

# **Radiant 2.0: A User's Guide**

By  
Matthew J. Christi

Department of Atmospheric Science  
Colorado State University  
Fort Collins, Colorado

Research was supported by DOE/ARM grant #DE-FG03-94ER61748

The logo for Colorado State University, featuring the words "Colorado" and "State" in a large, bold, serif font, with "University" in a smaller, sans-serif font below them.

**Colorado  
State  
University**

**Department of  
Atmospheric Science**

Paper No. 742

# Radiant 2.0: A User's Guide

**Matthew J. Christi**

Colorado State University  
Department of Atmospheric Science  
1371 Campus Delivery  
Fort Collins, Colorado 80523-1371

Research Supported by DOE/ARM Grant  
#DE-FG03-94ER61748

Colorado State University  
Department of Atmospheric Science  
Paper No. 742

July 2003

# Contents

<b>1</b>	<b>Why Another Radiative Transfer Solver?</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Doubling/Adding and DISORT . . . . .	2
1.3	Our Goal . . . . .	3
<b>2</b>	<b>Solving the Radiative Transfer Equation</b>	<b>5</b>
2.1	Radiative Transfer Theory . . . . .	5
2.2	Method of Solving the Radiative Transfer Equation . . . . .	7
2.3	Radiant: An Efficient Approach to Computing Radiative Transfer . . . . .	10
2.4	Modified Eigenmatrix Method . . . . .	11
<b>3</b>	<b>Radiant, Doubling/Adding, and DISORT: Accuracy and Timing Comparisons</b>	<b>15</b>
3.1	Accuracy Comparisons . . . . .	15
3.2	Timing Comparisons . . . . .	17
<b>4</b>	<b>Using Radiant</b>	<b>21</b>
4.1	Operational Modes . . . . .	21
4.2	Program Structure & Subroutine Description . . . . .	23
4.3	Input Parameters . . . . .	26
4.4	Special Program Provisions . . . . .	34

4.4.1 Singularity Busting . . . . .	34
4.4.2 Limitations Imposed by Some Time-Saving Features . . . . .	35

<b>A Global Transmission and Reflection Matrices: Numerical Considerations</b>	<b>37</b>
--------------------------------------------------------------------------------	-----------

# List of Figures

2.1	Physical processes affecting the transfer of radiation through a medium. . . . .	6
2.2	The interaction principle illustrated for a homogeneous layer. . . . .	10
3.1	Results of a speed comparison between the eigenmatrix method in <i>Radiant</i> and the doubling method. . . . .	18
3.2	Results of a speed comparison between <i>Radiant</i> and DISORT2. . . . .	19
4.1	Illustration of <i>Radiant's</i> normal mode of operation. . . . .	22
4.2	Illustration of <i>Radiant's</i> layer-saving mode of operation. . . . .	22
4.3	<i>Radiant's</i> program structure. . . . .	23

# List of Tables

1.1	Sample of organizations or authors and some of their radiative transfer codes.	2
3.1	Comparison of <b>upwelling</b> radiances generated by Van de Hulst Table 35 (VDH), a doubling/adding scheme (D/A), and DISORT (D) with those generated by <i>Radiant</i> (R) for a given layer of different optical parameters.	16
3.2	Comparison of <b>downwelling</b> radiances generated by Van de Hulst Table 35 (VDH), a doubling/adding scheme (D/A), and DISORT (D) with those generated by <i>Radiant</i> (R) for a given layer of different optical parameters.	16
3.3	Comparison of radiances generated by a doubling/adding scheme (D/A) with those generated by <i>Radiant</i> (R).	17
4.1	KFLAG and WVN_FLAG settings to use different operational modes in <i>Radiant</i> .	29

# Chapter 1

## Why Another Radiative Transfer Solver?

### 1.1 Introduction

There are quite a few radiative transfer (RT) codes currently available to the atmospheric science community. A list of some of the RT codes currently in use can be found in Table 1.1. Although the amount of code out there is not scarce, as one carefully observes from the table, the heart of many of these codes is based on one of two methods: the eigenmatrix method as implemented by the Discrete Ordinate Method of Radiative Transfer (DISORT) RT solver or the doubling/adding method. We mention in passing that not all of the RT codes listed in Table 1.1 compute the radiative transfer for plane-parallel media as DISORT, doubling/adding, or the code described in this work.

Although widely used, the above two methods have some drawbacks which limit their effectiveness in at least some applications. We begin with a brief discussion of some of the strengths and weaknesses of the doubling/adding method and of the eigenmatrix method as implemented by DISORT.

Table 1.1: Sample of organizations or authors and some of their radiative transfer codes (Source: the World Wide Web; \* = Not available)

Organization or Authors	Name of Code	RT Core
Air Force Research Lab	MODTRAN4	DISORT
Air Force Research Lab	MOSART	DISORT
Arve Kylling and Bernhard Mayer	LibRadTran	DISORT
Boston University	Streamer	DISORT
Brookhaven National Lab	*	doubling/adding
Environmental Systems Science Centre	DOORS	Similar to DISORT
Institut Für Meereskunde Kiel	MC-Layer	Monte Carlo
NASA - Ames	*	doubling/adding
NCAR	TUV	DOM
NOAA - GFDL	*	doubling/adding
Royal Netherlands Meteorological Inst.	DAK	doubling/adding
U. of Alaska Fairbanks	UVSPEC	DISORT
U. of Cal., San Diego	FEMRAD	FEM
U. of Cal., Santa Barbara	SBDART	DISORT
U. of Colorado	PolRadTran	DISORT
U. of Colorado	SHDOM	SHDOM
U. of Maryland	*	doubling/adding
U. of Texas, Arlington	*	doubling/adding
Universitetet I Oslo	RADTRAN	DISORT
U.S. Army Developmental Test Command	BLIRB	DOM

## 1.2 Doubling/Adding and DISORT

The idea of the doubling/adding method is to build up layers of atmosphere of relatively large optical depth, each with given optical properties, by taking slices of atmosphere of minute optical depth. This is done by a process of doubling the original optical depth iteratively until the desired optical depth of the layer is attained. The resulting individual, homogeneous layers of atmosphere so constructed are then added together to yield the overall optical properties of the atmosphere. On the other hand, the eigenmatrix method as implemented by DISORT makes use of eigenvalues and eigenvectors to solve a system of differential equations to obtain a solution to the radiative transfer equation.

Both methods have their strengths and weaknesses. A primary strength of the doubling/adding method is that, once a given atmospheric layer is constructed,



it does not need to be recomputed if the optical properties in the given layer do not change. Upon computing any other layers in the medium whose optical properties do change, the layers can then again be added together to obtain the RT solution. However, some of this computational efficiency can be lost if some layers that need recomputing are optically thick.

In contrast, the time taken to obtain the RT solution via the eigenmatrix approach as implemented by DISORT takes the same amount of time irrespective of the optical depths of the individual layers. However, when one wishes to account for changes that take place in the atmosphere, the entire system of differential equations needs to be re-solved to obtain the new solution.

Because of these limitations, the time required to perform retrievals of atmospheric constituents for example (where often many calls are made to a RT solver to perform forward model computations) is made unnecessarily expensive. We will demonstrate another way of going about solving the equation of transfer that, at least for some applications, can yield faster results without sacrificing the accuracy of those results.

### 1.3 Our Goal

The purpose of this paper is two-fold: (1) to introduce and describe a new multiple scattering plane-parallel atmospheric RT solver that takes advantage of the benefits of both the above methods while leaving some of their more undesirable characteristics behind and (2) to assist the user in using this solver.

To accomplish this, in chapter 2 we begin with some basic radiative transfer theory followed by a formulation for solving the equation of transfer. We then turn to a method of obtaining the solution to the equation of transfer as employed in the multiple scattering plane-parallel atmospheric RT solver that is the subject of this work: *Radiant*. In chapter 3, we discuss the results of accuracy and timing com-

parisons with both doubling/adding and DISORT. In chapter 4, which will probably be the most beneficial to users, a description of *Radiant's* operational modes is provided along with information regarding its program structure, input parameters, and some special considerations when using *Radiant*.

## Chapter 2

# Solving the Radiative Transfer Equation

### 2.1 Radiative Transfer Theory

The essence of radiative transfer (RT) theory can be described by equation (2.1) and Figure 2.1. The term on the lefthand side of equation (2.1) describes the change in intensity of radiation ( $I$ ) as it travels through a volume of space in a given medium. On the right side,  $\mu_{\odot}$  is the cosine of the solar zenith angle,  $\sigma_e$ ,  $\sigma_s$ , and  $\sigma_a$  are the extinction, scattering, and absorption coefficients, respectively,  $P$  is the scattering phase function, and  $B$  is the Planck function of emission. The terms on the right side can be interpreted as follows:

$$\begin{aligned} \mu \frac{dI(z, \mu, \phi)}{dz} = & -\sigma_e(z)I(z, \mu, \phi) \\ & + \frac{\sigma_s(z)}{4\pi} \int_0^{2\pi} \int_{-1}^1 P(z, \mu, \phi, \mu', \phi') I(z, \mu', \phi') d\mu' d\phi' \\ & + \frac{F_{\odot}}{4\pi} \sigma_s(z) P(z, \mu, \phi, \mu_{\odot}, \phi_{\odot}) e^{-\sigma_e(z_T - z)/\mu_{\odot}} \\ & + \sigma_a(z) B(T(z)) \end{aligned} \quad (2.1)$$

1st term - The attrition of photons that radiation undergoes due to absorption

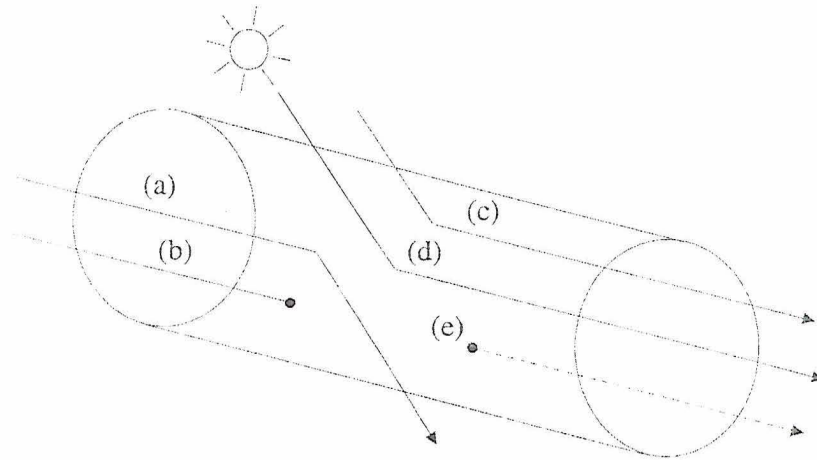


Figure 2.1: Physical processes affecting the transfer of radiation through a medium. These processes are (a) out-scattering, (b) absorption, (c) and (d) in-scattering of diffuse and direct radiation, respectively, and (e) emission.

and out-scattering (the scattering of a photon **out** of the path it had when it first entered a volume of space). The absorption and out-scattering together comprise what is commonly referred to in RT circles as the extinction. See *a* and *b* in Figure 2.1 for a depiction of out-scattering and absorption, respectively.

