click here to jump to Table of Contents

DISORT, a General-Purpose Fortran Program for Discrete-Ordinate-Method Radiative Transfer in Scattering and Emitting Layered Media: Documentation of Methodology

(version 1.1, Mar 2000)

by

Knut Stamnes (kstamnes@stevens-tech.edu)
Dept. of Physics and Engineering Physics
Stevens Institute of Technology
Hoboken, NJ 07030

Si-Chee Tsay (tsay@gsfc.nasa.gov)
Warren Wiscombe (wiscombe@gsfc.nasa.gov)
Climate and Radiation Branch
NASA Goddard Space Flight Center
Greenbelt, MD 20771

Istvan Laszlo (laszlo@atmos.umd.edu)
Department of Meteorology
University of Maryland
College Park, MD 20742

Dedication

While it is perhaps unique for some authors of a document to express their gratitude to others, three of us (Stamnes, Tsay, Wiscombe) want to highlight the remarkable contribution of our co-author Istvan Laszlo to the DISORT effort. It is hard to imagine that we would have finished it even now, more than ten years after starting down this path, without his constant attention and selfless dedication. When we three wandered off to other pursuits, he would quietly pull us back. When we could find no time to work on DISORT, he would find time to work on it alone. His detailed comments and criticisms of this Report, and his painstaking detective work to uncover small bugs and inconsistencies still lurking in the code, made the difference between "good enough" and "nearly perfect". So, from us, and from the thousands of users of DISORT worldwide—our heartfelt thanks, Istvan!

Preface

This report documents a state-of-the-art discrete ordinate algorithm called DISORT for monochromatic unpolarized radiative transfer in non-isothermal, vertically inhomogeneous, but horizontally homogeneous media. The physical processes included are Planckian thermal emission, scattering with arbitrary phase function, absorption, and surface bidirectional reflection. The system may be driven by parallel or isotropic diffuse radiation incident at the top boundary, as well as by internal thermal sources and thermal emission from the boundaries. Radiances, fluxes, and mean intensities are returned at user-specified angles and levels.

DISORT has enjoyed considerable popularity in the atmospheric science and other communities since its introduction in 1988, but its theoretical background and algorithmic developments are scattered across many journals over many years, and a few parts have never been fully described. Furthermore, the algorithm and code have evolved since 1988. This report brings the DISORT description up to date and provides a self-contained account of its theoretical basis (including all significant equations used in the program) as well as a discussion of the numerical implementation of that theory.

Two major new DISORT features are also described: intensity correction algorithms designed to compensate for the delta-M forward-peak scaling and give accurate intensities even in low orders of approximation; and a more general surface bidirectional reflection option.

DISORT has been designed to be an exemplar of good scientific software as well as a program of intrinsic utility. An extraordinary effort has been made to make it numerically well-conditioned, error-resistant, and user-friendly, and to take advantage of robust existing software tools. A thorough test suite is provided to verify the program both against published results, and for consistency where there are no published results. This careful attention to software design has been just as important in DISORT's popularity as its powerful algorithmic content.

The DISORT Fortran-77 (and soon Fortran-90) code is available at:

ftp://climate.gsfc.nasa.gov/pub/wiscombe/Multiple_Scatt/

(underlined blue items like this are hyperlinks).

Table of Contents *(blue items are hyperlinks)*

1. INTRODUCTION

The purpose of this report is to document our numerical implementation of the discrete-ordinate method for radiative transfer in vertically inhomogeneous layered media, as embodied in a Fortran computer code called DISORT. A synopsis of DISORT's basic methodology was published by Stamnes et al. (1988a) in the same year that version 1.0 of DISORT was released (snailmailed on PC diskettes and even decks of computer cards in those days). The synopsis paper over-optimistically promised a fuller exposition as a NASA Reference Publication, but we never managed to wrap up that report even though it was mostly complete by 1994 or so. This document is the promised report.

This report is sufficiently tardy that some of its equations have already appeared in the textbook by Thomas and Stamnes (1999). That book intended to refer to this report, rather than vice versa; however, regardless of their chronological order, they have quite distinct goals. The book has pedagogical goals while this report aims to make DISORT less of a "black box" by documenting the equations and algorithms used in the DISORT computer code. These equations and algorithms were scattered across many separate papers, or unpublished entirely. This report also documents the current version of DISORT (2.0), which has two major new features, rather than the final version in the previous line of development (1.3) which was mainly the result of small upgrades and bug fixes to the code of 1988.

Part of the delay in releasing this report was the explosion of interest in matters radiative in the 1990's, sparkplugged by the Dept. of Energy's ARM Program, which entrained all of us in a mad whirl of activity. This intense activity was good news for the radiation field as a whole, but bad news for DISORT development and writing. Another problem was the evolution of DISORT methodology, which no static report could track. Our solution, at least for the time being, is to release this report as a web-based document which will undergo further changes. As such, we invite comments from you, our readers; our e-mail addresses are on the first page. The current version is available from the same web site as the DISORT code package:

`ftp://climate.gsfc.nasa.gov/pub/wiscombe/Multiple_Scatt/`

DISORT considers the transfer of monochromatic unpolarized radiation in a scattering, absorbing and emitting plane parallel medium, with a specified bidirectional reflectivity at the lower boundary. (See Schulz et al., 1999, for a polarized version.) The medium can be forced by a parallel beam and/or diffuse incidence and/or Planck emission at either boundary. Intensities at

user-selected angles and levels are the normal output. These levels need not be subsets of the computational levels necessary to resolve the medium, nor need the angles be subsets of the quadrature angles necessary to do the integrals over angle. For example, it may require 10 levels to resolve the medium and 16 quadrature angles to do the integrals over angle accurately, but the user can ask for intensities at just two arbitrary levels and one arbitrary angle. In addition to intensities, fluxes, flux divergences, and mean intensities are available as byproducts (or, optionally, the only products).

DISORT represents the culmination of years of effort by Stamnes and his collaborators to make it the finest radiative transfer algorithm available. The prime motivation for this effort was to provide a well-documented, well-tested, robust and versatile radiative transfer code that others can use as a "software tool" (in the Unix sense) for various applications. The code is painstakingly documented, including references to equation numbers in published papers and in this report.

Computational speed was never a goal, and no shortcuts were taken in which accuracy was sacrificed for speed. Nevertheless, by strict adherence to the best precepts of software development, including simple interfaces, no COMMON blocks, internal error-resistance, modularity, and passing low-level tasks to high-quality software packages like LINPACK (Dongarra et al., 1979) or LAPACK (Anderson et al., 1995), DISORT is reasonably efficient as well. DISORT follows most of the timeless coding precepts in the classic Kernighan and Plauger (1978). The best recent reference on code development for scientists is McConnell (1993), although it lacks the conciseness of Kernighan and Plauger. One important lesson you learn from these references is that with enough up-front effort, it is possible to produce almost zero-defect code. Because of extensive testing at both the subroutine and whole-code levels, DISORT has been remarkably bug-free since its introduction in 1988.

The large number of letters and e-mails we have received about DISORT, from almost every country and continent, indicate that we were correct in our assessment that a tool of this kind was needed. It plays this role in larger models like MODTRAN (Berk et al., 1998) and SBDART (Richiazzi et al, 1998), among others. It has also become a kind of standard against which to compare other modeling results. We had anticipated that DISORT would be applied to terrestrial (e.g., Tsay et al., 1989, 1990; Tsay and Stamnes, 1992) and Martian (Lindner, 1988) atmospheric radiative transfer problems. What we hadn't anticipated was the wide range of other applications DISORT would be put to:

- "fluences in non-homogeneous layered tissue",

- "bulk scattering properties of spectralon",
- "the effects of air pollution on visibility",
- "scattering in paint films pigmented with TiO_2",
- "microwave radiative transfer simulation in support of NPOESS program",
- "simulation of spectra recorded by the Soviet entry probes Venera 11-14",

to name but a few. It has also been applied to solve the Boltzmann transport equation both for auroral electron bombardment of the Earth's atmosphere (Lummerzheim et al., 1989) and for the classical 'linear gas' problem (Stamnes et al., 1991).

Since 1988, there have been a few major and a large number of minor improvements in DISORT methodology and code. The current version (2.0) includes these improvements, although older versions continue to be provided for checking and backward compatibility. (Eventually, however, a new Fortran-90 version will become the only actively evolving version.) These new developments are described in this report.

Strongly forward-peaked scattering is treated by the δ–M method (Wiscombe, 1977) in which the forward peak is *separated* and approximated by a Dirac delta-function. (Conventional usage is "truncating" the forward peak, but we reserve that word exclusively for truncating infinite series.) δ–M has proven to be both accurate and efficient for flux computation, but it introduces spurious oscillations around the true intensity curve plotted as a function of angle, as illustrated later in this report. Correction procedures developed by Nakajima and Tanaka (1988) resolve this difficulty and provide accurate intensity values with low numbers of quadrature angles ("streams") while retaining the considerable advantages of the δ–M method. This approach may yield factors of 10 to 1000 increases in computational efficiency.

Thermal (Planck) emission is included, both internally and as a boundary condition, making DISORT suitable for thermal infrared and microwave applications in addition to solar-spectrum applications. To account for temperature variations across any layer, we approximate the Planck function as linear in optical depth (Appendix A of Wiscombe, 1976, analyzes the error in this approximation). Kylling and Stamnes (1992) developed a new exponential-linear-in-optical-depth approximation to the Planck function which may allow two to five times larger temperature changes across layers without loss of accuracy, but it has not been thoroughly tested against the linear approximation, it requires four special branches to avoid numerical problems, and it

assumes the temperature at the center of a layer is the average of the boundary temperatures (which may or may not be a good approximation). While we had originally intended to include this approximation in DISORT v2.0, we decided at the end that it needed more study.

The DISORT code package includes a set of test problems designed to:

(a) act as templates for how to call DISORT properly,

(b) demonstrate the basic capabilities of the code,

(c) exercise different logical pathways through the code, and

(d) provide comparisons with numerical results obtained by other investigators.

Both the code and the test problems contain further documentation which is carefully written and useful, but which we have chosen not to include in this report.

Section 2 gives a self-contained account of the theory, including the relevant equations used in the computer program. Section 3 discusses most aspects of DISORT's numerical implementation — most importantly, how to compute eigenvalues and eigenvectors reliably and efficiently, and how a simple scaling transformation avoids fatal overflows and ill-conditioning in the matrix inversion needed to determine the constants of integration.

Appendix A contains most of the details of the Nakajima/Tanaka intensity correction method. Appendix B is a "wish list" of things that we think should be added to DISORT, but that we haven't yet found the time to implement. Users are encouraged to send us ideas to add to this wish list. Appendix C contains some thoughts on the error from approximating finite sources as point sources.

2. THEORY

2.1 Basic Equations and Definitions

Here, we merely wish to present the basic radiative transfer formulas solved by DISORT in order to establish notation and conventions, with a minimum of definition and explanation. Our purpose is not to act as a textbook on radiative transfer; for that we refer the reader mainly to Thomas and Stamnes (1999), since it complements this report well and uses similar notation, or to Liou (1980). Thomas and Stamnes will be cited using the shorthand notation TS/2.28 where "2.28" is the equation number. Chandrasekhar's book (1960) will also be cited occasionally using, e.g., C/I.63 where "I" is the chapter and "63" is the equation number.

The equation describing the transfer of monochromatic radiation at wavelength λ through a plane parallel medium is given by (C/I.63; TS/2.28)

$$\mu \frac{\mathrm{d}I_\lambda^{tot}(\tau_\lambda,\mu,\phi)}{\mathrm{d}\tau_\lambda} = I_\lambda^{tot}(\tau_\lambda,\mu,\phi) - S_\lambda(\tau_\lambda,\mu,\phi) \qquad (1)$$

where $I_\lambda{}^{tot}$ is the specific intensity at vertical optical depth τ_λ (measured downward from the upper boundary) in a cone of unit solid angle along direction μ,ϕ (ϕ is the azimuthal angle and μ is the cosine of the polar angle). The polar angle is defined such that μ is positive in the upward hemisphere.

The wavelength dependence of all quantities is omitted in the rest of this report. DISORT makes no explicit use of wavelength except in the calculation of the Planck function, which requires a wavelength interval. Thus the medium and boundary properties, and the resulting radiant quantities calculated therefrom, can be regarded either as values at a particular wavelength, or as values *integrated over a range of wavelengths* over which the medium properties and boundary reflectance are invariant.

The "source function" S in (1) is the actual internal source Q plus the scattering into the direction μ,ϕ from all other directions (TS/5.63):

$$S(\tau,\mu,\phi) = Q(\tau,\mu,\phi) + \frac{\omega(\tau)}{4\pi}\int_0^{2\pi}\mathrm{d}\phi' \int_{-1}^{1}\mathrm{d}\mu'\, P(\tau,\mu,\phi;\mu',\phi')\, I^{tot}(\tau,\mu',\phi') \qquad (2a)$$

where ω is the "single scattering albedo" and P is the "scattering phase function" (confusing

terms imported from astronomy but now hallowed by tradition). *P* describes the angular scattering pattern of an infinitesimal volume (optically thin but large enough to contain many scatterers), and ω is the fraction of an incident beam which is scattered by that volume (the remainder being absorbed). No provision is made for preferentially oriented scatterers; thus, ω has no explicit dependence on the direction of the incident beam, and *P* depends only on the angle between the incident and scattered beam, not on the incident and scattered directions separately:

$$P(\tau, \mu, \phi; \mu', \phi') = P(\tau, \cos\Theta) \tag{2b}$$

where, from the cosine law of spherical trigonometry,

$$\cos\Theta = \mu\,\mu' + \sqrt{(1-\mu^2)(1-\mu'^2)}\ \cos(\phi - \phi') \tag{2c}$$

If the only internal source of radiation is thermal emission in local thermodynamic equilibrium (LTE), the source term *Q* reduces to

$$Q(\tau, \mu, \phi) = Q^{(\text{thermal})}(\tau) = \{1 - \omega(\tau)\}\ B[\lambda, T(\tau)] \tag{3a}$$

where $B(\lambda,T)$ is the Planck function at wavelength λ and temperature *T*.

If the medium is also forced by a 'direct' or parallel beam at the top boundary of intensity I_0 in direction μ_0,ϕ_0, then the intensity of this beam at any optical depth is trivially given by (TS/6.1)

$$I^{direct} = I_0\, e^{-\tau/\mu_0}\ \delta(\mu - \mu_0)\,\delta(\phi - \phi_0) \tag{3b}$$

Some thoughts and calculations about what happens when the beam is not quite parallel, that is, when the beam does not emanate from a point source, are given in Appendix C (this subject is omitted from all radiative transfer texts). Now if the usual diffuse-direct transformation is made (TS/6.2)

$$I^{tot} = I^{direct} + I^{diffuse} \tag{3c}$$

then (1,2) hold for the diffuse intensity only, provided that a new 'pseudo-beam' source term is added (TS/6.7,6.9):

$$Q^{(\text{beam})}(\tau, \mu, \phi) = \frac{\omega(\tau)\, I_0}{4\pi}\ P(\tau, \mu, \phi; -\mu_0, \phi_0)\ e^{-\tau/\mu_0} \tag{3d}$$

In general,

$$Q(\tau,\mu,\phi) = Q^{(\text{thermal})}(\tau) + Q^{(\text{beam})}(\tau,\mu,\phi) \tag{3e}$$

Many radiative transfer models ignore either the thermal or the beam source term; DISORT allows for both to be present, but takes computational shortcuts when one or the other is absent.

From here on, we shall for simplicity omit the 'diffuse' superscript

$$I \equiv I^{diffuse} \tag{4}$$

which of course become a tautology if there is no beam source.

2.1.1 Separation of ϕ–dependence

As noted above, DISORT assumes that the scattering phase function depends only on the angle Θ between the incident and scattered beams. This fact can be utilized to factor out the ϕ–dependence in (1) and (2) as follows. Expand the phase function in a series of 2*M* Legendre polynomials P_l (TS/6.28)

$$P(\tau,\cos\Theta) = \sum_{\ell=0}^{2M-1} (2\ell+1)\, g_\ell(\tau)\, P_\ell(\cos\Theta) \tag{5a}$$

where, by virtue of the orthogonality of Legendre polynomials, the expansion coefficients are given by

$$g_\ell(\tau) = \tfrac{1}{2}\int_{-1}^{+1} P_\ell(\cos\Theta)\, P(\tau,\cos\Theta)\, \mathrm{d}(\cos\Theta) \tag{5b}$$

Note that g_0=1 because the phase function, being a probability distribution, is normalized to unity. g_1 is called the "asymmetry factor", and ranges from 0.7 to 0.9 for aerosols and clouds in the solar spectrum. The g's generally decrease monotonically. For cloud phase functions they often decrease very slowly, and several hundred terms may be necessary in (5a) to adequately represent the phase function. This fact causes downstream problems in radiative transfer and basically doomed Chandrasekhar's analytic approach, which failed when any but the first three terms in (5a) were non-zero, that is, for any but the isotropic and Rayleigh scattering problems.

The next step is to apply the Addition Theorem for Spherical Harmonics (C/VI.86; TS/6.30) to (5a) to obtain

$$P(\tau,\cos\Theta)=\sum_{\ell=0}^{2M-1}(2\ell+1)\,g_\ell(\tau)\Big\{P_\ell(\mu)P_\ell(\mu')+2\sum_{m=1}^{\ell}\Lambda_\ell^m(\mu)\,\Lambda_\ell^m(\mu')\cos m(\phi-\phi')\Big\} \tag{5c}$$

Here, Λ_l^m is a *normalized* associated Legendre polynomial defined by (TS/6.31)

$$\Lambda_\ell^m(\mu)=\sqrt{\frac{(\ell-m)!}{(\ell+m)!}}\,P_\ell^m(\mu) \tag{5d}$$

where P_l^m is the usual associated Legendre polynomial. DISORT uses the Λ_l^m in preference to the P_l^m for numerical reasons given in Section 3.3.2.

The final step in factoring out the ϕ-dependence is to expand the intensity in a Fourier cosine series (C/VI.91; TS/6.34),

$$I(\tau,\mu,\phi)=\sum_{m=0}^{2M-1}I^m(\tau,\mu)\,\cos m(\phi_0-\phi) \tag{6}$$

Substitution of this equation, (2), and (5c) into the radiative transfer equation (1) causes (1) to split into $2M$ independent integro-differential equations, one for each azimuthal intensity component (TS/6.35):

$$\mu\frac{\mathrm{d}\,I^m(\tau,\mu)}{\mathrm{d}\tau}=I^m(\tau,\mu)-S^m(\tau,\mu)\quad(m=0,1,\ldots,2M-1) \tag{7}$$

where the source function is given by

$$S^m(\tau,\mu)=\int_{-1}^{1}D^m(\tau,\mu,\mu')\,I^m(\tau,\mu')\,\mathrm{d}\mu'+Q^m(\tau,\mu) \tag{8a}$$

and the other symbols are defined by (TS/6.33,35,36)

$$D^m(\tau,\mu,\mu')=\frac{\omega(\tau)}{2}\sum_{\ell=m}^{2M-1}(2\ell+1)\,g_\ell(\tau)\,\Lambda_\ell^m(\mu)\,\Lambda_\ell^m(\mu') \tag{8b}$$

$$Q^m(\tau,\mu) = X_0^m(\tau,\mu)\, e^{-\tau/\mu_0} + \delta_{m0}\, Q^{(\text{thermal})}(\tau) \tag{8c}$$

$$X_0^m(\tau,\mu) = \frac{\omega(\tau)\, I_0}{4\pi}\,(2-\delta_{m0}) \sum_{\ell=m}^{2M-1} (-1)^{\ell+m}\,(2\ell+1)\, g_\ell(\tau)\, \Lambda_\ell^m(\mu)\, \Lambda_\ell^m(\mu_0) \tag{8d}$$

$$\delta_{m0} = \begin{cases} 1 & \text{if } m = 0 \\ 0 & \text{otherwise} \end{cases}$$

This procedure factors out the azimuth (ϕ) dependence. It also uncouples the various Fourier components I^m in (7); that is, I^m does not depend on any I^{m+k} for $k \neq 0$. The only place ϕ is really used in DISORT is in reconstructing the intensity from (6) near the end of the DISORT subroutine.

Note that (8b) is inconsistent with TS (6.33). (8b) is correct and TS is in error in this regard. TS (6.33) should have the factor $(2-\delta_{m0})$ removed and placed under the summation in TS (6.32). With that change, TS (6.32–6.36) are all correct.

Having to compute the azimuthal series—that is, having to solve (7) repeatedly for many *m*—is responsible for much of the expense notoriously associated with 1–D radiative transfer calculations. The intensity expansion in (6) is the same length as the phase function expansion (5a), which can be hundreds or even thousands of terms long. In practice, however, the intensity expansion (6) is much shorter. What practically limits the length of both expansions is the later choice of the number of quadrature angles for approximating the angular integrals. Following Chandrasekhar (Ch. VI, Eq. 93) it has been customary not to extend these expansions beyond the number of quadrature angles. In his words, we are not "entitled" to do so. For example, for a four-stream approximation (four quadrature angles) we take four terms in (5a) and (6).

While it may seem strange that a purely numerical decision like picking the number of quadrature angles should react back on the length of a fundamental expansion, this has traditionally been the nature of the discrete ordinate approximation. There has been no systematic study (that we are aware of at least) to examine the potential merit of using more terms in the Legendre polynomial expansion of the phase function than the number of streams. Section 3.7 discusses all this somewhat further.

The code also carefully tests for all situations where the sum in (6) collapses to the *m*=0 term:

- no beam sources
- the beam source is at the zenith (μ_0=1)
- only fluxes and/or mean intensities are needed
- only zenith and/or nadir intensities are needed

We can understand these special cases in terms of their complete azimuthal symmetry about the vertical direction. LTE thermal emission is isotropic and scattering volumes behave isotropically (Eq. 2b), so they cannot create any azimuth dependence. In fact, azimuth dependence can only be created by the boundary conditions in DISORT, which basically means by a non-overhead beam source since at the lower boundary azimuth-dependent reflection, as for example from a plowed field, is not permitted (cf. Section 2.9).

2.1.2 Flux, flux divergence, and mean intensity

Radiation *fluxes* measure the total energy crossing a horizontal area per unit time, and are of importance for example in climatic energy budget studies. *Mean intensities* measure the total intensity hitting a volume from all sides, and are of importance for example to photochemical reactions. The only difference between them is that fluxes are *cosine-weighted* averages of the intensity while mean intensities are unweighted averages. Up (F^+) and down (F^-) fluxes are defined as follows (TS/6.25):

$$F^+(\tau) \equiv \int_0^{2\pi} \mathrm{d}\phi \int_0^1 \mu\, I(\tau, +\mu, \phi)\, \mathrm{d}\mu = 2\pi \int_0^1 \mu\, I^0(\tau, +\mu)\, \mathrm{d}\mu$$

$$F^-(\tau) \equiv \mu_0 I_0 e^{-\tau/\mu_0} + \int_0^{2\pi} \mathrm{d}\phi \int_0^1 \mu\, I(\tau, -\mu, \phi)\, \mathrm{d}\mu \qquad (9a)$$

$$= \mu_0 I_0 e^{-\tau/\mu_0} + 2\pi \int_0^1 \mu\, I^0(\tau, -\mu)\, \mathrm{d}\mu$$

where the direct-beam contribution has been included in the downward flux and where (6) has been used to give the second equality. Clearly azimuthal integration of (6) drops all but its m=0 term, which is hence known as the *azimuthally-averaged intensity* when expressed as:

$$I^0(\tau, \mu) \equiv \frac{1}{2\pi} \int_0^{2\pi} I(\tau, \mu, \phi)\, \mathrm{d}\phi \qquad (9b)$$

The hemispherical mean intensities are defined correspondingly as

$$\bar{I}^{+}(\tau) \equiv \frac{1}{2\pi}\int_0^{2\pi} \mathrm{d}\phi \int_0^1 I(\tau,+\mu,\phi)\,\mathrm{d}\mu = \int_0^1 I^0(\tau,+\mu)\,\mathrm{d}\mu$$

$$\bar{I}^{-}(\tau) \equiv \frac{1}{2\pi}\left[I_0 e^{-\tau/\mu_0} + \int_0^{2\pi} \mathrm{d}\phi \int_0^1 I(\tau,-\mu,\phi)\,\mathrm{d}\mu \right] \tag{9c}$$

$$= \frac{1}{2\pi} I_0 e^{-\tau/\mu_0} + \int_0^1 I^0(\tau,-\mu)\,\mathrm{d}\mu$$

The net flux and mean intensity are then (TS/6.26)

$$\begin{aligned} F(\tau) &\equiv F^{-}(\tau) - F^{+}(\tau) \\ \bar{I}(\tau) &\equiv \frac{1}{2}\left[\bar{I}^{-}(\tau) + \bar{I}^{+}(\tau)\right] \end{aligned} \tag{9d}$$

Finally, we have the flux divergence (TS/6.27):

$$\frac{dF}{d\tau} = 4\pi(1-\omega)\left\{ \bar{I} - B[\lambda, T(\tau)] \right\} \tag{9e}$$

This is related to the *radiative heating rate* (TS/5.76).

In general, quite different flux divergences are obtained from using a finite difference approximation $\Delta F/\Delta\tau$ to $\mathrm{d}F/\mathrm{d}\tau$, where "$\Delta$" indicates a difference between two levels. $\Delta F/\Delta\tau$ conserves energy much better and hence is preferred in climate modeling applications, while $\mathrm{d}F/\mathrm{d}\tau$ gives a better idea of the vertical variation of heating rate. This is simply because, trivially,

$$\int_{\tau_j}^{\tau_{j+1}} \left(\frac{dF}{d\tau}\right) d\tau \equiv F_{j+1} - F_j$$

but if this integral is approximated using only $\mathrm{d}F/\mathrm{d}\tau$ values at computational levels

$$\int_{\tau_j}^{\tau_{j+1}} \left(\frac{dF}{d\tau}\right) d\tau \cong \tfrac{1}{2}\left[\left(\frac{dF}{d\tau}\right)_{\tau_j} + \left(\frac{dF}{d\tau}\right)_{\tau_{j+1}} \right] \left(\tau_{j+1} - \tau_j\right) \neq F_{j+1} - F_j$$

(since only two integrands values are available, Trapezoidal Quadrature must be used) it then does not recapture the flux difference across the slab. (This of course ignores truncation error in F and $\mathrm{d}F/\mathrm{d}\tau$.due to taking too few streams.)

While DISORT quadrature rules are not discussed till Section 3.3.1, it is convenient to finish

up the subject of fluxes and mean intensities here and not return to it. DISORT uses Gaussian quadrature for the interval [0,1] with abscissae μ_i and weights w_i, i=1 to N. Thus, the DISORT approximations for flux and mean intensity are:

$$F^+(\tau) = 2\pi \int_0^1 \mu\, I^0(\tau, +\mu)\, \mathrm{d}\mu \cong 2\pi \sum_{i=1}^{N} w_i\, \mu_i\, I^0(\tau, +\mu_i)$$

$$F^-(\tau) = \mu_0 I_0 e^{-\tau/\mu_0} + 2\pi \int_0^1 \mu\, I^0(\tau, -\mu)\, \mathrm{d}\mu \tag{9f}$$

$$\cong \mu_0 I_0 e^{-\tau/\mu_0} + 2\pi \sum_{i=1}^{N} w_i\, \mu_i\, I^0(\tau, -\mu_i)$$

$$\bar{I}(\tau) = \frac{1}{4\pi} I_0 e^{-\tau/\mu_0} + \frac{1}{2}\left[\int_0^1 I^0(\tau, +\mu)\, \mathrm{d}\mu + \int_0^1 I^0(\tau, -\mu)\, \mathrm{d}\mu \right]$$

$$\cong \frac{1}{4\pi} I_0 e^{-\tau/\mu_0} + \frac{1}{2} \sum_{i=1}^{N} w_i \left[I^0(\tau, +\mu_i) + I^0(\tau, -\mu_i) \right] \tag{9g}$$

Version 1.x of DISORT returned azimuthally-averaged intensities to the user, but version 2.0 no longer does so (they are still calculated internally, however). No one apparently has a real application for this quantity, including ourselves (except in one former test problem, to compare to a published table). The only known cases where it has been measured involved flying in a circle while continuously targeting a surface pixel. In these cases, it was the individual intensities and not their azimuthal average which were of interest. Finally, because the Nakajima/Tanaka method (Sec. 3.6) is used to correct intensities, but not individual components in the azimuthal sum (6) leading to those intensities, the azimuthally-averaged intensities are inconsistent with the intensities returned to the user, even though they may not be inaccurate.

