

This work was written as part of one of the author's official duties as an Employee of the United States Government and is therefore a work of the United States Government. In accordance with 17 U.S.C. 105, no copyright protection is available for such works under U.S. Law.

Public Domain Mark 1.0

<https://creativecommons.org/publicdomain/mark/1.0/>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.



Matrix exponential in C/C++ version of vector radiative transfer code IPOL

Sergey Korkin^{a,b,*}, Alexei Lyapustin^b

^aUSRA GESTAR, 7178 Columbia Gateway Drive, Columbia, MD 21046, USA

^bNASA GSFC, 8800 Greenbelt Rd., Greenbelt, MD 20771, USA



ARTICLE INFO

Article history:

Received 7 January 2019

Revised 5 February 2019

Accepted 5 February 2019

Available online 7 February 2019

Keywords:

Matrix exponential

Left eigenvectors

Vector radiative transfer equation

ABSTRACT

We use only left eigenvectors to evaluate the matrix exponential in the method of discrete ordinates for the vector radiative transfer equation, which neglects circular polarization, in a plane-parallel atmosphere. This is contrary to a common practice of using the right eigenvectors to evaluate the matrix exponential combined with the left eigenvectors to avoid the inversion of the matrix of the right ones. Two numerical tests for Rayleigh and Aerosol scattering confirm our idea. For better explanation of our approach and for independent crosscheck of our results, we distribute an example in C/C++.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

In 2015, our Fortran 90/95 radiative transfer (RT) code IPOL, that simulates multiple scattering of Intensity and POLarization of the monochromatic solar radiation in a plane-parallel atmosphere, participated in a comprehensive polarized (vector) RT codes inter-comparison and confirmed high accuracy [9]. Since then, we use IPOL to account for the effect of polarization of light in the Multi-Angle Implementation of Atmospheric Correction (MAIAC) algorithm [23] and as a benchmark for our Successive ORDers RT code SORD [15]. Over time, we realized the need for further development of IPOL. This includes, but not limited to, the speed-up of the code to accelerate our research, and the translation from Fortran 90/95 into C/C++ for natural integration with the algorithm MAIAC.

Similar to RT code Pstar [26,27], IPOL simultaneously uses the method of discrete ordinates [3] and the matrix-operator method [29]. In such RT codes, evaluation of the matrix exponential is a key and the most time consuming part. For solution to that problem, the eigendecomposition is “likely to be most efficient for problems involving large matrices and repeated evaluation” of the matrix exponential [25: p. 20]. The eigendecomposition is a widely used linear algebra technique. We refer the reader to the Wolfram MathWorld’s “Eigen Decomposition”¹ or the Wikipedia’s “Eigendecomposition of a matrix”² (note different spelling) articles for in-depth reading. Later in Section 2, we use only a few relations from these articles.

position of a matrix”² (note different spelling) articles for in-depth reading. Later in Section 2, we use only a few relations from these articles.

In IPOL as a full vector (including circular polarization) discrete ordinates RT code, the eigendecomposition problem requires complex numbers [31,32,35]. However, in the Earth atmosphere, the degree of circular polarization is negligibly small [14,17: p. 483]. Modern space [4,11,20,21,24] and ground [19] passive polarimeters are not designed to measure the circular component. If it is dropped from numerical simulations, the vector RT codes deal with a 3×3 phase matrix and the eigendecomposition problem requires only real arithmetic, similar to the scalar case [32: Section 7.3.1.3, 27: p. 885]. Our experience also indicates that it is safe to skip the circular polarized light and still get accurate results for the linear polarization components [Q, U], not to mention the total intensity I, regardless the atmospheric conditions [15]. Moreover, “the order of the algebraic eigenvalue problem can be reduced by a factor of 2” [33: Abstract]. This reduction speeds up the simulations.

Naturally, we decided to skip the circular polarization while translating IPOL from Fortran into C/C++. When doing that, we noticed a way to use the eigendecomposition technique in a slightly more convenient manner, than it is usually done. This paper describes all necessary details of our approach, which seems to be new. We aim to use as few equations as possible without sacrificing clarity. To achieve this goal, we (a) developed and distribute a C/C++ source code that implements our approach, and (b) explic-

* Corresponding author.

E-mail address: sergey.v.korkin@nasa.gov (S. Korkin).

¹ <http://mathworld.wolfram.com/EigenDecomposition.html> (accessed 01/01/2019).

² https://en.wikipedia.org/wiki/Eigendecomposition_of_a_matrix (accessed 01/01/2019).

itly indicate equation numbers or, at worst, section or page number in references. The next Section 2 discusses theoretical background. In Section 3, we confirm our approach using two numerical examples and explain how these results can be reproduced with our C/C++ sample code. We conclude the paper with a summary.

