Radiant 1.0: A User's Guide

By
Matt Christi and Philip Gabriel

Department of Atmospheric Science
Colorado State University
Fort Collins, Colorado

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

Colorado
State
University

Department of
Atmospheric Science

Paper No. 732

RADIANT 1.0: A USER'S GUIDE

Matt Christi and Philip Gabriel

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

Radiant 1.0: A User's Guide

Matt Christi and Philip Gabriel

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. 732

March 2003

Contents

1 Why Another Radiative Transfer Solver? 1

1.1 Introduction 1

1.2 Doubling/Adding and DISORT 2

1.3 Our Goal 3

2 Solving the Radiative Transfer Equation 5

2.1 Radiative Transfer Theory 5

2.2 Solving the Radiative Transfer Equation 7

2.3 Radiant: An Efficient Approach to Computing Radiative Transfer 10

3 Computation Modes 13

3.1 Modified Eigenmatrix Method 13

3.2 Truncated Series Method 15

4 Radiant, Doubling/Adding, and DISORT: Accuracy and Timing Comparisons 17

4.1 Accuracy Comparisons 17

4.2 Timing Comparisons 19

5 Using Radiant 23

5.1 Operational Modes 23

5.2 Program Structure & Subroutine Description 25

5.3 Input Parameters 28

5.4 Sample Simulations 33

A Global Transmission and Reflection Matrices: Numerical Considerations 36

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

4.1 Results of a speed comparisòn between the eigenmatrix method in *Radiant* and the doubling method. 20

4.2 Results of a speed comparison between *Radiant* and DISORT. 21

5.1 Illustration of *Radiant's* normal mode of operation. 24

5.2 Illustration of *Radiant's* layer-saving mode of operation. 24

5.3 Illustration of *Radiant's* program structure. 25

List of Tables

1.1 Sample of organizations or authors and some of their radiative transfer codes. 2

4.1 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. 18

4.2 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. 18

4.3 Comparison of radiances generated by a doubling/adding scheme (D/A) with those generated by *Radiant* (R). 19

5.1 KFLAG and WVN_FLAG settings to use different operational modes in *Radiant*. 30

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 eigenmatix 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 two modes in which *Radiant* performs RT computations followed in chapter 4 by the results of speed and accuracy comparisons with both doubling/adding and DISORT. In chapter 5, which will probably the most beneficial to users, a description of *Radiant's* operational modes is provided along with information regarding its program structure and input parameters. Some examples are also provided to illustrate the setting of the parameters for a few simple scenarios.

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, σ_e, σ_s, and σ_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:

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

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

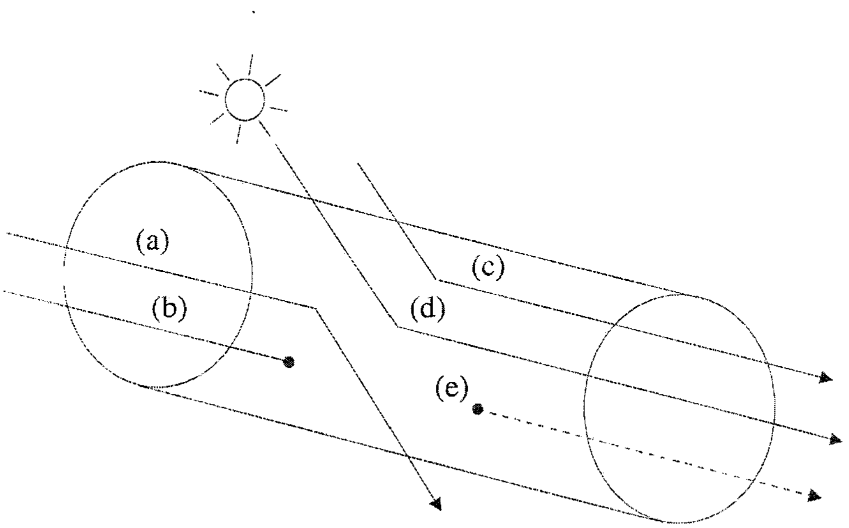


Figure 2.1: Physical processes affecting the transfer of radiation through a medium. These processes are (a) out-scattering, (b) absorption, (c) and (d) in-scattering of diffuse and direct radiation, respectively, and (e) emission.

