

Université de Montréal

**Transmission spectroscopy of TRAPPIST-1d with the
new Palomar/WIRC+Spec instrument: a
Karhunen-Loève transform based approach to
extracting spectrophotometry**

par

Jonathan Chan

Département de physique
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Astrophysique

December 23, 2019

Université de Montréal

Faculté des études supérieures et postdoctorales

Ce mémoire intitulé

Transmission spectroscopy of TRAPPIST-1d with the new Palomar/WIRC+Spec instrument: a Karhunen-Loève transform based approach to extracting spectrophotometry

présenté par

Jonathan Chan

a été évalué par un jury composé des personnes suivantes :

Paul Charbonneau

(président-rapporteur)

Björn Benneke

(directeur de recherche)

Laurence Perreault Levasseur

(membre du jury)

Résumé

Le système TRAPPIST-1 offre une opportunité sans précédent de caractériser les premières planètes potentiellement habitables en dehors de notre système solaire. Dans ce mémoire est décrit le développement d'un pipeline de réduction de données personnalisé pour le mode WIRC+Spec de la caméra infrarouge à grand champ récemment mise à niveau à l'observatoire Palomar. Nous introduisons une nouvelle approche d'ajustement de la fonction d'étalement du point basée sur la transformation de Karhunen-Loève pour extraire des courbes de lumière photométrique et spectroscopique de sources de forme irrégulière, que nous appliquons aux observations de l'exoplanète TRAPPIST-1d pour mesurer ses spectres de transmission dans les bandes J (1.1 à 1.4 μm) et K_s (1.95 à 2.35 μm). Un guide détaillé est présenté pour l'implémentation d'un calcul de profils de température incluant l'équilibre radiatif et convectif pour une modélisation atmosphérique efficace et précise. En comparant une multitude de scénarios atmosphériques aux observations de TRAPPIST-1d, nous obtenons des contraintes sur la composition et la structure de son atmosphère, excluant les scénarios sans nuages avec des métallicités inférieures à 300 fois la valeur solaire à 3σ .

Mots clés: Planètes et satellites: individuels (TRAPPIST-1d) – Planètes et satellites: atmosphères – Spectroscopie de transit – Méthodes numériques

Abstract

The TRAPPIST-1 system provides an unprecedented opportunity to characterize the first potentially habitable planets outside our solar system. In this work we describe the development of a custom data reduction pipeline for the WIRC+Spec mode of the recently upgraded Wide Field Infrared Camera instrument on Palomar Observatory. We introduce a novel, Karhunen-Loève transform based approach to extract photometric and spectroscopic light curves from irregularly shaped sources, which we apply to observations of the TRAPPIST-1d exoplanet to measure the J band (1.1 to 1.4 μm) and K_s band (1.95 to 2.35 μm) transmission spectra. We also present a detailed guide into the implementation of a self-consistent, radiative-convective temperature structure calculation for efficient and accurate atmospheric modelling. Comparing a host of atmospheric scenarios to the observations of TRAPPIST-1d to date, we constrain its atmosphere, ruling out cloud-free atmospheres with metallicities lower than 300 times the solar value at 3σ confidence.

Keywords: Planets and satellites: individual (TRAPPIST-1d) – Planets and satellites: atmospheres – Transit spectroscopy – Numerical methods

Contents

Résumé	5
Abstract	7
List of tables	13
List of figures	15
List of Acronyms and Abbreviations	19
Remerciements	21
Introduction	23
1. Background	23
1.1. Radial velocity method	23
1.2. Exoplanet transits	24
2. Transit spectroscopy with Palomar Observatory	25
2.1. WIRC Upgrades	25
2.2. Atmospheric characterization of the TRAPPIST-1 system	26
3. Contributions	27
References	28
First Article. Atmospheric characterization of TRAPPIST-1d using the new Palomar/WIRC+Spec mode and a new Karhunen-Loève Transform based approach for spectrophotometric extraction	29
1. Introduction	30
2. Observations and data reduction	31
2.1. Image calibration	33
2.1.1. Dark subtraction and flat fielding	33
2.1.2. Bad pixel removal	34

2.2.	Image preparation	35
2.2.1.	Signal-to-Noise Ratio (SNR) determination	37
3.	Light Curve Extraction	39
3.1.	Photometric PSF Fitting	39
3.2.	Spectroscopic PSF Fitting	42
4.	Light curve analysis	45
4.1.	Photometric Light Curves	46
4.1.1.	Instrumental Systematics	46
4.2.	Spectroscopic Light Curves	48
5.	Atmospheric Analysis	48
5.1.	Self-consistent modelling	49
5.1.1.	Self-consistent temperature structure using the Rybicki method	49
5.1.2.	Self-consistent Models of TRAPPIST-1d	51
5.2.	Atmospheric Retrieval	52
6.	Discussion and Conclusions	53
	References	54
	Summary and Conclusion	59
	Appendix A. Self Consistent Temperature Structure Calculation	61
A.1.	Overview	61
A.2.	Starting Procedure	61
A.2.1.	Estimate Composition and Solve for Hydrostatic Equilibrium	61
A.2.2.	Calculate Opacities	61
A.2.3.	Calculate Stellar Irradiation and Planck Function (B)	62
A.3.	Solve the Radiative Transfer Equation with Feautrier's Method	63
A.3.1.	Derivation	63
A.3.2.	Boundary Conditions	64
A.3.3.	Discretization	64
A.3.4.	Numerical Methods	66
A.3.4.1.	Middle Layers	66
A.3.4.2.	Top of the atmosphere	66
A.3.4.3.	Bottom of the atmosphere	68

A.3.4.4. Solution of the Matrix.....	68
A.4. Determining the Temperature Profile	73
A.4.1. Numerical Derivatives.....	73
A.4.2. Radiative Transfer.....	74
A.4.3. Radiative Equilibrium.....	79
A.4.3.1. Integral Form	80
A.4.3.2. Differential Form.....	81
A.4.4. Convective Equilibrium.....	82
A.4.4.1. Prerequisite Calculations.....	83
A.4.4.2. Integral Form	86
A.4.4.3. Differential form	87
A.4.5. Solving for the Temperature Correction.....	89
A.4.6. Arriving at a Self-Consistent Temperature Profile.....	90
References.....	96
Appendix B. Variable Names and Definitions	97

List of tables

3.1	Summary of all transit observations of TRAPPIST-1d obtained for this study....	32
3.2	Comparison of the photometric scatter of the TRAPPIST-1 light curves using the 1 st order PSF after differential photometry. Results are shown using both aperture photometry and PSF fitting methods. J band observations are consistent to within 5% difference while K _s band observations are consistent to within 13% difference.	42

List of figures

- 3.1 Main steps of the data reduction process. The steps are colour coded according to the relevant framework used: image calibration was performed with the WIRC_DRP; image preparation and light curve extraction was performed with the custom built WIRC module for light curve extraction in the EXOplanet Transits Eclipses and Phasecurves (ExoTEP) framework; light curve analysis was performed with the transit parameter fitting module of ExoTEP; and atmospheric modelling and retrieval was performed with the Self-Consistent Atmospheric Retrieval for Exoplanets (SCARLET) framework..... 33
- 3.2 Left: Sample raw frame from WIRC+Pol, adapted from Tinyanont et al. [28]. Right: Sample raw frame from WIRC+Spec. In the exposure taken by WIRC+Pol, the PSF is spread into symmetrical quadrants based on polarization..... 34
- 3.3 Histogram of pixel count frequency for an example 1st order PSF cut-out frame. The peak corresponds to the high frequency of background pixels in the frame. A red dashed line illustrates the Gaussian fit describing the peak; the local background value for a given frame is estimated to be the mean of the Gaussian fit..... 35
- 3.4 Elliptical aperture for an example 1st order trace. The white ellipse depicts the aperture while the green line depicts the horizontal axis from which the angle of the trace θ (illustrated in red) is taken from. The colour scale depicts the number of counts read by the detector..... 38
- 3.5 Reconstruction of photometric PSF. Panel a) depicts the reconstructed PSF model. Panels b) shows the original background subtracted exposure. The red outline corresponds to the aperture used to perform aperture photometry. A residual image obtained by subtracting the background subtracted exposure by the reconstructed PSF is presented in c)..... 39
- 3.6 Comparison of normalized white light curves extracted from the 1st diffraction order PSF of the TRAPPIST-1 and reference star using the PSF fitting method and aperture photometry for the night of UT 2017/08/12. The relative flux of the

	reference star has been offset for clarity. The TRAPPIST-1 light curve is later divided by the reference star light curve in order to remove atmospheric effects, as described in Section 4.	42
3.7	Reconstruction of spectroscopic PSF for 5 spectral bins. Panel a) depicts the original background subtracted exposure. Panels b) to f) show the 5 spectral bins which are then added together to construct the PSF model, depicted in g). A residual image obtained by subtracting the background subtracted exposure (a) by the reconstructed PSF (g) is presented in h). Panels a) to g) are on the same colour scale.	43
3.8	Wavelength solution for a sample 1 st order PSF of a J band exposure, calculated by matching the counts on the detector with the filter transmission cutoffs.	44
3.9	Rectangular apertures for spectroscopy, shown in red. Due to atmospheric effects, the PSF may grow or shrink on the detector. In order to maintain precise wavelength bins based on the wavelength solution, the apertures would also need to grow or shrink in size, leading to imprecise spectrophotometry.	45
3.10	Left: Joint fit broadband light curve for the two J band Palomar transit observations of TRAPPIST-1 d. The top panel shows the best fitting model light curves (black curve), overlaid with the systematics-corrected data (circles), binned to 30 seconds. Residuals from the light curve fits are shown in the middle panels. The bottom panels shows a histogram of the residuals normalized by the fitted photometric scatter parameter for each respective transit. The residuals follow the expected Gaussian distribution for photon noise limited observations. Right: Similarly for the two K _s band transit observations of TRAPPIST-1 d, binned to 120 seconds.	46
3.11	Blue and green data points show transit depth measurements from the J band observations and K _s band observations, respectively; vertical bars indicate the 1 σ transit depth uncertainties while the horizontal bars indicate the wavelength ranges of the measurements. A blue solid line represents the <i>Spitzer</i> 4.5 μ m transit depth with the dashed representing the upper and lower bounds [7]. Similarly, the red solid and dashed lines depict the average <i>HST</i> J band transit depth and error [6]. Our results are consistent with the <i>HST</i> and <i>Spitzer</i> transit depths, albeit with larger uncertainties.	47
3.12	The SCARLET temperature structure calculation framework. The black arrows represent the first run through of the code, which does not implement convection	

	or changes to hydrostatic equilibrium, which solves stability issues. After the first convergence of the temperature structure, the code follows the flowchart with the red arrows and the equations of hydrostatic equilibrium and radiative-convection equilibrium are solved.	50
3.13	Full spectrum with transit points from this work (blue), and previous <i>HST</i> (red) and <i>Spitzer</i> (black) studies [6, 7]. Vertical bars indicate the 1σ transit depth uncertainties while the horizontal bars indicate the wavelength ranges of the measurements. Bottom: The left, middle, and right plots zoom in on the J, K_s , and <i>Spitzer</i> bandpasses respectively.	52
3.14	Self-consistent temperature-pressure profiles for different atmospheric scenarios on TRAPPIST-1d. The dashed red line illustrates the calculated equilibrium temperature for TRAPPIST-1d while the solid curves depict the temperature-pressure profiles for various atmospheric cases at radiative-convective equilibrium. Note that the pressure axis is inverted as is commonly done for atmospheric temperature profiles.	53
3.15	Constraints on metallicity versus cloud top pressure. The shading indicates the normalized probability densities derived with SCARLET’s Bayesian atmospheric retrieval framework while the black contours show the 68%, 95%, and 99.7% Bayesian credible regions. Metallicities are described in reference to multiples of solar metallicity.	54

List of Acronyms and Abbreviations

ADU	Analog-to-Digital Unit
AU	Astronomical Unit
CCD	Charge-Coupled Device
ExoTEP	EXOplanet Transits Eclipses and Phasecurves
FWHM	Full Width at Half Maximum
HST	Hubble Space Telescope
JWST	James Webb Space Telescope
KLIP	Karhunen-Loève Image Projection
KLT	Karhunen-Loève Transform
MCMC	Markov Chain Monte Carlo
NIR	Near-Infrared
OSIRIS	OH-Suppressing Infrared Integral Field Spectrograph
PCA	Principal Component Analysis
PSF	Point Spread Function
RTE	Radiative Transfer Equation
RV	Radial Velocity
SCARLET	Self-Consistent Atmospheric Retrieval for Exoplanets
SNR	Signal to Noise Ratio
WIRC	Wide Field InfraRed Camera

Remerciements

I would like to thank everyone at iREx, without whom I would not have learned as much as I did during my time in Montréal. A special thanks goes out as well to my supervisor, Björn Benneke, for his never ending support, patience, and confidence throughout.

To the "fruitygang": thank you to Lana, for keeping me sane; Stefan, for never failing to supply nourishing brain food; Merrin, for her sporadic and insightful reflections; Prashansa, for sharing her expertise; Caroline, for her unwavering focus and authentic honesty; Steven, for his helpful hand; and Simon, for his refreshing taste in cinema.

Introduction

1. Background

The discovery of exoplanets, planets that orbit a star other than the Sun, heralded the exploration of a new topic in the field of astrophysics. Since the first exoplanets were found, techniques used to detect exoplanets have flourished, resulting in the discovery and confirmation of thousands of new exoplanets. Furthermore, the refinement of observational methods has massively improved our ability to study the nature of exoplanets and characterize their atmospheres, opening the door to the possibilities of finding habitable planets outside our solar system.

1.1. Radial velocity method

In 1995, Mayor and Queloz analyzed the changes in radial velocity of the star 51 Pegasi to determine the presence of a $0.5 M_J$ companion orbiting the star at 0.05 AU, which they named 51 Pegasi b [7]. Radial-velocity (RV) detection has been used ever since to aid in the detection of hundreds of exoplanets. This method of detection uses the principle of Newton's Third Law of Motion to determine that an exoplanet orbiting a star will cause a reflexive circular motion around the centre of mass of the two combined objects. For a circular orbit, the star's motion a_\star can be given in the following equation:

$$a_\star = a_p \frac{M_p}{M_\star} \quad (1.1)$$

where M_p and M_\star represent the mass of the planet and host star respectively and a_p is the radius of the planet's orbit [6]. The motion in the host star can then be observed by examining the Doppler shift in a star's spectra; as a star moves towards or away from an observer, the observed wavelength shortens and lengthens from its expected value. The Doppler effect is given by:

$$v_r = \frac{\delta\lambda}{\lambda} c \quad (1.2)$$

where v_r is the radial velocity of the host star, $\delta\lambda$ is observed Doppler shift, λ is the expected wavelength, and c is the speed of light. The radial velocity can be determined as a function of time, yielding a sinusoidal function with a semi-amplitude K . The planet's mass, period

P , inclination i , and eccentricity e can then be related in the equation:

$$K = \left(\frac{2\pi G}{P} \right)^{\frac{1}{3}} \frac{M_p \sin i}{(M_\star + M_p)^{\frac{2}{3}}} \frac{1}{\sqrt{1 - e^2}} \quad (1.3)$$

where G is the gravitational constant. Assuming the mass of the star is much greater than the mass of the planet and that the planet is in a circular orbit, the equation becomes:

$$K = \left(\frac{P}{\text{day}} \right)^{-\frac{1}{3}} \left(\frac{M_\star}{M_\odot} \right)^{-\frac{2}{3}} \left(\frac{M_p \sin i}{M_J} \right) 203 \frac{m}{s} \quad (1.4)$$

where M_J is the mass of Jupiter ($M_J = 1.898 \times 10^{27}$ kg). Examination of the above equation exposes some of the downfalls of the radial-velocity detection technique. The period and semi-amplitude have a weak inverse dependence, making radial-velocity best for the detection of planets with small periods. The semi-amplitude is also a function of the product of M_p and $\sin i$; as a result, the mass of the planet and inclination cannot be derived separately through radial-velocity detection. Radial-velocity detection also struggles in detecting multi-planet systems, especially when the system has low mass planets with large orbits; the Newtonian reflexive motion is often too small to differentiate from the effects of larger planets in the system since the effects of smaller objects can be absorbed by the radial-velocity effects of larger objects in the system. Despite these shortcomings, radial-velocity detection remains an important method of both the discovery of new exoplanets and confirmation of the existence of exoplanets discovered by other means.

1.2. Exoplanet transits

Within five years of Mayor and Queloz's initial discovery with RV, the discovery of a planet transiting its host star revealed a new technique for the study of exoplanets. In 2000, Charbonneau et al. reported the observation of a periodic dimming of the photometric flux received from the star HD 209458, explained by the crossing of an exoplanet, HD 209458b, in front of the stellar disk in the observers line of sight [4]. This technique allows for observers to study the radii of exoplanets as they cross in front of their host star since the photometric dimming, or the transit depth, scales as:

$$\text{Transit depth} = \frac{\Delta F}{F} = (R_p/R_\star)^2 \quad (1.5)$$

where F is the stellar flux, ΔF is the observed change in flux, and R_p and R_\star are the planetary and stellar radii respectively. Further refinement of the transit technique resulted in the first detection of an exoplanet atmosphere in 2002, a mere two years later [5]. Observing the transit of HD 209458 b with the Hubble Space Telescope (HST), Charbonneau et al. measured wavelength-dependent variations in transit depth, indicating wavelength-dependent changes in the heights at which the planet becomes opaque to tangential rays. Wavelengths

at which the planet blocks more stellar flux must therefore possess atoms or molecules in its atmosphere which absorb the flux or otherwise block the light. Observations are then compared to theoretical atmospheric models and spectra to determine the molecular species corresponding to the transit depth variations. For HD 209458b, the wavelength-dependent variations were concluded to be due to absorption from sodium in the atmosphere [5]. Since then, the *Kepler* mission has discovered thousands of exoplanets with the transit technique and has helped develop our understanding of the population statistics of exoplanets [8]. Recently, in 2019, the first detection of water on a habitable-zone planet was made with transit spectroscopy, a testament to the effectiveness of the technique towards the study of exoplanetary atmospheres [3].

2. Transit spectroscopy with Palomar Observatory

In this work, we explore the intricacies involved in characterizing exoplanetary atmospheres with transit spectroscopy using the Wide Field InfraRed Camera (WIRC) instrument on the 200-inch Hale Telescope at Palomar Observatory. Palomar Observatory is located in San Diego County, California in the Palomar Mountain Range and is owned and operated by the California Institute of Technology. The Hale Telescope is one of three active research telescopes at the observatory and from the time of its construction in 1949 until 1975, the 200-inch telescope was the largest telescope in the world. The WIRC instrument was first installed on the Hale Telescope in 2002 with a 2048-square Rockwell HAWAII-2 near-infrared (NIR) detector [30]. After the failure of this detector in 2012, WIRC temporarily employed an engineering grade detector until 2016, when it received the HAWAII-2 detector previously mounted on Keck/OSIRIS.

2.1. WIRC Upgrades

The WIRC instrument is optimized for the J, H, and K_s bands with an 8.7 arc minute field-of-view. In addition to the upgraded detector, a grism was installed, allowing for the development of the new WIRC+Spec mode and granting the capability for WIRC to perform low resolution ($R \approx 100$) slitless spectroscopy [28]. Traditionally, multi-object spectroscopy requires a mask with multiple slits to be made, which allows for the diffraction of multiple objects on the detector; however one of the downfalls of slit spectrographs lies in their slit loss, which defines the waste of unused light due to the small widths of the slit for high dispersion characteristics [10]. Slit loss becomes increasingly important when observing targets that are already faint. Slitless spectroscopy overcomes this downfall by removing the slit entirely, instead using a grism—a combination of a prism and a grating—which allows light from all sources on the detector to be dispersed into wavelength dependant spectra. In the case of WIRC+Spec, each source on the detector is diffracted into their 0th, 1st, and 2nd diffraction

orders. Slitless spectroscopy also allows us to perform differential photometry, where we take the photometry of a target star and one or more reference stars on the detector and divide the measurements of the target star by the reference star(s) in order to remove atmospheric effects. One of the drawbacks of slitless spectroscopy is the elevated background level; since the effective slit width for diffraction is the size of the detector, the background counts are usually much higher than for a slit spectrograph and the background also often has residual structure [10].

2.2. Atmospheric characterization of the TRAPPIST-1 system

This work exploits the advantages of slitless spectroscopy while addressing its downfalls in order to capitalize on the recent WIRC+Spec upgrades. One of the main challenges for atmospheric characterization of many exoplanetary systems around M-dwarfs lies in the faintness of their host stars. For example, even with multiple visits, the small apertures of the current space based telescopes (HST and Spitzer) are unable to capture enough photons to precisely characterize the atmospheres of the exciting TRAPPIST-1 planets. To tackle this problem, we harness the photon collecting power of the much larger Hale Telescope with the WIRC instrument. With slitless spectroscopy, we are able to perform spectrophotometry on the faint TRAPPIST-1 system without losing excessive amounts of light to slit losses. Furthermore, with multiple reference stars on the detector we can use differential photometry to remove the atmospheric effects that are characteristic of using a ground based telescope. To analyze the data taken by the WIRC+Spec instrument, we were required to develop a data reduction software capable of extracting photometric measurements from the irregularly shaped diffraction orders. In order to reduce the effects of the elevated background counts inherent to slitless spectroscopy, we also implemented a novel point spread function (PSF) fitting algorithm to optimally remove the background from our photometric measurement.

This work demonstrates the first applications of the WIRC+Spec mode in exoplanet transmission spectroscopy and atmospheric characterization. We first describe the custom-built data reduction pipeline used to extract photometric and spectroscopic light curves from raw images taken with the new WIRC+Spec mode. We recount each of the steps within the reduction, providing details on the algorithms used and explaining our PSF fitting approach. We then delve into the construction of realistic atmospheric models and our implementation of a self-consistent, radiative-convective temperature structure calculation to the Python-based Self-Consistent Atmospheric Retrieval for Exoplanets (SCARLET) framework [1, 4]. In Appendix A, we provide a step-by-step guide to solving the radiative transfer equation (RTE) with the second-order accurate Feautrier method and using the highly efficient Rybicki linearization method to converge upon radiative-convective equilibrium. This work combines the algorithms found in several different publications into one, easily accessible document, providing users with a side-by-side comparison of the mathematical expressions and the

corresponding Python implementation. Ultimately, Appendix A and B provide a reference for those wishing to familiarize themselves with SCARLET’s temperature structure calculation or to implement these features in other atmospheric modelling codes. Finally, we present an analysis of the TRAPPIST-1d exoplanet using our newly constructed data reduction software and improved atmospheric modelling capabilities in an article being submitted to *The Astrophysical Journal*.

3. Contributions

My contributions to this paper well summarizes the work presented in this thesis. In the summer of 2017, I personally took the 5 nights of observations of TRAPPIST-1 remotely with the WIRC instrument on Palomar Observatory. Advised by my supervisor, we decided on the exposure settings for the science, dark, and flat images and I closely monitored the observations and the telescope’s progress throughout the 5 observing nights. I used the WIRC_DRP built by Tinyanont et al. to perform the initial calibration of the observations, including dark subtraction, flat fielding, and bad pixel removal [28]. I then used the Python-based EXOplanet Transits Eclipses and Phasecurves (ExoTEP) pipeline to perform a light curve extraction and analysis. ExoTEP is a highly modular analysis framework which takes photometric or spectroscopic light curves and uses a Markov chain Monte Carlo (MCMC) to find the best fitting transit parameters for a given observation. To process the observations from WIRC+Spec with ExoTEP, I developed a module from scratch which performs photometry and spectrophotometry on the calibrated WIRC+Spec images in order to extract their photometric and spectroscopic light curves. I then fit the TRAPPIST-1d light curves with the ExoTEP pipeline to extract the transmission spectrum for each of the 5 observational nights. An atmospheric analysis was then carried out using the SCARLET framework. SCARLET calculates atmospheric models and synthetic planet spectra using a self-consistent photochemistry-thermochemistry model. In retrieval mode, SCARLET determines the best fitting set of parameters for a given observation set using a MCMC to explore the parameter space. To improve the speed, efficiency, and accuracy of the self-consistent models, I added a radiative-convective temperature structure calculation to the SCARLET framework, presented in Appendix A. Using the improved modelling framework, I modelled the transmission spectrum of TRAPPIST-1d and used the observed transmission spectra extracted with ExoTEP to run an atmospheric retrieval. Compiling the results from comparing the observed transmission spectrum with the modelled spectrum as well as the results of the atmospheric retrieval, I wrote the entirety of the paper presented within this thesis.

References

- [1] Bjoern Benneke and Sara Seager. ATMOSPHERIC RETRIEVAL FOR SUPER-EARTHS: UNIQUELY CONSTRAINING THE ATMOSPHERIC COMPOSITION WITH TRANSMISSION SPECTROSCOPY. *The Astrophysical Journal*, 753(2):100, July 2012.
- [2] Björn Benneke and Sara Seager. HOW TO DISTINGUISH BETWEEN CLOUDY MINI-NEPTUNES AND WATER/VOLATILE-DOMINATED SUPER-EARTHS. *The Astrophysical Journal*, 778(2):153, November 2013.
- [3] Björn Benneke, Ian Wong, Caroline Piaulet, Heather A. Knutson, Joshua Lothringer, Caroline V. Morley, Ian J. M. Crossfield, Peter Gao, Thomas P. Greene, Courtney Dressing, Diana Dragomir, Andrew W. Howard, Peter R. McCullough, Eliza M.-R. Kempton, Jonathan J. Fortney, and Jonathan Fraine. Water Vapor and Clouds on the Habitable-zone Sub-Neptune Exoplanet K2-18b. *The Astrophysical Journal*, 887(1):L14, December 2019.
- [4] David Charbonneau, Timothy M. Brown, David W. Latham, and Michel Mayor. Detection of Planetary Transits Across a Sun-like Star. *The Astrophysical Journal*, 529(1):L45–L48, January 2000.
- [5] David Charbonneau, Timothy M. Brown, Robert W. Noyes, and Ronald L. Gilliland. Detection of an Extrasolar Planet Atmosphere. *ApJ*, 568(1):377–384, March 2002.
- [6] Hubert Klahr and Wolfgang Brandner, editors. *Planet formation: theory, observations and experiments*. Number 1 in Cambridge astrobology. Cambridge University Press, Cambridge ; New York, 2006. OCLC: ocm64313073.
- [7] Michel Mayor and Didier Queloz. A Jupiter-mass companion to a solar-type star. *Nature*, 378(6555):355–359, November 1995.
- [8] Jason F. Rowe, Jeffrey L. Coughlin, Victoria Antoci, Thomas Barclay, Natalie M. Batalha, William J. Borucki, Christopher J. Burke, Steven T. Bryson, Douglas A. Caldwell, Jennifer R. Campbell, Joseph H. Catanzarite, Jessie L. Christiansen, William Cochran, Ronald L. Gilliland, Forrest R. Girouard, Michael R. Haas, Krzysztof G. Helminiak, Christopher E. Henze, Kelsey L. Hoffman, Steve B. Howell, Daniel Huber, Roger C. Hunter, Hannah Jang-Condell, Jon M. Jenkins, Todd C. Klaus, David W. Latham, Jie Li, Jack J. Lissauer, Sean D. McCauliff, Robert L. Morris, F. Mullally, Aviv Ofir, Billy Quarles, Elisa Quintana, Anima Sabale, Shawn Seader, Avi Shporer, Jeffrey C. Smith, Jason H. Steffen, Martin Still, Peter Tenenbaum, Susan E. Thompson, Joseph D. Twicken, Christa Van Laerhoven, Angie Wolfgang, and Khadeejah A. Zamudio. PLANETARY CANDIDATES OBSERVED BY *KEPLER* . V. PLANET SAMPLE FROM Q1–Q12 (36 MONTHS). *ApJS*, 217(1):16, March 2015.
- [9] Samaporn Tintanont, Maxwell A. Millar-Blanchaer, Ricky Nilsson, Dimitri Mawet, Heather Knutson, Tiffany Kataria, Gautam Vasisht, Charles Henderson, Keith Matthews, Eugene Serabyn, Jennifer W. Milburn, David Hale, Roger Smith, Shreyas Vissapragada, Louis D. Santos, Jason Kekas, and Michael J. Escuti. WIRC+Pol: A Low-resolution Near-infrared Spectropolarimeter. *PASP*, 131(996):025001, February 2019.
- [10] Jeremy Walsh, M. Kummel, and H. Kuntschner. Slitless Spectroscopy with HST Instruments. July 2010.
- [11] John C. Wilson, Stephen S. Eikenberry, Charles P. Henderson, Thomas L. Hayward, Joseph C. Carson, Bruce Pirger, Donald J. Barry, Bernhard R. Brandl, James R. Houck, Gregory J. Fitzgerald, and T. M. Stolberg. A Wide-Field Infrared Camera for the Palomar 200-inch Telescope. page 451, Waikoloa, Hawai'i, United States, March 2003.