2. Theoretical background

Solution to the RT equation using the method of discrete ordinates involves several main steps: expansion of the phase function or matrix over corresponding polynomials; the use of the Fourier series to decouple the dependence of the radiation field on azimuth; discretization of the scattering integral over cosine of the zenith angle μ using Gauss quadrature of an order $2N$, where N is the number of ordinates per hemisphere. For the 3 components of the Stokes vector per ordinate, these steps give a system of $6N$ differential equations over optical depth τ for each Fourier harmonic m . All systems are independent and identical. Further, we skip the Fourier index m . Numerous publications discuss these steps in details—see e.g. Lenoble [16], Thomas and Stamnes [37] and Hovenier et al. [12].

In this paper, we immediately start with integration of the system of differential equations over τ . For simplicity, we assume a homogeneous non-emitting atmosphere of the total optical depth τ_0 , and seek for a solution at $2N$ Gauss nodes. One computes the radiation field at arbitrary μ from the solution at the Gauss nodes using integration of the source function or the dummy node techniques [2]. In equations below, we use the lower case bold letters, e.g. α , for a $3N \times 3N$ matrices; “ \rightarrow ” for column vectors of $3N$ elements; and the capital bold letters, e.g. \mathbf{B} , for $6N \times 6N$ matrices. Radiation reflected from top and transmitted through bottom of the atmosphere, $\vec{\mathbf{i}}_-(\mu < 0, \tau = 0)$ and $\vec{\mathbf{i}}_+(\mu > 0, \tau = \tau_0)$ respectively, satisfy a system of equations [13: Eq. (8)]:

$$-\begin{bmatrix} \vec{\mathbf{i}}_-(0) \\ \vec{\mathbf{i}}_+(0) \end{bmatrix} + \exp(\mathbf{B}\tau_0) \begin{bmatrix} \vec{\mathbf{i}}_-(\tau_0) \\ \vec{\mathbf{i}}_+(\tau_0) \end{bmatrix} = \int_0^{\tau_0} \exp((\mathbf{B} - \mathbf{1}/\mu_0)\tau) d\tau \begin{bmatrix} \vec{\mathbf{f}}_-(\mu_0) \\ \vec{\mathbf{f}}_+(\mu_0) \end{bmatrix}. \quad (1)$$

In Eq. (1), $\vec{\mathbf{i}}_+(\mu > 0, \tau = 0)$ and $\vec{\mathbf{i}}_-(\mu < 0, \tau = \tau_0)$ are the boundary conditions; \mathbf{B} simulates multiple scattering from all Gauss directions to a given direction; $\mathbf{1}$ is a unit matrix. \mathbf{B} depends on the single scattering albedo ω_0 and a 3×3 scattering matrix expansion moments [12: Section 2.8]. Without circular polarization, \mathbf{B} has the following form [33: Section 3]

$$\mathbf{B} = \begin{bmatrix} -\alpha & \beta \\ -\beta & \alpha \end{bmatrix}. \quad (2)$$

Note, that the sign at β in our Eq. (2) differs from the one in Stamnes and Swanson (1980) due to opposite definition of the positive μ direction. In the right hand side of Eq. (1), the vectors $\vec{\mathbf{f}}_-(\mu_0)$ and $\vec{\mathbf{f}}_+(\mu_0)$ come from the single scattering of the direct solar beam; μ_0 is the cosine of the solar zenith angle.

In the method of spherical harmonics, Karp et al. [13: p.394] decomposed the matrix exponential using the right eigenvectors of the system matrix \mathbf{B} , Eq. (2):

$$\exp(\mathbf{B}\tau) = \mathbf{U}_R \exp(\mathbf{\Gamma}\tau) \mathbf{U}_R^{-1}, \quad (3)$$

where $\mathbf{\Gamma}$ is a diagonal matrix of eigenvalues, known to occur in pairs with equal magnitude and different sign [33: Eq. (12)] except for a conservative case, $\omega_0 = 1.0$, in which we are not practically interested. The columns of \mathbf{U}_R are the corresponding right eigenvectors.

Further, Karp et al. [13: Eq. (12)] applied $\mathbf{S}\mathbf{U}_R^{-1}$ to both sides of Eq. (1)

$$-\mathbf{S}\mathbf{U}_R^{-1} \begin{bmatrix} \vec{\mathbf{i}}_-(0) \\ \vec{\mathbf{i}}_+(0) \end{bmatrix} + \mathbf{H}\mathbf{U}_R^{-1} \begin{bmatrix} \vec{\mathbf{i}}_-(\tau_0) \\ \vec{\mathbf{i}}_+(\tau_0) \end{bmatrix} = \mathbf{S} \int_0^{\tau_0} \exp((\mathbf{\Gamma} - \mathbf{1}/\mu_0)\tau) d\tau \mathbf{U}_R^{-1} \begin{bmatrix} \vec{\mathbf{f}}_-(\mu_0) \\ \vec{\mathbf{f}}_+(\mu_0) \end{bmatrix}, \quad (4)$$

where \mathbf{S} is a diagonal scaling transformation matrix

$$\mathbf{S} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \exp(-\gamma\tau) \end{bmatrix} \quad (5)$$

that removes exponents with positive eigenvalues γ , and $\mathbf{H} = \mathbf{S} \exp(\mathbf{\Gamma}\tau_0)$. A particular form of \mathbf{S} differs, depending on the sequence of eigenvalues. These same steps are used in the methods of spherical harmonics [13: Eq. (8), 22: Eq. (6.15)] and discrete ordinates [1: Eqs. (7)–(9)].

