PICASO 3.0: A One-dimensional Climate Model for Giant Planets and Brown Dwarfs

Sagnick Mukherjee1, Natasha E. Batalha2, Jonathan J. Fortney1, and Mark S. Marley3

1 Department of Astronomy and Astrophysics, University of California, Santa Cruz, CA 95064, USA; samukher@ucsc.edu
2 NASA Ames Research Center, MS 245-3, Moffett Field, CA 94035, USA
3 Lunar and Planetary Laboratory, The University of Arizona, Tucson, AZ 85721, USA

Received 2022 June 6; revised 2022 October 17; accepted 2022 October 20; published 2023 January 12

Abstract

Upcoming James Webb Space Telescope observations will allow us to study exoplanet and brown dwarf atmospheres in great detail. The physical interpretation of these upcoming high signal-to-noise observations requires precise atmospheric models of exoplanets and brown dwarfs. While several 1D and 3D atmospheric models have been developed in the past three decades, these models have often relied on simplified assumptions like chemical equilibrium and are also often not open-source, which limits their usage and development by the wider community. We present a Python-based 1D atmospheric radiative-convective equilibrium (RCE) model. This model has heritage from the Fortran-based code, which has been widely used to model the atmospheres of solar system objects, brown dwarfs, and exoplanets. In short, the basic capability of the original model is to compute the atmospheric state of the object under RCE given its effective or internal temperature, gravity, and host-star properties (if relevant). In the new model, which has been included within the well-utilized code-base Picaso, we have added these original features as well as the new capability of self-consistently treating disequilibrium chemistry. This code is widely applicable to hydrogen-dominated atmospheres (e.g., brown dwarfs and giant planets).

Unified Astronomy Thesaurus concepts: Brown dwarfs (185); Exoplanets (498)

Supporting material: animations

1. Introduction

There are three broad categories of substellar atmosphere modeling frameworks: 1D physically self-consistent models, 3D general circulation models (GCMs), and atmospheric retrieval. Atmospheric models enable the understanding of physical and chemical processes driving the behavior of substellar atmospheres. When combined with observational data, they can be used to infer the physical state of atmospheres. Additionally, the adiabat calculated by the atmospheric models provides the upper boundary conditions of the interior evolutionary models of substellar objects like giant planets and brown dwarfs (Hubbard 1977; Burrows et al. 1997; Chabrier et al. 2000; Saumon & Marley 2008). Although substellar atmospheres generally constitute a small fraction (∼1%) of the total mass of their bodies, their radiative properties control the overall cooling of objects throughout their evolution. Therefore, atmospheric models are crucial for understanding both physical (e.g., climate and chemistry) and global properties of substellar objects (e.g., radius/luminosity evolution over time).

A variety of 1D radiative–convective–thermochemical equilibrium models of exoplanetary and brown dwarf atmospheres have been developed (e.g., Marley et al. 1996; Barman et al. 2001; Fortney et al. 2005, 2008; Phillips et al. 2020; Marley et al. 2021). These models originate both from stellar atmospheric modeling codes (e.g., Seager & Sasselov 1998; Barman et al. 2001; Sudarsky et al. 2003) and planetary atmospheric modeling codes (e.g., Marley et al. 1996; Baudino et al. 2015). These models rely on iterating to a self-consistent radiative-convective equilibrium (RCE) solution. Self-consistency here means that the model iterates on all components of the atmospheric structure simultaneously such that they are physically consistent with each other (e.g., chemical abundances are consistent with the current temperature profile). The models do not capture 3D dynamics, but instead are able to incorporate the effects of radiative/convective energy transport, chemistry, and clouds. These 1D models are also computationally efficient, which enhances their exploratory power significantly over 3D circulation models. For example, Zhang et al. (2021) used the Sonora Bobcat grid of models, generated using the 1D Radiative-Convective Thermochemical Equilibrium (RCTE) model described in Marley et al. (2021), to infer the physical properties of 55 T-dwarfs in a uniform analysis. The computational speed of these 1D self-consistent models along with their ability to treat atmospheric physics and chemistry self-consistently makes them a powerful tool for interpreting atmospheres of planets and brown dwarfs.