First Article.

**Atmospheric characterization
of TRAPPIST-1d using the
new Palomar/WIRC+Spec
mode and a new
Karhunen-Loève Transform
based approach for
spectrophotometric extraction**

by

Jonathan Chan¹, and Björn Benneke¹

(¹) Institute for Research on Exoplanets and Department of Physics, Université de Montréal, Montréal, QC, Canada

This article will be submitted in *The Astrophysics Journal*.

ABSTRACT. The TRAPPIST-1 system provides an unprecedented opportunity to characterize the first potentially habitable planets outside our solar system. Here we present the first spectroscopic results from the WIRC+Spec mode of the recently upgraded Wide Field Infrared Camera (WIRC) instrument on Palomar Observatory. We describe a novel, Karhunen-Loève transform (KLT) based, PSF fitting approach to extract photometric and spectroscopic light curves from 5 transit observations of TRAPPIST-1d with defocused, slitless multi-object spectroscopy, and present the measured J band (1.1 to 1.4 μm) and K_s band (1.95 to 2.35 μm) transmission spectra. For faint sources such as TRAPPIST-1 ($J = 11.35$), the observations are background noise limited; our approach allows us to extract the PSF from the background, lowering the effect of background noise, and also enables us to robustly extract the spectrophotometric light curves from the spectra trace despite substantial atmospheric effects on the spectral trace. We then use a newly implemented self-consistent radiative-convective temperature structure calculation to model and compare various atmospheric scenarios with observations to date, ruling out cloud-free atmospheres on TRAPPIST-1d with metallicities lower than $300\times$ the solar value at 3σ confidence.

Keywords: planets and satellites: individual (TRAPPIST-1d) – planets and satellites: atmospheres

1. Introduction

The TRAPPIST-1 system presents us with a unique opportunity to study and characterize temperate, Earth-like planets in a compact M-dwarf system. Discovered in 2017, the seven planet system orbiting an ultracool M8-dwarf has analogies to our own solar system and hosts three habitable-zone planets [18, 17]. TRAPPIST-1d in particular receives nearly the same amount of incident flux as Earth ($1.043 \pm 0.060 S_{\oplus}$) and likely houses a rocky core with a radius of $0.784 \pm 0.23 R_{\oplus}$ and a mass of $0.297 \pm 0.39 M_{\oplus}$. TRAPPIST-1d is also 60% less dense than Earth at $0.616 \rho_{\oplus}$, indicative of a gaseous envelope in the possible form of a thick atmosphere [7, 19]. Furthermore, the predicted equilibrium temperature of 282.1 ± 4.0 K (assuming a null Bond albedo) provides the right conditions for liquid water and complex organic molecules, a condition generally regarded as necessary for the existence of life as we know it [7]. Nonetheless, much about the habitability of planets around M-dwarfs is still unknown and the atmospheric characterization of the TRAPPIST-1 planets will shape our understanding and knowledge of factors such as atmospheric loss and the high energy environment of M-dwarf systems [9, 8].

Habitable-zone planets around M-dwarfs offer two key advantages over habitable-zone planets around Sun-like stars for atmosphere studies [23]. The small diameter of the star ($0.11 R_{\odot}$ for TRAPPIST-1) results in larger transit signatures since the amplitude of transit and atmospheric signals scale inversely with the square of the stellar radius. The short orbital periods (5-11 days for the TRAPPIST-1 system) for habitable-zone planets around M-dwarfs also enables the observation of repeated transits within a relatively short time frame. *Kepler* determined that roughly 40% of M-dwarfs host small planets ($1-2 R_{\oplus}$) in the habitable-zone

[21, 10, 11, 26, 12]; moreover, M-dwarf stars account for approximately 75% of all stars, further amplifying their importance in the search for life in the universe [31].

One of the main challenges for atmospheric characterization of M-dwarf systems lies in the faintness of their host stars. Even with multiple visits, the small apertures of the current space based telescopes (*HST* and *Spitzer*) are unable to capture enough photons to precisely characterize the atmospheres of the TRAPPIST-1 planets. The solution to this problem therefore requires observations from larger telescopes, either in space, with the upcoming *James Webb Space Telescope (JWST)*, or on the ground.

Taking advantage of a recent upgrade to the Wide-field InfraRed Camera (WIRC) on the 200-inch Hale Telescope at Palomar Observatory, our team uses the new WIRC+Spec mode to probe the atmosphere of TRAPPIST-1d with transmission spectroscopy. WIRC is a near-infrared (J, H, and K_s bands spanning from 1.1 – 2.3 μm) camera with an 8.7' \times 8.7' field of view at a prime focus of f/3.3 [30]. WIRC was outfitted with an upgraded science-grade HAWAII-2 detector (previously mounted on Keck/OSIRIS) along with the installation of a polarization grating for spectropolarimetry (WIRC+Pol mode) in January 2017 [28]. 32 channel read-out electronics were also upgraded from the previous 4 channels, shortening the minimum exposure time from the previous 3.23s to 0.92s. An update to WIRC's guiding software also granted the capability to guide on elongated traces (shown in Figure 3.2) with a 2-D cross correlation algorithm, as opposed to the previous script, which fit 2-D Gaussians to the 0th order PSFs of the field. The guiding update allows for improved guiding on faint sources, where the 0th order PSF of the star may be too dim and lack the required contrast from the background to guide on. The WIRC+Spec mode expands the capabilities of the WIRC instrument to perform low resolution ($R \approx 100$) slitless spectroscopy in the near-infrared (J, H, and K_s bands). With the photon collecting potential of the 5.08 m diameter Hale telescope, WIRC+Spec provides the scientific community with the opportunity to perform spectroscopic measurements of faint targets from the ground.

In this paper, we present observations gathered by the WIRC+SPEC instrument on Palomar Observatory of TRAPPIST-1d. In Section 2 we describe the observations and data reduction process. We then present an innovative Karhunen-Loève Transform (KLT) based point spread function (PSF) fitting approach to obtaining our photometric and spectroscopic results in Section 3. We perform a joint light curve analysis in Section 4 and present the best-fit broadband and spectroscopic light curves as well as the derived transmission spectrum. Our atmospheric modelling analysis is described in Section 5. Section 6 then provides a discussion of our results within the context of previous and future studies.

2. Observations and data reduction

We obtained time series spectroscopy during five transits of TRAPPIST-1d with the WIRC+Spec instrument on Palomar Observatory between UT 2017 August 8 and October

Planet(s)	Wavelength [μm]	UT Start Date	Notes
d	1.1 – 1.3	2017/08/08	
c, d	1.1 – 1.3	2017/08/12	Non-overlapping transits of planets c and d
b, d	1.1 – 1.3	2017/10/12	Overlapping transit of planets b and d
d	1.95 – 2.35	2017/10/16	
b, d, e	1.95 – 2.35	2017/10/24	Transits of b and e overlap, followed by transit of planet d

Table 3.1. Summary of all transit observations of TRAPPIST-1d obtained for this study.

24. We perform three of our observing nights with the J band filter, which spans the range of 1.1 to 1.4 μm , and the other two nights with the K_s band filter, spanning the range of 1.95 to 2.35 μm ; we list our observations in Table 3.1. Previous work with *HST* has already covered the J band, however these observations are the first to probe the TRAPPIST-1 system with the K_s bandpass [6]. Each transit observation included up to an hour of baseline before and after the expected transit mid-point, with the exception of the visit on UT 2017/10/24, in which the transit of TRAPPIST-1d was close to dawn and thus we cut the baseline short after the transit to avoid overexposing the detector. We use 15 s exposure times for the J band observations, allowing us to obtain sufficient signal to noise while maintaining a good cadence of observation throughout the transit. We find that with the K_s filter, TRAPPIST-1 is extremely faint on the detector compared to the background and we opted to use an exposure time of 2 s with 5 co-adds in order to maintain acceptable signal to noise without oversaturating the detector. The WIRC+SPEC instrument spreads photons from the observed star across the detector into three visible orders, as depicted in Figure 3.2. We defocus the instrument, preventing saturation of the detector while also further spreading the light across multiple pixels to minimize detector effects from single pixels. For simplicity, we describe the zeroth, first, and second diffraction order traces as point spread functions (PSFs). Starting with the raw images, we use the WIRC_DRP pipeline built by Tinyant et al. to handle dark subtraction, flat fielding, and bad pixel determination [28]. We then built a custom WIRC+Spec data reduction module for the Python based Exoplanet Transits, Eclipses and Phase Curves (ExoTEP) pipeline [5, 3]. The main steps of the data reduction process are outlined in Figure 3.1. The extraction procedure differs slightly between J band observations and K_s band observations; we will describe the steps taken for J band observations, indicating the steps taken differently for the K_s band datasets where applicable.

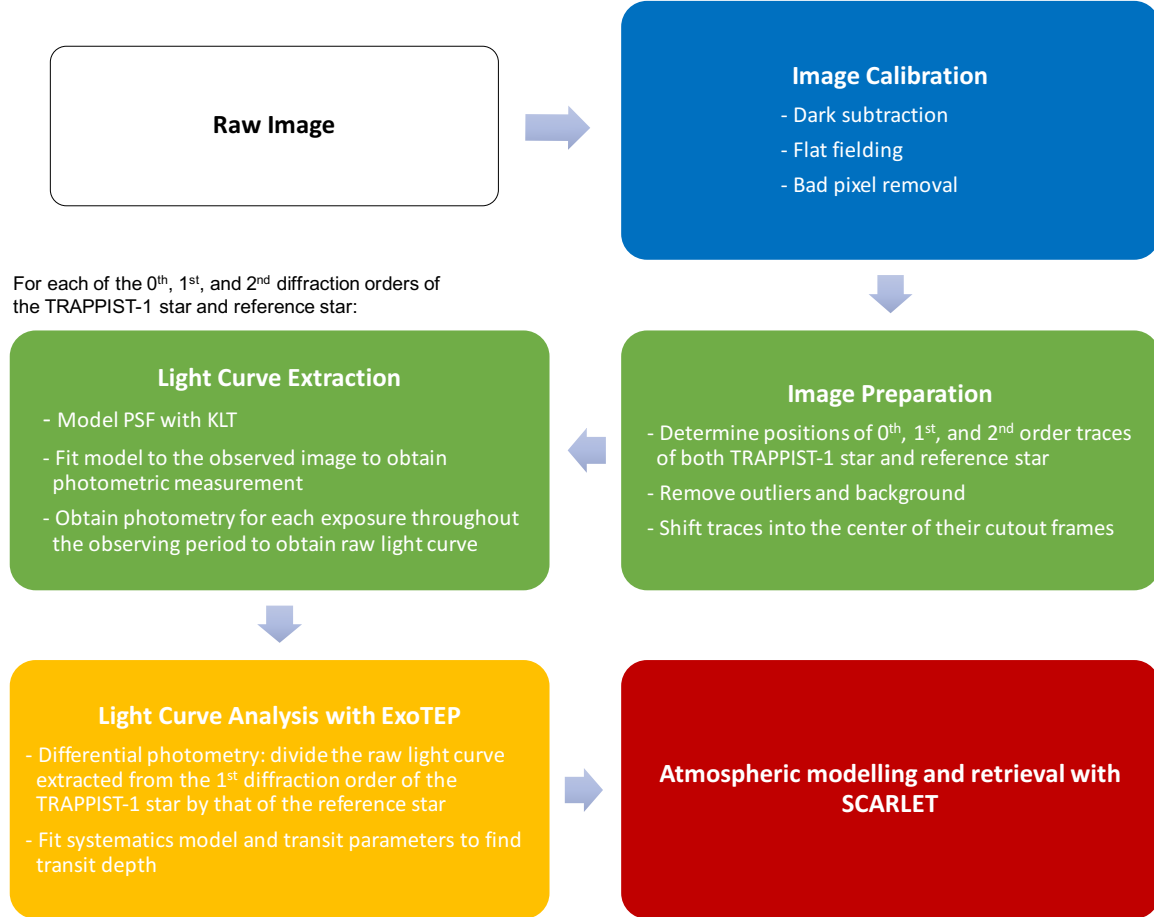


Fig. 3.1. Main steps of the data reduction process. The steps are colour coded according to the relevant framework used: image calibration was performed with the WIRC_DRP; image preparation and light curve extraction was performed with the custom built WIRC module for light curve extraction in the EXOplanet Transits Eclipses and Phasecurves (ExoTEP) framework; light curve analysis was performed with the transit parameter fitting module of ExoTEP; and atmospheric modelling and retrieval was performed with the Self-Consistent Atmospheric Retrieval for ExoplaneTs (SCARLET) framework.

2.1. Image calibration

We first calibrate the raw images with the WIRC_DRP pipeline designed for the WIRC+Pol spectropolarimetry mode of the WIRC instrument.

2.1.1. Dark subtraction and flat fielding

The WIRC_DRP median combines specified dark images to create a master dark frame, which is then subtracted from the science images. When the exposure times of the darks and science images are not the same, the darks are scaled to the exposure of the science image. Flat field images are median normalized and then median combined into a master flat frame, which is then divided from the science image. For J band observations, we use

dome flats with the lamp set to low while for K_s band, we use sky flats; we found that the dome lamp saturated the detector when used with the K_s filter.

2.1.2. *Bad pixel removal*

The WIRC_DRP uses both the dark and flat images to determine and remove bad pixels. First, pixels with unusual dark currents are determined from the dark frames by removing 5σ outliers from the mean of a median absolute deviation distribution of counts at each pixel. Dead pixels are then determined from the flat field images through a local and global filtering procedure. Local filtering detects isolated bad pixels by comparing neighboring pixels. The master flat frame is used to compute a map of standard deviations where the value at each pixel is determined by the standard deviation of a 11×11 box of pixels surrounding it. Pixels with values deviating by more than 3σ from surrounding values are rejected. Global filtering is then used to detect groups or patches of dead pixels, which may be overlooked by the local filtering since bad pixels are surrounded by equally bad pixels. First, a large-scale variation map is constructed by passing the master flat frame through a median filter. A map of pixel-to-pixel variations can then be determined by dividing the master flat frame by the large-scale variation map; 9σ outliers of the pixel-to-pixel map values are rejected. In total, Tinyanont et al. estimate $\sim 20,000$ pixels are rejected, accounting for 0.5% of the entire array [28].

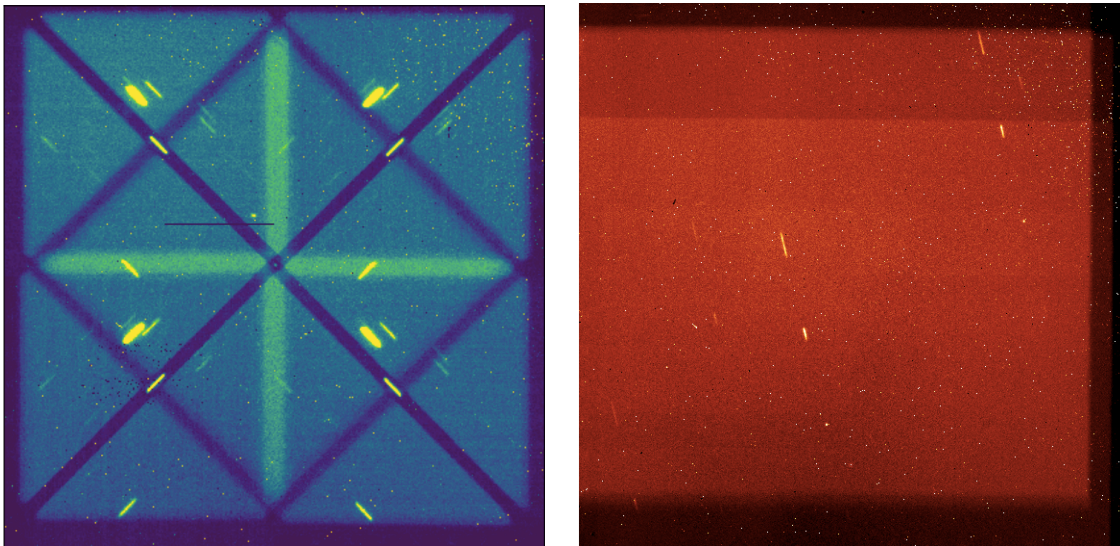


Fig. 3.2. Left: Sample raw frame from WIRC+Pol, adapted from Tinyanont et al. [28]. Right: Sample raw frame from WIRC+Spec. In the exposure taken by WIRC+Pol, the PSF is spread into symmetrical quadrants based on polarization.

2.2. Image preparation

The WIRC_DRP has the capability to perform photometric and spectral extractions for the WIRC+Pol spectropolarimetric mode, however the new WIRC+Spec mode introduces a different layout of the source across the detector, requiring a change in the reduction procedure. A comparison of a WIRC+Pol exposure and a WIRC+Spec exposure is shown in Figure 3.2. We create our own data reduction software as a module for the ExoTEP framework to customize the photometric and spectroscopic extraction to the WIRC+Spec exposures, allowing us to optimize each step to the specific needs of the new layout of the image.

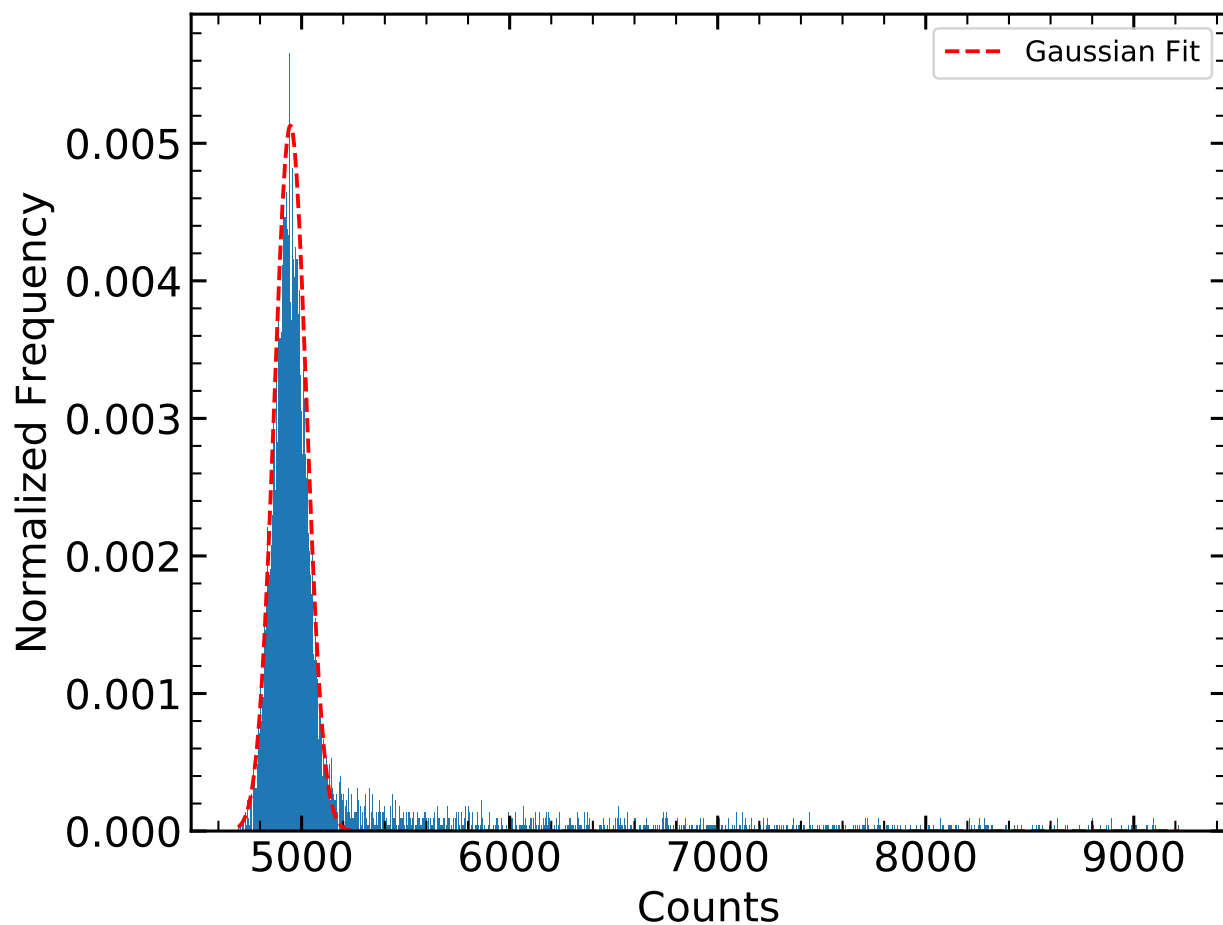


Fig. 3.3. Histogram of pixel count frequency for an example 1st order PSF cut-out frame. The peak corresponds to the high frequency of background pixels in the frame. A red dashed line illustrates the Gaussian fit describing the peak; the local background value for a given frame is estimated to be the mean of the Gaussian fit.

After calibrating the raw images with the WIRC_DRP pipeline, we prepare the images by finding the positions of the relevant traces, removing outliers and background, and shifting

the traces to the center of their respective frames to optimize the effectiveness and efficiency of the photometric and spectroscopic extraction.

We first obtain a data cube containing cutouts of the 0th, 1st, and 2nd order traces from the calibrated images in order to isolate the important/essentials while also reducing the amount of data being passed through the pipeline.

We specify the approximate locations of the center of the 0th order PSF and the 1st and 2nd order traces of the target and reference stars, which we determine manually for a single frame from the calibrated image. We then find the pixel coordinates of a box which encompasses the trace and save the data of the calibrated image inside the box within a numpy array. The pixel locations of the cutout edges are also saved for future use. This allows us to quickly access the relevant traces in a given frame without loading the entire calibrated image. The instrument’s guiding system does a sufficient job at keeping the traces within a few pixels between each frame so we are able to adjust the size of our cutouts to consistently capture the whole trace for every frame of a given night of observation. We choose a cutout size for each trace that allows for the trace to be framed by 10, 20, and 30 pixels for the 0th, 1st, and 2nd orders respectively, which gives us room to further isolate the individual traces in the next steps.

We use two different methods to determine the exact location of a trace on the detector, depending on the bandpass observed. For J band, we employ a flux-weighted positioning algorithm which iteratively finds the centroid of the 0th order PSF, given the calibrated image and a first guess [5]. The 1st and 2nd order traces are elliptically shaped, preventing us from using the flux-weighted positioning algorithm, which is designed for circular PSFs. Instead, we determine the vector describing the position of the 1st and 2nd order traces from the position of the 0th order PSF, which is a fixed value, intrinsic to the instrument’s grism. We can then calculate the position of the 1st and 2nd order traces from the 0th order PSF. For the K_s bandpass, the 0th order is much fainter on the detector and does not achieve the required contrast from the background noise to use a flux-weighted positioning algorithm effectively. Instead, we fit a 2-D Gaussian to the 0th order cutout of the calibrated image and use the mean x and y values of the fit to determine the position of the PSF. In some cases, the 0th order PSF of the star is completely lost in the background noise of a calibrated frame. To handle these exceptions, we also fit a 2-D Gaussian to the 1st order trace to determine their positions.

We estimate the local background for each trace by making a histogram of the flux values in each pixel contained by the cutout. We expect a double peaked distribution; background pixels will correspond to the lower flux peak while pixels containing the trace will be associated with the higher flux peak. In reality, since our cutout encompasses many more background pixels than pixels containing the trace, the histogram depicts a single peak for the background pixels along with a trailing tail corresponding to the trace pixels. We fit

a Gaussian distribution to the peak associated with the background pixels to determine the local background of a given trace. Figure 3.3 shows the histogram fit for a 1st order trace for a J band observation. We found that simply taking the median flux value of the cutout consistently overestimates the local background as opposed to fitting a Gaussian to the pixel flux values. Using the Gaussian fit, we then calculate the full width at half maximum (FWHM) of the 0th order PSF for the target and reference stars where:

$$\text{FWHM} = 2\sigma\sqrt{2\ln 2} \quad (2.1)$$

where σ is the standard deviation of the Gaussian fit.

At this point, we subtract the local background calculated at each time step for each source trace from the corresponding cutouts and also remove outlier pixels. We find the median and standard deviation images in time for each of the source traces and construct a mask where:

$$\text{Mask} = \frac{|\text{Image} - \text{Median Image}|}{\text{Standard Deviation Image}} \quad (2.2)$$

We then construct an outlier image where:

$$\text{Outlier Image} = \text{Mask} > \sigma \quad (2.3)$$

We reject 4.5σ outliers for J band observations. For K_s band nights, we forgo pixel outlier removal as we find the results do not change significantly even when rejecting up to 10σ outliers.