The explicit matrix inversion, \mathbf{U}_R^{-1} , is avoided by using the transposed matrix of left eigenvectors, \mathbf{U}_L^T [38: Eqs. (A3) and (A5), 27: Eq. (65), 8: Eq. (39)], and the following relations known from linear algebra (e.g., see the Wolfram MathWorld’s “Eigenvector”³)

$$\mathbf{U}_R^{-1} = \mathbf{U}_L^T, \quad \mathbf{\Gamma}_R = \mathbf{\Gamma}_L = \mathbf{\Gamma}, \quad \mathbf{U}_L(\mathbf{B}) = \mathbf{U}_R(\mathbf{B}^T). \quad (6)$$

In Eq. (6), $\mathbf{U}_L(\mathbf{B})$ means “left eigenvalues of matrix \mathbf{B} ”. Efremenko et al. [7: Abstract] reported that the use of the left eigenvectors instead of the matrix inversion accelerates computations by about 15%.

We believe, the use of both left and right eigenvectors [38: Eq. (A3)] for the scaling transformation in the form of Karp et al. [13], Eqs. (3)–(4), is inconvenient for the RT code development, debug, and support. Extra steps are required for their computation and normalization [5: p. 149, 6: p. 163, 8: Eqs. (32)–(33)]. Besides, a slow square root operation is involved (although, it is of course not a bottleneck in RT codes). We also believe the RT theory is complicated enough—the use of both right and left eigenvectors adds more complexity.

Instead of using both left and right eigenvectors, we can use only left eigenvectors and the following representation of the system matrix \mathbf{B} in Eq. (1)

$$\exp(\mathbf{B}\tau) = \mathbf{U}_L^{-1} \exp(\mathbf{\Gamma}\tau) \mathbf{U}_L. \quad (7)$$

Note, that the left eigenvectors are the row-vectors by definition. The scaling transformation $\mathbf{S}\mathbf{U}_L$ yields for Eq. (1)

$$-\mathbf{S}\mathbf{U}_L \begin{bmatrix} \vec{\mathbf{i}}_-(0) \\ \vec{\mathbf{i}}_+(0) \end{bmatrix} + \mathbf{H}\mathbf{U}_L \begin{bmatrix} \vec{\mathbf{i}}_-(\tau_0) \\ \vec{\mathbf{i}}_+(\tau_0) \end{bmatrix} = \mathbf{S} \int_0^{\tau_0} \exp((\mathbf{\Gamma} - \mathbf{1}/\mu_0)\tau) d\tau \mathbf{U}_L \begin{bmatrix} \vec{\mathbf{f}}_-(\mu_0) \\ \vec{\mathbf{f}}_+(\mu_0) \end{bmatrix}. \quad (8)$$

Eq. (8) does not use the inverse matrix and the right eigenvectors at all and normalization of \mathbf{U}_L is not important. Given that the transpose of the system matrix does not change its symmetry

$$\mathbf{B}^T = \begin{bmatrix} -\alpha^T & -\beta^T \\ \beta^T & \alpha^T \end{bmatrix}, \quad (9)$$

it is possible to efficiently solve the eigenproblem following Stamnes and Swanson [33: Section 3]. In particular, the matrix of the left eigenvectors is known to be symmetric [8: Eq. (37)]:

$$\mathbf{U}_L = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 \\ \mathbf{u}_2 & \mathbf{u}_1 \end{bmatrix}, \quad (10)$$

For numerical test of our idea, we developed a C/C++ code that solves Eq. (8). In the next section, we show two examples and describe how to reproduce our results.

³ <http://mathworld.wolfram.com/Eigenvector.html> (accessed 01/01/2019).