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 Solving the Radiative Transfer Equation

An overall description of the method used to compute radiances in *Radiant* model 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 $2n \geq N + 1$ equations when using Gaussian quadrature or $2n \geq N + 3$ equations when using Lobatto quadrature to represent the radiation field and allow the integrals to be computed exactly. Here, N is assumed to be odd 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

$$I(z, \mu, \phi) = \Sigma_{m=0}^{M} I_m(z, \mu) cos[m(\phi - \phi_{\odot})] \tag{2.2}$$

where z is altitude, μ is the cosine of the observation angle in reference to zenith, and ϕ 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 set of matrix equations, one for each m in the Fourier expansion of I:

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

Here, we have:

${}^mI^{\pm}$ - A vector describing the mth term in the Fourier expansion of I where (+) represents that portion of the vector representing upwelling radiances and (-) that portion representing downwelling radiances.

${}^mP^{\pm}$ and ${}^mP_{\odot}^{\pm}$ - 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.

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

$B(T(z))$ - The Planck function (assumed constant within a layer).

σ_a - The absorption coefficient of the medium.

σ_e - The extinction coefficient of the medium.

δ_{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

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

where

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

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

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

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

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

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 Σ, respectively, we arrive at the more concise expression

$$\frac{d}{dz}I = AI + \Sigma \tag{2.9}$$

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

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

where H is a fixed altitude above sea level.

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

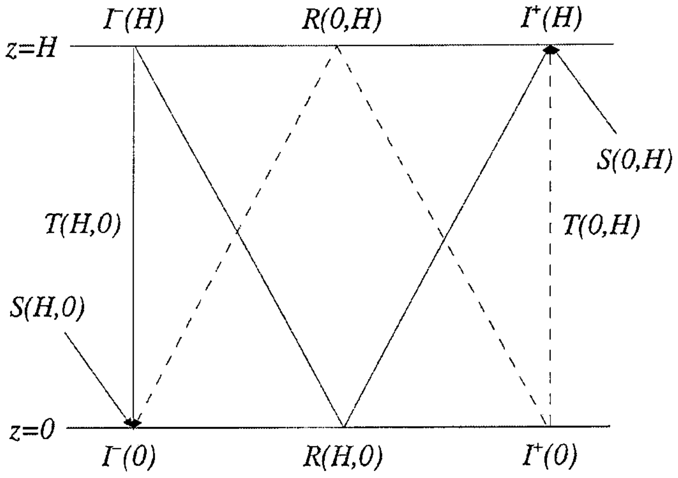


Figure 2.2: The interaction principle illustrated for a homogeneous layer where T(H,0) and R(H,0) are the downwelling global transmission and reflection matrices and T(0,H) and R(0,H) their upwelling counterparts. S(H,0) is the downwelling source vector and $S(0, H)$ the upwelling source vector, respectively.

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

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

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

where $I^+(H)$ is the upwelling radiation at the top of the atmosphere, $I^-(0)$ is the downwelling radiation at the surface, T and R are the global transmission and reflection matrices for the atmospheric state, 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 currently used to describe the influence of nature on radiant energy entering the atmosphere at wavelengths in the solar portion of the 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 has two computational modes for performing radiative transfer. The primary mode 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 other mode uses a truncated series approach for building very optically thin layers (those with $\tau < 0.003$). This will be referred to as the truncated series method (TSM). The rationale for the second mode is to assist *Radiant* in obtaining the fastest possible solutions for even these very small optical depths. This is needed because the eigenmatrix approach, being insensitive to optical depth, always takes the same amount of time to compute a given layer. For the vast majority of media, the eigenmatrix method will be faster than doubling; however, for $\tau < 0.003$, the doubling method is faster (see section 4.2 for timing results). This ensures faster layer-building regardless of optical depth.

No matter which method is used, MEM or TSM, 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. The MEM and TSM computational modes are described in more detail in the next chapter.

Chapter 3

Computation Modes

3.1 Modified Eigenmatrix Method

The eigenmatrix method, as implemented in *Radiant*, can be used to derive the T and R matrices for layers of any optical depth experienced in the real atmosphere. Again, this process will be referred to as the modified eigenmatrix method (MEM).

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

$$e^{A\tau} = Xe^{\Lambda\tau}X^{-1} \tag{3.1}$$

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 τ 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)

$$\begin{aligned} T(H,0) &= -u_+[E-(u_+^{-1}u_-)^2][(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)}] \\ &\quad \{E-[(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)}]^2\}^{-1}u_-^{-1} \end{aligned} \tag{3.2}$$

$$\begin{aligned} R(H,0) &= -u_+[E-(u_+^{-1}u_-)e^{-\Lambda^+\tau(H)}(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)} \\ &\quad \{E-[(u_+^{-1}u_-)^{-1}e^{-\Lambda^+\tau(H)}]^2\}^{-1}u_-^{-1} \end{aligned} \tag{3.3}$$