2nd and 3rd terms - The accumulation of photons that radiation experiences due to in-scattering (the scattering of photons from different directions **into** the path of the incident radiation being considered). The second term is associated with the in-scattering of diffuse radiation, whereas the third term is associated with the in-scattering from a direct source (in this case, the sun). These processes are depicted by *c* and *d* in Figure 2.1.

4th term - The addition of photons that radiation experiences due to the emission of photons by particles or gases within the medium into the direction of the incident radiation. In Figure 2.1, this is depicted by *e*.

## 2.2 Method of Solving the Radiative Transfer Equation

An overall description of the method used to compute radiances in *Radiant* is now described. A method of solving the equation of transfer (2.1) involves replacing the integrals in the equation by finite sums, thereby producing a discrete form of the equation. By using a quadrature scheme such as Gaussian or Lobatto quadrature, if one expresses the phase function  $P(z, \mu, \phi, \mu', \phi')$  as the sum of a suitable number of orthogonal polynomials, Legendre polynomials for instance, the integrals in (2.1) are exact. The number of discrete equations which are required to represent the radiation field will depend on the number of terms required to represent the phase function. For example, if the number of terms required to represent the phase function is  $N$ , then it will take  $n \geq N/2$  equations using Gauss quadrature,  $n \geq (N + 2)/2$  equations using Lobatto quadrature, or  $n \geq N$  equations using Double Gauss quadrature to represent the radiation field and allow the integrals to be computed exactly. Here,  $N$  is assumed to be even and  $n$  is the number of upward (or downward) streams used to describe the radiation field.

The radiance  $I$  can be described by the Fourier expansion

$$I(z, \mu, \phi) = \sum_{m=0}^{N-1} I_m(z, \mu) \cos[m(\phi - \phi_{\odot})] \quad (2.2)$$

where  $z$  is altitude,  $\mu$  is the cosine of the observation angle in reference to zenith, and  $\phi$  and  $\phi_{\odot}$  are the azimuth angle of the radiance and the sun, respectively, with respect to a given coordinate system.

The system of scalar equations resulting from the above discretization process of (2.1) can be expressed as a set of matrix equations, one for each  $m$  in the Fourier expansion of  $I$ :

$$\begin{aligned}
\frac{d^m I^\pm}{dz} &= \mp \sigma_e (M^{-1}) ({}^m I^\pm) \\
&\pm (1 + \delta_{0m}) \frac{\sigma_s}{4} [(M^{-1}) ({}^m P^\pm) C ({}^m I^+) + (M^{-1}) ({}^m P^\mp) C ({}^m I^-)] \\
&\pm \frac{\sigma_s}{4\pi} F_\odot (M^{-1}) ({}^m P_\odot^\mp) e^{[-\sigma_e (z_T - z) / \mu_\odot]} \\
&\pm \sigma_a (M^{-1}) ({}^m Y) B(T(z))
\end{aligned} \tag{2.3}$$

Here, we have:

${}^m I^\pm$  - A vector describing the  $m$ th term in the Fourier expansion of  $I$  where

(+) represents that portion of the vector representing upwelling radiances and (-) that portion representing downwelling radiances.

${}^m P^\pm$  and  ${}^m P_\odot^\pm$  - The phase function matrices for the forward (+) and backward (-) scattering of diffuse and direct radiation, respectively.

$M^{-1}$  - A matrix consisting of the reciprocals of quadrature roots.

$C$  - A matrix of quadrature weights.

${}^m Y$  - A vector of unity for  $m = 0$  and a vector of zeros otherwise.

$B(T(z))$  - The Planck function.

$\sigma_a$  - The absorption coefficient of the medium.

$\sigma_e$  - The extinction coefficient of the medium.

$\delta_{0m}$  - The Kronecker delta.

$F_\odot$  - The solar flux incident at the top of the medium.

$T(z)$  - The temperature at altitude  $z$ .

$\mu_\odot$  - The cosine of the solar zenith angle.

The set of equations represented by (2.3) can be rendered more compact in the following matrix equation

$$\frac{d}{dz} \begin{bmatrix} {}^m I^+ \\ {}^m I^- \end{bmatrix} = \begin{bmatrix} \hat{t}^m & -\hat{r}^m \\ \hat{r}^m & -\hat{t}^m \end{bmatrix} \begin{bmatrix} {}^m I^+ \\ {}^m I^- \end{bmatrix} + \begin{bmatrix} {}^m \hat{\Sigma}^+ \\ -{}^m \hat{\Sigma}^- \end{bmatrix} \tag{2.4}$$

where

$$\hat{t}^m = -\sigma_e(M^{-1}) + (1 + \delta_{om})\frac{\sigma_s}{4}[(M^{-1})({}^m P^+)C] \quad (2.5)$$

$$\hat{r}^m = -(1 + \delta_{om})\frac{\sigma_s}{4}[(M^{-1})({}^m P^-)C] \quad (2.6)$$

are matrices describing the local transmission and reflection properties of a given layer in the medium (i.e. the layer's intrinsic scattering properties) and

$${}^m \hat{\Sigma}^+ = \frac{\sigma_s}{4\pi} F_{\odot}(M^{-1})({}^m P_{\odot}^-) e^{[-\sigma_e(z_T - z)/\mu_{\odot}]} + \sigma_a(M^{-1})({}^m Y)B(z) \quad (2.7)$$

$${}^m \hat{\Sigma}^- = \frac{\sigma_s}{4\pi} F_{\odot}(M^{-1})({}^m P_{\odot}^+) e^{[-\sigma_e(z_T - z)/\mu_{\odot}]} + \sigma_a(M^{-1})({}^m Y)B(z) \quad (2.8)$$

are vectors describing sources of upwelling and downwelling radiation within the medium, respectively. Finally, denoting the matrix of local transmission and reflection functions by  $A$  and the radiance and source vectors by  $I$  and  $\Sigma$ , respectively, we arrive at the more concise expression

$$\frac{d}{dz}I = AI + \Sigma \quad (2.9)$$

where a dependence on  $m$  is understood. The above system of differential equations described by this matrix equation has the formal solution

$$I = e^{A\sigma_e H} I_o + \int_0^H \sigma_e e^{A\sigma_e(H-z)} \Sigma dz \quad (2.10)$$

where  $H$  is the geometric thickness of the homogeneous medium.

The solution to the RT equation can be obtained in a rather efficient manner by employing what is sometimes referred to as the interaction principle. The essence of this principle is displayed in Figure 2.2. By using the interaction principle, the solution to the RT equation can be expressed in terms of global transmission and

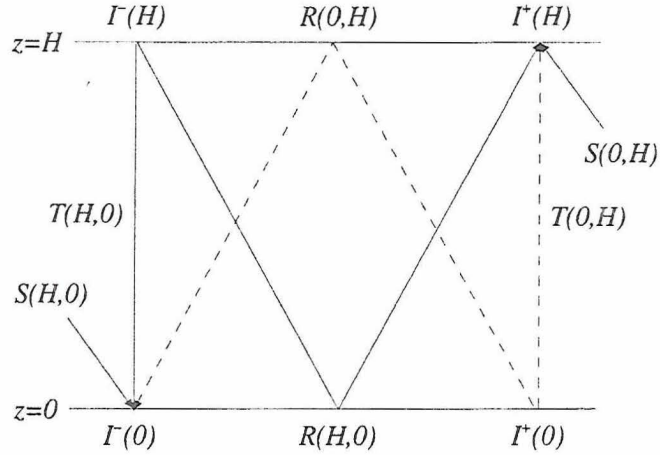


Figure 2.2: The interaction principle illustrated for a homogeneous layer where  $T(H,0)$  and  $R(H,0)$  are the downwelling global transmission and reflection matrices and  $T(0,H)$  and  $R(0,H)$  their upwelling counterparts.  $S(H,0)$  is the downwelling source vector and  $S(0,H)$  the upwelling source vector, respectively.

reflection matrices and two source vectors. Here, the overall solution is rendered

$$I^+(H) = T(0, H)I^+(0) + R(H, 0)I^-(H) + S(0, H) \quad (2.11)$$

$$I^-(0) = R(0, H)I^+(0) + T(H, 0)I^-(H) + S(H, 0) \quad (2.12)$$

where  $I^+(H)$  and  $I^-(H)$  are vectors of upwelling and downwelling radiation at the top of the layer,  $I^+(0)$  and  $I^-(0)$  the corresponding vectors at the bottom of the layer,  $T$  and  $R$  are the global transmission and reflection matrices for the layer, and  $S(0,H)$  and  $S(H,0)$  are the accompanying source vectors.

## 2.3 Radiant: An Efficient Approach to Computing Radiative Transfer

As discussed in chapter 1, it would be highly desirable to use a method that was not sensitive to optical depth (as the doubling/adding method) and at the same time would not demand the recomputation of the entire RT solution if the optical properties



in just one portion of the medium change (as done by DISORT). The idea is to take the optical depth insensitivity of the eigenmatrix approach and combine it with the "individual layeredness" of the doubling/adding method. By using the eigenmatrix method to compute the individual layers and then using adding to combine them, what results is an often faster yet accurate hybrid. These ideas have been united in a new radiative transfer code called *Radiant*.