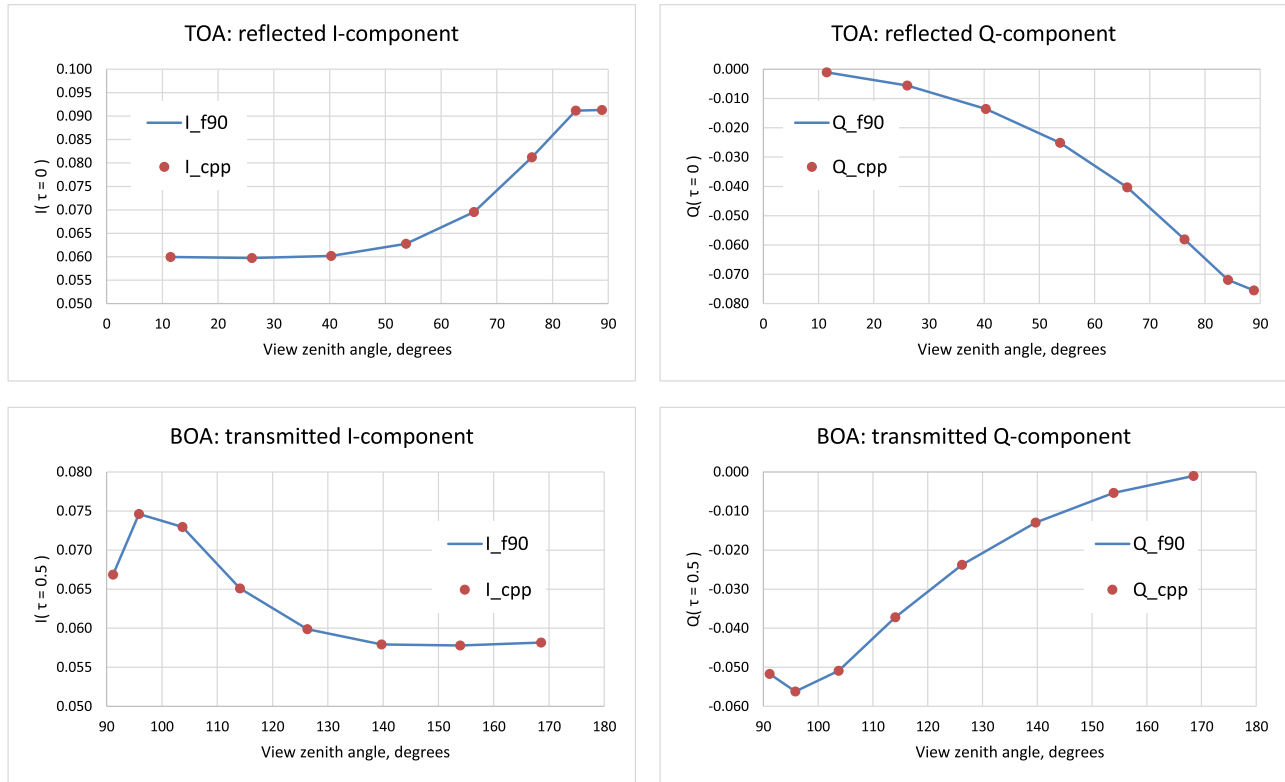


Fig. 1. I and Q components on top (TOA, $\tau = 0$) and bottom (BOA, $\tau = \tau_0$) of atmosphere for the Rayleigh case at Gauss nodes. Dot markers—C/C++, blue line—Fortran. See Table 1 for input. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Input parameters for numerical tests. See Section 3 for details.

Case	μ_0	τ_0	ω_0	K	N	$p(0^\circ)$	g	Reference
Rayleigh	1	0.5	1.0	3	8	1.5	0	[9: Case A1-1]
Aerosol	0.6	1	0.95	36	8	16	0.695	[39: Model 1]

3. Numerical results and discussion

In this section, we show results for two numerical simulations, Rayleigh [9: Case A1-1] and Aerosol [39: Model 1—prolate spheroids]. We use our full vector RT code IPOL as a benchmark. For simplicity, we consider only homogeneous atmospheres over black surface and compute solution at $2N=16$ double Gauss nodes [36], $N=8$ nodes per hemisphere. In the Rayleigh scenario, we replace the true conservative case $\omega_0=1.0$ with a “pseudo-conservative” $\omega_0=0.99999999$ (eight digits) which is a widely used practice [22,34]. In the Aerosol scenario, we compute only azimuthally averaged Stokes vector (the Fourier harmonic $m=0$) similar to Lyapustin and Rozanov [30] which makes $U=0$ in that scenario.

Tables 1 and 2 summarize the results. Table 1, left to right, defines the input parameters: name of the case; cosine of the solar zenith angle μ_0 ; the total optical thickness of the atmosphere τ_0 ; the single scattering albedo ω_0 ; the total number of the phase matrix expansion moments K ; the number of the Gauss nodes per hemisphere N ; the peak value of the phase function $p(0^\circ)$; the average scattering cosine g ; and a reference for each scenario. Table 2 shows numerical results generated by the C/C++ prototype of IPOL for both cases. We found perfect agreement in all the indicated digits, except for only 1 digit, marked in bold, which differs by 1 unit. Figs. 1 and 2 show the comparison for the total intensity I , and the Q -component versa the zenith angle.