2.2 Discrete Ordinate Approximation – Matrix Formulation

2.2.1. General

The discrete ordinate approximation to (7) is obtained by approximating the integral in (8a) by a quadrature sum and thus transforming the integro-differential equation (7) into the following system of ordinary differential equations (cf. Stamnes and Dale, 1981; Stamnes and Swanson,

1981)

$$\mu_i \frac{\mathrm{d}\, I^m(\tau,\mu_i)}{\mathrm{d}\tau} = I^m(\tau,\mu_i) - S^m(\tau,\mu_i) \quad (i = \pm 1,\ldots,\pm N) \tag{10a}$$

Each μ_i is called a "stream", and we call this a "$2N$–stream approximation". It is convenient to draw the μ_i from a Gaussian quadrature rule for [0,1] and to have them be mirror symmetric ($\mu_{-i}=-\mu_i$), but this is not important for the time being. If we write (8a) in quadratured form, S^m becomes a linear combination of I^m values at all quadrature angles μ_i,

$$S^m(\tau,\mu_i) = \sum_{\substack{j=-N \\ j\neq 0}}^{N} w_j\, D^m(\tau,\mu_i,\mu_j)\, I^m(\tau,\mu_j) + Q^m(\tau,\mu_i) \tag{10b}$$

which makes the system (10a) coupled in i (but not in m).

In a vertically inhomogeneous medium, the coefficients D^m in (10b) are functions of τ. This makes (10a) a system of $2N$ coupled differential equations with non-constant coefficients, for which there is no closed-form analytic solution in spite of the linearity. To obtain analytic solutions, DISORT assumes the medium consists of L adjacent homogeneous layers. The single-scattering albedo and phase function are assumed constant within each layer (but allowed to vary from layer to layer, cf. Figure 1). Thus a presumably continuous variation is approximated by a step-function variation. The thermal source term is approximated by a polynomial variation in τ within each layer.

To illustrate how things work without a welter of subscripts, we shall consider a single homogeneous layer $\tau_{p-1}\leq\tau\leq\tau_p$ but omit the p subscripts, which are implicit on all quantities, until later. The τ arguments of D and S will also be omitted since D and S are, by assumption, independent of τ in any one layer.

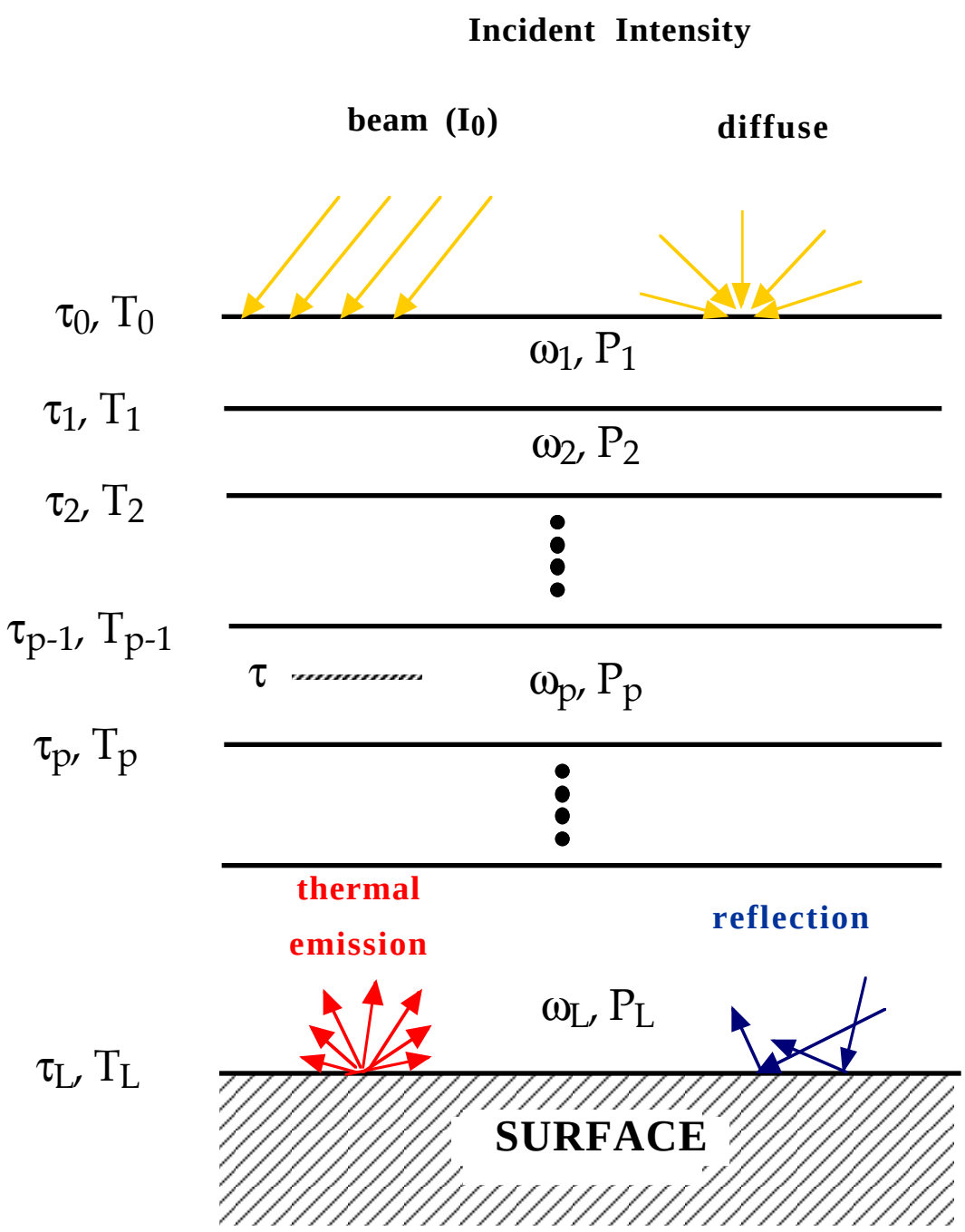

Figure 1. Schematic illustration of a multilayered optical medium driven by incident beam and diffuse intensity at the top boundary, and driven by reflection and thermal emission at the bottom boundary. Note that cumulative optical depth τ, temperature T, intensity, and flux are defined at layer interfaces while single-scatter albedo ω and phase function P are defined as layer averages.

We shall first describe the two- and four-stream cases (N=1 and 2) in order to see the DISORT equations in their simplest forms. It will then be easier to understand the multi-stream case. Since the two- and four-stream approximations cannot normally provide accurate intensities, we shall focus on the computation of fluxes and mean intensities. Hence we need only consider the m=0 component of the intensity.

Numerous papers have appeared since the early 1900's on two-stream (and closely related Eddington) approximations for radiative transfer. We shall not review this topic here, but rather refer the reader to the systematic presentation and comprehensive discussion of the two-stream method in a recent textbook (Thomas and Stamnes, 1999). Among the welter of published papers, we also recommend the following as having particular merit: Meador and Weaver, 1980; Zdunkowski et al., 1980; and King and Harshvardhan, 1986.

Note that the two-stream special case is not actually permitted in DISORT; it would give inferior results to TWOSTR, a further development by Kylling et al. (1995) based on DISORT. TWOSTR is a stand-alone program which is simpler than DISORT yet has more features specifically keyed to the two-stream application. Anyway, two-stream and even four-stream users prefer a custom-crafted program rather than a general-purpose program like DISORT, because speed is an important consideration. We provide TWOSTR mainly as a kind of benchmark and because, like DISORT but unlike many other published two- and four-stream algorithms, it eliminates the numerical ill-conditioning that occurs when two or more layers are combined. In the popular delta-Eddington code (Wiscombe, 1977) this problem was sidestepped by subdividing layers until each sub-layer was optically thin. No such artifices are necessary in TWOSTR, because the ill-conditioning problem is eliminated at its source.

2.2.2 Two-stream approximation (N=1)

The two-stream approximation is obtained by setting N=1 in (10a,b), which yields two coupled differential equations ($\mu_{-1}=-\mu_1$, $w_{-1}=w_1=1$) valid for any layer in Figure 1:

$$\mu_1 \frac{\mathrm{d}\, I^+(\tau)}{\mathrm{d}\tau} = I^+(\tau) - D(\mu_1,-\mu_1)\, I^-(\tau) - D(\mu_1,\mu_1)\, I^+(\tau) - Q^+(\tau) \qquad (11a)$$

$$-\mu_1 \frac{\mathrm{d}\, I^-(\tau)}{\mathrm{d}\tau} = I^-(\tau) - D(-\mu_1,-\mu_1)\, I^-(\tau) - D(-\mu_1,\mu_1)\, I^+(\tau) - Q^-(\tau) \qquad (11b)$$

where we have dropped the m=0 superscript. In (11a) and (11b) we have used the following

definitions

$$I^{\pm}(\tau) \equiv I(\tau, \pm\mu_1)$$

$$Q^{\pm}(\tau) \equiv Q(\tau, \pm\mu_1)$$

$$D(\mu_1, -\mu_1) = D(-\mu_1, \mu_1) = \frac{\omega}{2}(1 - 3g_1\mu_1^2) \equiv \omega\,\eta$$

$$D(\mu_1, \mu_1) = D(-\mu_1, -\mu_1) = \frac{\omega}{2}(1 + 3g_1\mu_1^2) \equiv \omega(1 - \eta)$$

where

$$\eta \equiv \frac{1}{2}\left(1 - 3g_1\,\mu_1^2\right) \tag{11c}$$

is called the backscatter ratio and g_1 is the first Legendre moment of the phase function as defined in (5c).

We may rewrite (11a) and (11b) in matrix form as

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\begin{bmatrix} I^+ \\ I^- \end{bmatrix} = \begin{bmatrix} -\alpha & -\beta \\ \beta & \alpha \end{bmatrix}\begin{bmatrix} I^+ \\ I^- \end{bmatrix} - \begin{bmatrix} Q'^+ \\ Q'^- \end{bmatrix} \tag{12}$$

where

$$Q'^{\pm} = \pm Q^{\pm} \big/ \mu_1$$

$$\alpha = \left[D(\mu_1, \mu_1) - 1\right] \big/ \mu_1 = \left[\omega(1-\eta) - 1\right] \big/ \mu_1$$

$$\beta = D(\mu_1, -\mu_1) \,/\mu_1 = \omega\,\eta \,/\mu_1$$

2.2.3 Four-stream approximation (N=2)

In this case we obtain 4 coupled differential equations from (10a,b) (again for any layer in Figure 1 and assuming a quadrature satisfying $\mu_{-i}=-\mu_i$, $\mu_i>0$, and $w_{-i}=w_i$). Already these equations are long and not particularly enlightening in scalar form, so we write them immediately in matrix form as follows:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\begin{bmatrix} I(\tau,\mu_1) \\ I(\tau,\mu_2) \\ I(\tau,-\mu_1) \\ I(\tau,-\mu_2) \end{bmatrix} = \begin{bmatrix} -\alpha_{11} & -\alpha_{12} & -\beta_{11} & -\beta_{12} \\ -\alpha_{21} & -\alpha_{22} & -\beta_{21} & -\beta_{22} \\ \beta_{11} & \beta_{12} & \alpha_{11} & \alpha_{12} \\ \beta_{21} & \beta_{22} & \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} I(\tau,\mu_1) \\ I(\tau,\mu_2) \\ I(\tau,-\mu_1) \\ I(\tau,-\mu_2) \end{bmatrix} - \begin{bmatrix} Q'(\tau,\mu_1) \\ Q'(\tau,\mu_2) \\ Q'(\tau,-\mu_1) \\ Q'(\tau,-\mu_2) \end{bmatrix} \tag{13}$$

where

$$Q'(\tau,\pm\mu_i) = \pm Q(\tau,\pm\mu_i)/\mu_i \quad (i=1,2) \tag{14a}$$

$$\begin{aligned} \alpha_{11} &= \left[w_1 D(\mu_1,\mu_1)-1\right]/\mu_1 = \left[w_1 D(-\mu_1,-\mu_1)-1\right]/\mu_1 \\ \alpha_{22} &= \left[w_2 D(\mu_2,\mu_2)-1\right]/\mu_2 = \left[w_2 D(-\mu_2,-\mu_2)-1\right]/\mu_2 \\ \alpha_{12} &= w_2 D(\mu_1,\mu_2)/\mu_1 = w_2 D(-\mu_1,-\mu_2)/\mu_1 \\ \alpha_{21} &= w_1 D(\mu_2,\mu_1)/\mu_2 = w_1 D(-\mu_2,-\mu_1)/\mu_2 \end{aligned} \tag{14b}$$

$$\begin{aligned} \beta_{11} &= w_1 D(\mu_1,-\mu_1)/\mu_1 = w_1 D(-\mu_1,\mu_1)/\mu_1 \\ \beta_{22} &= w_2 D(\mu_2,-\mu_2)/\mu_2 = w_2 D(-\mu_2,\mu_2)/\mu_2 \\ \beta_{12} &= w_2 D(\mu_1,-\mu_2)/\mu_1 = w_2 D(-\mu_1,\mu_2)/\mu_1 \\ \beta_{21} &= w_1 D(\mu_2,-\mu_1)/\mu_2 = w_1 D(-\mu_2,\mu_1)/\mu_2 \end{aligned} \tag{14c}$$

If we introduce the vectors

$$\mathbf{I}^{\pm} = \left\{I(\tau,\pm\mu_i)\right\}, \quad \mathbf{Q}'^{\pm} = \left\{Q'(\tau,\pm\mu_i)\right\} \qquad (i=1,2) \tag{14d}$$

then (13) can be written in a more compact form as

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\begin{bmatrix} \mathbf{I}^+ \\ \mathbf{I}^- \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\alpha} & -\boldsymbol{\beta} \\ \boldsymbol{\beta} & \boldsymbol{\alpha} \end{bmatrix} \begin{bmatrix} \mathbf{I}^+ \\ \mathbf{I}^- \end{bmatrix} - \begin{bmatrix} \mathbf{Q}'^+ \\ \mathbf{Q}'^- \end{bmatrix} \tag{14e}$$

where the matrixes $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are defined above. Note that this equation is very similar to the one obtained in the two-stream approximation (12) except that the scalars α and β have become 2x2 matrixes. Note also that matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ may be interpreted as layer transmission and reflection operators, respectively (cf. e.g. Stamnes, 1986).

2.2.4 Multi-stream approximation (N arbitrary)

From the preceding sketches of the two- and four-stream cases, it is fairly easy to guess the generalization to $2N$ streams: namely that, for any layer in Figure 1, (10a) may be written in matrix form identically to (14e) except that the matrix elements are now defined in a more general way:

$$\begin{aligned}
&\mathbf{I}^{\pm} = \left\{ I^m(\tau, \pm\mu_i) \right\} \quad (i = 1, \ldots, N) \\
&\mathbf{Q}'^{\pm} = \mathbf{M}^{-1}\, \mathbf{Q}^{\pm} \\
&\mathbf{Q}^{\pm} = \left\{ Q^m(\tau, \pm\mu_i) \right\} \quad (i = 1, \ldots, N) \\
&\mathbf{M} = \left\{ \mu_i \delta_{ij} \right\} \quad (i, j = 1, \ldots, N)
\end{aligned} \tag{15a}$$

$$\begin{aligned}
&\boldsymbol{\alpha} = \mathbf{M}^{-1} \left\{ \mathbf{D}^{+}\, \mathbf{W} - \Im \right\} \\
&\boldsymbol{\beta} = \mathbf{M}^{-1}\, \mathbf{D}^{-}\, \mathbf{W} \\
&\mathbf{W} = \left\{ w_i\, \delta_{ij} \right\} \quad (i, j = 1, \ldots, N) \\
&\mathbf{D}^{+} = \left\{ D^m(\mu_i, \mu_j) \right\} = \left\{ D^m(-\mu_i, -\mu_j) \right\} \quad (i, j = 1, \ldots, N) \\
&\mathbf{D}^{-} = \left\{ D^m(-\mu_i, \mu_j) \right\} = \left\{ D^m(\mu_i, -\mu_j) \right\} \quad (i, j = 1, \ldots, N)
\end{aligned} \tag{15b}$$

and where we have used $\Im$ for the identity matrix to distinguish it better from the intensity vectors.

The special structure of the ($2N$x$2N$) matrix in (14e),

$$\begin{bmatrix} -\boldsymbol{\alpha} & -\boldsymbol{\beta} \\ \boldsymbol{\beta} & \boldsymbol{\alpha} \end{bmatrix}$$

can be traced to the reciprocity principle, which for single scattering is due to the fact that the phase function depends only on the scattering angle Θ in (4) (C/IV.29). This special structure is also a consequence of having chosen a quadrature rule satisfying $\mu_{-i}=-\mu_i$ and $w_{-i}=w_i$. As we shall see later, this special structure leads to eigensolutions with eigenvalues occurring in positive/negative pairs. This allows a reduction in the order of the resulting algebraic eigenvalue problem by a factor of 2, which decreases the computational burden by a factor 8 (computation time for eigensolution algorithms grows roughly as the cube of the matrix dimension).

2.3 Quadrature Rule

There are many quadrature rules which satisfy the requirements given above, but the use of Gaussian quadrature is essential because it makes phase function renormalization unnecessary, i.e.

$$\sum_{\substack{j=-N \\ j\neq 0}}^{N} w_j \, D^0(\tau, \mu_i, \mu_j) = \sum_{\substack{i=-N \\ i\neq 0}}^{N} w_i \, D^0(\tau, \mu_i, \mu_j) = \omega(\tau) \tag{16}$$

implying that energy is conserved in the computation (cf. Wiscombe, 1977). The reason for this is simply that the Gaussian rule is based on the zeros of the Legendre polynomials which are also used for expanding the phase function.

The quadrature points and weights of the "Double-Gauss" scheme adopted here satisfy $\mu_{-i}=-\mu_i$ and $w_{-i}=w_i$. "Double-Gauss" is a quadrature rule suggested by Sykes (1951) in which Gaussian quadrature is applied separately to the half-ranges $-1<\mu<0$ and $0<\mu<1$. The main advantage is that the quadrature points (in even orders) are distributed symmetrically around $|\mu|=0.5$ and clustered both towards $|\mu|=1$ and $\mu=0$, whereas in the Gaussian scheme for the complete range $-1<\mu<1$, they are clustered only towards $\mu=-1$ and $\mu=+1$. The Double-Gauss clustering towards $\mu=0$ will give superior results anywhere the intensity varies rapidly across $\mu=0$, especially at the boundaries where the intensity is often discontinuous at $\mu=0$. Another advantage is that upward and downward fluxes are obtained immediately without any further approximations. Thomas and Stamnes (1999) provide a more detailed discussion of the Double-Gauss quadrature scheme.

2.4 Homogeneous Solution

It is traditional in solving linear ordinary differential equations (ODEs) to break the solution into two parts, called the *homogeneous* and *particular* solutions. The homogeneous solution satisfies the ODEs with no source term, or forcing, and doesn't have to satisfy the boundary conditions; it typically contains arbitrary constants. The particular solution is a solution with the source terms included and no arbitrary constants, but again not required to satisfy the boundary conditions. The general solution is the sum of the homogeneous and particular solutions, which is additionally required to satisfy the boundary conditions. The boundary conditions are satisfied by solving linear algebraic equations for the arbitrary constants in the homogeneous solution.

Thus the whole process breaks up into three distinct parts: finding the homogeneous solution,

finding the particular solution, and satisfying the boundary conditions.

2.4.1 Two-stream approximation (N=1):

Setting $Q'^{\pm}=0$ in (12) gives the homogeneous version of the two-stream approximation. Since these are just linear ODE's, it is traditional to seek solutions of the form

$$\mathbf{I}^{\pm} = G^{\pm} e^{-k\tau} \quad \text{where } G^{\pm} \equiv G(\pm\mu_1)$$

which leads to the following algebraic eigenvalue problem:

$$\begin{bmatrix} \alpha & \beta \\ -\beta & -\alpha \end{bmatrix} \begin{bmatrix} G^+ \\ G^- \end{bmatrix} = k \begin{bmatrix} G^+ \\ G^- \end{bmatrix} \tag{17a}$$

where k is an eigenvalue and G is an eigenvector. Expanding this into two scalar equations,

$$\alpha G^+ + \beta G^- = kG^+$$

$$-\beta G^+ - \alpha G^- = kG^-$$

and adding and subtracting these two equations, we find

$$(\alpha - \beta)(G^+ - G^-) = k(G^+ + G^-) \tag{17b}$$

$$(\alpha + \beta)(G^+ + G^-) = k(G^+ - G^-) \tag{17c}$$

Substitution of (17c) into (17b) yields

$$(\alpha - \beta)(\alpha + \beta)(G^+ + G^-) = k^2(G^+ + G^-) \tag{17d}$$

Clearly the scalar factor $(G^+ + G^-)$ can be canceled from both sides, and so

$$G^+ + G^- = \gamma \text{ (arbitrary scalar constant)} \tag{18a}$$

With this factor gone from both sides of (17d), taking the square root yields the two eigenvalues:

$$\begin{aligned} &k_1 = k, \; k_{-1} = -k \\ &k \equiv \sqrt{\alpha^2 - \beta^2} = \frac{1}{\mu}\sqrt{(1-\omega)(1-\omega+2\omega\eta)} > 0 \qquad (\omega < 1) \end{aligned} \tag{18b}$$

For $k_1 = k$, (17c) together with (18a) yields

$$G_1^+ - G_1^- = \frac{\alpha+\beta}{k}\gamma \tag{19}$$

(assuming $k \neq 0$ or, equivalently, $\omega \neq 1$), where the subscript "1" refers to the first eigenvalue. Adding and subtracting (18a) and (19) gives

$$\begin{aligned} G_1^+ &= \tfrac{1}{2}\left[1+\frac{\alpha+\beta}{k}\right]\gamma \\ G_1^- &= \tfrac{1}{2}\left[1-\frac{\alpha+\beta}{k}\right]\gamma \end{aligned} \tag{20a}$$

Picking γ such that $G_1^-=1$, the eigenvector corresponding to eigenvalue $k_1=k$ is:

$$\begin{bmatrix} G_1^+ \\ G_1^- \end{bmatrix} = \begin{bmatrix} R \\ 1 \end{bmatrix} \tag{20b}$$

where

$$R \equiv \frac{k+(\alpha+\beta)}{k-(\alpha+\beta)}$$

Repeating this for $k_{-1}=-k$, we find

$$\begin{bmatrix} G_{-1}^+ \\ G_{-1}^- \end{bmatrix} = \begin{bmatrix} 1 \\ R \end{bmatrix} \tag{20c}$$

The complete homogeneous solution is a linear combination of the exponential solutions for eigenvalues $k_1=k$ and $k_{-1}=-k$, i.e.

$$I(\tau,\mu_i) = \sum_{\substack{j=-1 \\ j\neq 0}}^{1} C_j \; G_j(\mu_i)\, e^{-k_j \tau} \quad (i=-1,+1) \tag{20d}$$

where C_{-1} and C_1 are constants of integration. We have written this in a more formal notation than necessary, for later generalization.

2.4.2 ω=1 special case

Note in (18b) that ω=1 leads to k=0. This in turn causes apparent infinities in (19) and (20). Vanishing eigenvalues is but one of many difficulties that the ω=1 special case causes.

We show no solutions for the special case ω=1. While ω=1 has received great theoretical attention, it cannot be realized in practice because all materials are absorbing, if only slightly so — a point made forcefully by Bohren and Huffman (1983) in their discussion of imaginary refractive indices. In fact, a complete lack of absorption is not physically possible. Thus, ω=1 can only be realized as an idealized limit from below (ω—>1–) as absorption approaches zero. As one would expect, the special-case formulas for ω=1 are proper limits of the general-case formulas,

$$\{general\ case\ \ formulas\} \xrightarrow[\omega \to 1-]{} \{\omega = 1\ special\ case\ \ formulas\}$$

but this limit involves 0/0 singularities which must be treated by L'Hospital's Rule. This leads to entirely different functions than the exponentials found in the general-case formula. Having different functions greatly complicates the interface conditions between layers with ω=1 and those with ω<1. Furthermore, the approach to ω=1 is cusp-like,

$$I(\omega) = I(\omega = 1) + c\,(1-\omega)^{1/2}$$

which can cause numerical difficulties.

Finally, there is the issue of introducing an arbitrary discontinuity. One has to specify how close to ω=1 one must get before invoking to the ω=1 solution. Using this solution only for ω=1 is not correct, because then cases close to ω=1 will suffer large ill-conditioning errors. Does one invoke the ω=1 solution when ω=1–10^{-6}, or ω=1–10^{-10}, or when ω=1 to machine precision? In the end, this choice is rather arbitrary, although necessarily guided by computer precision. Whatever choice one makes, a discontinuity in radiance is possible unless one carefully patches the ω=1 and ω<1 solutions together using perturbation expansions like that indicated above.

In an early version of DISORT we actually had a forest of IF statements dealing with this special case (although not in the correct perturbation-expansion way). But it seemed inelegant and wasteful to devote so much branching (a major code slowdown) to a case which is in reality only an idealized limit. The resulting code was hard to read and hard to debug, especially the interface conditions. So we decided to include only the ω<1 formulas and to deal with the ω=1 case by *dithering*. Dithering simply means replacing ω=1 by ω=1–ε wherever it occurs. (ε is taken to be 10 times machine precision or, in the recommended IEEE double precision mode of operation,

about 10^{-13}.) This basically gets the computer to do the L'Hospital's Rule computation, but digitally instead of analytically. It allows the ω<1 solution to be used in every layer, vastly simplifying the interface conditions. This procedure has worked well in practice, and has allowed us to match theoretical ω=1 solutions arbitrarily closely (except for semi-infinite media).

2.4.3 Four-stream approximation (N=2)

For the four-stream case, the structure of (14e) is similar to that of (12), and we may proceed as in the two-stream case by setting

$$\mathbf{I}^{\pm} = \mathbf{G}^{\pm} e^{-k\tau}, \quad \mathbf{G}^{\pm} = \begin{bmatrix} G(\pm\mu_1) \\ G(\pm\mu_2) \end{bmatrix}$$

Repetition of the procedure in the two-stream case then leads us directly to equations analogous to (17b-c), i.e.

$$(\boldsymbol{\alpha} - \boldsymbol{\beta})(\mathbf{G}^{+} - \mathbf{G}^{-}) = k(\mathbf{G}^{+} + \mathbf{G}^{-}) \tag{21a}$$

$$(\boldsymbol{\alpha} + \boldsymbol{\beta})(\mathbf{G}^{+} + \mathbf{G}^{-}) = k(\mathbf{G}^{+} - \mathbf{G}^{-}) \tag{21b}$$

and substitution of (21b) into (21a) yields

$$(\boldsymbol{\alpha} - \boldsymbol{\beta})(\boldsymbol{\alpha} + \boldsymbol{\beta})(\mathbf{G}^{+} + \mathbf{G}^{-}) = k^2(\mathbf{G}^{+} + \mathbf{G}^{-}) \tag{21c}$$

Since $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are 2x2 matrixes, there are two eigenvalues k_a^2 and k_b^2 and thus a total of four eigenvalues occurring in positive/negative pairs:

$$k_1 = k_a > 0, \quad k_{-1} = -k_a$$
$$k_2 = k_b > 0, \quad k_{-2} = -k_b$$

A solution of (21c) also yields eigenvectors

$$\mathbf{G}^{+} + \mathbf{G}^{-} = \boldsymbol{\gamma} \quad \text{(constant vector)} \tag{22a}$$

corresponding to eigenvalues k_a and k_b. Using (21b) we find that (assuming $k \neq 0$ or equivalently that ω<1)

$$\mathbf{G}^{+} - \mathbf{G}^{-} = \frac{1}{k}(\boldsymbol{\alpha} + \boldsymbol{\beta})\,\boldsymbol{\gamma} \tag{22b}$$

Combining (22a) and (22b) we find

$$\begin{aligned} \mathbf{G}^{+} &= \tfrac{1}{2}\left[\boldsymbol{\gamma} + \frac{1}{k}(\boldsymbol{\alpha}+\boldsymbol{\beta})\,\boldsymbol{\gamma}\right] \\ \mathbf{G}^{-} &= \tfrac{1}{2}\left[\boldsymbol{\gamma} - \frac{1}{k}(\boldsymbol{\alpha}+\boldsymbol{\beta})\,\boldsymbol{\gamma}\right] \end{aligned} \tag{22c}$$

which are the eigenvectors corresponding to the positive eigenvalues ($k=k_a$ and k_b). The eigenvectors corresponding to negative eigenvalues ($k=-k_a$ and $-k_b$) are similar:

$$\begin{aligned} \mathbf{G}^{+} &= \tfrac{1}{2}\left[\boldsymbol{\gamma} - \frac{1}{k}(\boldsymbol{\alpha}+\boldsymbol{\beta})\,\boldsymbol{\gamma}\right] \\ \mathbf{G}^{-} &= \tfrac{1}{2}\left[\boldsymbol{\gamma} + \frac{1}{k}(\boldsymbol{\alpha}+\boldsymbol{\beta})\,\boldsymbol{\gamma}\right] \end{aligned} \tag{22d}$$

Since we have now constructed a complete set of eigenvectors we may write down the homogeneous solution as a linear combination of the solutions for different eigenvalues as follows:

$$I(\tau,\mu_i) = \sum_{\substack{j=-2 \\ j\neq 0}}^{2} C_j\, G_j(\mu_i)\, e^{-k_j\tau} \quad (i=-2,-1,+1,+2)$$

where the C_j are constants of integration.