*Radiant* is used to describe the influence of nature on radiant energy at wavelengths in the solar and infrared portions of the electromagnetic spectrum and produce the resulting radiant intensities. It is a multi-stream, plane-parallel RT code that accounts for multiple scattering in the atmosphere and uses the ideas as described above: build individual (homogeneous) layers of atmosphere using the eigenmatrix method and then combine the layers using adding. This will be referred to as the modified eigenmatrix method (MEM). The  $T$  and  $R$  matrices and source vectors stated earlier are computed for each layer of atmosphere. Once these have been computed for a given layer, they are combined with those of other layers to build up the atmosphere for its current state. For layers whose optical properties do not change, they can be saved for subsequent use and again easily combined with those of other layers whose optical properties do change to quickly obtain the radiances for a new atmospheric state. We turn now to a more detailed description of the modified eigenmatrix method.

## 2.4 Modified Eigenmatrix Method

Although theoretically straight forward, the solution of the radiative transfer equation (2.10) is fraught with numerical difficulty as the instability of computing the exponential matrix is well known. Using eigenvalues and eigenvectors, the exponential matrix can be expressed as

$$e^{A\tau} = X e^{\Lambda\tau} X^{-1} \quad (2.13)$$

where  $e^{\Lambda\tau}$  is a diagonal matrix with the exponentials of the eigenvalues of the aforementioned  $A$  matrix on the diagonal,  $X$  is the matrix of associated eigenvectors and  $X^{-1}$  its inverse. To solve for the eigenvalues of  $A$ , polynomial deflation can be used to reduce the computational time as well as improve numerical stability (Stamnes and Swanson (1981); Stamnes et al. (1988)). The exponentials of the positive eigenvalues in  $e^{\Lambda\tau}$  can introduce numerical instability when the optical depth  $\tau$  becomes large; however, Stamnes and Conklin (1984) showed that this problem can be largely overcome by using a scaling transformation. Using a similar transformation, the global transmission and reflection matrices,  $T$  and  $R$  (recall Figure (2.2)), take the form (Benedetti et al., 2002)

$$\begin{aligned} T(H, 0) = & -u_+[E - (u_+^{-1}u_-)^2][(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)}] \\ & \{E - [(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)}]^2\}^{-1}u_-^{-1} \end{aligned} \quad (2.14)$$

$$\begin{aligned} R(H, 0) = & -u_+[E - (u_+^{-1}u_-)e^{-\Lambda^+\tau(H)}(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)}] \\ & \{E - [(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)}]^2\}^{-1}u_-^{-1} \end{aligned} \quad (2.15)$$

where  $u_+$  and  $u_-$  are matrices, when appropriately assembled, composing the matrix  $X$ ,  $\Lambda^+$  is a diagonal matrix containing the positive eigenvalues of the matrix  $A$ ,  $\tau(H)$  is the optical depth of a layer of geometric thickness  $H$ , and  $E$  is the identity matrix. Note that the expressions for  $T$  and  $R$  only contain decaying exponentials.

The global source vectors  $S(0,H)$  and  $S(H,0)$  in (2.11) and (2.12), respectively, can be rendered

$$S(0, H) = -T(0, H)(S_1^+ - S_3^+) - R(H, 0)(S_2^- - S_4^-) + S_2^+ - S_4^+ \quad (2.16)$$

$$S(H, 0) = -R(0, H)(S_1^+ - S_3^+) - T(H, 0)(S_2^- - S_4^-) + S_1^- - S_3^- \quad (2.17)$$

where the solar source components  $S_1^\pm$  and  $S_2^\pm$  are

$$S_1^\pm = \pm \left( \frac{E}{\mu_\odot} - A \right)^{-1} \frac{\omega_o}{4\pi} F_\odot M^{-1} P_\odot^\mp e^{-\frac{\tau}{\mu_\odot}} \quad (2.18)$$

$$S_2^\pm = \pm \left( \frac{E}{\mu_\odot} - A \right)^{-1} \frac{\omega_o}{4\pi} F_\odot M^{-1} P_\odot^\mp \quad (2.19)$$

and the thermal source components  $S_3^\pm$  and  $S_4^\pm$  are

$$S_3^\pm = \pm A^{-1} (1 - \omega_o) M^{-1} ({}^m Y) B_0 \pm (\sigma_e)^{-1} A^{-2} (1 - \omega_o) M^{-1} ({}^m Y) B_1 \quad (2.20)$$

$$S_4^\pm = S_3^\pm \pm H A^{-1} (1 - \omega_o) M^{-1} ({}^m Y) B_1 \quad (2.21)$$

Here,  $\mu_\odot$ ,  $A$ ,  $F_\odot$ ,  $M^{-1}$ ,  $P_\odot^\mp$ ,  ${}^m Y$ , and  $\sigma_e$  are the same as given in section 2.2,  $\omega_o$  is the single scatter albedo,  $B_0$  and  $B_1$  are constants describing the change of the Planck emission with height within the layer (assumed to vary linearly here), and  $H$  and  $E$  are again the geometric thickness of the layer and identity matrix, respectively.

Upon careful inspection of the expressions for  $T$  and  $R$ , one observes that some further numerical savings can be achieved by employing some substitutions, rearranging, and simplifying. A brief discussion of this and the resulting expressions for global  $T$  and  $R$  are given in appendix A.



## Chapter 3

# Radiant, Doubling/Adding, and DISORT: Accuracy and Timing Comparisons

### 3.1 Accuracy Comparisons

To test the trueness of *Radiant's* algorithms, calculations of radiant intensity were performed for a layer with different values of optical depth  $\tau$ , single-scatter albedo  $\omega_o$ , asymmetry factor  $g$ , and cosine of solar zenith angle  $\mu_\odot$  and compared with the radiance tables from VandeHulst (1980) as well as the values generated by two doubling/adding schemes and DISORT for the same optical parameters. Tables 3.1 and 3.2 show the results of a comparison between Van de Hulst Table 35, the doubling/adding scheme used in Gabriel et al. (1990), DISORT, and *Radiant* using the Henyey-Greenstein phase function. Both DISORT and *Radiant* were run in a 16-stream mode (8 upward and 8 downward radiances) during these tests. Table 3.3 reveals the results of a comparison with the doubling/adding scheme used in Miller et al. (2000) and also used for comparison by Benedetti et al. (2002) for the same optical parameters and values of degree  $m$  for the Henyey-Greenstein phase function.

Table 3.1: Comparison of **upwelling** radiances generated by Van de Hulst Table 35 (VDH), a doubling/adding scheme (D/A), and DISORT (D) with those generated by *Radiant* (R) for a given layer of different optical parameters.

$\tau$	$\omega_o$	$g$	$\mu$	$\mu_{\odot}$	VDH $I^+(H)$	D/A $I^+(H)$	D $I^+(H)$	R $I^+(H)$
1	1	0.75	1	0.1	1.5137E-01	1.5172E-01	1.5836E-01	1.4854E-01
1	1	0.75	1	0.5	1.0120E-01	1.0146E-01	1.0771E-01	1.0020E-01
1	1	0.75	1	1.0	0.3909E-01	0.3925E-01	0.2019E-01	0.3796E-01
2	1	0.75	1	0.1	2.0571E-01	2.0618E-01	2.1269E-01	2.0216E-01
2	1	0.75	1	0.5	2.0119E-01	2.0163E-01	2.0798E-01	1.9991E-01
2	1	0.75	1	1.0	1.0438E-01	1.0476E-01	8.3351E-02	1.0277E-01
4	1	0.75	1	0.1	2.8433E-01	2.8485E-01	2.9130E-01	2.7987E-01
4	1	0.75	1	0.5	3.4710E-01	3.4764E-01	3.5391E-01	3.4561E-01
4	1	0.75	1	1.0	2.5658E-01	2.5712E-01	2.3530E-01	2.5465E-01
8	1	0.75	1	0.1	3.7997E-01	3.8042E-01	3.8693E-01	3.7446E-01
8	1	0.75	1	0.5	5.1971E-01	5.2013E-01	5.2651E-01	5.1808E-01
8	1	0.75	1	1.0	4.9270E-01	4.9300E-01	4.7138E-01	4.9086E-01

Table 3.2: Comparison of **downwelling** radiances generated by Van de Hulst Table 35 (VDH), a doubling/adding scheme (D/A), and DISORT (D) with those generated by *Radiant* (R) for a given layer of different optical parameters.

$\tau$	$\omega_o$	$g$	$\mu$	$\mu_{\odot}$	VDH $I^-(0)$	D/A $I^-(0)$	D $I^-(0)$	R $I^-(0)$
1	1	0.75	1	0.1	2.1380E-01	2.1468E-01	2.1075E-01	2.1068E-01
1	1	0.75	1	0.5	2.6663E-01	2.6805E-01	2.6647E-01	2.6562E-01
1	1	0.75	1	1.0	3.0652E+00	3.0862E+00	2.9096E+00	3.0689E+00
2	1	0.75	1	0.1	2.7614E-01	2.7670E-01	2.7513E-01	2.7259E-01
2	1	0.75	1	0.5	4.2244E-01	4.2370E-01	4.2255E-01	4.2142E-01
2	1	0.75	1	1.0	2.8247E+00	2.8205E+00	2.7008E+00	2.8345E+00
4	1	0.75	1	0.1	2.9606E-01	2.9608E-01	2.9594E-01	2.9267E-01
4	1	0.75	1	0.5	5.0828E-01	5.0852E-01	5.0835E-01	5.0765E-01
4	1	0.75	1	1.0	1.5155E+00	1.5014E+00	1.4762E+00	1.5234E-01
8	1	0.75	1	0.1	2.3639E-01	2.3619E-01	2.3636E-01	2.3386E-01
8	1	0.75	1	0.5	4.2235E-01	4.2206E-01	4.2235E-01	4.2214E-01
8	1	0.75	1	1.0	6.7002E-01	6.6744E-01	6.6797E-01	6.7166E-01

Table 3.3: Comparison of radiances generated by a doubling/adding scheme (D/A) with those generated by *Radiant* (R). The optical parameters are:  $\tau = 1$ ,  $\omega = 1$ ,  $g = 0.8$ . Also,  $\mu_{\odot} = \cos 30^{\circ}$ .