Independent reproducibility is often omitted nowadays [10,18,28]. In order to maintain this essential criterion of scientific research, we developed and publicly distribute a C/C++ example code that reproduces our results. It is available from GitHub⁴ or by email request from the corresponding author. We link the code against the freeware C/C++ GNU Scientific Library,⁵ v.2.5. The package includes the following six files (five *.cpp source codes and one *.txt) zipped in `jqsr_2019a.zip`:

`domm0.cpp`—is a subroutine that implements equations from Section 2 using the method of discrete ordinates (`dom`) for a Fourier harmonic $m=0$. We emphasize, `domm0` is only a prototype subroutine: it does not compute quantities at arbitrary μ , it was not yet tested for an atmosphere with many layers (although a loop over layers is included), it defines excessive arrays to compute the U -component for $m > 0$, it does not include single scattering correction or reflecting surface. The sole purpose of `domm0` is to reproduce the reported results.

The `domm0` subroutine does not compute \mathbf{B} , Eq. (2), neither it computes \mathbf{B}^T , Eq. (9). Stamnes and Swanson [33, Eq. (12)], formulate and solve the half-space eigenproblem using $(\alpha - \beta)$ and $(\alpha + \beta)$. We do the same, except for the transposed matrices: $(\alpha^T - \beta^T)$ and $(\alpha^T + \beta^T)$. For efficiency, we create all the matrices already transposed instead of explicitly call the GSL transpose function `gsl_matrix_transpose()`. Further, we note $(\alpha^T - \beta^T)$ uses only even orders of the expansion coefficients, while $(\alpha^T + \beta^T)$ —only odd. This explains the requirement for the `domm0` input: the number of the used expansion moments nk must be odd with a minimum value of 3 for Rayleigh scattering. Thus, the considered Aerosol case uses $nk=2*N+1=17$ expansion mo-

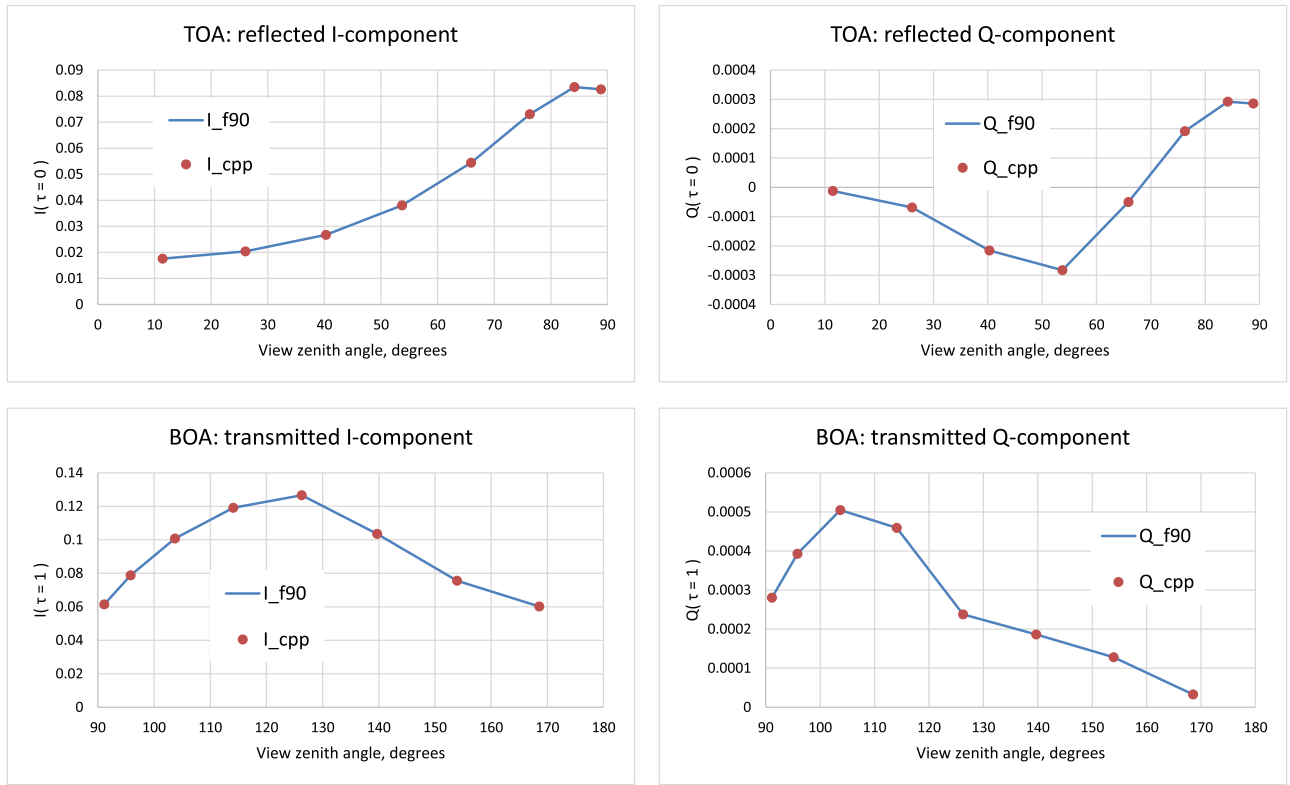
⁴ https://github.com/korkins/jqsr_2019a (accessed 01/01/2019).