In addition to the model used in Marley et al. (2021), several other independent 1D RCTE models have been widely used in the literature (e.g., Burrows et al. 2008; Tremblin et al. 2015; Gandhi & Madhusudhan 2017; Malik et al. 2017; Phillips et al. 2020; Marley et al. 2021). For example, the Genesis code from Gandhi & Madhusudhan (2017) was used for the high significance detection of various carbon and nitrogen bearing species in the atmosphere of HD 209458b by Giacobbe et al. (2021). The ATMO model (Tremblin et al. 2015; Phillips et al. 2020) was used by Goyal et al. (2018) to perform a uniform fitting analysis of the transmission spectra of 10 hot Jupiters, which led to important conclusions about their atmospheric composition. Despite the many insights gained from these models, their simplicity often makes them insufficient to explain a variety of atmospheric spectra (e.g., Noll et al. 1997; Oppenheimer et al. 1998; Saumon et al. 2015).
In this work, we focus on 1D physically motivated RCE models for substellar atmospheres. In addition, to include complexities like disequilibrium chemistry within these models, another important modification we must make to these codes is to transform them to open-sourced, "FAIR" codes (findable, accessible, interpretable, and reproducible). Doing so will enable the community to interpret the upcoming influx of data from missions like JWST (Pontoppidan et al. 2022). Currently, there are a handful of 1D climate models that are open-source (e.g., HELIOS, Malik et al. 2017, 2019b; and TLUSTY, Hubeny 1988; Hubeny et al. 2003; Sudarsky et al. 2003). HELIOS is a Python-based GPU dependent model and it has been used for modeling atmospheres of a variety of substellar objects (e.g., Malik et al. 2019a; Fossati et al. 2021; Deline et al. 2022; Yan et al. 2022). TLUSTY is a stellar atmosphere model written in FORTRAN77. This code has been modified to be applied to substellar atmospheres as well and is called COOLTLUSTY (Hubeny et al. 2003). COOLTLUSTY also has been widely used for brown dwarfs and exoplanets (e.g., Burrows et al. 2006; Spiegel et al. 2010; Spiegel & Burrows 2012; Lacy et al. 2019; Lacy & Burrows 2020).

In this work, we aim to add to develop a new open-source Python-based 1D RCE model for H-dominated substellar atmospheres—PICASO 3.0 that is both (1) open-source and (2) capable of including disequilibrium chemistry induced by vertical mixing self-consistently within the 1D RCE framework. PICASO 3.0 has been released publicly as an extension of the already open-source 1D and 3D radiative transfer tool PICASO (Batalha et al. 2019). It is also accompanied by detailed tutorials exploring all of its uses and limitations. Here, we focus on the description of the numerical techniques used in the code along with benchmarking to previous studies to demonstrate its functionality.

In Section 2, we discuss the methodology of PICASO 3.0, the new Python atmospheric model. We present the benchmarking of our model with other models in Section 3 followed by recommendations on how to use this model in Section 4. We demonstrate the calculation of various observables using PICASO 3.0 in Section 5. We briefly discuss ongoing and future improvements in Section 6 and conclude in Section 7.

2. Model Setup of PICASO 3.0

The heritage of our code is the Fortran-based EGP substellar atmospheric model. The EGP model has been used for substellar atmospheres including solar system moons and planets, Titan and Uranus (McKay et al. 1989; Marley & McKay 1999), the L–T–Y brown dwarf sequence (e.g., Marley et al. 1996; Morley et al. 2012, 2014; Karalidi et al. 2021; Marley et al. 2021; Zhang et al. 2021), cloudy atmospheres (e.g., Cushing et al. 2008; Stephens et al. 2009), and a wide variety of extrasolar planets (e.g., Fortney et al. 2005, 2007, 2008; Marley et al. 2012; Morley et al. 2015, 2017; Fortney et al. 2020) for over two decades. It has also been used as the primary radiative transfer scheme in the SPARC GCM (Showman et al. 2009). Briefly, this 1D model solves for the self-consistent temperature, chemical, and cloud structure of H-dominated atmospheres under the assumption of RCE. The code structure of PICASO 3.0 is composed of Python classes, which generally include multiple Python functions. We refer to these classes and functions as modules here. In general, Python is significantly slower than Fortran. However, with the use of

