The Journal of Open Source Software

PythonicDISORT: A Python reimplementation of the Discrete Ordinate Radiative Transfer package DISORT

Dion J. X. Ho[1]

1 Columbia University, Department of Applied Physics and Applied Mathematics, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Sophie Beck

Reviewers:

- @arjunsavel
- @simonrp84
- @pscicluna

Submitted: 11 February 2024
Published: unpublished

Summary

The Radiative Transfer Equation (RTE) models the processes of absorption, scattering and emission as electromagnetic radiation propagates through a medium. Consider a plane-parallel, horizontally homogeneous atmosphere with vertical coordinate τ (optical depth) increasing from top to bottom and directional coordinates ϕ for the azimuthal angle (positive is counterclockwise) and $\mu = \cos\theta$ for the polar direction (θ is the polar angle measured from the surface normal), with $\mu > 0$ pointing up following the convention of (K. Stamnes et al., 1988). Given three possible sources: blackbody emission from the atmosphere $s(\tau)$, scattering from a collimated beam of starlight with intensity I_0 and incident cosine polar and azimuthal angles μ_0, ϕ_0, and/or radiation from other atmospheric layers or the Earth's surface which is modeled by Dirichlet boundary conditions, the diffuse intensity $u(\tau, \mu, \phi)$ propagating in direction (μ, ϕ) is described by the 1D RTE (Chandrasekhar, 1960; K. Stamnes et al., 1988):

$$
\begin{aligned}
\mu\frac{\partial u(\tau,\mu,\phi)}{\partial\tau} = u(\tau,\mu,\phi) &- \frac{\omega}{4\pi}\int_{-1}^{1}\int_{0}^{2\pi} p\left(\mu,\phi;\mu',\phi'\right) u\left(\tau,\mu',\phi'\right) \mathrm{d}\phi' \mathrm{d}\mu' \\
&- \frac{\omega I_0}{4\pi} p\left(\mu,\phi;-\mu_0,\phi_0\right) \exp\left(-\mu_0^{-1}\tau\right) - s(\tau)
\end{aligned} \tag{1}
$$

Here ω is the single-scattering albedo and p the scattering phase function. These are assumed to be independent of τ, i.e. homogeneous in the atmospheric layer. An atmosphere with τ-dependent ω and p can be modelled by a multi-layer atmosphere with different ω and p for each layer.

The RTE is important in many fields of science and engineering, for example, in the retrieval of optical properties from measurements (McGuire et al., 2008; Teng et al., 2020; Torricella et al., 1999). The gold standard for numerically solving the 1D RTE is the Discrete Ordinate Radiative Transfer package `DISORT` which was coded in FORTRAN 77 and first released in 1988 (K. Stamnes et al., 1988; S. Stamnes, 1999). It has been widely used, for example by `MODTRAN` (Berk et al., 2014), `Streamer` (Key & Schweiger, 1998), and `SBDART` (Ricchiazzi et al., 1998), all of which are comprehensive radiative transfer models that are themselves widely used in atmospheric science, and by the three retrieval papers Torricella et al. (1999); McGuire et al. (2008); Teng et al. (2020). `DISORT` implements the Discrete Ordinates Method which has two key steps. First, the diffuse intensity function u and phase function p are expanded as the Fourier cosine series and Legendre series respectively:

$$
\begin{aligned}
u\left(\tau,\mu,\phi\right) &\approx \sum_{m=0} u^m\left(\tau,\mu\right)\cos\left(m\left(\phi_0-\phi\right)\right) \\
p\left(\mu,\phi;\mu',\phi'\right) = p\left(\cos\gamma\right) &\approx \sum_{\ell=0}(2\ell+1) g_\ell P_\ell\left(\cos\gamma\right)
\end{aligned}
$$

where γ is the scattering angle. These address the ϕ' integral in (1) and decompose the problem into solving

$$\mu \frac{du^m(\tau,\mu)}{d\tau} = u^m(\tau,\mu) - \int_{-1}^{1} D^m(\mu,\mu')\, u^m(\tau,\mu')\, \mathrm{d}\mu' - Q^m(\tau,\mu) - \delta_{0m} s(\tau)$$

for each Fourier mode of u. The terms D^m are derived from p and are thus also independent of τ. The second key step is to discretize the μ' integral using some quadrature scheme; `DISORT` uses the double-Gauss quadrature scheme from Sykes (1951). This results in a system of ordinary differential equations that can be solved using standard methods.