2.4.4 Multi-stream approximation (N arbitrary):

(14e) for N arbitrary is a system of $2N$ coupled, ordinary differential equations with constant coefficients. These coupled equations are linear and we can uncouple them using the same technique we used in the two- and four-stream cases.

Seeking solutions to the homogeneous version (**Q**'=0) of (14e) of the form

$$\mathbf{I}^{\pm} = \mathbf{G}^{\pm} e^{-k\tau} \tag{23a}$$

we find

$$\begin{bmatrix} \boldsymbol{\alpha} & \boldsymbol{\beta} \\ -\boldsymbol{\beta} & -\boldsymbol{\alpha} \end{bmatrix} \begin{bmatrix} \mathbf{G}^+ \\ \mathbf{G}^- \end{bmatrix} = k \begin{bmatrix} \mathbf{G}^+ \\ \mathbf{G}^- \end{bmatrix} \tag{23b}$$

which is a standard algebraic eigenvalue problem of order $2N$x$2N$ determining the eigenvalues k and the eigenvectors $\mathbf{G}^{\pm}$.

As noted previously, because of the special structure of the matrix in (23b), the eigenvalues occur in positive/negative pairs and the order of this algebraic eigenvalue problem may be reduced as follows (Stamnes and Swanson, 1981). Rewriting the homogeneous version of (14e) as

$$\frac{\mathrm{d}\mathbf{I}^+}{\mathrm{d}\tau} = -\boldsymbol{\alpha}\mathbf{I}^+ - \boldsymbol{\beta}\mathbf{I}^-$$

$$\frac{\mathrm{d}\mathbf{I}^-}{\mathrm{d}\tau} = \quad \boldsymbol{\beta}\mathbf{I}^+ + \boldsymbol{\alpha}\mathbf{I}^-$$

and then adding and subtracting these two equations, we find

$$\frac{\mathrm{d}(\mathbf{I}^+ + \mathbf{I}^-)}{\mathrm{d}\tau} = -(\boldsymbol{\alpha} - \boldsymbol{\beta})(\mathbf{I}^+ - \mathbf{I}^-) \tag{23c}$$

$$\frac{\mathrm{d}(\mathbf{I}^+ - \mathbf{I}^-)}{\mathrm{d}\tau} = -(\boldsymbol{\alpha} + \boldsymbol{\beta})(\mathbf{I}^+ + \mathbf{I}^-) \tag{23d}$$

Combining (23c) and (23d), we obtain

$$\frac{\mathrm{d}^2(\mathbf{I}^+ + \mathbf{I}^-)}{\mathrm{d}\tau^2} = (\boldsymbol{\alpha} - \boldsymbol{\beta})(\boldsymbol{\alpha} + \boldsymbol{\beta})(\mathbf{I}^+ + \mathbf{I}^-) \tag{23e}$$

or in view of (23a)

$$(\boldsymbol{\alpha} - \boldsymbol{\beta})(\boldsymbol{\alpha} + \boldsymbol{\beta})(\mathbf{G}^+ + \mathbf{G}^-) = k^2(\mathbf{G}^+ + \mathbf{G}^-) \tag{23f}$$

which completes the reduction of the order. Following Stamnes and Swanson (1981) we solve (23f) to obtain eigenvalues and eigenvectors ($\mathbf{G}^+ + \mathbf{G}^-$). We then use (23d) to determine ($\mathbf{G}^+ - \mathbf{G}^-$), and proceed as in the four-stream case to construct a complete set of eigenvectors.

2.5 Particular Solution

For beam sources (cf. 8c)

$$Q^{(beam)}(\tau,\mu) = X_0(\mu)\, e^{-\tau/\mu_0}$$

and it is easily verified that a particular solution of (10a) is (omitting *m*-superscript)

$$I(\tau,\mu_i) = Z_0(\mu_i)\, e^{-\tau/\mu_0} \tag{24a}$$

where the Z_0 are determined by the following system of linear algebraic equations

$$\sum_{\substack{j=-N \\ j\neq 0}}^{N} \left[\left(1+\frac{\mu_j}{\mu_0} \right) \delta_{ij} - w_j\, D(\mu_i,\mu_j) \right] Z_0(\mu_j) = X_0(\mu_i) \tag{24b}$$

In the two-stream case (24b) reduces to two algebraic equations with two unknowns, which is easily solved in closed form. The four-stream case involves four algebraic equations and may also be solved analytically, but it is quite difficult to construct an analytic solution which is perfectly protected from ill-conditioning (loss of accuracy due to subtraction of nearly-equal numbers). Thus, already at this level, it is preferable to use robust linear equation solving software, which protects against ill-conditioning automatically.

For thermal sources the emitted radiation is isotropic, so that

$$Q^0(\tau) = (1-\omega)\, B[T(\tau)]$$

$$Q^m(\tau) = 0 \quad (m>0)$$

The Planck function *B* is assumed to vary linearly in optical depth across each layer,

$$B[T(\tau)] = b_0 + b_1\tau \tag{24c}$$

which is an approximation hallowed by previous practice (Appendix A of Wiscombe, 1976, gives the most thorough extant analysis of it). The two coefficients b_0, b_1 are chosen to match *B* at the top and bottom layer boundaries τ_t and τ_b (where the temperature is known from user input)

$$B_t \equiv B\left[T(\tau_t)\right] = b_0 + b_1\tau_t$$

$$B_b \equiv B\left[T(\tau_b)\right] = b_0 + b_1\tau_b$$

Note that even though Planck emission itself is isotropic, the radiation emerging from a layer across which the temperature changes will not be isotropic (you 'see' down to different temperatures depending on what angle you look at).

Since the thermal source term assumes the form

$$Q^{(\text{thermal})}(\tau) = (1-\omega)\left(b_0 + b_1\tau\right) \tag{24d}$$

it suggests trying a particular solution of the form

$$I(\tau,\mu_i) = Y_0(\mu_i) + Y_1(\mu_i)\,\tau \tag{25}$$

Substituting (25) into (10) and equating coefficients of like powers in τ, we find

$$\sum_{\substack{j=-N \\ j\neq 0}}^{N}\left[\delta_{ij} - w_j\, D^0(\mu_i,\mu_j)\right] Y_1(\mu_j) = (1-\omega)\, b_1 \tag{26a}$$

$$\sum_{\substack{j=-N \\ j\neq 0}}^{N}\left[\delta_{ij} - w_j\, D^0(\mu_i,\mu_j)\right] Y_0(\mu_j) = (1-\omega)\; b_0 + \mu_i\, Y_1(\mu_i) \tag{26b}$$

(26) are a system of linear algebraic equations determining the $Y_l(\mu_i)$.

In the two- and four-stream cases, (24b) and (26) reduce to two and four algebraic equations, respectively, which may be solved analytically. But the numerical results obtained from the four-stream analytic expressions may be inferior to those obtained using linear equation solving software like LAPACK.

2.6 General Solution

The general solution to (10a) consists of a linear combination of all the homogeneous solutions, plus the particular solutions for beam and thermal emission sources (omitting *m*-superscript):

$$I(\tau,\mu_i) = \sum_{j=-N}^{N} C_j\, G_j(\mu_i)\, e^{-k_j\tau} + \delta_{m0}\left[Y_0(\mu_i) + Y_1(\mu_i)\,\tau\right] \tag{27a}$$

where for convenience the beam particular solution is included in the sum by defining, for j=0,

$$C_0\ G_0(\mu_i) = Z_0(\mu_i) \tag{27b}$$

$$k_0 = 1/\mu_0 \tag{27c}$$

The k_j and $G_j(\mu_i)$ for $j\neq 0$ are the eigenvalues and eigenvectors obtained as described in Section 2.4, the μ_i the cosines of the quadrature angles, and the C_j the constants of integration to be determined by the boundary and layer continuity conditions.

2.7 Intensities at Arbitrary Angles

For a slab of thickness τ_L, we may solve (7) formally to obtain ($\mu>0$)

$$I(\tau,+\mu) = I(\tau_L,+\mu)\, e^{-(\tau_L-\tau)/\mu} + \int_\tau^{\tau_L} S(t,+\mu)\, e^{-(t-\tau)/\mu}\, \frac{\mathrm{d}t}{\mu} \tag{28a}$$

$$I(\tau,-\mu) = I(0,-\mu)\, e^{-\tau/\mu} + \int_0^{\tau} S(t,-\mu)\, e^{-(\tau-t)/\mu}\, \frac{\mathrm{d}t}{\mu} \tag{28b}$$

where we have again omitted the m-superscript. These two equations show that if we know the source function $S(\tau,\pm\mu)$, we can find the intensity at arbitrary angles by integrating the source function. Below we shall use the discrete ordinate solutions to derive explicit expressions for the source function which can be integrated analytically. This procedure is sometimes referred to as the "iteration of the source-function technique", although as pointed out by Kourganoff (1963) and discussed by Stamnes (1982a) and Schulz and Stamnes (2000), it essentially amounts to an intelligent interpolation.

The source function given by (8) can, in view of (24d), be written as

$$S(\tau,\mu) = \sum_{\substack{i=-N \\ i\neq 0}}^{N} w_i\, D(\mu,\mu_i)\, I(\tau,\mu_i) + X_0(\mu)\, e^{-\tau/\mu_0} + \delta_{m0}\,(1-\omega)\left(b_0 + b_1\tau\right) \tag{29}$$

Substituting the general solution of (27) into (29), we find

$$S(\tau,\mu) = \sum_{j=-N}^{N} C_j\, G_j(\mu)\, e^{-k_j\tau} + \delta_{m0}\left[V_0(\mu) + V_1(\mu)\,\tau\right] \tag{30}$$

where

$$G_0(\mu) = Z_0(\mu) = \sum_{\substack{i=-N \\ i\neq 0}}^{N} w_i\, D(\mu, \mu_i)\, Z_0(\mu_i) + X_0(\mu) \tag{31a}$$

$$\text{for } j \neq 0,\ G_j(\mu) = \sum_{\substack{i=-N \\ i\neq 0}}^{N} w_i\, D(\mu, \mu_i)\, G_j(\mu_i) \tag{31b}$$

$$V_\ell(\mu) = \sum_{\substack{i=-N \\ i\neq 0}}^{N} w_i\, D^0(\mu, \mu_i)\, Y_\ell(\mu_i) + (1-\omega)\, b_\ell \quad (\ell = 1, 0) \tag{31c}$$

These formulas clearly reveal the interpolatory aspect mentioned above. Although we have generally omitted the *m*-superscript above, we have explicitly written D^0 in (26) and (31c) as a reminder that thermal emission contributes only to the azimuth-independent component of the intensity.

2.8 Angular Distributions

We now have all the information needed to compute $V_l(\mu)$ from (31c) and hence the source function (30). Thus, we may proceed to compute the angular distributions of the intensity. Before we treat the general multi-layer case with all its complicated formulas, however, we do the single-layer case (L=1) to try to achieve some understanding.

2.8.1. Single homogeneous layer

Plugging the source function (30) into the equations for the interpolated intensity (28) and doing the simple integrals of exponentials analytically, we find that for a single, homogeneous layer, the intensities become (μ>0):

$$I(\tau,+\mu) = I(\tau_1,+\mu)e^{-(\tau_1-\tau)/\mu} + \sum_{j=-N}^{N} C_j \frac{G_j(+\mu)}{1+k_j\mu}\left\{e^{-k_j\tau} - e^{-[k_j\tau_1+(\tau_1-\tau)/\mu]}\right\}$$

$$+\delta_{m0}\left\{ V_0(+\mu)\left[1-e^{-(\tau_1-\tau)/\mu}\right] \right. \tag{33a}$$

$$\left. +V_1(+\mu)\left[(\tau+\mu)-(\tau_1+\mu)e^{-(\tau_1-\tau)/\mu}\right]\right\}$$

$$I(\tau,-\mu) = I(0,-\mu)\, e^{-\tau/\mu} + \sum_{j=-N}^{N} C_j \frac{G_j(-\mu)}{1-k_j\mu}\left\{e^{-k_j\tau} - e^{-\tau/\mu}\right\}$$

$$+\delta_{m0}\left\{ V_0(-\mu)\left[1-e^{-\tau/\mu}\right] \right. \tag{33b}$$

$$\left. +V_1(-\mu)\left[(\tau-\mu)+\mu\, e^{-\tau/\mu}\right]\right\}$$

2.8.2 Multiple layers

In a multi-layered medium we evaluate the integral in (28a,b) by integrating layer by layer (cf. Stamnes, 1982a) as follows ($\tau_{p-1}\le\tau\le\tau_p$ and $\mu>0$):

$$\int_{\tau}^{\tau_L} S(t,+\mu)e^{-(t-\tau)/\mu}\frac{\mathrm{d}t}{\mu} = \int_{\tau}^{\tau_p} S_p(t,+\mu)e^{-(t-\tau)/\mu}\frac{\mathrm{d}t}{\mu}$$

$$+\sum_{n=p+1}^{L}\left\{\int_{\tau_{n-1}}^{\tau_n} S_n(t,+\mu)e^{-(t-\tau)/\mu}\frac{\mathrm{d}t}{\mu}\right\} \tag{34a}$$

$$\int_{0}^{\tau} S(t,-\mu)e^{-(\tau-t)/\mu}\frac{\mathrm{d}t}{\mu} = \sum_{n=1}^{p-1}\left\{\int_{\tau_{n-1}}^{\tau_n} S_n(t,-\mu)e^{-(\tau-t)/\mu}\frac{\mathrm{d}t}{\mu}\right\}$$

$$+\int_{\tau_{p-1}}^{\tau} S_p(t,-\mu)e^{-(\tau-t)/\mu}\frac{\mathrm{d}t}{\mu} \tag{34b}$$

Using (30) for $S_n(t,\mu)$ in (34), we find

$$I_p(\tau,+\mu) = I(\tau_L,+\mu)\,e^{-(\tau_L-\tau)/\mu} + \sum_{n=p}^{L}\left\{\begin{array}{l}\displaystyle\sum_{j=-N}^{N} C_{jn}\,\frac{G_{jn}(+\mu)}{1+k_{jn}\mu}\,E_{jn}(\tau,+\mu) \;+ \\ \delta_{m0}\left[V_{0n}(+\mu)\,F_{0n}(\tau,+\mu)+V_{1n}(+\mu)\,F_{1n}(\tau,+\mu)\right]\end{array}\right\} \tag{35a}$$

$$I_p(\tau,-\mu) = I(0,-\mu)\,e^{-\tau/\mu} + \sum_{n=1}^{p}\left\{\begin{array}{l}\displaystyle\sum_{j=-N}^{N} C_{jn}\,\frac{G_{jn}(-\mu)}{1-k_{jn}\mu}\,E_{jn}(\tau,-\mu) \;+ \\ \delta_{m0}\left[V_{0n}(-\mu)\,F_{0n}(\tau,-\mu)+V_{1n}(-\mu)\,F_{1n}(\tau,-\mu)\right]\end{array}\right\} \tag{35b}$$

where

$$E_{jn}(\tau,+\mu) = \exp\left(-k_{jn}\,\tau_{n-1}-(\tau_{n-1}-\tau)/\mu\right)-\exp\left(-k_{jn}\,\tau_n-(\tau_n-\tau)/\mu\right) \tag{36a}$$

$$F_{0n}(\tau,+\mu) = \exp\left(-(\tau_{n-1}-\tau)/\mu\right)-\exp\left(-(\tau_n-\tau)/\mu\right) \tag{36b}$$

$$\begin{aligned}F_{1n}(\tau,+\mu) = {} & \left(\tau_{n-1}+\mu\right)\exp\left(-(\tau_{n-1}-\tau)/\mu\right)- \\ & \left(\tau_n+\mu\right)\exp\left(-(\tau_n-\tau)/\mu\right)\end{aligned} \tag{36c}$$

with τ_{n-1} replaced by τ for $n=p$, and

$$E_{jn}(\tau,-\mu) = \exp\left\{-k_{jn}\,\tau_n-(\tau-\tau_n)/\mu\right\}-\exp\left\{-k_{jn}\,\tau_{n-1}-(\tau-\tau_{n-1})/\mu\right\} \tag{37a}$$

$$F_{0n}(\tau,-\mu) = \exp\left\{-(\tau-\tau_n)/\mu\right\}-\exp\left\{-(\tau-\tau_{n-1})/\mu\right\} \tag{37b}$$

$$\begin{aligned}F_{1n}(\tau,-\mu) = {} & \left(\tau_n-\mu\right)\exp\left\{-(\tau-\tau_n)/\mu\right\}- \\ & \left(\tau_{n-1}-\mu\right)\exp\left\{-(\tau-\tau_{n-1})/\mu\right\}\end{aligned} \tag{37c}$$

with τ_n replaced by τ for $n=p$.

One may immediately verify that, for one layer ($\tau_{n-1}=\tau$, $\tau_n=\tau_L=\tau_1$ in [35a]; $\tau_{n-1}=0$, $\tau_n=\tau$ in [35b]), (35) reduce to (33) as they should.

The basic soundness and merit of the intensity expressions given above has been discussed by Stamnes (1982a). We shall not repeat that discussion here, but the main findings were:

(1) For beam sources (thermal sources were not considered in that paper), (35a,b) when evaluated at the quadrature points yields results identical to (27a). This clearly demonstrates the interpolatory nature of (31a,b).

(2) (35a,b) have the merit of "correcting" the simpler expression (27a) for μ–values not coinciding with the quadrature points.

(3) (35a,b) satisfy the boundary and continuity conditions for all μ–values (even though we have imposed such conditions only at the quadrature points!).

2.9 Boundary and Interface Conditions

The treatment of the lower boundary has been considerably generalized in DISORT v2.0 compared to v1.x. The non-Lambertian surface reflection option in v1.x, where the surface BRDF depended *only* on the angle between incident and reflected beams, was unphysical and justly criticized by users (e.g. Godsalve, 1995). Now, a more general and realistic BRDF function can be input. From the user's point of view, it involves the following changes:

(a) the input variable HL, supplying the Legendre coefficients of the v1.x unphysical BRDF function (which depended on only a single angle and thus could be expanded in Legendre polynomials), is deleted;

(b) a subroutine BDREF must be supplied by the user, which provides the bidirectional reflectance of the surface as a function of three angles: incident polar angle, reflected polar angle, and azimuth angle between the incident and reflected directions (no provision is made for surfaces with a preferred orientation, like a plowed field, which would add a second azimuth angle);

(c) the user shoulders a good part of the responsibility to make sure that the supplied BRDF function is debugged, physically reasonable and conserves energy; other than gross violations (surface albedos below zero or above unity), DISORT has no way to test the supplied function; since most natural BRDF's do not even satisfy reciprocity (due to spatial and/or spectral averaging), DISORT cannot even test for this.

In general, the radiative transfer equation (1) must be solved subject to the boundary conditions

$$I(\boldsymbol{\tau} = \mathbf{0}, -\mu, \phi) = I_{top}(\mu, \phi) \tag{38a}$$

$$I(\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathbf{L}}, +\mu, \phi) = I_{g}(\mu, \phi) \tag{38b}$$

where $\mu>0$ restricts the conditions to the down and up hemispheres, respectively; I_{top} and I_g are the intensities incident at the top and bottom boundaries, respectively; and τ_L is the total optical depth of the entire medium.

The discrete ordinate method can handle boundary conditions of this generality. But to be specific, and to apply to planetary atmospheres, oceans, and ice, DISORT assumes that the medium is illuminated at the top boundary by known diffuse radiation I_{top} (parallel beams are treated as pseudo-sources) and that the bottom boundary has a known bidirectional reflectivity; these quantities are supplied by the user. (Emissivity is computed from the bidirectional reflectivity, if needed.) In the current DISORT program, I_{top} is restricted to be a constant, with the idea that this may usefully approximate a thermally emitting upper boundary, or a highly scattering one such as a cloud. But this restriction could easily be relaxed. Thus, the medium may be driven by a combination of uniform diffuse and parallel beam illumination at the top boundary.

Now we formulate the bottom boundary condition (38b) in terms of a bidirectional reflectivity and emissivity, as follows:

$$\begin{aligned} I(\boldsymbol{\tau} = \boldsymbol{\tau}_{\boldsymbol{L}}, +\mu, \phi) = \varepsilon(\mu)\, B(T_g) + \frac{1}{\pi}\, \mu_0\, I_0\, e^{-\tau_L/\mu_0}\, \rho_d(\mu, \phi; -\mu_0, \phi_0) \\ + \frac{1}{\pi} \int_0^{2\pi} \mathrm{d}\phi' \int_0^1 \rho_d(\mu, \phi; -\mu', \phi')\, I(\tau_L, -\mu', \phi')\, \mu'\, \mathrm{d}\mu' \end{aligned} \tag{39}$$

where ρ_d is the bidirectional reflectivity, ε the directional emissivity and T_g the temperature of the bottom boundary. I_0 is the intensity of the incident beam at the top boundary. Kirchhoff's law allows us to get the emissivity from the bidirectional reflectivity:

$$\varepsilon(\mu) + \frac{1}{\pi} \int_0^{2\pi} \mathrm{d}\phi' \int_0^1 \rho_d(\mu, \phi; -\mu', \phi')\, \mu'\, \mathrm{d}\mu' = 1$$

Note that there are several definitions of surface bidirectional reflectivity differing by various multiplicative factors. Our definition (39) follows Chapter 3 of Siegel and Howell (1992), which

provides the most thorough and consistent definitions we have seen. Hapke's (1993) definition must be multiplied by π and divided by the cosine of incident angle to agree with DISORT's definition.

Now we will make the usual assumption that the reflecting surface is isotropic; examples of non-isotropic surfaces (with preferred directions) would include plowed fields, corn fields, orchards, oriented sand dunes, and snow dunes in Antarctica ("sastrugi"). This isotropic assumption means that the azimuthal dependence of ρ_d depends only on the difference $\phi-\phi'$ between the azimuthal angles of the incident and reflected radiation. This enables us to separate the Fourier components by expanding the bidirectional reflectivity in a Fourier cosine series of $2M$ terms just like the intensity (6):

$$\rho_d(\mu,\phi;-\mu',\phi') = \rho_d(\mu,-\mu';\phi-\phi') = \sum_{m=0}^{2M-1} \rho_d^m(\mu,-\mu')\cos m(\phi-\phi') \qquad (40)$$

where the expansion coefficients are computed from

$$\rho_d^m(\mu,-\mu') = \frac{1}{\pi}\frac{2-\delta_{m0}}{2}\int_{-\pi}^{\pi} \rho_d(\mu,-\mu';\phi-\phi')\cos m(\phi-\phi')\ \mathrm{d}(\phi-\phi') \qquad (41)$$

Substituting (40) into (39) and using (6), we find that each Fourier component must satisfy the bottom boundary condition

$$I^m(\tau_L,+\mu) = I_g^m(\mu) \qquad (42a)$$

where

$$\begin{aligned} I_g^m(\mu) \equiv \delta_{m0}\,\varepsilon(\mu)\,B(T_g) + \frac{1}{\pi}\mu_0 I_0 e^{-\tau_L/\mu_0}\,\rho_d^m(\mu,-\mu_0) + \\ (1+\delta_{m0})\int_0^1 \rho_d^m(\mu,-\mu')\,I^m(\tau_L,-\mu')\,\mu'\,\mathrm{d}\mu' \end{aligned} \qquad (42b)$$

In a multilayered medium we must also require the intensity to be continuous across layer interfaces (Stamnes and Conklin, 1984). Thus, (10) must satisfy boundary and continuity conditions as follows

$$I_1^m(0,-\mu_i) = I_{top}^m(\mu_i) \quad (i=1,\ldots,N) \qquad (43a)$$

$$I_p^m(\tau_p,\mu_i) = I_{p+1}^m(\tau_p,\mu_i) \quad (p=1,\ldots,L-1;\ i=\pm 1,\ldots,\pm N) \tag{43b}$$

$$I_L^m(\tau_L,+\mu_i) = I_g^m(\mu_i) \quad (i=1,\ldots,N) \tag{43c}$$

where

$$I_g^m(\mu_i) = \delta_{m0}\,\varepsilon(\mu_i)\,B(T_g) + \frac{1}{\pi}\mu_0 I_0\, e^{-\tau_L/\mu_0}\,\rho_d^m(\mu_i,-\mu_0) + (1+\delta_{m0})\sum_{j=1}^{N} w_j\mu_j\,\rho_d^m(\mu_i,-\mu_j)\,I^m(\tau_L,-\mu_j) \tag{43d}$$

(this last equation being just the quadratured version of 42b). Since (43a) and (43c) introduce a fundamental distinction between downward directions (denoted by –μ) and upward directions (denoted by +μ), one should select a quadrature rule which integrates separately over the downward and upward directions. As noted previously, the Double-Gauss rule adopted in DISORT satisfies this requirement.

When discussing boundary conditions, it is convenient to write the discrete ordinate solution for the *p*th layer in the following form ($k_{jp}>0$ and $k_{-jp}=-k_{jp}$)

$$I_p(\tau,\mu_i) = \sum_{j=1}^{N}\left\{C_{jp}\,G_{jp}(\mu_i)\,e^{-k_{jp}\tau} + C_{-jp}\,G_{-jp}(\mu_i)\,e^{+k_{jp}\tau}\right\} + U_p(\tau,\mu_i) \tag{44}$$

where the sum contains the homogeneous solution involving the unknown coefficients (C_{jp}) to be determined, and U_p is the particular solution given by (27a):

$$U_p(\tau,\mu_i) = Z_0(\mu_i)\,e^{-\tau/\mu_0} + \delta_{m0}\left[Y_0(\mu_i) + Y_1(\mu_i)\,\tau\right] \tag{45}$$

Insertion of (44) into (43) yields (omitting *m*-superscript)

$$\sum_{j=1}^{N}\left\{C_{j1}\,G_{j1}(-\mu_i) + C_{-j1}\,G_{-j1}(-\mu_i)\right\} = I_{top}(-\mu_i) - U_1(0,-\mu_i) \quad \{i=1,\ldots,N\} \tag{46a}$$

$$\sum_{j=1}^{N} \Big\{ C_{jp}\, G_{jp}(\mu_i) e^{-k_{jp}\tau_p} + C_{-jp}\, G_{-jp}(\mu_i) e^{k_{jp}\tau_p} -$$

$$\Big[C_{j,p+1} G_{j,p+1}(\mu_i) e^{-k_{j,p+1}\tau_p} + C_{-j,p+1}\, G_{-j,p+1}(\mu_i) e^{k_{j,p+1}\tau_p} \Big] \Big\} \qquad (46b)$$

$$= U_{p+1}(\tau_p, \mu_i) - U_p(\tau_p, \mu_i) \qquad \{p = 1, \ldots, L-1;\ i = \pm 1, \ldots, \pm N\}$$

$$\sum_{j=1}^{N} \begin{Bmatrix} C_{jL}\, r_{jL}(\mu_i) G_{jL}(\mu_i) e^{-k_{jL}\tau_L} + \\ C_{-jL}\, r_{-jL}(\mu_i) G_{-jL}(\mu_i) e^{k_{jL}\tau_L} \end{Bmatrix} = \Gamma(\tau_L, \mu_i) \quad \{i = 1, \ldots, N\} \qquad (46c)$$

where

$$r_{jL}(\mu_i) = 1 - (1+\delta_{m0}) \sum_{n=1}^{N} \rho_d(\mu_i, -\mu_n)\, w_n \mu_n\, G_{jL}(-\mu_n) / G_{jL}(\mu_i) \qquad (47a)$$

$$\Gamma(\tau_L, \mu_i) = \delta_{m0}\, \varepsilon(\mu_i)\, B(T_g) - U_L(\tau_L, +\mu_i) +$$

$$\frac{1}{\pi} \mu_0 I_0\, e^{-\tau_L/\mu_0} \rho_d(\mu_i, -\mu_0) + \qquad (47b)$$

$$(1+\delta_{m0}) \sum_{j=1}^{N} \rho_d(\mu_i, -\mu_j)\, w_j \mu_j\, U_L(\tau_L, -\mu_j)$$

(46a-c) constitute a (2*NxL*)x(2*NxL*) system of linear algebraic equations from which the 2*NxL* unknown coefficients C_{jp} (j=±1,...,±*N*; p=1,...,*L*) must be determined. The numerical solution of this set of equations will be discussed in Section 3.5, but first we must deal with the fact that (46a-c) are intrinsically ill-conditioned. Fortunately, this ill-conditioning may be entirely eliminated by a simple scaling transformation discussed below.

