONTINUE
C
RETURN
C***************************************************************************
ENTRY LC
C
RETURN
C***************************************************************************
ENTRY OUTPUT
C
IF(ITER.EQ.0) WRITE(6,500)
WRITE(6,501) ITER, F(L1/2,M1/2,2,2),DMAX
IF(ITER.EQ.0) WRITE(7,500)
WRITE(7,501) ITER, F(L1/2,M1/2,2,2),DMAX
C
IF(KSTOP.EQ.1.OR.ITER.EQ.LAST) THEN
    PAUSE
    CALL HFLUX
C
    DO 521 J=1,M1
    DO 522 I=1,L1
    522       CONTINUE
    521    CONTINUE
C
    CALL PRINT
    PAUSE
C
    WRITE(6,502)
    WRITE(7,502)
C
    DO 510 I=2,L2
        WRITE(6,503) X(I),ABS(QPY(I,M1)-QMY(I,M1)),
                      1                        ABS(QPY(I,1)-QMY(I,1))
    510      CONTINUE
C
    WRITE(6,504)
    WRITE(7,504)
C
    DO 511 J=2,M2
        WRITE(6,503) Y(J),ABS(QPX(1,J)-QMX(1,J)),
                      1                        ABS(QPX(L1,J)-QMX(L1,J))
    511      CONTINUE
C
ENDIF
C
500  FORMAT(/3X,'ITER',8X,'F',12X,'DIFF-MAX'/1X,48('*'))
501  FORMAT(3X,I3,2(3X,1PE12.3))
502  FORMAT(/8X,'X',10X,'QTOP',7X,'QBOT'/1X,39('*'))
503  FORMAT(1X,3(1PE12.3))
504  FORMAT(/8X,'Y',9X,'QLEFT',7X,'QRITE'/1X,39('*'))
C
RETURN
C***************************************************************************
ENTRY GAMSOR
### 2.1-5 Results for Example 1

<table>
<thead>
<tr>
<th>ITER</th>
<th>F</th>
<th>DIFF-MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>1</td>
<td>3.158E-01</td>
<td>1.000E+00</td>
</tr>
<tr>
<td>2</td>
<td>3.158E-01</td>
<td>0.000E+00</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
I &= 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
X &= 0.00E+00 & 5.00E-02 & 1.50E-01 & 2.50E-01 & 3.50E-01 & 4.50E-01 & 5.50E-01 \\
XU &= 0.00E+00 & 0.00E+00 & 1.00E-01 & 2.00E-01 & 3.00E-01 & 4.00E-01 & 5.00E-01 \\
\end{align*}
\]

\[
\begin{align*}
I &= 8 & 9 & 10 & 11 & 12 \\
X &= 6.50E-01 & 7.50E-01 & 8.50E-01 & 9.50E-01 & 1.00E+00 \\
XU &= 6.00E-01 & 7.00E-01 & 8.00E-01 & 9.00E-01 & 1.00E+00 \\
\end{align*}
\]

\[
\begin{align*}
J &= 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
Y &= 0.00E+00 & 5.00E-02 & 1.50E-01 & 2.50E-01 & 3.50E-01 & 4.50E-01 & 5.50E-01 \\
YV &= 0.00E+00 & 0.00E+00 & 1.00E-01 & 2.00E-01 & 3.00E-01 & 4.00E-01 & 5.00E-01 \\
\end{align*}
\]

\[
\begin{align*}
J &= 8 & 9 & 10 & 11 & 12 \\
Y &= 6.50E-01 & 7.50E-01 & 8.50E-01 & 9.50E-01 & 1.00E+00 \\
YV &= 6.00E-01 & 7.00E-01 & 8.00E-01 & 9.00E-01 & 1.00E+00 \\
\end{align*}
\]

\[
\begin{align*}
L &= 1 & 2 & 3 & 4 & 5 & 6 \\
\theta &= 0.00E+00 & 7.85E-01 & 2.36E+00 & 3.14E+00 \\
\end{align*}
\]

\[
\begin{align*}
M &= 1 & 2 & 3 & 4 & 5 & 6 \\
\phi &= 0.00E+00 & 7.85E-01 & 2.36E+00 & 3.93E+00 & 5.50E+00 & 6.28E+00 \\
\end{align*}
\]

\[
\begin{align*}
I &= 8 & 9 & 10 & 11 & 12 \\
J &= 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\
\end{align*}
\]

\[
\begin{align*}
I &= 8 & 9 & 10 & 11 & 12 \\
J &= 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\
\end{align*}
\]
2.1-6 Discussion of Results

It can be seen that the solution is converged in one iteration. This is because the medium is non-scattering and the walls are non-reflecting walls. In the result lists for the present problem, along \( x \)-direction \( X(I) \) and \( XU(I) \) represents the value of \( X \) at grid location \( I \) and the value of \( X \) for the corresponding control volume face. Similarly, along \( y \)-direction \( Y(J) \) and \( YV(J) \) represents the value of \( Y \) at grid location \( J \) and value of \( Y \) for the corresponding control volume face. Angular grid contains grid related information in \( \theta \) and \( \phi \)-directions respectively. In the \( \theta \)-direction \( TH(L) \) represents the value of \( \theta \) at the grid location \( L \) and in the \( \phi \)-direction \( PH(M) \) represents the value of \( \phi \) at the grid location \( M \). The final field printout of incident radiation energy shows that the effect of hot gases inside the enclosure is to create a maximum irradiation (incident radiation energy), \( G(I,J) \) at the center of the enclosure whose non-dimensional value is numerically evaluated by finite-volume method as 0.995. The distribution of irradiation inside the medium is plotted as shown in Fig. 2.1.2 for absorption coefficients, \( \kappa = 10 \, m^{-1} \) and \( \kappa = 1 \, m^{-1} \). As we move towards the boundary from the center of the enclosure the magnitude of irradiation decreases and we get minimum irradiation at the boundaries. It was also noted that the irradiation is symmetrical about the vertical and horizontal centerlines of the enclosure. From Fig. 2.1.2 it is seen that as the value of absorption coefficient, \( \kappa \) decreases the magnitude of the incident radiation energy decreases. It is because the magnitude of intensity decreases along the path of travel as absorption coefficient decreases. The field printout of boundary heat fluxes shows that the net radiative heat fluxes are maximum at the center of each boundary and are symmetrical about the center of the boundaries.
Fig. 2.1.2 Distribution of incident radiation energy for absorption coefficients, $\kappa = 10 \, \text{m}^{-1}$ and $\kappa = 1 \, \text{m}^{-1}$.

2.1-7 Final Remarks

The distribution of incident radiation energy and wall heat fluxes in a square enclosure are presented in this section. The medium inside the enclosure is absorbing and emitting. It was noted that the solution converged well after one iteration since the walls are non-reflecting and temperature of all the four walls are known. Effect of imposing the symmetry condition at the boundaries can also be studied for this present problem which is discussed in the next example.

2.2 Black, Square Enclosure with Absorbing and Isothermal Medium with Symmetry Condition at Right and Bottom Boundaries (Example 2)

2.2-1 Problem Description

The problem under consideration is same as Example 1. Due to symmetries, one-quarter of the domain in Example 1 is simulated. It is a steady-state radiation problem in participating hot medium surrounded by a black enclosure of square shape as shown in Fig. 2.2.1. The medium is assumed absorbing, emitting but non-scattering. Left and top walls are at a prescribed temperature $T_w$. The medium has an uniform absorptivity $\kappa$. For the present problem the following values are used.

$$T_w = 0 \, \text{K}, \quad \varepsilon_w = 1, \quad \kappa = 10 \, \text{m}^{-1}, \quad T_g = \left( \frac{1}{\sigma} \right)^{\frac{1}{4}} \quad (2.2.1)$$
Fig. 2.2.1 Radiation in an absorbing and isothermal medium.

where \( \sigma \) is Stefan-Boltzmann constant. Our aim is to calculate the total irradiation and boundary radiative heat flux distribution.

### 2.2-2 Design of ADAPT

**GRID.** The title of the field printout is set to ‘G’ through TITLE (1). The output file (PROB2.DAT) is then specified via OPEN. By default, the angular domains are \( 0 \leq \theta \leq \pi \) and \( 0 \leq \phi \leq 2\pi \). These are specified in DEFLT using the variables TL and PL respectively. Two control angles represented by \( \text{NCVLT} = 2 \) are taken in the \( \theta \)-direction and 4 control angles represented by \( \text{NCVLP} = 4 \) are taken in the \( \phi \)-direction respectively. The default value of \( \text{POWERT} = 1 \) and \( \text{POWERP} = 1 \) are used to generate angular grids with uniform \( \Delta \theta \) and \( \Delta \phi \). The boundaries of the control angles are calculated by calling QUAD. The spatial domains are \( 0 \leq x \leq 0.5 \) and \( 0 \leq y \leq 0.5 \) which are specified through XL and YL respectively. Five control volumes are used in the \( x \) and \( y \) directions which are represented by \( \text{NCVLX} = 5 \) and \( \text{NCVLY} = 5 \) respectively. The default values of \( \text{POWERX} = 1 \) and \( \text{POWERY} = 1 \) are used. As a result, an uniform spatial grid is created by calling EZGRID.

**START.** Numerical values of all boundary conditions as given in Eq. (2.2.1) are set here. The maximum number of iterations for the present problem are set as \( \text{LAST} = 20 \). The value of absorption coefficient is taken as \( \text{ALPHA} = 10 \). Then we fill \( T(I,J) \) array by TEM, which serves as the temperature of hot gases at all interior control volumes. \( \text{KBCJ1(I)} = 2 \) and \( \text{KBCL1(J)} = 2 \) are set for symmetry along east (KBCL1) and south (KBCJ1) boundaries respectively. Boundary temperature is kept at the default value as west and north boundaries are at absolute zero temperature for the present problem.
Inhomogeneous medium is set here. The present problem is homogeneous as absorption coefficient of the medium is constant here. The homogeneous absorption coefficient was specified in START.

OUTPUT. For each iteration (ITER) the value of an actual intensity, \( F(I,J,L,M) \) at the center of the enclosure is printed and maximum difference in magnitude of the intensity from previous iteration, DMAX (which is printed as DIFF-MAX) over all control volumes and control angles is printed for verifying the convergence of the solution. The default value of ERROR = 1.E-6 is used in this example for satisfying the convergence criteria. When the maximum number of iterations has been reached (ITER = LAST), or when the maximum change of the radiative intensities between two successive iterations falls below the prescribed value (KSTOP = 1), the boundary heat fluxes and incident radiation energy are calculated by calling HFLUX. Then incident radiation energy, \( G(I,J) \) is nondimensionalised at all control volume nodes by dividing it by 4. All grid related variables and non-dimensional incident radiation energy at control volume nodes are printed by calling PRINT. QTOP represents the net radiative heat flux at the top boundary and QBOT represents the net radiative heat flux at the bottom boundary. \( X(I) \) represents the value of \( X \) at grid location \( I \). Magnitude of QTOP and magnitude of QBOT are printed here for different \( X(I) \). QLEFT represents the net radiative heat flux at the left boundary and QRITE represents the net radiative heat flux at the right boundary. \( Y(J) \) represents the value of \( Y \) at grid location \( J \). Magnitude of QLEFT and magnitude of QRITE are printed here for different \( Y(J) \).

GAMSOR. Irregular geometries and radiative equilibrium conditions are set here. Since in our present problem geometry of the enclosure is not irregular and it is in non-radiative equilibrium condition, hence no operation is performed here.

2.2-3 Additional Fortran Names

- DMAX: maximum of \( \left| \left( I - I_{OLD} \right) / I \right| \) calculated over all control volumes and control angles
- LAST: maximum number of iterations
- KBCL1(J): set for type of boundary condition along east boundary
- KBCJ1(I): set for type of boundary condition along south boundary
- ALPHA: absorption coefficient
- STFAN: Stefan-Boltzmann constant
- G(I,J): incident radiation
- TEM: non-dimensional gas temperature inside the enclosure
- QTOP: net radiative heat flux at top boundary
- QBOT: net radiative heat flux at bottom boundary
- QLEFT: net radiative heat flux at left boundary
- QRITE: net radiative heat flux at right boundary
2.2-4 Listing of ADAPT for Example 2

C*******************************************************************
SUBROUTINE ADAPT
C*******************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C*******************************************************************
C PROBLEM 2: BLACK, SQUARE ENCLOSURE WITH ABSORBING
AND ISOTHERMAL MEDIUM WITH SYMMETRIES
AT THE RIGHT AND BOTTOM BOUNDARIES
C*******************************************************************

ENTRY GRID

TITLE(1)= ' G '
OPEN(7,FILE='PROB2.DAT')

NCVLP=4
NCVLT=2

CALL QUAD

NCVLX=5
NCVLY=5

XL=0.5
YL=0.5

CALL EZGRID

RETURN
C*******************************************************************
ENTRY START

LAST=20
ALPHA=10.

TEM=(1./STFAN)**(1./4.)

DO 110 J=2,M2
   DO 111 I=2,L2
      T(I,J)=TEM
      KBCL1(J)=2
      KBCJ1(I)=2
   111   CONTINUE
110   CONTINUE

RETURN
C*******************************************************************
ENTRY LC

RETURN
C*******************************************************************
ENTRY OUTPUT

IF(ITER.EQ.0) WRITE(6,500)
WRITE(6,501) ITER, F(L1/2,M1/2,2,2),DMAX
IF(ITER.EQ.0) WRITE(7,500)
WRITE(7,501) ITER, F(L1/2,M1/2,2,2),DMAX

IF(KSTOP.EQ.1.OR.ITER.EQ.LAST) THEN
   PAUSE
   CALL HFLUX
C
DO 521 J=1,M1
DO 522 I=1,L1
522    CONTINUE
521 CONTINUE
C
CALL PRINT
PAUSE
C
WRITE(6,502)
WRITE(7,502)
C
DO 510 I=2,L2
WRITE(6,503) X(I),ABS(QPY(I,M1)-QMY(I,M1)),
1                     ABS(QPY(I,1)-QMY(I,1))
WRITE(7,503) X(I),ABS(QPY(I,M1)-QMY(I,M1)),
1                     ABS(QPY(I,1)-QMY(I,1))
510      CONTINUE
C
WRITE(6,504)
WRITE(7,504)
C
DO 511 J=2,M2
WRITE(6,503) Y(J),ABS(QPX(1,J)-QMX(1,J)),
1                        ABS(QPX(L1,J)-QMX(L1,J))
WRITE(7,503) Y(J),ABS(QPX(1,J)-QMX(1,J)),
1                        ABS(QPX(L1,J)-QMX(L1,J))
511      CONTINUE
C
ENDIF
C
500  FORMAT(/3X,'ITER',8X,'F',12X,'DIFF-MAX'/1X,48('*'))
501  FORMAT(3X,I3,2(3X,1PE12.3))
502  FORMAT(/8X,'X',10X,'QTOP',7X,'QBOT'/1X,39('*'))
503  FORMAT(1X,3(1PE12.3))
504  FORMAT(/8X,'Y',9X,'QLEFT',7X,'QRITE'/1X,39('*'))
C
RETURN
C*******************************************************************
ENTRY GAMSOR
C
RETURN
END
C*******************************************************************

2.2-5 Results for Example 2

<table>
<thead>
<tr>
<th>ITER</th>
<th>F</th>
<th>DIFF-MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>1</td>
<td>2.586E-01</td>
<td>1.000E+00</td>
</tr>
<tr>
<td>2</td>
<td>2.829E-01</td>
<td>3.998E-01</td>
</tr>
<tr>
<td>3</td>
<td>2.829E-01</td>
<td>1.648E-01</td>
</tr>
<tr>
<td>4</td>
<td>2.829E-01</td>
<td>0.000E+00</td>
</tr>
</tbody>
</table>