4 https://docs.python.org/3/tutorial/classes.html
the “numba” “just-in-time” framework (Lam et al. 2015) to create on-the-fly compiled machine code, PICASO 3.0 has comparable run times to the original Fortran version. Figure 1 shows a simplified schematic of PICASO 3.0’s workflow.

First, the user provides an initial set of physical properties for the object to be modeled, shown within the red boxes at the top of Figure 1. The model supports both nonirradiated (e.g., brown dwarf) and irradiated (e.g., planets) calculations. Therefore, we specify optional inputs with dashed outlines.

For example, for modeling a field brown dwarf atmosphere, the user only must specify the \( T_{\text{eff}} \) (\( T_{\text{int}} \) in case of a planet), gravity, atmospheric metallicity, and C/O ratio of the brown dwarf. For modeling an irradiated planet, the user must specify the internal temperature of the planet \( T_{\text{int}} \), planet gravity, atmospheric metallicity, C/O ratio, semimajor axis, and host-star properties (e.g., stellar temperature, metallicity, and gravity). \( T_{\text{int}} \) represents the temperature obtained by converting the internal heat flow of the planet via the Stefan–Boltzmann law. Along with these inputs, the user must also specify an initial pressure–temperature profile \( (T(P)) \) guess, which is divided into plane-parallel logarithmically spaced pressure layers (\( \sim 60–90 \)). Then, through an iterative process, which we describe in Section 2.1, the model iterates through computing chemistry, opacities, and net upwelling and downwelling fluxes until a radiative-convective equilibrium threshold has been met. During this iteration, the model solves for both radiative and convective parts of the atmosphere and takes into account the possibility of multiple radiative and convective zones. Ultimately, the model produces the final atmospheric state of the object \( (T(P) \) and associated chemistry). These can then be used to compute...
transmission, emission, and/or reflected light spectra used to compare with observations (see Figure 1). We discuss these physical and chemical aspects one by one in Section 2.1.

2.1. Physics and Chemistry of Substellar Atmospheres

2.1.1. Radiative-convective Equilibrium

The key physical basis of this model is the assumption of RCE in substellar atmospheres. The radiative equilibrium represents the physical scenario where within each atmospheric layer the energy emitted must be balanced by the energy absorbed. This means that each atmospheric layer must allow for the transfer of the same amount of radiative energy through it. Hubeny (2017) provided a detailed derivation of the RCE criteria. For completeness, we present some of the important steps here. Radiative-equilibrium scenario is represented by:

\[ \int_0^{\infty} (\chi_\nu J_\nu - I_\nu) d\nu = 0 \]  
(1)

where \( \chi_\nu \) and \( I_\nu \) are absorption and emission coefficients, respectively, and \( J_\nu \) is the first moment of the intensity field. Assumption of local thermodynamic equilibrium allows Equation (1) to be rewritten as:

\[ \int_\nu \kappa_\nu(J_\nu - B_\nu) d\nu = 0 \]  
(2)

where \( \kappa_\nu \) is the wavelength-dependent total absorption opacity of the layer, and \( B_\nu \) is the local Planck function of the layer. This is the integral form of the radiative-equilibrium condition for each atmospheric layer. The integral form of the radiative-equilibrium condition (Equation (2)) is applicable and numerically stable throughout the atmosphere. Another method of computing the radiative-equilibrium condition, which is used by our model, is referred to as the differential form. The differential form is numerically stable at optically thicker parts of the atmosphere and might become numerically unstable at parts of the atmosphere that are optically thin \((\tau \ll 1)\). In practice, this instability only appears at parts of the atmosphere with low temperatures and very small optical depths. In our practice, this instability only appears at parts of the atmosphere that are optically thin. We discuss these instabilities pointed out by Hubeny (2017) in a later update. In our model to make our solutions more accurate at the optically thinner parts of the atmosphere.

