The Journal of Open Source Software

SpectralModel: a high-resolution framework for petitRADTRANS 3

Doriann Blain [1¶], **Paul Mollière** [1], **and Evert Nasedkin** [1]

1 Max Planck Institut für Astronomie, DE ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Monica Bobra

Reviewers:

- @spectralcode
- @benhord

Submitted: 28 May 2024
Published: unpublished

Summary

Atmospheric characterisation from spectroscopic data is a key to understand planetary formation. Two types of observations can be performed for this kind of analysis. Space-based observations (e.g., using the James Webb Space Telescope, JWST), are not impeded by the Earth's atmosphere, but are currently limited to low resolving powers (< 3000), which can lead to ambiguities in some species detections. Ground-based observations (e.g., using the Very Large Telescope, VLT), on the other hand, can benefit from large resolving powers ($\approx 10^5$), allowing for unambiguous species detection, but are impacted by telluric spectral lines. `petitRADTRANS` (pRT) is a radiative transfer package used for computing emission or transmission spectra of planetary atmospheres (Mollière et al., 2019). The package has a non-negligible user base, the original article being cited in 264 refereed works at the time of writing. pRT is already relatively easy to use on space-based, low-resolution observations. However, while the package technically has the capacity to analyse high-resolution spectra, thanks to its ability to incorporate high-resolution ($\mathcal{R} = 10^6$) line lists, ground-based observations analysis is a complex and challenging task. The new `SpectralModel` object provides a powerful and flexible framework that streamlines the setup necessary to model and retrieve high-resolution spectra.

Statement of need

Calculating a spectrum using pRT's core object `Radtrans` is a two-step process in which the user first instantiates the object, giving parameters that control the loading of opacities. The second step is for the user to call one of the `Radtrans` function, giving "spectral" parameters such as the temperatures or the mass fractions of the atmosphere, that will be used in combination with the loaded opacities to generate the spectrum.

However, these two steps are by themselves often insufficient to build a spectrum in a real-life scenario. The spectral parameters may individually rely on arbitrarily complex models requiring their own parameters, and may depend on each other. For example, getting mass fractions from equilibrium chemistry requires knowing the temperature profile, and the mean molar mass requires knowing the mass fractions (see e.g. the built-in pRT functions). Common operations such as convolving the spectrum, scaling it to stellar flux, or more specifically for high-resolution spectra, Doppler-shifting the spectrum and including the transit effect, must be done by post-processing the `Radtrans`-generated spectrum. Finally, using a retrieval requires to code a "retrieval model" including all the steps described above. This induces, especially for first-time users, a significant setup cost. The alternative is to use one of pRT's built-in models, but this lacks flexibility.

The `SpectralModel` object extends the base capabilities of the petitRADTRANS package by providing a standardized but flexible framework for spectral calculations. It has been especially designed to effectively erase the setup cost of modelling the spectral Doppler-shift, the transit effect, and of implementing the preparation step necessary for ground-based high-resolution observations analysis. `SpectralModel` is also interfaced with pRT's `retrieval` module (Nasedkin et al., 2024), and as such is an easy-to-use tool to perform both high- and low-resolution atmospheric retrievals. Compared to other commonly used spectral modelling packages, for example ATMOSPHERIX (Klein et al., 2023), Brewster (Burningham et al., 2021), CHIMERA (Line et al., 2013), PSG (Villanueva et al., 2018), NEMESIS (Irwin et al., 2008), PICASO (Batalha et al., 2019), PLATON (Zhang et al., 2020), POSEIDON (MacDonald, 2023), TauREx (Al-Refaie et al., 2021), petitRADTRANS is currently, to our knowledge, the only one able to both generate time-varying high-resolution spectra and retrieve the corresponding data out-of-the-box[1].

The combination of ease-of-use and flexibility offered by `SpectralModel` makes it a powerful tool for high-resolution (but also low-resolution) atmospheric characterisation. With the upcoming first light of a new generation of ground based telescopes, such as the Extremely Large Telescope, `SpectralModel` makes petitRADTRANS ready for the new scientific discoveries that will be unveiled in the next era of high-resolution observations.