$\mu$	m	D/A $I^+(H)$	R $I^+(H)$	D/A $I^-(0)$	R $I^-(0)$
0.9894	0	9.9717E-03	9.9718E-03	1.6764E-01	1.6765E-01
0.7554	0	1.6232E-02	1.6232E-02	1.8942E-01	1.8942E-01
0.0950	0	4.8565E-02	4.8566E-02	6.9504E-02	6.9504E-02
0.9894	3	1.0576E-02	1.0576E-02	2.8254E-01	2.8254E-01
0.7554	3	2.1393E-02	2.1393E-02	6.9865E-01	6.9865E-01
0.0950	3	8.3972E-02	8.3973E-02	1.3309E-01	1.3309E-01
0.9894	7	1.0577E-02	1.0577E-02	2.8442E-01	2.8442E-01
0.7554	7	2.1415E-02	2.1416E-02	8.3781E-01	8.3781E-01
0.0950	7	8.4466E-02	8.4467E-02	1.3444E-01	1.3444E-01
0.9894	11	1.0577E-02	1.0577E-02	2.8441E-01	2.8442E-01
0.7554	11	2.1408E-02	2.1407E-02	8.5923E-01	8.5922E-01
0.0950	11	8.4532E-02	8.4533E-02	1.3445E-01	1.3449E-01
0.9894	15	1.0577E-02	1.0577E-02	2.8441E-01	2.8442E-01
0.7554	15	2.1406E-02	2.1405E-02	8.6311E-01	8.6310E-01
0.0950	15	8.4497E-02	8.4498E-02	1.3447E-01	1.3446E-01

## 3.2 Timing Comparisons

To test the speed of *Radiant's* algorithms, two speed comparisons were performed. First, *Radiant* was tested against the doubling/adding code used in Greenwald and Stephens (1988) to compare the time it took to compute radiances for layers of different optical depth. This was done to get a sense of how fast the eigenmatrix method was against the doubling method for building a given layer. Here, Figure 3.1 confirms that, as one expects, the doubling method takes longer to compute the global transmission, reflection, and source properties of the layer as the optical depth  $\tau$  increases (note that the abscissa on the plot is  $\log \tau$ ) whereas the eigenmatrix method, which is insensitive to optical depth, takes a fixed amount of time to compute the radiances. What is somewhat enlightening is the fact that the eigenmatrix method (at least when being run in a 16-stream mode as this was) is faster than the doubling method for the vast majority of optical depths experienced in the real atmosphere. The crossover point is at  $\approx \tau = 0.003$  with the eigenmatrix method being faster for

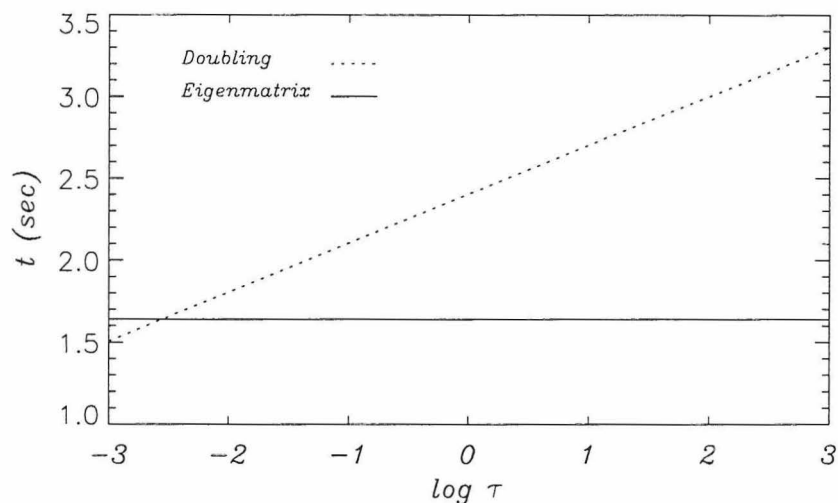


Figure 3.1: Results of a speed comparison between the eigenmatrix method in *Radiant* and the doubling method. The total times are the result of computing the radiances for a given atmospheric scene 500 times on a computer with a 400 MHz microprocessor.

every optical depth greater than this. For example, at  $\tau = 10$  its about 66% faster. This increase in speed, while not outstanding, can potentially save much valuable time over the course of a long series of computations.

For the second test, *Radiant* was tested against DISORT (DISORT2 to be exact) to see, for a given atmospheric state built up from a fixed number of layers, what kind of time savings can be achieved by using *Radiant* as opposed to DISORT2 when only the optical properties in one layer of atmosphere change and the radiances are recomputed. For example, this situation is faced in practice when doing profile retrievals of atmospheric gases where Jacobians are needed to perform the retrieval and computing elements of the Jacobian by finite difference is required.

Figure 3.2 shows the results of these tests. The solid line denoted "Radiant (1)" is the time it took *Radiant* to compute the radiance for a new atmospheric scene for the number of layers indicated. The dash dot line denoted "DISORT2" is the time it took DISORT2 to compute the radiance for the same scene and number of layers. The two dashed lines denoted "Radiant (2)" and "Radiant (3)" are the times



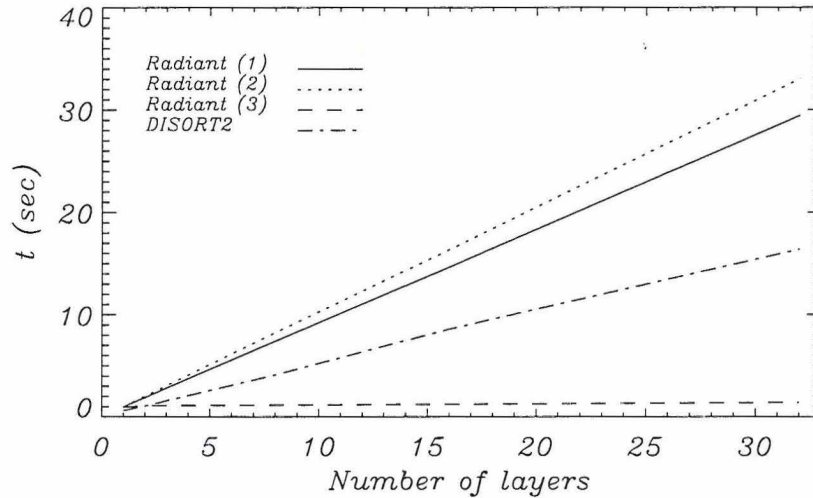


Figure 3.2: Results of a speed comparison between *Radiant* and DISORT2. The total times are the result of computing the radiances for a given atmospheric scene 500 times on a computer with a 550 MHz microprocessor. See text for details between *Radiant* (1), (2), and (3).

it took *Radiant* to perform Jacobian-related calculations. Specifically, "Radiant (2)" is the same as "Radiant (1)" except that additional layer computation and saving were performed to prepare for following calls to *Radiant* when it would be tasked to compute the radiances for new atmospheric scenes where only the optical properties of one layer would change. Again, this is done in practice when the computation of Jacobian elements by finite difference is required. The extra time spent up front here can yield big dividends as the dashed line denoted "Radiant (3)" reveals. When subsequent calls to *Radiant* are made in this scenario, aside from some rescaling of source terms in the layers below the affected layer, only the optical properties of the affected layer need recomputed - the others are saved in memory both as individual layers and blocks of layers. Following the recomputation of the affected layer, it only needs added to the other layers and/or blocks of layers that have already been saved to obtain the new radiances.

The savings observed is because, although using an eigenmatrix formulation

to obtain the solution to the radiative transfer equation, DISORT2 must re-solve the whole radiative transfer problem when the optical properties change in a single layer. This leads to much unnecessary computation in scenarios such as encountered in various retrieval work where only the recomputation of one or two layers may be required and the rest of the atmospheric state remains unchanged. As further evidence of the power of these ideas, when the saving features spoken of above were implemented in computing the elements of the Jacobian for NIR wavenumbers when performing retrievals of in CO<sub>2</sub> in a model atmosphere with 33 layers, the computation of the Jacobian was sped up by over a factor of 14!

The above illustrates some of the benefits that *Radiant* can provide in the way of saved time when working on certain problems requiring repeated calls to a radiative transfer model. Additional features are also planned to make the code more flexible and useful to the atmospheric science community at large.

# Chapter 4

## Using Radiant

### 4.1 Operational Modes

*Radiant* computes the radiances emerging from the boundaries of a homogeneous layer or heterogeneous block of layers with or without an underlying surface and has two operational modes: normal mode and layer-saving mode. In normal mode, each time *Radiant* is called by the parent program, it constructs the global transmission ( $T$ ) and reflection ( $R$ ) matrices and source vectors ( $S$ ) for each layer of atmosphere from the top down. As these are constructed for each layer, they are combined with those representing the ever growing block of layers as illustrated in Figure 4.1. When the composite  $T$  and  $R$  matrices and source vectors  $S$  for the entire atmospheric scene have been constructed, the boundary conditions are then applied and the RT solution for the scene obtained. Thus, in this mode, *Radiant* computes the RT solution for the scene "from scratch".

In contrast, when *Radiant* is called in layer-saving mode (see Figure 4.2), it assumes it has already been called once to compute the RT solution for a given atmospheric scene and that the optical properties of one of the layers has changed. For example, in Figure 4.2, layer 3 has changed (shaded). It then proceeds to look for the changed layer. Once it finds the layer, the  $T$  and  $R$  matrices and source vectors

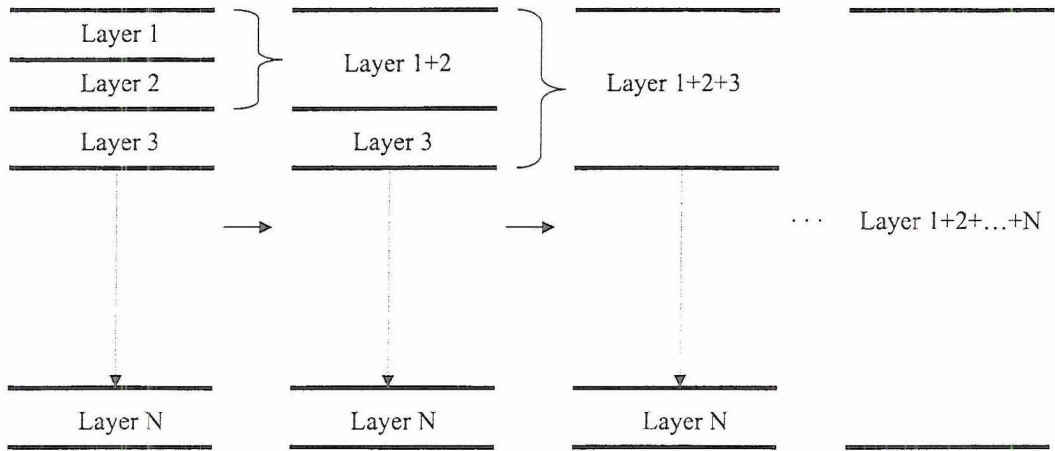


Figure 4.1: Illustration of *Radiant's* normal mode of operation.