where u_+ and u_- are matrices, when appropriately assembled, composing the matrix X, Λ^+ is a diagonal matrix containing the positive eigenvalues of the matrix A, $\tau(H)$ is the optical depth of a given layer of 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) can be rendered

$$S(0,H) = -T(0,H)S_1^+ - R(H,0)S_2^- + S_2^+ \tag{3.4}$$

$$S(H,0) = -R(0,H)S_1^+ - T(H,0)S_2^- + S_1^- \tag{3.5}$$

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

$$S_1^{\pm} = (\frac{I}{\mu_{\odot}} - A)^{-1}\frac{\omega_o}{4\pi}F_{\odot}M^{-1}P_{\odot}^{\mp}e^{-\frac{\tau}{\mu_{\odot}}} \tag{3.6}$$

$$S_2^{\pm} = (\frac{I}{\mu_{\odot}} - A)^{-1}\frac{\omega_o}{4\pi}F_{\odot}M^{-1}P_{\odot}^{\mp} \tag{3.7}$$

Here, $\mu_{\odot}$, A, $F_{\odot}$, M^{-1}, and $P_{\odot}^{\mp}$ are the same as in the previous chapter, ω_o is the single scatter albedo, and E is again the identity matrix.

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 discussion of this and the resulting expressions for global T and R are given in appendix A.

3.2 Truncated Series Method

The truncated series method, as implemented in *Radiant*, can be used to derive T and R matrices for layers whose optical depths are less than $\tau = 0.08$. Again, this process is denoted as the TRM. Its benefits and limitations were explored as a project by graduate student Brian McNoldy in a PhD-level course in radiative transfer at the CSU Department of Atmospheric Science. The method allows the computation of T and R matrices using $(n/2)^3$ operations rather than the n^3 operations required by the MEM.

The basic concept behind TRM is to do a series expansion of the exponential matrix and truncate it at an appropriate number of terms for a given accuracy. We start with

$$\begin{aligned} e^{A\tau} &= e^{\begin{bmatrix} t & -r \\ r & -t \end{bmatrix}\tau} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} t & -r \\ r & -t \end{bmatrix}\tau + \begin{bmatrix} t & -r \\ r & -t \end{bmatrix}^2 \frac{\tau^2}{2!} \\ &\quad + \begin{bmatrix} t & -r \\ r & -t \end{bmatrix}^3 \frac{\tau^3}{3!} + \begin{bmatrix} t & -r \\ r & -t \end{bmatrix}^4 \frac{\tau^4}{4!} + \cdots \end{aligned} \tag{3.8}$$

In order to experience the numerical savings desired while retaining a reasonable degree of accuracy, the number of terms retained in the series is set so three significant digits are retained for the radiances calculated. Depending on the optical depth of the layer under construction, as many as six or as few as three terms are used. For example, for optical depths in the range $0.004 \leq \tau \leq 0.02$, four terms are retained in the series. The formulations for the T and R matrices that result for the four-term case are

$$\begin{aligned} T(H,0) &= 1 - t\tau + (t^2 - r^2)\frac{\tau^2}{2!} - (t^3 + 2(r^2t + tr^2) + rtr)\frac{\tau^3}{3!} \\ &\quad + (t(6t^3 + 5r^2t + 10tr^2 + 3rtr) \\ &\quad -r(2r^3 - t^2r + 2rt^2 - 3trt))\frac{\tau^4}{4!} \end{aligned} \tag{3.9}$$

$$\begin{aligned} R(H,0) &= r\tau - (rt + tr)\frac{\tau^2}{2!} + (t(2rt + tr) - r(4r^2 - t^2))\frac{\tau^3}{3!} \\ &\quad - (t(5r^3 + t^2r + 3rt^2 + 3trt) \\ &\quad + r(t^3 - 7r^2t + 3tr^2 + 15rtr))\frac{\tau^4}{4!} \end{aligned} \tag{3.10}$$

where t and r are the local transmission and reflection matrices and τ is the optical depth of the layer. Since the matrix A is composed of these t and r matrices (recall eqs. (2.4) - (2.6)) and (2.9)), numerical savings are realized here due to the fact that if A is $2n$ x $2n$ for example, then t and r are only n x n; thus, even though there are more matrix multiplications required to compute T and R in the TRM as opposed to the MEM, the size of the matices being multiplied actually causes the computation of T and R to be less numerically expensive.

Chapter 4

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

4.1 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 τ, single-scatter albedo ω_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 4.1 and 4.2 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 4.3 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 4.1: 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.