I = 1 2 3 4 5 6 7
X = 0.00E+00 5.00E-02 1.50E-01 2.50E-01 3.50E-01 4.50E-01 5.00E-01
XU= 0.00E+00 0.00E+00 1.00E-01 2.00E-01 3.00E-01 4.00E-01 5.00E-01

J = 1 2 3 4 5 6 7
Y = 0.00E+00 5.00E-02 1.50E-01 2.50E-01 3.50E-01 4.50E-01 5.00E-01
YV= 0.00E+00 0.00E+00 1.00E-01 2.00E-01 3.00E-01 4.00E-01 5.00E-01
2.2-6 Discussion of Results

It can be seen that the solution is not converged in one iteration. This is because of the fact that due to the unknown temperature of right and bottom boundaries, intensities are unknown in these boundaries. Hence, iteration procedure starts by guessing intensity in the boundaries where temperature is unknown. In the result lists for the present problem, along x-direction \( X(I) \) and \( XU(I) \) represents the value of \( X \) at grid location \( I \) and the value of \( X \) for the corresponding control volume face. Similarly, along y-direction \( Y(J) \) and \( YV(J) \) represents the value of \( Y \) at grid location \( J \) and value of \( Y \) for the corresponding control volume face. Angular grid contains grid related information in \( \theta \) and \( \phi \)-directions respectively. In the \( \theta \)-direction \( TH(L) \) represents the value of \( \theta \) at the grid location \( L \) and in the \( \phi \)-direction \( PH(M) \) represents the value of \( \phi \) at the grid location \( M \). The final field printout of incident radiation energy shows that the effect of hot gases inside the enclosure is to create a maximum irradiation (incident radiation energy), \( G(I,J) \) along east and south walls of the enclosure whose maximum value is 0.995 which is same as we got from example 1. This is because the present problem is same as previous example, but we are solving only the one-forth of the previous example. The distribution of irradiation inside the medium is plotted as shown in Fig. 2.2.2 for absorption coefficients, \( \kappa = 10 \, m^{-1} \) and \( \kappa = 1 \, m^{-1} \). As we move towards north-west boundary from south-east boundary of the enclosure the magnitude of irradiation decreases and we get minimum irradiation in north and west boundaries. From Fig. 2.2.2 it is seen that as the value of absorption coefficient, \( \kappa \) decreases the magnitude incident radiation energy decreases. It is
because the magnitude of the intensity decreases along the path of travel as absorption coefficient decreases. The field printout of boundary heat fluxes shows that the net radiative heat fluxes are maximum at the top and left boundaries and minimum at bottom and right boundaries. This is because the bottom and right boundaries are hot. Hence, heat flux due to its own boundary temperature is existing which is nullifying the heat flux reaching to that boundary from all other boundaries. Therefore, the absolute value of net radiative heat flux decreases.

![Image of field printout showing boundary heat fluxes]

**Fig. 2.2.2** Distribution of incident radiation energy for absorption coefficients, $\kappa = 10 \, m^{-1}$ and $\kappa = 1 \, m^{-1}$.

### 2.2-7 Final Remarks

Effect of imposing the symmetry boundary condition on distribution of incident radiation energy and wall heat fluxes has been studied in this example problem. Here, the solution is not converged in one iteration as intensity at the symmetry boundaries are unknown due to unknown temperature on these boundaries. Till now, the examples we have discussed where medium is assumed non-scattering. Effect of imposing the scattering medium is also studied which is discussed in the next example.

### 2.3 Black, Square Enclosure with Absorbing and Anisotropically Scattering Medium and Hot Bottom Wall (Example 3)

#### 2.3-1 Problem Description

The problem under consideration is steady-state radiation in an absorbing and anisotropically scattering medium surrounded by a black enclosure of square shape as shown in Fig. 2.3.1. The bottom wall is kept hot and other three walls are kept at absolute zero temperature. The medium has an uniform absorptivity $\kappa$ and scattering coefficient $\sigma_s$. For the present problem the following values are used.
\[ T_{bottom} = \left( \frac{1}{\sigma} \right)^{\frac{1}{4}}, \quad T_{top} = T_{left} = T_{right} = 0 \text{ K}, \quad T_g = 0 \text{ K}, \quad \varepsilon_w = 1, \quad \kappa = 0.5 \text{ m}^{-1}, \quad \sigma_s = 0.5 \text{ m}^{-1} \quad (2.3.1) \]

Fig. 2.3.1 Radiation in an absorbing and anisotropically scattering medium.

where \( \sigma \) is Stefan-Boltzmann constant. Our aim is to calculate the total irradiation and boundary radiative heat flux distribution.

### 2.3-2 Design of ADAPT

**GRID.** The title of the field printout is set to ‘G’ through `TITLE (1)`. The output file (`PROB3.DAT`) is then specified via `OPEN`. By default, the angular domains are \( 0 \leq \theta \leq \pi \) and \( 0 \leq \phi \leq 2\pi \). These are specified in `DEFLT` using the variables `TL` and `PL` respectively. Two control angles represented by `NCVLTL = 2` are taken in the \( \theta \)-direction and 8 control angles represented by `NCVLP = 8` are taken in the \( \phi \)-direction respectively. The default value of `POWER = 1` and `POWERT = 1` are used to generate angular grids with uniform \( \Delta \theta \) and \( \Delta \phi \). The boundaries of the control angles are calculated by calling `QUAD`. The spatial domains are \( 0 \leq x \leq 1 \) and \( 0 \leq y \leq 1 \) which are specified through `XL` and `YL` respectively. Five control volumes are used in the \( x \) and \( y \) directions which are represented by `NCVLX = 5` and `NCVLY = 5` respectively. The default values of `POWERX = 1` and `POWERY = 1` are used. As a result, an uniform spatial grid is created by calling `EZGRID`.

**START.** Numerical values of all boundary conditions as given in Eq. (2.3.1) are set here. The maximum number of iterations for the present problem are set as `LAST = 20`. Anisotropic scattering medium is set as `KISO = 0`. Back scattering is specified by `KPHASE = 7`. The value of absorption coefficient and scattering coefficient are taken as `ALPHA = 0.5` and `SIG = 0.5` respectively. Then we fill `T(I,1)` array by `TEM`, which serves as the temperature of hot bottom wall. Other boundary temperatures are kept at the default value as all
the boundaries except the bottom boundary are at absolute zero temperature for the present problem.

**LC.** Inhomogeneous medium is set here. The present problem is homogeneous as absorption coefficient and scattering coefficient of the medium is constant here. The homogeneous absorption coefficient and scattering coefficient was specified in \texttt{START}.

**OUTPUT.** For each iteration (\texttt{ITER}) the value of an actual intensity, \( F(I,J,L,M) \) at the center of the enclosure is printed and maximum difference in magnitude of the intensity from previous iteration, \( \text{DMAX} \) (which is printed as \texttt{DIFF-MAX}) over all control volumes and control angles is printed for verifying the convergence of the solution. The default value of \( \texttt{EROR} = 1.E-6 \) is used in this example for satisfying the convergence criteria. When the maximum number of iterations has been reached (\texttt{ITER} = \texttt{LAST}), or when the maximum change of the radiative intensities between two successive iterations falls below the prescribed value (\texttt{KSTOP} = 1), the boundary heat fluxes and incident radiation energy are calculated by calling \texttt{HFLUX}. All grid related variables and incident radiation energy at control volume nodes are printed by calling \texttt{PRINT}. \texttt{QTOP} represents the net radiative heat flux at the top boundary and \texttt{QBOT} represents the net radiative heat flux at the bottom boundary. \( X(I) \) represents the value of \( X \) at grid location \( I \). Magnitude of \texttt{QTOP} and magnitude of \texttt{QBOT} are printed here for different \( X(I) \). \texttt{QLEFT} represents the net radiative heat flux at the left boundary and \texttt{QRITE} represents the net radiative heat flux at the right boundary. \( Y(J) \) represents the value of \( Y \) at grid location \( J \). Magnitude of \texttt{QLEFT} and magnitude of \texttt{QRITE} are printed here for different \( Y(J) \).

**GAMSOR.** Irregular geometries and radiative equilibrium conditions are set here. Since in our present problem geometry of the enclosure is not irregular and it is in non-radiative equilibrium condition, hence no operation is performed here.

### 2.3-3 Additional Fortran Names

- **DMAX**: maximum of \(|(I-I_{OLD})_I|\) calculated over all control volumes and control angles
- **LAST**: maximum number of iterations
- **ALPHA**: absorption coefficient
- **SIG**: scattering coefficient
- **KISO**: set for type of scattering
- **KPHASE**: set for back scattering
- **STFAN**: Stefan-Boltzmann constant
- **G(I,J)**: incident radiation energy
- **TEM**: non-dimensional temperature of bottom wall
- **QTOP**: net radiative heat flux at top boundary
- **QBOT**: net radiative heat flux at bottom boundary
- **QLEFT**: net radiative heat flux at left boundary
- **QRITE**: net radiative heat flux at right boundary
2.3-4 Listing of ADAPT for Example 3

C*******************************************************************
SUBROUTINE ADAPT
C*******************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C*******************************************************************
C PROBLEM 3: BLACK, SQUARE ENCLOSURE WITH PARTICIPATING MEDIUM
C AND HOT BOTTOM WALL
C*******************************************************************
C
ENTRY GRID
C
TITLE(1)= ' G ',
OPEN(7,FILE='PROB3.DAT')
C
NCVLP=8
NCVLT=2
C
CALL QUAD
C
NCVLX=5
NCVLY=5
C
XL=1.
YL=1.
C
CALL EZGRID
C
RETURN
C*******************************************************************
ENTRY START
C
LAST=20
KISO=0
KPHASE=7
ALPHA=0.5
SIG=0.5
C
TEM=(1./STFAN)**(1./4.)
C
DO 11 I=2,L1-1
     T(I,1)=TEM
11    CONTINUE
C
RETURN
C*******************************************************************
ENTRY LC
RETURN
C*******************************************************************
ENTRY OUTPUT
C
IF(ITER.EQ.0) WRITE(6,500)
WRITE(6,501) ITER, F(L1/2,M1/2,2,2),DMAX
IF(ITER.EQ.0) WRITE(7,500)
WRITE(7,501) ITER, F(L1/2,M1/2,2,2),DMAX
C
IF(KSTOP.EQ.1.OR.ITER.EQ.LAST) THEN
     CALL HFLUX
     CALL PRINT
C
RETURN
WRITE(6,502)
WRITE(7,502)
C
DO 510 I=2,L2
   WRITE(6,503) X(I),ABS(QPY(I,M1)-QMY(I,M1)),
   1           ABS(QPY(I,1)-QMY(I,1))
   WRITE(7,503) X(I),ABS(QPY(I,M1)-QMY(I,M1)),
   1           ABS(QPY(I,1)-QMY(I,1))
510 CONTINUE
C
WRITE(6,504)
WRITE(7,504)
C
DO 511 J=2,M2
   WRITE(6,503) Y(J),ABS(QPX(1,J)-QMX(1,J)),
   1           ABS(QPX(L1,J)-QMX(L1,J))
   WRITE(7,503) Y(J),ABS(QPX(1,J)-QMX(1,J)),
   1           ABS(QPX(L1,J)-QMX(L1,J))
511 CONTINUE
C
ENDIF
C
500 FORMAT(/3X,'ITER',8X,'F',12X,'DIFF-MAX'/1X,48('*'))
501 FORMAT(3X,I3,2(3X,1PE12.3))
502 FORMAT(/8X,'X',10X,'QTOP',7X,'QBOT'/1X,39('*'))
503 FORMAT(1X,3(1PE12.3))
504 FORMAT(/8X,'Y',9X,'QLEFT',7X,'QRITE'/1X,39('*'))
C
RETURN
C***************************************************************************
C
ENTRY GAMSOR
C
RETURN
END
C***************************************************************************

2.3-5 Results for Example 3

<table>
<thead>
<tr>
<th>ITER</th>
<th>F</th>
<th>DIFF-MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>1</td>
<td>4.147E-02</td>
<td>1.000E+00</td>
</tr>
<tr>
<td>2</td>
<td>5.099E-02</td>
<td>1.000E+00</td>
</tr>
<tr>
<td>3</td>
<td>5.363E-02</td>
<td>2.027E-01</td>
</tr>
<tr>
<td>4</td>
<td>5.413E-02</td>
<td>5.372E-02</td>
</tr>
<tr>
<td>5</td>
<td>5.424E-02</td>
<td>1.312E-02</td>
</tr>
<tr>
<td>6</td>
<td>5.426E-02</td>
<td>3.220E-03</td>
</tr>
<tr>
<td>7</td>
<td>5.427E-02</td>
<td>7.630E-04</td>
</tr>
<tr>
<td>8</td>
<td>5.427E-02</td>
<td>1.801E-04</td>
</tr>
<tr>
<td>9</td>
<td>5.427E-02</td>
<td>4.190E-05</td>
</tr>
<tr>
<td>10</td>
<td>5.427E-02</td>
<td>9.730E-06</td>
</tr>
<tr>
<td>11</td>
<td>5.427E-02</td>
<td>2.277E-06</td>
</tr>
<tr>
<td>12</td>
<td>5.427E-02</td>
<td>8.697E-07</td>
</tr>
</tbody>
</table>

I = 1 2 3 4 5 6 7
X = 0.00E+00 1.00E-01 3.00E-01 5.00E-01 7.00E-01 9.00E-01 1.00E+00
XU= 0.00E+00 0.00E+00 2.00E-01 4.00E-01 6.00E-01 8.00E-01 1.00E+00

J = 1 2 3 4 5 6 7
Y = 0.00E+00 1.00E-01 3.00E-01 5.00E-01 7.00E-01 9.00E-01 1.00E+00
YV= 0.00E+00 0.00E+00 2.00E-01 4.00E-01 6.00E-01 8.00E-01 1.00E+00
It can be seen that the solution is not converged in one iteration. This is because the medium is scattering. Hence, at each node source function is unknown. Therefore, at the beginning we are guessing the source function. On the basis of guess value of source function intensity is calculated at the next iteration and it is compared with intensity calculated from previous iteration. This iterative process will continue till the maximum difference in intensity between two successive iterations will be less than or equal to \( \text{EROR} \). Here it is seen that after twelve iterations solution is converged. In the result lists for the present problem, along \( x \) direction \( X(I) \) and \( XU(I) \) represents the value of \( X \) at grid location \( I \) and the value of \( X \) for the corresponding control volume face. Similarly, along \( y \) direction \( Y(J) \) and \( YV(J) \) represents the value of \( Y \) at grid location \( J \) and value of \( Y \) for the corresponding control volume face. Angular grid contains grid related information in \( \theta \) and \( \phi \)-directions respectively. In the \( \theta \)-direction \( TH(L) \) represents the value of \( \theta \) at the grid location \( L \) and in the \( \phi \)-direction \( PH(M) \) represents the value of \( \phi \) at the grid location \( M \). The final field printout of incident radiation energy shows that the effect of hot bottom wall is to create maximum irradiation (incident radiation energy), \( G(I,J) \) at the bottom of the enclosure which is numerically evaluated by finite volume method as 2.2. The distribution of irradiation inside the medium is plotted for backward and forward scattering as shown in Fig. 2.3.2 for absorption coefficient, \( \kappa = 0.5 \ m^{-1} \).
and scattering coefficient, $\sigma_s = 0.5 \text{ m}^{-1}$. As we go away from the bottom boundary of the enclosure towards the top, the magnitude of irradiation decreases and we get minimum irradiation at the top boundary. It is also seen that backward scattering gives more irradiation at the bottom wall compared to forward scattering. The field printout of boundary heat fluxes shows that the net radiative heat flux at the bottom wall is maximum and is symmetrical about the center of the bottom boundary. Net radiative heat fluxes at left and right boundaries are same and increases along these walls as we come closer to the hot bottom wall.