2.10 Scaling Transformation

2.10.1 General

As discussed by Stamnes and Conklin (1984), to avoid numerical ill-conditioning, it is necessary to remove the positive exponentials in (46) (remember $k_{jp}>0$ by convention). This is

achieved by the scaling transformation (Stamnes and Conklin, 1984)

$$C_{+jp} = \hat{C}_{+jp}\, e^{k_{jp}\tau_{p-1}} \quad \text{and} \quad C_{-jp} = \hat{C}_{-jp}\, e^{-k_{jp}\tau_p}. \tag{48}$$

Inserting (48) in (46) and solving for the *C*-hat's instead of the *C*'s, we find that all the exponential terms in the coefficient matrix have the form

$$\exp\left(-k_{jp}(\tau_p - \tau_{p-1})\right)$$

so they all have negative arguments ($k_{jp}>0$, $\tau_p>\tau_{p-1}$). This avoids numerical ill-conditioning and makes our numerical scheme for finding the *C*-hat's unconditionally stable for arbitrary total optical depths and arbitrary individual layer thicknesses.

To demonstrate how this scheme works, we shall again use the two-stream case as an example. It is instructive to consider first a one-layer medium and then a two-layer medium. The generalization to a multi-layered medium will then become obvious.

2.10.2 Two-stream, one-layer case (N=1, L=1)

In this simple case, (46a-c) reduce to

$$C_{1,1}\, G_{1,1}(-\mu_1) + C_{-1,1}\, G_{-1,1}(-\mu_1) = (\mathrm{RHS})_1$$

$$r_{1,1}\, C_{1,1}\, G_{1,1}(+\mu_1) e^{-k_1\tau_1} + r_{-1,1}\, C_{-1,1}\, G_{-1,1}(+\mu_1) e^{k_1\tau_1} = (\mathrm{RHS})_2$$

where "RHS" stands for "right hand side". We may write the left hand sides in matrix form as

$$\begin{bmatrix} 1 & R_1 \\ r_{1,1} R_1 e^{-k_1\tau_1} & r_{-1,1} e^{k_1\tau_1} \end{bmatrix} \begin{bmatrix} C_{1,1} \\ C_{-1,1} \end{bmatrix}$$

where we have used (20b) and (20c). This matrix is ill-conditioned when $k_1\tau_1$ becomes large due to the occurrence of the positive exponential term $\exp(k_1\tau_1)$. Scaling removes this positive exponential. Application of (48) yields

$$\begin{bmatrix} 1 & R_1 e^{-k_1\tau_1} \\ r_{1,1} R_1 e^{-k_1\tau_1} & r_{-1,1} \end{bmatrix} \begin{bmatrix} \hat{C}_{1,1} \\ \hat{C}_{-1,1} \end{bmatrix}$$

In the limit of large values of $k_1\tau_1$ this matrix becomes

$$\begin{bmatrix} 1 & 0 \\ 0 & r_{-1,1} \end{bmatrix}$$

which shows that the ill-conditioning problem has been entirely eliminated.

2.10.3 Two-stream, two-layer case (N=1, L=2)

Proceeding as in the one-layer case above, we write the left-hand side of (46) out explicitly in the two-stream approximation for a two-layer medium as follows:

$$C_{1,1}G_{1,1}(-\mu_1) + C_{-1,1}G_{-1,1}(-\mu_1) = (\text{RHS})_1$$

$$C_{1,1}G_{1,1}(-\mu_1)e^{-k_1\tau_1} + C_{-1,1}G_{-1,1}(-\mu_1)e^{k_1\tau_1} - C_{1,2}G_{1,2}(-\mu_1)e^{-k_2\tau_1} - C_{-1,2}G_{-1,2}(-\mu_1)e^{k_2\tau_1} = (\text{RHS}$$

$$C_{1,1}G_{1,1}(+\mu_1)e^{-k_1\tau_1} + C_{-1,1}G_{-1,1}(+\mu_1)e^{k_1\tau_1} - C_{1,2}G_{1,2}(+\mu_1)e^{-k_2\tau_1} - C_{-1,2}G_{-1,2}(+\mu_1)e^{k_2\tau_1} = (\text{RHS}$$

$$r_{1,2}C_{1,2}G_{1,2}(+\mu_1)e^{-k_2\tau_2} + r_{-1,2}C_{-1,2}G_{-1,2}(+\mu_1)e^{k_2\tau_2} = (\text{RHS})_4$$

Using (20b) and (20c), we may write the left hand sides of the above equations in matrix form as

$$\begin{bmatrix} 1 & R_1 & 0 & 0 \\ e^{-k_1\tau_1} & R_1e^{k_1\tau_1} & -e^{-k_2\tau_1} & -R_2e^{k_2\tau_1} \\ R_1e^{-k_1\tau_1} & e^{k_1\tau_1} & -R_2e^{-k_2\tau_1} & -e^{k_2\tau_1} \\ 0 & 0 & R_2r_{1,2}e^{-k_2\tau_2} & r_{-1,2}e^{k_2\tau_2} \end{bmatrix} \begin{bmatrix} C_{1,1} \\ C_{-1,1} \\ C_{1,2} \\ C_{-1,2} \end{bmatrix}$$

Introducing the scaling transformation (48), we obtain

$$\begin{bmatrix} 1 & R_1e^{-k_1\tau_1} & 0 & 0 \\ e^{-k_1\tau_1} & R_1 & -1 & -R_2e^{-k_2(\tau_2-\tau_1)} \\ R_1e^{-k_1\tau_1} & 1 & -R_2 & -e^{-k_2(\tau_2-\tau_1)} \\ 0 & 0 & R_2r_{1,2}e^{-k_2(\tau_2-\tau_1)} & r_{-1,2} \end{bmatrix} \begin{bmatrix} \hat{C}_{1,1} \\ \hat{C}_{-1,1} \\ \hat{C}_{1,2} \\ \hat{C}_{-1,2} \end{bmatrix} \qquad (49)$$

Thus, we have removed all the positive exponentials as desired, and we observe that the resulting matrix is well-conditioned for arbitrary total optical depths and arbitrary individual layer

thicknesses. In the limit of large values of $k_1\tau_1$ and $k_2(\tau_2-\tau_1)$ the matrix in (49) reduces to

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & R_1 & -1 & 0 \\ 0 & 1 & -R_2 & 0 \\ 0 & 0 & 0 & r_{-1,2} \end{bmatrix}$$

which again shows that the ill-conditioning problem has been entirely eliminated.

2.11 Scaled Solutions

We must now introduce the scaling transformation into our solutions. Since only the homogeneous solution is affected, we concentrate on this part and start by introducing (48) into (44), i.e.,

$$I_p(\tau,\mu_i)=\sum_{j=1}^{N}\left\{\hat{C}_{jp}G_{jp}(\mu_i)e^{-k_{jp}(\tau-\tau_{p-1})}+\hat{C}_{-jp}G_{-jp}(\mu_i)e^{-k_{jp}(\tau_p-\tau)}\right\} \qquad (50)$$

Since $k_{jp}>0$ and $\tau_{p-1}\le\tau\le\tau_p$, all exponentials in (50) have negative arguments as they should.

Next we look at the homogeneous part of (35a) which we rewrite as (using $k_{-jp}=-k_{jp}$)

$$I_p(\tau,+\mu)=\sum_{n=p}^{L}\sum_{j=1}^{N}\left\{C_{-jn}\frac{G_{-jn}(+\mu)}{1-k_{jn}\mu}E_{-jn}(\tau,+\mu)+C_{+jn}\frac{G_{+jn}(+\mu)}{1+k_{jn}\mu}E_{+jn}(\tau,+\mu)\right\} \qquad (51)$$

Introducing (48) into (51), we find

$$I_p(\tau,+\mu)=\sum_{n=p}^{L}\sum_{j=1}^{N}\left\{\hat{C}_{-jn}\frac{G_{-jn}(+\mu)}{1-k_{jn}\mu}\hat{E}_{-jn}(\tau,+\mu)+\hat{C}_{+jn}\frac{G_{+jn}(+\mu)}{1+k_{jn}\mu}\hat{E}_{+jn}(\tau,+\mu)\right\} \qquad (52a)$$

where

$$\hat{E}_{-jn}(\tau,+\mu)\equiv E_{-jn}(\tau,+\mu)e^{-k_{jn}\tau_n}=\exp\left[-(k_{jn}\Delta\tau_n+\delta\,\tau/\mu)\right]-\exp\left[-(\tau_n-\tau)/\mu\right] \qquad (52b)$$

with

$$\begin{cases} \Delta\tau_n = \tau_n - \tau_{n-1}, & \delta\tau = \tau_{n-1} - \tau & \text{for } n > p \\ \Delta\tau_p = \tau_p - \tau, & \delta\tau = 0 & \text{for } n = p \end{cases}$$

$$\begin{aligned} \hat{E}_{+jn}(\tau,+\mu) &\equiv E_{+jn}(\tau,+\mu)e^{k_{jn}\tau_{n-1}} \\ &= \exp\left[-\frac{\tau_{n-1}-\tau}{\mu}\right] - \exp\left\{-k_{jn}(\tau_n - \tau_{n-1}) - \frac{\tau_n - \tau}{\mu}\right\} \end{aligned} \tag{52c}$$

for n>p and

$$\hat{E}_{+jp}(\tau,+\mu) = \exp\left[-k_{jp}(\tau - \tau_{p-1})\right] - \exp\left\{-k_{jp}(\tau_p - \tau_{p-1}) - \frac{\tau_p - \tau}{\mu}\right\} \tag{52d}$$

Since $k_{jn}>0$ for n=p+1, p+2,...,L and $\tau_L>...>\tau_{n=p+1}>\tau_{n-1=p}>\tau$ and also $k_{jp}>0$ and $\tau_{p-1}<\tau<\tau_p$, all the exponentials in (52b-d) have negative arguments as they should.

Similarly, by introducing (48) into the homogeneous part of (35b), we find

$$I_p(\tau,-\mu) = \sum_{n=1}^{p}\sum_{j=1}^{N}\left\{\hat{C}_{-jn}\frac{G_{-jn}(-\mu)}{1+k_{jn}\mu}\hat{E}_{-jn}(\tau,-\mu) + \hat{C}_{+jn}\frac{G_{+jn}(-\mu)}{1-k_{jn}\mu}\hat{E}_{+jn}(\tau,-\mu)\right\} \tag{53a}$$

where

$$\hat{E}_{+jn}(\tau,-\mu) \equiv E_{+jn}(\tau,-\mu)e^{k_{jn}\tau_{n-1}} = \exp\left[-k_{jn}\Delta\tau_n - \frac{\delta\tau}{\mu}\right] - \exp\left[-\frac{\tau-\tau_{n-1}}{\mu}\right] \tag{53b}$$

with

$$\begin{cases} \Delta\tau_n = \tau_n - \tau_{n-1}, & \delta\tau = \tau - \tau_n & \text{for } n < p \\ \Delta\tau_p = \tau - \tau_{p-1}, & \delta\tau = 0 & \text{for } n = p \end{cases}$$

$$\begin{aligned} \hat{E}_{-jn}(\tau,-\mu) &\equiv E_{-jn}(\tau,-\mu)e^{-k_{jn}\tau_n} \\ &= \exp\left[-\frac{\tau-\tau_n}{\mu}\right] - \exp\left\{-k_{jn}(\tau_n - \tau_{n-1}) - \frac{\tau-\tau_{n-1}}{\mu}\right\} \end{aligned} \tag{53c}$$

for $n<p$ and

$$\hat{E}_{-jp}(\tau,-\mu)=\exp\left[-k_{jp}(\tau_p-\tau)\right]-\exp\left\{-k_{jp}(\tau_p-\tau_{p-1})-\frac{\tau-\tau_{p-1}}{\mu}\right\} \qquad (53d)$$

Again, we see that all exponentials involved have negative arguments since $k_{jn}>0$ and $\tau>\tau_n>\tau_{n-1}$ for $n=1,2,...,p-1$, and also $k_{jp}>0$ and $\tau_{p-1}<\tau<\tau_p$. The fact that the exponentials involved in the scaled solutions all have negative arguments ensures that fatal overflow errors are avoided in the computations.

3. NUMERICAL IMPLEMENTATION

Most of the DISORT code was based on equations in papers published long before this Report was written. However, we have taken considerable pains to ensure that the formulas in the Report agree with the code. Sometimes simple rearranging of terms is needed to see this, but we do not consider this as disagreement since the coding of complex equations often requires breaking them up into elemental components anyway. We found only two instances of substantial disagreement, although they can always be brought to the same form after some manipulation:

(1) Subroutine SOLEIG uses C_{ij} from Eq. 5 of Stamnes and Swanson (1981). The equivalent quantity in this Report would be (cf. Eq 15b):

$$\mathbf{C}^{\pm} = \mathbf{D}^{\pm}\,\mathbf{W}$$

(2) In this Report, the eigenvectors are interpolated to user angles using (31a). However, in subroutine TERPEV this is done based on Stamnes and Dale (1981) Eq. 8.

DISORT has a rather large list of input variables. This list is more easily comprehended if several simple facts are borne in mind :

* there is one vertical coordinate, measured in optical depth units from the top down, and two angular coordinates—one polar, one azimuthal;

* polar (zenith) angles are measured from the upward direction: straight up is 0° and straight down is 180° (for historical reasons, the cosine of the incident beam angle is taken positive, whereas according to the DISORT convention all other downward-directed intensities have negative polar angle cosines);

* azimuth angles are measured in an absolute frame of reference, rather than from the plane of the incident beam; hence the azimuth angle of the incident beam is an input variable (although it is only used in (6));

* the layers and polar angles necessary for computational purposes are *entirely decoupled* from the levels and polar angles at which the user desires results.

This final point is so important that it is worth restating: *the radiant quantities can be returned to the user at <u>any</u> level and <u>any</u> angle*. For example, a user may have picked 3 computational layers and 16 computational polar angles, but they can then request intensities just 1/3 of the way

into the 2nd layer, and in just the zenith direction.

The computational layering is usually constrained by the problem, in the sense that each computational layer must be reasonably homogeneous and not have a temperature variation of more than about ~10K across it (if thermal sources are important). For example, a dusty boundary layer topped by a cloud topped by clear sky would suggest three computational layers, in the absence of thermal sources. The number of computational polar angles ("streams") is constrained by the need for accuracy; for example, 4 streams may be enough for accurate fluxes, while 8 to 16 streams may be necessary for accurate intensities.

The radiant output units are determined by the sources of radiation driving the problem. Lacking thermal emission, the radiant output units are the same as the units of the beam source and the isotropic source incident at the top boundary. The whole problem could then be non-dimensionalized by setting these to unity. If thermal emission of any kind is included, subprogram PLKAVG determines the units. Then the beam and isotropic sources must have the same units as PLKAVG, which returns MKS values (W/m^2). Several users have rewritten PLKAVG to return just the temperature, an approximation which is widely used in the long-wavelength limit of microwave radiative transfer; in this case, all radiant quantities are in degrees Kelvin.

3.1 Structure of the FORTRAN Program

A flow chart of the DISORT algorithm (DIScrete Ordinate Radiative Transfer) is provided in Figure 2a. The philosophy behind the corresponding FORTRAN implementation, diagramed in Figure 2b, has been one of extensive modularization, with many individual subroutines each of which focuses on a relatively circumscribed task. These subroutines are designed to be self-contained, well-documented and readable. Note in Fig. 2b the highly downward, sequential flow of the computation, with little side branching.

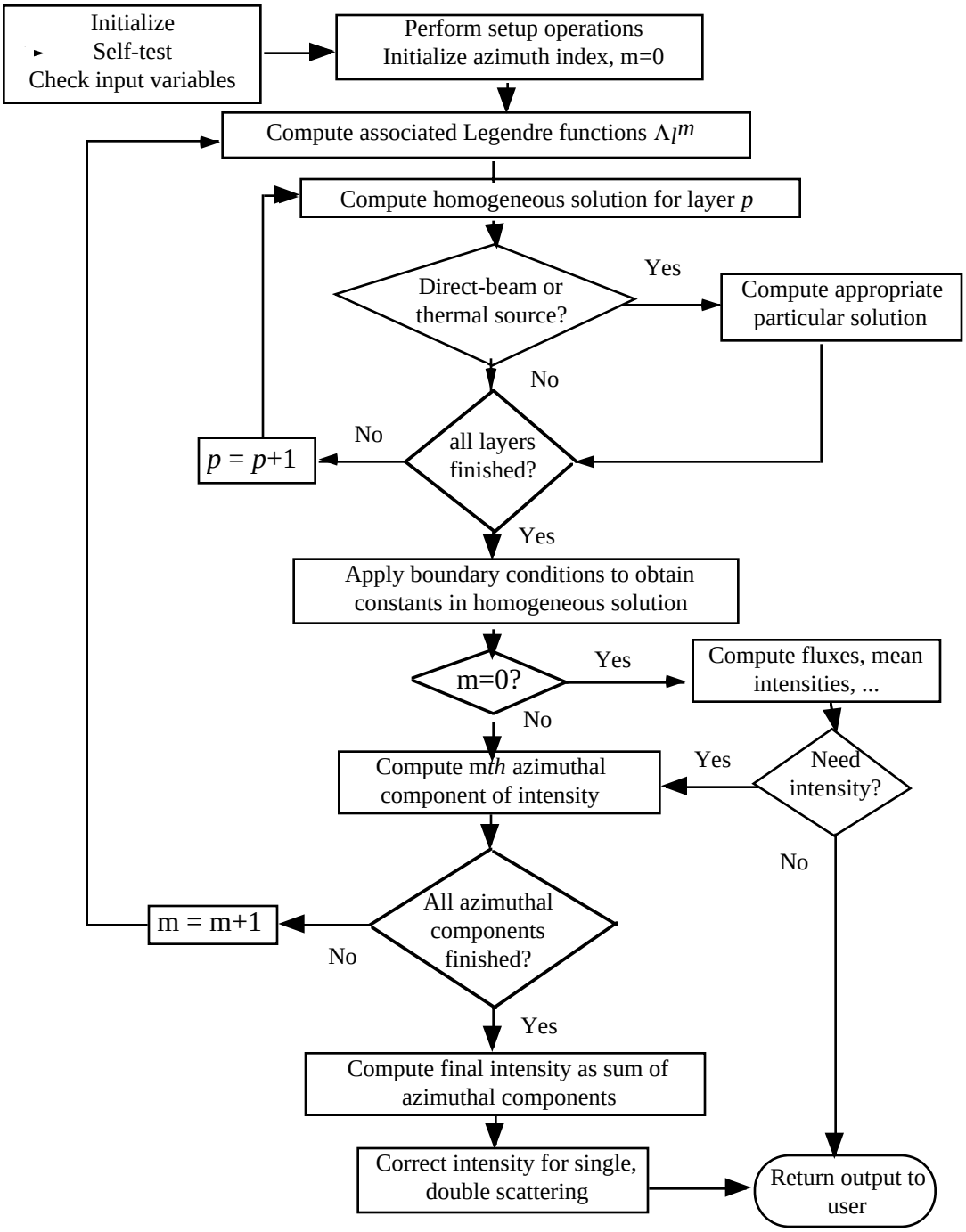

Figure 2a. Flow chart of the DISORT computer code.

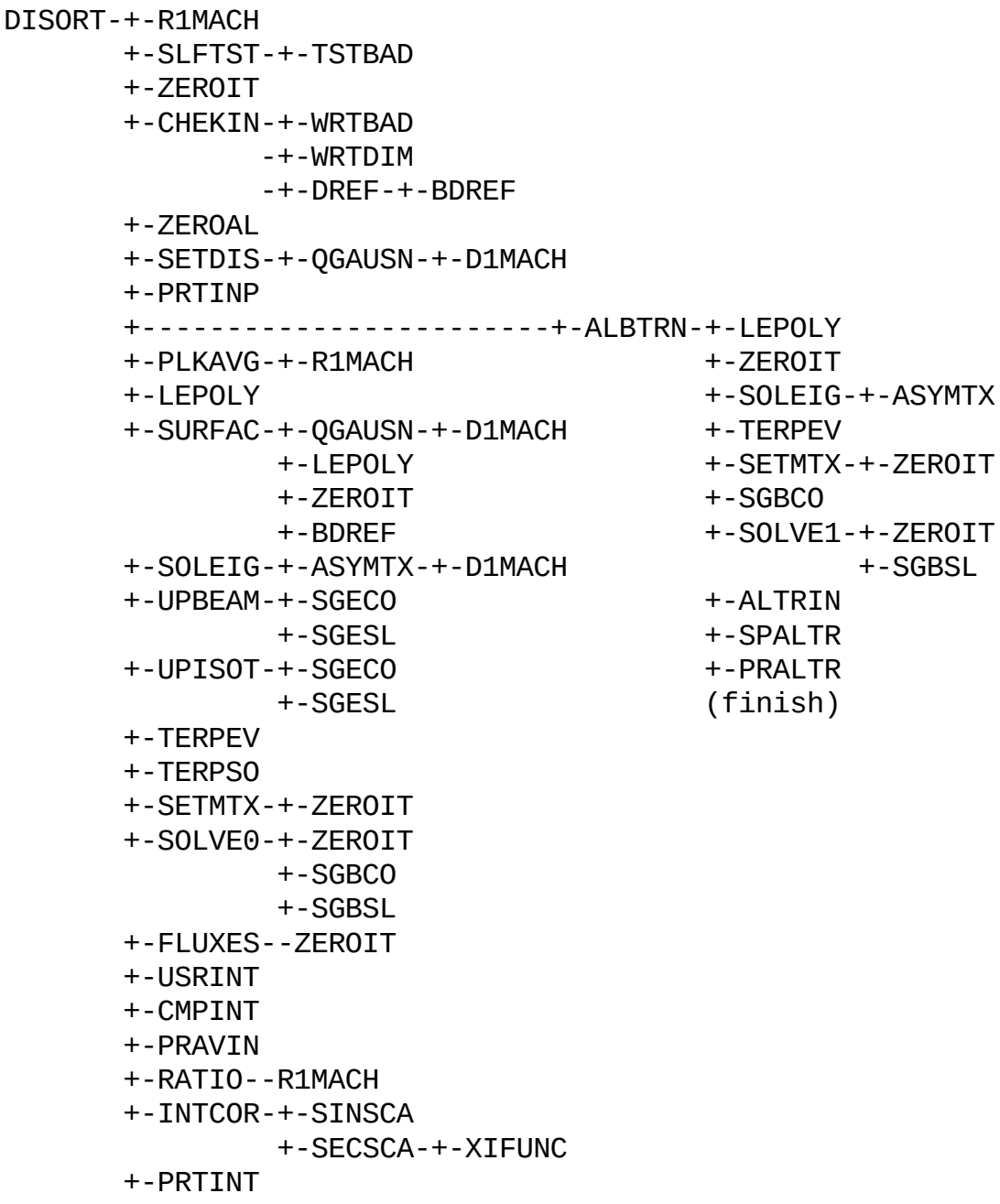

Figure 2b. Wiring diagram (call tree) of the DISORT computer code. The ubiquitous calls to ERRMSG are omitted. SG... are LAPACK routines.

The subroutines in the DISORT program can be grouped by function, as follows:

User-Supplied

BDREF	user-supplied FUNCTION subprogram for bottom-boundary bidirectional reflectivity (stub version which returns zero included, and special version provided with test problems)

Setup and Input

CHEKIN	checks input variables for errors
DREF	flux albedo as a function of incident angle when a bottom-boundary bidirectional reflectivity is specified
SLFTST	sets input for self-test and checks for failure
PLKAVG	computes integral of Planck function over a wavenumber interval
TSTBAD	prints message when self-test fails
WRTBAD	writes out names of erroneous input variables
WRTDIM	writes out names of too-small symbolic dimensions
ZEROAL	zeros a group of matrices

Core Calculation

ASYMTX	solves eigenfunction problem for real asymmetric matrix whose eigenvalues are known to be real
CMPINT	computes intensities at user levels and computational angles
FLUXES	computes upward and downward fluxes, flux divergences, and mean intensities
INTCOR	corrects intensity field by using Nakajima-Tanaka algorithm
SECSCA	calculates secondary scattered intensity for intensity correction
SINSCA	calculates single-scattered intensity for intensity correction
SETMTX	calculates coefficient matrix for linear equations embodying boundary and layer interface conditions
SOLVE0	solves linear equations embodying boundary and layer interface conditions (general boundary conditions)
SOLEIG	solves eigenfunction problem for a single layer
SURFAC	calculates surface bidirectional reflectivity and emissivity in form needed by rest of program
TERPSO	interpolates particular solutions to user angles

TERPEV	interpolates eigenvectors to user angles
UPBEAM	finds particular solution for beam source
UPISOT	finds particular solution for thermal emission source
USRINT	computes intensities at user levels and user angles
XIFUNC	calculates Xi function used in SECSCA (Eq A.17)

Service

LEPOLY	evaluates normalized associated Legendre polynomials
PRALTR	prints albedo and transmissivity in ALBTRN special case
PRAVIN	prints azimuthally averaged intensities (not executed unless an internal flag is turned on by altering the code)
PRTINP	prints input variables
PRTINT	prints intensities at user angles
QGAUSN	computes Gaussian quadrature points and weights
RATIO	computes ratio of two numbers in failsafe way
R1MACH	returns machine constants
SG**...**	linear-equation solvers from LINPACK
ZEROIT	zeros a given matrix

Special Case

ALBTRN	manages the special case described in Sec. 3.8 where the user wants only the flux albedo and transmission of the entire medium for many incident beam angles at once
ALTRIN	calculates azimuthally-averaged intensity (equal to albedo and/or transmissivity) at user angles
SOLVE1	solves linear equations embodying boundary and layer interface conditions
SPALTR	calculates spherical albedo, transmission

We shall briefly walk through the flow chart and concentrate on some important concepts of the numerical implementation. The user supplies all input to subroutine DISORT through an argument list. The most important input variables are:

(1) the number of computational angles (streams) and the actual angles at which output is desired (which need bear no relation to the computational angles);

(2) the number of computational layers to be employed and their optical properties (optical thickness, single scattering albedo, and phase function) and temperature; and the actual levels at which output is desired (which need bear no relation to the computational levels);

(3) the bidirectional reflectance or albedo of the surface, and its temperature.

The first time subroutine DISORT is called, it performs a self-test (SLFTST) by executing a simple test problem and making sure the answer is correct. This provides a modicum of security against the introduction of bugs, although this recursive construction may interfere with parallelization. If the self-test is passed, DISORT performs several setup operations:

(a) zeroing arrays (ZEROAL),

(b) checking input variables (CHEKIN),

(c) printing input variables (PRTINP),

(d) δ–M transformation for each computational layer, and a numerical shortcut to treat cases with large absorption optical depth (SETDIS),

(e) computation of thermal sources (PLKAVG).

As shown in Figure 2a, DISORT has two main loops. The outer loop is over azimuthal sum index m (6) while the inner loop is over layer index p. Prior to entering the inner loop, subroutine LEPOLY computes the normalized associated Legendre polynomial (5d), $\Lambda^l_m(\mu)$, of order m and all degrees l, at all quadrature and output polar angle cosines μ. These polynomials occur in (8b-d). The LEPOLY algorithm is described in Section 3.3, together with the computation of quadrature angles and weights (QGAUSN). The expression for D^m defined in (8b) is subsequently used in (10b), (24b), (26), (29) and (31).

In the inner loop over layers, the homogeneous solution for each layer is obtained in SOLEIG. The eigenvalues and eigenvectors of the $2Nx2N$ matrix equation (23b) are computed by solving the reduced NxN matrix equation (23f) utilizing our customized QR algorithm (ASYMTX). In Section 3.4, we discuss some of the numerical aspects of the eigenvalue problem. (It is well to remember that discrete ordinates was written off as a useless method until sophisticated algorithms for the eigenvalue problem became available, so that is why we call this “the core of DISORT”.)

Particular solutions are then computed for every layer for solar pseudo-sources (UPBEAM) and for thermal emission (UPISOT) allowing for linear-in-optical-depth variation of the Planck function within each layer.

When the loop over layers is completed, the boundary and interface continuity conditions are applied in SETMTX and SOLVE0 using (46–47). In order to avoid ill-conditioning when solving for the constants of integration, a scaling transformation is required. As discussed in Section 3.5, this scaling transformation guarantees unconditionally stable solutions for arbitrary individual layer and total optical thickness.