τ	ω_o	g	μ	$\mu_\odot$	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 4.2: 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.

τ	ω_o	g	μ	$\mu_\odot$	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 4.3: 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, $\mu_{\odot} = \cos 30^o$.

μ	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

4.2 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 4.1 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 τ increases (note that the abscissa on the plot is log τ) 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

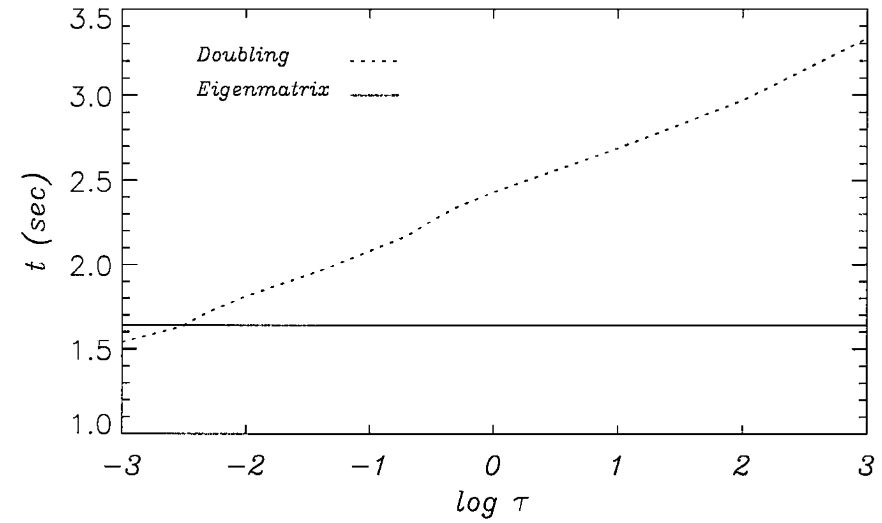


Figure 4.1: Results of a speed comparison between the eigenmatrix method in *Radiant* and the doubling method. The total times are the result of computing the radiances for a given atmospheric scene 500 times on a computer with a 400 MHz microprocessor.

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 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 DISORT when only the optical properties in one layer of atmosphere change and the radiances are recomputed. This situation is faced in practice when doing profile retrievals of atmospheric gases for example where Jacobians are needed to perform the retrieval and computing the elements of the Jacobian by finite difference is required.

Figure 4.2 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 "DISORT" is the time it took DISORT 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

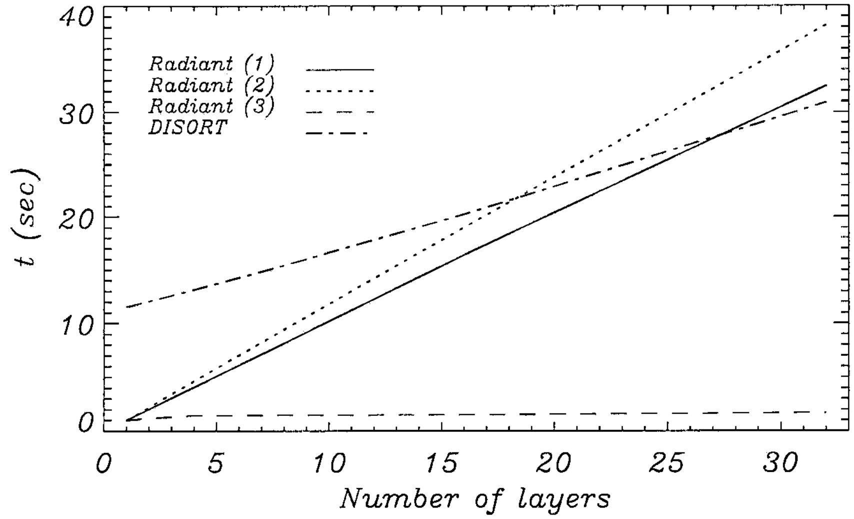


Figure 4.2: Results of a speed comparison between *Radiant* and DISORT. The total times are the result of computing the radiances for a given atmospheric scene 500 times on a computer with a 550 MHz microprocessor. See text for details between *Radiant* (1), (2), and (3).