The `SpectralModel` object

Main features

Spectral parameter calculation framework

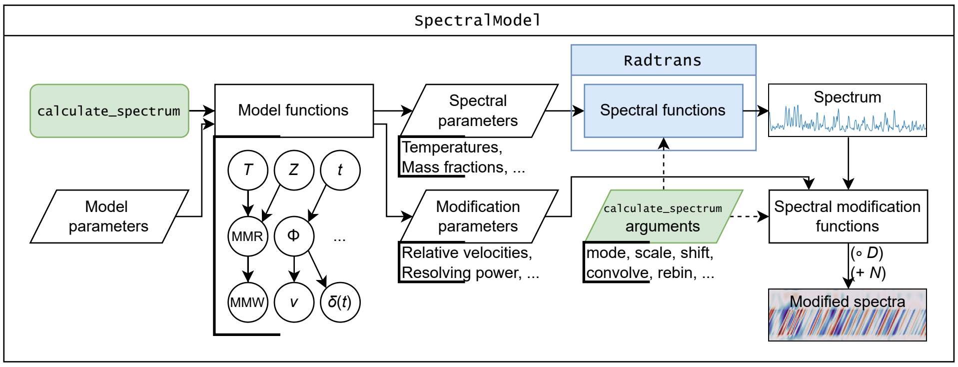

Figure 1: Flowchart of `SpectralModel.calculate_spectrum` function. The annotation below the model functions represents an example of execution order of these function after topological sorting, involving the temperature (T), the metallicity (Z), the time (t), the mass fractions (MMR), the mean molar masses (MMW), the orbital phases (ϕ), the relative velocities (v), and the transit effect (δ). Additional deformations (D) and noise (N) can also be included.

`SpectralModel` provides a framework to automatise the calculation of the spectral parameters. Each spectral parameter is linked to a function, called here "model function", which calculates its value. This feature can be extended to the parameters required for these functions, and so on. Before calculating spectra, the function's execution order is automatically determined through a topological sorting algorithm[2] (Kahn, 1962). `SpectralModel` comes with built-in functions (Blain et al., 2024) for all the spectral parameters, so that the object can be used "out-of-the-box". Parameters that ultimately do not depend on any function are called "model parameters", and must be given during instantiation.

---

[1] ATMOSPHERIX is able to make cross-correlation analysis of high-resolution spectra, but relies on petitRADTRANS to generate its templates. HYDRA-H (Gandhi et al., 2019) is a code able to perform high-resolution data retrievals, but is not publicly available. The other cited packages may have out-of-the-box single-time high-resolution spectral generation capabilities, but no time-varying high-resolution data retrieval framework, similarly to petitRADTRANS before the implementation of `SpectralModel`.

[2] Cyclic dependencies are not supported.

In addition, `SpectralModel` provides built-in functions (Blain et al., 2024) to scale, convolve, Doppler-shift, rebin, include planet transit effect, and prepare a spectrum after it has been calculated. Similarly to model functions, these "spectral modification functions" must be given, if used, their own model parameters during instantiation.

The spectral calculation is done within the `calculate_spectrum` function (see Figure 1). The spectral mode (emission or transmission), as well as which of the spectral modification to activate (i.e. only scaling, or both convolving and rebinning, etc.), are controlled through the function's arguments ("spectral modification parameters").

Automatic optimal wavelength range calculation

A way to slightly reduce the high[3] memory usage of high-resolution spectral analysis is to load exactly the wavelength range required for an analysis, instead of relying on manual inputs. This task is complicated in high-resolution retrievals due to parameters influencing the Doppler-shift (that is, the radial velocity semi-amplitude K_p, the rest frame velocity shift V_{rest}, and the mid transit time offset T_0) being retrieved. `SpectralModel` comes with a class method which takes into account the (uniform) prior range of these parameters to automatically calculate the optimal wavelength range to load.

Interface with pRT's `retrieval` module

In order to be able to perform high-resolution data retrievals, the `Retrieval` object has been extended to support spectra with up to 3 dimensions, intended to be spectral order, exposure (time), and spectral pixel (wavelength). Several improvements to the module have been implemented as well:

- The retrieved data can now be provided as arrays instead of requiring a file.
- Custom `Radtrans` (or by extension `SpectralModel`) objects can now be used for retrievals.

In addition, `SpectralModel`'s model parameters and spectral modification functions can be advantageously used to simplify the retrieval setup compared to `Radtrans`'. This removes the need for several steps:

- building the `RetrievalConfig` object, as this has been automated,
- declaring the fixed parameters, as all model parameters that are not retrieved parameters are *de facto* fixed parameters,
- writing the retrieval model function, as it is given by the `SpectralModel` itself.

Ground-based high-resolution spectra contain telluric and stellar lines that must be removed. This is usually done with a "preparing" pipeline (also called "detrending" or "pre-processing" pipeline). To this end, a new `retrieval.preparing` sub-module has been implemented, containing the "Polyfit" pipeline (Blain et al., 2024) and the "SysRem" pipeline (Tamuz et al., 2005). To perform a retrieval when the data are prepared with "Polyfit", the forward model must be prepared in the same way (Blain et al., 2024). This forward model preparation step can be activated when calculating a spectrum with `SpectralModel`.