Fig. 2.3.2 Distribution of incident radiation energy for backward and forward scattering with absorption coefficient, $\kappa = 0.5 \text{ m}^{-1}$ and scattering coefficient, $\sigma_s = 0.5 \text{ m}^{-1}$.

2.3-7 Final Remarks

Effect of scattering (forward and backward) on incident radiation energy and wall heat fluxes are discussed here. It was noted that the intensity is not converged after one iteration unlike Example 1. This is because of the fact that the source function due to in-scattering is unknown at each control volume nodes.

2.4 Black, Square Enclosure with Inhomogeneous Absorbing and Anisotropically Scattering Medium and Hot Bottom Wall (Example 4)

2.4-1 Problem Description

The problem under consideration is steady-state radiation in an inhomogeneous absorbing and anisotropically scattering medium surrounded by a black enclosure of square shape as shown in Fig. 2.4.1. The bottom wall is kept hot and other three walls are kept at absolute zero temperature. The absorption coefficients $\kappa_1$ and $\kappa_2$ and scattering coefficients $\sigma_{s,1}$ and $\sigma_{s,2}$ are shown in Fig. 2.4.1. For the present problem the following values are used.
\[ T_{\text{bottom}} = \left( \frac{1}{\sigma} \right)^{1/4}, \ T_{\text{top}} = T_{\text{left}} = T_{\text{right}} = 0 \, \text{K}, \ T_g = 0 \, \text{K}, \ \varepsilon_w = 1 \]  
\[ \kappa_1 = 0.5 \, m^{-1}, \ \sigma_{s,1} = 0.5 \, m^{-1} \quad \kappa_2 = 10 \, \kappa_1 \quad \sigma_{s,2} = 10 \, \sigma_{s,2} \]  

(2.4.1) \hspace{1cm} (2.4.2)

Fig. 2.4.1 Radiation in an inhomogeneous absorbing and anisotropically scattering medium.

where \( \sigma \) is Stefan-Boltzmann constant. Our aim is to calculate the total irradiation and boundary radiative heat flux distribution.

2.4-2 Design of ADAPT

**GRID.** The title of the field printout is set to ‘G’ through TITLE (1). The output file (PROB4.DAT) is then specified via OPEN. By default, the angular domains are \( 0 \leq \theta \leq \pi \) and \( 0 \leq \phi \leq 2\pi \). These are specified in DEFLT using the variables TL and PL respectively. Four control angles represented by NCVLT = 4 are taken in the \( \theta \)-direction and 8 control angles represented by NCVLP = 8 are taken in the \( \phi \)-direction respectively. The default value of POWERT = 1 and POWERP = 1 are used to generate angular grids with uniform \( \Delta \theta \) and \( \Delta \phi \). The boundaries of the control angles are calculated by calling QUAD. The spatial domains are \( 0 \leq x \leq 1 \) and \( 0 \leq y \leq 1 \) which are specified through XL and YL respectively. Ten control volumes are used in the \( x \) and \( y \) directions which are represented by NCVLX = 10 and NCVLY = 10 respectively. The default values of POWERX = 1 and POWERY = 1 are used. As a result, an uniform spatial grid is created by calling EZGRID.

**START.** Numerical values of all boundary conditions as given in Eqs. (2.4.1) and (2.4.2) are set here. The maximum number of iterations for the present problem are set as LAST = 30. Anisotropic scattering medium is set as KISO = 0. Forward scattering (F1 phase function) is specified by KPHASE = 2. The value of absorption coefficient \( \kappa_1 \) and scattering coefficient \( \sigma_{s,1} \) are taken as ALPHA = 0.5 and SIG = 0.5 respectively. Then we fill \( T(I,1) \) array by TEM, which serves as the temperature of hot bottom wall. Other boundary temperatures are kept at the default value as all the boundaries except the bottom boundary are at absolute zero temperature for the present problem.

**LC.** Inhomogeneous medium is set here. For \( 0.3 \leq x \leq 0.7 \) and \( 0.3 \leq y \leq 0.7 \), the absorption coefficient \( \kappa_2 \) and scattering coefficient \( \sigma_{s,2} \) are specified by modifying the CAPPA(I,J) and SIGMA(I,J) arrays respectively. The absorption coefficient and
scattering coefficient for the remainder of the domain remain unchanged and are equal to \( \kappa_1 \) and \( \sigma_{s,1} \) respectively.

**OUTPUT.** For each iteration (ITER) the value of an actual intensity, \( F(I, J, L, M) \) at the center of the enclosure is printed and maximum difference in magnitude of the intensity from previous iteration, DMAX (which is printed as DIFF–MAX) over all control volumes and control angles is printed for verifying the convergence of the solution. The default value of \( \text{ERROR} = 1. \times 10^{-6} \) is used in this example for satisfying the convergence criteria. When the maximum number of iterations has been reached (ITER = LAST), or when the maximum change of the radiative intensities between two successive iterations falls below the prescribed value (KSTOP = 1), the boundary heat fluxes and incident radiation energy are calculated by calling HFLUX. All grid-related variables and incident radiation energy at control volume nodes are printed by calling PRINT. QTOP represents the net radiative heat flux at the top boundary and QBOT represents the net radiative heat flux at the bottom boundary. \( X(I) \) represents the value of \( X \) at grid location \( I \). Magnitude of QTOP and magnitude of QBOT are printed here for different \( X(I) \). QLEFT represents the net radiative heat flux at the left boundary and QRITE represents the net radiative heat flux at the right boundary. \( Y(J) \) represents the value of \( Y \) at grid location \( J \). Magnitude of QLEFT and magnitude of QRITE are printed here for different \( Y(J) \).

**GAMSOR.** Irregular geometries and radiative equilibrium conditions are set here. Since in our present problem geometry of the enclosure is not irregular and it is in non-radiative equilibrium condition, hence no operation is performed here.

### 2.4-3 Additional Fortran Names

- **DMAX**: maximum of \( \left| I - I_{OLD} \right| / I \) calculated over all control volumes and control angles
- **LAST**: maximum number of iterations
- **ALPHA**: absorption coefficient
- **SIG**: scattering coefficient
- **KISO**: set for type of scattering
- **KPHASE**: set for back scattering
- **STFAN**: Stefan-Boltzmann constant
- **G(I,J)**: incident radiation energy
- **TEM**: non-dimensional temperature of bottom wall
- **QTOP**: net radiative heat flux at top boundary
- **QBOT**: net radiative heat flux at bottom boundary
- **QLEFT**: net radiative heat flux at left boundary
- **QRITE**: net radiative heat flux at right boundary

### 2.4-4 Listing of ADAPT for Example 4

```fortran
C*******************************************************************
SUBROUTINE USER
C*******************************************************************
INCLUDE 'PARAM.FOR'
```
INCLUDE 'COMMON.FOR'
C*******************************************************************
C PROBLEM 4: BLACK, SQUARE ENCLOSURE WITH INHOMOGENOUS MEDIUM
C AND HOT BOTTOM WALL
C*******************************************************************
C ENTRY GRID
C
TITLE(1)=' G '
OPEN(7,FILE='PROB4.DAT')
C
NCVLP=8
NCVLT=4
C
CALL QUAD
C
NCVLX=10
NCVLY=10
C
XL=1.
YL=1.
C
CALL EZGRID
C
RETURN
C*******************************************************************
C ENTRY START
C
LAST=30
KISO=0
KPHASE=2
ALPHA=0.5
SIG=0.5
C
TEM=(1./STFAN)**(1./4.)
C
DO 11 I=2,L1-1
   T(I,1)=TEM
11   CONTINUE
C
RETURN
C*******************************************************************
C ENTRY LC
C
DO 100 J=1,M1
   DO 101 I=1,L1
      IF(X(I).GT.0.3.AND.X(I).LT.0.7.AND.
         Y(J).GT.0.3.AND.Y(J).LT.0.7) THEN
         CAPPA(I,J)=10*ALPHA
         SIGMA(I,J)=10*SIG
      ENDIF
101 CONTINUE
100 CONTINUE
C
RETURN
C*******************************************************************
C ENTRY OUTPUT
C
IF(ITER.EQ.0) WRITE(6,500)
WRITE(6,501) ITER, F(L1/2,M1/2,2,2),DMAX
IF(ITER.EQ.0) WRITE(7,500)
WRITE(7,501) ITER, F(L1/2,M1/2,2,2),DMAX
C
IF(KSTOP.EQ.1.OR.ITER.EQ.LAST) THEN
   CALL HFLUX
   CALL PRINT
C
   WRITE(6,502)
   WRITE(7,502)
C
   DO 510 I=2,L2
      WRITE(6,503) X(I),ABS(QPY(I,M1)-QMY(I,M1)),
                      1                   ABS(QPY(I,1)-QMY(I,1))
      WRITE(7,503) X(I),ABS(QPY(I,M1)-QMY(I,M1)),
                      1                   ABS(QPY(I,1)-QMY(I,1))
   510 CONTINUE
C
   WRITE(6,504)
   WRITE(7,504)
C
   DO 511 J=2,M2
      WRITE(6,503) Y(J),ABS(QPX(1,J)-QMX(1,J)),
                      1                   ABS(QPX(L1,J)-QMX(L1,J))
   511 CONTINUE
C
ENDIF
C
500  FORMAT(/3X,'ITER',8X,'F',12X,'DIFF-MAX'/1X,48('*'))
501  FORMAT(3X,I3,2(3X,1PE12.3))
502  FORMAT(/8X,'Y',9X,'QLEFT',7X,'QRITE'/1X,39('*'))
503  FORMAT(1X,2(F10.5))
504  FORMAT(/8X,'Y',9X,'QLEFT',7X,'QRITE'/1X,39('*'))
C
RETURN
C*******************************************************************
ENTRY GAMSOR
C
RETURN
C
END
C*******************************************************************

2.4-5 Results for Example 4

<table>
<thead>
<tr>
<th>ITER</th>
<th>F</th>
<th>DIFF-MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>1</td>
<td>1.724E-03</td>
<td>1.000E+00</td>
</tr>
<tr>
<td>2</td>
<td>8.829E-03</td>
<td>1.000E+00</td>
</tr>
<tr>
<td>3</td>
<td>1.191E-02</td>
<td>5.533E-01</td>
</tr>
<tr>
<td>4</td>
<td>1.282E-02</td>
<td>2.462E-01</td>
</tr>
<tr>
<td>5</td>
<td>1.307E-02</td>
<td>9.393E-02</td>
</tr>
<tr>
<td>6</td>
<td>1.314E-02</td>
<td>3.203E-02</td>
</tr>
<tr>
<td>7</td>
<td>1.316E-02</td>
<td>1.018E-02</td>
</tr>
<tr>
<td>8</td>
<td>1.316E-02</td>
<td>3.085E-03</td>
</tr>
<tr>
<td>9</td>
<td>1.316E-02</td>
<td>9.060E-04</td>
</tr>
<tr>
<td>10</td>
<td>1.316E-02</td>
<td>2.606E-04</td>
</tr>
<tr>
<td>11</td>
<td>1.316E-02</td>
<td>7.393E-05</td>
</tr>
<tr>
<td>12</td>
<td>1.316E-02</td>
<td>2.086E-05</td>
</tr>
<tr>
<td>13</td>
<td>1.316E-02</td>
<td>5.856E-06</td>
</tr>
<tr>
<td>I</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>X</td>
<td>14</td>
<td>1.316E-02</td>
</tr>
<tr>
<td>XU</td>
<td>15</td>
<td>1.316E-02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>6.50E-01</td>
<td>7.50E-01</td>
<td>8.50E-01</td>
<td>9.50E-01</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>XU</td>
<td>6.00E-01</td>
<td>7.00E-01</td>
<td>8.00E-01</td>
<td>9.00E-01</td>
<td>1.00E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>J</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>0.00E+00</td>
<td>5.00E-02</td>
<td>1.50E-01</td>
<td>2.50E-01</td>
<td>3.50E-01</td>
<td>4.50E-01</td>
<td>5.50E-01</td>
</tr>
<tr>
<td>YV</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>1.00E-01</td>
<td>2.00E-01</td>
<td>3.00E-01</td>
<td>4.00E-01</td>
<td>5.00E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>J</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>6.50E-01</td>
<td>7.50E-01</td>
<td>8.50E-01</td>
<td>9.50E-01</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>YV</td>
<td>6.00E-01</td>
<td>7.00E-01</td>
<td>8.00E-01</td>
<td>9.00E-01</td>
<td>1.00E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>TH</td>
<td>0.00E+00</td>
<td>3.93E-01</td>
<td>1.18E+00</td>
<td>1.96E+00</td>
<td>2.75E+00</td>
<td>3.14E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH</td>
<td>0.00E+00</td>
<td>3.93E-01</td>
<td>1.18E+00</td>
<td>1.96E+00</td>
<td>2.75E+00</td>
<td>3.53E+00</td>
<td>4.32E+00</td>
</tr>
</tbody>
</table>

| M  | 8  | 9  | 10 |
|----|----|----|
| PH | 5.11E+00 | 5.89E+00 | 6.28E+00 |

******            G           ******
-------------------------

<table>
<thead>
<tr>
<th>I</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>12</td>
<td>0.00E+00</td>
<td>1.59E-01</td>
<td>1.45E-01</td>
<td>1.39E-01</td>
<td>1.14E-01</td>
<td>1.07E-01</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>1.47E-01</td>
<td>1.60E-01</td>
<td>1.46E-01</td>
<td>1.40E-01</td>
<td>1.15E-01</td>
<td>1.08E-01</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.80E-01</td>
<td>1.98E-01</td>
<td>1.88E-01</td>
<td>1.78E-01</td>
<td>1.24E-01</td>
<td>1.08E-01</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>2.20E-01</td>
<td>2.45E-01</td>
<td>2.43E-01</td>
<td>2.30E-01</td>
<td>1.28E-01</td>
<td>1.05E-01</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>2.69E-01</td>
<td>3.04E-01</td>
<td>3.16E-01</td>
<td>3.03E-01</td>
<td>1.27E-01</td>
<td>1.02E-01</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>3.32E-01</td>
<td>3.82E-01</td>
<td>4.14E-01</td>
<td>4.07E-01</td>
<td>1.94E-01</td>
<td>1.69E-01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>4.13E-01</td>
<td>4.87E-01</td>
<td>5.44E-01</td>
<td>5.51E-01</td>
<td>3.17E-01</td>
<td>2.98E-01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.19E-01</td>
<td>6.27E-01</td>
<td>7.19E-01</td>
<td>7.54E-01</td>
<td>5.61E-01</td>
<td>5.56E-01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6.42E-01</td>
<td>8.08E-01</td>
<td>9.43E-01</td>
<td>1.03E+00</td>
<td>1.07E+00</td>
<td>1.09E+00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7.63E-01</td>
<td>1.03E+00</td>
<td>1.20E+00</td>
<td>1.29E+00</td>
<td>1.34E+00</td>
<td>1.36E+00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>8.96E-01</td>
<td>1.37E+00</td>
<td>1.56E+00</td>
<td>1.65E+00</td>
<td>1.69E+00</td>
<td>1.71E+00</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.00E+00</td>
<td>2.04E+00</td>
<td>2.04E+00</td>
<td>2.04E+00</td>
<td>2.04E+00</td>
<td>2.05E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>12</td>
<td>1.14E-01</td>
<td>1.39E-01</td>
<td>1.45E-01</td>
<td>1.59E-01</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>1.15E-01</td>
<td>1.40E-01</td>
<td>1.46E-01</td>
<td>1.60E-01</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.24E-01</td>
<td>1.78E-01</td>
<td>1.88E-01</td>
<td>1.98E-01</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1.28E-01</td>
<td>2.30E-01</td>
<td>2.43E-01</td>
<td>2.45E-01</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1.27E-01</td>
<td>3.03E-01</td>
<td>3.16E-01</td>
<td>3.04E-01</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1.94E-01</td>
<td>4.07E-01</td>
<td>4.14E-01</td>
<td>3.82E-01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3.17E-01</td>
<td>5.51E-01</td>
<td>5.44E-01</td>
<td>4.87E-01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.61E-01</td>
<td>7.54E-01</td>
<td>7.19E-01</td>
<td>6.27E-01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.07E+00</td>
<td>1.03E+00</td>
<td>9.43E-01</td>
<td>8.08E-01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.34E+00</td>
<td>1.29E+00</td>
<td>1.20E+00</td>
<td>1.03E+00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.69E+00</td>
<td>1.65E+00</td>
<td>1.56E+00</td>
<td>1.37E+00</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2.05E+00</td>
<td>2.04E+00</td>
<td>2.04E+00</td>
<td>2.04E+00</td>
</tr>
</tbody>
</table>
2.4-6 Discussion of Results