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 eigenmatix formulation to obtain the solution to the radiative transfer equation, DISORT 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_2 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 5

Using Radiant

5.1 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 5.1. When the composite global transmission (T) and reflection (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 5.2),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 5.2, layer 3 has changed (shaded). It then proceeds to look for the changed layer. Once it finds the layer, the global transmission (T) and reflection

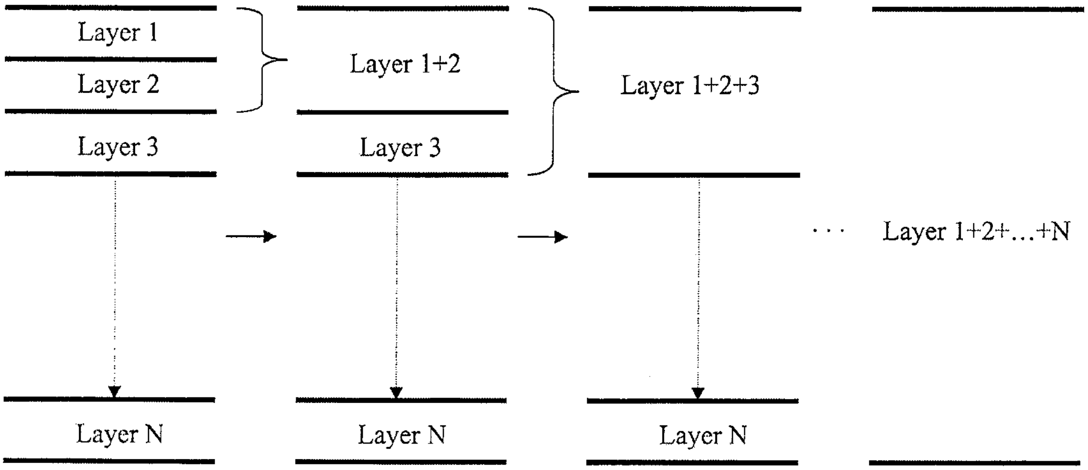


Figure 5.1: Illustration of *Radiant's* normal mode of operation.

(R) matrices and source vectors (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 5.2, these are the unshaded blocks. The boundary conditions are again applied and the RT solution for the new scene obtained.

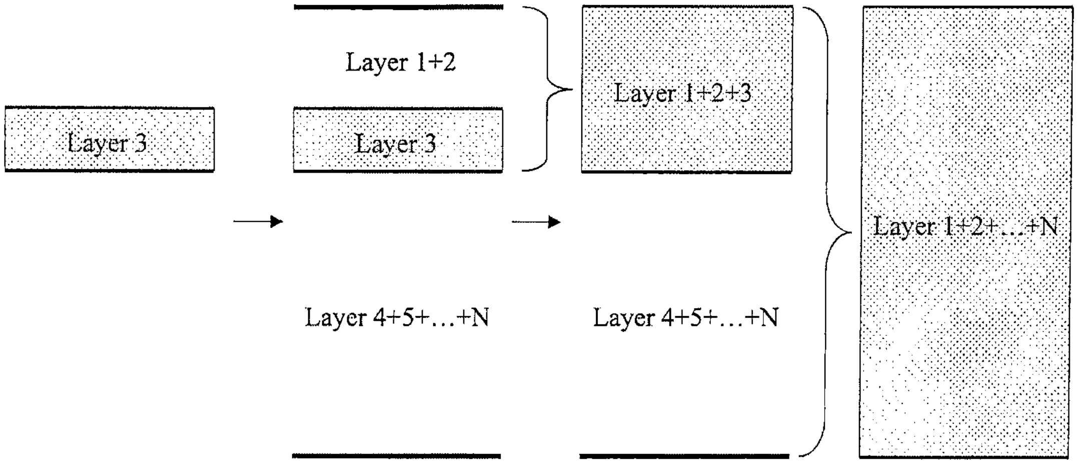
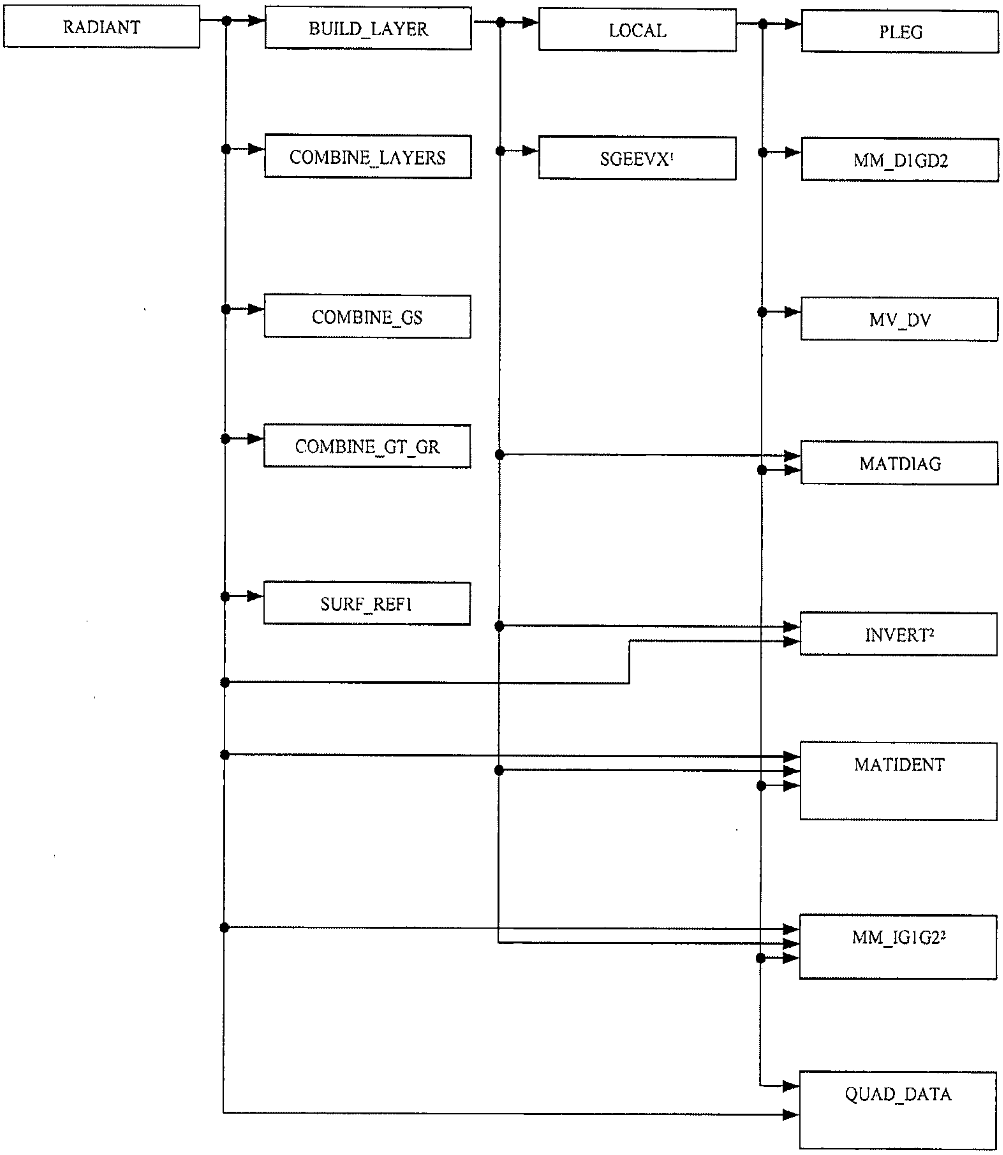