The first time through the outer loop (azimuthal index $m=0$) we compute fluxes, flux divergences, mean intensities, and azimuth-averaged intensities using (9) in FLUXES. If only these latter quantities are desired by the user, this completes the computation; otherwise DISORT loops over azimuthal index m, calling USRINT repeatedly to get the azimuthal components of the intensity I^m at user-specified polar angles. To obtain the intensity at arbitrary (user-specified) azimuth angles, DISORT sums (6) over all I^m, stopping when the criterion in Sec. 3.7 is satisfied.

Finally, the Nakajima-Tanaka (1988) algorithms highlighted in Section 3.6 are applied at the end of DISORT to correct the intensity calculation in the presence of strongly forward-peaked scattering. Such corrections are essential if one desires to use low-order discrete ordinate approximations to reduce the computational burden. These algorithms used in conjunction with the δ–M method may provide acceptable accuracy for as few as 8 streams. Without these algorithms similar accuracy will typically require a quadrupling of the number of streams, implying that they provide computational savings of the order of $4^3 = 64$.

Simple expressions for the plane albedo and transmissivity are presented in Section 3.8 for media without thermal sources. These expressions, implemented in ALBTRN, offer substantial computational advantages when flux albedo and transmissivity are required at many incident beam angles. We conclude with some remarks about computational speed in Section 3.9 and computer precision in Section 3.10.

3.2 Setup Operations

3.2.1 Constraints on input/output variables

Subroutine CHEKIN checks the input variables and array-dimension specifications; it mainly

performs sanity and max-min checking, plus imposing a few physical constraints. For instance, it sanity-tests whether any layer optical depth is negative; and max-min-tests whether each layer single scattering albedo is between zero and unity. No such checking can be perfect — subtle combinations of bad input are always possible — but the paucity of user complaints about, or suggested additions to, CHEKIN indicate that its simple checks catch most user input errors.

Sanity and max-min checking of DISORT output variables would also be desirable; however, this is not done at present, partly because output intensities and fluxes can both be negative under certain circumstances; these negative values are negligible compared to the driving sources, but would require incisive tests based on machine precision. While testing output variables may seem superfluous, it provides a final sentinel that code changes by the user or very unusual input do not trigger bad behavior in DISORT.

Subroutine CHEKIN prints an error message containing the name of the variable in error, then continues until all variables have been checked. At the end of CHEKIN, if the error count is one or more, execution is terminated; while some of these errors may be mild, the DISORT philosophy is that even small errors may conceal bigger problems, or misconceptions on the part of the user. In addition, a warning message is issued if the temperature change across any computational layer exceeds 10K, but execution is allowed to proceed because this does not always lead to inaccurate results.

CHEKIN also checks if the dimensions of internal arrays (contained in PARAMETER statements in subroutine DISORT) are too small. The resultant error message will give the name of the PARAMETER which needs to be increased. This will become obsolete when DISORT is converted to Fortran-90; Fortran-90's dynamic memory allocation features allow internal arrays to assume exactly the dimensions needed, neither too large nor too small.

3.2.2 δ–M transformation

DISORT uses the δ–M transformation (Wiscombe, 1977) to achieve optimum computational efficiency and accuracy for strongly forward-peaked phase functions. The essence of the δ–M method is to separate the phase function P into the sum of a delta-function in the forward direction and a truncated phase function P' which we expand in a (presumably short) series of Legendre polynomials:

$$P(\tau,\cos\Theta) = f\,P''(\tau,\cos\Theta) + (1-f)P'(\tau,\cos\Theta)$$
$$\approx 2f\,\delta(1-\cos\Theta) + (1-f)\sum_{\ell=0}^{2N-1}(2\ell+1)g'_\ell(\tau)P_\ell(\cos\Theta) \qquad (54a)$$

(The term "truncation" is frequently used to describe the separation of the forward peak, but we feel this usage is confusing since "truncation" has a well-established meaning in mathematics; in the DISORT context, "truncation" will refer exclusively to stopping finite or infinite series before their natural ending point.)

The use of N for the upper limit of the sum in (54a) is not accidental. It is meant to indicate that our goal is to reduce the length of the phase function expansion (5a) from $2M$, which can be 100's to 1000's, to $2N$, where $2N$ is the number of computational quadrature angles and might be as little as 4 to 8.

Note that the actual forward-peak phase function P'' is approximated by a delta-function, but once the expansion coefficients g_l' are known it could be recaptured using the first part of (54a). The δ–M Legendre expansion coefficients g_l' are obtained from the Legendre coefficients g_l of the phase function P by

$$g'_\ell(\tau) = \frac{g_\ell(\tau) - f}{1-f} \quad (\ell = 0,\ldots,2N-1) \qquad (54b)$$

and the separated fraction f is chosen by setting

$$f = g_{2N}(\tau) \qquad (54c)$$

One way to see why δ–M is an improvement is as follows. $g_l=0$ for $l\geq 2N$ in the ordinary phase function expansion in Legendre polynomials. In δ–M, $g'_l=0$ for $l\geq 2N$, which by (54b,c) is equivalent to $g_l=g_{2N}$ for $l\geq 2N$, and this latter approximation is bound to be better than $g_l=0$ since typically the g_l decrease only slowly with l.

If (54a-c) are substituted into the radiative transfer equation (1), an identical form of the transfer equation is obtained, but with P', τ', ω' replacing P, τ, ω, respectively, where

$$d\tau' = (1-\omega f)\, d\tau$$
$$\omega' = \frac{1-f}{1-\omega f}\,\omega \qquad (54d)$$

The δ–M transformation (54b–d) is applied in SETDIS. SETDIS also multiplies the phase function Legendre coefficients g_l by (2l+1) and by the (scaled) single scatter albedo, since they always occur in this combination.

Application of the δ–M method will artificially enhance the direct-beam component of flux at the expense of the diffuse component, but the sum of the two will be computed accurately. The true direct-beam component is of course trivially calculated as an exponential attenuation of the incident direct beam using the unscaled optical depth. Unlike the δ–M direct beam, this true direct-beam does not depend on the number of streams chosen, which makes physical sense. Thus, DISORT returns the true direct-beam. To preserve the sum of direct and diffuse, DISORT must therefore "unscale" the returned diffuse flux, essentially adding back that part which was lumped into the δ–M direct beam. This unscaling of the diffuse flux is done in FLUXES.

The limitations of the δ–M transformation for calculating intensities are discussed in Section 3.6, where a method for improving δ–M based on treating single- and double-scattering more correctly is described.

3.2.3 Integrated Planck function

PLKAVG calculates the integral of the Planck blackbody radiation function across a spectral interval. This is somewhat of an inconsistency since DISORT is strictly monochromatic, but is furnished as a service for problems where the only thing varying significantly across a spectral interval is the Planck function. The DISORT spectral unit is wavenumber ν=1/λ (the standard variable in spectroscopy), where λ is wavelength. We define the "Planck function" at temperature T [K] and wavenumber ν [cm^{-1}] as the emitted power per unit area from a blackbody surface (not the energy density inside a blackbody, which differs by a factor c/4):

$$B[\upsilon, T] = \frac{10^8 c_1 \upsilon^3 / \pi}{e^{100 c_2 \upsilon / T} - 1} \quad [\mathrm{W/m^2}] \qquad (55a)$$

where the 10^8 factor accounts for the units of ν being [cm^{-1}] not [m^{-1}], and

$$c_1 = 2\pi hc^2 = 3.741832\times10^{-16}\ [\mathrm{W/m^2}]$$
$$c_2 = hc/k = 1.438786\times10^{-2}\ [\mathrm{m\text{-}K}] \qquad (55b)$$

are the so-called "first and second radiation constants" with h = Planck's constant = 6.626176×10^{-34} J-s, c = speed of light = 2.99792458×10^{8} m/s, and k = Boltzmann's constant = 1.380662×10^{-23} J/K (values from CRC Handbook of Chemistry and Physics).

Note that an extra factor of π is divided out of (55a), compared to the normal definition one finds in textbooks. This is because we want (55a) to be the intensity leaving a blackbody surface, not the flux; since the blackbody intensity is isotropic, this merely means dividing the flux by π, which can be seen by pulling the intensity in (9a) outside the integral signs and doing the integrals.

Integrating (55a) over all wavenumbers leads to the famous Stefan-Boltzmann T^4 law:

$$\int_0^\infty B[\upsilon,T]\,d\upsilon = \frac{\sigma T^4}{\pi} \qquad (55c)$$

where σ is the Stefan-Boltzmann constant:

$$\sigma = \frac{2\pi^5}{15}\frac{k^4}{c^2h^3} = 5.67032\mathrm{e}-8 \quad [\mathrm{W m^{-2} K^{-4}}]$$

In integrating the radiative transfer equation across a spectral interval $[\nu_1,\nu_2]$, denoted by $\Delta\nu$, we encounter integrals of the form

$$\int_{\Delta\upsilon} B[\upsilon,T]\,\Psi_\upsilon\,\mathrm{d}\upsilon$$

where Ψ is the product of the intensity and possibly other wavenumber-dependent factors. The DISORT procedure, which conserves energy and is rigorously correct in the limit of zero or infinite absorption, is to approximate such integrals as

$$\int_{\Delta\upsilon} B[\upsilon,T]\,\Psi_\upsilon\,\mathrm{d}\upsilon = \frac{\xi_{\Delta\upsilon}}{\Delta\upsilon}\int_{\Delta\upsilon}\Psi_\upsilon\,\mathrm{d}\upsilon \qquad (55d)$$

where

$$\xi_{\Delta\upsilon} = \int_{\upsilon_1}^{\upsilon_2} B[\upsilon', T]\,\mathrm{d}\upsilon' \equiv \xi_{\upsilon_2} - \xi_{\upsilon_1} \quad \text{where} \quad \xi_{\upsilon} \equiv \int_0^{\upsilon} B[\upsilon', T]\,\mathrm{d}\upsilon' \tag{55e}$$

(The usual Mean-Value-Theorem procedure — pulling B through the integral sign in (55d) and evaluating it at some wavenumber inside $\Delta\nu$ — is to be avoided; it neither conserves energy nor gives the correct answer in limiting cases such as a vacuum above a black surface.)

If the spectral interval is very narrow (ν_1 and ν_2 are close), a direct Simpson's-Rule quadrature is used to calculate $\xi_{\Delta\nu}$ to avoid an ill-conditioned subtraction in (55e). The Simpson Rule is iterated until a goal of 6-significant-digit accuracy is achieved. But in the general case the subtraction formula in (55e) is used, and thus approximations for the integral ξ_{ν} are required. (That integral cannot be done analytically; even version 4.0 of Mathematica fails to find a closed form for it.) We begin by defining a non-dimensional version E of ξ_{ν} by making the change of variable

$$t = 100\, c_2 \upsilon' / T$$

to obtain

$$\xi_{\upsilon} = \frac{c_1 T^4}{\pi {c_2}^4} \int_0^{100\, c_2 \upsilon / T} \frac{t^3}{e^t - 1}\,\mathrm{d}t \equiv \frac{\sigma T^4}{\pi} E\left(\frac{100\, c_2 \upsilon}{T}\right) \tag{55f}$$

where

$$E(x) \equiv \frac{15}{\pi^4} \int_0^x \frac{t^3}{e^t - 1}\,\mathrm{d}t, \quad E(\infty) = 1 \tag{55g}$$

It is also useful to define

$$H(x) \equiv 1 - E(x) = \frac{15}{\pi^4} \int_x^{\infty} \frac{t^3}{e^t - 1}\,\mathrm{d}t, \quad H(0) = 1 \tag{55h}$$

For x large, e^t in the integrand in (55h) will be large, and thus it makes sense to expand in inverse powers of it:

$$\left(e^t - 1\right)^{-1} = e^{-t}\left(1 + e^{-t} + e^{-2t} + \ldots\right)$$

and integrate term-by-term to obtain

$$H(x)=\frac{15}{\pi^4}\sum_{n=1}^{\infty}\frac{e^{-nx}}{n^4}\Big(6+nx\big(6+nx\,(3+nx)\big)\Big) \tag{55i}$$

We have found that this infinite series gives 6 significant digits of accuracy if we cut it off at the following numbers n_{max} of terms:

x-range	nmax
1.5 – 1.9	7
1.9 – 2.3	6
2.3 – 2.9	5
2.9 – 3.9	4
3.9 – 5.7	3
5.7 –	2
10.25 – ∞	1

For x small, the integrand of $E(x)$ in (55g) can be expanded in powers of t and a term-by-term integration performed, to yield

$$E(x)=\frac{15x^3}{\pi^4}\left(\frac{1}{3}-\frac{x}{8}+\frac{x^2}{60}-\frac{x^4}{5040}+\frac{x^6}{272160}-\frac{x^8}{13305600}+O(x^{10})\right) \tag{55j}$$

This power series gives 6 significant digits of accuracy when $x\le 1.5$.

Thus, we get $E(x)$ directly from (55j) when $x\le 1.5$, or else indirectly from (55i), as $1-H(x)$, when $x>1.5$, using the number of terms in the above table. Then PLKAVG implements (55e) as follows:

(1) If $x_1 \le x_2 \le 1.5$, where $x_i = 100\ c_2\ \nu_i\ /T$, then

$$\xi_{\Delta\upsilon}=\frac{\sigma T^4}{\pi}\Big\{E(x_2)-E(x_1)\Big\} \tag{56a}$$

(2) If $x_1 \le 1.5 < x_2$, then since

$$\int_{x_1}^{x_2} = \int_{x_1}^{0} + \int_{0}^{\infty} + \int_{\infty}^{x_2} = \int_{0}^{\infty} - \int_{0}^{x_1} - \int_{x_2}^{\infty}$$

we take

$$\xi_{\Delta\upsilon} = \frac{\sigma T^4}{\pi}\left\{1 - E(x_1) - H(x_2)\right\} \tag{56b}$$

(3) If $1.5 < x_1 \le x_2$, then

$$\xi_{\Delta\upsilon} = \frac{\sigma T^4}{\pi}\left\{H(x_1) - H(x_2)\right\} \tag{56c}$$

We might still worry about the subtractions in (56a–c) causing ill-conditioning. But since x_1 close to x_2 is treated as a special case using a Simpson Rule quadrature, and since E and H are monotonic by virtue of having positive integrands, this is not a concern for (56a,c). And because of the ranges of x_1 and x_2 in (56b), neither $E(x_1)$ nor $H(x_2)$ ever come close to unity [E(1.5)=0.095], so (56b) suffers no ill-conditioning either.

3.2.4 Computational shortcuts

For highly absorbing media without thermal emission, very little radiation will reach levels for which the absorbing optical depth is greater than, say 10. In such cases we may ignore anything happening below this depth (i.e. set all the radiant quantities to zero), redefine the "surface" to occur at this depth, and set the surface reflection to zero. SETDIS determines where this depth occurs, and the rest of the program applies this knowledge.

Since the computational time is proportional to the number of computational layers picked by the user, this may result in considerable computational savings for wavelengths at which the absorption optical depth is large. The savings would be especially noticeable when DISORT is used inside a line-by-line model with the atmosphere having large absorption optical depths near line centers.

3.3 Angle-Related Computations

3.3.1 Quadrature weights and abscissae

The quadrature weights and abscissae are calculated by QGAUSN. It is vital that this

subroutine be run in the highest possible precision — at least IEEE double precision, or 14 significant digits. This helps to conserve energy in DISORT. Negligible computer time penalty is incurred, because QGAUSN never takes more than a few percent of DISORT's computing time. For many years, DISORT was distributed in two Fortran-77 versions: one where ASYMTX and QGAUSN were the only routines in double precision; and one where all routines were single precision but with a warning to use an autodoubling compiler (that converts everything to double precision) if the underlying single precision was not at least 14-digits. (Cray and DEC Alpha always had 14-digit single precision.)

In DISORT, a Double-Gauss quadrature rule is utilized to transform the analytically insoluble integro-differential equation of radiative transfer (1) into a finite set of coupled ordinary differential equations for which analytic solutions are feasible. This rule requires N weights w_i and N abscissae μ_i for quadrature on the interval [0,1] corresponding to the interval [0°,90°] in polar angle θ, such that the integral of an arbitrary function $f(\mu)$ from 0 to 1 can be approximated as:

$$\int_0^1 f(\mu)\, d\mu \cong \sum_{i=1}^{N} w_i\, f(\mu_i)$$

By the definition of Gaussian quadrature, this is exact when $f(\mu)$ is a polynomial of degree $2N-1$ or less. This fact was used to test DISORT's subroutine QGAUSN, a fast yet extremely accurate routine for computing these weights and abscissae.

The QGAUSN algorithm, described in the classic book by Davis and Rabinowitz (1975), is as follows. The abscissae μ_i for [–1,1] are roots of the Legendre polynomial $P_N(x)$. They are calculated iteratively using a cubically convergent variant of Newton's Method:

$$\mu_i^{(k+1)} = \mu_i^{(k)} - \frac{P_N}{P_N'}\left(1 + \frac{P_N P_N''}{2(P_N')^2}\right)$$

where k is the iteration number, and the omitted argument of P_N and its derivatives is just the current iterate, $\mu_i^{(k)}$. The iteration is continued until two successive iterates are within ten times machine precision of each other. Values of P_N are calculated using upward recurrence starting from P_0=1 and $P_1(\mu)$=μ,

$$P_n = \frac{1}{n}\left((2n-1)\,\mu_i^{(k)}\,P_{n-1} - (n-1)\,P_{n-2}\right)$$

which also provides P_{N-1} to calculate the derivatives of P_N needed for the Newton iteration:

$$P'_N = \frac{N\left(P_{N-1} - \mu_i^{(k)}\,P_N\right)}{1-\left[\mu_i^{(k)}\right]^2}$$

$$P''_N = \frac{2\,\mu_i^{(k)}P'_N - N(N+1)\,P_N}{1-\left[\mu_i^{(k)}\right]^2}$$

Since Newton's method can easily diverge, a very good initial guess is essential; fortunately Davis/Rabinowitz furnish one:

$$\mu_i^{(0)} = \cos\left(t + \frac{N-1}{8N^3\tan t}\right) \quad \text{where} \quad t \equiv \frac{4i-1}{4N+2}\pi$$

With this starting value and the cubically-convergent Newton Method, it requires only one iteration to compute abscissae correct to 7 figures and two iterations for 14 figures. In our experience, this scheme has never led to divergence even for *N* up to 1000.

The abscissae for [–1,1] are mirror symmetric about zero. Thus the above iteration finishes after finding just *N*/2 abscissae; the other *N*/2 are obtained by symmetry.

Knowing the abscissae μ_i, the weights are obtained from:

$$w_i = 2\,\frac{1-\mu_i^{\,2}}{\left(N\,P_{N-1}\right)^2}$$

The weights for two mirror-symmetric abscissae are identical.

For *N* odd, analytic expressions for the middle (M) abscissa and weight are used:

$$\mu_M = 0, \quad w_M = 2 \Bigg/ \left(\prod_{\substack{k=3 \\ k\ odd}}^{N} \frac{k}{k-1} \right)^2$$

Since the resulting weights and abscissae are for the interval [–1,1], they are transformed to the interval [0,1] at the end (see Thomas and Stamnes, 1999, for details about this transformation):

$$\mu_i \longrightarrow \tfrac{1}{2}\left(1+\mu_i\right)$$

$$w_i \longrightarrow \tfrac{1}{2} w_i$$

3.3.2 Associated Legendre polynomials

DISORT computes the *normalized* associated Legendre polynomials Λ_l^m (5d) in LEPOLY. The Λ_l^m are preferred over the usual associated Legendre polynomials P_l^m because the Λ_l^m remain bounded whereas the P_l^m can become quite large and overflow the computer; as a side benefit, use of Λ_l^m somewhat simplifies the formulas. We follow the Λ_l^m algorithm of Dave and Armstrong (1970). However, we carried their derivations farther in some respects.

Which Λ_l^m are actually needed in DISORT? The only places they occur are (8b,d), which need only

$$\Lambda_\ell^m(\mu) \quad \text{for} \quad \ell = m, \ldots, 2N-1 \tag{57}$$

Since the intensity component calculation for each m occurs in order of increasing m, starting at m=0, and is independent of the calculation for every other m, there is no need to calculate a triangular matrix of values of Λ_l^m for all possible l and m. This would also not be possible in DISORT because exit from the loop over m is controlled by a convergence criterion; thus the upper limit for m is not known in advance and is often much lower than the theoretical upper limit. DISORT just calculates the vector (57) as required, for each m in turn, using certain values from the previous value of m for initialization of recurrences. This accounts for the somewhat unusual structure of LEPOLY whereby Λ_l^m is both an input (when m>0) and an output variable.

The only *stable* recurrence for the $\Lambda_l{}^m$ is the one involving its subscript, which is fortunate since that is exactly what is required in (57). We take this recurrence from Dave/Armstrong Eq. (10) but substitute the more customary notations $l=k+1$ and $m=n-1$ to get:

$$\Lambda_\ell^m(\mu) = \frac{(2\ell-1)\mu\,\Lambda_{\ell-1}^m(\mu) - \sqrt{(\ell+m-1)(\ell-m-1)}\,\Lambda_{\ell-2}^m(\mu)}{\sqrt{(\ell-m)(\ell+m)}} \quad (\ell = m+2,\ldots) \qquad (58a)$$

To initialize the recurrence (58a), we need $\Lambda^m{}_m$ and $\Lambda^m{}_{m+1}$; Dave and Armstrong express these two quantities as multi-step recurrences, but we have derived simpler expressions requiring only a single recursive step. First we deal with $\Lambda^m{}_m$. Putting $m=n-1$ in Dave/Armstrong Eq. (11–12) leads to

$$\Lambda_m^m(\mu) \equiv (1-\mu^2)^{m/2}\, y_m \qquad (58b)$$

where

$$y_m = -\sqrt{\frac{2m-1}{2m}}\, y_{m-1} \qquad (58c)$$

Multiplying both sides of (58c) by $(1-\mu^2)^{m/2}$ and using (58b) gives

$$\Lambda_m^m(\mu) = -\sqrt{\frac{(2m-1)(1-\mu^2)}{2m}}\, \Lambda_{m-1}^{m-1}(\mu) \qquad (58d)$$

Thus, starting with $\Lambda^0{}_0=1$, (58d) gives all $\Lambda^m{}_m$.

Next we find $\Lambda^m{}_{m+1}$. Putting $m=n-1$ in Dave/Armstrong Eq. (13-14) leads to

$$\Lambda_{m+1}^m(\mu) \equiv \mu(1-\mu^2)^{m/2}\, z_{m+1} \qquad (58e)$$

where

$$z_{m+1} = -\sqrt{\frac{2m+1}{2m}}\; z_m \qquad (58f)$$

Dividing (58e) by (58b) leads to

$$\frac{\Lambda^m_{m+1}}{\Lambda^m_m} = \mu \frac{z_{m+1}}{y_m} \tag{58g}$$

The ratio on the right-hand side of (58g) is much simpler than that provided by Dave/Armstrong. Using the recurrences (58c,f) for *y* and *z* leads to:

$$\frac{z_{m+1}}{y_m} = \sqrt{\frac{2m+1}{2m-1}}\,\frac{z_m}{y_{m-1}} = \sqrt{\frac{2m+1}{2m-3}}\,\frac{z_{m-1}}{y_{m-2}} = \ldots = \sqrt{2m+1}\,\frac{z_1}{y_0} = \sqrt{2m+1}$$

so that (58g) reduces to

$$\Lambda^m_{m+1}(\mu) = \mu \sqrt{2m+1}\,\Lambda^m_m(\mu) \tag{58h}$$

Thus, knowing only Λ^{m-1}_{m-1}, which comes into LEPOLY as input from subroutine DISORT, we can get Λ^m_m and Λ^m_{m+1} from (58d,h), then $\Lambda_l{}^m$ for $l=m+2,\ldots,N-1$ from the general recurrence (58a). (When m=0 we initialize the process by setting $\Lambda^0_0 = 1$ rather than expecting input from subroutine DISORT.)

There are two other computational details. First, because the square roots of integers appear so often in the recurrence formulas, LEPOLY saves the square roots of the integers 1 to 1000 on its first execution, then uses them on subsequent executions to obviate calls to the SQRT function (this might need rethinking for parallel computation). Second, back in subroutine DISORT, advantage is taken of the relation

$$\Lambda^m_\ell(-\mu) = (-1)^{\ell-m}\,\Lambda^m_\ell(\mu) \tag{59}$$

to bypass LEPOLY calls for the negative computational μ's, since these are just negatives of the positive computational μ's (the Gaussian abscissae on [0,1] from Section 3.3.1).

3.4 Computation of Eigenvalues and Eigenvectors

The eigen-computation is handled entirely by ASYMTX. It is vital that this subroutine be run in the highest possible precision — at least IEEE double precision, or 14 significant digits. This is because the eigenproblem becomes degenerate as single-scatter albedo approaches unity, and in general because it is a sensitive computation requiring high precision. For many years, DISORT was distributed in two Fortran-77 versions: one where ASYMTX and QGAUSN were the only

routines in double precision; and one where all routines were single precision but with a warning to use an autodoubling compiler (that converts everything to double precision) if the underlying single precision was not at least 14-digits.

Stamnes and Swanson (1981) obtained eigenvalues and eigenvectors by solving (23f) and showed that only real eigenvalues were obtained in their numerical calculations. Since the matrix in (23f)

$$\mathbf{A} = (\boldsymbol{\alpha} - \boldsymbol{\beta})\,(\boldsymbol{\alpha} + \boldsymbol{\beta})$$

is real but asymmetric, they adopted subroutines in EISPACK (cf. Cowell, 1980) which utilize the double-QR algorithm. This algorithm applies to a general real matrix (which typically has complex eigenvalues/eigenvectors) and thus must use complex arithmetic. This wastes a large amount of computation since only real arithmetic is needed for the discrete-ordinates application. Therefore co-author Tsay developed ASYMTX as a customized version of the EISPACK solver using only real arithmetic. ASYMTX also collapsed the entire EISPACK algorithm, which was scattered across many subroutines, into a single subroutine. ASYMTX is really the heart of DISORT, and accounts for the program's relative compactness.

We are aware that LAPACK, the successor to LINPACK and EISPACK, includes improved eigenproblem algorithms. However, the appropriate subroutine SGEEV (Single-precision GEneral EigenValue) requires 29 dependent routines containing over 8000 lines not including the supplied machine constant routines (which are unnecessary in Fortran-90). We found this somewhat offputting for several reasons:

(1) it is not clear if the new LAPACK routines are really needed — the improvement in accuracy might be subtle at best, since EISPACK was already pretty good;

(2) it is not obvious that the new routines, being so much larger and requiring so much argument-passing, would be faster computationally;

(3) it would require extensive testing to see what differences are caused by the new routines;

(4) an aesthetic point — since all of DISORT exclusive of ASYMTX is only about 6000 lines, we felt a certain lack of proportion if we adopted the new routines, especially since ASYMTX is only about 700 lines.

Of course, there are also some advantages to switching to the LAPACK eigensolver. First, the new algorithms are undoubtedly more accurate and more adapted to modern computers. Second,

and of direct concern to DISORT, cases where single-scattering albedo is very near unity would probably be handled better. These are currently the cases of most concern for ASYMTX, since the eigenvalue problem becomes singular in this limit. Third, eventually all of LAPACK will be just a big library which is automatically linked to Fortran programs (as already happens on Cray supercomputers), and then concerns about size would become irrelevant — using LAPACK would become as simple as using Fortran intrinsic functions.

Converting SGEEV and its subroutines to real arithmetic, as Tsay did with ASYMTX, is too daunting a task for us to undertake without help. Thus, we communicated several times with the principal author of LAPACK and also with one of the numerical analysts who runs the famous numerical analysis web site "netlib", about providing eigenvalue solvers specialized to the discrete-ordinates application, but so far to no avail. The market must be too small to interest them.

As discussed by Stamnes et al. (1988b), the advantage of solving the algebraic eigenvalue problem involving the *asymmetric* matrix A in (23f) is that only one matrix multiplication is necessary. Since the eigenvalues are real, one might suspect that it is possible to transform this algebraic eigenvalue problem into one involving *symmetric* matrices. This is indeed the case, but such a transformation and subsequent solution procedures introduced by Nakajima and Tanaka (1986) involved matrix operations in which the effect of rounding error became significant. Stamnes et al. (1988b) introduced the Cholesky decomposition to symmetrize the matrix and compared this new procedure both with Nakajima/Tanaka's and with the old one of Stamnes/Swanson. They summarize the outcome of these comparisons as follows:

(1) All three methods work well when executed in double precision.

(2) In single precision the Stamnes/Swanson method is more accurate than the Cholesky method which in turn is slightly more accurate than the Nakajima/Tanaka method.

(3) All three methods take similar computer times, within 30%; there is no great speedup from symmetrizing the eigenproblem, because of the overhead of extra matrix multiplications.

Based on these findings, Stamnes et al. (1988b) decided to examine the possibility of speeding up the Stamnes/Swanson method, since surprisingly it had turned out to be the most accurate. It was decided that the greatest speedup would come from eliminating complex arithmetic in the EISPACK QR-algorithm program that had been used as a "black box" by Stamnes/Swanson. This was the motive for the creation of DISORT's subroutine ASYMTX.

