Journal of Quantitative Spectroscopy & Radiative Transfer

www.elsevier.com/locate/jqsrt

New secondary-scattering correction in disort with increased efficiency for forward scattering

Robert Buras, Timothy Dowling, Claudia Emde

PII: S0022-4073(11)00138-5
DOI: doi:10.1016/j.jqsrt.2011.03.019
Reference: JQSRT 3758

To appear in: *Journal of Quantitative Spectroscopy & Radiative Transfer*

Received date: 1 March 2011
Revised date: 28 March 2011
Accepted date: 29 March 2011

Cite this article as: Robert Buras, Timothy Dowling and Claudia Emde, New secondary-scattering correction in disort with increased efficiency for forward scattering, *Journal of Quantitative Spectroscopy & Radiative Transfer*, doi:10.1016/j.jqsrt.2011.03.019

New secondary-scattering correction in DISORT with increased efficiency for forward scattering

Robert Buras[a,*], Timothy Dowling[b], Claudia Emde[a]

[a] Meteorologisches Institut München, Ludwig-Maximilian-Universität München, Theresienstrasse 37, 80333 München, Germany
[b] Dept. of Physics & Astronomy, University of Louisville, Kentucky 40292, USA

Abstract

We present an alternative method to calculate the directional distribution after secondary scattering of light in an atmosphere, and apply it to the correction developed by Nakajima and Tanaka [1] as implemented in the DISORT radiative transfer solver. This method employs the scattering phase functions directly, instead of expanding over their Legendre moments as in the original formulation, and hence is not compromised in cases where a prohibitive number of moments is required to maintain accuracy. The new approach is designed to be particularly efficient in the strongly forward-scattering case, which arises for example in problems involving cloud-ice or dust particles. We have implemented this in a newly rewritten C-code version of DISORT that provides additional computational efficiencies via dynamic and cache-aware memory allocation. The new version uses less memory and runs considerably faster than the original, while producing results with equal or greater accuracy.

Keywords: radiative transfer, DISORT, intensity correction, phase function, secondary scattering, ice clouds, dust

1. Introduction

Atmospheric radiative transfer has been a topic of increasing interest for decades, and a variety of computational algorithms for solving the radiative transfer equations are available [e.g. 2]. For the calculation of radiance, which is the quantity of interest when simulating remote sensing instruments, two methods dominate the field: the discrete-ordinate method (DOM) [3] and the Monte Carlo method [4–6]. The former is computationally the faster of the two, but has the disadvantage that it becomes increasingly complex when polarisation [7, 8], spherical geometry [9], or even horizontally inhomogeneous atmospheres [10, 11] are of interest. The latter handles these effects more readily, and is often used for validation of DOM results, but is generally more computationally expensive, although recent developments have enabled Monte Carlo methods to compete with discrete-ordinate methods for some applications [12–14].

The discrete-ordinate method calculates the radiance (often called the intensity) for a preset number of streams or directions, n_{str}, such that the scattering phase function (SPF) is described by appropriate Legendre coefficients [15]. It is in the nature of the DOM that it can only take into account the first $(n_{\mathrm{str}} - 1)$ Legendre coefficients, the n_{str} coefficient is then usually handled by the delta-M scaling method [16]. For those scattering directions where the radiance is sensitive to small-scale detail in the phase function, in particular for forward scattering and any special angles related to particle geometry (e.g. rainbows), errors can be significantly reduced, often below 1%, by employing an intensity correction method (ICM) [1]. This method corrects the radiances obtained by the DOM for small-scale features from single and double scattered light. However, the original ICM approach was designed with moderately formed scattering phase functions in mind, since at the time computational power was limited. Today, strongly forward-peaked SPFs, such as those that arise in the study of ice clouds [17], are of great interest, and for such extreme SPFs, Legendre expansions with tens of thousands of coefficients are necessary to obtain sufficient precision, which can render the original ICM approach computationally expensive, especially for the calculation of the double scattered light corrections.

To address this issue, we present an alternative method to calculate the intensity corrections that uses the SPFs directly as a function of scattering angle, rather than as an expansion in Legendre polynomials. In Section 2 we formulate the double-scattering problem in terms of the squared SPF expressed both ways, in Section 3 we describe the new correction algorithm, and in Section 4 we show how to optimize the scattering-angle grid on which the SPF is defined to reduce memory requirements. The method has been implemented into a new C-code version

[*] Corresponding author
Email addresses: robert.buras@lmu.de (Robert Buras), dowling@louisville.edu (Timothy Dowling), claudia.emde@lmu.de (Claudia Emde)

Preprint submitted to Journal of Quantitative Spectroscopy & Radiative Transfer *March 31, 2011*

of DISORT, `c_disort`[1], which employs dynamic memory allocation and cache-aware structures that significantly reduce run time, as described in Appendix A. Since DISORT does require $n_{\rm str}$ Legendre coefficients to work, we present a fast yet accurate method to calculate them from the SPF in Appendix C.