$S$  describing the optical properties of the changed layer are determined. These are then combined with those of other layers or blocks of layers already saved in memory from the first time *Radiant* was called to compute the RT solution for the original atmospheric scene. In Figure 4.2, these are the unshaded blocks. The boundary conditions are again applied and the RT solution for the new scene obtained.

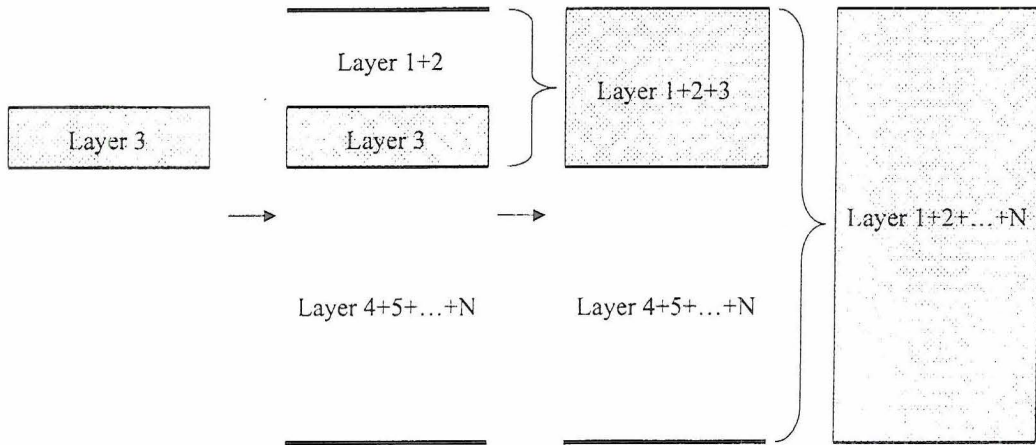
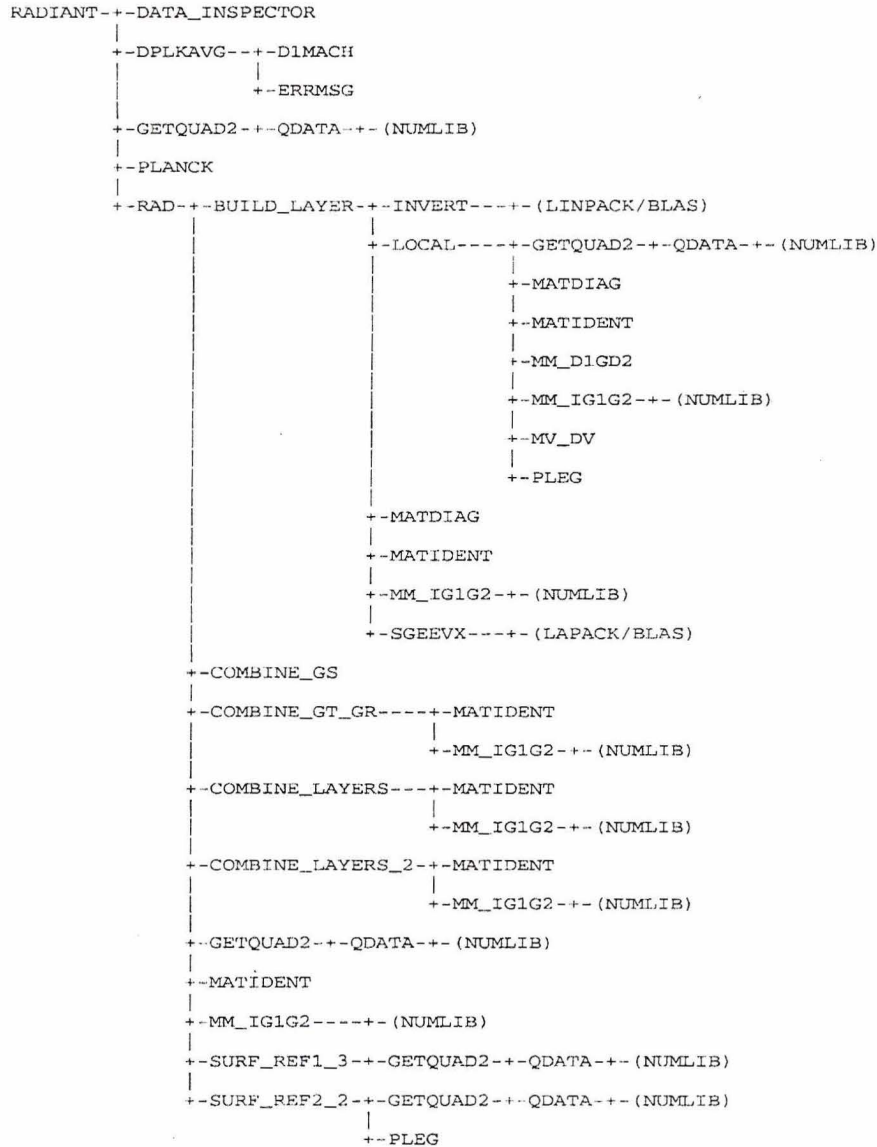


Figure 4.2: Illustration of *Radiant's* layer-saving mode of operation.

## 4.2 Program Structure & Subroutine Description

*Radiant's* program structure is illustrated in figure 4.3:



LAPACK/BLAS = LAPack and BLAS Subroutines from Oak Ridge National Laboratory  
 LINPACK/BLAS = LINPack and BLAS Subroutines from Oak Ridge National Laboratory  
 NUMLIB = Numlib Subroutines from Oak Ridge National Laboratory

Figure 4.3: *Radiant's* program structure.

where the subroutines listed perform the following functions (listed in alphabetical order):

- BUILD\_LAYER - BUILDS ONE LAYER OF ATMOSPHERE FOR THE PARAMETERS INPUT. THIS IS DONE THROUGH CONSTRUCTING THE GLOBAL TRANSMISSION AND REFLECTION MATRICES AND SOURCE VECTORS FOR THE LAYER.
- COMBINE\_GS - COMBINES THE GLOBAL SOURCE VECTORS FROM TWO SEPARATE LAYERS TO CONSTRUCT THOSE OF A COMBINED LAYER.
- COMBINE\_GT\_GR - COMBINES THE GLOBAL TRANSMISSION & REFLECTION MATRICES FROM TWO SEPARATE LAYERS TO CONSTRUCT THOSE OF A COMBINED LAYER.
- COMBINE\_LAYERS - COMBINES THE GLOBAL TRANSMISSION & REFLECTION MATRICES ALONG WITH EITHER THE GLOBAL SOLAR OR THERMAL SOURCE VECTORS FROM TWO SEPARATE LAYERS TO CONSTRUCT THOSE OF A COMBINED LAYER.
- COMBINE\_LAYERS\_2 - COMBINES THE GLOBAL TRANSMISSION & REFLECTION MATRICES ALONG WITH BOTH THE GLOBAL SOLAR AND THERMAL SOURCE VECTORS FROM TWO SEPARATE LAYERS TO CONSTRUCT THOSE OF A COMBINED LAYER.
- DATA\_INSPECTOR - INSPECTS THE SUITABILITY OF THE INPUT DATA PASSED TO RADIANT.
- DPLKAVG - COMPUTES THERMALLY-EMITTED RADIANCES USING THE PLANCK FUNCTION OF EMISSION INTEGRATED BETWEEN TWO WAVENUMBERS (A DISORT-DERIVED SUBROUTINE).
- INVERT - COMPUTES THE INVERSE OF A SQUARE MATRIX (USES LINPACK & BLAS SUBROUTINES).
- LOCAL - CONSTRUCTS THE LOCAL REFLECTANCE AND TRANSMITTANCE MATRICES AND SOURCE VECTORS FOR A LAYER OF ATMOSPHERE.
- MATDIAG - CONSTRUCTS A DIAGONAL MATRIX.
- MATIDENT - CONSTRUCTS AN IDENTITY MATRIX.
- MM\_D1GD2 - COMPUTES THE PRODUCT OF A GENERAL MATRIX "G" AND TWO DIAGONAL MATRICES D1 & D2 ( $D1 \times G \times D2$ ).
- MM\_IG1G2 - COMPUTES THE PRODUCT OF THE INVERSE OF A GENERAL MATRIX G1 AND A GENERAL MATRIX G2 (USES LINPACK & BLAS SUBROUTINES).

MV.DV - COMPUTES THE PRODUCT OF A DIAGONAL MATRIX "D" AND A VECTOR "V".

PLANCK - COMPUTES THERMALLY-EMITTED SPECTRAL RADIANCES USING THE PLANCK FUNCTION OF EMISSION.

PLEG - COMPUTES RENORMALIZED ASSOCIATED LEGENDRE POLYNOMIALS.

GETQUAD2 - COMPUTES THE POSITIVE QUADRATURE ROOTS AND ASSOCIATED WEIGHTS FOR (1) THE INTERVAL [-1,1] FOR EITHER GAUSS OR LOBATTO QUADRATURE OR (2) THE INTERVAL [0,1] FOR DOUBLE GAUSS QUADRATURE.

RAD - COMBINES ATMOSPHERIC LAYERS (IF APPLICABLE), APPLIES BOUNDARY CONDITIONS, AND COMPUTES RESULTING UPWELLING RADIANCES AT THE TOP OF THE LAYER OR BLOCK OF LAYERS AND UPWELLING AND DOWNWELLING RADIANCES AT THE BOTTOM OF THE LAYER OR BLOCK OF LAYERS.

RADIANT - ACTS AS THE INTERFACE BETWEEN RADIANT AND CALLING ROUTINE AND SERVES AS A PREPROCESSOR FOR SUBROUTINE RAD.

SGEEVX - COMPUTES THE EIGENVALUES AND ASSOCIATED EIGENVECTORS FOR THE COMPUTATION OF THE GLOBAL TRANSMISSION & REFLECTION MATRICES OF EACH ATMOSPHERIC LAYER CONSTRUCTED (USES LAPACK & BLAS SUBROUTINES).