One thing which is a little confusing is how the actual eigenvectors $\mathbf{G}^+$ and $\mathbf{G}^-$ are calculated.

This is done in SOLEIG, which recovers the eigenvectors from their sum and difference:

$$\mathbf{G}^+ = \frac{1}{2}\left[\left(\mathbf{G}^+ + \mathbf{G}^-\right) + \left(\mathbf{G}^+ - \mathbf{G}^-\right)\right]$$

$$\mathbf{G}^- = \frac{1}{2}\left[\left(\mathbf{G}^+ + \mathbf{G}^-\right) - \left(\mathbf{G}^+ - \mathbf{G}^-\right)\right]$$

The sum and difference are what is actually produced when the calculation is organized along the lines of Eqs. (21) and (23).

Before discussing the general case, we shall first show how ASYMTX handles the special cases N=1 (two-stream) and N=2 (four-stream). The original EISPACK routines did not have special branches for these cases.

3.4.1. Two-stream case

The two-stream case is trivial—just the scalar equation $A_{11}x=kx$, where x is the eigenvector of A corresponding to eigenvalue k. The final result is

$$k_1 = A_{11}, \quad x_1 = 1 \tag{60}$$

where of course any convenient value can be chosen for x_1.

3.4.2 Four-stream case

Our eigenvalue problem (23f) in the four-stream case is

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = k \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Even for this simple 2x2 case, it turns out to be impossible to extract the solution formulas from ASYMTX, because knowledge of the numerical values of the matrix elements is required during the "balancing" process. It is only possible in the two special cases A_{21}=0 and A_{12}=0. Once we see how ASYMTX handles these two special cases, we can construct a general solution which reduces correctly to them. Here are the ASYMTX results:

***A_{21}*=0:**

$$k_1 = A_{11}; \quad k_2 = A_{22}$$

$$\mathbf{x}^{(1)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{61a}$$

$$\mathbf{x}^{(2)} = \begin{cases} \begin{bmatrix} A_{12}/(A_{22} - A_{11}) \\ 1 \end{bmatrix} & \text{for } A_{11} \neq A_{22} \\ \begin{bmatrix} -A_{12}/\Delta \\ 1 \end{bmatrix} & \text{for } A_{11} = A_{22} \end{cases}$$

***A_{12}*=0:**

$$k_1 = A_{22}; \quad k_2 = A_{11}$$

$$\mathbf{x}^{(1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{61b}$$

$$\mathbf{x}^{(2)} = \begin{cases} \begin{bmatrix} 1 \\ A_{21}/(A_{11} - A_{22}) \end{bmatrix} & \text{for } A_{11} \neq A_{22} \\ \begin{bmatrix} 1 \\ -A_{21}/\Delta \end{bmatrix} & \text{for } A_{11} = A_{22} \end{cases}$$

$$\Delta = \Xi \left(|A_{11}| + |A_{12}| + |A_{21}| + |A_{22}| \right)$$

where Ξ is close to machine precision. The eigenvector element in which Δ occurs is thus very large compared to any matrix element of A. This is designed to allow the eigenvector to be correct to machine precision. Obviously there is a good deal of arbitrariness in this case, since the eigenvalue problem is degenerate, and it is valuable to see how ASYMTX handles it.

In the second case, there is no particular reason for inverting the order of eigenvalues k compared to the first case, so we shall ignore that reordering in making our general solution agree with the above special cases.

The general solution involves solving the quadratic equation

$$\begin{vmatrix} A_{11} - k & A_{12} \\ A_{21} & A_{22} - k \end{vmatrix} = 0 \quad \Longrightarrow \quad k^2 - (A_{11} + A_{22})\,k + (A_{11}A_{22} - A_{12}A_{21}) = 0$$

The sign in the quadratic solution must be chosen carefully in order to reduce correctly to the special cases, and also to avoid dividing by zero in certain eigenvector elements. The correct choice is:

$$k_1 = \begin{cases} k_+ & \text{if } A_{11} \geq A_{22} \\ k_- & \text{if } A_{11} < A_{22} \end{cases} \qquad k_2 = \begin{cases} k_- & \text{if } A_{11} \geq A_{22} \\ k_+ & \text{if } A_{11} < A_{22} \end{cases} \tag{62}$$

where

$$k_+ \equiv \tfrac{1}{2}\left\{A_{11} + A_{22} + \sqrt{(A_{11} - A_{22})^2 + 4A_{12}A_{21}}\right\}$$

$$k_- \equiv \tfrac{1}{2}\left\{A_{11} + A_{22} - \sqrt{(A_{11} - A_{22})^2 + 4A_{12}A_{21}}\right\}$$

which produces the following limits:

$$k_1 \xrightarrow[A_{12}, A_{21} \to 0]{} A_{11}$$

$$k_2 \xrightarrow[A_{12}, A_{21} \to 0]{} A_{22}$$

To get eigenvectors $\mathbf{x}j$, we assume the jth element of $\mathbf{x}_j$ is always unity, which agrees with the special cases for which we know the ASYMTX solution. Thus, for $\mathbf{x}1$ we take x_1=1 and solve for x_2 from either

$$(A_{11} - k_1)\, x_1 + A_{12}\, x_2 = 0 \quad \Longrightarrow \quad x_2 = \frac{k_1 - A_{11}}{A_{12}}$$

or from

$$A_{21}\, x_1 + (A_{22} - k_1)\, x_2 = 0 \quad \Longrightarrow \quad x_2 = \frac{A_{21}}{k_1 - A_{22}}$$

We prefer the latter solution because it is well-behaved as A_{12}-->0 (due to the way we picked k_1) while the former solution is not.

For $\mathbf{x}2$, similar reasoning leads us to take x_2=1 and

$$(A_{11} - k_2)\, x_1 + A_{12}\, x_2 = 0 \quad \Longrightarrow \quad x_1 = \frac{A_{12}}{k_2 - A_{11}}$$

Thus, finally, the matrix whose columns are the eigenvectors (and which is returned by ASYMTX) is

$$\left[\mathbf{x}^{(1)} \quad \mathbf{x}^{(2)}\right] = \begin{bmatrix} 1 & \dfrac{A_{12}}{k_2 - A_{11}} \\ \dfrac{A_{21}}{k_1 - A_{22}} & 1 \end{bmatrix} \tag{63a}$$

This reduces correctly to the special cases A_{21}=0 and A_{12}=0 we pulled out of ASYMTX above, provided $A_{11} \neq A_{22}$.

When A_{11}=A_{22}, the eigenvalue expressions above apply, but the eigenvector expressions will blow up if either A_{21}=0 or A_{12}=0, so we have the following special cases

$$\begin{aligned} \mathrm{x}_1 &= \begin{bmatrix} 1 \\ A_{21}/\Delta \end{bmatrix} \quad \text{if } A_{12} = 0 \\ \mathrm{x}_2 &= \begin{bmatrix} A_{12}/\Delta \\ 1 \end{bmatrix} \quad \text{if } A_{21} = 0 \end{aligned} \tag{63b}$$

Given how tricky this seemingly simple case becomes, it is a daunting prospect to develop the analytic solution for the 3x3 case, although this might be worthwhile if six-stream approximations were for some reason to overtake two- and four-stream approximations in popularity (highly unlikely in our opinion).

3.4.3 Removable singularities in the intensities

Certain combinations of the directions and eigenvalues may lead to 0/0-type singularities in the intensities and in (35a,b). These singularities are removed by applying L'Hospital's Rule. The resulting equations, below, are applied in USRINT.

The discrete ordinate solutions for the p^{th} layer (35a,b) contain the following expressions:

$$I_p(\tau,+\mu) \propto \frac{E_{jn}(\tau,+\mu)}{1+k_{jn}\mu} \exp(k_{jn}\tau_n) \equiv f_1(\tau,k_{jn}\mu)$$

$$I_p(\tau,-\mu) \propto \frac{E_{jn}(\tau,-\mu)}{1-k_{jn}\mu} \exp(k_{jn}\tau_{n-1}) \equiv f_2(\tau,k_{jn}\mu)$$

The E_{jn} are given in (36a, 37a). The exponential factors on the right hand sides come from applying the scaling transformation (48). Whenever $|k_{jn}\mu|=1$, the upward and downward intensities have a 0/0-type singularity for the eigenvalues $k_{jn}<0$ and $k_{jn}>0$, respectively. Applying L'Hospital's Rule leads to

$$f_1(\tau,k_{jn}\mu=-1) = \frac{\tau_n-\tau_{n-1}}{\mu} \exp(-(\tau_n-\tau)/\mu)$$

$$f_2(\tau,k_{jn}\mu=+1) = \frac{\tau_n-\tau_{n-1}}{\mu} \exp(-(\tau-\tau_{n-1})/\mu)$$

The particular solution of the downward intensity for the beam source [$I_p(\tau,-\mu)$] is proportional to

$$f(\tau,-\mu,\mu_0) \equiv \frac{E_{jn}(\tau,-\mu)}{1-\mu/\mu_0}$$

This is undefined at $\mu=\mu_0$, that is again avoided by applying L'Hospital's Rule leading to

$$f(\tau,-\mu=\mu_0,\mu_0) = \frac{\tau_n-\tau_{n-1}}{\mu_0} \exp(-\tau/\mu_0)$$

3.5 Numerical Solution for the Constants of Integration

Applying the boundary conditions plus continuity conditions at layer interfaces led to (46), which constituted a (*2N*x*L*) x (*2N*x*L*) system of linear algebraic equations for the *2N*x*L* unknown coefficients C_{jp} ($j=\pm1,...,\pm N$; $p=1,...,L$). This set of equations is ill-conditioned for large total optical thickness or large individual layer thickness, so a scaling transformation was introduced in (48) which entirely eliminates this problem.

Fortunately, the coefficient matrix for this system of algebraic equations is a banded matrix

with ($6N$–1) diagonals (Stamnes and Conklin, 1984). For instance, there are 11 diagonals in the four-stream case (N=2). Both LINPACK and LAPACK include solvers specifically designed for banded matrices. In DISORT we use the LINPACK (cf. Dongarra et al., 1979) subroutine SGBSL to solve (46) for the C-hat's.

Since we are dealing with a sparse matrix, the run time will be proportional to the number of layers. This is so because a band matrix of order I and band width J can be solved in a time proportional to IJ^2. In our case, I=2NL and J=N implying that the run time will increase as LN^3 (linearly with the number of layers and cubically with the number of streams).

3.6 Correction of the Intensity Field

Intensity correction is a new feature in DISORT v2.x which was not in v1.x. It is implemented in INTCOR and several subroutines which INTCOR calls. The corrections are applied in subroutine DISORT after the intensities have been computed in the normal (v1.x) manner. Thus, they are essentially a post-processing step.

The purpose of intensity correction is to achieve high intensity accuracy with very few streams N, thereby dramatically shrinking the computational burden. With the corrections, N=4–6 will usually give better than 1% intensity accuracy; this level of accuracy previously required N=20 or more.

Intensity correction involves the following changes to the input/output arguments of DISORT:

(a) the full phase function Legendre expansion (all the significant Legendre coefficients) must be provided, not just the Legendre coefficients from 0 to 2N as in v1.x;

(b) the DELTAM option is no longer an input argument and is always turned on (it can still be turned off internally for testing); inexperienced users should not be given the option of turning DELTAM off, since that will almost always result in poorer results; DELTAM is not needed in cases of weakly varying phase function, e.g. Rayleigh scattering, but it does no harm in such cases either;

(c) the azimuthally-averaged intensity is no longer returned as an output; it is hard to calculate corrections for it, and it did not seem worth the effort since user demand for it is basically non-existent;

In all radiative transfer algorithms for strongly anisotropic scattering, the selection of the point

at which series are truncated (manifesting itself in DISORT as the choice of N=number of streams) is an age-old problem. The difficulty stems from the need to use large values of N, sometimes more than several hundred, to accurately resolve the anisotropic scattering pattern if one solves the problem in a brute-force way. This can consume enormous amounts of computing resources, since the computational burden grows roughly as N^3. Therefore, truncation procedures which provide accurate intensities while keeping N small are highly desirable.

The δ–M transformation (cf. Section 3.2.2) has proven to be a very useful truncation procedure for *flux* computations, because it captures the kind of anisotropy (strong forward peak due to diffraction) found in almost all natural scattering phenomena. Nakajima and Tanaka (1988) showed, however, that the δ–M method produces unacceptable error for *intensity* computations of practical size (N<10–20), especially in the bright region close to the direct beam (aureole). But they went on to show that an ingenious correction of the δ–M intensities for single and double scattering can reduce the intensity error below 1%.

The problem with δ–M is illustrated in Figures 3 and 4. Figure 3a shows an aerosol phase function and the corresponding δ–M approximation for N=10. The δ–M approximation oscillates around the correct phase function, which becomes even clearer in the difference plot, Figure 3b. The oscillations would get worse for smaller N. Figure 4a,b show the corresponding intensity fields for a slab with optical depth 0.8; the δ–M intensity oscillates around the correct intensity in synchronism with the corresponding phase function oscillations. This suggests that the single-scattered intensity, which is proportional to the phase function, contributes most to the oscillation, and that δ–M is misrepresenting single scattering rather badly. Thus, we should first construct the exact solution for single-scattered intensity and use it to correct the DISORT δ–M intensities.

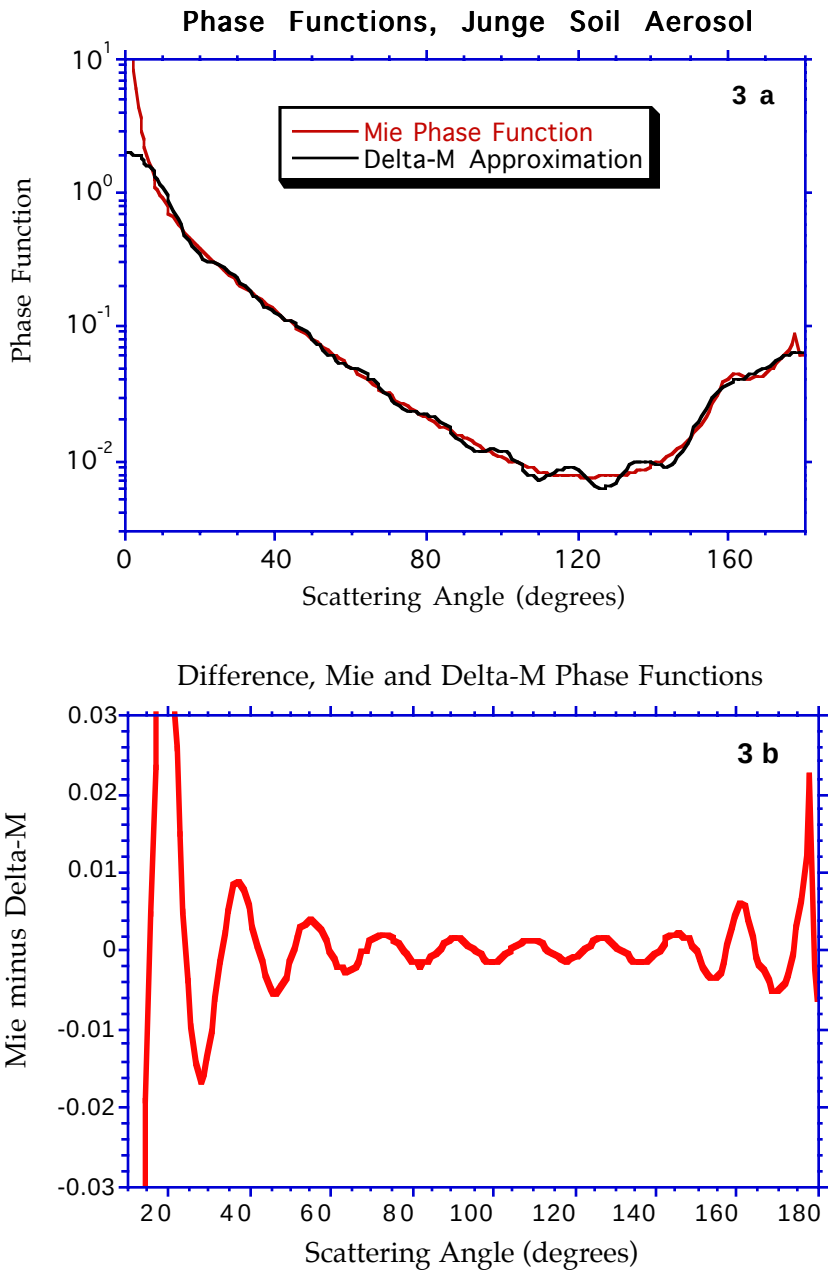

Figure 3. (a) Mie scattering phase function for a typical aerosol in the atmosphere, and the corresponding δ–M approximation for N=10 (20 streams); (b) difference between the two phase functions in (a). Note oscillations.

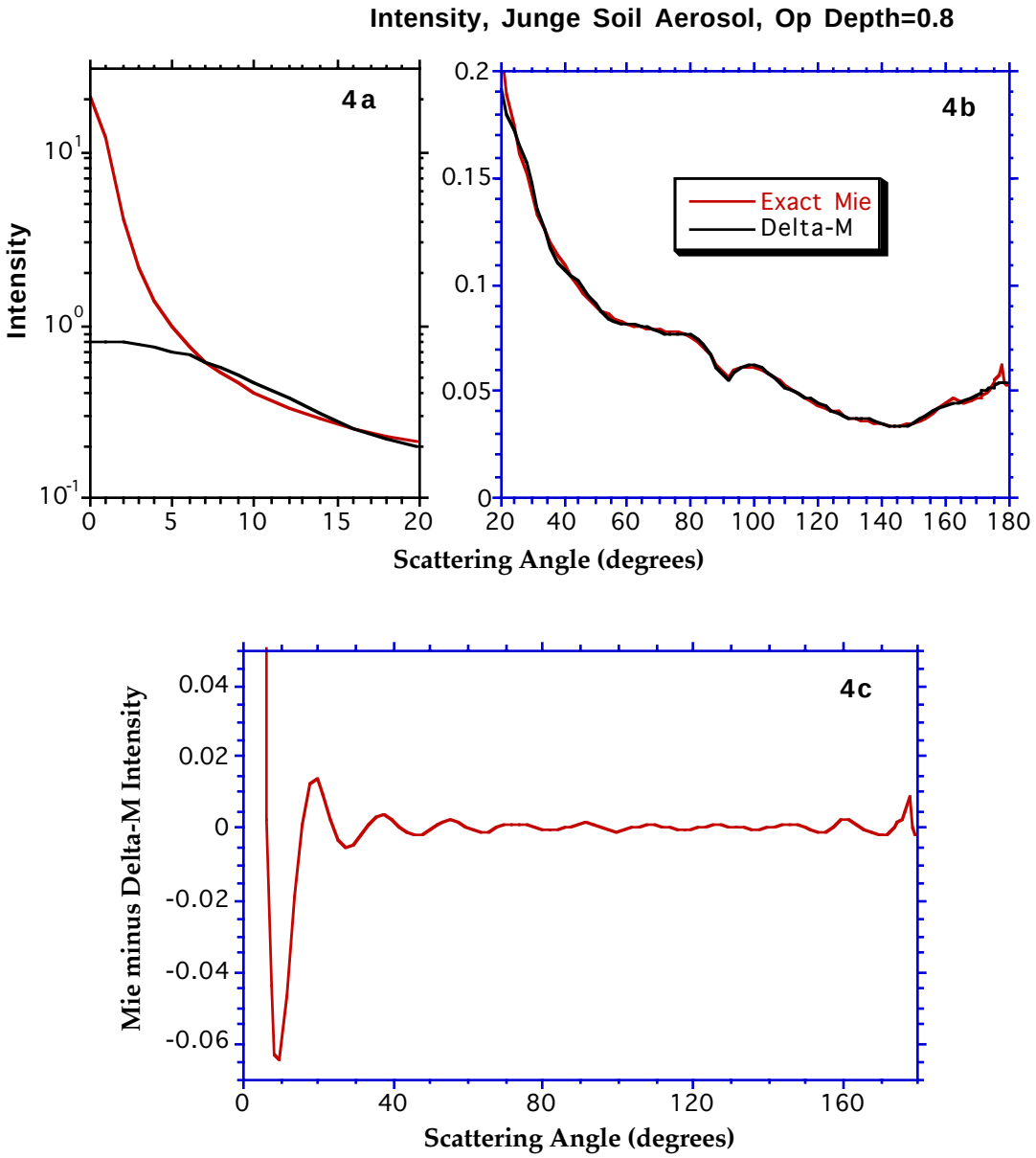

Figure 4. The intensities vs. angle corresponding to the phase function in Fig. (3a) for an overhead incident sun and total optical depth of 0.8 (after Nakajima and Tanaka, 1988). (a and b) show the intensities using a log scale for small angles, while (c) shows their difference. Note how the δ–M phase function oscillations in Figure 3 are mirrored in the intensities.

3.6.1 Single scattering solution

The single-scattering solution is implemented in SINSCA. This is where *all* the Legendre moments of the phase function are required, in order to reconstruct the phase function accurately. Neglecting the multiple scattering source term in (2) leads to the single-scattering special case of the radiative transfer equation (1):

$$\pm\mu \frac{\mathrm{d}I_{ss}(\tau,\pm\mu,\phi)}{\mathrm{d}\tau} = I_{ss}(\tau,\pm\mu,\phi) - \frac{I_0}{4\pi}\,\omega(\tau)\,P(\tau,\pm\mu,\phi;-\mu_0,\phi_0)\,e^{-\tau/\mu_0} \qquad (64)$$

where the subscript "ss" refers to single-scattering. This is the simplest nontrivial ODE, treated in every elementary calculus textbook. Considering first the upward intensity field (indicated by the notation "+μ" with μ>0), we multiply (64) by an integrating factor exp(–τ/μ) and integrate both sides from τ_L to τ (cf. Figure 1) subject to the upgoing intensity at the lower boundary condition $I(\tau_L)$:

$$\int_{\tau_L}^{\tau} \frac{\mathrm{d}\left[I_{ss}(t,+\mu,\phi)e^{-\tau/\mu}\right]}{dt}\,dt = -\frac{I_0}{4\pi\mu}\int_{\tau_L}^{\tau} \omega(t)\,P(t,+\mu,\phi;-\mu_0,\phi_0)\,e^{-t/\mu-t/\mu_0}\,\mathrm{d}t \qquad (65a)$$

For simplicity, we may set $I(\tau_L)$=0 and carry out the integration layer by layer

$$I_{ss}(\tau,+\mu,\phi) = \frac{I_0}{4\pi(1+\mu/\mu_0)}\sum_{n=p}^{L}\omega_n\,P_n(+\mu,\phi;-\mu_0,\phi_0)\,\times \left\{\exp\left(-\frac{\tau_{n-1}-\tau}{\mu}-\frac{\tau_{n-1}}{\mu_0}\right)-\exp\left(-\frac{\tau_n-\tau}{\mu}-\frac{\tau_n}{\mu_0}\right)\right\} \qquad (65b)$$

with τ_{n-1} replaced by τ for $n=p$. Since $\tau_L>...>\tau_n>\tau_{n-1}\geq\tau$, all the exponentials in (65b) have negative arguments as they should.

Likewise, for the downward intensity field (indicated by the notation "–μ" with μ>0), we multiply (64) by an integrating factor exp(τ/μ) and integrate both sides from τ_0 to τ subject to the downgoing intensity at the upper boundary condition $I(\tau_0)$:

$$\int_{\tau_0}^{\tau} \frac{\mathrm{d}\left[I_{ss}(t,-\mu,\phi)e^{+t/\mu}\right]}{dt}\,dt = \frac{I_0}{4\pi\mu}\int_{\tau_0}^{\tau} \omega(t)\,P(t,-\mu,\phi;-\mu_0,\phi_0)\exp\left(\frac{t}{\mu}-\frac{t}{\mu_0}\right)\mathrm{d}t \qquad (65c)$$

Also, we may let $I(\tau_0)=0$ for $\tau_0=0$ and carry out the integration layer by layer for $\mu\neq\mu_0$:

$$I_{ss}(\tau,-\mu,\phi) = \frac{I_0}{4\pi(1-\mu/\mu_0)}\sum_{n=1}^{p}\omega_n\,P_n(-\mu,\phi;-\mu_0,\phi_0)\,\times \left\{\exp\left(-\frac{\tau-\tau_n}{\mu}-\frac{\tau_n}{\mu_0}\right)-\exp\left(-\frac{\tau-\tau_{n-1}}{\mu}-\frac{\tau_{n-1}}{\mu_0}\right)\right\} \qquad (65d)$$

with τ_n replaced by τ for $n=p$. Again, since $\tau>\ldots>\tau_n>\tau_{n-1}\geq 0$, all the exponentials in (65d) have negative arguments. In the special case $\mu=\mu_0$, this reduces to:

$$I_{ss}(\tau,-\mu,\phi) = \frac{I_0}{4\pi\mu_0}e^{-\tau/\mu_0}\sum_{n=1}^{p}\omega_n\,P_n(-\mu,\phi;-\mu_0,\phi_0)(\tau_n-\tau_{n-1}) \qquad (65e)$$

with τ_n replaced by τ for $n=p$. This simpler form is obtained directly from (65c), or alternatively by the application of L'Hospital's rule to (65d). (65b,d,e) are the formulas used in SINSCA.

3.6.2 Nakajima/Tanaka intensity corrections

Intensity corrections based on the paper of Nakajima and Tanaka (1988) are applied in INTCOR. Nakajima/Tanaka (hereinafter N/T) developed several correction procedures to improve the accuracy of the δ–M intensity and eliminate oscillatory errors like those we saw in Figures 3-4. Recall that the δ–M approximation to the phase function is:

$$P_{exact}(\tau,\cos\Theta) \approx 2f\,\delta(1-\cos\Theta)+(1-f)\,P'(\tau,\cos\Theta) \qquad (66)$$

where P' is expanded in $2N$ Legendre polynomials (cf. 54a). The N/T procedures adjust the corresponding δ–M (primed) intensities by adding a correction term ΔI

$$I_{\text{corrected (TMS or IMS)}} = I' + \Delta I_{\text{(TMS or IMS)}} \qquad (67)$$

which first subtracts out the "wrong" single/double scattering part of the DISORT solution, then adds back the "correct" single/double scattering solution.

The simplest correction procedure, N/T's so-called *MS method*, which corrects only for using the δ–M phase function but not for the δ–M scaling of optical depth or single-scatter albedo, leads to a considerable improvement in accuracy, as shown in Figure 5a for the same aerosol case as in Figures 3–4. But we bypass the MS method in favor of a second N/T method, the so-called *TMS method*, which in addition corrects for the δ–M scaling. The improvement of TMS relative to MS is illustrated in Figure 5b; there is a factor of 20 reduction in y-axis scale in Figure 5b relative to the MS error plot in Figure 5a. In Figure 5b, the TMS error is acceptably low (a few tenths of a percent) except in the solar aureole near zero degrees scattering angle. In general, the TMS-method yields very satisfactory results except in this aureole region. Further corrections can be done in the aureole region, using N/T's so-called *IMS method*. The improvement from going over to the more complicated IMS method is indicated in Figure 5b. INTCOR includes both TMS and IMS correction procedures.