2.2.1. *Signal-to-Noise Ratio (SNR) determination*

We determine the expected SNR for aperture photometry in order to gauge the effectiveness of our photometric extraction. We first construct an aperture based on the order of the trace: we use a circular aperture for the 0th order PSF and an elliptical aperture for the 1st and 2nd order traces.

We manually approximate the radius of the 0th order PSF in pixels from a single calibrated frame of an observing night, adding up to 5 pixels to account for the PSF's change in size with time from atmospheric effects. We then construct a `CircularAperture` with the `photutils` package. To construct an elliptical aperture for the 1st and 2nd order traces, we must first determine the parameters describing the elliptical traces. We approximate the semi-major axis in pixels of the traces manually for a single calibrated image, adding up to 10 pixels to account for changes in the size of the trace throughout the observing night. Taking the vector describing the position of the 1st and 2nd order traces from the position of the 0th order PSF, we determine the angle θ between the trace and the horizontal axis, measured counter-clockwise from the horizontal axis, using simple trigonometry. Estimating the size of the trace by eye, we then calculate the length of the semi-major (a) and semi-minor (b)

axes from the rotation angle (θ) and the approximate size (r) of the trace as:

$$\alpha = \pi - \theta \quad (2.4)$$

$$a = \frac{r}{2\sin\alpha} \quad (2.5)$$

$$b = \frac{r}{2\tan\alpha} \quad (2.6)$$

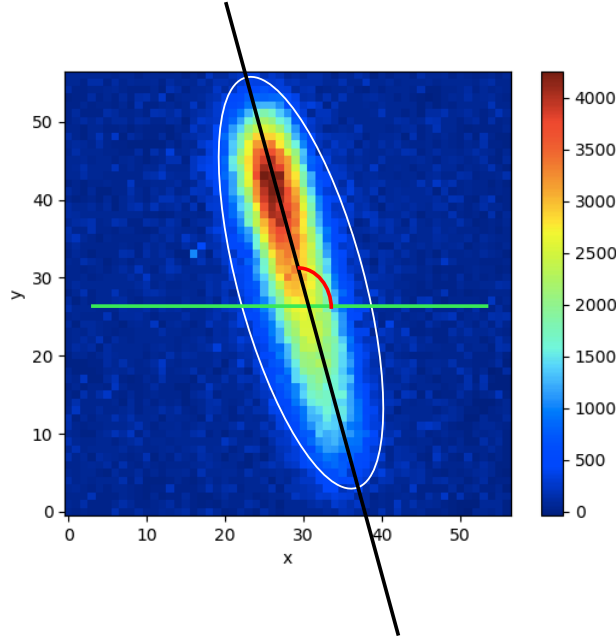


Fig. 3.4. Elliptical aperture for an example 1st order trace. The white ellipse depicts the aperture while the green line depicts the horizontal axis from which the angle of the trace θ (illustrated in red) is taken from. The colour scale depicts the number of counts read by the detector.

Given the parameters of the elliptical trace, we construct an `EllipticalAperture` with the `photutils` package; we show the aperture for a 1st order trace in Figure 3.4. We multiply the aperture mask by the background subtracted cutout, which we assume represents only the counts from the source, and taking into account the instrument’s gain of 1.2, we determine the accumulated source electrons detected by the charge-coupled device (CCD). Similarly, to determine the accumulated background electrons detected we multiply the aperture mask by the calculated local background value, taking into account a gain of 1.2 to convert from ADU to e^- . The expected SNR is then:

$$\text{SNR} = \frac{\text{Source } e^-}{\sqrt{\text{Source } e^- + \text{Background } e^-}} \quad (2.7)$$

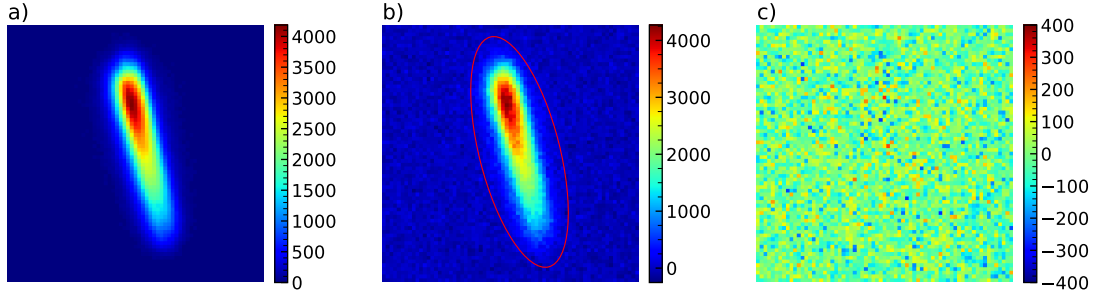


Fig. 3.5. Reconstruction of photometric PSF. Panel a) depicts the reconstructed PSF model. Panels b) shows the original background subtracted exposure. The red outline corresponds to the aperture used to perform aperture photometry. A residual image obtained by subtracting the background subtracted exposure by the reconstructed PSF is presented in c).

and the individual scatter for the target and reference stars in parts per million (ppm) are:

$$\text{Scatter}_{\text{Source}} = \frac{1}{\text{SNR}} \times 10^6 \quad (2.8)$$

To account for the division of the target star light curve by the reference star light curve with differential photometry, which reduces the effects of changes in the Earth’s atmosphere, we perform error propagation to determine the expected light curve scatter to be:

$$\text{Total Scatter} = \sqrt{\text{Scatter}_{\text{Target}}^2 + \text{Scatter}_{\text{Reference}}^2} \quad (2.9)$$

3. Light Curve Extraction

Our photometric extraction employs a novel PSF fitting approach with a Karhunen-Loève (KL) transform—closely related to the widely known principal component analysis (PCA)—which is commonly used in image processing. We use the KLT to construct a PSF model, which we then scale to the data in order to extract our photometric and spectroscopic data. Compared to aperture photometry, this technique allows us to reduce the effects of background noise on our extraction and also provides a robust approach to extracting spectroscopy from the irregularly shaped PSFs.

3.1. Photometric PSF Fitting

We adapt the KL image projection (KLIP) algorithm, first introduced by Soummer et al., in our PSF fitting procedure [27, 29]. Soummer et al. initially introduced the the KLIP algorithm as a direct imaging technique, allowing for the optimal subtraction of a PSF from a given image, thus exposing any nearby faint astronomical signals (planets or other binaries) around the annulus of a target star [27]. KLIP first uses a KLT to model the PSF of a given frame, then subtracts the PSF from the original frame to find otherwise undetectable

sources. Similarly, we tailor the algorithm to accurately model the irregularly shaped PSFs of the 1st and 2nd order traces from WIRC+Spec, which then allows us to remove unnecessary background from our photometry. The KLIP algorithm is as follows:

- (1) The target image and set of reference images are chosen and the average value of each frame is subtracted so they have zero mean over each frame. In our reduction, all exposures which do not contain the transit are used as the set of reference images; exposures within the transit event are excluded from the reference since they will vary from the baseline.
- (2) The Karhunen-Loève transform is computed to find the eigenvectors and eigenvalues of the covariance matrix of the reference set.
- (3) The number of components used to estimate the target image is chosen. We use 20 components to rebuild the PSFs of our J band observations and 100 components to rebuild the PSFs of our K_s band observations; these numbers yielded the lowest photometric light curve scatter.
- (4) The best estimate of the target image PSF is then computed from the projection of the target image on the Karhunen-Loève basis. Figure 3.5 illustrates a comparison of the modelled PSF with the original target image.
- (5) The model target PSF is subtracted from the original target image, leaving only the background and, for direct imaging applications, any faint astronomical signals otherwise masked by the host PSF. The result of this step is depicted in Figure 3.5 as the residuals of subtracting the original image from the modelled PSF.

The KLT used in this algorithm share close similarities to the widely known principal component analysis (PCA) method; some authors use both terms interchangeably. Fundamentally, both methods are dimensionality reduction techniques which allow a set of data to be described using fewer variables than originally, however the process to achieve this objective are slightly different between the KLT and PCA. The KLT has two defining features which are not necessarily characteristic of a PCA. First of all, KLT completely decorrelates the signal. That is to say, after KLT, the covariance matrix of the signal is diagonalized and the covariance between any two components is always zero. Secondly, the geometric interpretation of the KLT is merely a rotation of the coordinate system while the PCA first shifts the origin of the coordinate system before performing the same rotation. A more in depth comparison of the KLT and PCA can be found in Gerbrands and Soummer et al. [16, 27].

We first perform a sub-pixel shift with the calculated positions of the 0th, 1st, and 2nd order traces of the TRAPPIST-1 star, shifting the position of the cut-out frame so that the PSF is in the center of each cut-out. This removes the variability of the PSF’s movement on the detector from the KLT; we treat this effect with an instrumental systematics model later on. Following steps 1 to 4 of the KLIP algorithm, we extract a model PSF for each cut-out

frame, assuming the PSF follows a Gaussian noise model. We then normalize the model PSF to sum to unity and scale the normalized model PSF to best fit the original calibrated image as follows:

$$\text{Data} = A \times \text{Model} \quad (3.1)$$

Where A is a scaling factor. Following Gagné et al., we define the goodness-of-fit as:

$$G = \sum_{i=1}^N \left(\frac{\text{Data} - A \times \text{Model}}{\sigma_{\text{Data}}} \right)^2 \quad (3.2)$$

We then fit a scaling factor A that minimizes $G_{k,j}$ for each image [14]. The best fit for A can be expressed analytically as:

$$A = \frac{\sum_{i=1}^N \frac{\text{Data} \times \text{Model}}{\sigma_{\text{Data}}^2}}{\sum_{i=1}^N \frac{\text{Model}^2}{\sigma_{\text{Data}}^2}} \quad (3.3)$$

During the reduction we do not consider a measurement error, σ_{Data} , for each image, but instead derive error from the photometric scatter of the light curve after photometry has been extracted. Ignoring σ_{Data} , A can then be further simplified to:

$$A = \frac{\sum_{i=1}^N \text{Data} \times \text{Model}}{\sum_{i=1}^N \text{Model}^2} \quad (3.4)$$

We take A to be the photometric measurement of the cut-out frame. For each exposure, we use this technique to obtain A for each of the 0th, 1st, and 2nd order PSFs for the TRAPPIST-1 star. We then repeat the process for the 0th, 1st, and 2nd order PSFs of a nearby reference star; thus, we obtain 6 time-series raw light curves in total for each night of observation: 3 for the TRAPPIST-1 star and 3 for the reference star, based on the 0th, 1st, and 2nd order PSFs. In Section 4, we then use the light curves of the 1st order PSFs from the TRAPPIST-1 star and the reference star to perform differential photometry.

We find that our PSF fitting method is consistent with results obtained with aperture photometry: a comparison of calculated light curve scatter from the two techniques is presented in Table 3.2. These values are calculated from the light curve extracted from the 1st order PSF after differential photometry (eg. the TRAPPIST-1 1st order light curve has been divided by that of the reference star). For the J band observations, the photometric scatter values of the light curves obtained from aperture photometry and our PSF fitting method are within 5% difference. The K_s band observations a higher percent difference, which may be a result of the overall increased noise and variability in the observations. We believe the increased scatter on UT 2017/10/24 is due to the lack of baseline after the transit, which was cut short due to the brightening of the sky as dawn approached. We show the raw white light curves of UT 2017/08/12 from the 1st order PSF, extracted using both aperture photometry and PSF fitting, in Figure 3.6. The light curves extracted with PSF fitting exhibit

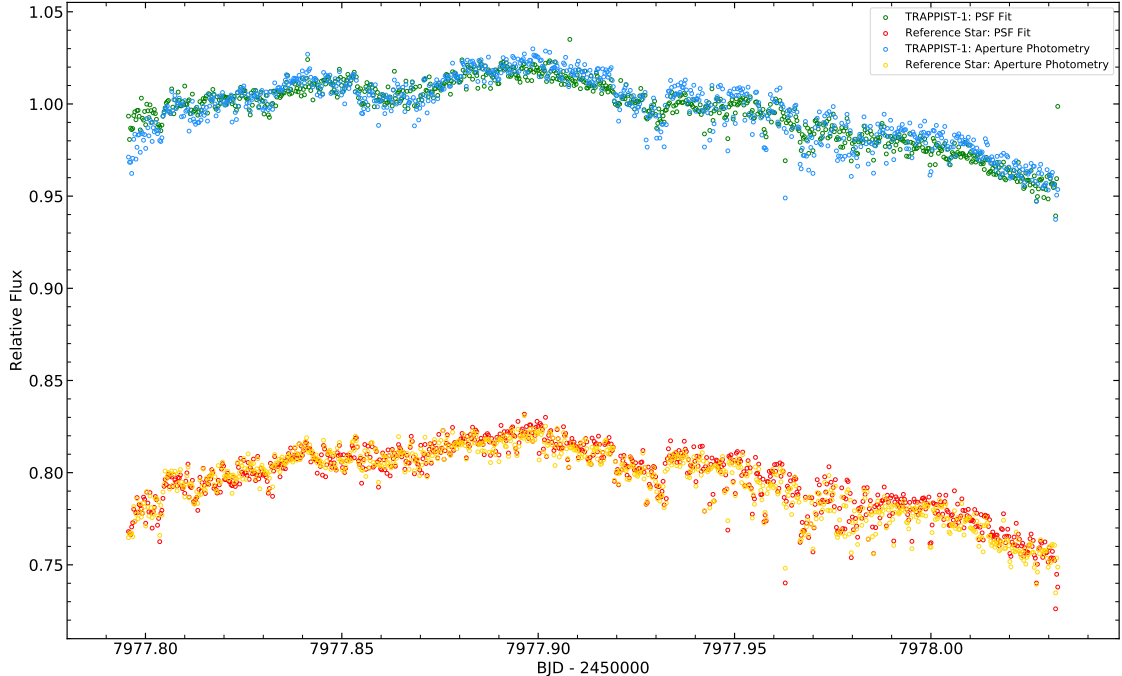


Fig. 3.6. Comparison of normalized white light curves extracted from the 1st diffraction order PSF of the TRAPPIST-1 and reference star using the PSF fitting method and aperture photometry for the night of UT 2017/08/12. The relative flux of the reference star has been offset for clarity. The TRAPPIST-1 light curve is later divided by the reference star light curve in order to remove atmospheric effects, as described in Section 4.

UT Night	Band	Aperture Photometry	PSF Fitting	Percent Difference
2017/08/08	J	4998 ppm	5148 ppm	2.956%
2017/08/12	J	2624 ppm	2532 ppm	3.569%
2017/10/12	J	3460 ppm	3338 ppm	3.589%
2017/10/16	K _s	5892 ppm	5201 ppm	12.46%
2017/10/24	K _s	6081 ppm	6844 ppm	11.81%

Table 3.2. Comparison of the photometric scatter of the TRAPPIST-1 light curves using the 1st order PSF after differential photometry. Results are shown using both aperture photometry and PSF fitting methods. J band observations are consistent to within 5% difference while K_s band observations are consistent to within 13% difference.

less variability, as confirmed by Table 3.2. Our PSF fitting method also provides us with a robust approach to extracting the spectroscopic light curves, which we discuss below.

3.2. Spectroscopic PSF Fitting

To perform spectroscopic extraction on a given frame, we obtain the wavelength solution of the detector and using the wavelength at each pixel location, we construct a forward model which retrieves the best fitting spectroscopic PSF model. We first determine the distribution

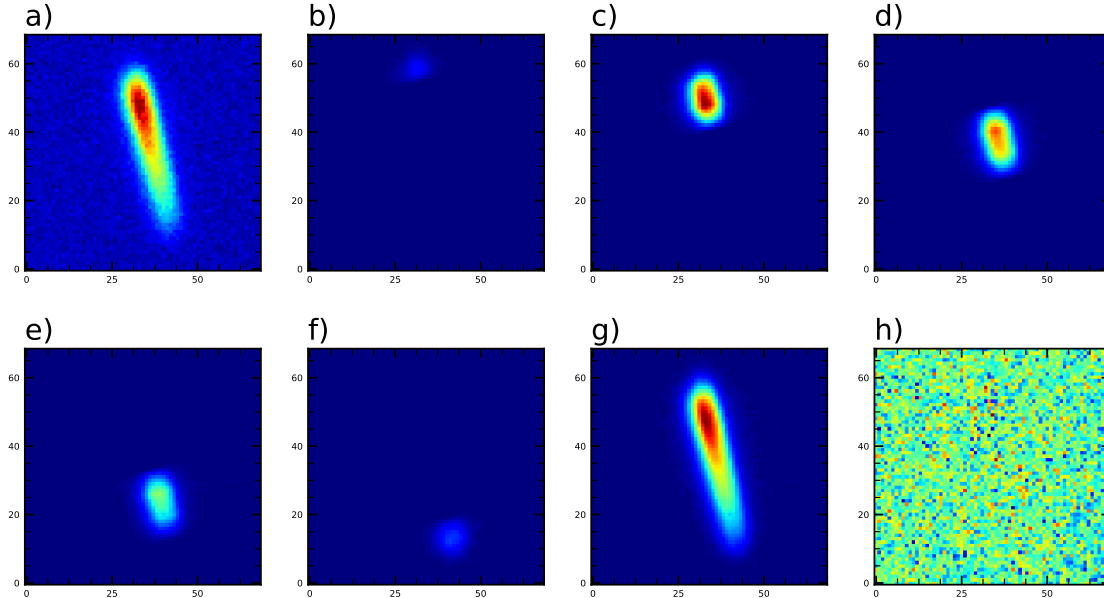


Fig. 3.7. Reconstruction of spectroscopic PSF for 5 spectral bins. Panel a) depicts the original background subtracted exposure. Panels b) to f) show the 5 spectral bins which are then added together to construct the PSF model, depicted in g). A residual image obtained by subtracting the background subtracted exposure (a) by the reconstructed PSF (g) is presented in h). Panels a) to g) are on the same colour scale.

of flux across the source trace as a function of pixel location and match the falloff of counts with respect to pixel location to the filter transmission cutoffs. At low spectral resolution, we can assume that wavelength is a linear function of pixel position and thus we obtain the absolute wavelength solution [28]. Figure 3.8 shows the wavelength grid for a J band observation.

From there, we take the model PSF constructed by the KLIP algorithm and divide it into segments according to the number of spectroscopic bins. We pass each segment through a Gaussian convolution and use the wavelength solution to determine the segment’s wavelength coverage based on its pixel location on the detector. We then construct a forward model which combines the separate spectroscopic segments to reconstruct the model PSF and scales each segment by a scaling factor A . For N spectroscopic bins, the data can be fit as:

$$\text{Data} = A_1 \times \text{Segment}_1 + A_2 \times \text{Segment}_2 + \dots A_N \times \text{Segment}_N$$

Our forward model handles the scaling of each PSF model segment and combines each segment to reconstruct the PSF model. We then use a least squares minimization to determine the best fitting scaling values for all spectroscopic PSF segments at the same time, ultimately reconstructing the a PSF that best estimates the original data. A visualization of the spectroscopic PSF reconstruction for 5 spectral bins is shown in Figure 3.7. For our

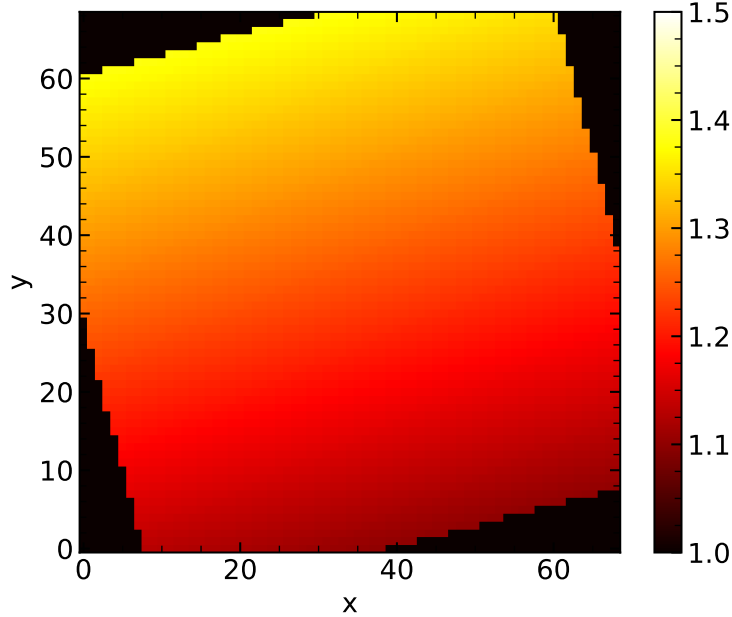


Fig. 3.8. Wavelength solution for a sample 1st order PSF of a J band exposure, calculated by matching the counts on the detector with the filter transmission cutoffs.

analysis, we chose to extract 10 spectroscopic bins from our J band observations and 5 bins from the K band observations. This allows us to optimize spectral resolution while maintaining reasonable signal-to-noise within each bin. Again, we take A for each bin as the spectrophotometric measurement and we obtain A for the 0th, 1st, and 2nd order PSF's of the TRAPPIST-1 star as well as a nearby reference star for each exposure, ultimately giving us 6 spectroscopic time-series light curves: 3 for TRAPPIST-1 and 3 for the reference star.

In comparison to an aperture photometry approach, PSF fitting allows us to robustly and accurately extract spectroscopic data from the 1st and 2nd order traces. For an aperture photometry approach, we create N rectangular apertures to capture the counts in each spectroscopic bin, as shown in Figure 3.9. This allows us to obtain a rough estimate of the flux in each bin, however it does not account for the Gaussian profile of the flux in each wavelength bin. Although it is also possible to rotate each bin to intersect the PSF perpendicularly, we find this approach makes lining up each rectangular bin along the PSF without gaps or missed pixels to be unnecessarily difficult and inefficient. Furthermore, this approach requires much more manual input to determine the best apertures based on the size of the PSF, which can vary throughout the night due to atmospheric effects and changes from one observational night to another. As the PSF grows, the wavelength solution changes as well; we find that maintaining precise wavelength bins is unmanageable with an aperture approach since the size of the aperture would need to change constantly.

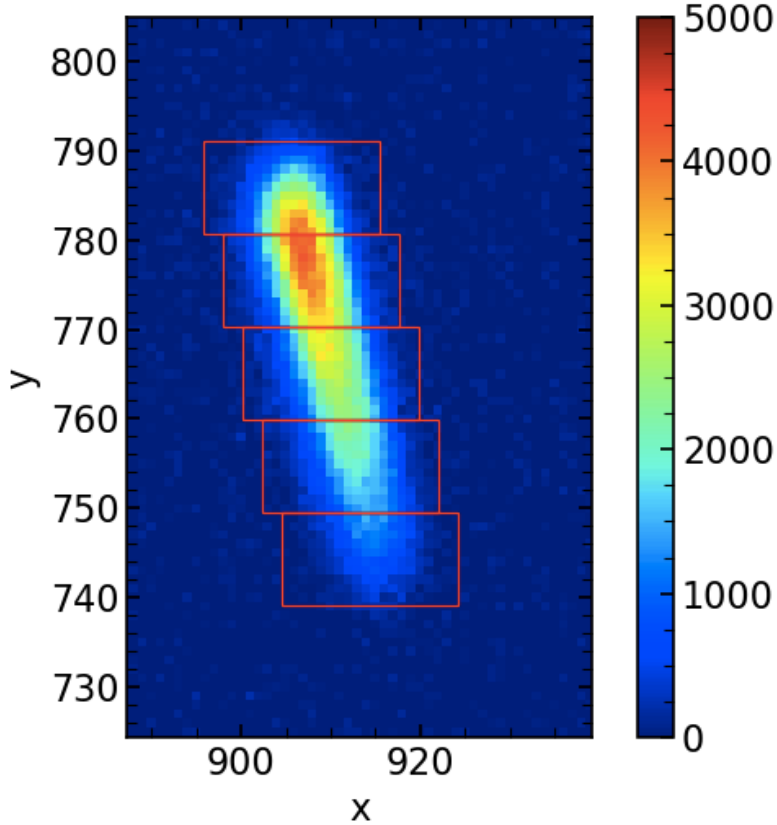


Fig. 3.9. Rectangular apertures for spectroscopy, shown in red. Due to atmospheric effects, the PSF may grow or shrink on the detector. In order to maintain precise wavelength bins based on the wavelength solution, the apertures would also need to grow or shrink in size, leading to imprecise spectrophotometry.

4. Light curve analysis

We carry out our light curve analysis of the TRAPPIST-1d observations within the ExoTEP analysis framework [5, 3]. In our analysis, we use the photometric results of the 1st order PSFs, which are the brightest of the three differential orders per star, and yield the cleanest light curves. We perform differential photometry, dividing the light curve extracted from the 1st order source of the TRAPPIST-1 star by that of the reference star in order to broadly account for the effects of the atmosphere. We were unable to fit the overlapping transit of planets b and d well, and thus chose to omit the visit of UT 2017/10/12 from our analysis.

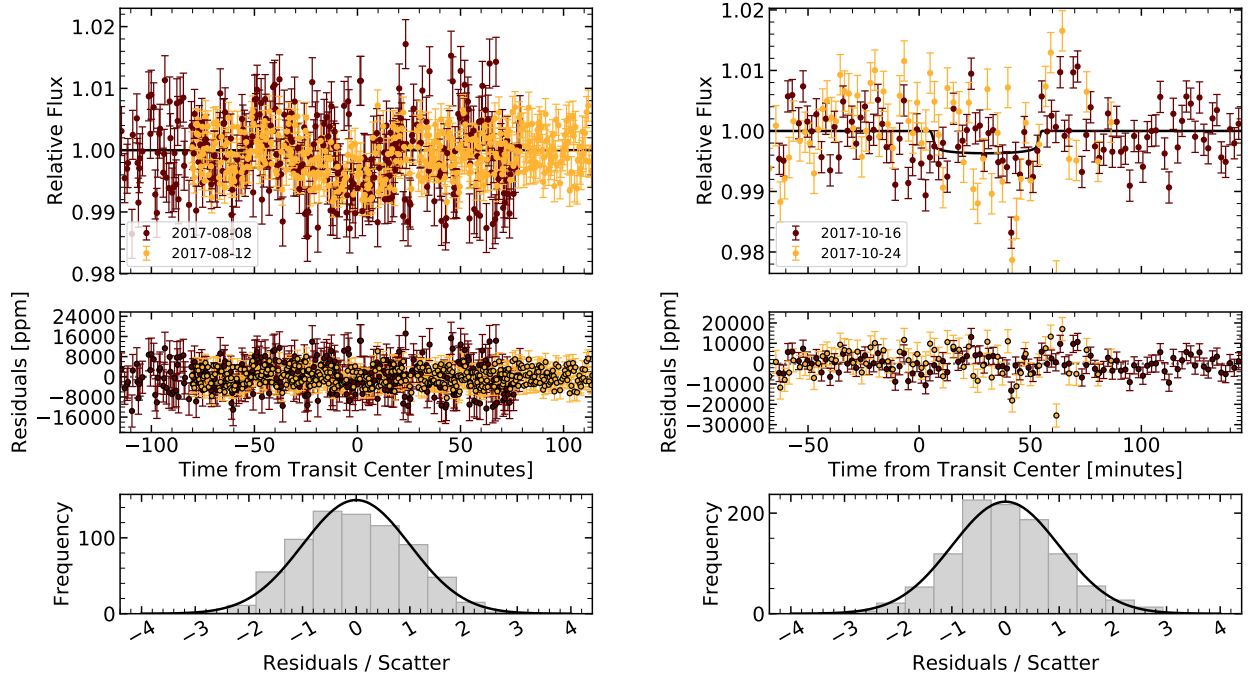


Fig. 3.10. Left: Joint fit broadband light curve for the two J band Palomar transit observations of TRAPPIST-1 d. The top panel shows the best fitting model light curves (black curve), overlaid with the systematics-corrected data (circles), binned to 30 seconds. Residuals from the light curve fits are shown in the middle panels. The bottom panels shows a histogram of the residuals normalized by the fitted photometric scatter parameter for each respective transit. The residuals follow the expected Gaussian distribution for photon noise limited observations. Right: Similarly for the two K_s band transit observations of TRAPPIST-1 d, binned to 120 seconds.