SURF\_REF1.3 - RETURNS A MATRIX INDICATIVE OF THE REFLECTION CHARACTERISTICS OF A LAMBERTIAN SURFACE FOR A MODEL ATMOSPHERE/SURFACE SYSTEM.

SURF\_REF2.2 - RETURNS A MATRIX INDICATIVE OF THE REFLECTION CHARACTERISTICS OF A SURFACE READ IN FROM FILE FOR A MODEL ATMOSPHERE/SURFACE SYSTEM.

### 4.3 Input Parameters

*Radiant* and most of its subroutines are written in FORTRAN 90 with various Linear Algebra PACKage (LAPACK), LINear algebra PACKage (LINPACK), and Basic Linear Algebra Subprogram (BLAS) routines used which were originally written in FORTRAN 77. However, comment and continuation lines were modified early during model development so that the entire code could be viewed by a FORTRAN 90 compiler as an entirely FORTRAN 90 code. *Radiant* can be called within a FORTRAN program by a call statement such as:

```
CALL RADIANT( N, NUMLAY, VIEW_FLAG, VIEW_ANGLE, SRC_FLAG, FSUN, &  
             SZA, ITMS, ITMT, TTOP, TEMISS, TLEV, TSURF, PLANCK_FLAG, WVN, &  
             WVNLO, WVNHI, PHI, QUAD, DG, NUMDEG, DELTAZ, SIGS_CLD, &  
             SIGE_CLD, SIGS_AERO, SIGE_AERO, SIGS_RAY, SIGE_GAS, G1_CLD, &  
             G2_CLD, G_CLD, G1_AERO, G2_AERO, G_AERO, PF_CLD, PF_CLD_FILE, &  
             PF_AERO, PF_AERO_FILE, DELTA_M_CLD, DELTA_M_AERO, SURF, &  
             ALBEDO, SURF_FILE, WVN_FLAG, KFLAG, FLUXES, WARNING_FLAG, &  
             P_IO, R_IO, FLUX_IO, L_IO, B_IO, INV_IO, ITM, IBMTOT, IBPTOT, ITPTOT, &  
             FIT, FOT, FOB, FIB )
```

The following is a brief explanation of the parameters that are found above in *Radiant's* current argument list (listed alphabetically). Note that since *Radiant* builds a block of atmospheric layers from the top down, each vector of dimension NUMLAY should be defined such that the FIRST element of those vectors corresponds to the value the user desires for the TOP layer in a block of layers.

ALBEDO - (input) DOUBLE PRECISION  
SURFACE ALBEDO WHEN SURF = 1.

B.IO - (input) INTEGER  
I/O FLAG FOR TOGGING THE DISPLAY OF CERTAIN DATA FROM  
SUBROUTINE "BUILD\_LAYER" ON OR OFF.  
0 = OFF  
1 = ON



DELTA\_M\_AERO - (input) INTEGER, DIMENSION(NUMLAY)  
 VECTOR OF FLAGS TO INDICATE WHETHER OR NOT DELTA-M SCALING  
 IS USED IN SUBROUTINE "LOCAL" FOR AEROSOL. DELTA-M SCALING  
 ALLOWS THE PHASE FUNCTION TO ASSUME A SHARPER FORWARD  
 PEAK (SEE Wiscombe (1977))  
 0 = NO  
 1 = YES

DELTA\_M\_CLD - (input) INTEGER, DIMENSION(NUMLAY)  
 VECTOR OF FLAGS TO INDICATE WHETHER OR NOT DELTA-M SCALING  
 IS USED IN SUBROUTINE "LOCAL" FOR CLOUD. DELTA-M SCALING  
 ALLOWS THE PHASE FUNCTION TO ASSUME A SHARPER FORWARD  
 PEAK (SEE Wiscombe (1977))  
 0 = NO  
 1 = YES

DELTAZ - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
 VECTOR CONTAINING GEOMETRIC THICKNESSES OF LAYERS UNDER  
 COMPUTATION.

DG - (input) INTEGER  
 FLAG TO INDICATE WHICH VERSION OF GAUSS QUADRATURE WILL BE  
 USED IN SUBROUTINE LOCAL IF QUAD = 0 IS SELECTED.  
 0 = NORMAL GAUSS  
 1 = DOUBLE GAUSS

FIB - (output) DOUBLE PRECISION  
 FLUX INCOMING AT THE BOTTOM BOUNDARY OF THE ATMOSPHERE.

FIT - (output) DOUBLE PRECISION  
 FLUX INCOMING AT THE TOP BOUNDARY OF THE ATMOSPHERE.

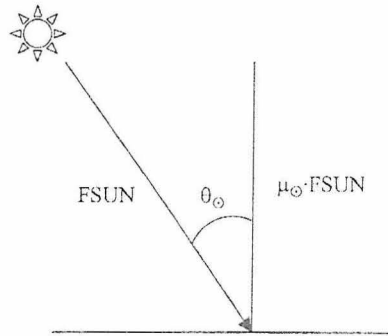
FOB - (output) DOUBLE PRECISION  
 FLUX OUTGOING AT THE BOTTOM BOUNDARY OF THE ATMOSPHERE.

FOT - (output) DOUBLE PRECISION  
 FLUX OUTGOING AT THE TOP BOUNDARY OF THE ATMOSPHERE.

FLUX\_IO - (input) INTEGER  
 I/O FLAG FOR TOGGING THE DISPLAY OF FLUX CONSERVATION TEST  
 DATA FROM SUBROUTINE "RAD" ON OR OFF WHEN THE AZIMUTHAL  
 COMPONENT OF THE RADIANCE FIELD IS COMPUTED.  
 0 = OFF  
 1 = ON

FLUXES - (input) INTEGER  
 FLAG TO PERMIT RADIANT TO GENERATE FLUXES/SPECTRAL FLUXES  
 FROM COMPUTED RADIANCES/SPECTRAL RADIANCES.  
 0 = OFF  
 1 = ON

FSUN - (input) DOUBLE PRECISION  
SOLAR FLUX AT THE TOP OF THE ATMOSPHERE AT A GIVEN WAVENUMBER.  
FOR CLARITY, FSUN IS DEFINED AS INDICATED IN THE FOLLOWING  
FIGURE.



G1\_AERO - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF AEROSOL ASYMMETRY PARAMETERS FOR DOUBLE  
HENYEY-GREENSTEIN - FORWARD SCATTERING DIRECTION.

G2\_AERO - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF AEROSOL ASYMMETRY PARAMETERS FOR DOUBLE  
HENYEY-GREENSTEIN - BACKWARD SCATTERING DIRECTION.

G\_AERO - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF AEROSOL ASYMMETRY PARAMETERS FOR DOUBLE  
HENYEY-GREENSTEIN - EFFECTIVE ASYMMETRY PARAMETER.

G1\_CLD - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF CLOUD ASYMMETRY PARAMETERS FOR DOUBLE  
HENYEY-GREENSTEIN - FORWARD SCATTERING DIRECTION.

G2\_CLD - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF CLOUD ASYMMETRY PARAMETERS FOR DOUBLE  
HENYEY-GREENSTEIN - BACKWARD SCATTERING DIRECTION.

G\_CLD - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF CLOUD ASYMMETRY PARAMETERS FOR DOUBLE  
HENYEY-GREENSTEIN - EFFECTIVE ASYMMETRY PARAMETER.

IBMTOT - (output) DOUBLE PRECISION, DIMENSION(N)  
VECTOR OF TOTAL DOWNWELLING DIFFUSE RADIANCES AT THE  
BOTTOM OF THE ATMOSPHERE (OR LAYER OR BLOCK OF LAYERS)  
FROM ALL FOURIER COMPONENTS COMPUTED.

IBPTOT - (output) DOUBLE PRECISION, DIMENSION(N)  
VECTOR OF TOTAL UPWELLING DIFFUSE RADIANCE AT THE BOTTOM OF  
THE ATMOSPHERE (OR LAYER OR BLOCK OF LAYERS)  
FROM ALL FOURIER COMPONENTS COMPUTED.

INV\_IO - (input) INTEGER  
 I/O FLAG FOR TOGGING THE DISPLAY OF CERTAIN DATA FROM  
 SUBROUTINE "INVERT" (WHICH COMPUTES THE INVERSE OF A MATRIX)  
 ON OR OFF.  
 0 = OFF  
 1 = ON

ITMS - (input) DOUBLE PRECISION, DIMENSION(N)  
 VECTOR OF DOWNWELLING DIFFUSE SOLAR RADIANCE AT THE TOP OF  
 THE LAYER OR BLOCK OF LAYERS IF A MID-ATMOSPHERIC LAYER OR  
 BLOCK OF LAYERS IS BEING SIMULATED (SHOULD BE A VECTOR  
 OF ZEROS FOR A FULL ATMOSPHERIC SCENARIO).

ITMT - (input) DOUBLE PRECISION, DIMENSION(N)  
 VECTOR OF DOWNWELLING DIFFUSE THERMAL RADIANCE AT THE TOP OF  
 THE LAYER OR BLOCK OF LAYERS IF A MID-ATMOSPHERIC LAYER OR  
 BLOCK OF LAYERS IS BEING SIMULATED (SHOULD BE A VECTOR  
 OF ZEROS FOR A FULL ATMOSPHERIC SCENARIO).

ITPTOT - (output) DOUBLE PRECISION, DIMENSION(N)  
 VECTOR OF TOTAL UPWELLING DIFFUSE RADIANCE AT THE TOP OF  
 THE ATMOSPHERE (OR LAYER OR BLOCK OF LAYERS)  
 FROM ALL FOURIER COMPONENTS COMPUTED.

KFLAG - (input) INTEGER  
 FLAG INDICATING WHETHER OR NOT LAYER-SAVING MODE IS  
 ACTIVE (E.G. WHEN COMPUTING ELEMENTS OF A JACOBIAN BY  
 FINITE DIFFERENCE).  
 0 = NO (NORMAL MODE)  
 1 = YES (LAYER-SAVING MODE)

**NOTE: KFLAG IS USED IN CONJUNCTION WITH WVN\_FLAG.  
 SEE TABLE BELOW FOR APPROPRIATE SETTINGS.**

Table 4.1: KFLAG and WVN\_FLAG settings to use different operational modes in *Radiant*.