Figure 5.2: Illustration of *Radiant's* layer-saving mode of operation.

5.2 Program Structure & Subroutine Description

Radiant has the following program structure:



1. Uses LAPACK & BLAS Subroutines
2. Uses LINPACK & BLAS Subroutines

Figure 5.3: Illustration of *Radiant's* program structure. Parent routines call the subroutines to which they point and are directly connected (e.g. build_layer calls local, sgeevx, matdiag, invert, matident, and mm_ig1g2).

where the above subroutines 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 AND GLOBAL SOURCE VECTORS FROM TWO SEPARATE LAYERS TO CONSTRUCT THOSE OF A COMBINED LAYER.

INVERT - COMPUTES THE INVERSE OF A SQUARE MATRIX (USES LINPACK & BLAS SUBROUTINES).

LOCAL - CONSTRUCTS THE LOCAL REFLECTANCE AND TRANSMITTANCE MATRICES AND SOLAR 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 (D1xGxD2).

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”.

PLEG - COMPUTES RENORMALIZED ASSOCIATED LEGENDRE POLYNOMIALS.

QUAD_DATA - RETURNS THE POSITIVE QUADRATURE ROOTS AND ASSOCIATED WEIGHTS FOR THE INTERVAL [-1,1] FROM EITHER GAUSS OR LOBATTO QUADRATURE FOR THE FOLLOWING NUMBER OF STREAMS IN THE RADIANCE FIELD: 2, 4, 8, 16, 32 (USING 16 OR 32 STREAMS IS RECOMMENDED).

RADIANT - 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.