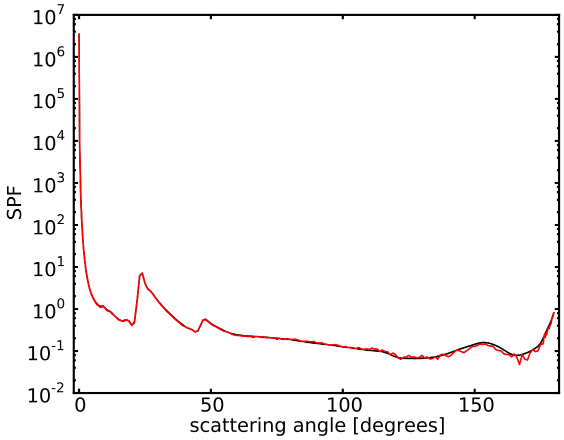

Figure 1: Scattering phase function (SPF) with strong forward peak vs. scattering angle. The black line depicts the SPF for ice crystals using the parameterization by Baum [17] with effective radius $r_{\rm eff} = 60\mu$m and wavelength $\lambda = 500$nm; the red line shows the SPF resulting when expanding the original SPF into 20000 Legendre coefficients and then transforming the coefficients back to a SPF. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2. Calculating the squared SPF

The objective is to find the directional distribution after double scattering for a given scattering phase function, $P(\mathbf{\Omega}, \mathbf{\Omega_0})$, where $\mathbf{\Omega_0}$ and $\mathbf{\Omega}$ are unit vectors in the initial and final directions, respectively. We will only consider randomly oriented and spherical particles such that $P(\mathbf{\Omega}, \mathbf{\Omega_0}) = P(\mu)$, where $\mu = \mathbf{\Omega} \cdot \mathbf{\Omega_0} = \cos\theta$, and θ is the scattering angle. The formal solution is

$$P^2(\mu) = \frac{1}{4\pi} \int_{4\pi} \mathrm{d}^2\mathbf{\Omega_1}\, P(\mathbf{\Omega} \cdot \mathbf{\Omega_1})\, P(\mathbf{\Omega_1} \cdot \mathbf{\Omega_0}), \qquad (1)$$

where the two-dimensional integration over all directions after the first scattering, $\mathbf{\Omega_1}$, takes into account all possible combinations of scattering directions.

In the original intensity-correction method, Eq. (1) is solved by expanding the SPF in terms of Legendre polynomials, $P_l(\mu)$,

$$P(\mu) = \sum_{l=0}^{m} (2l+1) g_l P_l(\mu) \qquad (2)$$

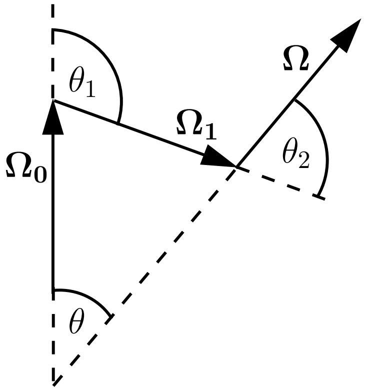

Figure 2: Illustration of the angles and directions used in the integration of Eq. (1) shown in the x-z plane. Here, $\cos\theta_1 = \mu_1$ and $\cos\theta_2 = \mu_2$. Note that in this example $\phi = \phi_1 = 0$.

where m is the truncation limit of the infinite series of Legendre coefficients, g_l, which may be computed via orthogonality such that

$$g_l = \frac{1}{2} \int_{-1}^{1} \mathrm{d}\mu\, P_l(\mu) P(\mu). \qquad (3)$$

Equation (2) is substituted into Eq. (1) and the addition theorem for spherical harmonics is applied (see e.g. Chandrasekhar [19, chapter VI, page 86]), yielding the form used by Nakajima and Tanaka [1],

$$P^2(\mu) = \sum_{l=0}^{m} (2l+1) g_l^2 P_l(\mu)\,. \qquad (4)$$

Figure 1 illustrates the strong forward peak typically associated with ice crystals. For such problems, to increase accuracy the truncation limit m must be taken so large that it not only renders the Legendre expansion computationally inefficient, but the recursive calculation of the Legendre polynomials becomes numerically unstable. In the example of Figure 1, the average accuracy of the Legendre expansion reaches a minimum of few % using 20000 Legendre coefficients, after which the differences between the original SPF and the SPF corresponding to the Legendre polynoms increase again.

As an alternative, here we seek to numerically integrate Eq. (1) itself in a manner that enhances the sampling of the SPF where it is needed. We start by rewriting Eq. (1) in terms of a double integration over the cosines of the two sequential scattering angles, as depicted in Fig. 2. Because Eq. (1) depends only on μ, we set $\mathbf{\Omega_0} = (0,0,1)$ and define

$$\mathbf{\Omega}(\mu, \phi) = \left(\sqrt{1-\mu^2}\cos\phi, \sqrt{1-\mu^2}\sin\phi, \mu\right)\,, \qquad (5)$$

where ϕ is an azimuthal angle, with an analogous expression for $\mathbf{\Omega_1}(\mu_1, \phi_1)$, such that Eq. (1) becomes

$$P^2(\mu) = \frac{1}{4\pi} \int_{-1}^{1} \mathrm{d}\mu_1 \int_{0}^{2\pi} \mathrm{d}\phi_1 P(\mu_2)\, P(\mu_1)\,. \qquad (6)$$

[1] The source code is freely available as part of the radiative transfer package libRadtran [18] at *www.libradtran.org*.