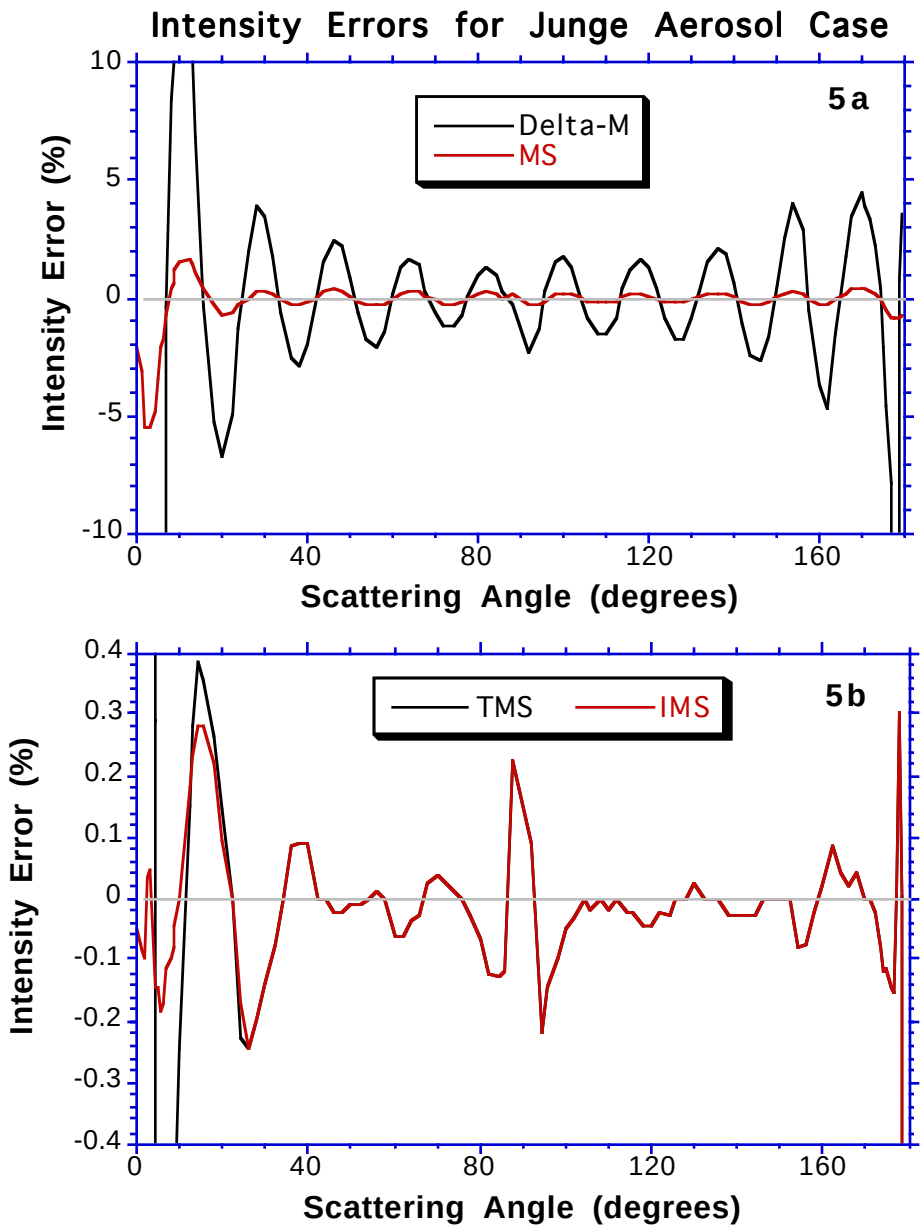

Figure 5. Relative error (%) of intensity fields computed by (a) the δ–M and MS methods, and (b) the TMS and IMS methods for the same aerosol case as in Figs. 3–4 (after Nakajima and Tanaka, 1988). TMS and IMS agree past about 30 degrees. Note change in scale in (b) relative to (a).

We begin by describing the TMS method. We need to define a somewhat schematic notation for the single-scattering solution of Section 3.6.1:

$$\Im[\text{Phase function, optical depth}] \equiv \text{single scattering solution } I_{ss}$$

The reason for introducing this functional is that we want to use it with both the *exact* phase function *P, calculated from all available terms in the Legendre expansion,* and the δ–*M* phase function *P´, calculated from just 2N terms.* The TMS method is quite simple, although one needs to gaze at it a while before it begins to seem completely natural. Its essence is to subtract the single-scattered intensity obtained from the δ–M method, then add back the τ–scaled intensity based on the exact phase function and the unscaled single scattering albedo:

$$\Delta I_{TMS} = \Im\left[\frac{\omega P}{1-\omega f}, \tau'\right] - \Im[\omega' P', \tau'] \tag{68a}$$

where the factor $(1-\omega f)$ is a consequence of the δ–M τ–scaling (54d). Another way to say this is: the TMS method attempts to correct the single-scattered intensity obtained from the δ–M scaling by replacing it with the single-scattered intensity computed without scaling the single scattering albedo and the phase function. Since the single-scattering solution is linear in the phase function, (68a) can be simplified to

$$\Delta I_{TMS} = \Im\left[\frac{\omega P - (1-f)\omega P'}{1-\omega f}, \tau'\right] \tag{68b}$$

where we have used the δ–M scaling equations (54d) to replace ω´.

The N/T IMS correction attempts to further correct for the errors of the TMS approximation in the forward scattering angular region. Few in the radiative transfer community have been able to understand the IMS method based on the description in N/T's paper. Furthermore, it is rather complicated and requires the introduction of considerable notation to reveal the essence of the method. Thus, we thought it better to relegate our interpretation of this method to Appendix A. After applying the IMS correction, the example in Figure 5b shows that all emergent intensities agree with the exact solutions to within 1%. This is typical of the accuracy of the IMS method.

3.6.3 Restrictions and limitations

The Nakajima/Tanaka corrections are not completely general, although they could perhaps be made more general. The main restrictions and limitations are as follows:

(1) the derivation ignores thermal emission (Planck) sources; the corrections are still applied when there are Planck sources, on the grounds that accounting correctly for single scattering is always a good idea, but very little is known about the more subtle interactions of Planck sources with these corrections;

(2) the derivation ignores surface reflection [$I(\tau_L)=0$ in Eq 65b]; however, surface reflected photons can be regarded as multiply scattered and thus part of the multiply scattered field; still, if the surface were specularly reflecting, this could be a problem;

(3) the derivation ignores any diffuse intensity incident on the top boundary [$I(\tau_0)=0$ in Eq 65d]; this probably doesn't matter much;

(4) the corrections are not applied to flux, flux divergence or mean intensity (or azimuthally-averaged intensity, but we are dropping that as an output variable); this makes these quantities technically inconsistent with the returned intensities, although only slightly so; part of the justification is that the δ-M method has a long history of computing flux accurately, and it also conserves flux exactly—applying corrections to it might violate flux conservation and possibly make the accuracy worse;

It also must be admitted that the intensity correction methods have not been completely tested for all optical depths, single-scatter albedos, and phase functions. They were originally devised for aerosol cases with optical depths no larger than unity, and indeed single scattering has a big effect in just such cases. When we turned on the intensity corrections for our existing DISORT test cases, however, the results usually improved and never got worse. If this is any indication, then the corrections work for a wide range of conditions.

3.7 Azimuthal Convergence

From (6) one would conclude that $2M$ terms are needed in the azimuthal series for intensity—as many terms as in the phase function Legendre expansion (5a), which can be quite long. However, a peculiar feature of the discrete ordinates method, first noted by its inventor Chandrasekhar (Ch. VI, Eq. 93), is that the number of streams needs to exceed the number of terms in the phase function expansion:

$$2N \geq 2M$$

In practice, for typical phase functions, this condition would be impossible to satisfy; it would require hundreds of streams and impossibly large computations. Thus, this inequality is reversed, and M determined from N rather than vice versa. The number of streams $2N$ is picked based on how much computation the user can afford (and whether they want only fluxes not intensities); and the azimuthal series is merely truncated at $2M=2N$ terms. It is inconsistent to carry it any

farther out. The error this causes is simply tolerated.

Work by King (1983) and others has shown that, if one can use an arbitrarily large number of streams $2N$, the azimuthal series typically converges well before $2N$ terms; usually no more than 10–20 terms are needed. Since each extra term means a lot of extra computation (the azimuthal summing loop is the outermost loop in DISORT), it is important to carefully monitor the convergence of this series and not take any more terms than absolutely necessary, at the same time avoiding prematurely truncating the series. DISORT monitors the azimuthal convergence using a Cauchy criterion. Azimuth summing successively creates the partial sums

$$I_K(\tau,\mu,\phi) = \sum_{m=0}^{K} I^m(\tau,\mu)\ \cos m(\phi_0 - \phi) \quad (K < 2M-1)$$

The Cauchy convergence criterion is that

$$\max_{\tau,\mu,\phi} \frac{\left| I^m(\tau,\mu)\ \cos m(\phi_0 - \phi) \right|}{\left| I_K(\tau,\mu,\phi) \right|} \le \varepsilon_{azim}$$

(where the max is over the user optical depths and angles, not the computational ones). In order to avoid accidentally terminating the series before true convergence, azimuth summing will only stop when this criterion is satisfied *twice* . This is conservative and probably wastes computation in the vast majority of cases, but it protects against bad results in cases with few user optical depths or angles. ε_{azim} should be below 0.01 to avoid the risk of serious non-convergence.

3.8 Simplified Albedo and Transmissivity Computations

DISORT has a special case, managed by ALBTRN, which allows one to get the flux albedo and transmission of the whole medium for many incident beam angles at once, for the same cost as getting these quantities for just one beam angle in a normal DISORT calculation. The special-case code calls a few of the same subroutines as the normal case code, but is essentially self-contained; it even has its own printing routine. (Note that this special case excludes thermal sources and non-Lambertian lower boundary reflectivities.)

By appealing to the reciprocity principle, Stamnes (1982b) derived simple expressions for the albedo and transmissivity of a vertically inhomogeneous, plane parallel medium lacking thermal sources:

$$a(\mu) = I^0(0,\mu) \qquad t(\mu) = I^{0*}(0,\mu)$$
$$a^*(\mu) = I^{0*}(\tau_L,-\mu) \qquad t^*(\mu) = I^0(\tau_L,-\mu) \tag{69}$$

where the asterisk refers to illumination from below and τ_L is, as usual, the total optical thickness of the inhomogeneous medium. The explanations of the top two equations in (69) are as follows:

(a) the albedo for a given angle μ of parallel-beam incidence equals the azimuthally-averaged reflected intensity *for isotropic unit intensity incident at the top boundary;*

(b) the transmissivity equals the azimuthally-averaged transmitted intensity *for isotropic unit intensity incident at the bottom boundary*.

(69) offers substantial computational advantages. The conventional procedure to compute $a(\mu)$ and $t(\mu)$ for a particular angle of incidence μ requires, first, computing the angular distribution of the azimuthally-averaged intensity, and, second, the quadrature of this intensity over angle. This requires the computation of the particular solution (Section 2.5) for each layer and every direction of incidence considered, which can be quite costly in an inhomogeneous (multi-layered) medium. In contrast, by applying an isotropic boundary condition and using (69), one may solve for all desired angles of incidence simultaneously and entirely avoid the computation of the particular solution.

The numerical implementation of this special case involves the following two steps (Stamnes, 1982b):

(1) Compute albedo and transmissivity for the so-called "standard problem" (no reflecting lower boundary) for isotropic illumination of unit intensity from above and below using (69). For illumination from above, the following explicit expressions [cf. (52a) and (53a)] are used to calculate two of the right-hand sides in (69):

$$\begin{Bmatrix} I^0(0,+\mu) \\ I^0(\tau_L,-\mu) \end{Bmatrix} = \sum_{n=1}^{L}\sum_{j=1}^{N} \left\{ \hat{C}^0_{-jn} \frac{G^0_{-jn}(\pm\mu)}{1\mp k^0_{jn}\mu} \hat{E}^0_{-jn}(\pm\mu) + \hat{C}^0_{+jn} \frac{G^0_{+jn}(\pm\mu)}{1\pm k^0_{jn}\mu} \hat{E}^0_{+jn}(\pm\mu) \right\} \tag{70}$$

where the 0-superscript on *C, E, k* refers to the azimuth-independent case (*m*=0) and $\hat{E}$ is defined in (52b-d) and (53b-d). Two equations have been collapsed into one here, because the right-hand sides in (70) are identical except for some sign flips; the two left-hand sides are stacked inside the curly brackets.

(70) gives us $a(\mu)$ and $t^*(\mu)$. The same formulas also give us $a^*(\mu)$ and $t(\mu)$. Only the constants of integration C are different. Thus, only the final step in the solving procedure, which determines these constants, needs to be repeated for illumination from below.

(2) Modify the solution from step (1) to account for a Lambert (isotropic) reflecting lower boundary. Chandrasekhar (1960) called this the "planetary problem" as opposed to the "standard problem" which has zero surface reflection. He derived simple expressions for the planetary-problem solution in terms of the standard-problem solution for a homogeneous slab. Stamnes (1982b) generalized this derivation to an inhomogeneous atmosphere above a surface that reflects radiation anisotropically but with no dependence on the direction of incidence. Such a surface seems rather unphysical, and certainly doesn't satisfy reciprocity, so in DISORT only true Lambert surface reflection is allowed in this special case.

3.9 Computational Speed

Unlike other radiative transfer methods like adding-doubling, DISORT computation time is basically independent of optical thickness (because the solutions are closed-form analytic functions of optical thickness) — but it does depend linearly on the number of computational layers needed to resolve the vertical structure. Since the DISORT solutions are analytic, the number of computational layers can be determined by the actual structure of the problem rather than, as in some other methods, by mere numerical constraints. (The only exception to this rule is that temperature changes across layers should be less than about 10K to avoid significant errors in the thermal emission computation.) In general the number of computational layers will be determined by how accurately one wants to resolve the optical properties (absorption/scattering characteristics) of the medium, which are taken to be constant across any layer. For example, if one is doing ultraviolet radiative transfer for a cloud embedded in a pure Rayleigh atmosphere, one need only specify three computational layers to DISORT, no matter how optically thick the cloud is, provided that the single-scatter albedo and phase function are constant throughout the cloud.

DISORT's computation time goes roughly as the third power of the number of "streams" (quadrature angles). This is clearly the main factor affecting DISORT computation time. For strongly forward-peaked phase functions it is difficult to get accurate intensities with fewer than 16 streams, and even with 16 streams accuracy can be poor at some angles. Thus, careful users have been forced to use 32 or even 64 streams to be sure of getting 1% accuracy (increasingly important as radiometers get better and better). Fortunately, the new Nakajima/Tanaka improvements in DISORT v2.0 allow intensities to be achieved with as few as 4–8 streams, affording factors of

10–100 increases in computational speed for a given level of accuracy.

Godsalve (1996) discovered some ways to speed up the DISORT particular solution computation; however, the speedups (10% in multi-layer cases) did not seem compelling enough to try to alter DISORT to include them.

3.10 Remarks on Computer Precision

DISORT has certain intrinsic limitations because of computer precision. These limitations are related to ordinary computer "roundoff error" and have nothing to do with more easily controllable (through the number of streams) "truncation error". DISORT is free of the *catastrophic* growth of roundoff error that plagued pre-1980 discrete ordinate programs, but certain parts of the calculation (the eigenvalue/vector and Gauss quadrature rule computations, and to a much lesser extent the linear equation solving computations) are just inherently more sensitive to roundoff than the rest of DISORT, because they involve so many arithmetic operations.

The reason DISORT was for many years distributed with the eigenvalue/vector and Gauss quadrature rule subroutines in double precision is that DISORT was originally developed on 32-bit-single-precision computers (VAXes and IBMs) which were too inaccurate for those computations. Running DISORT on a typical 32-bit-single-precision computer usually gives results precise to at least 2–3 significant digits, although for certain special situations the precision can fall to one significant digit. Results below about 10^{-8} times the driving radiation can even have *no* significant digits, as one can see in the fluxes in some of the test problems. To alleviate these difficulties, we recommend that DISORT be run entirely in double precision wherever possible. This is overkill for some subroutines, but is simpler than juggling mixed precision. With double precision hardware on almost all computers these days, the computer time penalty is rarely more than 20% (much less than in 1988 when some computers still performed double precision arithmetic in software).

On some Unix workstations, certain compiler options can also cause IEEE underflows to trigger thousands of error messages and/or abort the run. DISORT will never pass underflow tests because it has potential underflows everywhere, especially in the computations of exponentials.

4. REFERENCES

Anderson, E., Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, 1999: LAPACK User's Guide, 3rd Edition, Society for Industrial and Applied Mathematics, 3600 University City Science Center, Philadelphia, Pennsylvania 19104-2688 (web-based equivalent at http://www.netlib.org/lapack/lug/lapack_lug.html)

Berk, A., L. Bernstein, G.P. Anderson, P. Acharya, D. Robertson, J. Chetwynd, S. Adler-Golden, 1998: MODTRAN cloud and multiple scattering upgrades with application to AVIRIS, Rem. Sens. Envir. 65, 367-375.

Bohren, C. and D. Huffman, 1983: **Absorption and Scattering of Light by Small Particles**, Wiley, New York

Chandrasekhar, S., 1960: **Radiative Transfer**, Dover, New York.

Cowell, W.R. (Ed.), 1980: **Sources and Developments of Mathematical Software**, Prentice Hall, Englewood Cliffs, New Jersey.

Dave, J.V. and B. Armstrong, 1970: Computations of high-order associated Legendre polynomials, J. Quant. Spectrosc. Radiat. Transfer 10, 557-562

Davis, P. and P. Rabinowitz, 1975, 1984: **Methods of Numerical Integration**, Academic Press, New York.

Dongarra, J., C. Moler, J. Bunch and G.W. Stewart, 1979: **LINPACK User's Guide**. Society for Industrial and Applied Mathematics (SIAM) Press, Philadelphia.

Godsalve, C., 1995: The inclusion of reflectances with preferred directions in radiative-transfer calculations, J. Quant. Spectrosc. Radiat. Transfer 53, 289-305.

Godsalve, C., 1996: Accelerating the discrete ordinates method for the solution of the radiative transfer equation for planetary atmospheres, J. Quant. Spectrosc. Radiat. Transfer 56, 609-616.

Hapke, B., 1993: **Theory of Reflectance and Emittance Spectroscopy**, Cambridge University Press, 333 pp.

Kernighan, B. and P. Plauger, 1978: **The Elements of Programming Style**, 2nd Edition, McGraw-Hill, 168 pp.

King, M.D., 1983: Number of terms required in the Fourier exapnsion of the reflection function for optically thick atmospheres, J. Quant. Spectrosc. Radiat. Transfer 30, 143-161.

King, M.D. and Harshvardhan, 1986: Comparative accuracy of selected multiple scattering approximations. J. Atmos. Sci. 43, 784-801.

Kourganoff, V., 1963: **Basic Methods in Transfer Problems**, Dover Press, New York.

Kylling, A. and K. Stamnes, 1992: Efficient yet accurate solution of the linear transport equation in the presence of internal sources: The exponential-linear-in-depth approximation. J.

Comput. Physics. 102, 265-276.

Kylling, A., K. Stamnes, and S.-C. Tsay, 1995: A reliable and efficient two-stream algorithm for radiative transfer: Documentation of accuracy in realistic layered media. J. Atmos. Chem. 21, 115-150.

Lindner, B., 1988: Ozone on Mars: The effects of clouds and airborne dust. Planet. Space Sci. 36, 125-144.

Liou, K.N., 1980: **An Introduction to Atmospheric Radiation**, Academic Press, Orlando, Florida (out of print but amazon.com will search for used copies).

Lummerzheim, D, M. Rees and H.R. Anderson, 1989: Angular dependent transport of auroral electrons in the upper atmosphere. Planet. Space Sci. 37, 109-129.

McConnell, S., 1993: **Code Complete: A Practical Handbook of Software Construction**, Microsoft Press, 857 pp.

Meador, W. and W. Weaver, 1980: Two-stream approximations to radiative transfer in planetary atmospheres: A unified description of existing methods and a new improvement. J. Atmos. Sci. 37, 630-643.

Nakajima, T. and M. Tanaka, 1986: Matrix formulations for the transfer of solar radiation in a plane-parallel atmosphere. J. Quant. Spectrosc. Radiat. Transfer 35, 13-21.

Nakajima, T. and M. Tanaka, 1988: Algorithms for radiative intensity calculations in moderately thick atmospheres using a truncation approximation. J. Quant. Spectrosc. Radiat. Transfer 40, 51-69.

Press, W., S. Teukolsky, W. Vetterling and B. Flannery, 1992: **Numerical Recipes in Fortran: The Art of Scientific Computing**, Second Edition, Cambridge Univ. Press, 963 pp.

Richiazzi, P., S. Yang, C. Gautier and D. Sowle, 1998: SBDART: A research and teaching software tool for plane-parallel radiative transfer in the Earth's atmosphere, Bull. Amer. Meteor. Soc. 79, 2101-2114.

Schulz, F., K. Stamnes and F. Weng, 1999: VDISORT: An improved and generalized discrete ordinate method for polarized (vector) radiative transfer, J. Quant. Spectrosc. Radiat. Transfer 61, 105-122.

Schulz, F. M., and K. Stamnes, Angular distribution of the Stokes vector in a plane parallel, vertically inhomogeneous medium in the vector discrete ordinate radiative transfer (VDISORT) model, J. Quant. Spectrosc. Radiat. Transfer, 65, 609-620, 2000.

Siegel, R. and J. Howell, 1992: **Thermal Radiation Heat Transfer, 3rd ed.**, Taylor & Francis, Washington DC, 1072 pp.

Stamnes, K., 1982a: On the computation of angular distributions of radiation in planetary atmospheres. J. Quant. Spectrosc. Radiat. Transfer 28, 47-51.

Stamnes, K., 1982b: Reflection and transmission by a vertically inhomogeneous planetary atmosphere. Planet. Space Sci. 30, 727-732.

Stamnes, K., 1986: The theory of multiple scattering of radiation in plane parallel atmospheres. Rev. Geophys. 24, 299-310.

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 H. Dale, 1981: A new look at the discrete ordinate method for radiative transfer calculation in anisotropically scattering atmospheres. II: Intensity computations. J. Atmos. Sci. 38, 2696-2706.

Stamnes, K., O. Lie-Svendsen and M. Rees, 1991: The linear Boltzmann equation in slab geometry: Development and verification of a reliable and efficient solution, Planet. Space Sci. 39, 1453-1463.

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.-C. Tsay, W. Wiscombe and K. Jayaweera, 1988a: Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media. Appl. Opt. 27, 2502-2509.

Stamnes, K., S.-C. Tsay and T. Nakajima, 1988b: Computation of eigenvalues and eigenvectors for discrete ordinate and matrix operator method radiative transfer, J. Quant. Spectrosc. Radiat. Transfer 39, 415-419.

Sykes, J., 1951: Approximate integration of the equation of transfer, Mon. Not. Roy. Astron. Soc. 11, 377-386.

Thomas, G. E. and K. Stamnes, 1999: **Radiative Transfer in the Atmosphere and Ocean**, Cambridge University Press.

Tsay, S.-C., K. Stamnes and K. Jayaweera, 1989: Radiative energy budget in the cloudy and hazy Arctic, J. Atmos. Sci. 46, 1002-1018.

Tsay, S.-C., K. Stamnes and K. Jayaweera, 1990: Radiative transfer in stratified atmospheres: Development and verification of a unified model, J. Quant. Spectrosc. Radiat. Transfer 43, 133-148.

Tsay, S.-C. and K. Stamnes, 1992: Ultraviolet radiation in the Arctic: The impact of potential ozone depletions and cloud effects, J. Geophy. Res. 97, 7829-7840.

Wiscombe, W., 1976: Extension of the doubling method to inhomogeneous sources, J. Quant. Spectrosc. Radiat. Transfer 16, 477-489.

Wiscombe, W., 1977: The delta-M method: Rapid yet accurate radiative flux calculations for strongly asymmetric phase functions, J. Atmos. Sci. 34, 1408-1422.

Zdunkowski, W.G., R.M. Welch, and G. Korb, 1980: An investigation of the structure of typical two-stream methods for the calculations of solar fluxes and heating rates in clouds, Contrib. Atmos. Phys. 53, 147-166.

APPENDIX A: Second-Order Intensity Corrections

The Nakajima/Tanaka (N/T) "IMS method" is an improved version of their TMS method intended to correct for errors in the aureole (near-forward-scattering) region by approximating secondary or higher orders of scattering. It is implemented in DISORT subroutine SECSCA (SECondary SCAttering).

The derivation makes use of several equations given earlier, which we repeat here for convenience with the same equation numbers, but defining a functional notation for phase function integrals which will be of great use later:

(radiative transfer equation)
$$\mu \frac{\mathrm{d}I(\tau,\mu,\phi)}{\mathrm{d}\tau} = I(\tau,\mu,\phi) - S(\tau,\mu,\phi) \tag{1}$$

(sources)
$$S(\tau,\mu,\phi) = Q^{(beam)}(\tau,\mu,\phi) + \omega(\tau)[P * I](\tau,\mu,\phi) \tag{2}$$

(notation)
$$[P * I](\tau,\mu,\phi) \equiv \frac{1}{4\pi}\int_0^{2\pi} \mathrm{d}\phi' \int_{-1}^{1} \mathrm{d}\mu'\, P(\tau,\mu,\phi;\mu',\phi')\, I(\tau,\mu',\phi')$$

$$Q^{(\mathrm{beam})}(\tau,\mu,\phi) = \frac{\omega(\tau)\, I_0}{4\pi}\, P(\tau,\mu,\phi;-\mu_0,\phi_0)\, e^{-\tau/\mu_0} \tag{3b}$$

$$P(\tau,\cos\Theta) = f\, P''(\tau,\cos\Theta) + (1-f)P'(\tau,\cos\Theta)$$

(δ–M)
$$\approx 2f\,\delta(1-\cos\Theta) + (1-f)\sum_{\ell=0}^{2N-1}(2\ell+1)g'_\ell(\tau)P_\ell(\cos\Theta) \tag{54a}$$

(notation)
$$P \equiv \text{exact phase function}$$
$$P' = \delta - \mathrm{M}\ \text{ phase function}$$

(δ–M transformations)
$$d\tau' = (1-\omega f)\,d\tau$$
$$\omega' = \frac{1-f}{1-\omega f}\,\omega \tag{54d}$$

(single-scatt eqn)
$$\pm\mu \frac{\mathrm{d}I_{ss}(\tau,\pm\mu,\phi)}{\mathrm{d}\tau} = I_{ss}(\tau,\pm\mu,\phi) - \frac{I_0}{4\pi}\,\omega(\tau)\,P(\tau,\pm\mu,\phi;-\mu_0,\phi_0)\,e^{-\tau/\mu_0} \tag{64}$$

$$\Im[\text{Phase function, optical depth}] \equiv \text{single scattering solution } I_{ss}$$

$$I'_{ss} \equiv \Im[\omega' P', \tau']$$

$$I_{ss}^{\text{no } \delta-\text{M}} \equiv \Im\left[\frac{\omega P}{1-\omega f}, \tau'\right]$$

(Nakajima/Tanaka method) $$I_{\text{TMS}} = I' + I_{ss}^{\text{no } \delta-\text{M}} - I'_{ss} \qquad (67, 68a)$$

Let the remaining error in the intensity field be denoted by

$$\Delta I_{IMS} \equiv I_{TMS} - I_{true} \qquad (A.1)$$

where all intensity variables implicitly have argument list $(\tau,-\mu,\phi)$ from now on, with the $-\mu$ indicating that we correct only downward not upward intensities. N/T derive an equation for ΔI_{IMS}. First differentiate both sides of the above definition for ΔI_{IMS} and use the δ–M scaling relations (54d) and the definition of I_{TMS} (67,68a) to obtain

$$\begin{aligned} -\mu \frac{\mathrm{d}\{\Delta I_{IMS}\}}{\mathrm{d}\tau} &= -\mu\left[\frac{\mathrm{d}I_{TMS}}{\mathrm{d}\tau} - \frac{\mathrm{d}I_{true}}{\mathrm{d}\tau}\right] \\ &= \mu \frac{\mathrm{d}I_{true}}{\mathrm{d}\tau} - \mu[1-\omega(\tau) f(\tau)] \frac{\mathrm{d}}{\mathrm{d}\tau'}\left\{I' + I_{ss}^{\text{no } \delta-\text{M}} - I'_{ss}\right\} \end{aligned} \qquad (A.2)$$

Substituting (1-3) to get rid of I_{true}, then using the δ–M approximation for the phase function (54a) and the equation for the single-scattering solution (64), and finally algebraically rearranging, one finds

$$-\mu \frac{\mathrm{d}\{\Delta I_{IMS}\}}{\mathrm{d}\tau} = \Delta I_{IMS} - \left[Q_1 + Q_2 + Q_3\right] - \omega(\tau)\left[P * \Delta I_{IMS}\right](\tau, -\mu, \phi) \qquad (A.3)$$

where the Q's, like the intensity variables, have argument list $(\tau,-\mu,\phi)$. The Q's are specified in terms of a "δ–M multiple-scattered intensity"

$$I'_{mult} = I' - I'_{ss} \qquad (A.4)$$

and the "δ–M residual phase function" P'' derived from (54a):

$$P''(\tau,\cos\Theta)=\frac{1}{f}\{P(\tau,\cos\Theta)-(1-f)P'(\tau,\cos\Theta)\} \tag{A.5}$$

Using these definitions, the Q's are:

$$\begin{aligned}
Q_1 &= \omega(\tau)f(\tau)\left[I'_{mult}-\left[P''*I'_{mult}\right](\tau,-\mu,\phi)\right]\\
Q_2 &= \omega(\tau)f(\tau)\left[I_{ss}^{\text{no }\delta-\text{M}}-\left[P''*I_{ss}^{\text{no }\delta-\text{M}}\right](\tau,-\mu,\phi)\right]\\
Q_3 &= \frac{I_0\omega(\tau)}{4\pi}P(\tau,-\mu,\phi;-\mu_0,\phi_0)\left[e^{-\tau'/\mu_0}-e^{-\tau/\mu_0}\right] \;-\\
&\qquad \omega(\tau)[1-f(\tau)]\left[P'*\left(I_{ss}^{\text{no }\delta-\text{M}}-I'_{ss}\right)\right](\tau,-\mu,\phi)
\end{aligned} \tag{A.6}$$

Solving (A.3) directly is even more difficult than solving the original radiative transfer problem (1-3), unless appropriate approximations are made.