4.1. Photometric Light Curves

The astrophysical model is derived using the `batman` package, described in Kreidberg (2015) [22], which models the transit signal of a planet as it crosses in front of its host star. We then fit the transit depth and mid-transit time of the time series photometry with a Markov chain Monte Carlo (MCMC), as described in Benneke et al. [5, 3], fixing all other parameters of the planet and star to the values reported in the literature [7, 6]. The resulting joint white light curves are depicted in Figure 3.10.

4.1.1. Instrumental Systematics

The systematics model aims to correct for instrumental and atmospheric time-correlated effects in the dataset. For the J band measurements, we find the expansion and contraction of the PSF on the detector due to atmospheric effects has the greatest influence in the photometry. To account for this, we model the systematics with the following analytical function:

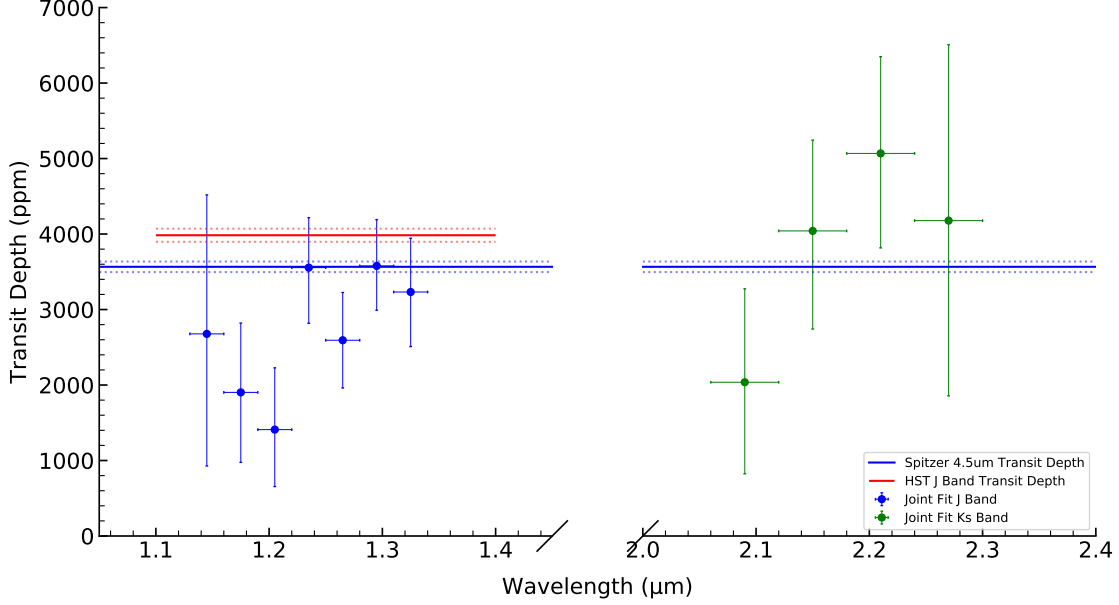


Fig. 3.11. Blue and green data points show transit depth measurements from the J band observations and K_s band observations, respectively; vertical bars indicate the 1σ transit depth uncertainties while the horizontal bars indicate the wavelength ranges of the measurements. A blue solid line represents the *Spitzer* 4.5 μm transit depth with the dashed representing the upper and lower bounds [7]. Similarly, the red solid and dashed lines depict the average *HST* J band transit depth and error [6]. Our results are consistent with the *HST* and *Spitzer* transit depths, albeit with larger uncertainties.

$$S_{J\text{Band}}(t) = c + v_x x(t) + v_y y(t) + v_{FWHM} FWHM(t) \quad (4.1)$$

where c is a normalization constant, v_x is the slope of the x position with respect to time $x(t)$, v_y is the slope of the y position with respect to time $y(t)$, and v_{FWHM} is the slope of the full width at half maximum (FWHM) of the PSF as a function of time $FWHM(t)$. Detrending against the FWHM and the position of the PSF on the detector allows us to correct for changes in pixel sensitivities across the detector as well as atmospheric effects which may cause the PSF to grow or shrink throughout the period of observation.

For K_s band observation sets, we found the 0th order PSF did not always offer high enough contrast above the background noise to extract the FWHM, thus we instead correct for positional effects on the detector and the effects of background with respect to time. The systematics function is as follows:

$$S_{K_s\text{Band}}(t) = c + v_x x(t) + v_y y(t) + v_b bkg(t) + v_t t \quad (4.2)$$

where v_b is the slope of the changes in background with time $bkg(t)$ and v_t is the slope of the time scale, t . As can be seen in Figure 3.10, the K_s band transit observations show a

fair amount of systematic noise, visible both in the best fitting model light curve and in the residuals. This suggests that the systematics model insufficiently describes the instrumental and atmospheric time-correlated effects. The justification for this systematic noise is twofold. Firstly, due to the nature of the K_s band observations, the systematics model could not take into account the FWHM of the PSF, depriving the systematics model of corrections to the light curve based on variations in the PSF’s size. Secondly, the effects of the brightening sky as dawn approached during the transit observation on the night of UT 2017/10/24 may not have been well constrained by the systematics model, resulting in increased noise and systematic variability.

4.2. Spectroscopic Light Curves

We fit the spectroscopic light curves using the same model as the broadband light curves. We perform a joint MCMC fit of a transit and systematics model to all spectroscopic light curves, keeping the mid-transit time fixed to the best-fitting value from the broadband light curve to obtain posterior distributions for individual planet-to-star radius ratios in each spectroscopic bin. The transmission spectrum from our joint analysis of the J band and K_s band nights are given in Figure 3.11. The first bin (1.10 - 1.13 μm) and the last two bins (1.34 - 1.40 μm) of the J band observations and the first bin (2.0 - 2.06 μm) of the K_s band observations hold too little light due to the WIRC filter cut offs, resulting in noisy, unreliable light curves. We choose to exclude these points in our analysis.

5. Atmospheric Analysis

Atmospheric characterization of exoplanets is firmly dependent on the accuracy of the available model spectra. Advances in transit, eclipse, and direct imaging techniques and instrumentation constantly improve the quality and resolution of observed spectra. To explain the observed phenomena, modern atmospheric models follow two main approaches: forward modelling and retrievals [25, 2]. Forward models take into account the numerous physical processes at work within an atmosphere and generate simulated spectra based on a defined set of model parameters. Algorithms describing the atmospheric chemistry, energy transport mechanisms, clouds and hazes, and thermodynamics, etc. all play a complex role in calculating a self-consistent atmospheric model. Model spectra can then be compared and fit to the observed spectra by tweaking the model parameters until a desired match is obtained. Forward models excel in their ability to take into account prior knowledge of atmospheric chemistry and physics in order to gain physical insights into the modeled atmospheric processes, however they lack the ability to treat observational uncertainties and the models are based upon the assumption that our understanding of the physical and chemical processes are absolute [15, 1, 4]. In comparison, atmospheric retrieval methods attempt to fit models

to observed spectra by fitting free parameters which describe the atmospheric composition and characteristics. The method explores the model parameter space to find the best fitting model, deriving statistically robust uncertainties for derived chemical abundances and temperature profiles. One of the major downfalls associated with this method arises from the lack of physical consistency in the derived models. The retrieved models take chemical abundances and temperature profiles as free parameters and do not take into account the atmospheric processes at play. As a result, retrievals can yield unphysical outcomes when the retrieved temperature profiles are incompatible with the retrieved abundances [2].

5.1. Self-consistent modelling

In this work, we build upon the Python based Self-Consistent Atmospheric Retrieval framework for Exoplanets (SCARLET), which tries to bridge the gap between forward modelling and retrievals in order to constrain the uncertainties of relevant model parameters while calculating self-consistent atmospheric models to match observed data [1, 4, 2]. SCARLET calculates atmospheric models and synthetic planet spectra using a self-consistent photochemistry-thermochemistry model. Chemical abundances are determined using either a chemical equilibrium model, which minimizes Gibb’s free energy in each layer of the atmosphere, or a more detailed kinetics and transport model, which solves the coupled mass-continuity equations for each molecular species.

5.1.1. *Self-consistent temperature structure using the Rybicki method*

Critical to both forward models and retrievals, the treatment of a planet’s temperature structure dictates how and where energy is deposited within the atmosphere, affecting the possible chemical abundances, as well as the resulting model spectra.

Here, we implement the highly efficient Rybicki method into SCARLET to converge upon radiative-convective equilibrium. Previously, the equations of hydrostatic equilibrium, radiative transfer, radiative equilibrium and convection were solved individually, iterating between all equations to converge upon the temperature structure. As a result, the iterations were slow and thousands of iterations were required to relax towards the solution. In contrast, the Rybicki method linearizes all equations, thus allowing us to solve all structural equations simultaneously. The method is essentially a Newton-Raphson procedure for solving a set of nonlinear algebraic equations, and allows us to efficiently advance towards the converged solution much more quickly; in most cases only 10 to 20 iterations are required as opposed to thousands previously. We describe the Rybicki method in its entirety in Appendix A, Section A.4. These additions allow SCARLET to evolve as a next-generation atmospheric modelling framework, increasing the speed, efficiency, and accuracy of the self-consistent temperature profile calculation. This code self-consistently handles both incident stellar irradiation as well as internal flux, enabling it to treat a variety of possible applications.

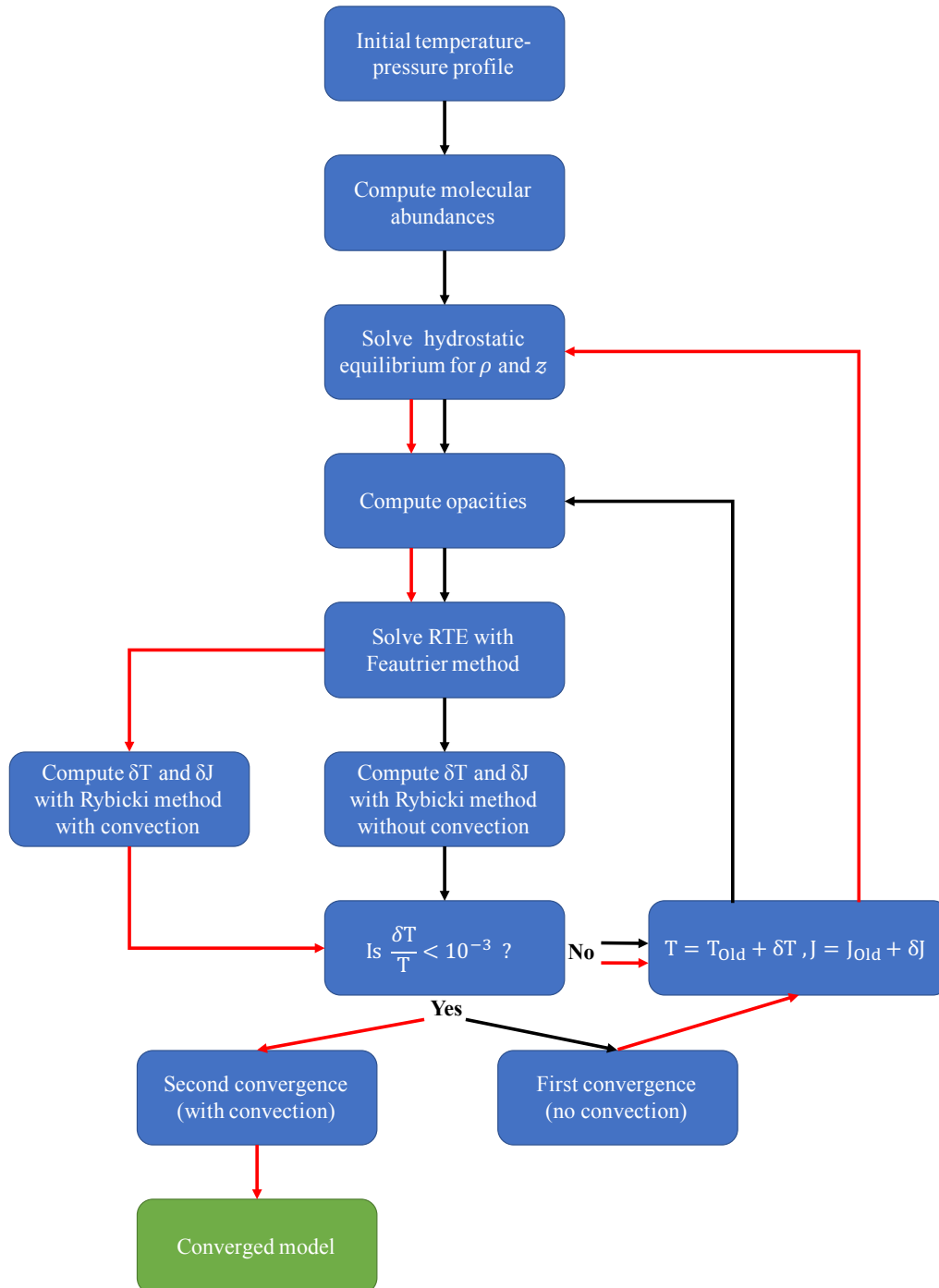


Fig. 3.12. The SCARLET temperature structure calculation framework. The black arrows represent the first run through of the code, which does not implement convection or changes to hydrostatic equilibrium, which solves stability issues. After the first convergence of the temperature structure, the code follows the flowchart with the red arrows and the equations of hydrostatic equilibrium and radiative-convection equilibrium are solved.

An overview of the basic architecture of the self-consistent temperature structure calculation framework is illustrated in Figure 3.12. An initial temperature profile is either generated from the planet properties (eg. a uniform temperature structure based on the exoplanet’s equilibrium temperature), or otherwise parameterized by the user. After initializing a pressure grid representative of the layers of the atmosphere, the chemical composition is calculated with a self-consistent photochemistry-thermochemistry model for the initial temperature profile and the hydrostatic equilibrium equations are solved to obtain the altitude, mean molecular mass, and scale heights for each atmospheric layer. Extinction, absorption, and scattering coefficients at each layer for every wavelength are then interpolated from pre-calculated opacity tables. We then use the Feautrier method to solve the radiative transfer equation (RTE) for every layer and wavelength, described in Section A.3 of Appendix A[13]. From there, the Rybicki method handles the modelling of energy transport, solving the equations of radiative equilibrium to yield a correction to the temperature for each layer which will bring the temperature profile closer to the self-consistent solution (see Section A.4 of Appendix A)[1, 24]. We apply the temperature correction to the previous temperature profile and then recalculate the opacities based on the new temperature profile. The RTE is then resolved for the new temperature profile and a new temperature correction is calculated with the Rybicki method. We iterate until the temperature correction is sufficiently small to converge upon an initial solution. For stability, we do not account for convective equilibrium or change the chemical composition until the algorithm has converged upon a temperature structure which first satisfies the conditions for radiative equilibrium. After an initial solution has been found, we update the chemical composition using the photochemistry-thermochemistry model and solve the hydrostatic equilibrium equations at each iteration. We then also account for convection within our energy transport algorithms, allowing us to finally converge upon a self-consistent temperature profile. Convection is treated within the Rybicki method using mixing length theory. Appendix A provides a detailed description of the self-consistent temperature calculation.

5.1.2. *Self-consistent Models of TRAPPIST-1d*

We construct a host of possible combinations of atmospheric metallicity and cloud top pressures. Figure 3.14 depicts the calculated self-consistent temperature structures for each case. We choose a solar carbon to oxygen (C/O) ratio, a Bond albedo of 0.1, and a heat distribution factor of 0.25 and use SCARLET’s chemically consistent mode to compute a model spectrum at radiative-convective and chemical equilibrium. SCARLET first computes the molecular abundances in chemical and hydrostatic equilibrium and the molecule opacities. A self-consistent temperature structure is then calculated using the Feautrier method to solve the RTE and the Rybicki method to converge upon radiative-convective equilibrium. We treat clouds as a gray opacity source parameterized by the cloud top pressures, which

we vary between 1 mbar and 10^{10} mbar. We also probe metallicities between $0.01 \times$ solar to $100 \times$ solar. The resulting spectra are computed with line-by-line radiative transfer. We compare the calculated spectra to the WIRC (this work), *HST* [6], and *Spitzer* [7] observations to date with a reduced chi square (χ^2), as illustrated in Figure 3.13. The uncertainty of our results do not allow us to constrain the atmosphere of TRAPPIST-1d better than previous studies using *HST*, however our results are consistent in ruling out cloud-free, hydrogen dominated atmospheres. Lower reduced χ^2 s are found with high metallicity cases and lower cloud top pressures, indicating a better fit to the observations.

5.2. Atmospheric Retrieval

Using the *HST*, *Spitzer*, and Palomar observations, we also run a retrieval with SCARLET’s chemically consistent mode. In retrieval mode, SCARLET determines the best fitting set of parameters for a given observation set using a MCMC to explore the parameter space. We take atmospheric metallicity, C/O ratio, cloud top pressure, and the vertical temperature structure as free fitting parameters with log-uniform priors. For a probed set of parameters, SCARLET calculates a self-consistent atmospheric model and generates a transmission spectra; the modelled spectra is then compared with the observations to evaluate its likelihood. With a Bayesian MCMC analysis combined with the chemically-consistent atmospheric forward model, we found we were able to constrain the metallicity and cloud top pressure of

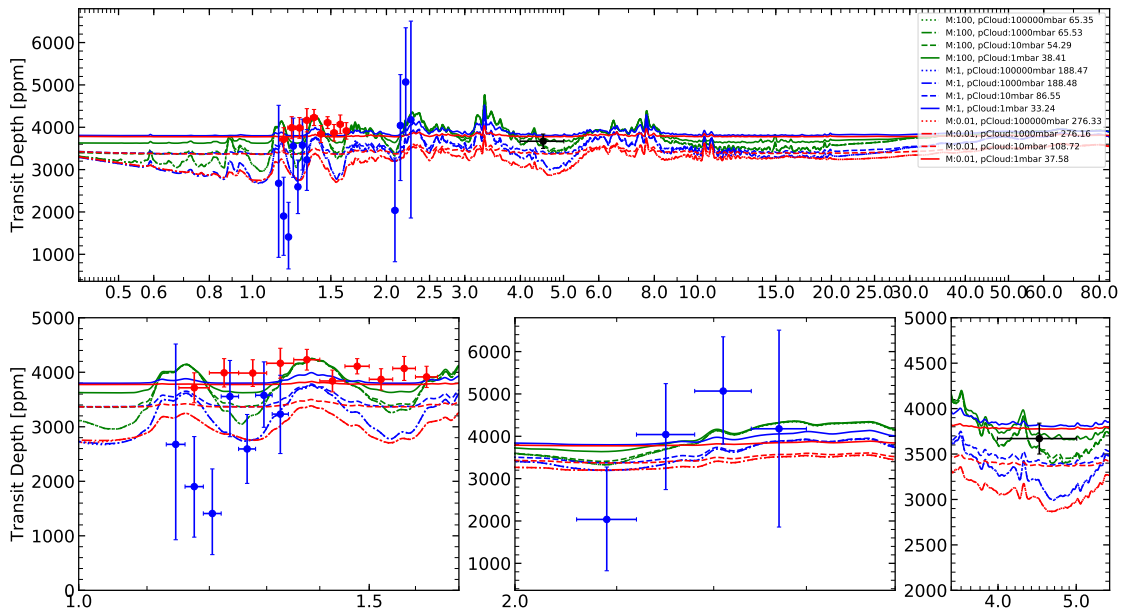


Fig. 3.13. Full spectrum with transit points from this work (blue), and previous *HST* (red) and *Spitzer* (black) studies [6, 7]. Vertical bars indicate the 1σ transit depth uncertainties while the horizontal bars indicate the wavelength ranges of the measurements. Bottom: The left, middle, and right plots zoom in on the J, K_s , and *Spitzer* bandpasses respectively.

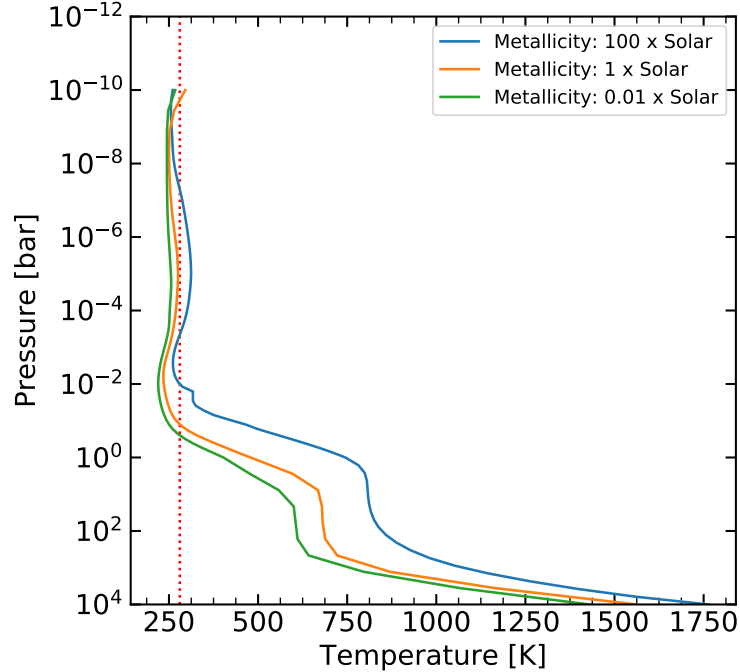


Fig. 3.14. Self-consistent temperature-pressure profiles for different atmospheric scenarios on TRAPPIST-1d. The dashed red line illustrates the calculated equilibrium temperature for TRAPPIST-1d while the solid curves depict the temperature-pressure profiles for various atmospheric cases at radiative-convective equilibrium. Note that the pressure axis is inverted as is commonly done for atmospheric temperature profiles.

TRAPPIST-1d, as depicted in Figure 3.15. Shaded regions represent normalized probability densities where darker shading indicate higher probability regions. We find cloud top pressures above 10000 mbar excluded from the 3σ bounds from metallicities of 0.01 to $300\times$ the solar value, indicating that clouds become optically thick below 10000 mbar. Low cloud top pressures and high metallicities have the same effect on observed and modelled spectra: both attenuate the observed or modelled transmission spectra. Thus, at high metallicities the effect of varying cloud top pressures becomes indistinguishable, as can be seen in Figure 3.15 at metallicities over $300\times$ solar. Similar to our results from self-consistent modelling, our joint retrieval also rules out cloud free atmospheres (with cloud top pressures lower than 10000 mbar) to 3σ at metallicities lower than $300\times$ the solar value.

6. Discussion and Conclusions

In this work we present the first spectroscopic results with the WIRC+Spec mode on Palomar Observatory’s WIRC instrument. Using a custom built data reduction pipeline, we apply a novel, KLT based, PSF fitting approach to extracting photometric and spectroscopic light curves from our observations of TRAPPIST-1d with the WIRC+Spec mode. We show the photometric results of our PSF fitting technique are consistent with the results of using

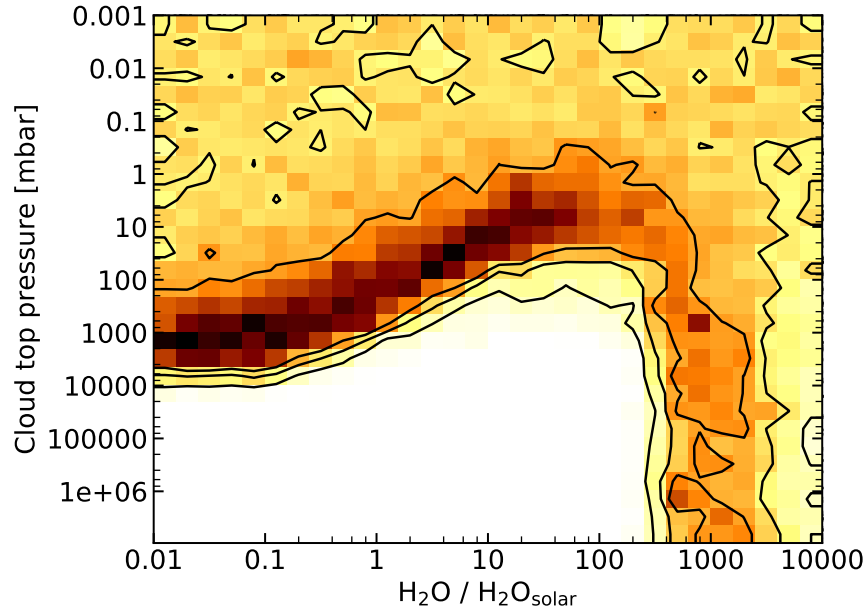


Fig. 3.15. Constraints on metallicity versus cloud top pressure. The shading indicates the normalized probability densities derived with SCARLET’s Bayesian atmospheric retrieval framework while the black contours show the 68%, 95%, and 99.7% Bayesian credible regions. Metallicities are described in reference to multiples of solar metallicity.

aperture photometry. This technique also provides a robust approach to extracting spectroscopy while taking into account the changes in PSF size due to atmospheric effects; this is not feasible with aperture spectrophotometry as discussed in Section 3.2. Using a newly implemented self-consistent radiative-convective temperature structure calculation, we model and compare various atmospheric scenarios with observations to date, confirming that a cloud-free, hydrogen dominated atmosphere is highly unlikely on TRAPPIST-1d. Although we were unable to characterize the atmosphere of TRAPPIST-1d better than previous works, we demonstrate WIRC+Spec’s capabilities as a tool for transit spectroscopy, even for a faint, challenging target such as TRAPPIST-1. In the future, we plan to apply our techniques to characterize the atmospheres of hot Jupiters and obtain more observations of TRAPPIST-1 to improve the signal of our current results. The *James Web Space Telescope (JWST)* and other upcoming next generation observatories will bring further clarity into the nature of the TRAPPIST-1 system.