Description	WVN_FLAG Setting	KFLAG Setting
Compute radiances only	NEW	0
Compute radiances; save layers for next call	NEW	1
Compute radiances using saved layers	OLD	1

L\_IO - (input) INTEGER  
 I/O FLAG FOR TOGGING THE DISPLAY OF CERTAIN DATA FROM  
 SUBROUTINE "LOCAL" ON OR OFF.  
 0 = OFF  
 1 = ON

N - (input) INTEGER  
 NUMBER OF UPWARD (OR DOWNWARD) STREAMS (RANGE: HAS BEEN TESTED  
 FROM N = 4 (8-STREAM) UP TO N = 126 (256-STREAM)).

NUMDEG - (input) INTEGER  
TOTAL NUMBER OF FOURIER COMPONENTS TO BE COMPUTED - 1  
(E.G. FOR COMPUTING THE RADIANCE FIELD FOR DEGREES 0  
THRU 9, NUMDEG = 9. FOR ONLY AZIMUTHAL COMPONENT OF  
RADIANCE FIELD, NUMDEG = 0).

NUMLAY - (input) INTEGER  
NUMBER OF LAYERS IN THE MODEL ATMOSPHERE.

P\_IO - (input) INTEGER  
I/O FLAG FOR TOGGING THE DISPLAY OF CERTAIN DATA FROM  
SUBROUTINE "RADIANT" (THE PREPROCESSOR SUBROUTINE) ON OR OFF.  
0 = OFF  
1 = ON

PF\_AERO - (input) INTEGER, DIMENSION(NUMLAY)  
VECTOR OF FLAGS TO DETERMINE HOW COEFFICIENTS OF AEROSOL  
PHASE FUNCTION EXPANSION ARE OBTAINED.  
0 = NO PHASE FUNCTION  
1 = DOUBLE HENYEY-GREENSTEIN PHASE FUNCTION  
2 = USER-DEFINED PHASE FUNCTION READ IN FROM FILE

PF\_AERO\_FILE - (input) CHARACTER\*7, DIMENSION(NUMLAY)  
VECTOR OF FILE INDICES CONTAINING COEFFICIENTS OF AEROSOL PHASE  
FUNCTION EXPANSION (USED IF P\_AERO = 2).

PF\_CLD - (input) INTEGER, DIMENSION(NUMLAY)  
VECTOR OF FLAGS TO DETERMINE HOW COEFFICIENTS OF CLOUD  
PHASE FUNCTION EXPANSION ARE OBTAINED.  
0 = NO PHASE FUNCTION  
1 = DOUBLE HENYEY-GREENSTEIN PHASE FUNCTION  
2 = USER-DEFINED PHASE FUNCTION READ IN FROM FILE

PF\_CLD\_FILE - (input) CHARACTER\*7, DIMENSION(NUMLAY)  
VECTOR OF FILE INDICES CONTAINING COEFFICIENTS OF CLOUD PHASE  
FUNCTION EXPANSION (USED IF PF\_CLD = 2).

PHI - (input) DOUBLE PRECISION  
THE AZIMUTHAL ANGLE FOR WHICH THE RADIANCE VECTOR IS  
COMPUTED (IN degrees). THE AZIMUTH ANGLE OF THE SUN IS  
TAKEN TO BE ZERO.

PLANCK\_FLAG - (input) INTEGER  
FLAG TO CHOOSE WHETHER SPECTRAL RADIANCES OR RADIANCES  
WILL BE USED FOR THERMAL SOURCES.  
1 = SPECTRAL RADIANCES (in  $mW/(m^2cm^{-1}sr)$ )  
2 = RADIANCES (in  $mW/(m^2sr)$ )

**NOTE: WHEN USING THERMAL SOURCES, THE OUTPUT WILL  
HAVE THE ABOVE UNITS DEPENDING ON THE VALUE OF  
PLANCK\_FLAG. THUS, WHEN USING SOLAR SOURCES WITH  
THERMAL SOURCES, INSURE THAT THE SOLAR INPUT HAS UNITS**

OF  $mW/(m^2cm^{-1})$  (SPECTRAL FLUX) WHEN PLANCK\_FLAG = 1  
AND  $mW/m^2$  (FLUX) WHEN PLANCK\_FLAG = 2.

- QUAD - (input) INTEGER  
FLAG TO INDICATE WHICH QUADRATURE SCHEME WILL BE UTILIZED  
TO DETERMINE ANGLES FOR WHICH RADIANCES IN THE RADIANCE  
FIELD ARE EXPLICITLY COMPUTED.  
0 = GAUSS (SEE VARIABLE "DG" FOR MORE ON THIS OPTION)  
1 = LOBATTO
- R.IO - (input) INTEGER  
I/O FLAG FOR TOGGING THE DISPLAY OF CERTAIN DATA FROM  
SUBROUTINE "RAD" ON OR OFF.  
0 = OFF  
1 = ON
- SIGE\_AERO - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF EXTINCTION COEFFICIENTS FOR AEROSOL  
(IN UNITS OF INVERSE LENGTH).
- SIGE\_CLD - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF EXTINCTION COEFFICIENTS FOR CLOUD  
(IN UNITS OF INVERSE LENGTH).
- SIGE\_GAS - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF EXTINCTION COEFFICIENTS FOR ATMOSPHERIC GAS  
(IN UNITS OF INVERSE LENGTH).
- SIGS\_AERO - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF SCATTERING COEFFICIENTS FOR AEROSOL  
(IN UNITS OF INVERSE LENGTH).
- SIGS\_CLD - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF SCATTERING COEFFICIENTS FOR CLOUD  
(IN UNITS OF INVERSE LENGTH).
- SIGS\_RAY - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)  
VECTOR OF SCATTERING COEFFICIENTS FOR RAYLEIGH SCATTER  
(IN UNITS OF INVERSE LENGTH).
- SRC\_FLAG - (input) INTEGER  
FLAG INDICATING TYPE OF SOURCES INCLUDED IN COMPUTING  
RADIANCES.  
1 = SOLAR ONLY  
2 = SOLAR AND THERMAL  
3 = THERMAL ONLY
- SURF - (input) INTEGER  
FLAG INDICATING TYPE OF REFLECTING SURFACE.  
0 = NO UNDERLYING SURFACE  
1 = LAMBERTIAN SURFACE  
2 = USER-DEFINED SURFACE. BIDIRECTIONAL REFLECTIVITY  
FUNCTION READ IN FROM FILE

SURF\_FILE - (input) CHARACTER\*7  
 INDEX OF FILE FROM WHICH COEFFICIENTS FOR LEGENDRE POLYNOMIAL  
 EXPANSION OF SURFACE BIDIRECTIONAL REFLECTIVITY ARE OBTAINED  
 (USED IF SURF = 2).

SZA - (input) DOUBLE PRECISION  
 THE SOLAR ZENITH ANGLE (in degrees from zenith).

TEMISS - (input) DOUBLE PRECISION  
 THERMAL EMISSIVITY OF THE TOP BOUNDARY.

TLEV - (input) DOUBLE PRECISION, DIMENSION(NUMLAY+1)  
 VECTOR OF TEMPERATURES AT BOUNDARIES OF ATMOSPHERIC LAYERS  
 (in K).

TSURF - (input) DOUBLE PRECISION  
 THE TEMPERATURE OF THE SURFACE - DISTINCT FROM TEMPERATURE  
 AT BOTTOM OF BOTTOM ATMOSPHERIC LAYER (in K).

TTOP - (input) DOUBLE PRECISION  
 TEMPERATURE OF TOP BOUNDARY - DISTINCT FROM TEMPERATURE  
 AT TOP OF TOP ATMOSPHERIC LAYER (in K).

VIEW\_ANGLE - (input) DOUBLE PRECISION  
 USER-DEFINED VIEW ANGLE (in degrees from zenith).

VIEW\_FLAG - (input) INTEGER  
 FLAG INDICATING IF USER-DEFINED VIEW ANGLE IS IN USE.  
 0 = OFF  
 1 = ON

WARNING\_FLAG - (input) INTEGER  
 I/O FLAG FOR TOGGING THE DISPLAY OF CERTAIN WARNING DATA  
 ON OR OFF WHEN READING PHASE FUNCTION DATA FOR AN ATMOSPHERIC  
 LAYER OR BIDIRECTIONAL REFLECTIVITY DATA FOR THE SURFACE FROM  
 EXTERNAL FILES.  
 0 = OFF  
 1 = ON

WVN - (input) DOUBLE PRECISION  
 WAVENUMBER USED FOR COMPUTATION OF PLANCK FUNCTION OF  
 EMISSION WHEN THERMAL SOURCES ARE ACTIVE (I.E. SRC\_FLAG  
 = 2 OR 3) AND PLANCK\_FLAG = 1.

WVN\_FLAG - (input) CHARACTER\*3  
 FLAG TO INDICATE WHETHER OR NOT SAVED MATRICES WILL  
 BE USED TO COMPUTE RADIANCES ON CURRENT CALL TO RADIANT  
 (SEE NOTE UNDER "KFLAG").  
 'NEW' = NO  
 'OLD' = YES

WVNHI - (input) DOUBLE PRECISION  
WAVENUMBER USED FOR UPPER LIMIT IN INTEGRATION OF  
PLANCK FUNCTION OF EMISSION WHEN THERMAL SOURCES ARE  
ACTIVE (I.E. SRC\_FLAG = 2 OR 3) AND PLANCK\_FLAG = 2.

WVNLO - (input) DOUBLE PRECISION  
WAVENUMBER USED FOR LOWER LIMIT IN INTEGRATION OF  
PLANCK FUNCTION OF EMISSION WHEN THERMAL SOURCES ARE  
ACTIVE (I.E. SRC\_FLAG = 2 OR 3) AND PLANCK\_FLAG = 2.

## 4.4 Special Program Provisions

### 4.4.1 Singularity Busting

Due to the nature of the discrete ordiante method, *Radiant* has to deal with some of the same singularity issues as DISORT. These singularities are observed when doing computations (1) in a situation where the cosine of the solar zenith angle ( $\mu_{\odot}$ ) is equal to the cosine of one of the quadrature angles ( $\mu_i$ ) used for numerical integration or the cosine of the user-defined viewing angle ( $\mu_{user}$ ) and (2) in a conservatively scattering medium (i.e. single scatter albedo  $\omega_0 = 1$ ). The first will be referred to as a " $\mu_{\odot}$ -singularity" and the second as an " $\omega_0$ -singularity". In *Radiant*, these singularity types are dealt with as described below.