The approximations are as follows.

(1) Since the IMS-method correction should be concentrated around the solar aureole region ($\mu=\mu_0$), the source function Q_1 and the angular integral term in (A.3) can be neglected due to their small contribution, leading to:

$$-\mu\frac{\mathrm{d}\left\{\Delta I_{IMS}\right\}}{\mathrm{d}\tau}=\Delta I_{IMS}-\left[Q_2+Q_3\right] \tag{A.7}$$

(2) In the δ–M method, because P´´ is a delta-function,

$$\frac{1}{4\pi}\int_0^{2\pi}\mathrm{d}\phi'\int_{-1}^{1}\mathrm{d}\mu'\,P''(\tau,-\mu,\phi;\mu',\phi')\,P'(\tau,\mu',\phi';-\mu_0,\phi_0)=P'(\tau,-\mu,\phi;-\mu_0,\phi_0) \tag{A.8}$$

(3) The source term Q_2 becomes, by substituting the solutions for the two single-scattered intensities (subscript *ss*),

$$\begin{aligned}
Q_2(\tau,-\mu,\phi)\approx\frac{I_0}{4\pi}\frac{[\omega(\tau)f(\tau)]^2}{1-\omega(\tau)f(\tau)}\Big[&P''(\tau,-\mu,\phi;-\mu_0,\phi_0)-\\
&\frac{1}{4\pi}\int_0^{2\pi}\mathrm{d}\phi'\int_{-1}^{1}\mathrm{d}\mu'\,P''(\tau,-\mu,\phi;\mu',\phi')P''(\tau,\mu',\phi';-\mu_0,\phi_0)\Big]\varphi(\tau',-\mu_0,-\mu_0)
\end{aligned} \tag{A.9}$$

Here, the approximation is that the geometrical factor φ

$$\varphi(\tau,-\mu,-\mu_0)\equiv\frac{e^{-\tau/\mu}}{\mu}\int_0^\tau e^{\left(1/\mu-1/\mu_0\right)t}\,\mathrm{d}t \tag{A.10}$$

is nearly independent of μ near $\mu=\mu_0$ and can be brought outside the angular integrals by setting $\mu=\mu_0$.

(4) By using (54d), the exponential term in Q_3 can be approximated as

$$\begin{aligned} &e^{-\tau'/\mu_0}-e^{-\tau/\mu_0}=e^{-\tau'/\mu_0}\left[1-e^{-\omega f\tau/\mu_0}\right] \\ &=-e^{-\tau'/\mu_0}\sum_{n=1}^{\infty}\frac{1}{n!}\left[\frac{-\omega(\tau)f(\tau)\tau}{\mu_0}\right]^n\approx\frac{\omega(\tau)f(\tau)}{1-\omega(\tau)f(\tau)}\varphi(\tau',-\mu_0,-\mu_0) \end{aligned} \tag{A.11}$$

if only the first term (n=1) of the Taylor series expansion is considered.

(5) Using (A.11) and following the pattern of (A.9), the source term Q_3 becomes

$$\begin{aligned} Q_3(\tau,-\mu,\phi)\approx&\frac{I_0}{4\pi}\frac{\omega^2(\tau)f(\tau)}{1-\omega(\tau)f(\tau)}\Bigg[P(\tau,-\mu,\phi;-\mu_0,\phi_0)-\frac{1-f(\tau)}{4\pi}\times \\ &\int_0^{2\pi}\mathrm{d}\phi'\int_{-1}^{1}\mathrm{d}\mu'\,P'(\tau,-\mu,\phi;\mu',\phi')P''(\tau,\mu',\phi';-\mu_0,\phi_0)\Bigg]\varphi(\tau',-\mu_0,-\mu_0) \\ =&\frac{I_0}{4\pi}\frac{[\omega(\tau)f(\tau)]^2}{1-\omega(\tau)f(\tau)}P''(\tau,-\mu,\phi;-\mu_0,\phi_0)\varphi(\tau',-\mu_0,-\mu_0) \end{aligned} \tag{A.12}$$

The last approximation concerns the treatment of vertical inhomogeneity of the medium. Since the IMS correction is applied only to the transmitted intensity around the solar aureole, the contribution to secondary-scattered intensity from backward scattering is relatively small. Thus, vertically averaged optical properties of the medium should be sufficient. (A.7) then has an analytical solution of

$$\begin{aligned} \Delta I_{IMS}\approx&\frac{I_0}{4\pi}\frac{\left(\bar{\omega}\bar{f}\right)^2}{1-\bar{\omega}\bar{f}}\left[2P''(-\mu,\phi;-\mu_0,\phi_0)-P''^2(-\mu,\phi;-\mu_0,\phi_0)\right]\times \\ &\chi(\tau,-\mu,-\mu_0',-\mu_0') \end{aligned} \tag{A.13}$$

where the square of P´´ is defined as an operator:

$$P''^{2}(\tau,-\mu,\phi;-\mu_0,\phi_0) \equiv \frac{1}{4\pi}\int_0^{2\pi} \mathrm{d}\phi' \int_{-1}^{1} \mathrm{d}\mu'\, P''(\tau,-\mu,\phi;\mu',\phi')\, P''(\tau,\mu',\phi';-\mu_0,\phi_0) \qquad \text{(A.14)}$$

and the mean optical properties of the atmosphere are

$$\bar{\omega} = \sum_{n=1}^{p} \omega_n \tau_n \Big/ \sum_{n=1}^{p} \tau_n$$

$$\bar{f} = \sum_{n=1}^{p} f_n \omega_n \tau_n \Big/ \sum_{n=1}^{p} \omega_n \tau_n$$

$$P''(\cos\Theta) = \sum_{\ell=0}^{N_{\max}} (2\ell+1)\,\bar{g}_\ell\, P_\ell(\cos\Theta) \qquad \text{(A.15)}$$

$$\bar{g}_\ell = \sum_{n=1}^{p} g'_{\ell,n} \omega_n \tau_n \Big/ \sum_{n=1}^{p} f_n \omega_n \tau_n$$

$$g'_{\ell,n} \equiv \begin{cases} f_n & \ell \le 2N-1 \\ g_{\ell,n} & \ell > 2N-1 \end{cases}$$

(Note that P'' is not a delta-function, but rather the *residual phase function*, which is only *approximated* as a delta-function in the δ–M method. $N_{\max}$ is the number of terms necessary to converge the Legendre expansion of the exact phase function P.) Also, the transformation of the geometrical factor

$$\varphi(\tau',-\mu_0,-\mu_0) = \varphi(\tau,-\mu_0',-\mu_0'), \quad \mu_0' \equiv \mu_0 / (1-\bar{\omega}\bar{f})$$

is used to obtain the function χ as

$$\chi(\tau,-\mu,-\mu',-\mu'') = \frac{e^{-\tau/\mu}}{\mu\mu'} \int_0^{\tau} \mathrm{d}t\, e^{t(1/\mu-1/\mu')} \int_0^{t} e^{t'(1/\mu'-1/\mu'')}\, \mathrm{d}t' \qquad \text{(A.16)}$$

(A.16) is of course a trivial integral, but it is still necessary to distinguish five different cases, as follows (for all μ, μ´, μ´´ > 0):

$$\chi = \frac{1}{\mu\mu'}\begin{cases} \dfrac{\tau^2 e^{-\tau/\mu}}{2} & \mu' = \mu'',\ \mu = \mu' \\ \dfrac{1}{x_1}\left[\left(\tau - \frac{1}{x_1}\right)e^{-\tau/\mu'} + \dfrac{e^{-\tau/\mu}}{x_1}\right] & \mu' = \mu'',\ \mu \neq \mu' \\ \dfrac{1}{x_2}\left[\dfrac{e^{-\tau/\mu''} - e^{-\tau/\mu}}{x_2} - \tau e^{-\tau/\mu}\right] & \mu' \neq \mu'',\ \mu = \mu' \\ \dfrac{1}{x_1}\left[\dfrac{e^{-\tau/\mu'} - e^{-\tau/\mu}}{x_1} - \tau e^{-\tau/\mu}\right] & \mu' \neq \mu'',\ \mu = \mu'' \\ \dfrac{1}{x_2}\left[\dfrac{e^{-\tau/\mu''} - e^{-\tau/\mu}}{x_2} - \dfrac{e^{-\tau/\mu'} - e^{-\tau/\mu}}{x_1}\right] & \mu \neq \mu' \neq \mu'' \end{cases} \tag{A.17a}$$

where

$$x_1 = \frac{1}{\mu} - \frac{1}{\mu'}$$
$$x_2 = \frac{1}{\mu} - \frac{1}{\mu''} \tag{A.17b}$$

The branching in (A.17a) is done in XIFUNC, which is called by SECSCA.

Substituting the definition (A.15) of the residual phase function *P″* into (A.14), and expressing the phase functions in terms of the directions $(-\mu,\phi)$ and $(-\mu_0,\phi_0)$ by using the addition theorem for spherical harmonics (5b), and then carrying out the integration leads to

$$P''^2\left(\tau,-\mu,\phi;-\mu_0,\phi_0\right) = \sum_{\ell=0}^{N_{\max}} (2\ell+1)\,\bar{g}_\ell{}^2\; P_\ell(\cos\Theta) \tag{A.18}$$

where

$$\cos\Theta = \mu\,\mu_0 + \sqrt{(1-\mu^2)\,(1-\mu_0{}^2)}\,\cos\left(\phi-\phi_0\right)$$

Substituting (A.15) and (A.18) into (A.13), the IMS intensity correction term becomes

$$\Delta I_{IMS} \approx \frac{I_0}{4\pi} \frac{\left(\bar{\omega}\,\bar{f}\right)^2}{1-\bar{\omega}\,\bar{f}} \left[\sum_{\ell=0}^{N_{\max}} (2\ell+1)\left(2\bar{g}_\ell - \bar{g}_\ell{}^2\right) P_\ell(\cos\Theta)\right] \chi\left(\tau, -\mu, -\mu_0', -\mu_0'\right) \qquad \text{(A.19)}$$

APPENDIX B: DISORT Wish List

In any large effort such as DISORT, the authors always have an idea file — a "wish list" — of improvements they would like to make in the future. Unfortunately, DISORT is a volunteer effort; we have never had a dollar of explicit funding support for the DISORT activity. Thus, there is no guarantee that any of these improvements will in fact be made, nor any time schedule for making them. So, we offer this appendix both to capture these ideas before they are lost, and to encourage those with initiative and desire to try to implement these changes themselves.

B.1 Zeroing computational noise in output quantities

Sometimes computational noise (roundoff and truncation error occasionally amplified by ill-conditioning) causes output fluxes and/or intensities returned by DISORT to be small but non-zero when they should in fact be zero. Users are especially alarmed when these noisy values are negative. This was only a minor annoyance at first, especially in the test problem suite where we had to manually scan the output and judge whether each such small value was in fact merely noise or perhaps a telltale sign of a larger problem. But over the years it has grown to be a *major* annoyance both to ourselves and to users: to ourselves, because every time we change the code or run on a new computer, these noise results change, and we have to look at them all over again; and to users, because they see spurious non-unit ratios in the output from the provided test problems, instead of 1.00000. Furthermore, these noise results make automatic maintenance of the program painful, because they always show up as spurious differences in a file-compare between new and old test problem outputs.

Let us define ε=machine precision (e.g. roughly 10^{-7} for IEEE single precision) and "noise"=ratio of radiant quantity to the forcing of that quantity (beam or diffuse incidence or thermal emission). The noise results fall into three categories:

(1) $|\text{noise}| < 10\varepsilon$ (ω not too close to unity and none of the problem specifications large enough to cause major buildup of roundoff error)

(2) $10\varepsilon < |\text{noise}| < 10^3\,\varepsilon$ (ω not too close to unity but one or more of the problem specifications, say the number of layers, large enough to cause major buildup of roundoff error)

(3) $|\text{noise}| > 10^3\,\varepsilon$ ($\omega = 1$ but dithered by program to be slightly less than unity;

perhaps exacerbated by large problem specifications)

Thus, the noise can either explode upwards as ω-->1 (case 3) or just creep upwards in a relatively normal way (case 2).

Case 3 noise has a fundamentally different cause than cases 1 and 2; we suspect it is mainly due to exploding inaccuracy in the eigenvalue-eigenvector calculation as ω-->1 which could perhaps be drastically reduced by introducing special-case branches for ω near 1. However, at this point this is only speculation and the cause could lie elsewhere.

Cases 1 and 2 noise are more easily dealt with. One would calculate a "noise floor" I_{min} and then zero every DISORT output intensity such that $|I|<I_{min}$. Analogous procedures would apply to fluxes, flux divergences, and mean intensities. I_{min} would be the maximum of all the sources stimulating radiation in the medium:

$$I_{min} = \max(\text{beam source, diffuse source, Planck source})$$

B.2 Removable 0/0-type singularities

We have already described in Sec. 2.4.2 how DISORT deals with the 0/0-type singularities arising when single-scatter albedo ω=1 (by dithering). Another removable 0/0-type singularity condition arises when the beam angle coincides with one of the angles at which output intensities are desired. In this case, the user would be advised to slightly change their sun angle or their output angle so that they no longer coincide. The program handles this case using L'Hospital's Rule to get the correct limit, but it is not sophisticated and may amplify the error by not expanding to a higher level of approximation.

This type of singularity also occurs when the beam angle coincides with one of the quadrature angles, but the latter are not under direct user control, and they take such unusual values that the odds of such a coincidence are practically zero. The problem is most likely to occur when the number of streams divided by 2 is odd and cos(beam angle) = 0.5, and it can easily be corrected by changing the number of streams.

In general, it may be better to avoid requiring intensities exactly at the beam angle—neither directly backward nor directly forward. Real phase functions are most poorly known in the direct backscattering region, especially in the low order of Legendre approximation in which they are represented in DISORT, and if one is looking for the hot-spot or opposition effect such as seen

when flying over vegetation, forget it—DISORT doesn't calculate that at all. The region of direct forward scattering is also difficult for DISORT, because in order to do as well as it does at other angles it has to fiddle with the photons scattered within a few degrees of the forward direction; thus its exact forward intensity may be less accurate than at other angles. The intensity correction of Nakajima and Tanaka, however, can usually improve the results in both the forward and backward directions.

B.3 Fortran-90 implementation

Fortran-90 offers a number of new features that would simplify DISORT and make it even more robust. However, there are so many new features that it is important to prioritize them, so that the conversion to Fortran-90 can proceed in a step-by-step manner as time allows, and thus not be overwhelming. In our view, the most important conversions, after the trivial ones such as converting fixed-form (columns 6 to 72) to free-form source code are accomplished, are:

(1) conversion of the internal arrays from fixed-size to "automatic" arrays, taking their dimensions from DISORT's argument list; this eliminates both the PARAMETER's now required to dimension internal arrays and an annoying source of input error messages in CHEKIN;

(2) use of new array notation wherever possible;

(3) use of new array intrinsic functions—mainly SUM, MATMUL, and DOT_PRODUCT;

(4) packaging subroutines inside MODULEs so interface blocks are not needed;

(5) deleting the integers describing array dimensions from argument lists to DISORT subroutines; re-dimensioning those arrays as "assumed-shape" arrays;

(6) replacing R1MACH with calls to f90 intrinsic functions EPSILON, TINY, and HUGE;

B.4 Summing series by Clenshaw's recurrence

When single-scattering is treated exactly, as in the various Nakajima methods, actual values of the phase function must be calculated. In DISORT, this is done by having the user provide the Legendre coefficients so that the code can sum the Legendre polynomial series. Clenshaw's recurrence formula (Press et al., 1992, Ch. 5) is an elegant and efficient way to sum a series of functions that obey a recurrence formula, as Legendre polynomials do. Suppose the desired sum is

$$f(x) = \sum_{k=0}^{N} c_k \, F_k(x)$$

and that F_k obeys the recurrence relation

$$F_{n+1}(x) = \alpha(n,x)\, F_n(x) + \beta(n,x)\, F_{n-1}(x)$$

Define y_k by the recurrence

$$y_{N+2} = y_{N+1} = 0$$
$$y_k = \alpha(k,x) y_{k+1} + \beta(k+1,x) y_{k+2} + c_k \quad (k = N, N-1, \ldots, 1)$$

If you solve this last relation for c_k and plug the result back into the sum for $f(x)$, there is a great deal of cancellation leading to

$$f(x) = \beta(1,x)\, F_0(x)\, y_2 + F_1(x)\, y_1 + F_0(x)\, c_0$$

The advantage of this recurrence is that it is always stable (except in one rare case to be mentioned below), independent of whether the F_k recurrence is stable in the up or down direction; and since it starts with the smallest c_k (for large k) and goes backwards, it preserves the maximum amount of significance.

For the special case of Legendre polynomials, $F_k = P_k$, we have

$$\alpha(n,x) = \frac{2n+1}{n+1}\, x = (1+\gamma_n)x \quad \beta(n,x) = -\frac{n}{n+1} = -\gamma_n \quad (\gamma_n \equiv \frac{n}{n+1})$$
$$f(x) = -\tfrac{1}{2} y_2 + x y_1 + c_0$$

where in practice the γ_n would be pre-computed and stored.

The rare problem case is when the first two terms in the sum for $f(x)$ nearly cancel. In the Legendre polynomial case, a criterion for this would be:

$$\left| -\tfrac{1}{2} y_2 + x y_1 \right| < 10\, \varepsilon \left| y_2 \right| \quad (\varepsilon = \text{machine precision})$$

This occurs when the F_k are small for large k, and at the same time c_k are small for small k.

Since for phase functions the c_k typically decrease monotonically, it is unlikely that this problem case would ever be triggered; however, testing for it is a wise precaution and makes the algorithm more robust. If the problem case is ever encountered, Press et al. say that the solution is to use the Clenshaw recurrence in the upward direction. However, this involves knowing F_N and F_{N-1} which would require an up-recurrence for the F's also, and the economy of the Clenshaw method is seemingly lost.

B.5 Special case of no sources

If DISORT is unforced by any sources, the output variables would be identically zero. DISORT should test for this condition and, if true, return immediately after the output arrays are zeroed. Obviously, if only a few cases are being done manually, this is not a serious concern, but in cases where DISORT is buried inside a larger program and called many times without the input being manually monitored, it could be very wasteful indeed. Also, DISORT doesn't check if the user input is contradictory in the Planck case; that is, if PLANK=TRUE but the Planck sources are all in fact zero. Carrying along the thermal emission part of the computation is also wasteful in this case.

APPENDIX C: If Beam Source is Not Quite Parallel

The standard beam source in radiative transfer is just (3c):

$$I^{direct} = I_0 \, e^{-\tau/\mu_0} \, \delta(\mu - \mu_0) \, \delta(\phi - \phi_0) \tag{C.1}$$

but in reality all sources have a finite angular width, so that the delta-functions are really boxcar functions. The Sun has an angular width of 0.52–0.54° from the Earth, for example. Such extended sources can of course be regarded as made up of dozens of point sources at slightly different angles, and a separate radiative transfer problem can be solved for each point source. The results of these separate calculations can be averaged together (even weighting the results by limb darkening toward the edge of the source) to get a more correct answer than with the delta-function approximation.

However, this is a lot of extra work. It is natural to want to avoid it. Thus, we should ask "what error penalty is exacted if we just ignore the angular width of the source and use (C.1) with μ_0,ϕ_0 referring to the center of the source?" There are of course other small errors arising from the fundamental assumptions in DISORT—atmospheres are spherical not plane-parallel, for example, and radiation is partially polarized by scattering not unpolarized—so we want to at least be sure that errors from a non-parallel beam are no larger than these other errors, and also smaller than typical errors in intensity measurements.

While it would be hard to prove analytically, our intuition tells us that the problem is much ameliorated when there is a lot of scattering to diffuse the radiation. Scattered radiation quickly loses its memory of the beam angle and is probably much less sensitive to it than direct-beam radiation. The largest errors are probably incurred in the direct beam. (If this intuition is true, Venus solar radiances would probably suffer less error than Earth ones even though it is twice as close to the Sun, because of its complete cloud cover.)

It is relatively trivial to calculate errors in the direct beam (C.1) from an extended source, if we ignore limb darkening:

$$\varepsilon(\tau, \mu_0, \Delta\mu) \equiv 100 \left(1 - \frac{e^{-\tau/\mu_0}}{\frac{1}{2\Delta\mu} \int_{\mu_0 - \Delta\mu}^{\mu_0 + \Delta\mu} e^{-\tau/\mu} \, d\mu} \right) \tag{C.2}$$

For simplicity, this expression makes a "ring of suns" approximation rather than trying to account for the finite azimuthal size of the beam source corresponding to the factor $\delta(\phi-\phi_0)$ in (C.1). The integral in (C.2) can be written in terms of exponential integrals:

$$\int_{\mu_0-\Delta\mu}^{\mu_0+\Delta\mu} e^{-\tau/\mu}\,d\mu = (\mu_0+\Delta\mu)\,E_2\left(\frac{\tau}{\mu_0+\Delta\mu}\right) - (\mu_0-\Delta\mu)\,E_2\left(\frac{\tau}{\mu_0-\Delta\mu}\right) \qquad \text{(C.3)}$$

allowing (C.2) to be written in the form

$$\varepsilon(x\equiv\tau/\mu_0,\beta\equiv\Delta\mu/\mu_0)\equiv 100\left[1-\frac{2e^{-x}}{\left(\frac{1}{\beta}+1\right)E_2\left(\frac{x}{1+\beta}\right)-\left(\frac{1}{\beta}-1\right)E_2\left(\frac{x}{1-\beta}\right)}\right] \qquad \text{(C.4)}$$

which depends only on the two combination parameters x and β, not on three parameters separately as in (C.2).

In order to study (C.4) numerically, we need realistic ranges of x and β. x can realistically range from zero to arbitrarily large values, which can occur either because optical depth is large or the beam source is near the horizon (making μ_0 small). β, on the other hand, is more constrained, at least for beam sources that don't violate our idea of a "beam". The following table gives an idea of the range of β for beam sources with widths from 0.5° (our Sun from Earth) to 1° (our Sun from Venus) to 2° (another solar system) and for various values of $\theta_0=\cos^{-1}(\mu_0)$. β is only large for beam sources near the horizon, obviously.

θ0	0.5°	1°	1.5°	2°
89°	0.250	0.500	0.750	
88°	0.125	0.250	0.375	0.500
87°	0.083	0.167	0.250	0.333
86°	0.062	0.125	0.187	0.250
84°	0.042	0.083	0.125	0.167
82°	0.031	0.062	0.093	0.124
80°	0.025	0.050	0.074	0.099
75°	0.016	0.032	0.049	0.065
70°	0.012	0.024	0.036	0.048

Using an exponential integral routine from Numerical Recipes (Press et al., 1992), we obtained the results in Figure C.1. The conclusion to be reached from data such as that in Fig. C.1 is that the percent error ε:

(1) increases to tens of percent as $x=\tau/\mu_0$ increases past 5–10; thus large optical depth for any beam angle, or the beam close to the horizon for any optical depth, will lead to large errors;

(2) for moderate optical depths, only becomes worrisome when the beam source is close enough to the horizon to also need to worry about spherical geometry corrections (which are usually applied, however, without taking into account any corrections due to the finite width of the beam source).

Note also that the *absolute* error (compared to the magnitude of an overhead beam at the top of the medium) is quite a different story. In the cases where relative error is large (larger τ, smaller μ_0), absolute error tends to be small.

This effect may be important when the beam source remains close to the horizon for the entire day, as in Arctic and Antarctic spring and fall.

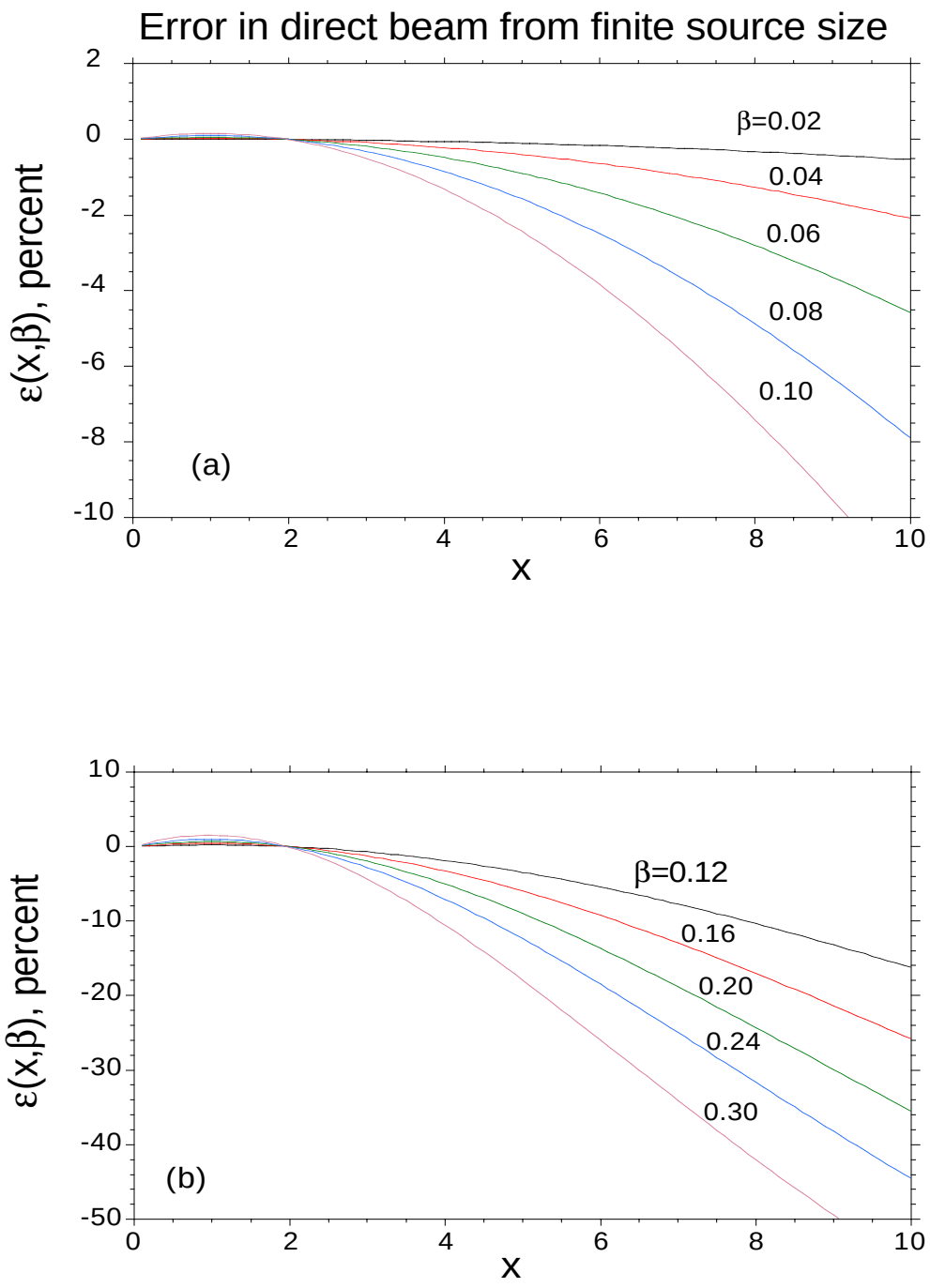

Fig C.1 Errors in approximating direct beam as a point source when in fact it is finite, from Eq. (C.4).

Appendix D: DISORT Software Design Principles

Developing a package such as DISORT is a humbling experience. Every time we thought it was finally free of errors, further testing would reveal another. What seemed only a 6-month project thus stretched into 3 years; however, we think the result is worth it. We believe this package to be freer of errors than any other similar package available today, and more full-featured to boot.

Of course, we would be foolhardy to claim that a package as large and complex as this one is entirely error-free. We have followed two cardinal principles of software development in an effort to minimize errors:

(1) offloading hard but standard computational tasks onto excellent software written by experts (in our case, the linear equation and eigenvalue/vector computations)

(2) "unit testing" many subroutines outside the program, using specially developed test drivers (for example, the Gauss quadrature and Planck function routines)

We are confident that the remaining errors are subtle and unlikely to be encountered by the average user. If you do find any errors, please report them to the authors, and we will do our best, time permitting, to find a solution.

It is very easy to introduce errors into this package. We did it many times ourselves in the course of developing it. The most seemingly innocent, casual changes are fraught with danger. After a several-year debugging process, we are not prepared to find bugs that *you* introduce. If you change the code, you are on your own.