⁵ <https://www.gnu.org/software/gsl/> (accessed 01/01/2019).

Table 2

Numerical results for the C/C++ code. All digits agree with the Fortran computations, except for the last digit in I ($\mu=0.1017$), Rayleigh case, differs by one unit – marked bold. Negative and positive Gauss nodes, μ , correspond to radiation at top (reflected) and bottom (transmitted) of atmosphere, respectively. See Table 1 for input.

Case μ	Rayleigh		Aerosol	
	I	Q	I	Q
–0.0199	9.13232E – 02	–7.54817E – 02	8.26495E – 02	2.85711E – 04
–0.1017	9.11630E – 02	–7.19422E – 02	8.34655E – 02	2.92456E – 04
–0.2372	8.12421E – 02	–5.80813E – 02	7.30460E – 02	1.91619E – 04
–0.4083	6.95400E – 02	–4.02927E – 02	5.44124E – 02	–5.04486E – 05
–0.5917	6.27695E – 02	–2.51031E – 02	3.80289E – 02	–2.82670E – 04
–0.7628	6.01544E – 02	–1.35367E – 02	2.67002E – 02	–2.15704E – 04
–0.8983	5.97108E – 02	–5.52871E – 03	2.03950E – 02	–6.85303E – 05
–0.9801	5.99419E – 02	–1.05280E – 03	1.75805E – 02	–1.22715E – 05
0.0199	6.68614E – 02	–5.16898E – 02	6.13960E – 02	2.80410E – 04
0.1017	7.46133E – 02	–5.62263E – 02	7.87672E – 02	3.93080E – 04
0.2372	7.29682E – 02	–5.09112E – 02	1.00791E – 01	5.05020E – 04
0.4083	6.50852E – 02	–3.71982E – 02	1.19207E – 01	4.59117E – 04
0.5917	5.98697E – 02	–2.37405E – 02	1.26586E – 01	2.37564E – 04
0.7628	5.79301E – 02	–1.29607E – 02	1.03563E – 01	1.86095E – 04
0.8983	5.77966E – 02	–5.32793E – 03	7.55802E – 02	1.28067E – 04
0.9801	5.81608E – 02	–1.01767E – 03	6.02075E – 02	3.24825E – 05

**Fig. 2.** Same as Fig. 1 except for the Aerosol case. See Table 1 for input.

ments out of 34 total. For an adequate comparison, the single scattering correction was switched off in the Fortran version of IPOL.

Another benefit of our approach, Eq. (7), specifically related to the C/C++ language, is that the left eigenvectors are the row vectors. Thus the matrix of left eigenvectors, Eq. (10), is located in memory in a native C/C++ row-major order. In Fortran, U_L must be transposed for efficiency.

gauszv.cpp, rayxk.cpp, and read2d.cpp – compute the Gauss nodes, Rayleigh scattering matrix expansion moments for the first test, and read in a txt file with the Aerosol expansion moments for the second test, respectively. They are thoroughly commented and should be self-explanatory;

test_domm0.cpp is the main program. It creates input for both cases, calls domm0.cpp and other subroutines, and prints out the result on the screen;

Finally, Xk0036_0695.txt is an ASCII file with the expansion moments for the Aerosol case. In that file, the columns left to right are: k —the number of the expansion moment starting from 0 followed by 6 columns for expansion moments for the elements of the scattering matrix $[a_1 \ a_2 \ a_3 \ a_4 \ b_1 \ b_2]$ — a_4 and b_2 are not used [12: Appendix C].

A natural question is the gain of speed. Since we do not compute the right eigenvectors at all and do not transpose matrices, we definitely anticipate a slightly higher gain in performance than

the 15% reported earlier [8: Abstract]. Unfortunately, a detailed performance analysis against our Fortran 90/95 version of IPOL is not possible. Mostly because the Fortran version of IPOL computes the full Stokes vector using complex arithmetic, do not rely on the solution of the eigenproblem in a half-space, and hence very time consuming. However, we believe the main advantage of our approach lies in some simplification of theoretical background and coding, especially in C/C++.