References

- [1] Bjoern Benneke and Sara Seager. ATMOSPHERIC RETRIEVAL FOR SUPER-EARTHS: UNIQUELY CONSTRAINING THE ATMOSPHERIC COMPOSITION WITH TRANSMISSION SPECTROSCOPY. *The Astrophysical Journal*, 753(2):100, July 2012.
- [2] Björn Benneke. Strict Upper Limits on the Carbon-to-Oxygen Ratios of Eight Hot Jupiters from Self-Consistent Atmospheric Retrieval. *arXiv:1504.07655 [astro-ph]*, April 2015. arXiv: 1504.07655.

- [3] Björn Benneke, Heather A. Knutson, Joshua Lothringer, Ian J.M. Crossfield, Julianne I. Moses, Caroline Morley, Laura Kreidberg, Benjamin J. Fulton, Diana Dragomir, Andrew W. Howard, Ian Wong, Jean-Michel Désert, Peter R. McCullough, Eliza M.-R. Kempton, Jonathan Fortney, Ronald Gilliland, Drake Deming, and Joshua Kammer. A sub-Neptune exoplanet with a low-metallicity methane-depleted atmosphere and Mie-scattering clouds. *Nat Astron*, 3(9):813–821, September 2019.
- [4] Björn Benneke and Sara Seager. HOW TO DISTINGUISH BETWEEN CLOUDY MINI-NEPTUNES AND WATER/VOLATILE-DOMINATED SUPER-EARTHS. *The Astrophysical Journal*, 778(2):153, November 2013.
- [5] Björn Benneke, Michael Werner, Erik Petigura, Heather Knutson, Courtney Dressing, Ian J. M. Crossfield, Joshua E. Schlieder, John Livingston, Charles Beichman, Jessie Christiansen, Jessica Krick, Varujan Gorjian, Andrew W. Howard, Evan Sinukoff, David R. Ciardi, and Rachel L. Akeson. *SPITZER* OBSERVATIONS CONFIRM AND RESCUE THE HABITABLE-ZONE SUPER-EARTH K2-18b FOR FUTURE CHARACTERIZATION. *ApJ*, 834(2):187, January 2017.
- [6] J. de Wit, H. R. Wakeford, N. Lewis, L. Delrez, M. Gillon, F. Selsis, J. Leconte, B.-O. Demory, E. Bolmont, V. Bourrier, A. J. Burgasser, S. Grimm, E. Jehin, S. M. Lederer, J. Owen, V. Stamenkovic, and A. H. M. J. Triaud. Atmospheric reconnaissance of the habitable-zone Earth-sized planets orbiting TRAPPIST-1. *Nature Astronomy*, 2(3):214–219, March 2018. arXiv: 1802.02250.
- [7] Laetitia Delrez, Michael Gillon, Amaury H. M. J. Triaud, Brice-Olivier Demory, Julien de Wit, James G. Ingalls, Eric Agol, Emeline Bolmont, Artem Burdanov, Adam J. Burgasser, Sean J. Carey, Emmanuel Jehin, Jeremy Leconte, Susan Lederer, Didier Queloz, Franck Selsis, and Valerie Van Grootel. Early 2017 observations of TRAPPIST-1 with $\text{\textit{Spitzer}}$. *Monthly Notices of the Royal Astronomical Society*, 475(3):3577–3597, April 2018. arXiv: 1801.02554.
- [8] Vera Dobos, Amy C. Barr, and László L. Kiss. Tidal Heating and the Habitability of the TRAPPIST-1 Exoplanets. *A&A*, 624:A2, April 2019. arXiv: 1902.03867.
- [9] Chuanfei Dong, Meng Jin, Manasvi Lingam, Vladimir S. Airapetian, Yingjuan Ma, and Bart van der Holst. Atmospheric escape from the TRAPPIST-1 planets and implications for habitability. *Proceedings of the National Academy of Sciences*, 115(2):260–265, January 2018. arXiv: 1705.05535.
- [10] Courtney D. Dressing and David Charbonneau. THE OCCURRENCE RATE OF SMALL PLANETS AROUND SMALL STARS. *The Astrophysical Journal*, 767(1):95, March 2013.
- [11] Courtney D. Dressing and David Charbonneau. THE OCCURRENCE OF POTENTIALLY HABITABLE PLANETS ORBITING M DWARFS ESTIMATED FROM THE FULL *KEPLER* DATASET AND AN EMPIRICAL MEASUREMENT OF THE DETECTION SENSITIVITY. *ApJ*, 807(1):45, June 2015.
- [12] Will M. Farr, Jonathan R. Gair, Ilya Mandel, and Curt Cutler. Counting and confusion: Bayesian rate estimation with multiple populations. *Phys. Rev. D*, 91(2):023005, January 2015.
- [13] P. Feautrier. Sur la resolution numerique de l’equation de transfert. *Comptes Rendus Academie des Sciences (serie non specifiée)*, 258, 1964.
- [14] Jonathan Gagné, Jacqueline K. Faherty, Kelle L. Cruz, David Lafrenière, René Doyon, Lison Malo, Adam J. Burgasser, Marie-Eve Naud, Étienne Artigau, Sandie Bouchard, John E. Gizis, and Loïc Albert. BANYAN. VII. A NEW POPULATION OF YOUNG SUBSTELLAR CANDIDATE MEMBERS OF NEARBY MOVING GROUPS FROM THE BASS SURVEY. *The Astrophysical Journal Supplement Series*, 219(2):33, August 2015.

- [15] Siddharth Gandhi and Nikku Madhusudhan. GENESIS: New Self-Consistent Models of Exoplanetary Spectra. *Monthly Notices of the Royal Astronomical Society*, 472(2):2334–2355, December 2017. arXiv: 1706.02302.
- [16] Jan J. Gerbrands. On the relationships between SVD, KLT and PCA. *Pattern Recognition*, 14(1):375–381, January 1981.
- [17] Michael Gillon, Amaury H. M. J. Triaud, Brice-Olivier Demory, Emmanuel Jehin, Eric Agol, Katherine M. Deck, Susan M. Lederer, Julien de Wit, Artem Burdanov, James G. Ingalls, Emeline Bolmont, Jeremy Leconte, Sean N. Raymond, Franck Selsis, Martin Turbet, Khalid Barkaoui, Adam Burgasser, Matthew R. Burleigh, Sean J. Carey, Aleksander Chaushev, Chris M. Copperwheat, Laetitia Delrez, Catarina S. Fernandes, Daniel L. Holdsworth, Enrico J. Kotze, Valerie Van Grootel, Yaseen Almaleky, Zouhair Benkhaldoun, Pierre Magain, and Didier Queloz. Seven temperate terrestrial planets around the nearby ultracool dwarf star TRAPPIST-1. *Nature*, 542(7642):456–460, February 2017. arXiv: 1703.01424.
- [18] Michaël Gillon, Emmanuël Jehin, Susan M. Lederer, Laetitia Delrez, Julien de Wit, Artem Burdanov, Valérie Van Grootel, Adam Burgasser, Amaury H. M. J. Triaud, Cyrielle Opitom, Brice-Olivier Demory, Devendra K. Sahu, Daniella Bardalez Gagliuffi, Pierre Magain, and Didier Queloz. Temperate Earth-sized planets transiting a nearby ultracool dwarf star. *Nature*, 533(7602):221–224, May 2016.
- [19] Simon L. Grimm, Brice-Olivier Demory, Michaël Gillon, Caroline Dorn, Eric Agol, Artem Burdanov, Laetitia Delrez, Marko Sestovic, Amaury H. M. J. Triaud, Martin Turbet, Émeline Bolmont, Anthony Caldas, Julien de Wit, Emmanuël Jehin, Jérémy Leconte, Sean N. Raymond, Valérie Van Grootel, Adam J. Burgasser, Sean Carey, Daniel Fabrycky, Kevin Heng, David M. Hernandez, James G. Ingalls, Susan Lederer, Franck Selsis, and Didier Queloz. The nature of the TRAPPIST-1 exoplanets. *Astronomy & Astrophysics*, February 2018. arXiv: 1802.01377.
- [20] Ivan Hubeny and Dimitri Mihalas. *Theory of Stellar Atmospheres: An Introduction to Astrophysical Non-equilibrium Quantitative Spectroscopic Analysis*. Princeton University Press, October 2014. Google-Books-ID: TmuYDwAAQBAJ.
- [21] Ravi Kumar Kopparapu. A REVISED ESTIMATE OF THE OCCURRENCE RATE OF TERRESTRIAL PLANETS IN THE HABITABLE ZONES AROUND *KEPLER* M-DWARFS. *ApJ*, 767(1):L8, March 2013.
- [22] Laura Kreidberg. batman : BASic Transit Model cAlculatioN in Python. *Publications of the Astronomical Society of the Pacific*, 127(957):1161–1165, November 2015.
- [23] Philip Nutzman and David Charbonneau. Design Considerations for a Ground-Based Transit Search for Habitable Planets Orbiting M Dwarfs. *PUBL ASTRON SOC PAC*, 120(865):317–327, March 2008.
- [24] G.B. Rybicki. A modified Feautrier method. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 11(6):589–595, June 1971.
- [25] S. Seager and D. D. Sasselov. Extrasolar Giant Planets under Strong Stellar Irradiation. *The Astrophysical Journal*, 502(2):L157–L161, August 1998.
- [26] Ari Silburt, Eric Gaidos, and Yanqin Wu. A STATISTICAL RECONSTRUCTION OF THE PLANET POPULATION AROUND *KEPLER* SOLAR-TYPE STARS. *ApJ*, 799(2):180, January 2015.
- [27] Rémi Soummer, Laurent Pueyo, and James Larkin. DETECTION AND CHARACTERIZATION OF EXOPLANETS AND DISKS USING PROJECTIONS ON KARHUNEN-LOÈVE EIGENIMAGES. *The Astrophysical Journal*, 755(2):L28, August 2012.
- [28] Samaporn Tinyanont, Maxwell A. Millar-Blanchaer, Ricky Nilsson, Dimitri Mawet, Heather Knutson, Tiffany Kataria, Gautam Vasisht, Charles Henderson, Keith Matthews, Eugene Serabyn, Jennifer W.

- Milburn, David Hale, Roger Smith, Shreyas Vissapragada, Louis D. Santos, Jason Kekas, and Michael J. Escuti. WIRC+Pol: A Low-resolution Near-infrared Spectropolarimeter. *PASP*, 131(996):025001, February 2019.
- [29] Jason J. Wang, James R. Graham, Laurent Pueyo, Paul Kalas, Maxwell A. Millar-Blanchaer, Jean-Baptiste Ruffio, Robert J. De Rosa, S. Mark Ammons, Pauline Arriaga, Vanessa P. Bailey, Travis S. Barman, Joanna Bulger, Adam S. Burrows, Andrew Cardwell, Christine H. Chen, Jeffrey K. Chilcote, Tara Cotten, Michael P. Fitzgerald, Katherine B. Follette, René Doyon, Gaspard Duchêne, Alexandra Z. Greenbaum, Pascale Hibon, Li-Wei Hung, Patrick Ingraham, Quinn M. Konopacky, James E. Larkin, Bruce Macintosh, Jérôme Maire, Franck Marchis, Mark S. Marley, Christian Marois, Stanimir Metchev, Eric L. Nielsen, Rebecca Oppenheimer, David W. Palmer, Rahul Patel, Jenny Patience, Marshall D. Perrin, Lisa A. Poyneer, Abhijith Rajan, Julien Rameau, Fredrik T. Rantakyö, Dmitry Savransky, Anand Sivaramakrishnan, Inseok Song, Remi Soummer, Sandrine Thomas, Gautam Vasisht, David Vega, J. Kent Wallace, Kimberly Ward-Duong, Sloane J. Wiktorowicz, and Schuyler G. Wolff. THE ORBIT AND TRANSIT PROSPECTS FOR PICTORIS b CONSTRAINED WITH ONE MILLIARCSECOND ASTROMETRY. *The Astronomical Journal*, 152(4):97, October 2016.
- [30] John C. Wilson, Stephen S. Eikenberry, Charles P. Henderson, Thomas L. Hayward, Joseph C. Carson, Bruce Pirger, Donald J. Barry, Bernhard R. Brandl, James R. Houck, Gregory J. Fitzgerald, and T. M. Stolberg. A Wide-Field Infrared Camera for the Palomar 200-inch Telescope. page 451, Waikoloa, Hawai'i, United States, March 2003.
- [31] Jennifer G. Winters, Todd J. Henry, Wei-Chun Jao, John P. Subasavage, Joseph P. Chatelain, Ken Slatten, Adric R. Riedel, Michele L. Silverstein, and Matthew J. Payne. The Solar Neighborhood. XLV. The Stellar Multiplicity Rate of M Dwarfs Within 25 pc. *AJ*, 157(6):216, May 2019.

Summary and Conclusion

In this work, we demonstrated the capabilities of the WIRC+Spec mode on the WIRC instrument for exoplanet atmospheric characterization. Starting with observations from WIRC+Spec taken from August to October 2017, we developed a pipeline to reduce the raw exposures into time series photometric and spectroscopic light curves. We innovatively adapt a KLT-based algorithm to model the observed PSFs, allowing us to use a PSF fitting approach to extract our photometry and spectroscopy in an efficient and robust manner. Following the reduction and light curve analysis of our data, we compare the measured transmission spectrum against state-of-the-art atmospheric models. These models use the newly implemented self-consistent temperature structure calculation, which linearizes the RTE in order to quickly converge upon radiative-convective equilibrium. Our analysis of TRAPPIST-1d rules out cloud free atmospheres (with cloud top pressures lower than 10000 mbar) at metallicities lower than $300\times$ solar to 3σ confidence. In the future, we plan to apply our data reduction software to analyse hot Jupiter observations taken by WIRC+Spec in 2017. These targets are generally much less faint and therefore less challenging to observe. With a higher signal-to-noise ratio, we will be able to reduce our uncertainties and extract much cleaner light curves and resulting transmission spectra. These results will further establish WIRC+Spec as a candidate for future exoplanet observations and atmospheric characterization with transmission spectroscopy.

Appendix A

Self Consistent Temperature Structure Calculation

A.1. Overview

In the following sections, we describe the numerical methods from Hubeny and Mihalas as they have been implemented in SCARLET [1]. The source code for each step is also included for reference and a "cheat sheet" of the described variables can be found in Appendix B. For those using this appendix as a reference to implement a self-consistent temperature structure calculation, please be aware of the subtleties in nomenclature used (eg. κ vs. k or ω vs. w). Although the best effort has been made to clarify variable nomenclature, some variables may differ from one section of the code to the next (eg. \mathbf{A}_d in the Feautrier code and \mathbf{A}_k in the Rybicki method) in order to best emulate the original published algorithms.

A.2. Starting Procedure

We start by defining a pressure grid, which define the layers of the modelled atmosphere. The grid is defined at the boundary of each layer with the 0th layer being at the top of the atmosphere and the N^{th} layer being at the bottom, or surface of the planet.

A.2.1. Estimate Composition and Solve for Hydrostatic Equilibrium

Given a set of estimated parameters for the celestial object, we first estimate the chemical composition and solve for hydrostatic equilibrium using the `calcComposition` and `calcHydroEqui` functions respectively.

$$\frac{dP}{dz} = -\rho g \tag{A.2.1}$$

$$P = \frac{\rho k_B T}{\bar{m}} \tag{A.2.2}$$

A.2.2. Calculate Opacities

We then determine the extinction, absorption, and scattering coefficients with `calcOpacities`, taking into account any cloud opacities depending on the desired atmospheric

model.

$$\chi_\nu = \kappa_\nu + \sigma_\nu \quad (\text{A.2.3})$$

where χ_ν is the extinction coefficient, κ_ν is the absorption coefficient, and σ_ν is the scattering coefficient. We divide the extinction coefficient by the density at each layer to calculate the change in optical depth at each layer, $\Delta\tau_\nu$.

$$\omega_\nu = \frac{\chi_\nu}{\rho} \quad (\text{A.2.4})$$

In SCARLET, $\Delta\tau_\nu$ is defined *at the boundary of the top layer* ($\Delta\tau[0]$), and then *in the middle of each layer* for all other layers down to the exoplanet's surface ($\Delta\tau[1:]$). This allows us to calculate $\Delta\tau$ at half layers (eg. $\Delta\tau[0] = \Delta\tau_0$, $\Delta\tau[1] = \Delta\tau_{3/2}$).

$$\Delta\tau_\nu(d=0) = \frac{\omega_\nu(d=0)P(d=0)}{g(d=0)} \quad (\text{A.2.5})$$

$$\Delta\tau_{\nu,d>0} = \frac{1}{2} \frac{(\omega_{\nu,d} + \omega_{\nu,d-1})(P_d - P_{d-1})}{g_d} \quad (\text{A.2.6})$$

```

1 dtau = np.zeros([self.nLay,self.nWave])
2 dtau[0,:] = (mExtinctCoef[0,:]*self.p[0])/self.grav[0]
3 for i in range(1,self.nLay):
4     dtau[i,:] = 0.5*(mExtinctCoef[i,:] +
5         mExtinctCoef[i-1,:])*(self.p[i]-self.p[i-1])/self.grav[i]

```

Algorithm A.1. Defining dtau within the `solveRTE` function

A.2.3. Calculate Stellar Irradiation and Planck Function (B)

We assume the star is a black body and determine the stellar irradiance (H_{ext}) at the subsolar point for the given planet.

$$H_{ext} = F_\star \left(\frac{R_\star}{a} \right)^2 \quad (\text{A.2.7})$$

where $F_\star = B(T_{eff}, \nu)$ is the flux at the star's surface, R_\star is the radius of the star, a is the semi-major axis of the planet, and the Planck function is calculated as follows:

$$B(T, \nu) = \frac{2h\nu^3}{c^2} \frac{1}{e^{\frac{h\nu}{k_B T}} - 1} \quad (\text{A.2.8})$$

A.3. Solve the Radiative Transfer Equation with Feautrier's Method

A.3.1. Derivation

We start by splitting the radiative transfer equation into two sub-equations: one for incident radiation and another for emergent radiation.

$$\mu \frac{\partial I_\nu(\tau_\nu, \mu)}{\partial \tau_\nu} = I_\nu(\tau_\nu, \mu) - S_\nu(\tau_\nu) \quad (\text{A.3.1})$$

becomes:

$$+ \mu \frac{\partial I_\nu(+\mu)}{\partial \tau_\nu} = I_\nu(+\mu) - S_\nu \quad (\text{A.3.2})$$

$$- \mu \frac{\partial I_\nu(-\mu)}{\partial \tau_\nu} = I_\nu(-\mu) - S_\nu \quad (\text{A.3.3})$$

We assume the source function is symmetric so: $S_\nu(-\mu) = S_\nu(+\mu) = S_\nu$

We then define $I_\nu^\pm = I_\nu(\pm\mu)$ and:

$$u_{\mu\nu} \equiv \frac{1}{2}(I_\nu^+ + I_\nu^-) \quad (\text{A.3.4})$$

$$v_{\mu\nu} \equiv \frac{1}{2}(I_\nu^+ - I_\nu^-) \quad (\text{A.3.5})$$

Adding A.3.2 and A.3.3, we get:

$$\mu \frac{\partial}{\partial \tau_\nu} (I_\nu^+ - I_\nu^-) = (I_\nu^+ + I_\nu^-) - 2S_\nu \quad (\text{A.3.6})$$

$$\mu \frac{\partial v_{\mu\nu}}{\partial \tau_\nu} = u_{\mu\nu} - S_\nu \quad (\text{A.3.7})$$

Subtracting A.3.2 and A.3.3 gives us:

$$\mu \frac{\partial}{\partial \tau_\nu} (I_\nu^+ + I_\nu^-) = (I_\nu^+ - I_\nu^-) \quad (\text{A.3.8})$$

$$\mu \frac{\partial u_{\mu\nu}}{\partial \tau_\nu} = v_{\mu\nu} \quad (\text{A.3.9})$$

Taking the derivative of A.3.9 with respect to τ_ν ,

$$\mu \frac{\partial^2 u_{\mu\nu}}{\partial \tau_\nu^2} = \frac{\partial v_{\mu\nu}}{\partial \tau_\nu} \quad (\text{A.3.10})$$

We substitute A.3.10 into A.3.7 to obtain Feautrier's Equation:

$$\mu^2 \frac{\partial^2 u_{\mu\nu}}{\partial \tau_\nu^2} = u_{\mu\nu} - S_\nu \quad (\text{A.3.11})$$

The source function of the RTE is:

$$S_\nu = \frac{\kappa_\nu B_\nu + \sigma_\nu J_\nu}{\kappa_\nu + \sigma_\nu} \quad (\text{A.3.12})$$

where we define

$$\lambda_\nu = \frac{\kappa_\nu}{\kappa_\nu + \sigma_\nu} = \frac{\kappa_\nu}{\chi_\nu} \quad (\text{A.3.13})$$

We can then define J_ν in terms of $u_{\mu\nu}$ as

$$J_\nu = \frac{1}{2} \int_{-1}^1 I_\nu d\mu = \frac{1}{2} \left(\int_0^1 I_\nu^+ d\mu + \int_{-1}^0 I_\nu^- d\mu \right) \quad (\text{A.3.14})$$

$$J_\nu = \frac{1}{2} \int_0^1 (I_\nu^+ + I_\nu^-) d\mu \quad (\text{A.3.15})$$

$$J_\nu = \int_0^1 u_{\mu\nu} d\mu \quad (\text{A.3.16})$$

Similarly,

$$K_\nu = \int_0^1 u_{\mu\nu} \mu^2 d\mu \quad (\text{A.3.17})$$

A.3.2. Boundary Conditions

At $\tau_\nu = 0$, $I_\nu^- = 0$. Therefore, $u_{\mu\nu}(0) = v_{\mu\nu}(0)$ and A.3.9 becomes:

$$\mu \left(\frac{\partial u_{\mu\nu}}{\partial \tau_\nu} \right)_{\tau_\nu=0} = u_{\mu\nu}(0) - H_{ext} \quad (\text{A.3.18})$$

where H_{ext} is the incoming stellar radiation.

At $\tau_\nu = \tau_{max}$, the diffusion approximation gives us:

$$I_\nu^\pm(\tau_{max}) = B_\nu(\tau_{max}) \pm \mu \left(\frac{dB_\nu}{d\tau_\nu} \right)_{\tau_{max}} \quad (\text{A.3.19})$$

$$u_{\mu\nu}(\tau_{max}) = \frac{1}{2} [I_\nu^+(\tau_{max}) + I_\nu^-(\tau_{max})] = B_\nu(\tau_{max}) \quad (\text{A.3.20})$$

A.3.3. Discretization

In order to solve the RTE, we discretize and express the equations using finite differences. We first introduce the index d to represent each layer and j to represent each angle μ .

$$\tau_d, \quad d = 0, \dots, ND - 1$$

where ND is the total number of layers, and

$$\mu_j, \quad j = 0, \dots, M - 1$$

where M is the total number of angles probed.

The source function can then be expressed as:

$$S_\nu = \lambda_\nu B_\nu + (1 - \lambda_\nu) J_\nu$$

$$S_d = \lambda_d B_d + (1 - \lambda_d) \sum_{j=0}^{M-1} w_j u_{j,d} \quad (\text{A.3.21})$$

where w_j is a weighting function used to perform the integral numerically.

Then expressing a given function X in terms of finite differences of τ :

$$\frac{X_d - X_{d-1}}{\tau_d - \tau_{d-1}} \equiv \left(\frac{dX}{d\tau} \right)_{d-1/2} \quad (\text{A.3.22})$$

$$\frac{X_{d+1} - X_d}{\tau_{d+1} - \tau_d} \equiv \left(\frac{dX}{d\tau} \right)_{d+1/2} \quad (\text{A.3.23})$$

$$\left(\frac{d^2 X}{d\tau^2} \right)_d = \frac{\left(\frac{dX}{d\tau} \right)_{d+1/2} - \left(\frac{dX}{d\tau} \right)_{d-1/2}}{\Delta\tau_d} \quad (\text{A.3.24})$$

where

$$\begin{aligned} \Delta\tau_d &= \frac{1}{2}(\Delta\tau_{d+1/2} + \Delta\tau_{d-1/2}) \\ \Delta\tau_{d-1/2} &= \tau_d - \tau_{d-1} \\ \Delta\tau_{d+1/2} &= \tau_{d+1} - \tau_d \end{aligned} \quad (\text{A.3.25})$$

Note that equation A.3.25 and equation A.2.6 are equivalent.

Using a similar discretization for equation A.3.11, we can write:

$$\begin{aligned} \mu^2 \frac{\partial^2 u_{\mu\nu}}{\partial \tau_\nu^2} &= u_{\mu\nu} - S_\nu \\ \mu_j^2 \frac{\left[\left(\frac{du_j}{d\tau} \right)_{d+1/2} - \left(\frac{du_j}{d\tau} \right)_{d-1/2} \right]}{\Delta\tau_d} &= \frac{\mu_j^2}{\Delta\tau_d} \left[\frac{(u_{j,d+1} - u_{j,d})}{\Delta\tau_{d+1/2}} - \frac{(u_{j,d} - u_{j,d-1})}{\Delta\tau_{d-1/2}} \right] = u_{j,d} - S_d \quad (\text{A.3.26}) \\ \frac{\mu_j^2}{\Delta\tau_d \Delta\tau_{d-1/2}} u_{j,d-1} - \frac{\mu_j^2}{\Delta\tau_d} \left(\frac{1}{\Delta\tau_{d+1/2}} + \frac{1}{\Delta\tau_{d-1/2}} \right) u_{j,d} + \frac{\mu_j^2}{\Delta\tau_d \Delta\tau_{d+1/2}} u_{j,d+1} \\ &= u_{j,d} - \lambda_d B_d - (1 - \lambda_d) \sum_{k=0}^{M-1} w_k u_{k,d} \end{aligned} \quad (\text{A.3.27})$$

Here k replaces the index j in the summation term of the source function for reasons of clarity.