Our package `PythonicDISORT` is a Python 3 reimplementation of `DISORT` that replicates most of its functionality while being easier to install, use, and modify, though at the cost of computational speed. It has `DISORT`'s main features: multi-layer solver, delta-M scaling, Nakajima-Tanaka (NT) corrections, only flux option, direct beam source, isotropic internal source (blackbody emission), Dirichlet boundary conditions (diffuse flux boundary sources), Bi-Directional Reflectance Function (BDRF) for surface reflection, and more. In addition, `PythonicDISORT` has been tested against `DISORT` on `DISORT`'s own test problems. While packages that wrap `DISORT` in Python already exist (Connour & Wolff, 2020; Hu, 2017), `PythonicDISORT` is the first time `DISORT` has been reimplemented from scratch in Python.

Statement of need

`PythonicDISORT` is not meant to replace `DISORT`. Due to fundamental differences between Python and FORTRAN, `PythonicDISORT`, though optimized, remains about an order of magnitude slower than `DISORT`. Thus, projects that prioritize computational speed should still use `DISORT`. I will continue to optimize `PythonicDISORT`; there remain avenues for code vectorization among other optimizations. It is unlikely that `PythonicDISORT` can be optimized to achieve the speed of `DISORT` though. In addition, `PythonicDISORT` currently lacks `DISORT`'s latest features, most notably its pseudo-spherical correction, though I am open to adding new features and I added a subroutine to compute actinic fluxes to satisfy a user request.

`PythonicDISORT` is instead designed with three goals in mind. First, it is meant to be a pedagogical and exploratory tool. `PythonicDISORT`'s ease of installation and use makes it a low-barrier introduction to Radiative Transfer and Discrete Ordinates Solvers. Even researchers who are experienced in the field may find it useful to experiment with `PythonicDISORT` before deciding whether and how to upscale with `DISORT`. Installation of `PythonicDISORT` through `pip` should be system agnostic as `PythonicDISORT`'s core dependencies are only `NumPy` (Harris et al., 2020) and `SciPy` (Virtanen et al., 2020). I also intend to implement `conda` installation. In addition, using `PythonicDISORT` is as simple as calling the Python function `pydisort`. In contrast, `DISORT` requires FORTRAN compilers, has a lengthy and system dependent installation, and each call requires shell script for compilation and execution.

Second, `PythonicDISORT` is designed to be modified by users to suit their needs. Given that Python is a widely-used high-level language, `PythonicDISORT`'s code should be understandable, at least more so than `DISORT`'s FORTRAN code. Moreover, `PythonicDISORT` comes with a Jupyter Notebook (Kluyver et al., 2016) – our *Comprehensive Documentation* – that breaks down both the mathematics and code behind the solver. Users can in theory follow the Notebook to recode `PythonicDISORT` from scratch; it should at least help them make modifications.

Third, `PythonicDISORT` is intended to be a testbed. For the same reasons given above, it should be easier to implement and test experimental features in `PythonicDISORT` than in `DISORT`. This should expedite research and development for `DISORT` and similar algorithms.

`PythonicDISORT` was first released on PyPI and GitHub on May 30, 2023. I know of its use in at least three ongoing projects: on the Two-Stream Approximations, on atmospheric photolysis, and on the topographic mapping of Mars through photoclinometry. I will continue to maintain and upgrade `PythonicDISORT`. Our latest version: `PythonicDISORT v0.8.0` was released on June 10, 2024.