It can be seen that the solution is not converged in one iteration. This is because the medium scatters energy. Hence, at each node source function is unknown. Therefore, at the beginning we are guessing the source function. On the basis of guess value of source function intensity is calculated at the next iteration and it is compared with intensity calculated from previous iteration. This iterative process will continue till the maximum difference in intensity between two successive iterations will be less than or equal to \( \text{EROR} \). Here it is seen that after fifteen iterations solution is converged. In the result lists for the present problem, along \( x \) direction \( X(I) \) and \( XU(I) \) represents the value of \( X \) at grid location \( I \) and the value of \( X \) for the corresponding control volume face. Similarly, along \( y \) direction \( Y(J) \) and \( YV(J) \) represents the value of \( Y \) at grid location \( J \) and value of \( Y \) for the corresponding control volume face. Angular grid contains grid related information in \( \theta \) and \( \phi \)-directions respectively. In the \( \theta \)-direction \( TH(L) \) represents the value of \( \theta \) at the grid location \( L \) and in the \( \phi \)-direction \( PH(M) \) represents the value of \( \phi \) at the grid location \( M \). The final field printout of incident radiation energy shows that the effect of hot bottom wall is to create maximum irradiation (incident radiation energy), \( G(I,J) \) at the bottom of the enclosure which is numerically evaluated by finite volume method as 2.05. The distributions of irradiation inside the medium for homogeneous and inhomogeneous media are shown for a forward scattering phase function (F1) in Fig. 2.4.2. Due to the large absorption coefficient in the middle of the inhomogeneous medium, there is a sharp decrease in the incident radiation. It should be noted that the incident radiation for the homogeneous medium with F1 scattering phase function shown in Figs. 2.3.2 and 2.4.2 are different. This is due to the different spatial and angular grids employed in the two examples. The field printout of boundary heat fluxes shows that the net radiative heat flux at the bottom wall is maximum and is symmetrical about the center of the bottom boundary. Net radiative heat fluxes at left and right boundaries are same and increases along these walls as we come closer to the hot bottom wall.
Effect of inhomogeneous optical properties on incident radiation energy and wall heat fluxes are discussed here.

### 2.5 Black, Square Enclosure with Absorbing-Emitting Medium with a radiative source (Example 5)

#### 2.5-1 Problem Description

The problem under consideration is steady-state radiation in an absorbing-emitting medium with a radiative source. In this example, the radiative source is specified as \( q_{\text{gen}} = 5 \text{ kW/m}^3 \).

The medium intensity can be calculated from

\[
\nabla \cdot q = q_{\text{gen}} = \kappa (4 \pi I_b - G)
\]

(2.5.1)

Once the blackbody intensity is obtained from Eq. (2.5.1), the gas temperature can be calculated using

\[
E_b = \pi I_b = \sigma T_g^4
\]

(2.5.2)

The boundary conditions are specified as

\[
y = 0 \quad T = 1200 \text{ K} \quad (2.5.3a)
\]

\[
y = 1 \text{m} \quad T = 400 \text{ K} \quad (2.5.3b)
\]
Our aim is to calculate the temperature of the medium.

### 2.5-2 Design of ADAPT

**GRID.** The title of the field printout is set to ‘G’ through TITLE (1). The output file (PROB5.DAT) is then specified via OPEN. By default, the angular domains are \(0 \leq \theta \leq \pi\) and \(0 \leq \phi \leq 2\pi\). These are specified in DEFLT using the variables TL and PL respectively. Four control angles represented by NCVLT = 4 are taken in the \(\theta\)-direction and 8 control angles represented by NCVLP = 8 are taken in the \(\phi\)-direction respectively. The default value of POWERT = 1 and POWERP = 1 are used to generate angular grids with uniform \(\Delta \theta\) and \(\Delta \phi\). The boundaries of the control angles are calculated by calling QUAD. The spatial domains are \(0 \leq x \leq 1\) and \(0 \leq y \leq 1\) which are specified through XL and YL respectively. Ten control volumes are used in the \(x\) and \(y\) directions which are represented by NCVLX = 10 and NCVLY = 10 respectively. The default values of POWERX = 1 and POWERY = 1 are used. As a result, an uniform spatial grid is created by calling EZGRID.

**START.** Numerical values of all boundary conditions as given in Eq. (2.5.3) are set here. The maximum number of iterations for the present problem are set as LAST = 30. The value of absorption coefficient \(\kappa\) is taken as ALPHA = 0.5. The radiative heat source is specified as QGEN. Then we fill the boundary temperatures according to Eq. (2.5.3).

**LC.** Since the medium is homogeneous, no addition treatment is done here.

**OUTPUT.** For each iteration (ITER) the temperature of the medium is calculated using Eq. (2.5.2). The value of the temperature, \(T(I, J)\) at the center of the enclosure and maximum difference in magnitude of the intensity from previous iteration, DMAX (which is printed as DIFF-MAX) over all control volumes and control angles are printed for verifying the convergence of the solution. The default value of EROR = 1.E-6 is used in this example for satisfying the convergence criteria. When the maximum number of iterations has been reached (ITER = LAST), or when the maximum change of the radiative intensities between two successive iterations falls below the prescribed value (KSTOP = 1), the boundary heat fluxes and incident radiation energy are calculated by calling HFLUX. All grid-related variables and incident radiation energy at control volume nodes are printed by calling PRINT. The user is of course free to use the temperature as the convergence monitoring variable. You will need to write this algorithm yourself in ENTRY OUTPUT.

**GAMSOR.** The effect of the radiative source given by Eq. (2.5.1) is incorporated in here. The incident radiation \(G\) is calculated by calling HFLUX. The intensities of all internal control volumes are then calculated according to Eq. (2.5.1). Radiative equilibrium is a degenerate case of this example with \(q_{gen} = 0\).

### 2.5-3 Additional Fortran Names
DMAX maximum of \( |I_i - I_{OLD_i}| \) calculated over all control volumes and control angles
LAST maximum number of iterations
ALPHA absorption coefficient
STFAN Stefan-Boltzmann constant
G(I,J) incident radiation energy
QGEN radiative heat source

2.5-4 Listing of ADAPT for Example 5

```fortran
C*******************************************************************
SUBROUTINE USER
C*******************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C*******************************************************************
C PROBLEM 5: BLACK, SQUARE ENCLOSURE WITH THE MEDIUM
C SUBJECT TO A RADIATIVE HEAT SOURCE
C*******************************************************************
CENTRY GRID
C
TITLE(1)=' G,'
OPEN(7,FILE='PROB5.DAT')
C
NCVLP=8
NCVLT=4
C
CALL QUAD
C
NCVLX=10
NCVLY=10
C
XL=1.
YL=1.
C
CALL EZGRID
C
RETURN
C*******************************************************************
CENTRY START
C
LAST=30
ALPHA=0.5
QGEN=5000
C
DO 100 J=2,M2
T(1,J)=800
T(L1,J)=800
100 CONTINUE
C
DO 101 I=2,L2
T(I,1)=1200
T(I,M1)=400
101 CONTINUE
C
RETURN
C*******************************************************************
C```

31
2.5-5 Results for Example 5

<table>
<thead>
<tr>
<th>ITER</th>
<th>T</th>
<th>DIFF-MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>1</td>
<td>4.582E+02</td>
<td>1.000E+00</td>
</tr>
<tr>
<td>2</td>
<td>8.733E+02</td>
<td>9.055E-01</td>
</tr>
<tr>
<td>3</td>
<td>9.327E+02</td>
<td>2.185E-01</td>
</tr>
<tr>
<td>4</td>
<td>9.481E+02</td>
<td>6.207E-02</td>
</tr>
<tr>
<td>5</td>
<td>9.523E+02</td>
<td>1.768E-02</td>
</tr>
<tr>
<td>6</td>
<td>9.534E+02</td>
<td>4.985E-03</td>
</tr>
<tr>
<td>7</td>
<td>9.537E+02</td>
<td>1.394E-03</td>
</tr>
<tr>
<td>8</td>
<td>9.538E+02</td>
<td>3.882E-04</td>
</tr>
<tr>
<td>9</td>
<td>9.539E+02</td>
<td>1.079E-04</td>
</tr>
<tr>
<td>10</td>
<td>9.539E+02</td>
<td>2.996E-05</td>
</tr>
<tr>
<td>11</td>
<td>9.539E+02</td>
<td>8.328E-06</td>
</tr>
<tr>
<td>12</td>
<td>9.539E+02</td>
<td>2.460E-06</td>
</tr>
<tr>
<td>13</td>
<td>9.539E+02</td>
<td>8.153E-07</td>
</tr>
</tbody>
</table>
### 2.5-6 Discussion of Results

<table>
<thead>
<tr>
<th>I</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0.00E+00</td>
<td>5.00E-02</td>
<td>1.50E-01</td>
<td>2.50E-01</td>
<td>3.50E-01</td>
<td>4.50E-01</td>
<td>5.50E-01</td>
</tr>
<tr>
<td>XU</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>1.00E-01</td>
<td>2.00E-01</td>
<td>3.00E-01</td>
<td>4.00E-01</td>
<td>5.00E-01</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Y</td>
<td>0.00E+00</td>
<td>5.00E-02</td>
<td>1.50E-01</td>
<td>2.50E-01</td>
<td>3.50E-01</td>
<td>4.50E-01</td>
<td>5.50E-01</td>
</tr>
<tr>
<td>YV</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>1.00E-01</td>
<td>2.00E-01</td>
<td>3.00E-01</td>
<td>4.00E-01</td>
<td>5.00E-01</td>
</tr>
<tr>
<td>J</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>6.50E-01</td>
<td>7.50E-01</td>
<td>8.50E-01</td>
<td>9.50E-01</td>
<td>1.00E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YV</td>
<td>6.00E-01</td>
<td>7.00E-01</td>
<td>8.00E-01</td>
<td>9.00E-01</td>
<td>1.00E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>TH</td>
<td>0.00E+00</td>
<td>3.93E-01</td>
<td>1.18E+00</td>
<td>1.96E+00</td>
<td>2.75E+00</td>
<td>3.14E+00</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>PH</td>
<td>0.00E+00</td>
<td>3.93E-01</td>
<td>1.18E+00</td>
<td>1.96E+00</td>
<td>2.75E+00</td>
<td>3.53E+00</td>
<td>4.32E+00</td>
</tr>
<tr>
<td>M</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PH</td>
<td>5.11E+00</td>
<td>5.89E+00</td>
<td>6.28E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>I</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>J</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>0.00E+00</td>
<td>8.17E+04</td>
<td>8.44E+04</td>
<td>8.82E+04</td>
<td>9.18E+04</td>
<td>9.39E+04</td>
<td>9.39E+04</td>
<td></td>
</tr>
<tr>
<td>1.05E+05</td>
<td>9.63E+04</td>
<td>9.51E+04</td>
<td>9.72E+04</td>
<td>1.00E+05</td>
<td>1.02E+05</td>
<td>1.02E+05</td>
<td></td>
</tr>
<tr>
<td>1.11E+05</td>
<td>1.09E+05</td>
<td>1.09E+05</td>
<td>1.12E+05</td>
<td>1.16E+05</td>
<td>1.18E+05</td>
<td>1.18E+05</td>
<td></td>
</tr>
<tr>
<td>1.19E+05</td>
<td>1.19E+05</td>
<td>1.22E+05</td>
<td>1.26E+05</td>
<td>1.31E+05</td>
<td>1.33E+05</td>
<td>1.33E+05</td>
<td></td>
</tr>
<tr>
<td>1.26E+05</td>
<td>1.30E+05</td>
<td>1.35E+05</td>
<td>1.41E+05</td>
<td>1.45E+05</td>
<td>1.48E+05</td>
<td>1.48E+05</td>
<td></td>
</tr>
<tr>
<td>1.35E+05</td>
<td>1.41E+05</td>
<td>1.48E+05</td>
<td>1.55E+05</td>
<td>1.60E+05</td>
<td>1.62E+05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.45E+05</td>
<td>1.54E+05</td>
<td>1.64E+05</td>
<td>1.71E+05</td>
<td>1.76E+05</td>
<td>1.78E+05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.55E+05</td>
<td>1.69E+05</td>
<td>1.82E+05</td>
<td>1.90E+05</td>
<td>1.94E+05</td>
<td>1.96E+05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.67E+05</td>
<td>1.87E+05</td>
<td>2.03E+05</td>
<td>2.12E+05</td>
<td>2.17E+05</td>
<td>2.19E+05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.79E+05</td>
<td>2.09E+05</td>
<td>2.29E+05</td>
<td>2.40E+05</td>
<td>2.45E+05</td>
<td>2.48E+05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.90E+05</td>
<td>2.40E+05</td>
<td>2.61E+05</td>
<td>2.72E+05</td>
<td>2.77E+05</td>
<td>2.79E+05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00E+00</td>
<td>2.94E+05</td>
<td>2.97E+05</td>
<td>2.98E+05</td>
<td>2.99E+05</td>
<td>2.99E+05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>I</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>J</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>9.18E+04</td>
<td>8.82E+04</td>
<td>8.44E+04</td>
<td>8.17E+04</td>
<td>0.00E+00</td>
<td></td>
</tr>
<tr>
<td>1.00E+05</td>
<td>9.72E+04</td>
<td>9.51E+04</td>
<td>9.63E+04</td>
<td>1.05E+05</td>
<td></td>
</tr>
<tr>
<td>1.16E+05</td>
<td>1.12E+05</td>
<td>1.09E+05</td>
<td>1.09E+05</td>
<td>1.11E+05</td>
<td></td>
</tr>
<tr>
<td>1.31E+05</td>
<td>1.26E+05</td>
<td>1.22E+05</td>
<td>1.19E+05</td>
<td>1.19E+05</td>
<td></td>
</tr>
<tr>
<td>1.45E+05</td>
<td>1.41E+05</td>
<td>1.35E+05</td>
<td>1.30E+05</td>
<td>1.26E+05</td>
<td></td>
</tr>
<tr>
<td>1.60E+05</td>
<td>1.55E+05</td>
<td>1.48E+05</td>
<td>1.41E+05</td>
<td>1.35E+05</td>
<td></td>
</tr>
<tr>
<td>1.76E+05</td>
<td>1.71E+05</td>
<td>1.64E+05</td>
<td>1.54E+05</td>
<td>1.45E+05</td>
<td></td>
</tr>
<tr>
<td>1.94E+05</td>
<td>1.90E+05</td>
<td>1.82E+05</td>
<td>1.69E+05</td>
<td>1.55E+05</td>
<td></td>
</tr>
<tr>
<td>2.17E+05</td>
<td>2.12E+05</td>
<td>2.03E+05</td>
<td>1.87E+05</td>
<td>1.67E+05</td>
<td></td>
</tr>
<tr>
<td>2.45E+05</td>
<td>2.40E+05</td>
<td>2.29E+05</td>
<td>2.09E+05</td>
<td>1.79E+05</td>
<td></td>
</tr>
<tr>
<td>2.77E+05</td>
<td>2.72E+05</td>
<td>2.61E+05</td>
<td>2.40E+05</td>
<td>1.90E+05</td>
<td></td>
</tr>
<tr>
<td>2.99E+05</td>
<td>2.98E+05</td>
<td>2.97E+05</td>
<td>2.94E+05</td>
<td>0.00E+00</td>
<td></td>
</tr>
</tbody>
</table>
As expected, the solution did not converge in one iteration. This is because the blackbody intensity of the medium is not known. Hence, the source function is unknown. Therefore, at the beginning we are guessing the source function. On the basis of guess value of source function intensity is calculated at the next iteration and it is compared with intensity calculated from previous iteration. This iterative process will continue till the maximum difference in intensity between two successive iterations will be less than or equal to \textit{EROR}. Here it is seen that after thirteen iterations solution is converged. In the result lists for the present problem, along \( x \) direction \( X(I) \) and \( XU(I) \) represents the value of \( X \) at grid location \( I \) and the value of \( X \) for the corresponding control volume face. Similarly, along \( y \) direction \( Y(J) \) and \( YV(J) \) represents the value of \( Y \) at grid location \( J \) and value of \( Y \) for the corresponding control volume face. Angular grid contains grid related information in \( \theta \) and \( \phi \)-directions respectively. In the \( \theta \)-direction \( TH(L) \) represents the value of \( \theta \) at the grid location \( L \) and in the \( \phi \)-direction \( PH(M) \) represents the value of \( \phi \) at the grid location \( M \). Figure 2.5.1 shows the temperature distribution due to the effect of the radiative heat source. The black lines show that locations of the centers of the control volumes for this uniformly divided spatial domain.

![Temperature distribution due to the radiative source.](image)

2.5-7 Final Remarks

This example shows how radiative heat source is modeled using RAT. The next example shows the same problem modeled using non-uniform spatial grids.

2.6 Black, Square Enclosure with Absorbing-Emitting Medium with a radiative source (Example 6)

2.6-1 Problem Description
The physical parameters of this problem are identical to that of Example 5. This problem demonstrates the use of non-uniform spatial grids. The problem under consideration is steady-state radiation in an absorbing-emitting medium with a radiative source. In this example, the radiative source is specified as \( q_{\text{gen}} = 5 \text{kW/m}^3 \). The medium intensity can be calculated from

\[
\nabla \cdot q = q_{\text{gen}} = \kappa(4\pi I_b - G)
\]

(2.6.1)

Once the blackbody intensity is obtained from Eq. (2.6.1), the gas temperature can be calculated using

\[
E_b = \pi I_b = \sigma T_g^4
\]

(2.6.2)

The boundary conditions are specified as

\[
\begin{align*}
y = 0 & \quad T = 1200 \text{ K} \\
y = 1 \text{ m} & \quad T = 400 \text{ K} \\
\text{Others} & \quad T = 800 \text{ K}
\end{align*}
\]

(2.6.3a,b,c)

Our aim is to calculate the temperature of the medium.

### 2.6-2 Design of ADAPT

**GRID.** The title of the field printout is set to ‘G’ through `TITLE(1)`. The output file (`PROB6.DAT`) is then specified via `OPEN`. By default, the angular domains are \(0 \leq \theta \leq \pi\) and \(0 \leq \phi \leq 2\pi\). These are specified in `DEFLT` using the variables `TL` and `PL` respectively. Four control angles represented by `NCVLT = 4` are taken in the \(\theta\)-direction and 8 control angles represented by `NCVLP = 8` are taken in the \(\phi\)-direction respectively. The default value of `POWERT = 1` and `POWERP = 1` are used to generate angular grids with uniform \(\Delta \theta\) and \(\Delta \phi\). The boundaries of the control angles are calculated by calling `QUAD`. The spatial domain is divided into two zones in the \(x\) direction through `NZX = 2`. `XZONE(1)` specifies the length of the first zone to 0.5 which is half the size of the enclosure. The first zone is divided into five control volumes using `NCVX(1) = 5`. The widths of the control volumes are arranged to expand in the positive \(x\) direction using `POWRX(1) = 1.5`. Similar to the first zone, the width of the second zone is set to 0.5 using `XZONE(2)`. Five control volumes fill this space and is set using `NCVX(2) = 5`. The widths of these control volumes contract towards the wall (in the positive \(x\) direction). This is arranged using `POWRX(2) = -1.5`. Note that in this program, positive (> 1.0) `POWRX` implies expanding grids, while negative (< -1) indicates contracting grids. The same magnitude (1.5 in this example) ensures that the meshes are symmetrical about the centerline (in this example). More complete explanation can be found in Patankar (1991). The same concept is used in specifying the grids in the \(y\) direction. The meshes are generated by calling `ZGRID` (not `EZGRID` as in the previous example.)
START. Numerical values of all boundary conditions as given in Eq. (2.6.3) are set here. The maximum number of iterations for the present problem are set as \( \text{LAST} = 30 \). The value of absorption coefficient \( \kappa \) is taken as \( \text{ALPHA} = 0.5 \). The radiative heat source is specified as \( \text{QGEN} \). Then we fill the boundary temperatures according to Eq. (2.6.3).

LC. Since the medium is homogeneous, no addition treatment is done here.

OUTPUT. For each iteration (\( \text{ITER} \)) the temperature of the medium is calculated using Eq. (2.6.2). The value of the temperature, \( T(I,J) \) at the center of the enclosure and maximum difference in magnitude of the intensity from previous iteration, \( \text{DMAX} \) (which is printed as \( \text{DIFF-MAX} \)) over all control volumes and control angles are printed for verifying the convergence of the solution. The default value of \( \text{EROR} = 1. \times 10^{-6} \) is used in this example for satisfying the convergence criteria. When the maximum number of iterations has been reached (\( \text{ITER} = \text{LAST} \)), or when the maximum change of the radiative intensities between two successive iterations falls below the prescribed value (\( \text{KSTOP} = 1 \)), the boundary heat fluxes and incident radiation energy are calculated by calling \( \text{HFLUX} \). All grid-related variables and incident radiation energy at control volume nodes are printed by calling \( \text{PRINT} \). The user is of course free to use the temperature as the convergence monitoring variable. You will need to write this algorithm yourself in \( \text{ENTRY OUTPUT} \).

GAMSOR. The effect of the radiative source given by Eq. (2.6.1) is incorporated in here. The incident radiation \( G \) is calculated by calling \( \text{HFLUX} \). The intensities of all internal control volumes are then calculated according to Eq. (2.6.1). Radiative equilibrium is a degenerate case of this example with \( q_{\text{gen}} = 0 \).

### 2.6-3 Additional Fortran Names

- \( \text{DMAX} \) maximum of \( \left| I_{\text{OLD}} - I \right| \) calculated over all control volumes and control angles
- \( \text{LAST} \) maximum number of iterations
- \( \text{ALPHA} \) absorption coefficient
- \( \text{STFAN} \) Stefan-Boltzmann constant
- \( G(I,J) \) incident radiation energy
- \( \text{QGEN} \) radiative heat source

### 2.6-4 Listing of ADAPT for Example 6