This equation forms a matrix of the form:

$$\begin{aligned} \begin{pmatrix} a_{00} & 0 & \cdots & 0 \\ 0 & a_{11} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{M-1M-1} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{M-1} \end{pmatrix}_{d-1} + \begin{pmatrix} b_{00} & b_{01} & \cdots & b_{0M} \\ b_{10} & b_{11} & \cdots & b_{1M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{M0} & b_{M1} & \cdots & b_{M-1M-1} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{M-1} \end{pmatrix}_d \\ + \begin{pmatrix} c_{00} & 0 & \cdots & 0 \\ 0 & c_{11} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{M-1M-1} \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{M-1} \end{pmatrix}_{d+1} = \begin{pmatrix} l_0 \\ l_1 \\ \vdots \\ l_{M-1} \end{pmatrix}_d \end{aligned}$$

Which can also be written as:

$$-\mathbf{A}_d \mathbf{u}_{d-1} + \mathbf{B}_d \mathbf{u}_d - \mathbf{C}_d \mathbf{u}_{d+1} = \mathbf{L}_d \quad (\text{A.3.28})$$

A.3.4. Numerical Methods

A.3.4.1. Middle Layers

With equation A.3.27 and A.3.28, we find for all layers excluding the top and bottom of the atmosphere:

$$\begin{aligned} a_{jj} &= -\frac{\mu_j^2}{\Delta\tau_d \Delta\tau_{d-1/2}} \\ a_{jk} &= 0 \\ b_{jj} &= -\frac{\mu_j^2}{\Delta\tau_d} \left(\frac{1}{\Delta\tau_{d+1/2}} + \frac{1}{\Delta\tau_{d-1/2}} \right) - 1 + (1 - \lambda_d)w_j \\ b_{jk} &= (1 - \lambda_d)w_k \\ c_{jj} &= -\frac{\mu_j^2}{\Delta\tau_d \Delta\tau_{d+1/2}} \\ c_{jk} &= 0 \\ l_j &= -\lambda_d B_d \end{aligned} \quad (\text{A.3.29})$$

```

1 # Middle -----
2 for d in range(1,l-1):
3     for j in range(imu):
4         A[j,:] = -mu[j]**2/(dtau[d,:]*1.0/2.0*(dtau[d,:]+dtau[d+1,:]))
5         C[j,:] = -mu[j]**2/(dtau[d+1,:]*1.0/2.0*(dtau[d,:]+dtau[d+1,:]))
6         L[j,:] = -lmbda[d,:]*plnk[d,:]
7         for k in range(imu):
8             B[j,k,:] = (1.0-lmbda[d,:])*weight[k]
9         B[j,j,:] = B[j,j,:] + A[j,:] + C[j,:] - 1

```

Algorithm A.2. Solving the matrices of the middle layers within the `feautrierRTE` function

A.3.4.2. Top of the atmosphere

At the top of the atmosphere, equation A.3.18

$$\mu \left(\frac{\partial u_{\mu\nu}}{\partial \tau_\nu} \right)_{\tau_\nu=0} = u_{\mu\nu}(0) - H_{ext}$$

becomes:

$$\mu_j \left(\frac{u_{j1} - u_{j0}}{\Delta\tau_{3/2}} \right) = u_{j0} - H_{ext} \quad (\text{A.3.30})$$

We perform a Taylor expansion to obtain:

$$u_{j1} = u_{j0} - H_{ext} + \Delta\tau_{3/2} \left(\frac{du}{d\tau} \right)_0 + \frac{1}{2} \Delta\tau_{3/2}^2 \left(\frac{d^2u}{d\tau^2} \right)_0 \quad (\text{A.3.31})$$

where from the Feautrier's Equation (A.3.11) and the upper boundary condition:

$$\left(\frac{d^2u}{d\tau^2} \right)_0 = \frac{1}{\mu_j^2} (u_{j0} - S_0)$$

$$\left(\frac{du}{d\tau} \right)_0 = \frac{u_{j0}}{\mu_j}$$

and therefore,

$$(u_{j1} - u_{j0}) = \Delta\tau_{3/2} \frac{u_{j0}}{\mu_j} + \frac{1}{2} \frac{\Delta\tau_{3/2}^2}{\mu_j^2} (u_{j0} - S_0) - H_{ext} \quad (\text{A.3.32})$$

$$\mu_j \frac{(u_{j1} - u_{j0})}{\Delta\tau_{3/2}} = u_{j0} + \frac{1}{2} \frac{\Delta\tau_{3/2}}{\mu_j} (u_{j0} - S_0) - H_{ext} \quad (\text{A.3.33})$$

We rearrange the above to write a matrix equation of the form:

$$\mathbf{B}_0 \mathbf{u}_0 - \mathbf{C}_0 \mathbf{u}_1 = \mathbf{L}_0 \quad (\text{A.3.34})$$

where

$$b_{jj} = -\frac{\mu_j}{\Delta\tau_{3/2}} - 1 - \frac{1}{2} \frac{\Delta\tau_{3/2}}{\mu_j} (1 - (1 - \lambda_0)w_j) \quad (\text{A.3.35})$$

$$b_{jk} = \frac{1}{2} \frac{\Delta\tau_{3/2}}{\mu_j} (1 - \lambda_0)w_k \quad (\text{A.3.36})$$

$$c_{jj} = -\frac{\mu_j}{\Delta\tau_{3/2}} \quad (\text{A.3.37})$$

$$l_j = -\frac{1}{2} \frac{\Delta\tau_{3/2}}{\mu_j} \lambda_0 B_0 - H_{ext} \quad (\text{A.3.38})$$

```

1 # TOA -----
2 for j in range(imu):
3     C[j,:] = -mu[j]/dtau[1,:]
4     L[j,:] = -1.0/2.0*dtau[1,:]/mu[j]*lmbda[0,:]*plnk[0,:] - Hext
5
6     for k in range(imu):
7         B[j,k,:] = 1.0/2.0*dtau[1,:]/mu[j]*(1.0-lmbda[0,:])*weight[k]
8
9     B[j,j,:] = B[j,j,:] + C[j,:] - 1.0 - 1.0/2.0*dtau[1,:]/mu[j]
```

Algorithm A.3. Solving the matrices of the top layers within the `feautrierRTE` function

A.3.4.3. *Bottom of the atmosphere*

At the bottom of the atmosphere, $u_{\mu\nu}(\tau_{max}) = B_\nu(\tau_{max})$ and $\mathbf{A}_{ND-1} = \mathbf{C}_{ND-1} = 0$ so:

$$\mathbf{B}_{ND-1}\mathbf{u}_{ND-1} = \mathbf{L}_{ND-1} \quad (\text{A.3.39})$$

where

$$b_{jj} = 1 \quad (\text{A.3.40})$$

$$b_{jk} = 0 \quad (\text{A.3.41})$$

$$l_j = B_{ND-1} \quad (\text{A.3.42})$$

where B is the Planck function.

```

1 # Bottom -----
2 for j in range(imu):
3     U[j,1-1,:] = plnk[1-1,:]

```

Algorithm A.4. Solving the matrices of the bottom layers within the `feautrierRTE` function

The following grand matrix can then be constructed:

$$\begin{pmatrix} \mathbf{B}_0 & -\mathbf{C}_0 & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ -\mathbf{A}_1 & \mathbf{B}_1 & -\mathbf{C}_1 & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}_2 & \mathbf{B}_2 & -\mathbf{C}_2 & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \ddots & \ddots & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & \vdots & \vdots & \cdots & -\mathbf{A}_{ND-2} & \mathbf{B}_{ND-2} & -\mathbf{C}_{ND-2} \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} & -\mathbf{A}_{ND-1} & \mathbf{B}_{ND-1} \end{pmatrix} \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \vdots \\ \mathbf{u}_{ND-2} \\ \mathbf{u}_{ND-1} \end{pmatrix} = \begin{pmatrix} \mathbf{L}_0 \\ \mathbf{L}_1 \\ \mathbf{L}_2 \\ \vdots \\ \vdots \\ \mathbf{L}_{ND-2} \\ \mathbf{L}_{ND-1} \end{pmatrix} \quad (\text{A.3.43})$$

A.3.4.4. *Solution of the Matrix*

We solve this system of equations using a Gauss-Jordan elimination, consisting of a forward elimination from the top layer down, followed by a back substitution at the bottom layer back up to the top.

Starting from the top layer:

$$\mathbf{B}_0 \mathbf{u}_0 - \mathbf{C}_0 \mathbf{u}_1 = \mathbf{L}_0 \quad (\text{A.3.44})$$

$$\mathbf{u}_0 = \underbrace{\mathbf{B}_0^{-1} \mathbf{C}_0}_{\mathbf{D}_0} \mathbf{u}_1 + \underbrace{\mathbf{B}_0^{-1} \mathbf{L}_0}_{\mathbf{R}_0} \quad (\text{A.3.45})$$

$$\mathbf{u}_0 = \mathbf{D}_0 \mathbf{u}_1 + \mathbf{R}_0 \quad (\text{A.3.46})$$

and then for the middle layers:

$$-\mathbf{A}_1 \mathbf{u}_0 + \mathbf{B}_1 \mathbf{u}_1 - \mathbf{C}_1 \mathbf{u}_2 = \mathbf{L}_1 \quad (\text{A.3.47})$$

where we can substitute A.3.46 for \mathbf{u}_0 to obtain:

$$-\mathbf{A}_1 (\mathbf{D}_0 \mathbf{u}_1 + \mathbf{R}_0) + \mathbf{B}_1 \mathbf{u}_1 - \mathbf{C}_1 \mathbf{u}_2 = \mathbf{L}_1 \quad (\text{A.3.48})$$

$$(\mathbf{B}_1 - \mathbf{A}_1 \mathbf{D}_0) \mathbf{u}_1 = \mathbf{C}_1 \mathbf{u}_2 + \mathbf{A}_1 \mathbf{R}_0 + \mathbf{L}_1 \quad (\text{A.3.49})$$

$$\mathbf{u}_1 = \underbrace{(\mathbf{B}_1 - \mathbf{A}_1 \mathbf{D}_0)^{-1} \mathbf{C}_1}_{\mathbf{D}_1} \mathbf{u}_2 + \underbrace{(\mathbf{B}_1 - \mathbf{A}_1 \mathbf{D}_0)^{-1} (\mathbf{A}_1 \mathbf{R}_0 + \mathbf{L}_1)}_{\mathbf{R}_1} \quad (\text{A.3.50})$$

Then in general up to $d = ND - 2$:

$$\mathbf{u}_d = \mathbf{D}_d \mathbf{u}_{d+1} + \mathbf{R}_d \quad (\text{A.3.51})$$

$$\mathbf{D}_d = (\mathbf{B}_d - \mathbf{A}_d \mathbf{D}_{d-1})^{-1} \mathbf{C}_d \quad (\text{A.3.52})$$

$$\mathbf{R}_d = (\mathbf{B}_d - \mathbf{A}_d \mathbf{D}_{d-1})^{-1} (\mathbf{A}_d \mathbf{R}_{d-1} + \mathbf{L}_d) \quad (\text{A.3.53})$$

We then calculate all \mathbf{D}_d and \mathbf{R}_d of the system. At the bottom of the atmosphere, $\mathbf{A}_{ND-1} = \mathbf{C}_{ND-1} = 0$ and we can then solve for \mathbf{u}_{ND-1} :

$$\mathbf{u}_{ND-1} = \mathbf{B}_{ND-1}^{-1} \mathbf{L}_{ND-1} \quad (\text{A.3.54})$$

We can then use

$$\mathbf{u}_d = \mathbf{D}_d \mathbf{u}_{d+1} + \mathbf{R}_d \quad (\text{A.3.55})$$

to find the rest of the \mathbf{u}_d , finally allowing us to calculate J_ν and K_ν

$$J_d = \sum_{j=0}^{M-1} w_j u_{jd} \quad (\text{A.3.56})$$

$$K_d = \sum_{j=0}^{M-1} w_j \mu_j^2 u_{jd} \quad (\text{A.3.57})$$

and to find H :

$$H_0 = \sum_{j=0}^{M-1} w_j \mu_j u_{j0} \quad (\text{A.3.58})$$

$$H_d = \frac{K_d - K_{d-1}}{\Delta \tau_d}$$

We then define f_ν and g_ν as:

$$f_\nu \equiv \frac{K_\nu}{J_\nu} \quad (\text{A.3.59})$$

$$g_\nu \equiv \frac{H_\nu(\tau = 0)}{J_\nu(\tau = 0)} \quad (\text{A.3.60})$$

We show below the complete **feautrierRTE** function including the calculations of the matrices at each layer and the Gauss-Jordan elimination as implemented in SCARLET. Please refer to Appendix A for a complete list of variables for the Feautrier method (as described in this section) and their equivalents in SCARLET.

```

1  def feautrierRTE(self,lmbda,dtau,B,Hext = 0, mu =
    np.array([0.8872983346,0.5,0.1127016654]), weight =
    np.array([0.27777777,0.4444444,0.27777777])):
2
3      [imu,l,w] = [mu.size,self.nLay,self.nWave]
4
5      plnk = B
6      J = np.zeros([l,w])
7      K = np.zeros([l,w])
8      H = np.zeros([l,w])
9
10     fn = np.zeros([l,w])
11     Gfn = np.zeros([w])
12
13     # Tier 1: mu
14     A = np.zeros([imu,w])
15     C = np.zeros([imu,w])
16     L = np.zeros([imu,w])
17     X = np.zeros([imu,w])
18
19     # Tier 2: mu*mu
20     B = np.zeros([imu,imu,w])
21     N = np.zeros([imu,imu,w])
22
23     # Tier 3: layered
24     U = np.zeros([imu,l,w])
25     R = np.zeros([imu,l,w])
26     D = np.zeros([imu,imu,l,w])

```

```

27
28 # TOA -----
29 for j in range(imu):
30     C[j,:] = -mu[j]/dtau[1,:]
31     L[j,:] = -1.0/2.0*dtau[1,;]/mu[j]*lmbda[0,]*plnk[0,] - Hext
32
33     for k in range(imu):
34         B[j,k,:] = 1.0/2.0*dtau[1,;]/mu[j]*(1.0-lmbda[0,])*weight[k]
35
36     B[j,j,:] = B[j,j,:] + C[j,:] - 1.0 - 1.0/2.0*dtau[1,;]/mu[j]
37
38 # Gauss-Jordan forward elimination
39 #Inverse B
40 Bflip = B
41 for i in range(imu-1,0,-1):
42     Bflip = np.swapaxes(Bflip,i-1,i)
43 B = np.linalg.inv(Bflip)
44 for i in range(0,imu-1):
45     B = np.swapaxes(B,i,i+1)
46
47 for j in range(imu):
48     for k in range(imu):
49         D[j,k,0,:] = B[j,k,:]*C[k,:]
50
51 for j in range(imu):
52     R[j,0,:] = 0
53     for k in range(imu):
54         R[j,0,:] = R[j,0,:] + B[j,k,:]*L[k,:]
55
56 # Middle -----
57 for d in range(1,l-1):
58     for j in range(imu):
59         A[j,:] = -mu[j]**2/(dtau[d,]*1.0/2.0*(dtau[d,]+dtau[d+1,]))
60         C[j,:] = -mu[j]**2/(dtau[d+1,]*1.0/2.0*(dtau[d,]+dtau[d+1,]))
61         L[j,:] = -lmbda[d,]*plnk[d,]
62         for k in range(imu):
63             B[j,k,:] = (1.0-lmbda[d,])*weight[k]
64         B[j,j,:] = B[j,j,:] + A[j,:] + C[j,:] - 1
65
66     for j in range(imu):

```

```

67         for k in range(imu):
68             N[j,k,:] = B[j,k,:] - A[j,:]*D[j,k,d-1,:]
69
70     for j in range(imu):
71         X[j,:] = A[j,:]*R[j,d-1,:]+L[j,:]
72
73     # Gauss-Jordan forward elimination
74     # Inverse N
75     Nflip = N
76     for i in range(imu-1,0,-1):
77         Nflip = np.swapaxes(Nflip,i-1,i)
78     N = np.linalg.inv(Nflip)
79     for i in range(0,imu-1):
80         N = np.swapaxes(N,i,i+1)
81
82     for j in range(imu):
83         for k in range(imu):
84             D[j,k,d,:] = N[j,k,:]*C[k,:]
85
86     for j in range(imu):
87         for k in range(imu):
88             R[j,d,:] = R[j,d,:] + N[j,k,:]*X[k,:]
89
90     # Bottom -----
91     for j in range(imu):
92         U[j,l-1,:] = plnk[l-1,:]
93
94     # Reverse Substitution -----
95     for d in range(l-2,-1,-1):
96         for j in range(imu):
97             for k in range(imu):
98                 U[j,d,:] = U[j,d,:] + D[j,k,d,:]*U[k,d+1,:]
99                 U[j,d,:] = U[j,d,:] + R[j,d,:]
100
101
102     # Calculate J,K,H
103     for d in range(l):
104         for j in range(imu):
105             J[d,:] = J[d,:]+weight[j]*U[j,d,:]
106             K[d,:] = K[d,:]+weight[j]*mu[j]**2*U[j,d,:]

```



```

107         fn[d,:] = K[d,:]/J[d,:]
108
109     for d in range(1,l):
110         H[d,:] = (K[d,:] - K[d-1,:])/dtau[d,:]
111
112     for j in range(imu):
113         H[0,:] = H[0,:] + weight[j]*mu[j]*U[j,0,:]
114
115     Gfn = H[0,:]/J[0,:]
116     return J,K,H,fn,Gfn

```

Algorithm A.5. The complete `feautrierRTE` function to solve the radiative transfer equation

In this section, we have solved the radiative transfer equation using the Feautrier method.

A.4. Determining the Temperature Profile

We follow the Rybicki method outlined in Hubeny & Mihalas (2014) to converge upon the correct temperature-pressure profile. The method follows a Newton-Raphson procedure, linearizing the solution to the RTE and radiative-convective equilibrium in order to solve for a temperature difference that iteratively brings us closer towards the converged solution. The linearization couples bordering layers and also takes into account the wavelength dependent opacities for the entire wavelength band at once, reducing both computational power and required time.

The Rybicki method can be divided into three segments: solving for radiative transfer, radiative equilibrium, and convective equilibrium. The derivation of Rybicki’s linearization is beyond the scope of this work and we instead focus on the implementation of Rybicki’s result.

A.4.1. Numerical Derivatives

We first calculate the necessary derivatives of κ , χ , ω , λ , κB and B at the current temperature and pressure numerically. For example, to calculate the derivative of a variable $X(T)$:

$$\frac{\partial X_d}{\partial T_d} = \frac{X(1.01T_d) - X(T_d)}{1.01T_d - T_d} \quad (\text{A.4.1})$$

Below, we show the `diffVar` function used to calculate needed numerical derivatives. For numerical stability we do not update the chemistry of the model until the temperature-pressure profile has converged once.

```

1  def diffVar(self,T,plnk,lmbda,mExtinctCoef,extinctCoef,absorbCoef,loop):
2      newT = T*1.01
3      diff = newT-T
4      diff = np.tile(diff[:,np.newaxis],(1,self.nWave))
5
6      if loop == 1: # Add stability by calculation chemistry/hydroequil only
7          after 1st convergence
8          self.qmol_lay =
9              self.calcComposition(self.modelSetting,self.params,newT)
10             self.z,self.dz,self.grav,self.ntot,self.nmol,self.MuAve,
11             self.scaleHeight,self.RpBase,self.r=
12             self.calcHydroEqui(self.modelSetting,self.params,newT)
13
14             extinctCoef2,absorbCoef2, scatCoef =
15                 self.calcOpacities(self.modelSetting,self.params,newT)
16             rho = np.tile((self.ntot*self.MuAve)[: ,np.newaxis],(1,self.nWave))
17
18             mExtinctCoef2 = extinctCoef2/rho
19
20             lmbda2 = absorbCoef2/extinctCoef2
21
22             B2 = rad.PlanckFct(np.tile(newT[:,np.newaxis],(1,self.nWave)),
23                 np.tile(self.f[np.newaxis,:],(len(newT),1)),'Hz','W/(m**2*Hz)','rad')
24
25             diffExt = (extinctCoef2 - extinctCoef)/diff
26             diffAbs = (absorbCoef2 - absorbCoef)/diff
27             diffMExt = (mExtinctCoef2 - mExtinctCoef)/diff
28             diffLmbda = (lmbda2 - lmbda)/diff
29             diffN = (absorbCoef2*B2 - absorbCoef*plnk)/diff
30             diffPlnk = (B2 - plnk)/diff
31
32             return diffExt,diffAbs,diffMExt,diffLmbda,diffN,diffPlnk

```

Algorithm A.6. Determining the numerical derivatives of κ , χ , ω , λ , κB and B in the `diffVar` function

A.4.2. Radiative Transfer

The linearized form of the RTE results in the following matrix equation:

$$\mathbf{U}_k \delta \mathbf{J}_k + \mathbf{V}_k \delta \mathbf{T} = \mathbf{E}_k \quad (\text{A.4.2})$$

where

$$k = 0, \dots, NF - 1$$

and NF is the total number of frequencies considered. \mathbf{U}_k and \mathbf{V}_k are tridiagonal matrices to account for the coupling between any given layer and the layers above and below it; for one frequency, the matrix is of the form:

$$\begin{pmatrix} \mathbf{U}_{B,d=0} & \mathbf{U}_{C,d=0} & \mathbf{0} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ -\mathbf{U}_{A,1} & \mathbf{U}_{B,1} & -\mathbf{U}_{C,1} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & -\mathbf{U}_{A,2} & \mathbf{U}_{B,2} & -\mathbf{U}_{C,2} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \ddots & \ddots & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & \vdots & \vdots & \dots & -\mathbf{U}_{A,ND-2} & \mathbf{U}_{B,ND-2} & -\mathbf{U}_{C,ND-2} \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{0} & -\mathbf{U}_{A,ND-1} & \mathbf{U}_{B,ND-1} \end{pmatrix} \quad (\text{A.4.3})$$

where

$$d = 0, \dots, ND - 1$$

and ND is the total number of layers. \mathbf{V}_k is similarly structured.

Below, we describe the equations to build the linearized matrices of the RTE from the components of the Feautrier method.

At the top of the atmosphere:

$$\mathbf{U}_{B,0} = \frac{f_0}{\Delta\tau_{3/2}} + g_\nu + \frac{\Delta\tau_{3/2}}{2} \lambda_0 \quad (\text{A.4.4})$$

$$\mathbf{U}_{C,0} = \frac{f_1}{\Delta\tau_{3/2}} \quad (\text{A.4.5})$$

$$\begin{aligned} \mathbf{V}_{B,0} = & \left[-\frac{f_0 J_0 - f_1 J_1}{\Delta\tau_{3/2}^2} + \frac{1}{2} \lambda_0 (J_0 - B_0) \right] \frac{\Delta\tau_{3/2}}{\omega_0 + \omega_1} \frac{\partial\omega_0}{\partial T_0} \\ & + \frac{\Delta\tau_{3/2}}{2} \left[\frac{\partial\lambda_0}{\partial T_0} J_0 - \lambda_0 B_0 \left(\frac{1}{\kappa_0 B_0} \frac{\partial(\kappa_0 B_0)}{\partial T_0} - \frac{1}{\chi_0} \frac{\partial\chi_0}{\partial T_0} \right) \right] \end{aligned} \quad (\text{A.4.6})$$

$$\mathbf{V}_{C,0} = - \left[\left(\frac{f_0 J_0 - f_1 J_1}{\Delta\tau_{3/2}^2} \right) \frac{\Delta\tau_{3/2}}{\omega_1 + \omega_2} \frac{\partial\omega_1}{\partial T_1} \right] \quad (\text{A.4.7})$$

$$\mathbf{E}_0 = -\frac{f_0 J_0 - f_1 J_1}{\Delta\tau_{3/2}} - g_\nu J_0 + H_{ext} - \frac{\Delta\tau_{3/2}}{2} (J_0 - B_0) \lambda_0 \quad (\text{A.4.8})$$

1 ##### Top of atmosphere #####

2 `U = np.zeros([self.nLay, self.nLay, self.nWave])`

3 `V = np.zeros([self.nLay, self.nLay, self.nWave])`

```

4 E = np.zeros([self.nLay,self.nWave])
5 # Top Layer
6 # at the top layer A = 0 and so we only have B and C
7 # B equation 18.64a
8 U[0,0,:] = fn[0,;]/dtau[1,:] + Gfn + dtau[1,:] * lambda[0,:] / 2.0 # B at top
9 # C equation 18.64b
10 U[0,1,:] = fn[1,;]/dtau[1,:] # C at top
11 # B equation 18.65a
12 V[0,0,:] = ((-1.0*(fn[0,]*J[0,] - fn[1,]*J[1,])/(dtau[1,]**2)) + 0
      .5*lambda[0,]*(J[0,] - plnk[0,]))*(dtau[1,]*domega_dT[0,]/(omega[0,] +
13      omega[1,])) + (0.5*dtau[1,])*(dlambda_dT[0,]*J[0,] -
      lambda[0,]*plnk[0,]*(deta_dT[0,]/(kappa[0,]*plnk[0,]) +
      dchi_dT[0,]/chi[0,]))
14 # C equation 18.65b
15 V[0,1,:] = -1.0*((fn[0,]*J[0,] -
      fn[1,]*J[1,])/(dtau[1,]**2))*(dtau[1,]*domega_dT[1,]/(omega[0,] +
      omega[1,]))
16 # L in Mihalas equation 18.66
17 E[0,:] = (-1.0*(fn[0,]*J[0,] - fn[1,]*J[1,])/dtau[1,]) - Gfn*J[0,] + Hext -
18      (0.5*dtau[1,]*lambda[0,]*(J[0,] - plnk[0,]))

```

Algorithm A.7. Solving the matrices of the top layers within the `correct` function