Acknowledgements

I acknowledge funding from NSF through the Learning the Earth with Artificial intelligence and Physics (LEAP) Science and Technology Center (STC) (Award #2019625). I am also grateful to my Columbia University PhD advisor Dr. Robert Pincus and co-advisor Dr. Kui Ren for their advice and contributions.

References

Berk, A., Conforti, P., Kennett, R., Perkins, T., Hawes, F., & Bosch, J. van den. (2014). MODTRAN® 6: A major upgrade of the MODTRAN® radiative transfer code. *2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 1–4. https://doi.org/10.1109/WHISPERS.2014.8077573

Chandrasekhar, S. (1960). *Radiative transfer*. Dover.

Connour, K., & Wolff, M. (2020). *pyRT_DISORT: A pre-processing front-end to help make DISORT simulations easier in Python* (Version 1.0.0). https://github.com/kconnour/pyRT_DISORT

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hu, Z. (2017). *pyDISORT* (Version 0.8). https://github.com/chanGimeno/pyDISORT

Key, J. R., & Schweiger, A. J. (1998). Tools for atmospheric radiative transfer: Streamer and FluxNet. *Computers & Geosciences*, *24*(5), 443–451. https://doi.org/10.1016/S0098-3004(97)00130-1

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. (2016). *Jupyter notebooks – a publishing format for reproducible computational workflows* (F. Loizides & B. Schmidt, Eds.; pp. 87–90). IOS Press.

McGuire, P. C., Wolff, M. J., Smith, M. D., Arvidson, R. E., Murchie, S. L., Clancy, R. T., Roush, T. L., Cull, S. C., Lichtenberg, K. A., Wiseman, S. M., Green, R. O., Martin, T. Z., Milliken, R. E., Cavender, P. J., Humm, D. C., Seelos, F. P., Seelos, K. D., Taylor, H. W., Ehlmann, B. L., … Malaret, E. R. (2008). MRO/CRISM retrieval of surface lambert albedos for multispectral mapping of mars with DISORT-based radiative transfer modeling: Phase 1—using historical climatology for temperatures, aerosol optical depths, and atmospheric pressures. *IEEE Transactions on Geoscience and Remote Sensing*, *46*(12), 4020–4040. https://doi.org/10.1109/TGRS.2008.2000631

Ricchiazzi, P., Yang, S., Gautier, C., & Sowle, D. (1998). SBDART: A research and teaching software tool for plane-parallel radiative transfer in the earth's atmosphere. *Bulletin of the American Meteorological Society*, *79*(10), 2101–2114. https://doi.org/10.1175/1520-0477(1998)079%3C2101:SARATS%3E2.0.CO;2

Stamnes, K., Tsay, S.-C., Wiscombe, W., & Jayaweera, K. (1988). Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media. *Appl. Opt.*, *27*(12), 2502–2509. https://doi.org/10.1364/AO.27.002502

Stamnes, S. (1999). LLLab disort website. In *Light and Life Lab (LLLab)*. http://www.rtatmocn.com/disort/

Sykes, J. B. (1951). Approximate Integration of the Equation of Transfer. *Monthly Notices of the Royal Astronomical Society*, *111*(4), 377–386. https://doi.org/10.1093/mnras/111.4.377

Teng, S., Liu, C., Zhang, Z., Wang, Y., Sohn, B.-J., & Yung, Y. L. (2020). Retrieval of ice-over-water cloud microphysical and optical properties using passive radiometers. *Geophysical Research Letters*, *47*(16), e2020GL088941. https://doi.org/10.1029/2020GL088941

Torricella, F., Cattani, E., Cervino, M., Guzzi, R., & Levoni, C. (1999). Retrieval of aerosol properties over the ocean using global ozone monitoring experiment measurements: Method and applications to test cases. *Journal of Geophysical Research: Atmospheres*, *104*(D10), 12085–12098. https://doi.org/10.1029/1999JD900040

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2