The differential form can be derived by using Equation (2) along with the second moment of the radiative transfer equation,

\[ \int_\mu \mu dI_\nu(\mu) d\mu = \int_\mu \mu(I_\nu - B_\nu) d\mu, \]  
(3)

which under local thermodynamic equilibrium leads to the form,

\[ \frac{dH_\nu}{dz} = \kappa_\nu(J_\nu - B_\nu) \]  
(4)

where \( H_\nu \) is the wavelength-dependent second moment of the specific intensity field. Comparing Equations (2) and (4) leads to the condition,

\[ \int_\nu \frac{dH_\nu}{dz} d\nu = 0, \]  
(5)

which means that the integral of the second moment of the intensity, \( H_\nu \), must be constant at all layers. Mathematically, this can be written as:

\[ \int_\nu H_\nu d\nu - \frac{\sigma_b T_{\text{eff}}^4}{4\pi} = 0 \]  
(6)

where \( \sigma_b \) is the Stefan–Boltzmann constant, and \( T_{\text{eff}} \) is the effective temperature of the object. This calculation can be found in the \( \text{t}_{\text{start}} \) module in the code. The model tries to achieve this radiative equilibrium in all of the atmospheric layers that are stable against convection.

Given Equation (6), which defines the radiative-equilibrium condition, the general procedure of the model is to start with an initial guess of the \( T(P) \) profile of the atmosphere. This can be seen in the required inputs box at the top of Figure 1. Radiative fluxes through all of the atmospheric layers are then calculated with a chemical state of the atmosphere. These radiative fluxes are used to compute the wavelength integrated flux carried by each layer \( H(z) \). If the atmospheric layers are not in radiative equilibrium, this quantity \( H(z) \) would not be equal to the target flux given by \( \frac{\sigma_b T_{\text{eff}}^4}{4\pi} \), and the \( T(P) \) solution would be perturbed until the convergence criteria set by Equation (6) is met. The methodology of the convergence is discussed further in Section 2.2.

2.1.2. Radiative Transfer

In order to compute RCE with Equation (6), radiative fluxes must be calculated for all atmospheric layers. The fundamental equation describing atmospheric radiative transfer is

\[ \mu \frac{dI_\nu(\mu, \tau_\nu, \phi)}{d\tau_\nu} = I_\nu(\mu, \tau_\nu, \phi) - S_\nu(\mu, \tau_\nu, \phi) \]

\[ - \frac{\omega_0}{4\pi} \int_0^{2\pi} \int_{\phi'}^1 P_\nu(\mu, \mu', \phi, \phi') I_\nu(\mu', \tau_\nu, \phi') d\mu' d\phi' \]  
(7)

where \( \mu = \cos(i) \), in which \( i \) is the incident angle of the ray, \( \phi \) is the azimuthal angle, \( I_\nu \) is the specific intensity, \( \tau_\nu \) is the frequency-dependent optical depth, \( S_\nu \) is the frequency-dependent source function, and \( P_\nu \) is the scattering phase function. The first term on the right-hand side of Equation (7) describes the attenuation of the intensity with increasing optical depth. The source function \( S_\nu \) is used to describe the emission of the atmosphere itself (e.g., thermal emission) or any external radiation source (e.g., incident stellar radiation). The third term in Equation (7) represents scattering of the intensity within the atmosphere where the phase function \( P_\nu(\mu, \mu', \phi, \phi') \) is the probability that intensity \( I_\nu(\mu', \tau_\nu, \phi') \) will be scattered from the direction \((\mu', \phi')\) to the direction \((\mu, \phi)\).