For the middle layers:

$$\mathbf{U}_{A,d} = - \left[\frac{f_{d-1}}{\Delta\tau_{d-1/2}\Delta\tau_d} \right] \quad (\text{A.4.9})$$

$$\mathbf{U}_{B,d} = \frac{f_d}{\Delta\tau_d} \left(\frac{1}{\Delta\tau_{d-1/2}} + \frac{1}{\Delta\tau_{d+1/2}} \right) + \lambda_d \quad (\text{A.4.10})$$

$$\mathbf{U}_{C,d} = - \left[\frac{f_{d+1}}{\Delta\tau_{d+1/2}\Delta\tau_d} \right] \quad (\text{A.4.11})$$

$$\mathbf{V}_{A,d} = - \left[a_d \left(\frac{\partial\omega_{d-1}}{\partial T_{d-1}} \right) \right] \quad (\text{A.4.12})$$

$$\mathbf{V}_{B,d} = - (a_d + c_d) \frac{\partial\omega_d}{\partial T_d} + \frac{\partial\lambda_d}{\partial T_d} J_d - \lambda_d B_d \left(\frac{1}{\kappa_d B_d} \frac{\partial(\kappa_d B_d)}{\partial T_d} - \frac{1}{\chi_d} \frac{\partial\chi_d}{\partial T_d} \right) \quad (\text{A.4.13})$$

$$\mathbf{V}_{C,d} = - \left[c_d \left(\frac{\partial\omega_{d+1}}{\partial T_{d+1}} \right) \right] \quad (\text{A.4.14})$$

where

$$\alpha_d = \frac{f_d J_d - f_{d-1} J_{d-1}}{\Delta\tau_{d-1/2} \Delta\tau_d} \quad (\text{A.4.15})$$

$$\gamma_d = \frac{f_d J_d - f_{d+1} J_{d+1}}{\Delta\tau_{d+1/2} \Delta\tau_d} \quad (\text{A.4.16})$$

$$\beta_d = \alpha_d + \gamma_d \quad (\text{A.4.17})$$

$$a_d = \frac{\alpha_d + \frac{1}{2}\beta_d \left(\frac{\Delta\tau_{d-1/2}}{\Delta\tau_d} \right)}{\omega_{d-1} + \omega_d} \quad (\text{A.4.18})$$

$$c_d = \frac{\gamma_d + \frac{1}{2}\beta_d \left(\frac{\Delta\tau_{d+1/2}}{\Delta\tau_d} \right)}{\omega_{d+1} + \omega_d} \quad (\text{A.4.19})$$

$$\mathbf{E}_d = -\beta_d - \lambda_d (J_d - B_d) \quad (\text{A.4.20})$$

```

1 ##### Middle of atmosphere #####
2 mid_lyrs = np.arange(self.nLay-2)+1 # all layers excepts the top and bottom ones
3 # A          equation 18.68a
4 U[mid_lyrs,mid_lyrs-1,:] =
      -1.0*(fn[mid_lyrs-1,:]/(dtau[mid_lyrs,:]*0.5*(dtau[mid_lyrs,:]+
5      dtau[mid_lyrs+1,:]))
6 # B          equation 18.68b
7 U[mid_lyrs,mid_lyrs,:] =
      (fn[mid_lyrs,:]/(0.5*(dtau[mid_lyrs,:]+dtau[mid_lyrs+1,:])))*
8      (1.0/dtau[mid_lyrs,:] + 1.0/dtau[mid_lyrs+1,:]) + lmbda[mid_lyrs,:]
9 # C          equation 18.68c
10 U[mid_lyrs,mid_lyrs+1,:] =
      -1.0*(fn[mid_lyrs+1,:]/(dtau[mid_lyrs+1,:]*0.5*(dtau[mid_lyrs,:]+
11      dtau[mid_lyrs+1,:]))
12 # now for V we need to compute equations 18.70a-e
13 alpha_di = (fn[mid_lyrs,:]*J[mid_lyrs,:] -
      fn[mid_lyrs-1,:]*J[mid_lyrs-1,:])/(dtau[mid_lyrs,:]*0.5*(dtau[mid_lyrs,:]+
14      dtau[mid_lyrs+1,:]))
15 gamma_di = (fn[mid_lyrs,:]*J[mid_lyrs,:] -
      fn[mid_lyrs+1,:]*J[mid_lyrs+1,:])/(dtau[mid_lyrs+1,:]*0.5*(dtau[mid_lyrs,:]+
16      dtau[mid_lyrs+1,:]))
17 beta_di = alpha_di + gamma_di
18 a_di = (alpha_di + 0.5*beta_di*dtau[mid_lyrs,:]/(0.5*(dtau[mid_lyrs,:]+
19      dtau[mid_lyrs+1,:])))/
20      (omega[mid_lyrs-1,:]+omega[mid_lyrs,:])

```

```

21 c_di = (gamma_di + 0.5*beta_di*dtau[mid_lyrs+1,:]/(0.5*(dtau[mid_lyrs,:]+
22   dtau[mid_lyrs+1,:])))/
23   (omega[mid_lyrs+1,:]+omega[mid_lyrs,:])
24 # A           equation 18.69a
25 V[mid_lyrs,mid_lyrs-1,:] = -1.0*(a_di*domega_dT[mid_lyrs-1,:])
26 # B           equation 18.69b
27 V[mid_lyrs,mid_lyrs,:] = -1.0*(a_di+c_di)*domega_dT[mid_lyrs,:] +
   dlambda_dT[mid_lyrs,:]*J[mid_lyrs,:] -
   lmbda[mid_lyrs,:]*plnk[mid_lyrs,:]*(deta_dT[mid_lyrs,:]/(kappa[mid_lyrs,:]*
28   plnk[mid_lyrs,:]) - dchi_dT[mid_lyrs,:]/chi[mid_lyrs,:])
29 # C           equation 18.69c
30 V[mid_lyrs,mid_lyrs+1,:] = -1.0*(c_di*domega_dT[mid_lyrs+1,:])
31 # L           equation 18.71
32 E[mid_lyrs,:] = -1.0*beta_di - lmbda[mid_lyrs,:]*(J[mid_lyrs,:] -
   plnk[mid_lyrs,:])

```

Algorithm A.8. Solving the matrices of the middle layers within the **correctT** function

At the bottom of the atmosphere:

$$\mathbf{U}_{A,ND-1} = - \left[\frac{f_{d-1}}{\Delta\tau_{d-1/2}} \right] \quad (\text{A.4.21})$$

$$\mathbf{U}_{B,ND-1} = \frac{f_d}{\Delta\tau_{d-1/2}} + \frac{1}{2} + \frac{\Delta\tau_{d-1/2}}{2} \lambda_d \quad (\text{A.4.22})$$

$$\mathbf{V}_{A,ND-1} = - \left[\left(\frac{f_d J_d - f_{d-1} J_{d-1}}{\Delta\tau_{d-1/2}^2} - b_d \right) \frac{\Delta\tau_{d-1/2}}{\omega_d + \omega_{d-1}} \frac{\partial \omega_{d-1}}{\partial T_{d-1}} - \frac{1}{3\Delta\tau_{d-1/2}} \left(\frac{\partial B}{\partial T} \right)_{d-1} \right] \quad (\text{A.4.23})$$

$$\begin{aligned} \mathbf{V}_{B,ND-1} = & \left[-\frac{f_d J_d - f_{d-1} J_{d-1}}{\Delta\tau_{d-1/2}^2} + b_d + \frac{1}{2} \lambda_d (J_d - B_d) \right] \frac{\Delta\tau_{d-1/2}}{\omega_d + \omega_{d-1}} \frac{\partial \omega_d}{\partial T_d} \\ & + \frac{\Delta\tau_{d-1/2}}{2} \left[\frac{\partial \lambda_d}{\partial T_d} J_d - \lambda_d B_d \left(\frac{1}{\kappa_d B_d} \frac{\partial (\kappa_d B_d)}{\partial T_d} - \frac{1}{\chi_d} \frac{\partial \chi_d}{\partial T_d} \right) \right] \\ & - \left(\frac{1}{2} + \frac{1}{3\Delta\tau_{d-1/2}} \right) \left(\frac{\partial B}{\partial T} \right)_d \end{aligned} \quad (\text{A.4.24})$$

where

$$b_d = \frac{1}{3} \frac{B_d - B_{d-1}}{\Delta\tau_{d-1/2}^2} \quad (\text{A.4.25})$$

$$\mathbf{E}_{ND-1} = -\frac{f_d J_d - f_{d-1} J_{d-1}}{\Delta\tau_{d-1/2}} - \frac{1}{2} (J_d - B_d) + \frac{1}{3} \frac{B_d - B_{d-1}}{\Delta\tau_{d-1/2}} - \frac{\Delta\tau_{d-1/2}}{2} \lambda_d (J_d - B_d) \quad (\text{A.4.26})$$

```

1 ##### Bottom of atmosphere
2 # A          equation 18.72b
3 U[-1,-2,:] = -1.0*fn[-2,:]/dtau[-1,:]
4 # B          equation 18.72a
5 U[-1,-1,:] = fn[-1,:]/dtau[-1,:] + 0.5 + 0.5*dtau[-1,:]*lmbda[-1,:]
6 # for V first calculate b_i (equation 18.73)
7 b_i = (1.0/3.0)*(plnk[-1,:]-plnk[-2,:])/(dtau[-1,:]**2)
8 # A          equation 18.72d
9 V[-1,-2,:] = -1.0*(((fn[-1,:]*J[-1,:] - fn[-2,:]*J[-2,:])/(dtau[-1,:]**2) -
    b_i)*dtau[-1,:]*domega_dT[-2,:]/(omega[-1,:]+omega[-2,:])) -
    dplnk_dT[-2,:]/(3.0*dtau[-1,:]))
10 # B         equation 18.72c
11 V[-1,-1,:] = ((-1.0*((fn[-1,:]*J[-1,:] - fn[-2,:]*J[-2,:])/(dtau[-1,:]**2))) +
12    b_i + 0.5*lmbda[-1,:]*(J[-1,:]-plnk[-1,:]))*(dtau[-1,:]*domega_dT[-1,:]/
13    (omega[-1,:]+omega[-2,:])) + 0.5*dtau[-1,:]*(dlambda_dT[-1,:]*J[-1,:] -
14    lmbda[-1,:]*plnk[-1,:]*(deta_dT[-1,:]/(kappa[-1,:]*plnk[-1,:]) -
15    dchi_dT[-1,:]/chi[-1,:])) - dplnk_dT[-1,:]*(0.5 + 1.0/(3.0*dtau[-1,:]))
16 # L         equation 18.74
17 E[-1,:] = -1.0*((fn[-1,:]*J[-1,:] - fn[-2,:]*J[-2,:])/dtau[-1,:]) - 0.5*(J[-1,:]
    - plnk[-1,:]) +
18    (plnk[-1,:] - plnk[-2,:])/(3.0*dtau[-1,:]) - 0
    .5*dtau[-1,:]*lmbda[-1,:]*(J[-1,:] - plnk[-1,:])

```

Algorithm A.9. Solving the matrices of the bottom layers within the `correct` function

A.4.3. Radiative Equilibrium

For a single layer, the equation for radiative equilibrium can be written as:

$$\int_0^{\infty} \kappa_{\nu} (J_{\nu} - B_{\nu}) d\nu = 0 \quad (\text{A.4.27})$$

$$\int_0^{\infty} \frac{d(f_{\nu} J_{\nu})}{d\tau_{\nu}} d\nu = \frac{\sigma_{SB}}{4\pi} T_{int}^4 \quad (\text{A.4.28})$$

where σ_{SB} is the Stefan-Boltzmann constant and T_{int} is the internal heat of the planet. For numerical stability, the layers of the atmosphere are split into two regimes in order to solve for radiative equilibrium. The integral form (equation A.4.27) is used in the top of the atmosphere ($P < 1 \times 10^5 Pa$) while the differential form (equation A.4.28) is used for the bottom of the atmosphere ($P \geq 1 \times 10^5 Pa$).

The linearized equations of radiative equilibrium leave us with the following matrix equation:

$$\sum_{k=0}^{NF-1} \mathbf{X}_k \delta \mathbf{J}_k + \mathbf{A} \delta \mathbf{T} = \mathbf{F} \quad (\text{A.4.29})$$

where, like \mathbf{U}_k and \mathbf{V}_k in the linearized RTE, \mathbf{X}_k and \mathbf{A} are again tridiagonal matrices of the form:

$$\begin{pmatrix} \mathbf{X}_{B,0} & -\mathbf{X}_{C,0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ -\mathbf{X}_{A,1} & \mathbf{X}_{B,1} & -\mathbf{X}_{C,1} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{X}_{A,2} & \mathbf{X}_{B,2} & -\mathbf{X}_{C,2} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \ddots & \ddots & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & \vdots & \vdots & \cdots & -\mathbf{X}_{A,ND-2} & \mathbf{X}_{B,ND-2} & -\mathbf{X}_{C,ND-2} \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} & -\mathbf{X}_{A,ND-1} & \mathbf{X}_{B,ND-1} \end{pmatrix} \quad (\text{A.4.30})$$

A.4.3.1. Integral Form

Linearizing equation A.4.27 for the upper atmosphere builds the following matrices for convective layers:

$$\mathbf{X}_{B,d} = w_k \kappa_d \quad (\text{A.4.31})$$

$$\mathbf{A}_{B,d} = \sum_{k=0}^{NF-1} w_k \left(\frac{\partial \kappa_d}{\partial T_d} J_d - \frac{\partial (\kappa_d B_d)}{\partial T_d} \right) \quad (\text{A.4.32})$$

$$\mathbf{F}_d = - \sum_{k=0}^{NF-1} w_k \kappa_d (J_d - B_d) \quad (\text{A.4.33})$$

Here the integration weights w_k are calculated within the **calcWeight** function of SCARLET, which determines the required integration weights based on the wavelength grid of the model.

```

1  def calcWeight(self):
2      weight = np.zeros([self.nWave])
3      for i in range(1,self.nWave-1):
4          weight[i] = 0.5*(self.f[i+1]-self.f[i-1])
5
6      weight[0] = 0.5*(self.f[1]-self.f[0])
7      weight[-1] = 0.5*(self.f[-1]-self.f[-2])
8
9      if self.f[0] > self.f[-1]:
10         return weight * -1.0          #FLIPPED FREQ GRID SO -VE
11     else:
```


Algorithm A.10. Calculating integration weights with the `calcWeight` function.

Solving for the integral matrices with SCARLET is show below:

```

1 # integral form at top half of atmosphere
2 # B          equation 18.78a
3 X[int_top,int_top,:] = weights*kappa[int_top,:]
4 # B          equation 18.78b - note that we think there should be a sum here
5 A[int_top,int_top] = np.sum((dkappa_dT[int_top,:]*J[int_top,:] -
        deta_dT[int_top,:])*weights,
6                             axis=1)
7 # L          equation 18.78c
8 F[int_top] = -1.0*np.sum(weights*kappa[int_top,:]*(J[int_top,:] -
        plnk[int_top,:]), axis=1)

```

Algorithm A.11. Solving the integral form matrices of radiative equilibrium withing the `correctT` function

A.4.3.2. Differential Form

Similarly, linearizing equation A.4.28 gives the following terms for convective layers:

$$\mathbf{X}_{A,d} = - \left[w_k \frac{f_{d-1}}{\Delta\tau_{d-1/2}} \right] \quad (\text{A.4.34})$$

$$\mathbf{X}_{B,d} = w_k \frac{f_d}{\Delta\tau_{d-1/2}} \quad (\text{A.4.35})$$

$$\mathbf{A}_{A,d} = - \left[\sum_{k=0}^{NF-1} w_k \frac{f_d J_d - f_{d-1} J_{d-1}}{\Delta\tau_{d-1/2}^2} \frac{\Delta\tau_{d-1/2}}{\omega_d + \omega_{d-1}} \frac{\partial \omega_{d-1}}{\partial T_{d-1}} \right] \quad (\text{A.4.36})$$

$$\mathbf{A}_{B,d} = - \sum_{k=0}^{NF-1} w_k \frac{f_d J_d - f_{d-1} J_{d-1}}{\Delta\tau_{d-1/2}^2} \frac{\Delta\tau_{d-1/2}}{\omega_d + \omega_{d-1}} \frac{\partial \omega_d}{\partial T_d} \quad (\text{A.4.37})$$

$$\mathbf{F}_d = \frac{\sigma_{SB}}{4\pi} T_{int}^4 - \sum_{k=0}^{NF-1} w_k \frac{f_d J_d - f_{d-1} J_{d-1}}{\Delta\tau_{d-1/2}} \quad (\text{A.4.38})$$

At the top of the atmosphere:

$$\mathbf{X}_{B,0} = w_k g_\nu \quad (\text{A.4.39})$$

$$\mathbf{F}_0 = \frac{\sigma_{SB}}{4\pi} T_{int}^4 - \sum_{k=0}^{NF-1} w_k (g_\nu J_0 - H_{ext}) \quad (\text{A.4.40})$$

These last two equations for the top of the atmosphere are generally not applicable when the integral and differential forms are combined since the differential form of linearized

radiative equilibrium is not used at the top of the atmosphere, however we include them for completeness.

```

1 # Differential form at bottom part of the atmosphere
2 # A          equation 18.79a
3 X[diff_bot,diff_bot-1,:] = -1.0*weights*fn[diff_bot-1,:]/dtau[diff_bot,:]
4 # B          equation 18.79b
5 X[diff_bot,diff_bot,:] = weights*fn[diff_bot,:]/dtau[diff_bot,:]
6 # A          equation 18.79c
7 A[diff_bot,diff_bot-1] = -1.0*np.sum(weights*((fn[diff_bot,:]*J[diff_bot,:] -
8         fn[diff_bot-1,:]*J[diff_bot-1,:])/(dtau[diff_bot,:]**2))*
9         (omega_dT[diff_bot-1,:]*dtau[diff_bot,:]/
10        (omega[diff_bot,:]+omega[diff_bot-1,:])),axis=1)
11 # B          equation 18.79d
12 A[diff_bot,diff_bot] = -1.0*np.sum(weights*((fn[diff_bot,:]*J[diff_bot,:] -
13        fn[diff_bot-1,:]*J[diff_bot-1,:])/(dtau[diff_bot,:]**2))*
14        (omega_dT[diff_bot,:]*dtau[diff_bot,:]/(omega[diff_bot,:]+
15        omega[diff_bot-1,:])),axis=1)
16 # L          equation 18.80
17 F[diff_bot] = (sigmaSB*self.params['Tint']**4)/(4.0*np.pi) -
18        np.sum(weights*(fn[diff_bot,:]*J[diff_bot,:] -
19        fn[diff_bot-1,:]*J[diff_bot-1,:])/dtau[diff_bot,:],axis=1)
20
21 # top boundary condition of differential method - generally not applicable for
22    combined form since differential form is only used at BOA
23 top = np.where(diff_bot==0)[0]
24 # B          equation 18.81a
25 X[top,top,:] = weights*Gfn
26 # L          equation 18.81b
27 F[top] = (sigmaSB*self.params['Tint']**4)/(4.0*np.pi) -
28        np.sum(weights*(Gfn*J[0,:] - Hext))

```

Algorithm A.12. Solving the differential form layers within the **correct** function

A.4.4. Convective Equilibrium

An atmosphere is convectively unstable when the Schwarzschild criteria for convective instability is met:

$$\nabla > \nabla_{ad} \quad (\text{A.4.41})$$

where $\nabla = d\ln T/d\ln P$ is the logarithmic temperature gradient and ∇_{ad} is the adiabatic gradient. We consider $\nabla_{ad} = 2/7$, which holds for an ideal gas and is a sufficient estimate for our purposes. If the temperature gradient is greater than the adiabatic gradient, the gas will be warmer than its surroundings and therefore rise due to convective instability [1]. The Schwarzschild criteria is often met at layers deep in the atmosphere, where the atmosphere is opaque and radiative transfer is inefficient at transporting energy. In our implementation, we treat convective layers with Mixing Length Theory. A parcel of gas within a convective layer rises when it is warmer than its surroundings. The Mixing Length Theory stipulates that after the gas has travelled the distance of a mixing length ℓ , it dissipates the energy into its surroundings, allowing for the efficient transport of heat from one altitude of the atmosphere to another; the resulting temperature gradient is less than if the heat were transported by radiation alone [2].

For convective layers, we add a convective flux term to Equations A.4.27 and A.4.28 to give us:

$$\int_0^\infty \kappa_\nu (J_\nu - B_\nu) d\nu + \frac{\rho g}{4\pi} \frac{\partial F_{conv}}{\partial P} = 0 \quad (\text{A.4.42})$$

$$\int_0^\infty \frac{d(f_\nu J_\nu)}{d\tau_\nu} d\nu + \frac{F_{conv}}{4\pi} = \frac{\sigma_{SB}}{4\pi} T_{int}^4 \quad (\text{A.4.43})$$

where

$$F_{conv} = \left(\sqrt{\frac{gQH_P}{32}} (\rho c_P T) (\nabla - \nabla_{el})^{3/2} \left(\frac{\ell}{H_P} \right)^2 \right) \quad (\text{A.4.44})$$

Here H_P is the scale height, c_P is the specific heat at constant pressure, $Q \equiv -(d\ln \rho/d\ln T) = 1$ for an ideal gas, ℓ is the mixing length, taken to be $\approx H_P$, and

$$\nabla - \nabla_{el} = \frac{1}{2} \mathcal{B}^2 + (\nabla - \nabla_{ad}) - \mathcal{B} \left(\frac{1}{4} \mathcal{B}^2 + (\nabla - \nabla_{ad}) \right)^{1/2} \quad (\text{A.4.45})$$

where

$$\mathcal{B} = \frac{16\sqrt{2}\sigma_{SB}T^3}{\rho c_P (gQH_P)^{1/2} (\ell/H_P)} \frac{\tau_{el}}{1 + \frac{1}{2}\tau_{el}^2} \quad (\text{A.4.46})$$

and $\tau_{el} = \chi_R \ell$ is the optical thickness of the mixing length. We also define $H_{conv} = F_{conv}/4\pi$.

A.4.4.1. Prerequisite Calculations

We first calculate the temperature gradient ∇ of each layer for the current temperature profile to determine the convective layers where:

$$\nabla_d = \frac{d\ln T_d}{d\ln P_d} \quad (\text{A.4.47})$$

Hubeny and Mihalas also calculate H_{conv} , $\partial H_{conv}/\partial T$, and $\partial H_{conv}/\partial P$ at the middle of each layer so we determine the corresponding temperature and pressures in preparation.

```

1 grad = np.zeros([self.nLay])
2
3 grad[1:] = np.diff(np.log(T))/np.diff(np.log(self.p))
4 grad[0] = grad[1]
5
6 grad_ad = 2./7.
7 convLay = np.where(grad >= grad_ad)[0]
8
9 self.convLay = deepcopy(convLay)
10
11
12 Tmid = np.hstack([T[0],0.5*(T[:self.nLay-1]+T[1:self.nLay])])
13 Pmid = np.hstack([self.p[0],0.5*(self.p[:self.nLay-1]+self.p[1:self.nLay])])

```

Algorithm A.13. Calculate ∇ and temperatures and pressures at the middle of each layer

We then solve for F_{conv} , $\nabla - \nabla_{el}$, and H_{conv} with the function `getHconv`, as well as $\partial H_{conv}/\partial T$, and $\partial H_{conv}/\partial P$ with the function `getdHconv`.

```

1 Hconv,grad_diff,rho = self.getHconv(Tmid,Pmid,grad,grad_ad,chi)
2
3
4 def getHconv(self,T,P,grad,grad_ad,chi):
5     z,dz,grav,ntot,nmol,MuAve,scaleHeight,RpBase,r =
6         self.calcHydroEqui(self.modelSetting,self.params,T)
7
8     rho = ntot*MuAve
9     Hp = scaleHeight #P/(self.grav*rho)
10    Q = 1.
11    sizel = Hp
12    MuAveLay =
13        np.hstack([MuAve[0],0.5*(self.MuAve[:self.nLay-1]+self.MuAve[1:self.nLay])])
14    Rconst = 8.3144598 / (MuAveLay/uAtom * 0.001) #J/mol*K / kg/mol = J/kg*K
15    Cp = 7./2. * Rconst # 5./2.*R Monotomic, 7./2.*R #Diatomic, 4*R #Polyatomic
16        units of J/kg K
17    dplnk_dT = self.Planck_deriv(T)
18    tau_el=np.zeros([self.nLay])
19    for i in range(self.nLay):
20        tau_el[i] = self.ChiRoss_not_over_rho(dplnk_dT[i,:],chi[i])*sizel[i]

```

```

19
20     beta =
21         ((16.*np.sqrt(2.)*sigmaSB*T**3.)/(rho*Cp*(self.grav*Q*Hp)**0.5*(size1/Hp)))*
22         (tau_el/(1.+0.5*tau_el**2.))
23     grad_diff = (grad-grad_ad) + (0.5*beta**2. -
24         beta*(0.25*beta**2.+(grad-grad_ad)**0.5)
25     Fconv = (self.grav*Q*Hp/32. )**0.5*(rho*Cp*T)*(grad_diff)**1.5*(size1/Hp)**2.
26     Hconv = Fconv/(4.*np.pi)
27
28     return Hconv,grad_diff,rho

```

Algorithm A.14. Finding F_{conv} , $\nabla - \nabla_{el}$, and H_{conv} within the `getHconv` function

```

1  dHconv_T,dHconv_P = self.getdHconv(Tmid,Pmid,grad,grad_ad,Hconv,chi)
2
3  def getdHconv(self,T,P,grad,grad_ad,Hconv,chi):
4      dT = T*0.0001
5      dp = P*0.0001
6
7      chiNew = self.calcOpacities(self.modelSetting,self.params,T+dT)[0]
8      Hconv_T, grad_diff,rho = self.getHconv(T + dT,P,grad,grad_ad,chiNew)
9      Hconv_P, grad_diff,rho = self.getHconv(T,P + dp,grad,grad_ad,chi)
10
11     dHconv_T = (Hconv_T - Hconv)/dT
12     dHconv_P = (Hconv_P - Hconv)/dp
13     return dHconv_T,dHconv_P

```

Algorithm A.15. Calculating $\partial H_{conv}/\partial T$ and $\partial H_{conv}/\partial P$ within the `getdHconv` function.

Linearizing equations A.4.42 and A.4.43 yield the convective terms that must be added to the integral and differential form matrices for the appropriate layers. More specifically, the \mathbf{A} and \mathbf{F} matrices, corresponding to radiative equilibrium, are modified to account for radiative-convective equilibrium. We also introduce the \mathbf{Q} and \mathbf{R} matrices, corresponding to elements of the grand matrix which are related to ∇ .