Ground-based data simulation

Data (F) taken from ground telescopes can be expressed as $F = M_\Theta \circ D + N$ (Blain et al., 2024), where M_Θ is an exact model with true parameters Θ, D ("deformation matrix") represents the combination of telluric lines, stellar lines, and instrumental deformations (pseudo-continuum, blaze function, …), and N is the noise. The operator "$\circ$" represents the element-wise product. Telluric lines, noise, and other deformations can be included in a `SpectralModel` object. A time-varying airmass can be added as model parameter to better model the telluric

---

[3] Loading a typical pRT line-by-line opacity file between 1 and 2 μm takes 804 MB of RAM, according to `numpy.ndarray.nbytes`.

lines. Finally, a command-line interface (CLI) with ESO's SKYCALC sky model calculator has been implemented, adapting the CLI provided on the ESO's website.

Workflows

Examples for these workflows are available in the pRT's documentation.

Spectra calculation

Calculating spectra with `SpectralModel` is done in two steps:

1. Instantiation: similarly to `Radtrans`, this step is done to load the opacities, and thus requires the same parameter as a `Radtrans` instantiation. In addition, the user can provide model parameters, that will give the spectral parameters and the modification parameters. Finally, a custom `dict` can be given if the user desires to use different functions than the built-in ones.
2. Calculation: spectral calculation is done with a unique function. The spectrum type (emission or transmission), as well as modification flags (for scaling, Doppler-shifting, etc.) are given as arguments.

Retrievals

Retrieving spectra with `SpectralModel` is done in seven steps:

1. Loading the data,
2. For high-resolution ground-based data: preparing the data,
3. Setting the retrieved parameters, this is done by filling a `dict`,
4. Setting the forward model, by instantiating a `SpectralModel` object,
5. Instantiating a `Data` object with the `SpectralModel` dedicated function,
6. Instantiating a `Retrieval` object from the previously built `Data` object(s),
7. Running the retrieval.

In addition, a new corner plot function, based on the `corner` package (Foreman-Mackey, 2016), has been implemented to ease the representation of the retrieval results with this framework.

The petitRADTRANS 3 update

Test	pRT 2.7.7 time (s)	pRT 3.1.0 time (s)	pRT 2.7.7 RAM (MB)	pRT 3.1.0 RAM (MB)
Opacity loading, `'c-k'`	3.2	0.9	–	–
Opacity loading, `'lbl'`	6.3	0.4	–	–
Emission, `'c-k'`	6.4	5.2	2428	1472
Emission, `'lbl'`	7.8	4.4	3929	2643
Transmission, `'c-k'`	1.2	0.6	992	757
Transmission, `'lbl'`	6.6	3.1	3929	2230

- Times are measured using the `cProfile` standard library, from the average of 7 runs.
- "RAM": peak RAM usage as reported by the `tracemalloc` standard library.
- `'c-k'`: using correlated-k opacities (CH_4 and H_2O), from 0.3 to 28 μm.
- `'lbl'`: using line-by-line opacities (CO and H_2O), from 0.9 to 1.2 μm.
- All spectra calculations are done using 100 pressure levels. Emission scattering is activated in `'c-k'` mode.
- Results obtained on Debian 12.5 (WSL2), CPU: AMD Ryzen 9 3950X @ 3.50 GHz.

Fully and seamlessly implementing `SpectralModel` into pRT required major changes and refactors to pRT's code. The changes focus on optimisations (both for speed and RAM usage) for high-resolution spectra computing, but this also impacts the correlated-k (low-resolution) part of the code (see Table 1). To speed-up "input data" (opacities, pre-calculated equilibrium chemistry table, star spectra table) loading times, pRT's loading system has been overhauled and the loaded files have been converted from a mix of ASCII, Fortran unformatted and HDF5 files to HDF5-only. Opacities now also follow an extended ExoMol database naming and structure convention. The package's installation process has been made compatible with Python ≥ 3.12[4]. Finally, several quality-of-life features (e.g., missing requested opacities can be automatically downloaded from the project's Keeper library, or the `Planet` object) have been implemented.

Acknowledgements

We thank the pRT users, who greatly helped improving the package by sharing their suggestions and reporting their issues.