The explicit dependence of μ_2 on the other angles may be derived via trigonometry and substitution of $\phi_1 - \phi$ with ϕ_1,

$$\mu_2 = \mu\mu_1 + \sqrt{1-\mu^2}\sqrt{1-\mu_1^2}\cos\phi_1. \tag{7}$$

To optimize the sampling points for the integration, we employ the cumulative distribution function of the SPF,

$$F(\mu) \equiv \int_{-1}^{\mu} \mathrm{d}\hat{\mu}\, P(\hat{\mu})\,; \quad \mathrm{d}F_1 \equiv P(\mu_1)\,\mathrm{d}\mu_1\,, \tag{8}$$

such that Eq. (6) becomes

$$P^2(\mu) = \frac{2}{4\pi}\int_0^2 \mathrm{d}F_1 \int_0^{\pi} \mathrm{d}\phi_1\, P(\mu_2)\ , \tag{9}$$

where use has been made of the symmetry about π of the $\mathrm{d}\phi_1$ integration, $\mu_2(\mu,\mu_1,\phi_1) = \mu_2(\mu,\mu_1,2\pi-\phi_1)$. By sampling with equidistant $\mathrm{d}F_1$ intervals instead of $\mathrm{d}\mu_1$ intervals, the resolution is naturally enhanced in the first forward peak in Eq. (6), $P(\mu_1)$. This achieves the first half of our goal.

What remains is to simultaneously resolve the second forward peak in an efficient manner. To accomplish this, we use the following symmetry: In Eq. (1), interchanging μ_1 and μ_2 leads to the same contribution to the integral. Therefore, we can drop all parts of the integral where $\mu_2 > \mu_1$ and multiply the remaining integral by a factor of 2, such that Eq. (9) becomes

$$P^2(\mu) = 2\frac{2}{4\pi}\int_0^2 \mathrm{d}F_1 \int_0^{\pi} \mathrm{d}\phi_1\, P(\mu_2)\,\Xi(\mu_1-\mu_2). \tag{10}$$

where $\Xi(x)$ is the Heaviside step function, equal to 1 for $x > 0$ and 0 for $x < 0$. This integral is well sampled for phase functions that have a strong forward peak, because the sampling point density is high where $P(\mu_1)$ is large, i.e. in the forward peak, and $P(\mu_2)$ cannot be much larger than $P(\mu_1)$ because $\mu_2 < \mu_1$ (see also the SPF in e.g. Fig. 1).

Finally, the $\mathrm{d}\phi$ integration can be performed analytically, as is shown in Appendix B. The integration over $\mathrm{d}F_1$ is performed using the trapezoidal-rule integration.

3. Intensity correction in DOMs

The above method was used to alter the original intensity correction [1] as implemented in the discrete-ordinate solver DISORT [3]. Defining the residual phase function, $P''(\mu)$, to be the SPF minus the delta-M scaled SPF, the intensity correction is proportional to [3, A.13]

$$\Delta I_{\mathrm{ICM}} \propto 2P''(\mu) - P''^2(\mu)\,. \tag{11}$$

The original ICM requires calculation of P'' and P''^2 using the Legendre moments. Here, P'' is calculated from the SPF minus the delta-M scaled SPF, and P''^2 is calculated from P'' using the method described in Section 2, with the following consideration.