4. Summary

We apply only left eigenvectors with the Karp's et al. [13] scaling transformation to evaluate the matrix exponential in a plane-parallel vector radiative transfer equation. This is contrary to a widely used right eigenvectors approach, which either requires an explicit inversion of a matrix of the right eigenvectors (one per each optical layer) or the combined use of the right and left eigenvectors to avoid this inversion. Our approach avoids complications related to the proper normalization when both the left and right eigenvectors are used, and hence makes an RT code easier to develop and debug. It also uses the known half-space eigenproblem and hence applicable for efficient numerical simulation of unpolarized or linear polarized light scattering with the methods of discrete ordinates or spherical harmonics. Due to the left eigenvectors are essentially row vectors, the implementation of our approach in C/C++ is preferable. We demonstrate two numerical examples and offer a GNU Scientific Library-dependent C/C++ source code, https://github.com/korkins/jqsrt_2019a, for independent crosscheck of the reported results. We presented numerical results for azimuthally averaged quantities only. However, for cases with azimuthal dependence and non-zero U -component of the Stokes vector, the T -polynomials possess the same symmetry with respect to $\pm\mu$ as the Legendre and R -polynomials, involved in our tests [12: Appendix B].

Acknowledgments

The research of Alexei Lyapustin and Sergey Korkin was funded by NASA Science for Terra, Aqua and SNPP (17-TASNPP17-0116; solicitation NNH17ZDA001NTASNPP). The authors are thankful to an anonymous Reviewer for independent validation of our approach in his RT code and for valuable suggestions that helped clarify the paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.jqsrt.2019.02.009](https://doi.org/10.1016/j.jqsrt.2019.02.009).