```fortran
C*******************************************************************
SUBROUTINE USER
C*******************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C*******************************************************************
C PROBLEM 6: BLACK, SQUARE ENCLOSURE IN A MEDIUM
C WITH A RADIATIVE HEAT SOURCE
C*******************************************************************
C ENTRY GRID
C```
TITLE(1)=' G '
OPEN(7,FILE='PROB6.DAT')

NCVLP=8
NCVLT=4

CALL QUAD

NZX=2
XZONE(1)=0.5
NCVX(1)=5
POWRX(1)=1.5
XZONE(2)=0.5
NCVX(2)=5
POWRX(2)=-1.5

NZY=2
YZONE(1)=0.5
NCVY(1)=5
POWRY(1)=1.5
YZONE(2)=0.5
NCVY(2)=5
POWRY(2)=-1.5

CALL ZGRID

RETURN

C*******************************************************************
CENTRY START
C
LAST=30
ALPHA=0.5
QGEN=5000

DO 100 J=2,M2
   T(1,J)=800
   T(L1,J)=800
100 CONTINUE

DO 101 I=2,L2
   T(I,1)=1200
   T(I,M1)=400
101 CONTINUE

RETURN

C*******************************************************************
CENTRY LC
C
RETURN

C*******************************************************************
CENTRY OUTPUT
C
DO 200 J=2,M2
   DO 201 I=2,L2
      T(I,J)=(PI*RIB(I,J)/STFAN)**0.25
201 CONTINUE
200 CONTINUE

IF(ITER.EQ.0) WRITE(6,500)
WRITE(6,501) ITER, T(L1/2,M1/2),DMAX
IF(ITER.EQ.0) WRITE(7,500)
WRITE (7, 501) ITER, T(L1/2, M1/2), DMAX

C
IF (KSTOP.EQ.1. OR. ITER.EQ.LAST) THEN
CALL HFLUX
CALL PRINT
ENDIF
C
500 FORMAT (/3X, 'ITER', 8X, 'T', 12X, 'DIFF-MAX' / 1X, 48('*'))
501 FORMAT (3X, I3, 2(3X, 1PE12.3))
C
RETURN
C*******************************************************************
ENTRY GAMSOR
C
CALL HFLUX
C
DO 300 J=2, M2
  DO 301 I=2, L2
    RIB(I, J) = (QGEN/CAPPA(I, J) + G(I, J)) / (4.*PI)
  CONTINUE
300 CONTINUE
C
RETURN
END
C*******************************************************************

2.6-5 Results for Example 6

ITER  T  DIFF-MAX

<table>
<thead>
<tr>
<th>ITER</th>
<th>T</th>
<th>DIFF-MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>1</td>
<td>4.582E+02</td>
<td>1.000E+00</td>
</tr>
<tr>
<td>2</td>
<td>8.755E+02</td>
<td>9.031E-01</td>
</tr>
<tr>
<td>3</td>
<td>9.359E+02</td>
<td>2.205E-01</td>
</tr>
<tr>
<td>4</td>
<td>9.517E+02</td>
<td>6.284E-02</td>
</tr>
<tr>
<td>5</td>
<td>9.561E+02</td>
<td>1.804E-02</td>
</tr>
<tr>
<td>6</td>
<td>9.573E+02</td>
<td>5.131E-03</td>
</tr>
<tr>
<td>7</td>
<td>9.576E+02</td>
<td>1.449E-03</td>
</tr>
<tr>
<td>8</td>
<td>9.577E+02</td>
<td>4.075E-04</td>
</tr>
<tr>
<td>9</td>
<td>9.577E+02</td>
<td>1.143E-04</td>
</tr>
<tr>
<td>10</td>
<td>9.577E+02</td>
<td>3.198E-05</td>
</tr>
<tr>
<td>11</td>
<td>9.577E+02</td>
<td>9.083E-06</td>
</tr>
<tr>
<td>12</td>
<td>9.577E+02</td>
<td>2.553E-06</td>
</tr>
<tr>
<td>13</td>
<td>9.577E+02</td>
<td>8.696E-07</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0.00E+00</td>
<td>2.24E-02</td>
<td>8.56E-02</td>
<td>1.79E-01</td>
<td>2.95E-01</td>
<td>4.29E-01</td>
<td>5.71E-01</td>
</tr>
<tr>
<td>XU</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>4.47E-02</td>
<td>1.26E-01</td>
<td>2.32E-01</td>
<td>3.58E-01</td>
<td>5.00E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>7.05E-01</td>
<td>8.21E-01</td>
<td>9.14E-01</td>
<td>9.78E-01</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>XU</td>
<td>6.42E-01</td>
<td>7.68E-01</td>
<td>8.74E-01</td>
<td>9.55E-01</td>
<td>1.00E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>J</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>0.00E+00</td>
<td>2.24E-02</td>
<td>8.56E-02</td>
<td>1.79E-01</td>
<td>2.95E-01</td>
<td>4.29E-01</td>
<td>5.71E-01</td>
</tr>
<tr>
<td>YV</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>4.47E-02</td>
<td>1.26E-01</td>
<td>2.32E-01</td>
<td>3.58E-01</td>
<td>5.00E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>J</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>7.05E-01</td>
<td>8.21E-01</td>
<td>9.14E-01</td>
<td>9.78E-01</td>
<td>1.00E+00</td>
</tr>
</tbody>
</table>
### 2.6-6 Discussion of Results

As expected, the solution did not converge in one iteration. This is because the blackbody intensity of the medium is not known. Hence, the source function is unknown. Therefore, at the beginning we are guessing the source function. On the basis of guess value of source function intensity is calculated at the next iteration and it is compared with intensity calculated from previous iteration. This iterative process will continue till the maximum difference in intensity between two successive iterations will be less than or equal to $\text{EROR}$. Here it is seen that after thirteen iterations solution is converged. In the result lists for the present problem, along $x$ direction $X(I)$ and $XU(I)$ represents the value of $X$ at grid location $I$ and the value of $X$ for the corresponding control volume face. Similarly, along $y$ direction $Y(J)$ and $YV(J)$ represents the value of $Y$ at grid location $J$ and value of $Y$ for the corresponding control volume face. Angular grid contains grid related information in $\theta$ and $\phi$-directions respectively.
In the $\theta$-direction $\text{TH}(L)$ represents the value of $\theta$ at the grid location $L$ and in the $\phi$-direction $\text{PH}(M)$ represents the value of $\phi$ at the grid location $M$. Figure 2.6.1 shows the temperature distribution due to the effect of the radiative heat source. The black lines show that locations of the centers of the control volumes for this uniformly divided spatial domain. Note that the nodes are finer near the four walls.

![Temperature distribution due to the radiative source.](image)

Fig. 2.6.1 Temperature distribution due to the radiative source.

### 2.6-7 Final Remarks
APPENDIX A

LISTING OF THE INVARIANT PART OF RAT

A.1 Include File “PARAM.FOR”

A.2 Include File “COMMON.FOR”
A.3 Invariant Part of RAT

```fortran
PROGRAM RAT2D
C***********************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C***********************************************************************
CALL DEFLT
CALL GRID
C
CALL SETUP1
C
CALL START
C
CALL SETUP2
C
CALL LC
C
CHECK FOR SCATTERING
C
DO 5 I=2,L2
   DO 6 J=2,M2
      IF(SIGMA(I,J).NE.0.) KSIG=1
   6 CONTINUE
5 CONTINUE
C
IF(KSIG.EQ.1) THEN
   IF(KISO.EQ.1) THEN
      CALL ISOTRP
   ELSE
      CALL ANISO
   ENDIF
ENDIF
C
CALL OUTPUT
DO 10 ITER1=1,LAST
   ITER=ITER1
   IF(KSTOP.EQ.1) STOP
   CALL GAMSOR
   CALL BNDRY
   CALL SMBM
   CALL HEART
   CALL BOUND
   CALL OUTPUT
10 CONTINUE
C
END
```

```
SUBROUTINE DEFLT
C***********************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C***********************************************************************
LAST=5
MODE=1
C
ERROR=1.E-6
SMALL=1.E-10
BIG=1.E20
C
PI=4.0*ATAN(1.0)
PIBY2=PI/2.
PI32=3.*PI/2.
```
PI4 = 4. * PI

XL = 1.0
YL = 1.0

TL = PI
PL = 2. * PI

POWERX = 1.0
POWERY = 1.0
POWERP = 1.0
POWERT = 1.0

NCVLX = 5
NCVLY = 5

DO 10 NZ = 1, NZMX
XZONE(NZ) = 1.
YZONE(NZ) = 1.
NCVX(NZ) = 5
NCVY(NZ) = 5
POWRX(NZ) = 1.
POWRY(NZ) = 1.
10 CONTINUE

EPSJ1 = 1.
EPSI1 = 1.
EPSM1 = 1.
EPSL1 = 1.