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

5.3 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( FSUN, MU0, PHI, QUAD, N, NUMDEG, DELTA_M, NUMLAY, &
    ALBEDO, SURF, G1_CLD, G2_CLD, G_CLD, G1_AERO, G2_AERO, G_AERO, &
    SIGS_CLD, SIGE_CLD, SIGS_AERO, SIGE_AERO, SIGS_RAY, SIGE_GAS, &
    DELTAZ, R_IO, FLUX_IO, L_IO, B_IO, INV_IO, WVN_FLAG, KFLAG, ITM, &
    IBMTOT, IBPTOT, ITPTOT )
```

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

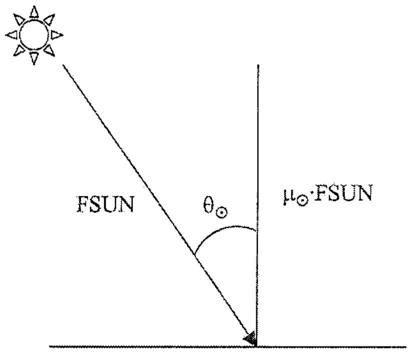
B_IO - (input) INTEGER
I/O FLAG FOR TOGGLING THE DISPLAY OF CERTAIN DATA FROM
SUBROUTINE "LOCAL" ON OR OFF.
0 = OFF
1 = ON

DELTA_M - (input) INTEGER
FLAG TO INDICATE WHETHER OR NOT DELTA_M SCALING
IS USED IN SUBROUTINE "LOCAL". DELTA_M SCALING
ALLOWS THE PHASE FUNCTION TO ASSUME A SHARPER PEAK
(SEE Wiscombe (1977))
0 = NO
1 = YES

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

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

FSUN - (input) DOUBLE PRECISION
SOLAR FLUX AT THE TOP OF THE ATMOSPHERE AT A GIVEN WAVELENGTH. 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 TOGGLING THE DISPLAY OF CERTAIN DATA FROM
SUBROUTINE "INVERT" (WHICH COMPUTES THE INVERSE OF A MATRIX).
ON OR OFF
0 = OFF
1 = ON

ITM - (input) DOUBLE PRECISION, DIMENSION(N)
VECTOR OF DOWNWELLING DIFFUSE 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 5.1: 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 TOGGLING THE DISPLAY OF CERTAIN DATA FROM SUBROUTINE "BUILD_LAYER" ON OR OFF.
0 = OFF
1 = ON

MU0 - (input) DOUBLE PRECISION

COSINE OF THE SOLAR ZENITH ANGLE.

N - (input) INTEGER

NUMBER OF UPWARD (OR DOWNWARD) STREAMS (CURRENTLY, CAN BE 2, 4, 8, OR 16).

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.

PHI - (input) DOUBLE PRECISION

THE AZIMUTHAL ANGLE FOR WHICH THE RADIANCE VECTOR IS COMPUTED (IN radians). NOTE: THE AZIMUTH ANGLE OF THE SUN IS CURRENTLY ASSUMED TO BE ZERO.

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
1 = LABATTO

NOTE: SEE SUBROUTINE QUAD_DATA FOR COSINES OF OBSERVING ANGLES CURRENTLY SUPPORTED WHEN RUNNING A PARTICULAR NUMBER OF STREAMS IN THE RADIANCE FIELD

R_IO - (input) INTEGER

I/O FLAG FOR TOGGLING THE DISPLAY OF CERTAIN DATA FROM SUBROUTINE "RADIANT" ON OR OFF.
0 = OFF
1 = ON

SIGE_AERO - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)

VECTOR OF EXTINCTION COEFFICIENTS FOR AEROSOL (IN km^{-1}).

SIGE_CLD - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)

VECTOR OF EXTINCTION COEFFICIENTS FOR CLOUD (IN km^{-1}).

SIGE_GAS - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)
VECTOR OF EXTINCTION COEFFICIENTS FOR ATMOSPHERIC GAS
(IN km^{-1}).

SIGS_AERO - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)
VECTOR OF SCATTERING COEFFICIENTS FOR AEROSOL
(IN km^{-1}).

SIGS_CLD - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)
VECTOR OF SCATTERING COEFFICIENTS FOR CLOUD
(IN km^{-1}).

SIGS_RAY - (input) DOUBLE PRECISION, DIMENSION(NUMLAY)
VECTOR OF SCATTERING COEFFICIENTS FOR RAYLEIGH SCATTER
(IN km^{-1}).

SURF - (input) INTEGER
FLAG INDICATING TYPE OF REFLECTING SURFACE.
0 = NO UNDERLYING SURFACE
1 = LAMBERTIAN SURFACE

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

5.4 Sample Simulations

To assist in understanding how the aforementioned input parameters are used in practice, this section contains some examples of how input parameters would be set for different desired simulations. The simple simulations are briefly described below with the actual parameters associated with the simulation included in the driver routine "radiant_driver.f90". The magnitudes of the parameters are for demonstration only and do not necessarily constitute the actual parameters one should use in a given "real" atmospheric scene (e.g. the asymmetry parameter settings in cloud cases 1 and 2). Atmospheric scenes containing aerosol would be treated in an analogous manner as the scenes containing cloud.

When using the driver for the first time, it is recommended that the user only change the two variables entitled SIM and INPUT_IO in the INPUT BLOCK section that begins at line 102 of the driver. SIM allows the user to select one of the sample simulations below while INPUT_IO allows one to display the input parameters used in that simulation.