References

- [1] Budak VP, Kaloshin GA, Shagalov OV, Zheltov VS. Numerical modeling of the radiative transfer in a turbid medium using the synthetic iteration. *Opt Express* 2015;23(15):A829–40.
- [2] Chalhoub ES, Garcia RDM. The equivalence between two techniques of angular interpolation for the discrete-ordinates method. *J Quant Spectrosc Radiat Transfer* 2000;64:517–35.
- [3] Chandrasekhar S. Radiative transfer. London: Oxford University Press; 1950.
- [4] Deschamps P, Breon F, Leroy M, Podaire A, Bricaud A, Buriez J, Seze G. The POLDER mission: instrument characteristics and scientific objectives. *IEEE Trans Geosci Remote Sens* 1994;32(3):598–615.
- [5] Doicu A, Trautmann T. Discrete-ordinate method with matrix exponential for a pseudo-spherical atmosphere: scalar case. *J Quant Spectrosc Radiat Transfer* 2009;110:146–58.
- [6] Doicu A, Trautmann T. Discrete-ordinate method with matrix exponential for a pseudo-spherical atmosphere: vector case. *J Quant Spectrosc Radiat Transfer* 2009;110:159–72.
- [7] Efremenko DS, Doicu A, Loyola D, Trautmann T. Acceleration techniques for the discrete ordinate method. *J Quant Spectrosc Radiat Transfer* 2013;114:73–81.
- [8] Efremenko DS, Garcia VM, Garcia SG, Doicu A. A review of the matrix-exponential formalism in radiative transfer. *J Quant Spectrosc Radiat Transfer* 2017;196:17–45.
- [9] Emde C, Barlakas V, Cornet C, Evans F, Korkin S, Ota Y, C-Labonnote L, Lyapustin A, Macke A, Mayer B, Wendisch M. IPRT polarized radiative transfer model intercomparison project – phase A. *J Quant Spectrosc Radiat Transfer* 2015;164:8–36.
- [10] Fomel S, Claerbout JF. Reproducible research. *Comput Sci Eng* 2009;11(5):5–7.
- [11] Fougne B, Marbach T, Lacan A, Lang R, Schlüssel P, Poli G, Munro R, Couto A. The multi-viewing multi-channel multi-polarisation imager – overview of the 3MI polarimetric mission for aerosol and cloud characterization. *J Quant Spectrosc Radiat Transfer* 2018;219:23–32.
- [12] Hovenier JW, van der Mee C, Domke H. Transfer of polarized light in planetary atmospheres. Basic concepts and practical methods. Dordrecht, Netherlands: Kluwer Academic Publishers; 2004.
- [13] Karp AH, Greenstadt J, Fillmore JA. Radiative transfer through an arbitrary thick scattering atmosphere. *J Quant Spectrosc Radiat Transfer* 1980;24(5):391–406.
- [14] Kawata Y. Circular polarization of sunlight reflected by planetary atmospheres. *Icarus* 1978;33:217–32.
- [15] Korkin S, Lyapustin A, Sinyuk A, Holben B, Kokhanovsky A. Vector radiative transfer code SORD: performance analysis and quick start guide. *J Quant Spectrosc Radiat Transfer* 2017;200:295–310.
- [16] Lenoble J, editor. Radiative transfer in scattering and absorbing atmospheres: standard computational procedures. Hampton VA: A. Deepak Publishing; 1985.
- [17] Lenoble J, Herman M, Deuzé J L, Lafrance B, Santer R, Tanré D. A successive order of scattering code for solving the vector equation of transfer in the earth's atmosphere with aerosols. *J Quant Spectrosc Radiat Transfer* 2007;107:479–507.
- [18] LeVeque RJ, Mitchell IM, Stodden V. Reproducible research for scientific computing: tools and strategies for changing the culture. *Comput Sci Eng* 2012;14(13):13–17.
- [19] Li L, Li Z, Li K, Blarel L, Wendisch M. A method to calculate Stokes parameters and angle of polarization of sky light from polarized CIMEL sun/sky radiometers. *J Quant Spectrosc Radiat Transfer* 2014;149:334–46.
- [20] Li Z, Hou W, Hong J, Zheng F, Luo D, Wang J, Gu X, Qiao Y. Directional Polarimetric Camera (DPC): monitoring aerosol spectral optical properties over land from satellite observation. *J Quant Spectrosc Radiat Transfer* 2018;218:21–37.
- [21] Liu Y, Diner D. Multi-angle imager for aerosols: a satellite investigation to Benefit Public Health. *Public Health Rep* 2017;132(1):14–17.
- [22] Lyapustin A, Muldashev T, Wang Y. Code SHARM: fast and accurate radiative transfer over spatially variable anisotropic surfaces. In: Kokhanovsky AA, editor. Light scattering reviews 5. Single light scattering and radiative transfer. Chichester: Springer; 2010. p. 205–47.
- [23] Lyapustin A, Wang Y, Korkin S, Huang D. MODIS collection 6 MAIAC algorithm. *Atmos Meas Tech* 2018;11:5741–65.
- [24] Mishchenko M, Cairns B, Kopp G, Schueler C, Fafaul D, Hansen J, Hooker R, Itchkawich T, Maring H, Travis L. Accurate monitoring of terrestrial aerosols and total solar irradiance: introducing the Glory mission. *Bull Am Meteorol Soc* 2007;88:677–91.
- [25] Moler CB, van Loan CF. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev* 2003;45:3–49.
- [26] Nakajima T, Tanaka M. Matrix formulations for the transfer of solar radiation in a plane-parallel scattering atmosphere. *J Quant Spectrosc Radiat Transfer* 1986;35(1):13–21.
- [27] Ota Y, Higurashi A, Nakajima T, Yokota T. Matrix formulations of radiative transfer including the polarization effect in a coupled atmosphere–ocean system. *J Quant Spectrosc Radiat Transfer* 2010;111:878–94.
- [28] Peng RD. Reproducible research in computational science. *Science* 2011;334(6060):1226–7.
- [29] Plass GN, Kattawar GW, Catchings FE. Matrix operator theory of radiative transfer. 1: Rayleigh scattering. *Appl Opt* 1973;12:314–29.
- [30] Rozanov VV, Lyapustin AI. Similarity of radiative transfer equation: error analysis of phase function truncation techniques. *J Quant Spectrosc Radiat Transfer* 2010;111:1964–79.
- [31] Rozanov VV, Rozanov AV, Kokhanovsky AA, Burrows JP. Radiative transfer through terrestrial atmosphere and ocean: software package SCIATRAN. *J Quant Spectrosc Radiat Transfer* 2014;133:13–71.
- [32] Spurr RJD. LIDORT and VLIDORT: linearized pseudo-spherical scalar and vector discrete ordinate radiative transfer models for use in remote sensing retrieval problems. In: Kokhanovsky AA, editor. Light scattering reviews 3. Light scattering and reflection. Chichester: Springer; 2008. p. 229–75.
- [33] Stamnes K, Swanson RA. A new look at the discrete ordinate method for radiative transfer calculations in anisotropically scattering atmospheres. *J Atmos Sci* 1981;38(2):387–9.
- [34] Stamnes K, Tsay SC, Wiscombe W, Jayaweera K. Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media. *Appl Opt* 1988;27(12):2502–9.
- [35] Siewert CE. A discrete-ordinates solution for radiative-transfer models that include polarization effects. *J Quant Spectrosc Radiat Transfer* 2000;64:227–54.
- [36] Sykes JB. Approximate integration of the equation of transfer. *Mon Not R Astron Soc* 1951;111:378–86.
- [37] Thomas GE, Stamnes K. Radiative transfer in the atmosphere and ocean. Cambridge University Press; 1999.
- [38] Waterman PC. Matrix-exponential description of radiative transfer. *J Opt Soc Am* 1981;71(4):410–22.
- [39] Wauben WMF, de Haan JF, Hovenier JW. Low orders of scattering in plane-parallel homogeneous atmosphere. *Astron Astrophys* 1993;276:589–602.