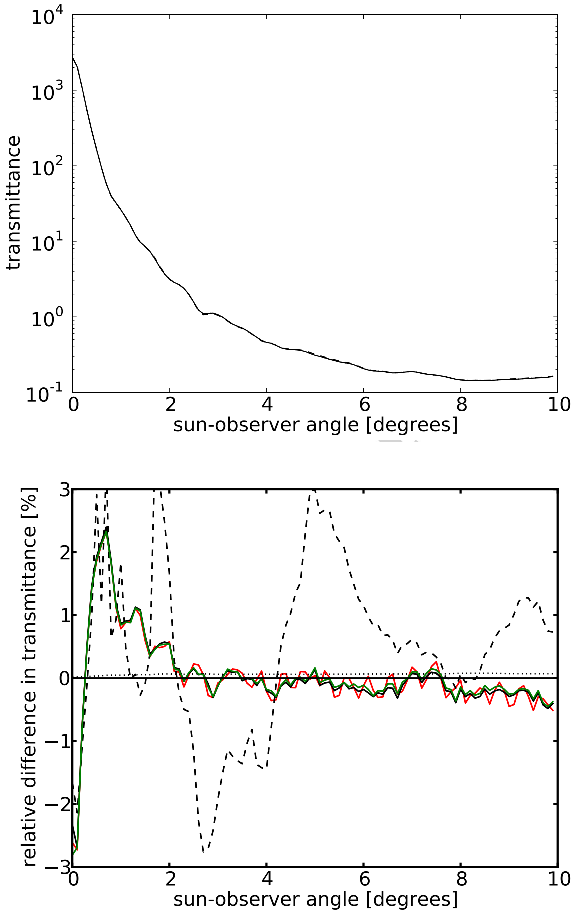

Figure 3: Radiance calculations as a function of viewing angle with respect to the sun; the sun's zenith angle is 30°. The upper panel shows transmittance, defined as radiance divided by extraterrestrial irradiance, for the original and new intensity-correction methods (ICMs), and for Monte Carlo MYSTIC calculations. The results are practically indistinguishable. The lower panel compares various DISORT calculations, using Monte Carlo MYSTIC output as the baseline. The statistical error for the MYSTIC calculation is shown as the dotted-black line, with maximum value 0.07%. The red line is for the original ICM, the solid-black line is for the new ICM with moderate resolution ($N_F = 100$), and the dashed-black and green lines are for low and high resolutions ($N_F = 40$ and $N_F = 250$), respectively. Note that the double-scattering correction as implemented by Nakajima and Tanaka is only performed for sun-observer angles less than 10°. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Note that P'' often has negative values, since it is the residual SPF, such that the cumulative distribution F from Eq. (8) is no longer monotonic. For this reason, we choose the sampling points according to the cumulative distribution of the *absolute* value of P'', F_{abs}, by substituting the integrand $\mathrm{d}F_1$ with

$$\frac{2}{F_{\mathrm{abs}}(\mu=1)}\,\mathrm{sign}\,(P''(\mu))\,\mathrm{d}F_{\mathrm{abs}}, \tag{12}$$

where the first fraction is a normalization factor.

We have experimented with the original and the new treatment of the ICM for a range of setups: i) wavelength varied from 400nm to 2100nm, ii) sun zenith angle varied from 0° to 70°, iii) observation angle varied from looking directly into the sun to 10° off (the traditional range for applying the ICM), iv) water clouds with the optical depth from 0.1 to 10, and effective radius from 1μm to 25μm, v) ice clouds with optical depth from 0.01 to 2, and the effective radius from 5μm to 40μm, using the Baum parameterization [17], and vi) all ten aerosols types from the OPAC database [20] with optical depth from 0.01 to 2. We have also created scenes with mixed effective radii and extinction coefficients, and scenes with various combinations of water clouds, ice clouds and aerosols. The number of sampling points used for the $\mathrm{d}F_1$ integration in the new ICM, N_F, was set to 100, n_{str} was 16.

For an independent comparison, we ran Monte Carlo simulations for a subset of the setups with the MYSTIC code [4] using the variance reduction method VROOM [13], which is also part of *libRadtran*. The radiances were calculated using 10^7 photons each, leading to a relative statistical error of at most 0.07%.

Figure 3 shows a comparison of radiance calculations for a sample setup with DISORT using both versions of the ICM, and with MYSTIC. The DISORT calculations were performed using the new C-code version of DISORT (`c_disort`; see Appendix A), which is part of the open-source radiative transfer package *libRadtran*. The example shown consists of a parameterized [17] thin ice cloud layer with $\tau = 0.814$, $r_{\mathrm{eff}} = 20\mu$m between the altitudes 5–6 km, sun zenith angle 30°, and wavelength $\lambda = 500$nm; the radiance is evaluated at the surface. The two ICMs agree to better than 0.2%, as seen in the top panel of Fig. 3. We use the MYSTIC calcuation as the baseline in the bottom panel of Fig. 3 by differencing the MYSTIC and DISORT results. We see that the differences between DISORT and MYSTIC are larger than the differences between the two DISORT ICM methods. In fact, for the ice-cloud cases we have examined, which are the most difficult for the original method to handle, the new ICM results are never less accurate than the original. Thus, the new ICM is a valid replacement. The question then turns to computational performance.

Table 1 compares the computational times of the two different ICM algorithms run in `c_disort`. We see that the new ICM is always faster for the calculation of radiances. In particular, when many directions and/or altitudes are calculated, the speed-up can be more than a factor of eight. The computational time needed for the ICM only can even exceed a speedup factor of 50. Note that the speed-up is less pronounced for SPFs needing fewer Legendre coefficients (e.g. water clouds).

We conclude that replacing the old ICM by the new ICM using $N_F = 100$ significantly speeds up DISORT without loss of accuracy.

N_μxN_z	no ICM	original ICM	new ICM
# of directions	[s]	[s]	[s]
1x1	0.14	0.19	0.15
1x12	0.20	0.34	0.21
100x1	0.33	3.03	0.36
100x12	1.27	12.6	1.47

Table 1: Computational times for the calculation of radiances with DISORT for the simulations shown in Fig. 3. The left-most column denotes the number of radiances calculated in a single DISORT call. N_μ is the number of directions for which the radiance was calculated, N_z the number of altitudes.