ALPHA = 0.0
SIG = 0.0

STFAN = 5.6696E-8
KISO = 1

KBOUND = 0

DO 95 I = 1, NI
   DO 96 J = 1, NJ
      DO 97 L = 1, NT
         DO 98 M = 1, NP
            F(I, J, L, M) = 0.
            BM(I, J, L, M) = 0.
            SM(I, J, L, M) = 0.
            SC(I, J, L, M) = 0.
            SP(I, J, L, M) = 0.
            RIBI1(J, L, M) = 0.
            RIBL1(J, L, M) = 0.
            RIBJ1(I, L, M) = 0.
            RIBM1(I, L, M) = 0.
98 CONTINUE
97 CONTINUE
96 CONTINUE
95 CONTINUE
DO 123 L=2,K2
    DO 124 M=2,J2
        DO 125 LL=2,K2
            PHASE(L,M,LL,MM)=0.0
        125    CONTINUE
    124    CONTINUE
 123  CONTINUE
C RETURN
END

SUBROUTINE SETUP
C***********************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C***********************************************************************
DC(ARGU)=0.5*ARGU-0.25*SIN(2.*ARGU)
SIN2(ARG)=(1.-COS(2.*ARG))/2.0
AM(ARGU)=AMAX1(ARGU,0.0)
C***********************************************************************
ENTRY SETUP1
C L2=L1-1
L3=L2-1
M2=M1-1
M3=M2-1
K2=K1-1
K3=K2-1
J2=J1-1
J3=J2-1
C COME HERE TO CALCULATE X-DIRECTION GRID
C X(1)=XU(2)
    DO 5 I=2,L2
         X(I)=0.5*(XU(I+1)+XU(I))
         XCV(I)=XU(I+1)-XU(I)
    5    CONTINUE
X(L1)=XU(L1)
C COME HERE TO CALCULATE Y-DIRECTION GRID
C Y(1)=YV(2)
    DO 10 J=2,M2
         Y(J)=0.5*(YV(J+1)+YV(J))
         YCV(J)=YV(J+1)-YV(J)
    10   CONTINUE
Y(M1)=YV(M1)
C CALCULATIONS OF CONTROL VOLUME VOLUMES
C DO 15 J=2,M2
    DO 16 I=2,L2
         VOL(I,J)=XCV(I)*YCV(J)
    16    CONTINUE
15 CONTINUE
C CALCULATIONS OF THETA-DIRECTION GRID
C THETA(1)=THETAI(2)
DO 20 L=2,K2
   THETA(L)=0.5*(THETAI(L+1)+THETAI(L))
20 CONTINUE
THETA(K1)=THETAI(K1)

CALCULATIONS OF PHI-DIRECTION GRID

PHI(1)=PHII(2)
DO 30 M=2,J2
   PHI(M)=0.5*(PHII(M+1)+PHII(M))
30 CONTINUE
PHI(J1)=PHII(J1)

CALCULATIONS OF CONTROL ANGLES AND "DIRECTION COSINES"

DO 40 M=2,J2
   PHIM=PHI(M)
   IF(PHIM.LT.PI BY2) MPHI2=M
   IF(PHIM.LT.PI) MPHI3=M
   IF(PHIM.LT.PI/32) MPHI4=M
   TERM1=COS(PHII(M+1))-COS(PHII(M))
   TERM2=SIN(PHII(M+1))-SIN(PHII(M))
   TERM3=PHII(M+1)-PHII(M)
   DO 41 L=2,K2
      IF(THETA(L).LT.PI BY2) LTETA2=L
      TERM4=DC(THETAI(L+1))-DC(THETAI(L))
      TERM5=(SIN2(THETAI(L+1))-SIN2(THETAI(L)))/2.
      DCX(L,M)=TERM2*TERM4
      DCY(L,M)=-TERM1*TERM4
      IF(ABS(DCX(L,M)).LT.1.E-5) DCX(L,M)=0.0
      IF(ABS(DCY(L,M)).LT.1.E-5) DCY(L,M)=0.0
      DOM(L,M)=-(COS(THETAI(L+1))-COS(THETAI(L)))*TERM3
41 CONTINUE
40 CONTINUE

MP2P1=MPHI2+1
MP3P1=MPHI3+1
MP4P1=MPHI4+1
LT2P1=LTETA2+1

CAUTION**** XCV(1),XCV(L1),YCV(1),YCV(M1) ARE NOT USED. ****
XCV(1)=SMALL
XCV(L1)=SMALL
YCV(1)=SMALL
YCV(M1)=SMALL

PRINT 55

2 FORMAT//(//15X,
   1 'COMPUTATION IN TWO-DIMENSIONAL CARTESIAN COORDINATES')
55 FORMAT(14X,56(1H*),//)
C***********************************************************************
ENTRY SETUP2
C
COME HERE TO SPECIFY BOUNDARY EMISSION FROM THE EAST AND WEST WALLS
C
DO 60 J=2,M2
   IF(KBCI1(J).EQ.1) THEN
      DO 62 L=2,K2
         DO 63 M=2,MPHI2
            RIBI1(J,L,M)=STFAN*T(1,J)**4/PI
         63            CONTINUE
         DO 64 M=MP4P1,J2
            RIBI1(J,L,M)=STFAN*T(1,J)**4/PI
         64            CONTINUE
      62         CONTINUE
   ENDIF
   IF(KBCL1(J).EQ.1) THEN
      DO 65 L=2,K2
         DO 66 M=MP2P1,MPHI4
            RIBL1(J,L,M)=STFAN*T(L1,J)**4/PI
         66            CONTINUE
      65         CONTINUE
   ENDIF
60   CONTINUE
C
COME HERE TO SPECIFY BOUNDARY EMISSIONS FROM THE NORTH AND SOUTH WALLS
C
DO 70 I=2,L2
   IF(KBCJ1(I).EQ.1) THEN
      DO 72 L=2,K2
         DO 73 M=2,MPHI3
            RIBJ1(I,L,M)=STFAN*T(I,1)**4/PI
         73            CONTINUE
      72         CONTINUE
   ENDIF
   IF(KBCM1(I).EQ.1) THEN
      DO 75 L=2,K2
         DO 76 M=MP3P1,J2
            RIBM1(I,L,M)=STFAN*T(I,M1)**4/PI
         76            CONTINUE
      75         CONTINUE
   ENDIF
70   CONTINUE
C
COME HERE TO SPECIFY EMISSION FROM THE MEDIUM
C
DO 90 I=2,L2
   DO 91 J=2,M2
      RIB(I,J)=STFAN*T(I,J)**4/PI
      CAPPA(I,J)=ALPHA
      SIGMA(I,J)=SIG
      BETA(I,J)=CAPPA(I,J)+SIGMA(I,J)
91         CONTINUE
90   CONTINUE
C
COME HERE TO UPDATE REFLECTIVITIES
C
RHOJ1=1-EPSJ1
RHOI1=1-EPSI1
RHOM1=1-EPSM1
RHOL1=1-EPSL1
C
RETURN
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE HEART
C***********************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C***********************************************************************
AM(ARGU)=AMAX1(ARGU,0.0)
C***********************************************************************
KSTOP=1
DMAX=-1000.
C
COME HERE TO START FROM THE SOUTH-WEST CORNER (I=2, J=2)
DO 10 J=2,M2
    DO 11 I=2,L2
        DO 12 L=2,K2
            DO 13 M=2,MPHI2
                VOLM=VOL(I,J)*DOM(L,M)
                RNUM=AX(J,L,M)*F(I-1,J,L,M)+
                    AY(I,L,M)*F(I,J-1,L,M)+
                    VOLM*SM(I,J,L,M)
                DENO=AX(J,L,M)+AY(I,L,M)+BM(I,J,L,M)*VOLM
                FOLD=F(I,J,L,M)
                F(I,J,L,M)=RNUM/(DENO+SMALL)
                DIFF=ABS(F(I,J,L,M)-FOLD)/(F(I,J,L,M)+SMALL)
                DMAX=AMAX1(DMAX,DIFF)
                IF(DMAX.GT.EROR) KSTOP=0
            13 CONTINUE
        12 CONTINUE
    11 CONTINUE
10 CONTINUE
C
COME HERE TO START FROM THE SOUTH-EAST CORNER (I=L2, J=2)
DO 20 J=2,M2
    DO 21 I=L2,2,-1
        DO 22 L=2,K2
            DO 23 M=MP2P1,MPHI3
                VOLM=VOL(I,J)*DOM(L,M)
                RNUM=AX(J,L,M)*F(I+1,J,L,M)+
                    AY(I,L,M)*F(I,J-1,L,M)+
                    VOLM*SM(I,J,L,M)
                DENO=AX(J,L,M)+AY(I,L,M)+BM(I,J,L,M)*VOLM
                FOLD=F(I,J,L,M)
                F(I,J,L,M)=RNUM/(DENO+SMALL)
                DIFF=ABS(F(I,J,L,M)-FOLD)/(F(I,J,L,M)+SMALL)
                DMAX=AMAX1(DMAX,DIFF)
                IF(DMAX.GT.EROR) KSTOP=0
            23 CONTINUE
        22 CONTINUE
    21 CONTINUE
20 CONTINUE
C
COME HERE TO START FROM THE NORTH-WEST CORNER (I=2, J=M2)
DO 110 J=M2,2,-1
    DO 111 I=2,L2
        DO 112 L=2,K2
            DO 113 M=MP4P1,J2
                VOLM=VOL(I,J)*DOM(L,M)
                RNUM=AX(J,L,M)*F(I-1,J,L,M)+
                    AY(I,L,M)*F(I,J+1,L,M)+
                    VOLM*SM(I,J,L,M)
                DENO=AX(J,L,M)+AY(I,L,M)+BM(I,J,L,M)*VOLM
                FOLD=F(I,J,L,M)
                F(I,J,L,M)=RNUM/(DENO+SMALL)
DIFF = ABS(F(I, J, L, M) - FOLD) / (F(I, J, L, M) + SMALL)
DMAX = AMAX1(DMAX, DIFF)
IF(DMAX.GT.EROR) KSTOP=0

COME HERE TO START FROM THE NORTH-EAST CORNER (I=L2, J=M2)
DO 120 J=M2,2,-1
   DO 121 I=L2,2,-1
      DO 122 L=2,K2
         DO 123 M=MP3P1,MPHI4
            VOLM=VOL(I,J)*DOM(L,M)
            RNUM=AX(J,L,M)*F(I+1,J,L,M)+
                  AY(I,L,M)*F(I,J+1,L,M)+
                  VOLM*SM(I,J,L,M)
            DENO=AX(J,L,M)+AY(I,L,M)+BM(I,J,L,M)*VOLM
            FOLD=F(I,J,L,M)
            F(I,J,L,M)=RNUM/(DENO+SMALL)
            DIFF=ABS(F(I,J,L,M) - FOLD) / (F(I,J,L,M) + SMALL)
            DMAX=AMAX1(DMAX, DIFF)
            IF(DMAX.GT.EROR) KSTOP=0
         123        CONTINUE
      122     CONTINUE
   121   CONTINUE
120  CONTINUE
C
RETURN
END

SUBROUTINE SMBM
C
DO 11 J=2,M2
   DO 12 I=2,L2
      DO 13 L=2,K2
         DO 14 M=2,J2
            SMSUM=0.
            IF(SIGMA(I,J).NE.0.) THEN
               DO 15 LL=2,K2
                  DO 16 MM=2,J2
                     SMSUM=SMSUM+PHASE(LL,MM,L,M)*DOM(LL,MM)*F(I,J,LL,MM)
16                     CONTINUE
15                  CONTINUE
               14            CONTINUE
13         CONTINUE
12      CONTINUE
11   CONTINUE
C
RETURN
END

-----------------------------------
SUBROUTINE SMBM
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C
DO 11 J=2,M2
   DO 12 I=2,L2
      DO 13 L=2,K2
         DO 14 M=2,J2
            SMSUM=0.
            IF(SIGMA(I,J).NE.0.) THEN
               DO 15 LL=2,K2
                  DO 16 MM=2,J2
                     SMSUM=SMSUM+PHASE(LL,MM,L,M)*DOM(LL,MM)*F(I,J,LL,MM)
16                     CONTINUE
15                  CONTINUE
               14            CONTINUE
13         CONTINUE
12      CONTINUE
11   CONTINUE
C
RETURN
END

-----------------------------------
SUBROUTINE BNDRY
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
DO 100 I=2,L2
   IF(KBCJ1(I).EQ.1) THEN
      CALL WLSUM(I,1,K2,J2,-1,DCY,F,SSUM)
      DO 102 L=2,K2
         DO 103 M=2,MPHI3
            F(I,1,L,M)=EPSJ1*RIBJ1(I,L,M)+RHOJ1*SSUM/PI
            CONTINUE
         CONTINUE
   ELSEIF(KBCJ1(I).EQ.2) THEN
      DO 104 L=2,K2
         DO 105 M=2,MPHI3
            MM=J2-(M-2)
            F(I,1,L,M)=F(I,1,L,MM)
            CONTINUE
         CONTINUE
   ENDIF
   IF(KBCM1(I).EQ.1) THEN
      CALL WLSUM(I,M1,K2,J2,1,DCY,F,RNSUM)
      DO 412 L=2,K2
         DO 413 M=MP3P1,J2
            F(I,M1,L,M)=EPSM1*RIBM1(I,L,M)+RHOM1*RNSUM/PI
            CONTINUE
         CONTINUE
   ELSEIF(KBCM1(I).EQ.2) THEN
      DO 414 L=2,K2
         DO 415 M=MP3P1,J2
            MM=MPHI3-(M-MP3P1)
            F(I,M1,L,M)=F(I,M1,L,MM)
            CONTINUE
         CONTINUE
   ENDIF
C
   IF(KBCI1(J).EQ.1) THEN
      CALL WLSUM(1,J,K2,J2,-1,DCX,F,WSUM)
      DO 501 L=2,K2
         DO 502 M=2,MPHI2
            F(1,J,L,M)=EPSI1*RIBI1(J,L,M)+RHOI1*WSUM/PI
            CONTINUE
         CONTINUE
   ELSEIF(KBCI1(J).EQ.2) THEN
      DO 504 L=2,K2
         DO 505 M=2,MPHI2
            MM=MPHI3-(M-2)
            F(1,J,L,M)=F(1,J,L,MM)
            CONTINUE
         CONTINUE
   ENDIF
C
DO 500 J=2,M2
IF(KBCL1(J).EQ.1) THEN  
   CALL WLSUM(L1,J,K2,J2,1,DCX,F,ESUM)  
   DO 508 L=2,K2  
       DO 509 M=MP2P1,MPHI4  
           F(L1,J,L,M)=EPSL1*RIBL1(J,L,M)+RHOL1*ESUM/PI  
   CONTINUE
   508  CONTINUE  
   ELSEIF(KBCL1(J).EQ.2) THEN  
       DO 1510 L=2,K2  
           DO 1511 M=MP2P1,MPHI3  
               MM=MPHI2-(M-MP2P1)  
               F(L1,J,L,M)=F(L1,J,L,MM)  
       CONTINUE  
           DO 1514 M=MP3P1,MPHI4  
               MM=J2-(M-MP3P1)  
               F(L1,J,L,M)=F(L1,J,L,MM)  
       CONTINUE
   1510  ENDIF
   500  CONTINUE
C
   RETURN
END