A.4.4.2. Integral Form

For convective layers in the top of the atmosphere ($P < 1 \times 10^5 Pa$), we linearize equation A.4.42 to find the additional terms required to account for convection.

$$\mathbf{A}_{A,d_{conv}}^{conv} = - \left[\left(\left(\frac{\partial H_{conv}}{\partial T} \right)_{d-1/2} + \left(\frac{\partial H_{conv}}{\partial P} \right)_{d-1/2} kN_{d-1} \right) \frac{\rho_d}{2\Delta m_d} \right] \quad (\text{A.4.48})$$

$$\mathbf{A}_{B,d_{conv}}^{conv} = \left[\left(\frac{\partial H_{conv}}{\partial T} \right)_{d+1/2} + \left(\frac{\partial H_{conv}}{\partial P} \right)_{d+1/2} kN_d \right. \\ \left. - \left(\frac{\partial H_{conv}}{\partial T} \right)_{d-1/2} - \left(\frac{\partial H_{conv}}{\partial P} \right)_{d-1/2} kN_d \right] \frac{\rho_d}{2\Delta m_d} \quad (\text{A.4.49})$$

$$\mathbf{A}_{C,d_{conv}}^{conv} = - \left[- \left(\left(\frac{\partial H_{conv}}{\partial T} \right)_{d+1/2} + \left(\frac{\partial H_{conv}}{\partial P} \right)_{d+1/2} kN_{d-1} \right) \frac{\rho_d}{2\Delta m_d} \right] \quad (\text{A.4.50})$$

$$\mathbf{Q}_{A,d_{conv}}^{conv} = - \left[\frac{H_{conv,d-1/2}}{\nabla_{d-1/2} - \nabla_{el,d-1/2}} \frac{3\rho_d}{4\Delta m_d} \right] \quad (\text{A.4.51})$$

$$\mathbf{Q}_{B,d_{conv}}^{conv} = \left(\frac{H_{conv,d+1/2}}{\nabla_{d+1/2} - \nabla_{el,d+1/2}} - \frac{H_{conv,d-1/2}}{\nabla_{d-1/2} - \nabla_{el,d-1/2}} \right) \frac{3\rho_d}{2\Delta m_d} \quad (\text{A.4.52})$$

$$\mathbf{Q}_{C,d_{conv}}^{conv} = - \left[\left(\frac{H_{conv,d+1/2}}{\nabla_{d+1/2} - \nabla_{el,d+1/2}} \right) \frac{3\rho_d}{2\Delta m_d} \right] \quad (\text{A.4.53})$$

$$\mathbf{F}_{d_{conv}}^{conv} = - \frac{\rho_d}{\Delta m_d} (H_{conv,d+1/2} - H_{conv,d-1/2}) \quad (\text{A.4.54})$$

```

1 #if layer uses integral form
2 if np.where(int_top == d)[0].size == 1:
3     A[d,d-1] += -((dHconv_T[d]+dHconv_P[d]*(self.p[d-1]/T[d-1]))*0.5*rho[d]/
4                 (self.p[d+1]/self.grav[d]-self.p[d-1]/self.grav[d]))
5     A[d,d]   += (((dHconv_T[d+1]+dHconv_P[d+1]*(self.p[d]/T[d])) -
6                 (dHconv_T[d-1]+dHconv_P[d-1]*(self.p[d]/T[d])))*0.5*rho[d]/
7                 (self.p[d+1]/self.grav[d]-self.p[d-1]/self.grav[d]))
8
9     if grad_diff[d+1] == 0: # Do not allow division by 0
10        Q[d,d-1] = -((Hconv[d]/grad_diff[d])*(1.5*rho[d])/
11                    (self.p[d+1]/self.grav[d]-self.p[d-1]/self.grav[d]))

```

```

11     Q[d,d] = ((0-Hconv[d]/grad_diff[d])*(3.*rho[d]/
12             (self.p[d+1]/self.grav[d]-self.p[d-1]/self.grav[d])))
13     Q[d,d+1] = 0
14     else:
15     Q[d,d-1] = -((Hconv[d]/grad_diff[d])*(1.5*rho[d])/
16             (self.p[d+1]/self.grav[d]-self.p[d-1]/self.grav[d]))
17     Q[d,d] =
18             ((Hconv[d+1]/grad_diff[d+1]-Hconv[d]/grad_diff[d])*(3.*rho[d]/
19             (self.p[d+1]/self.grav[d]-self.p[d-1]/self.grav[d])))
20     Q[d,d+1] = -((Hconv[d+1]/(grad_diff[d+1]))*(3.*rho[d]/
21             (self.p[d+1]/self.grav[d]-self.p[d-1]/self.grav[d])))
22
23     F[d] += -(Hconv[d+1]-Hconv[d]*2.)*rho[d]/
24             (self.p[d+1]/self.grav[d]-self.p[d-1]/self.grav[d])

```

Algorithm A.16. Solving the additional matrices for the integral form convective layers within the `correcT` function.

A.4.4.3. Differential form

Similarly, for convective layers at the bottom of the atmosphere ($P \geq 1 \times 10^5 Pa$), we linearize equation A.4.43 to find the additional terms required to account for convection. For exoplanets, convective layers are generally deep within the atmosphere and thus usually are modelled with the differential form.

$$\mathbf{A}_{A,d_{conv}}^{conv} = - \left[-\frac{1}{2} \left(\frac{\partial H_{conv}}{\partial T} \right)_{d-1/2} \quad -\frac{1}{2} \left(\frac{\partial H_{conv}}{\partial P} \right)_{d-1/2} \quad kN_{d-1} \right] \quad (\text{A.4.55})$$

$$\mathbf{A}_{B,d_{conv}}^{conv} = \frac{1}{2} \left(\frac{\partial H_{conv}}{\partial T} \right)_{d-1/2} + \frac{1}{2} \left(\frac{\partial H_{conv}}{\partial P} \right)_{d-1/2} \quad kN_d \quad (\text{A.4.56})$$

$$\mathbf{Q}_{B,d_{conv}}^{conv} = \frac{\frac{3}{2} H_{conv, d-1/2}}{\nabla_{d-1/2} - \nabla_{el, d-1/2}} \quad (\text{A.4.57})$$

$$\mathbf{F}_{d_{conv}}^{conv} = -H_{conv, d-1/2} \quad (\text{A.4.58})$$

$$\frac{\partial H_{conv}}{\partial T} = \frac{H_{conv}(T + \delta T, P, \nabla) - H_{conv}(T, P, \nabla)}{\delta T} \quad (\text{A.4.59})$$

$$\frac{\partial H_{conv}}{\partial P} = \frac{H_{conv}(P + \delta P, T, \nabla) - H_{conv}(P, T, \nabla)}{\delta P} \quad (\text{A.4.60})$$

```

1 # If layer uses differential form
2 if np.where(diff_bot == d)[0].size == 1:

```

```

3   # A           equation 18.88b
4   A[d,d-1] += -(-(0.5*dHconv_T[d])-(0.5*dHconv_P[d]))*(self.p[d-1]/T[d-1]))
5   # B           equation 18.88d
6   A[d,d]  +=   (0.5*dHconv_T[d])+(0.5*dHconv_P[d]))*(self.p[d]/T[d])
7   # B           equation 18.88e
8   Q[d,d] = (3./2. * Hconv[d]) / (grad_diff[d])
9   # L           equation 18.88f
10  F[d] += -(Hconv[d])

```

Algorithm A.17. Solving the additional matrices for the differential form convective layers within the `correct` function.

Finally, we calculate the additional convective terms that are universal for both the integral and differential forms:

$$\mathbf{R}_{A,d_{conv}}^{conv} = - \left[\frac{2T_d \nabla_d}{T_d^2 - T_{d-1}^2} - \frac{2P_d \nabla_d}{P_d^2 - P_{d-1}^2} k N_{d-1} \right] \quad (\text{A.4.61})$$

$$\mathbf{R}_{B,d_{conv}}^{conv} = \frac{2T_{d-1} \nabla_d}{T_d^2 - T_{d-1}^2} - \frac{2P_{d-1} \nabla_d}{P_d^2 - P_{d-1}^2} k N_d \quad (\text{A.4.62})$$

$$\mathbf{F}_{d_{conv}}^{conv} = \nabla_{d-1/2} - \frac{T_d - T_{d-1}}{T_d + T_{d-1}} \cdot \frac{P_d + P_{d-1}}{P_d - P_{d-1}} = 0 \quad (\text{A.4.63})$$

```

1  R[d,d-1] = ((2*T[d]*grad[d])/(T[d]**2-T[d-1]**2) -
2             (2*self.p[d]*grad[d])/(self.p[d]**2-self.p[d-1]**2))*
3             (self.p[d-1]/T[d-1]))
4  R[d,d]   = ((2*T[d-1]*grad[d])/(T[d]**2-T[d-1]**2) -
5             (2*self.p[d-1]*grad[d])/(self.p[d]**2-self.p[d-1]**2))*(self.p[d]/T[d]))

```

Algorithm A.18. Calculate the additional matrices universal to both the integral and differential forms for convective layers within the `correct` function.

A.4.5. Solving for the Temperature Correction

Finally, combining the matrix equations from the linearized RTE and radiative-convective equilibrium, we obtain the following grand matrix:

$$\begin{pmatrix} \mathbf{U}_0 & 0 & \cdots & \cdots & 0 & \mathbf{V}_0 \\ 0 & \mathbf{U}_1 & 0 & \cdots & \vdots & \mathbf{V}_1 \\ \vdots & 0 & \mathbf{U}_2 & 0 & \vdots & \mathbf{V}_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{U}_{NF-1} & \mathbf{V}_{NF-1} \\ \mathbf{X}_0 & \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_{NF-1} & \mathbf{A}_{r+c} \end{pmatrix} \begin{pmatrix} \delta \mathbf{J}_0 \\ \delta \mathbf{J}_1 \\ \delta \mathbf{J}_2 \\ \vdots \\ \delta \mathbf{J}_{NF-1} \\ \delta \mathbf{T} \end{pmatrix} \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{E}_1 \\ \mathbf{E}_2 \\ \vdots \\ \mathbf{E}_{NF-1} \\ \mathbf{F}_{r+c} \end{pmatrix} \quad (\text{A.4.64})$$

where the convective terms are incorporated into the \mathbf{A}_{r+c} and \mathbf{F}_{r+c} terms by:

$$\mathbf{A}_{r+c} = (\mathbf{A} + \mathbf{A}_{conv}) + \mathbf{Q}_{conv} \mathbf{R}_{conv} \quad (\text{A.4.65})$$

$$\mathbf{F}_{r+c} = \mathbf{F} + \mathbf{F}_{conv} \quad (\text{A.4.66})$$

We can write the $\delta \mathbf{J}_k$ in terms of $\delta \mathbf{T}$ to obtain:

$$\delta \mathbf{J}_k = (\mathbf{U}_k^{-1} \mathbf{E}_k) - (\mathbf{U}_k^{-1} \mathbf{V}_k) \delta \mathbf{T} \quad (\text{A.4.67})$$

and substituting the above into A.4.29 allows us to solve for $\delta \mathbf{T}$

$$\begin{aligned} & \sum_{k=0}^{NF-1} \mathbf{X}_k \delta \left((\mathbf{U}_k^{-1} \mathbf{E}_k) - (\mathbf{U}_k^{-1} \mathbf{V}_k) \delta \mathbf{T} \right) + \mathbf{A}_{r+c} \delta \mathbf{T} = \mathbf{F}_{r+c} \\ & \left(\mathbf{A}_{r+c} - \sum_{k=0}^{NF-1} \mathbf{X}_k (\mathbf{U}_k^{-1} \mathbf{V}_k) \right) \delta \mathbf{T} = \left(\mathbf{F}_{r+c} - \sum_{k=0}^{NF-1} \mathbf{X}_k (\mathbf{U}_k^{-1} \mathbf{E}_k) \right) \end{aligned} \quad (\text{A.4.68})$$

We show how $\delta \mathbf{T}$ and $\delta \mathbf{J}_k$ are calculated in SCARLET below:

```

1  ## Now we solve for delta T
2  U_moved = np.moveaxis(U,-1,0)
3  U_inv_moved = np.linalg.inv(U_moved)
4  V_moved = np.moveaxis(V,-1,0)
5  UV_moved = np.matmul(U_inv_moved,V_moved) # the 3 matmul lines are taking most
      of the time
6  UV = np.moveaxis(UV_moved,0,-1)
7  X_moved = np.moveaxis(X,-1,0)
8  XUV = np.matmul(X_moved, UV_moved)
9  XUV_summed = np.sum(XUV,0)
10 A_conv = A + np.matmul(Q,R)

```

```

11 LHS = A_conv - XUV_summed
12 U_inv = np.moveaxis(U_inv_moved,0,-1)
13
14 UE = np.einsum('ijk,jk->ik', U_inv, E)
15 XUE = np.einsum('ijk,jk->ik', X, UE)
16 XUE_summed = np.sum(XUE, axis=1)
17
18 RHS = F - XUE_summed
19 deltaT = np.dot(np.linalg.inv(LHS),RHS)
20
21 dT = np.tile(deltaT[:,np.newaxis],(1,self.nWave))
22 UVT = np.einsum('ijk,jk->ik', UV, dT)
23
24 deltaJ = UE - UVT

```

Algorithm A.19. Solving for the temperature correction within the `correct` function.

A.4.6. Arriving at a Self-Consistent Temperature Profile

The Rybicki method solves the RTE and radiative-convective equilibrium equations, giving us a correction for the temperature at each layer. We then apply this temperature correction to the temperature profile, yielding a new temperature structure that is closer to the converged solution and iterate until reaching a threshold of $\delta\mathbf{T}/\mathbf{T} = 10^{-3}$.

```

1 def
   calcNonGrayTpProf(self,modelSetting,params,firstIter,LucyUnsold,runConvection):
2     # Linearized Tp model
3     if (modelSetting['maxNonGrayIter'] is None) or
        (modelSetting['maxNonGrayIter'] > self.nonGrayIter):
4         print 'Calculate Self-Consistent TP:'
5
6         if self.Teq<400:
7             print 'Quadrupling the max # of iterations to accomodate low Teq of
                planet'
8             modelSetting['maxIterForNonGrayTpCalc'] =
                modelSetting['maxIterForNonGrayTpCalc']*4
9
10        elif self.Teq<800:
11            print 'Tripling the max # of iterations to accomodate low Teq of
                planet'

```

```

12         modelSetting['maxIterForNonGrayTpCalc'] =
13             modelSetting['maxIterForNonGrayTpCalc']*3
14
15     #Set initital conditions for temperature profile T
16     if LucyUnsold:
17         # Use Lucy Unsold tp profile as starting point if it was used
18         T = ut.loadpickle('test.pkl') # Load from Lucy Unsold
19     else:
20         if (firstIter is True):
21             if 'Tprof' in params:
22                 T = np.interp(np.log(self.p),np.log(params['Tprof'][0]),
23                             params['Tprof'][1])
24             else:
25                 T=self.Teq*np.ones_like(self.p)
26         else:
27             T=self.T #Use temperature structure from previous iteration as
28                 initital condition
29
30     # Initialize chemistry, hydrostatic equilibrium, and opacities
31     self.qmol_lay = self.calcComposition(modelSetting,params,T)
32     self.z,self.dz,self.grav,self.ntot,self.nmol,self.MuAve,self.scaleHeight,
33     self.RpBase,self.r=
34         self.calcHydroEqui(modelSetting,params,T)
35     self.extinctCoef,self.absorbCoef, self.scatCoef =
36         self.calcOpacities(modelSetting,params,T)
37
38     # calculate integration weights so that later we can sum instead of
39     integrate
40     weights = self.calcWeight()
41
42     # Run RTE solver
43     B,J,K,H,fn,Gfn,dtau,lmbda,extinctCoef,mExtinctCoef,absorbCoef =
44         self.solveRTE(T,modelSetting,params,self.IrradStarEffIntensityPerHz,
45         TpCorrLin = True)
46
47     # Save iterations of T and dT
48     TList = deepcopy(T)
49     dTList = np.zeros([self.nLay])
50     errorList = np.zeros([self.nLay])

```

```

47     # Iterate until convergence criteria of T are met
48     runConvection = False # Don't run convection until after 1st convergence
        for stability
49
50     for loop in range(2):
51         i = 1
52         not_converged = True
53         prev_best_dT = 500.
54
55         prev_iter = np.zeros(self.nLay)
56
57         while(not_converged):
58             print i,
59
60             nNonConvergedLay = 0
61
62             # Call Tp linearization method
63             dT,dJ=
                self.correcT(T,B,J,fn,Gfn,self.IrradStarEffIntensityPerHz,dtau,
64                 lambda,mExtinctCoef,extinctCoef,absorbCoef,weights,loop,
65                 runConvection)
66
67
68             #Slow down jumps if dT is too large (for stability)
69             for k in range(self.nLay):
70                 if np.abs(dT[k])> 0.6*T[k]:
71                     dT[k] = 0.6*T[k]*np.sign(dT[k])
72                 if self.Teq<400:
73                     if np.abs(dT[k])>0.3*T[k]:
74                         dT[k] = 0.3*T[k]*np.sign(dT[k])
75
76                 if runConvection and self.Teq<800: # Slow down even more when
                    running convection
77                     if np.abs(dT[k])>0.4*T[k]:
78                         dT[k] = 0.4*T[k]*np.sign(dT[k])
79
80                 elif runConvection and self.Teq<400:
81                     if np.abs(dT[k])>0.2*T[k]:
82                         dT[k] = 0.2*T[k]*np.sign(dT[k])
83

```

```

84         # If iteration is -ve of the previous iteration, only apply
           1/2 of the dT
85         if np.round(dT[k]) == np.round(prev_iter[k])*-1.:
86             dT[k] = dT[k] / 2.
87
88     prev_iter = deepcopy(dT)
89     # Calculate new T (in this next iteration)
90     T = T+dT
91
92     # Check temperature convergence of all layers
93     for d in range(self.nLay):
94         if np.abs(dT[d]/(T[d]-dT[d])) < 1e-3: #Convergence criteria
95             nNonConvergedLay += 0
96         else:
97             nNonConvergedLay += 1
98
99     if nNonConvergedLay == 0:
100         #--> Tp profile is converged for this molecular composition
101         not_converged = False
102         if loop == 0:
103             print ''
104             print 'Checking chemistry/hydrostatic equilibrium
                    convergence',
105             if modelSetting['TempType']=='NonGrayConv':
106                 print 'and adding convection'
107                 runConvection = True
108                 self.convLay = '(Not yet computed)'
109             print ''
110         if loop == 1:
111             print ''
112             print 'dJ: ', np.amax(np.abs(dJ))
113             print 'Successfully converged'
114
115     elif np.amax(np.abs(dT)) < prev_best_dT:
116         # Save best iteration
117         best_T = deepcopy(T)
118         prev_best_dT = np.amax(np.abs(dT))
119         if runConvection:
120             prev_best_convLay = self.convLay
121

```

```

122
123     # Do not allow for inversions when running convection for
124     # challenging target
125     if runConvection and self.Teq<400:
126         if len(self.convLay) != 0:
127             for k in range(self.nLay):
128                 if k >= self.convLay[0] and T[k] < T[k-1]:
129                     dT[k] = T[k-1]-T[k]
130                     T[k] = T[k-1]
131
132     # Save iterations of T and dT
133     TList = np.c_[TList,T]
134     dTList = np.c_[dTList,dT]
135
136     errorList = np.c_[errorList,np.abs(dT/(T-dT))]
137
138     self.errorList = errorList
139     self.dTList = dTList
140     self.TList = TList
141
142     if not_converged:
143         # Iterate
144         if loop >= 1: # Add stability by changing
145             # chemistry/hydroequilibrium only after 1st convergence
146             self.qmol_lay = self.calcComposition(modelSetting,params,T)
147             self.z,self.dz,self.grav,self.ntot,self.nmol,
148                 self.MuAve,self.scaleHeight,self.RpBase,self.r =
149                 self.calcHydroEqui(modelSetting,params,T)
150
151             B,J,K,H,fn,Gfn,dtau,lmbda,
152                 extinctCoef,mExtinctCoef,absorbCoef =
153                 self.solveRTE(T,modelSetting,params,
154                 self.IrradStarEffIntensityPerHz,TpCorrLin=True)
155
156             i+=1
157
158     if self.plotTpChangesEveryIteration:
159         self.plotTpChanges(save=True,close=True,loop=loop)
160

```

```

159         #Display some convergence measures (just for user to see
           convergence)
160         IdownPerHz = self.IrradStarEffIntensityPerHz
161         IupPerHz = 4.0 * H[0,:] - IdownPerHz
162         FupPerHz = pi * IupPerHz
163         TotalFluxUpTOA = - np.trapz(x=self.f,y=FupPerHz) #Minus sign
           because self.f goes from high to low
164         TotalIrradTOA = -
           np.trapz(x=self.f,y=self.IrradStarEffIntensityPerHz) * np.pi
           #Minus sign because self.f goes from high to low
165         netFluxTOA=TotalFluxUpTOA-TotalIrradTOA
166         print '\nTotal flux up at TOA: %6g W/m**2'%(TotalFluxUpTOA)
167         print 'Total irradiation at TOA: %6g W/m**2'%(TotalIrradTOA)
168         if runConvection:
169             print 'dTmax = {:7g} | dJmax = {:7g} | flux residual: {:7g}
           W/m**2 ({:5g}%) |
           Conv={}' .format(np.amax(np.abs(dT)),np.amax(np.abs(dJ)),
170             netFluxTOA, netFluxTOA/TotalIrradTOA*100,str(runConvection)+
           ' for layers:' + str(self.convLay))
171         else:
172             print 'dTmax = {:7g} | dJmax = {:7g} | flux residual: {:7g}
           W/m**2 ({:5g}%) |
           Conv={}' .format(np.amax(np.abs(dT)),np.amax(np.abs(dJ)),
173             netFluxTOA,netFluxTOA/TotalIrradTOA*100,runConvection)
174
175
176         #Check whether maximum number of iterations is reached (user
           specified)
177         if i == modelSetting['maxIterForNonGrayTpCalc']:
178             print 'Reached max # of iterations: using Tp profile with
           lowest dT (dTmax = {:7g})' .format(prev_best_dT)
179             self.TList = TList
180             not_converged=False
181             T = deepcopy(best_T)
182             self.convLay = prev_best_convLay
183
184
185         self.TList = TList
186         self.nonGrayIter += 1
187
188     return T

```

Algorithm A.20. Calculating the self-consistent temperature profile in the `calcNonGrayTpProf` function.

References

- [1] Ivan Hubeny and Dimitri Mihalas. *Theory of Stellar Atmospheres: An Introduction to Astrophysical Non-equilibrium Quantitative Spectroscopic Analysis*. Princeton University Press, October 2014. Google-Books-ID: TmuYDwAAQBAJ.
- [2] Edward A. Spiegel. A Generalization of the Mixing-Length Theory of Turbulent Convection. *The Astrophysical Journal*, 138:216, July 1963.

Appendix B

Variable Names and Definitions

Variable	Name in SCARLET	Definition
General variables:		
T	T	Temperature
P	P	Pressure
τ		Optical depth
ν	<code>self.freqs</code> converted to <code>self.wave</code>	Frequency (converted to wave-lengths for SCARLET)
z	z	Altitude
dz	dz	Change in altitude
ρ	rho	Density
k_B	kBoltz	Boltzmann constant
g	grav	Boltzmann constant
\bar{m}	MuAve	Mean molecular mass
κ_ν	absorbCoef	Absorption coefficient
ω_ν	scatCoef	Scattering coefficient
χ_ν	extinctCoef	Extinction coefficient
λ_ν	lmbda	κ_ν/χ_ν
ω_ν	mExtinctCoef	Extinction coefficient by mass
$\Delta\tau_\nu$	dtau	Change in optical depth
B	B	Planck Function
μ	mu	Cosine of angle θ relative to the vertical of a layer
I_ν		Specific intensity
S_ν		Source function
$u_{\mu\nu}$		$u_{\mu\nu} \equiv 1/2(I_\nu^+ + I_\nu^-)$
$v_{\mu\nu}$		$v_{\mu\nu} \equiv 1/2(I_\nu^+ - I_\nu^-)$

w_j	weight	Integration weights for Feautrier method
J_ν	J	Mean intensity of radiation
H_ν	H	1 st moment of specific intensity
K_ν	K	2 nd moment of specific intensity
H_{ext}	Hext	External stellar irradiation
f_ν	fn	K_ν/J_ν
g_ν	Gfn	$H_\nu(\tau = 0)/J_\nu(\tau = 0)$

Solving the RTE (within **FeautrierRTE**):

\mathbf{A}_d	A	See equation A.3.28
\mathbf{B}_d	B	See equation A.3.28
\mathbf{C}_d	C	See equation A.3.28
\mathbf{L}_d	L	See equation A.3.28
\mathbf{u}_d	U	See equation A.3.28
\mathbf{D}_d	D	See equations A.3.45 and A.3.50
\mathbf{R}_d	R	See equations A.3.45 and A.3.50
$(\mathbf{B}_d - \mathbf{A}_d\mathbf{D}_{d-1})$	N	See equation A.3.52
$(\mathbf{A}_d\mathbf{R}_{d-1} + \mathbf{L}_d)$	X	See equation A.3.53

Rybicki method (within **correctT**):

\mathbf{U}_k	U	See equation A.4.2
\mathbf{V}_k	V	See equation A.4.2
\mathbf{E}_k	E	See equation A.4.2
$\delta\mathbf{J}_k$	deltaJ	Change in J_ν for each layer (see equation A.3.28)
$\delta\mathbf{T}$	deltaT	Temperature correction for each layer (see equation A.3.28)
α_d	alpha_di	See equation A.4.15
γ_d	gamma_di	See equation A.4.16
β_d	beta_di	See equation A.4.17
a_d	a_di	See equation A.4.18
c_d	c_di	See equation A.4.19
b_d	b_i	See equation A.4.25
α_d	alpha_di	See equation A.4.15
γ_d	gamma_di	See equation A.4.16
β_d	beta_di	See equation A.4.17

$\partial\chi_d/\partial T_d$	dchi_dT	Numerically solved partial derivatives from diffVar
$\partial\kappa_d/\partial T_d$	dkappa_dT	Numerically solved partial derivatives from diffVar
$\partial\omega_d/\partial T_d$	domega_dT	Numerically solved partial derivatives from diffVar
$\partial\lambda_d/\partial T_d$	dlambda_dT	Numerically solved partial derivatives from diffVar
$\partial(\kappa_d B_d)/\partial T_d$	deta_dT	Numerically solved partial derivatives from diffVar
$\partial B_d/\partial T_d$	dplnk_dT	Numerically solved partial derivatives from diffVar
\mathbf{X}_k	X	See equation A.4.29
\mathbf{A}_k	A	See equation A.4.29
\mathbf{F}_k	F	See equation A.4.29
w_k	weights	Integration weights for Rybicki method
σ_{SB}	sigmaSB	Stefan-Boltzmann constant
T_{int}	Tint	Internal heat of the planet

Convection prerequisites

(within **getHconv** and **getdHconv**):

∇_{ad}	grad_ad	Adiabatic gradient
∇_d	grad	Logarithmic temperature gradient
$\nabla_d - \nabla_{el,d}$	grad_diff	Difference between the temperature gradient and the mixing length gradient (See equation A.4.45).
H_P	Hp	Scale height of the atmosphere
Q	Q	$= -(d\ln\rho/d\ln T) = 1$ for an ideal gas
τ_{el}	tau_el	Optical thickness of the mixing length
H_{conv}	Hconv	H convective (See section A.4.4)
F_{conv}	Fconv	F convective (See equation A.4.44)
$(\partial H_{conv}/\partial T)_d$	dHconv_T	Numerically solved partial derivatives
$(\partial H_{conv}/\partial P)_d$	dHconv_P	Numerically solved partial derivatives

Convection (within **correcT**):

kN_d	self.p[d]/T[d]	k , the Boltzmann constant, multiplied
--------	----------------	--

		by N_d , the total particle number density
Δm_d	<code>self.p[d+1]/self.grav[d]- self.p[d-1]/self.grav[d])</code>	Relates pressure and gravity changes in neighboring layers
\mathbf{A}_d^{conv}	A	See section A.4.4
\mathbf{Q}_d^{conv}	Q	See section A.4.4
\mathbf{R}_d^{conv}	R	See section A.4.4
\mathbf{F}_d^{conv}	F	See section A.4.4