4. Optimizing grid for SPF

The grid on which the SPF is defined as a function of the scattering angle can also be optimized in order to save memory consumption. This may be obtained by the following procedure: First, we calculate the SPF on a fine, equidistant angular grid θ_{fine} (e.g. $\Delta\theta_{\mathrm{fine}} = 0.1°$), for example using the Mie tool by Wiscombe [21] which is integrated in *libRadtran*. Then we start the construction of our optimized grid with two grid points, $\theta_1 = 0°$ and $\theta_n = 180°$, $n = 2$. Then we iteratively add intermediate grid points using the following procedure: Given two adjacent grid points θ_i and θ_{i+1}, $i \in [1, n-1]$, we calculate the error we would obtain in the SPF for each θ_{fine} if we would interpolate linearly between them,

$$\Delta P(\theta_{\mathrm{fine}}) = \left| P(\theta_{\mathrm{fine}}) - \tilde{P} \right| , \tag{13}$$

where

$$\tilde{P} \equiv P(\theta_i) + \frac{\theta_{\mathrm{fine}} - \theta_i}{\theta_{i+1} - \theta_i} \left(P(\theta_{i+1}) - P(\theta_i) \right) , \tag{14}$$

for all $\theta_i < \theta_{\mathrm{fine}} < \theta_{i+1}$. We choose the θ_{fine} for which $\Delta P(\theta_{\mathrm{fine}})$ is maximal and add this point to the optimized grid. The procedure is repeated iteratively until the maximum $\Delta P(\theta_{\mathrm{fine}})$ is smaller than a critical value, say 1%. If it happens that the final resolution of the optimized grid is, in places, equal to the fine grid, then a refinement is performed for these regions, i.e. the procedure is continued using a finer initial grid.

Figure 4 shows the first 25 grid points chosen by the algorithm for two examples of SPFs at $\lambda = 500$nm, for a Mie calculation for water droplets with $r_{\mathrm{eff}} = 10\mu$m and a gamma size distribution with an effective variance of 0.1, and for ice crystals with an effective radius $r_{\mathrm{eff}} = 60\mu$m using the parameterization by Baum [17]. For the Mie example the angular resolution of the initial grid is set to $\Delta\theta = 0.01°$, corresponding to 18000 grid points, and the Baum parameterization provides the SPF on 498 grid points with varying angular resolution, the highest resolution being in the forward direction with $\Delta\theta = 0.01°$.

The optimized grid for the water droplets has 147 grid points; the one for the ice crystals has 189. Comparing the former with the number of Legendre coefficients needed to describe the SPF properly, 827, we see that we can

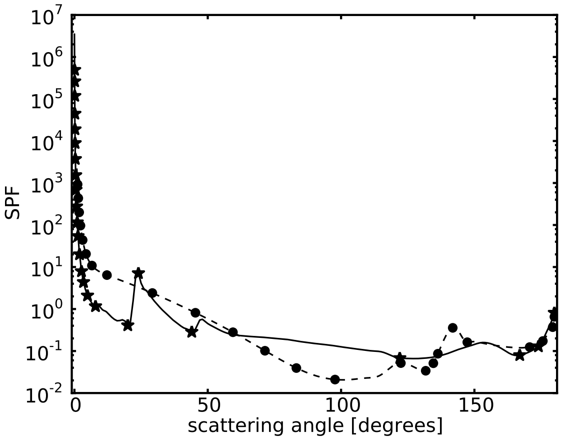

Figure 4: Example of optimized grids for water droplets (dashed curve) and ice crystals (solid curve) parameterized by [17]. The figure shows the SPF as a function of the scattering angle. The first 25 grid points chosen by the algorithm described in Section 4 are depicted by asterixes and circles for ice crystals and water droplets, respectively (the rest are omitted for clarity).

save memory by saving the SPF as a function of scattering angle instead of saving the Legendre coefficients. This is even the case if we additionally save the first 128 Legendre coefficients, which are needed by DISORT if one wants to calculate with 128 streams. For water droplets with larger effective radius the memory reduction is even more pronounced.

5. Conclusions

We have described an alternate way to calculate the double scattering phase function (SPF) introduced by Nakajima and Tanaka [1], which uses the SPF as a function of the scattering angle, instead of the Legendre coefficients. With this new method, the calculation of the intensity correction for radiances in discrete-ordinate radiative transfer codes is performed considerably faster, especially in the presence of strongly forward-scattering hydrometeors (ice crystals, large water droplets) and dust, without loss of accuracy. In particular for large ice crystals, the original method for intensity correction often cannot produce correct results because in practice the Legendre expansion of the phase function does not converge. We have also demonstrated how to significantly reduce the memory and disk space requirements needed for the SPFs. For ice clouds, the memory reduction can be up to a factor of 22, and disk space is reduced by a factor of 8 in the solar spectrum. The simulations performed in this paper were done using the new C-code version of DISORT 2.1, `c_disort`, which has been written to initialize and use memory significantly more efficiently than the original Fortran version, as described in Appendix A.

6. Acknowledgements

The authors thank Franziska Schnell, Arve Kylling, Tommy Greathouse, Tim Mattox, and Joseph Harrington for useful discussions. TD acknowledges NASA grant NNX08AE64G.

Appendix A. C_DISORT

DISORT 2.1 is a widely used discrete-ordinate radiative-transfer solver written in Fortran 77 [3]. In Summer 2010, T. Dowling rewrote the code, including its LINPACK components, plus the two-stream option added by A. Kylling, as a 100% C, double precision package called `c_disort`. This appendix provides a brief overview, complete details may be found at *www.libradtran.org*.