SUBROUTINE BOUND
C
   INCLUDE 'PARAM.FOR'
   INCLUDE 'COMMON.FOR'
C
   DO 10 J=2,M2  
       DO 11 L=2,K2  
           DO 12 M=MP2P1,MPHI4  
               F(1,J,L,M)=F(2,J,L,M)  
       CONTINUE
       DO 13 M=2,MPHI2  
           F(L1,J,L,M)=F(L2,J,L,M)  
       CONTINUE
       DO 14 M=MP4P1,J2  
           F(L1,J,L,M)=F(L2,J,L,M)  
       CONTINUE
11  CONTINUE
10  CONTINUE
C
   DO 20 I=2,L2  
       DO 21 L=2,K2  
           DO 22 M=2,MPHI3  
               F(I,M1,L,M)=F(I,M2,L,M)  
           CONTINUE
           DO 23 M=MP3P1,J2  
               F(I,1,L,M)=F(I,2,L,M)  
       CONTINUE
21  CONTINUE
20  CONTINUE
C
   RETURN
END

SUBROUTINE WLSUM(I,J,K2,J2,INDX,DC,FC,SUM)
C
   INCLUDE 'PARAM.FOR'
   DIMENSION DC(NT,NP),FC(NI,NJ,NT,NP)
C
   SUM=0.
IF(INDX.GT.0) THEN
   DO 10 L=2,K2
      DO 20 M=2,J2
         SUM=SUM+FC(I,J,L,M)*AMAX1(DC(L,M),0.0)
   20      CONTINUE
10      CONTINUE
ELSE
      DO 30 L=2,K2
         DO 40 M=2,J2
            SUM=SUM+FC(I,J,L,M)*AMAX1(-DC(L,M),0.0)
   40      CONTINUE
30      CONTINUE
ENDIF
C
RETURN
END

SUBROUTINE PHASEF
C
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
DIMENSION PSUB(NP,NPSUB),TSUB(NT,NPSUB),
1 PSUBI(NP,NPSUB),TSUBI(NT,NPSUB)

ENTRY ISOTRP
C
DO 10 L=2,K2
   DO 20 M=2,J2
      DO 30 LL=2,K2
         DO 40 MM=2,J2
            PHASE(LL,MM,L,M)=1.0
   40            CONTINUE
30         CONTINUE
20      CONTINUE
10   CONTINUE
C
RETURN
END

ENTRY ANISO
C
nsub=5
CALL PCOEF
C
DO 2000 M=2,J2
   PLOW=PHII(M)
   PHIGH=PHII(M+1)
   CALL PHISUB(PLOW,PHIGH,nsub,PSUB,PSUBI,M)
   DPHI=(PHIGH-PLOW)/nsub
   DO 2100 L=2,K2
      TLOW=THETAI(L)
      THIGH=THETAI(L+1)
      CALL TETASUB(TLOW,THIGH,DPHI,nsub,TSUB,TSUBI,L)
   2100 CONTINUE
2000 CONTINUE
C
DO 2200 M=2,J2
   PHIL=PHI(M)
   DO 2210 L=2,K2
      TETAK=THETA(L)
      DO 2300 MM=2,J2
         DO 2310 LL=2,K2
            SUMLL=0.0
            SUMM=SUM+FC(I,J,L,M)*AMAX1(DC(L,M),0.0)
   2310        CONTINUE
2300 CONTINUE
2210 CONTINUE
2200 CONTINUE
C
PLSSUB=PSUB(M,MS)
TERMA=PSUBI(M,MS+1)-PSUBI(M,MS)
DO 2230 LS=2,NSUB+1
   TKSSUB=TSUB(L,LS)
   TERMB=COS(TSUBI(L,LS+1))-COS(TSUBI(L,LS))
   DOMA=-TERMA*TERMB
   XMU=SIN(TKSSUB)*COS(PLSSUB)
   PSI=SIN(TKSSUB)*SIN(PLSSUB)
   ETA=COS(TKSSUB)
   DO 2410 MMS=2,NSUB+1
      PLSUB=PSUB(MM,MMS)
      TERM1=PSUBI(MM,MMS+1)-PSUBI(MM,MMS)
      DO 2420 LLS=2,NSUB+1
         TKSUB=TSUB(LL,LLS)
         XMUL=SIN(TKSUB)*COS(PLSUB)
         PSIIL=SIN(TKSUB)*SIN(PLSUB)
         ETAL=COS(TKSUB)
         ANG=XMU*XMUL+PSI*PSIIL+ETA*ETAL
         SUM=1.0
         DO 2500 MO=1,MORDER
            SUM=SUM+A(MO)*PLCOS(MO,ANG,SMALL)
         2500             CONTINUE
         TERM2=COS(TSUBI(LL,LLS+1))-COS(TSUBI(LL,LLS))
         DOMS=-TERM1*TERM2
         SUMLL=SUMLL+SUM*DOMS*DOMA
   2420                  CONTINUE
   2410             CONTINUE
   2230          CONTINUE
   2220       CONTINUE
   PHASE(LL,MM,L,M)=SUMLL/DOM(LL,MM)/DOM(L,M)
2210          CONTINUE
   2200       CONTINUE
C
   CALL PNORM
   RETURN
END

SUBROUTINE PHISUB(PLOW,PHIGH,NSUB,PSUB,PSUBI,L)
   INCLUDE 'PARAM.FOR'
   DIMENSION PSUBI(NP,NPSUB),PSUB(NP,NPSUB)
   NSUBK=NSUB+2
   PSUBI(L,2)=PLOW
   PSUBI(L,NSUBK)=PHIGH
   DPHI=(PHIGH-PLOW)/FLOAT(NSUB)
   DO 10 LL=3,NSUBK-1
      PSUBI(LL,LL)=PSUBI(LL,LL-1)+DPHI
   10 CONTINUE
C
   DO 20 LL=2,NSUBK-1
      PSUB(LL,LL)=0.5*(PSUBI(LL,LL)+PSUBI(LL,LL+1))
20 CONTINUE
   RETURN
END
SUBROUTINE TETASUB(TLOW, THIGH, DPHI, NSUB, TSUB, TSUBI, K)

INCLUDE 'PARAM.FOR'

DIMENSION TSUBI(NT, NPSUB), TSUB(NT, NPSUB)

NSUBL=NSUB+2
TSUBI(K,2)=TLOW
TSUBI(K,NSUBL)=THIGH
DPHI=(THIGH-TLOW)/FLOAT(NSUB)
DO 10 KK=3, NSUBL-1
    TSUBI(K,KK)=TSUBI(K,KK-1)+DPHI
10 CONTINUE

TSUB(K,1)=TSUBI(K,2)
DO 20 KK=2, NSUBL-1
    TSUB(K,KK)=0.5*(TSUBI(K,KK)+TSUBI(K,KK+1))
20 CONTINUE

TSUB(K,NSUBL)=TSUBI(K,NSUBL)

RETURN
END

SUBROUTINE PCOEF

C******************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
COMPLEX IOR
C**********************************************************************

C   KPHASE     PHASE FUNCTION
C -------------------------------
C
C     2             F1
C     3             F2
C     6             B1
C     7             B2
C
C
IF(KPHASE.EQ.1) IPH=0
IF(KPHASE.EQ.2) IPH=2
IF(KPHASE.EQ.3) IPH=4
IF(KPHASE.EQ.4) IPH=5
IF(KPHASE.EQ.5) IPH=6
IF(KPHASE.EQ.6) IPH=10
IF(KPHASE.EQ.7) IPH=20
C--- FORWARD SCATTERING PHASE FUNCTIONS.
    IF(IPH.EQ.100) THEN
        MORDER =1
    ENDIF
    A(0) =1.0
    A(1) =3.*0.30

C-----
    IF(IPH.EQ.0) THEN
        XSIZ = 999.
        IOR = (999.,999.)
        QSCA = 999.
        QABS = 999.
        MORDER = 26
    ENDIF
    A(0) = 1.0
    A(1) = 2.78197
A(2) = 4.25856
A(3) = 5.38653
A(4) = 6.19015
A(5) = 6.74492
A(6) = 7.06711
A(7) = 7.20999
A(8) = 7.20063
A(9) = 7.03629
A(10) = 6.76587
A(11) = 6.35881
A(12) = 5.83351
A(13) = 5.22997
A(14) = 4.47918
A(15) = 3.69000
A(16) = 2.81577
A(17) = 1.92305
A(18) = 1.11502
A(19) = 0.50766
A(20) = 0.20927
A(21) = 0.07138
A(22) = 0.02090
A(23) = 0.00535
A(24) = 0.00120
A(25) = 0.00024
A(26) = 0.00004

ENDIF
C -----.GetComponent("__after__")

IF(IPH.EQ.1) THEN
  XSIZE =10.0
  IOR = (1.33,0.0)
  QSCA=2.20654869
  QABS=0.0
  GFAC=0.71245915
  MORDER=20
  C
  A(0)=1.0000000
  A(1)=2.1373777
  A(2)=2.9336057
  A(3)=2.8347003
  A(4)=2.7405264
  A(5)=2.6281443
  A(6)=2.5408404
  A(7)=2.6078286
  A(8)=2.7392752
  A(9)=2.9413168
  A(10)=3.2081311
  A(11)=3.4706223
  A(12)=3.7823384
  A(13)=4.1312394
  A(14)=4.2848716
  A(15)=4.7362566
  A(16)=4.4471574
  A(17)=4.3439150
  A(18)=3.3834202
  A(19)=2.2265594
  A(20)=1.3012373
ENDIF
C ----- GetComponent("__after__")

IF(IPH.EQ.2) THEN
  XSIZE = 5.0
  IOR = (1.33,0.0)
  QSCA = 3.59103251
  QABS = 0.0
  GFAC = 0.84534043

54
MORDER =  12

C
A(0) =   1.0
A(1) =   2.5360217
A(2) =   3.5654900
A(3) =   3.9797626
A(4) =   4.0029206
A(5) =   3.6640084
A(6) =   3.0160117
A(7) =   2.2330437
A(8) =   1.3025078
A(9) =   0.5346286
A(10) =  0.2013563
A(11) =  0.0547964
A(12) =  0.0109929
ENDIF

C-----

IF(IPH.EQ.3) THEN
XSIZE =3.0
IOR =(1.33,0.0)
QSCA=1.75339794
QABS=0.0
GFAC=0.78320068
MORDER=9
C
A(0) =   1.0000000
A(1) =   2.3496020
A(2) =   2.7382560
A(3) =   2.3145776
A(4) =   1.3907945
A(5) =   0.5534959
A(6) =   0.1744258
A(7) =   0.0401137
A(8) =   0.0069153
A(9) =   0.0008899
ENDIF

C-----

IF(IPH.EQ.4) THEN
XSIZE =  2.0
IOR   = (1.33,0.0)
QSCA  =  0.71294856
QABS  =  0.0
MORDER =  8
C
A(0) =   1.0
A(1) =   2.0091653
A(2) =   1.5633900
A(3) =   0.6740690
A(4) =   0.2221484
A(5) =   0.0472529
A(6) =   0.0067132
A(7) =   0.0006743
A(8) =   0.0000494
ENDIF

C-----

IF(IPH.EQ.5) THEN
XSIZE =1.0
IOR =(1.33,0.0)
QSCA=9.39240903E-02
QABS=0.0
GFAC=0.18451715
MORDER=6
C
A(0) =   1.0000000
A(1) =   0.5535514
A(2) =   0.5600496
A(3) =   0.1157242
A(4) =   0.0107823
ENDIF
A(5) = 0.0005812
A(6) = 0.0000230
ENDIF

C-----
IF(IPH.EQ.6) THEN
XSIZE = 999.
IOR   = (999.,999.)
QSCA  = 999.
QABS  = 999.
MORDER = 2
C   A(0) = 1.0
A(1) = 1.2
A(2) = 0.5
ENDIF
C

C-----
IF(IPH.EQ.7) THEN
XSIZE = 999.
IOR   = (999.,999.)
QSCA  = 999.
QABS  = 999.
MORDER = 1
C   A(0) = 1.0
A(1) = 1.0
ENDIF
C
C-----------------------------------------------------------
C--- BACKWARD SCATTERING PHASE FUNCTIONS.
IF(IPH.EQ.10) THEN
XSIZE = 999.
IOR   = (999.,999.)
QSCA  = 999.
QABS  = 999.
MORDER = 5
C   A(0) = 1.0
A(1) = -0.56524
A(2) = 0.29783
A(3) = 0.08571
A(4) = 0.01003
A(5) = 0.00063
ENDIF

C-----
IF(IPH.EQ.20) THEN
XSIZE = 999.
IOR   = (999.,999.)
QSCA  = 999.
QABS  = 999.
MORDER = 2
C   A(0) = 1.0
A(1) = -1.2
A(2) = 0.5
ENDIF
C
C-----
IF(IPH.EQ.30) THEN
XSIZE = 999.
IOR   = (999.,999.)
QSCA  = 999.
QABS  = 999.
MORDER = 1
C   A(0) = 1.0
A(1) = -1.0
A(1) = -1.0
ENDIF
C
RETURN
SUBROUTINE PNORM
C******************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
DIMENSION PHOLD(NT,NP,NT,NP)
C***********************************************************************
DO 10 L=2,K2
   DO 20 M=2,J2
      SUM=0.0
      DO 30 LL=2,K2
         DO 40 MM=2,J2
            SUM=SUM+PHASE(LL,MM,L,M)*DOM(LL,MM)
         40 CONTINUE
      30 CONTINUE
      C
      FACT=SUM/(4.*PI)
      C
      DO 50 LL=2,K2
         DO 60 MM=2,J2
            PHOLD(LL,MM,L,M)=PHASE(LL,MM,L,M)
            PHASE(LL,MM,L,M)=PHASE(LL,MM,L,M)/(FACT+SMALL)
         60 CONTINUE
      50 CONTINUE
20   CONTINUE
10   CONTINUE
C
RETURN
END

FUNCTION PLCOS(L,X,SMALL)
C     COMPUTES THE LEGENDRE POLYNOMIAL P-SUB-L.
C     X IS IN THE RANGE FROM -1 TO +1.
C     MODIFIED FROM "NUMERICAL RECIPES, THE ART OF SCIENTIFIC COMPUTING' BY
C     W. H. PRESS, B. P. FLANNERY, S. A. TEUKOLSKY, W. T. VETERLING
C ORIGINAL FUNCTION NAME = PLGNDR
cc IF(L.LT.1.OR.ABS(X).GT.1.+SMALL) PRINT*, 'CHECK ANG',X
C
PMM=1.
PMM1=X
IF(L.EQ.1) THEN
   PLCOS=PMMP1
ELSE
   DO 12 LL=2,L
      PLL=(X*(2*LL-1)*PMMP1-(LL-1)*PMM)/(LL)
      PMM=PMMP1
      PMMP1=PLL
12 CONTINUE
   PLCOS=PLL
ENDIF
RETURN
END

SUBROUTINE SUPPLY
C******************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C******************************************************************
7 FORMAT(/1X,6(1H*),3X,A18,3X,6(1H*)/9X,20(1H-))
8 FORMAT('/', I =',I6,6I9)
ENTRY EZGRID

CONSTRUCT THE X-DIRECTION GRID
L1=NCVLX+2
XU(1)=XU(2)
XU(L1)=XL+XU(2)
L2=L1-1
FCVLX=FLOAT(NCVLX)
DO 21 I=3,L2
DD=FLOAT(I-2)/FCVLX
IF(POWERX.GT.0.) THEN
   XU(I)=XL*DD**POWERX+XU(2)
ELSE
   XU(I)=XL*(1.-(1.-DD)**(-POWERX))+XU(2)
ENDIF
21 CONTINUE