The output of each simulation will be four sets of radiances. The first two sets are the total downwelling diffuse radiances impinging at the top of the set of atmospheric layers (denoted "top of the atmosphere" or TOA) and those exiting out the bottom of the set of atmospheric layers (denoted "bottom of the atmosphere" or BOA), respectively. The second two sets are the total upwelling diffuse radiances for the BOA and TOA. The column denoted MU are the cosines of the angles at which the radiances in the adjacent RADIANCE column are computed. Positive cosines are associated with upwelling angles and negative cosines with downwelling angles. Thus, a cosine of +1 is associated with a radiance directed straight up (used for a satellite viewing nadir for example) and a cosine of -1 with a radiance directed straight down (used for a ground instrument viewing zenith for example).

Simulation 1: Isolated Cloud Layer, Nadir and Zenith viewing

This simulates the radiances obtained when viewing a plane-parallel cloud 1 km thick with single Henyey-Greenstein scattering. The first element in the total upwelling diffuse radiance vector at the TOA is the radiance for nadir viewing with the other elements at non-nadir viewing angles as indicated by their angle cosines. The last element in the total downwelling diffuse radiance vector at the BOA is the radiance for zenith viewing the other elements at non-zenith viewing angles as indicated by their angle cosines.

Simulation 2: Isolated Cloud Layer, Non-nadir and Non-zenith viewing
(Benedetti Test Case)

This simulates the radiances obtained when viewing a plane-parallel cloud 1 km thick with single Henyey-Greenstein scattering. All elements in the total upwelling and downwelling diffuse radiance vectors are for the angles as indicated by their angle cosines.

One may compare the results from this simulation with the results in Table 4.3 for the case where m = 15. Note that in the table however only the magnitude of the angle cosine is used. Thus, $\mu = 0.9894$ is positive for the case of upwelling radiances $I^{+}(H)$ in the table and $\mu = 0.9894$ is negative for the case of the downwelling radiances $I^{-}(0)$ in the table.

Simulation 3: 1 Layer with Rayleigh Scattering, Lambertian surface

This simulates the radiances obtained when viewing a plane-parallel Rayleigh scattering layer 1 km thick with a Lambertian surface beneath. Unlike the other simulations, 32 streams (16 up, 16 down) are employed in the radiance field for an example. The surface albedo is 0.5.

Simulation 4: 32 Rayleigh-Scattering Layers with Optically Thin Cloud at 12 km,
Nadir viewing and Zenith viewing, Lambertian surface

This simulates the radiances obtained when viewing a block of 32 plane-parallel Rayleigh scattering layers with a cloud of optical depth $\tau = 0.1$ (using single Henyey-Greenstein scattering) at an altitude of 12 km and Lambertian surface beneath. As in simulation 1, the first element in the total upwelling diffuse radiance vector at the TOA is the radiance for nadir viewing and the last element in the total downwelling diffuse radiance vector at the BOA is the radiance for zenith viewing. The surface albedo is 0.2.

Simulation 5: 32 Layers with Gaseous Absorption, Nadir and Zenith viewing,
Lambertian surface (using layer-saving)

This simulates the radiances obtained when viewing a block of 32 plane-parallel layers with an absorbing gas active and Lambertian surface beneath. The surface albedo is 0.1.

Unlike the previous simulations, *Radiant* is called twice to compute the radiances. The first time, normal mode is used to compute the radiances for the scene. The second time, the extinction coefficient is perturbed downward in the layer corresponding to an altitude of 10 km and the radiances recomputed for the new scene using layer-saving. Note that, since the atmosphere is totally absorbing in this simulation, there are no diffuse downwelling radiances into the surface (all energy is in the direct solar beam which is not displayed).

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

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

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

where u_+ and u_- are matrices, when appropriately assembled, composing the matrix X from equation (3.1), Λ^+ is a diagonal matrix containing the positive eigenvalues of the matrix A, $\tau(H)$ is the optical depth of a given 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:

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

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

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 multisensor 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.