Structurally, the biggest break from the Fortran DISORT 2.1 code is the replacement of static memory with dynamic memory throughout `c_disort`. The Fortran version is a mix of mostly single precision with some double precision for critical linear-algebra operations, whereas the C version is 100% double precision. Significantly, occasional erroneous numerical spikes seen in the Fortran version are absent in the C version, which we attribute to the uniform double precision of the latter.

Most Fortran "`GO TO`" jumps are eliminated, some by replacing with logically equivalent `while` or `break` statements, and others by appropriate use of `if else` statements (a few C `goto` calls remain in the LINPACK components in `c_disort`). Cosmetic changes include replacing the 6-letter names in the Fortran version with more readable names, for example

`secsca()` → `c_secondary_scat()`
`terpev()` → `c_interp_eigenvec()`

The consistent use of the `c_` prefix is adopted to prevent name collisions with other *libradtran* packages. All the comments from the Fortran version are preserved, and additional comments are added to improve source-code clarity.

In terms of runtime efficiency, the most helpful change turns out to be re-writing how arrays are initialized. In the Fortran version, all of the multidimensional array elements are set to zero using nested loops, whether or not this is necessary. In the C version, dynamic memory is allocated with zeros using `calloc()`, which is efficient because modern operating systems give idle cpu cores the housekeeping task of zeroing out pages of unused memory, yielding a "poor man's parallel" advantage. Moreover, only those arrays that require it are subsequently zeroed after being allocated, and fast `memset()` calls are used rather than nested loops wherever possible.

The input and output variables are organized into structures, as mentioned in the previous section. The Fortran version of DISORT is known for its long argument list. In contrast, the C version passes one input and one output structure pointer, which streamlines argument-reference overhead and also makes the code easier to read. In addition, internal arrays are also organized into arrays of structures where feasible. The result is that the data values are stored in a striped manner, which improves cache locality. In contrast, the Fortran version stores each array separately, and in a padded manner, which leads to cache misses and reduced performance. The key paradigm shift being addressed here is the recognition that the modern cpu core is starved for data (because cpu speed has outstripped memory recall speed).

As a performance benchmark we used `disotest`, the nearly exhaustive suite of example test cases that is delivered with DISORT. The new C version runs `disotest` 12 times faster than the Fortran version. This is in addition to the speed-ups mentioned in the body of the paper, which only compare different ICM algorithms running the C version.

Finally, an advantage of the C version in terms of 'customer experience' is that its dynamic memory allocation ensures that the end user cannot configure memory in a wasteful manner, which experience shows occurs frequently with the Fortran version. The speed-up is less dramatic if the static array sizes in the Fortran version are trimmed to be as small as possible for a given job (which requires recompiling), but if the Fortran arrays are set too large, we have timed the Fortran version in field applications to be running over twenty times slower than the C version.

Appendix B. Analytic dϕ integration

We want to find an analytic solution to the integral

$$\int_0^\pi \mathrm{d}\phi_1 P(\mu_2)\, \Xi(\mu_1-\mu_2)\,. \tag{B.1}$$

First, we substitute the integration variable ϕ_1 by μ_2:

$$\begin{aligned} \mathrm{d}\phi_1 &= \left(\frac{\mathrm{d}\mu_2}{\mathrm{d}\phi_1}\right)^{-1} \mathrm{d}\mu_2 \\ &= \left(-\sqrt{1-\mu^2}\sqrt{1-\mu_1^2}\sin\phi\right)^{-1} \mathrm{d}\mu_2\,. \end{aligned} \tag{B.2}$$

Using Eq. (7) to write $\sin\phi$ as a function of μ, μ_1 and μ_2, and inserting Eq. (B.2) into Eq. (B.1), we obtain

$$\begin{aligned} &\int_0^\pi \mathrm{d}\phi_1 P(\mu_2)\, \Xi(\mu_1-\mu_2) \\ &= \int_{\mu_{\min}}^{\mu_{\max}} \mathrm{d}\mu_2 \frac{P(\mu_2)}{\sqrt{1-\mu^2-\mu_1^2-\mu_2^2+2\mu\mu_1\mu_2}}, \end{aligned} \tag{B.3}$$

where

$$\begin{aligned} \mu_{\min} &= \mu_2(\phi_1=\pi) \\ \mu_{\max} &= \min\left(\mu_2(\phi_1=0),\ \mu_1\right) \end{aligned} \tag{B.4}$$

can be obtained using Eq. (7). Between two grid points on which the SPF is defined, μ_i and μ_{i+1}, we can linearize the SPF in μ,

$$P(\mu) = C_i + D_i\mu; \qquad D_i = \frac{P(\mu_{i+1})-P(\mu_i)}{\mu_{i+1}-\mu_i}; \qquad C_i = P(\mu_i) - D_i\mu_i\,. \tag{B.5}$$

Inserting this into Eq. (B.3) we can solve the integral analytically between any two grid points. The solution is

$$\begin{aligned} &\int \mathrm{d}\mu_2 \frac{P(\mu_2)}{\sqrt{1-\mu^2-\mu_1^2-\mu_2^2+2\mu\mu_1\mu_2}} \\ &= -D_i\sqrt{1-\mu^2-\mu_1^2-\mu_2^2+2\mu\mu_1\mu_2} \\ &\quad -(D_i\mu\mu_1+C_i)\arcsin\frac{-\mu_2+\mu\mu_1}{\sqrt{1-\mu^2}\sqrt{1-\mu_1^2}}\,. \end{aligned} \tag{B.6}$$

For the special cases $\mu_2(\phi_1=0)$ and $\mu_2(\phi_1=\pi)$ this term simplifies to $+(D_i\mu\mu_1+C_i)\pi/2$ and $-(D_i\mu\mu_1+C_i)\pi/2$, respectively.