References

Al-Refaie, A. F., Changeat, Q., Waldmann, I. P., & Tinetti, G. (2021). TauREx 3: A fast, dynamic, and extendable framework for retrievals. *The Astrophysical Journal*, *917*(1), 37. https://doi.org/10.3847/1538-4357/ac0252

Batalha, N. E., Marley, M. S., Lewis, N. K., & Fortney, J. J. (2019). Exoplanet reflected-light spectroscopy with PICASO. *The Astrophysical Journal*, *878*(1), 70. https://doi.org/10.3847/1538-4357/ab1b51

Blain, D., Sánchez-López, A., & Mollière, P. (2024). A formally motivated retrieval framework applied to the high-resolution transmission spectrum of HD 189733 b. *The Astronomical Journal*, *167*(4), 179. https://doi.org/10.3847/1538-3881/ad2c8b

Burningham, B., Faherty, J. K., Gonzales, E. C., Marley, M. S., Visscher, C., Lupu, R., Gaarn, J., Fabienne Bieger, M., Freedman, R., & Saumon, D. (2021). Cloud busting: enstatite and quartz clouds in the atmosphere of 2M2224-0158. *Monthly Notices of the Royal Astronomical Society*, *506*(2), 1944–1961. https://doi.org/10.1093/mnras/stab1361

Foreman-Mackey, D. (2016). Corner.py: Scatterplot matrices in python. *Journal of Open Source Software*, *1*(2), 24. https://doi.org/10.21105/joss.00024

Gandhi, S., Madhusudhan, N., Hawker, G., & Piette, A. (2019). HyDRA-h: Simultaneous hybrid retrieval of exoplanetary emission spectra. *The Astronomical Journal*, *158*(6), 228. https://doi.org/10.3847/1538-3881/ab4efc

Irwin, P. G. J., Teanby, N. A., de Kok, R., Fletcher, L. N., Howett, C. J. A., Tsang, C. C. C., Wilson, C. F., Calcutt, S. B., Nixon, C. A., & Parrish, P. D. (2008). The NEMESIS planetary atmosphere radiative transfer and retrieval tool. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *109*(6), 1136–1150. https://doi.org/https://doi.org/10.1016/j.jqsrt.2007.11.006

Kahn, A. B. (1962). Topological sorting of large networks. *Commun. ACM*, *5*(11), 558–562. https://doi.org/10.1145/368996.369025

Klein, B., Debras, F., Donati, J.-F., Hood, T., Moutou, C., Carmona, A., Ould-elkhim, M., Bézard, B., Charnay, B., Fouqué, P., Masson, A., Vinatier, S., Baruteau, C., Boisse, I., Bonfils, X., Chiavassa, A., Delfosse, X., Dethier, W., Hebrard, G., … Wyttenbach, A. (2023). ATMOSPHERIX: I- an open source high-resolution transmission spectroscopy pipeline for

---

[4] pRT 2 used the `numpy.distutils module` to compile its Fortran extensions. This module is deprecated and is removed for Python 3.12. pRT 3 uses the Meson build system instead, with almost unnoticeable changes for users.

exoplanets atmospheres with SPIRou. *Monthly Notices of the Royal Astronomical Society*, *527*(1), 544–565. https://doi.org/10.1093/mnras/stad2607

Line, M. R., Knutson, H., Deming, D., Wilkins, A., & Desert, J.-M. (2013). A NEAR-INFRARED TRANSMISSION SPECTRUM FOR THE WARM SATURN HAT-p-12b. *The Astrophysical Journal*, *778*(2), 183. https://doi.org/10.1088/0004-637X/778/2/183

MacDonald, R. J. (2023). POSEIDON: A multidimensional atmospheric retrieval code for exoplanet spectra. *Journal of Open Source Software*, *8*(81), 4873. https://doi.org/10.21105/joss.04873

Mollière, P., Wardenier, J. P., Boekel, R. van, Henning, Th., Molaverdikhani, K., & Snellen, I. A. G. (2019). petitRADTRANS: A Python radiative transfer package for exoplanet characterization and retrieval. *Astronomy & Astrophysics*, *627*, A67. https://doi.org/10.1051/0004-6361/201935470

Nasedkin, E., Mollière, P., & Blain, D. (2024). Atmospheric retrievals with petitRADTRANS. *Journal of Open Source Software*, *9*(96), 5875. https://doi.org/10.21105/joss.05875

Tamuz, O., Mazeh, T., & Zucker, S. (2005). Correcting systematic effects in a large set of photometric light curves. *Monthly Notices of the Royal Astronomical Society*, *356*(4), 1466–1470. https://doi.org/10.1111/j.1365-2966.2004.08585.x

Villanueva, G. L., Smith, M. D., Protopapa, S., Faggi, S., & Mandell, A. M. (2018). Planetary spectrum generator: An accurate online radiative transfer suite for atmospheres, comets, small bodies and exoplanets. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *217*, 86–104. https://doi.org/https://doi.org/10.1016/j.jqsrt.2018.05.023

Zhang, M., Chachan, Y., Kempton, E. M.-R., Knutson, H. A., & Chang, W. (Happy). (2020). PLATON II: New capabilities and a comprehensive retrieval on HD 189733b transit and eclipse data. *The Astrophysical Journal*, *899*(1), 27. https://doi.org/10.3847/1538-4357/aba1e6