To avoid  $\mu_{\odot}$ -singularities,  $\mu_{\odot}$  is compared to each  $\mu_i$  and  $\mu_{user}$  (if used). If the absolute value of the difference between  $\mu_{\odot}$  and any of the  $\mu_i$  or  $\mu_{user}$  is less than  $10^{-11}$ , then  $10^{-10}$  is added or subtracted from  $\mu_{\odot}$  as appropriate to avoid program disruption. When this occurs, the following message is displayed to alert the user that this event has occurred

SZA PERTURBED TO AVOID SINGULARITY. NOW, SZA = (adjusted  $\mu_{\odot}$  value)

Since *Radiant* employs double precision in its computations, this adjustment should not adversely affect the outcome of the resulting computations.

To avoid  $\omega_0$ -singularities, the following tests are performed for each layer in the medium and the corresponding procedures employed for the corresponding situations:

CASE #1: When both cloud and aerosol extinction are present, but atmospheric gas extinction is not ( $\sigma_{e,clld} > 10^{-12}$ ,  $\sigma_{e,aer} > 10^{-12}$ , and  $\sigma_{e,gas} \leq 10^{-12}$ ), then the single scatter albedo for both cloud and aerosol are summed. If the sum is greater than 1.9999999999, then  $\sigma_{s,aer}$  is multiplied by 0.9999999999 to avoid an  $\omega_0$ -singularity and the following message is displayed to alert the user that this event has occurred.



SIGS\_AERO PERTURBED TO AVOID SINGULARITY.

NOW, SIGS\_AERO = (adjusted  $\sigma_{s,aer}$  value) FOR LAYER (layer no. of affected layer)

CASE #2: When cloud extinction is present, but aerosol and atmospheric gaseous extinction are not ( $\sigma_{e,clد} > 10^{-12}$ ,  $\sigma_{e,aer} \leq 10^{-12}$ , and  $\sigma_{e,gas} \leq 10^{-12}$ ), then the single scatter albedo for cloud is evaluated. If it is greater than 0.9999999999, then  $\sigma_{s,clد}$  is multiplied by 0.9999999999 to avoid an  $\omega_0$ -singularity and the following message is displayed to alert the user that this event has occurred.

SIGS\_CLD PERTURBED TO AVOID SINGULARITY.

NOW, SIGS\_CLD = (adjusted  $\sigma_{s,clد}$  value) FOR LAYER (layer no. of affected layer)

CASE #3: When aerosol extinction is present, but cloud and atmospheric gaseous extinction are not ( $\sigma_{e,clد} \leq 10^{-12}$ ,  $\sigma_{e,aer} > 10^{-12}$ , and  $\sigma_{e,gas} \leq 10^{-12}$ ), then the single scatter albedo for aerosol is evaluated. If it is greater than 0.9999999999, then  $\sigma_{s,aer}$  is multiplied by 0.9999999999 to avoid an  $\omega_0$ -singularity and a message like in case #1 is displayed to alert the user that this event has occurred.

In addition to the above tests, a test is also performed for the case when neither cloud, aerosol, or gaseous extinction is present and when one is operating in the solar portion of the EM spectrum where Rayleigh scattering is present. Here, we have  $\omega_{0,ray} = 1$ . In this case, we set  $\omega_{0,ray} = 0.9999999999$  to avoid an  $\omega_0$ -singularity.

#### 4.4.2 Limitations Imposed by Some Time-Saving Features

When *Radiant* is called to perform a given set of computations, the variables N, NUMDEG, SZA, and NUMLAY must not be changed during that set of computations. This is because

- (1) N is used to determine the size of matrices when a given number of streams is employed. Some of these matrices are saved for potential reuse as each layer in the medium is built so that certain computations do not have to be needlessly repeated.
- (2) NUMDEG is used to determine how many orders of associated Legendre

polynomials are needed for a required set of radiance (or flux) computations. Because associated Legendre polynomials of higher order can be efficiently computed by using those of lower order by way of a recurrence relation, all the needed polynomials are computed "up front" for a given computation and saved in an appropriately-sized 3-D array.

(3) SZA is also needed to compute all the associated Legendre polynomials needed for a required set of radiance (or flux) computations. Thus, it may not be changed without conflicting with the originally computed set of associated Legendre polynomials.

(4) NUMLAY is used to determine the number of layers in the medium. If *Radiant* is called in layer-saving mode and NUMLAY has been changed, a conflict will arise because certain multi-dimensional arrays have been defined using NUMLAY and saved for subsequent reuse in layer-saving mode. It MIGHT be alright to change NUMLAY during a set of computations if not using layer-saving mode, but this has not been thoroughly explored yet. Thus, this restriction is in place until more thorough testing can be performed in this area.

If one of these variables is changed during a set of computations, the user will receive the following error message:

```
ERROR: ONCE RADIANT HAS BEEN CALLED THE FIRST TIME,  
N, NUMDEG, SZA, AND NUMLAY MUST REMAIN CONSTANT FOR ALL  
SUBSEQUENT CALLS TO RADIANT FOR THE CURRENT RUN TO  
MAINTAIN CONSISTENCY WITH CERTAIN COMPUTATIONAL  
EFFICIENCY FEATURES IN RADIANT'S ALGORITHMS
```

Along with this, information is provided regarding which of the four variables has changed.

# Appendix A

## Global Transmission and Reflection Matrices: Numerical Considerations

In chapter 3, it was asserted that the global transmission ( $T$ ) and global reflection ( $R$ ) matrices can be obtained from the expressions

$$\begin{aligned} T(H, 0) = & -u_+[E - (u_+^{-1}u_-)^2][(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)}] \\ & \{E - [(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)}]^2\}^{-1}u_-^{-1} \end{aligned} \quad (\text{A.1})$$

$$\begin{aligned} R(H, 0) = & -u_+[E - (u_+^{-1}u_-)e^{-\Lambda^+\tau(H)}(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)} \\ & \{E - [(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)}]^2\}^{-1}u_-^{-1} \end{aligned} \quad (\text{A.2})$$

where  $u_+$  and  $u_-$  are matrices, when appropriately assembled, composing the matrix  $X$  from equation (2.13),  $\Lambda^+$  is a diagonal matrix containing the positive eigenvalues of the matrix  $A$ ,  $\tau(H)$  is the optical depth of a layer of thickness  $H$ , and  $E$  is the identity matrix.

If one uses the above formulation for  $T$  exactly, one will need to perform eight matrix multiplications and four matrix inversions. This amounts to twelve matrix operations proportional to  $n^3$ . Similarly, if the exact expression for  $R$  is used, this leads to an additional four matrix multiplications (assuming that some of the matrix products used in calculating  $T$  are used again so as to avoid unnecessary recomputation). This leads to a total of sixteen  $n^3$  operations; however, there are some substitutions and simplifications that can be done to the above expressions. Employing these techniques leads to the following equivalent expressions for  $T$  and  $R$ :

$$T(H, 0) = [u_- - u_+ u_-^{-1} u_+] e^{-\Lambda + \tau(H)} \{E - [u_-^{-1} u_+ e^{-\Lambda + \tau(H)}]^2\}^{-1} u_-^{-1} \quad (\text{A.3})$$

$$R(H, 0) = [u_- e^{-\Lambda + \tau(H)} u_-^{-1} u_+ e^{-\Lambda + \tau(H)} - u_+] \{E - [u_-^{-1} u_+ e^{-\Lambda + \tau(H)}]^2\}^{-1} u_-^{-1} \quad (\text{A.4})$$

If one carefully observes,  $T$  now only requires seven matrix multiplications and two matrix inversions and  $R$  an additional three matrix multiplications leading to a total of twelve  $n^3$  operations to obtain both these matrices. Furthermore, if one employs an  $A^{-1}B$  algorithm (an algorithm in which both the inverse of the matrix A and the multiplication of it by matrix B are both done at the same time), one can save an additional two  $n^3$  operations; thus, by trimming some of the "numerical fat" as it were, one can save six  $n^3$  operations every time these very heavily used matrices need recomputed and lowers the total number of  $n^3$  operations required to ten. The formulations for  $T$  and  $R$  given in Benedetti et al. (2002) help make *Radiant* a more numerically stable code for higher optical depths while the above modifications help make it more efficient.

# Bibliography

- Benedetti, A., P. Gabriel, and G. Stephens: 2002, Properties of reflected sunlight derived from a Green's function method. *J. Quant. Spectrosc. Radiat. Transfer*, **72**, 201–225.
- Gabriel, P., S. Lovejoy, A. Davis, D. Schertzer, and G. Austin: 1990, Discrete angle radiative transfer 2. Renormalization approach for homogeneous and fractal clouds. *J. Geophys. Res.*, **95**, 11717–11728.
- Greenwald, T. and G. Stephens: 1988, Application of a doubling-adding radiation model to visibility problems. Technical report, Cooperative Institute for Research in the Atmosphere.
- Miller, S., G. Stephens, C. Drummond, A. Heidinger, and P. Partain: 2000, A multi-sensor diagnostic satellite cloud property retrieval scheme. *J. Geophys. Res.*, **105**, 19955–19971.
- Stamnes, K. and P. Conklin: 1984, A new multi-layer discrete ordinate approach to radiative transfer in vertically inhomogeneous atmospheres. *J. Quant. Spectrosc. Radiat. Transfer*, **31**, 273–282.
- Stamnes, K. and R. Swanson: 1981, A new look at the discrete ordinate method for radiative transfer calculations in anisotropically scattering atmospheres. *J. Atmos. Sci.*, **38**, 387–399.
- Stamnes, K., S. Tsay, and T. Nakajima: 1988, Computation of eigenvalues and eigenvectors for the discrete ordinate and matrix operator methods in radiative transfer. *J. Quant. Spectrosc. Radiat. Transfer*, **39**, 415–419.
- Vandehulst, H.: 1980, *Multiple Light Scattering: Tables, Formulas, and Applications, Vol. I and II*. Academic Press, New York, New York.
- Wiscombe, W.: 1977, The delta-m method: rapid yet accurate radiative flux calculations for strongly asymmetric phase functions. *J. Atmos. Sci.*, **34**, 1408–1422.