Appendix C. Fast derivation of Legendre coefficients

Between each pair of neighboring sampling points of the SPF we interpolate the SPF linearly, as in Eq. (B.5). Then the formula for calculating the Legendre coefficient becomes

$$g_l = \frac{1}{2}\sum_i \int_{\mu_i}^{\mu_{i+1}} (C_i + D_i\mu) P_l(\mu)\mathrm{d}\mu \tag{C.1}$$

This formula contains two integrals, the solutions of which are:

$$\int P_l(\mu)\mathrm{d}\mu = \frac{1}{l}\left(P_{l+1}(\mu) - \mu P_l(\mu)\right) \tag{C.2}$$

and

$$\int \mu P_l(\mu)\mathrm{d}\mu = \frac{1}{l-1}\left(\frac{l+2}{l+1}\mu P_{l+1}(\mu) - \mu^2 P_l(\mu) - \frac{1}{l+1}P_{l+2}(\mu)\right). \tag{C.3}$$

Here we show the derivation of Eqs. (C.2) and (C.3).

We start with Rodrigues' formula for the Legendre polynomial, and perform some transformations:

$$\begin{aligned} P_{l+1}(x) &= \frac{1}{2^{l+1}(l+1)!}\frac{\mathrm{d}^{l+1}}{\mathrm{d}x^{l+1}}\left[(x^2-1)^{l+1}\right] \\ &= \frac{1}{2^{l+1}(l+1)!}\frac{\mathrm{d}^{l}}{\mathrm{d}x^{l}}\left[(x^2-1)^{l}(l+1)2x\right] \\ &= \frac{2(l+1)}{2^{l+1}(l+1)!}\frac{\mathrm{d}^{l}}{\mathrm{d}x^{l}}\left[(x^2-1)^{l}x\right] \\ &= \frac{1}{2^l l!}\left\{x\frac{\mathrm{d}^{l}}{\mathrm{d}x^{l}}\left[(x^2-1)^{l}\right] + l\frac{\mathrm{d}^{l-1}}{\mathrm{d}x^{l-1}}\left[(x^2-1)^{l}\right]\right\} \\ &= xP_l(x) + l\int_{x_0}^{x} P_l(x')\mathrm{d}x'. \end{aligned} \tag{C.4}$$

The first and last part of Eq. (C.4) can be rewritten as

$$\int_{x_0}^{x} P_l(x')\mathrm{d}x' = \frac{1}{l}\left(P_{l+1}(x) - xP_l(x)\right) \qquad \text{(C.5)}$$

This is the solution to Eq. (C.2); x_0 drops out when the integration borders are defined.

The integral Eq. (C.3) is obtained as follows (the first step is partial integration):

$$\begin{aligned}
&\int_{x_1}^{x} x' P_l(x')\mathrm{d}x \\
= \;& x\int_{x_1}^{x} P_l(x')\mathrm{d}x' - \int_{x_1}^{x}\left[\int_{x_1}^{x'} P_l(x'')\mathrm{d}x''\right]\mathrm{d}x' \\
= \;& x\int_{x_1}^{x} P_l(x')\mathrm{d}x' - \int_{x_1}^{x}\left[\frac{P_{l+1}(x') - x'P_l(x')}{l}\right. \\
&\qquad\qquad\qquad \left. - \frac{P_{l+1}(x_1) - x_1P_l(x_1)}{l}\right]\mathrm{d}x' \\
= \;& x\int_{x_1}^{x} P_l(x')\mathrm{d}x' - \frac{1}{l}\int_{x_1}^{x} P_{l+1}(x')\mathrm{d}x' \\
& + \frac{1}{l}\int_{x_1}^{x} x'P_l(x')\mathrm{d}x \\
& + \left[\frac{P_{l+1}(x_1) - x_1P_l(x_1)}{l}\right](x - x_1) \qquad \text{(C.6)}
\end{aligned}$$

The third term in the last equation needs to be moved to the left side of the equation. Then, using Eq. (C.5) in the second step, we obtain

$$\begin{aligned}
&\left(1 - \frac{1}{l}\right)\int_{x_1}^{x} x'P_l(x')\mathrm{d}x \\
= \;& x\int_{x_1}^{x} P_l(x')\mathrm{d}x' - \frac{1}{l}\int_{x_1}^{x} P_{l+1}(x')\mathrm{d}x' \\
& + \left[\frac{P_{l+1}(x_1) - x_1P_l(x_1)}{l}\right](x - x_1) \\
= \;& x\frac{P_{l+1}(x) - xP_l(x)}{l} - x\frac{P_{l+1}(x_1) - x_1P_l(x_1)}{l} \\
& - \frac{P_{l+2}(x) - xP_{l+1}(x)}{l(l+1)} + \frac{P_{l+2}(x_1) - x_1P_{l+1}(x_1)}{l(l+1)} \\
& + \left[\frac{P_{l+1}(x_1) - x_1P_l(x_1)}{l}\right](x - x_1) \qquad \text{(C.7)}
\end{aligned}$$

This results in

$$\begin{aligned}
&\int_{x_1}^{x_2} x'P_l(x')\mathrm{d}x = \frac{1}{l-1} \qquad \text{(C.8)} \\
&\left[\frac{l+2}{l+1}xP_{l+1}(x) - x^2P_l(x) - \frac{1}{l+1}P_{l+2}(x)\right]_{x_1}^{x_2},
\end{aligned}$$

which is Eq. (C.3).