CONSTRUCT THE Y-DIRECTION GRID
M1=NCVLY+2
YV(2)=0.
YV(M1)=YL
M2=M1-1
FCVLY=FLOAT(NCVLY)
DO 31 J=3,M2
DD=FLOAT(J-2)/FCVLY
IF(POWERY.GT.0.) THEN
   YV(J)=YL*DD**POWERY
ELSE
   YV(J)=YL*(1.-(1.-DD)**(-POWERY))
ENDIF
31 CONTINUE

RETURN

ENTRY ZGRID

CONSTRUCT THE GRID ZONE-BY-ZONE

CONSIDER THE X DIRECTION
XU(1)=XU(2)
I2=2
DO 1101 NZ=1,NZX
FCVLX=FLOAT(NCVX(NZ))
ILAST=I2
I1=ILAST+1
I2=ILAST+NCVX(NZ)

1101 CONTINUE
DO 1101 I=I1,I2
DD=FLOAT(I-ILAST)/FCVLX
IF(POWRX(NZ).GT.0.) THEN
  XU(I)=XU(ILAST)+XZONE(NZ)*DD**POWRX(NZ)
ELSE
  XU(I)=XU(ILAST)+XZONE(NZ)*(1.-(1.-DD)**(-POWRX(NZ)))
ENDIF
1101 CONTINUE
L1=I2
C

CONSIDER THE Y DIRECTION
C
YV(2)=0.
JJ2=2
DO 1100 NZ=1,NZY
FCVLY=FLOAT(NCVY(NZ))
JLAST=JJ2
JJ1=JLAST+1
JJ2=JLAST+NCVY(NZ)
DO 1100 J=JJ1,JJ2
DD=FLOAT(J-JLAST)/FCVLY
IF(POWRY(NZ).GT.0.) THEN
  YV(J)=YV(JLAST)+YZONE(NZ)*DD**POWRY(NZ)
ELSE
  YV(J)=YV(JLAST)+YZONE(NZ)*(1.-(1.-DD)**(-POWRY(NZ)))
ENDIF
1100 CONTINUE
M1=JJ2
RETURN
C

CONSTRUCT THE PHI-DIRECTION GRID
C
J1=NCVLP+2
PHII(2)=0.
PHII(J1)=PL
J2=J1-1
FCVLP=FLOAT(NCVLP)
DO 4 M=3,J2
DD=FLOAT(M-2)/FCVLP
IF(POWERP.GT.0.) THEN
  PHII(M)=PL*DD**POWERP
ELSE
  PHII(M)=PL*(1.-(1.-DD)**(-POWERP))
ENDIF
4    CONTINUE
C

CONSTRUCT THE THETA-DIRECTION GRID
C
K1=NCVLT+2
THETAI(2)=0.
THETAI(K1)=TL
K2=K1-1
FCVLT=FLOAT(NCVLT)
DO 5 L=3,K2
DD=FLOAT(L-2)/FCVLT
IF(POWERT.GT.0.) THEN
  THETAI(L)=TL*DD**POWERT
ELSE
  THETAI(L)=TL*(1.-(1.-DD)**(-POWERT))
ENDIF
5    CONTINUE
C
RETURN
ENTRY PRINT

C
PRINT 50
WRITE(7,50)
IEND=0
301 IF(IEND.EQ.L1) GO TO 310
IBEG=IEND+1
IEND=IEND+7
IEND=MIN0(IEND,L1)
PRINT 50
WRITE(7,50)
PRINT 51,(I,I=IBEG,IEND)
WRITE(7,51)(I,I=IBEG,IEND)
IF(MODE.EQ.3) GO TO 302
PRINT 52,(X(I),I=IBEG,IEND)
PRINT 62,(XU(I),I=IBEG,IEND)
WRITE(7,52)(X(I),I=IBEG,IEND)
WRITE(7,62)(XU(I),I=IBEG,IEND)
GO TO 303
302 PRINT 53,(X(I),I=IBEG,IEND)
WRITE(7,53)(X(I),I=IBEG,IEND)
303 GO TO 301
310 JEND=0
PRINT 50
WRITE(7,50)
311 IF(JEND.EQ.M1) GO TO 320
JBEG=JEND+1
JEND=JEND+7
JEND=MIN0(JEND,M1)
PRINT 50
PRINT 54,(J,J=JBEG,JEND)
PRINT 55,(Y(J),J=JBEG,JEND)
PRINT 55,(YV(J),J=JBEG,JEND)
WRITE(7,50)
WRITE(7,54)(J,J=JBEG,JEND)
WRITE(7,55)(Y(J),J=JBEG,JEND)
WRITE(7,65)(YV(J),J=JBEG,JEND)
GO TO 311
320 JEND=0
PRINT 50
WRITE(7,50)
331 IF(JEND.EQ.K1) GO TO 340
JBEG=JEND+1
JEND=JEND+7
JEND=MIN0(JEND,K1)
PRINT 50
PRINT 58,(J,J=JBEG,JEND)
PRINT 59,(THETA(J),J=JBEG,JEND)
WRITE(7,50)
WRITE(7,58)(J,J=JBEG,JEND)
WRITE(7,59)(THETA(J),J=JBEG,JEND)
GO TO 331
340 JEND=0
PRINT 50
WRITE(7,50)
341 IF(JEND.EQ.J1) GO TO 350
JBEG=JEND+1
JEND=JEND+7
JEND=MIN0(JEND,J1)
PRINT 50
PRINT 60,(J,J=JBEG,JEND)
PRINT 61, (PHI(J), J=JBEG, JEND)
WRITE(7,50)
WRITE(7,60) (J, J=JBEG, JEND)
WRITE(7,61) (PHI(J), J=JBEG, JEND)
GO TO 341
350 CONTINUE
C
WRITE(6,7) TITLE(1)
WRITE(7,7) TITLE(1)
IBEG=1
JBEG=1
IEND=1
JEND=m1
IREP=(IEND-IBEG+7)/7
DO 551 KP=1,IREP
   INCR=MIN(6, IEND-IBEG)
   ISTOP=IBEG+INCR
   WRITE(6,8) (I, I=IBEG, ISTOP)
   WRITE(6,9)
   WRITE(7,8) (I, I=IBEG, ISTOP)
   WRITE(7,9)
   DO 552 J=JEND, JBEG, -1
      WRITE(6,40) J, (G(I,j), I=IBEG, ISTOP)
      WRITE(7,40)
   552 CONTINUE
   IBEG=ISTOP+1
551 CONTINUE
C
RETURN
END
CCCCCCCCCC
SUBROUTINE HFLUX
C******************************************************************
INCLUDE 'PARAM.FOR'
INCLUDE 'COMMON.FOR'
C******************************************************************
DO 500 I=1, L1
   DO 510 J=1, M1
      SQPY=0.
      SQMY=0.
      SQPX=0.
      SQMX=0.
   DO 520 L=2, K2
      DO 530 M=2, J2
         ADCY=ABS(DCY(L,M))
         ADCX=ABS(DCX(L,M))
         IF(I.GT.1.AND.I.LT.L1) THEN
            IF(DCY(L,M).GT.0.0) THEN
               SQPY=SQPY+ADCY*F(I,J,L,M)
            ELSE
               SQMY=SQMY+ADCY*F(I,J,L,M)
            ENDIF
         ENDIF
      530 END
      520 CONTINUE
   510 CONTINUE
   SQPY(I,J)=SQPY
   SQMY(I,J)=SQMY
C
530 CONTINUE
520 CONTINUE
QPY(I,J)=SQPY
QMY(I,J)=SQMY
CCCCCCCCCC
QPX(I,J)=SQPX
QMX(I,J)=SQMX

DO 540 I=1,L1
  DO 550 J=1,M1
    GSUM=0.
    DO 560 L=2,K2
      DO 570 M=2,J2
        GSUM=GSUM+F(I,J,L,M)*DOM(L,M)
      570 CONTINUE
    560 CONTINUE
    G(I,J)=GSUM
  551 CONTINUE
  550 CONTINUE
540 CONTINUE

DO 580 J=1,M1
  IF(KBCI1(J).EQ.2) THEN
    QPY(1,J)=QPY(2,J)
    QMY(1,J)=QMY(2,J)
    G(1,J)=G(2,J)
  ENDIF
  IF(KBCL1(J).EQ.2) THEN
    QPY(L1,J)=QPY(L2,J)
    QMY(L1,J)=QMY(L2,J)
    G(L1,J)=G(L2,J)
  ENDIF
580 CONTINUE

DO 600 I=1,L1
  IF(KBCJ1(I).EQ.2) THEN
    QPX(I,1)=QPX(I,2)
    QMX(I,1)=QMX(I,2)
    G(I,1)=G(I,2)
  ENDIF
  IF(KBCM1(I).EQ.2) THEN
    QPX(I,M1)=QPX(I,M2)
    QMX(I,M1)=QMX(I,M2)
    G(I,M1)=G(I,M2)
  ENDIF
600 CONTINUE

RETURN
END
# FUNCTIONS OF VARIOUS SUBROUTINES AND ENTRIES

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFLT</td>
<td>default values are set here.</td>
</tr>
<tr>
<td>GRID*</td>
<td>geometry of the problem is specified here.</td>
</tr>
<tr>
<td>SETUP1</td>
<td>initial setup.</td>
</tr>
<tr>
<td>START*</td>
<td>properties and temperatures are set here.</td>
</tr>
<tr>
<td>SETUP2</td>
<td>final setup.</td>
</tr>
<tr>
<td>LC*</td>
<td>inhomogeneous medium is set here.</td>
</tr>
<tr>
<td>ISOTRP</td>
<td>set the phase function to the isotropic phase function.</td>
</tr>
<tr>
<td>ANISO</td>
<td>calculates anisotropic phase function.</td>
</tr>
<tr>
<td>OUTPUT*</td>
<td>output routine.</td>
</tr>
<tr>
<td>GAMSOR*</td>
<td>irregular geometries and radiative equilibrium conditions are set here.</td>
</tr>
<tr>
<td>HEART</td>
<td>main solution loop.</td>
</tr>
<tr>
<td>PRINT</td>
<td>print grid related variables and incident radiation energy.</td>
</tr>
<tr>
<td>EZGRID</td>
<td>usually called in entry grid to construct the spatial grids.</td>
</tr>
<tr>
<td>QUAD</td>
<td>usually called in entry grid to construct the angular grids.</td>
</tr>
<tr>
<td>HFLUX</td>
<td>heat fluxes and incident radiation energy are calculated here.</td>
</tr>
</tbody>
</table>

* Denotes entries in the ADAPT subroutine.
LAST : maximum number of iterations.
MODE : indicator for coordinate system.
ERROR : convergence criteria.
SMALL : a small number.
BIG : a big number.
PI : \( \pi \)
PIBY2 : \( \pi/2 \)
PI32 : \( 3\pi/2 \)
PI4 : \( 4\pi \)
TL : \( \theta \)-direction length of the calculation domain = \( \pi \)
PL : \( \phi \)-direction length of the calculation domain = \( 2\pi \)
XL : \( x \)-direction length of the calculation domain
YL : \( y \)-direction length of the calculation domain
POWERT : non-uniformity index for the \( \theta \)-direction grid.
POWERP : non-uniformity index for the \( \phi \)-direction grid.
POWEX : non-uniformity index for the \( x \)-direction grid.
POWERY : non-uniformity index for the \( y \)-direction grid.
NCVLP : number of \( \phi \)-direction control volume widths in the domain.
Presently, please use NCVLP that is divisible by 4.
NCVLT : number of \( \theta \)-direction control volume widths in the domain.
Presently, please use NCVLP that is divisible by 2.
NCVX : number of \( x \)-direction control volume widths in the domain.
NCVY : number of \( y \)-direction control volume widths in the domain.
XZONE(NZ) : \( x \)-direction length of a zone.
YZONE(NZ) : \( y \)-direction length of a zone.
NCVX(NZ) : number of \( x \)-direction control volume widths in a zone.
NCVY(NZ) : number of \( y \)-direction control volume widths in a zone.
POWEX(NZ) : non-uniformity index for the \( x \)-direction grid in a zone.
POWERY(NZ) : non-uniformity index for the \( y \)-direction grid in a zone.
EPSJ1 : emissivity of the south wall.
EPM1 : emissivity of the north wall.
EPSL1 : emissivity of the east wall.
EPSI1 : emissivity of the west wall.
RHOJ1 : \( 1. - \) EPSJ1.
RHOM1 : \( 1. - \) EPSM1.
RHOL1 : 1 - EPSL1.
RHOL1 : 1 - EPSL1.
ALPHA : absorption coefficient, \( \alpha \).
SIG : scattering coefficient.
STFAN : Stefan-Boltzmann constant, \( \sigma \).
KISO : = 1; isotropic scattering.
KISO : = 0; anisotropic scattering.
L1 : maximum number of \( x \)-direction grid locations = NCVLX + 2.
M1 : maximum number of \( y \)-direction grid locations = NCVLY + 2.
K1 : maximum number of \( \theta \)-direction grid locations = NCVLT + 2.
K1 : maximum number of \( \phi \)-direction grid locations = NCVLP + 2.
FCOLJ1() : intensity of the collimated beam at the south wall.
F() : actual nodal intensity.
FOLD() : actual nodal intensity from the previous iteration.
RIBI1() : \( \pi \sigma^4 / \lambda \) at the west wall.
RIBL1() : \( \pi \sigma^4 / \lambda \) at the east wall.
RIBJ1() : \( \pi \sigma^4 / \lambda \) at the south wall.
RIBM1() : \( \pi \sigma^4 / \lambda \) at the north wall.
T(I,J) : temperature.
RIB(I,J) : \( \pi \sigma^4 / \lambda \) of the medium.
KBCI1() : boundary condition indicator for the west boundary:
KBCI1() : boundary condition indicator for the west boundary:
: = 1; given temperature.
: = 2; symmetry
: = 3; periodic.
CAPPA(I,J) : absorption coefficient, \( \kappa \).
SIGMA(I,J) : scattering coefficient, \( \sigma_s \).
BETA(I,J) : extinction coefficient, \( \beta = \kappa + \sigma_s \).
BM() : modified extinction coefficient.
SM() : modified source coefficient.
PHASE() : phase function.
L1 : value of I for the right-boundary grid line.
M1 : value of J for the top-boundary grid line.
K1 : value of \( L \) for the \( \theta = 180^\circ \) grid line.
J1 : value of M for the \( \phi = 360^\circ \) grid line.
L2 : \( L1 - 1 \)
L3 : \( L2 - 1 \)
M2 : \( M1 - 1 \)
M3 : \( M2 - 1 \)
K2 : \( K1 - 1 \)
K3 : \( K2 - 1 \)
J2 : \( J1 - 1 \)

\(^\dagger\) KBCL1(), KBCJ1() and KBCM1() are similar to KBCI1().
J3 : J2 - 1
X(I) : value of X at grid location I.
XU(I) : value of X at the control-volume face.
    XU(1) is meaningless.
XCV(I) : x-direction width of control-volume.
    XCV(1) and XCV(L1) are meaningless.
Y(J) : value of Y at grid location J.
YV(J) : value of Y at the control-volume face.
    YV(1) is meaningless.
YCV(J) : y-direction width of control-volume.
    YCV(1) and YCV(M1) are meaningless.
AX(J) : YCV(J)
AY(I) : XCV(I)
VOL(I,J) : XCV(I) * YCV(J)
THETA(L) : value of θ at grid location L.
THETAI(L) : value of θ at the control-volume face.
    THETAI(1) is meaningless.
PHI(M) : value of φ at grid location M.
PHII(M) : value of φ at the control-volume face.
    PHII(1) is meaningless.
DCX() : $D_{CX}$
DCY() : $D_{CY}$
DOM() : $\Delta \Omega$
ISOLID() : index denoting blockage.
QPY() : $q^+_y$
QMY() : $|q^-_y|$
QPX() : $q^+_x$
QMX() : $|q^-_x|$
G : incident radiation energy.
REFERENCES