References

[1] T. Nakajima, M. Tanaka, Algorithms for radiative intensity calculations in moderately thick atmospheres using a truncation approximation, J. Quant. Spectrosc. Radiat. Transfer 40 (1988) 51–69.

[2] J. Lenoble (Ed.), Radiative Transfer in Scattering and Absorbing Atmospheres: Standard Computational Procedures, Deepak, 1988.

[3] K. Stamnes, S.-C. Tsay, W. Wiscombe, I. Laszlo, DISORT, a General-Purpose Fortran Program for Discrete-Ordinate-Method Radiative Transfer in Scattering and Emitting Layered Media: Documentation of Methodology, Tech. rep., Dept. of Physics and Engineering Physics, Stevens Institute of Technology, Hoboken, NJ 07030 (2000).

[4] B. Mayer, Radiative transfer in the cloudy atmosphere, Eur. Phys. J. Conferences 1 (2009) 75–99.

[5] G. I. Marchuk, G. A. Mikhailov, M. A. Nazaraliev, The Monte Carlo methods in atmospheric optics, Springer Series in Optical Sciences, Berlin: Springer, 1980, 1980.

[6] A. Marshak, A. Davis, 3D Radiative Transfer in Cloudy Atmospheres, Springer, 2005, iSBN-13 978-3-540-23958-1.

[7] V. V. Rozanov, A. A. Kokhanovsky, The solution of the vector radiative transfer equation using the discrete ordinates technique: Selected applications, Atmos. Res. 79 (2006) 241–265.

[8] Y. Ota, A. Higurashi, T. Nakajima, T. Yokota, Matrix formulations of radiative transfer including the polarization effect in a coupled atmosphere-ocean system, J. Quant. Spectrosc. Radiat. Transfer 111 (2010) 878–894.

[9] A. Rozanov, V. Rozanov, M. Buchwitz, A. Kokhanovsky, J. P. Burrows, SCIATRAN 2.0 A new radiative transfer model for geophysical applications in the 175 2400 nm spectral region, ASR 36 (2005) 1015–1019.

[10] C. Emde, S. A. Buehler, C. Davis, P. Eriksson, T. R. Sreerekha, C. Teichmann, A polarized discrete ordinate scattering model for simulations of limb and nadir long-wave measurements in 1-d/3-d spherical atmospheres, J. Geophys. Res. D: Atmospheres 109 (24) (2004) 1–20.

[11] K. F. Evans, The spherical harmonics discrete ordinate method for three–dimensional atmospheric radiative transfer, J. Atmos. Sci. 55 (1998) 429–446.

[12] C. Emde, B. Mayer, Simulation of solar radiation during a total solar eclipse: A challenge for radiative transfer, Atmos. Chem. Phys. 7 (2007) 2259–2270.

[13] R. Buras, B. Mayer, Efficient unbiased variance reduction techniques for monte carlo simulations of radiative transfer in cloudy atmospheres: The solution, J. Quant. Spectrosc. Radiat. Transfer 112 (3) (2011) 434–447.

[14] C. Emde, R. Buras, B. Mayer, ALIS: An efficient method to compute high spectral resolution polarized solar radiances using the Monte Carlo approach, J. Quant. Spectrosc. Radiat. Transfer, in press.

[15] K. Stamnes, On the computation of angular distributions of radiation in planetary atmospheres, J. Quant. Spectrosc. Radiat. Transfer 28 (1) (1982) 47–51.

[16] W. J. Wiscombe, The delta-M method: Rapid yet accurate radiative flux calculations for strongly asymmetric phase functions, J. Atmos. Sci. 34 (1977) 1408–1421.

[17] B. Baum, A. Heymsfield, P. Yang, S. Bedka, Bulk scattering models for the remote sensing of ice clouds. part 1: Microphysical data and models, J. Appl. Meteorology 44 (2005) 1885–1895.

[18] B. Mayer, A. Kylling, Technical note: The libradtran software package for radiative transfer calclations – description and examples of use, Atmos. Chem. Phys. 5 (2005) 1855–1877.

[19] S. Chandrasekhar, Radiative Transfer, Dover, New York, 1960.

[20] M. Hess, P. Koepke, I. Schult, Optical properties of aerosols and clouds: The software package OPAC, Bull. Am. Meteorological Soc. 79 (5) (1998) 831–844.

[21] W. Wiscombe, Improved Mie scattering algorithms, Applied Optics 19 (9) (1980) 1505–1509.