In order to solve Equation (7), we follow the two-stream radiative transfer methodology described in Toon et al. (1989) for the calculation of radiative fluxes. The specific intensity \( I_\nu(\mu, \tau_\nu, \phi) \) can be integrated over the azimuthal angle \( \phi \) to calculate the azimuthally integrated intensity \( I_\nu(\mu, \tau_\nu) \).
upward and downward fluxes can also be defined by

\[ F^+_\nu = \int_0^1 \mu I^+_{\nu}(\mu, \tau) d\mu \]
\[ F^-_\nu = \int_0^1 \mu I^-_{\nu}(\mu, \tau) d\mu \]  

(8)

where \( I^+_{\nu}(\mu, \tau) \) is the azimuthally integrated intensity for upward values of \( \mu \), and \( I^-_{\nu}(\mu, \tau) \) is the azimuthally integrated intensity for downward values of \( \mu \). Equation (7) can be integrated to produce two separate coupled equations in terms of the upward and downward fluxes instead of \( \mu \) and \( \phi \) dependent intensities. These equations for the upward and downward fluxes are\(^5\)

\[ \frac{\partial F^+_\nu}{\partial \tau} = \gamma_1 F^+_\nu - \gamma_2 F^-_\nu - S^+_\nu \]
\[ \frac{\partial F^-_\nu}{\partial \tau} = \gamma_2 F^+_\nu - \gamma_1 F^-_\nu + S^-_\nu \]  

(9)

where \( \gamma_1 \) and \( \gamma_2 \) are functions of the scattering properties of the medium, and \( S^+_\nu \) and \( S^-_\nu \) are modified versions of the source function. The radiative transfer calculation within our model is divided into two distinct parts: (1) the transfer of thermal emission through the atmosphere, and (2) the transfer of the reflected external (e.g., stellar) radiation throughout the atmosphere. Equation (9) is applicable to both of these components, but the functions \( \gamma_1, \gamma_2, S^+_\nu \), and \( S^-_\nu \) are defined differently for each. A brief description of the radiative transfer of thermal radiation in our model is provided below followed by a discussion of the radiative transfer of external stellar radiation.

We use the two-stream source function technique described in Toon et al. (1989) to calculate the thermal upward and downward fluxes in each layer. The hemispheric-mean approximation approach is used in this method where \( \gamma_1 \) is given by \( 2 - \omega_0(1 + g) \) and \( \gamma_2 = \omega_0(1 + g) \). \( \omega_0 \) and \( g \) are the single scattering albedo and the scattering asymmetry parameter of the atmospheric layer, respectively. This hemispheric-mean approximation was used to obtain functional forms for these source functions, the upward azimuthally averaged intensities at the top \( (I^+_n(0, \mu)) \) and bottom \( (I^-_n(\tau, \mu)) \) of an atmospheric layer with optical depth \( \tau \) can be written as,

\[ I^+_n(0, \mu) = I^+_n(\tau, \mu) e^{-\tau/\mu} + \frac{G}{\lambda_\mu - 1} (e^{\lambda_\mu} e^{-\tau/\mu} - 1) \]
\[ + \frac{H}{\lambda_\mu + 1} (1 - e^{\lambda_\mu} e^{-\tau/\mu}) + \alpha_1 (1 - e^{-\tau/\mu}) \]
\[ + \alpha_2 (\mu - (\tau + \mu) e^{-\tau/\mu}). \]  

(10)

This calculation can be found in the get_thermal_1d_gfluxi module. Similarly the downward azimuthally averaged intensities at the top \( (I^-_n(0, -\mu)) \) and bottom \( (I^-_n(\tau, -\mu)) \) of the same atmospheric layer can be written as,

\[ I^-_n(\tau, -\mu) = I^-_n(0, -\mu) e^{-\tau/\mu} + \frac{K}{\lambda_\mu - 1} (e^{-\tau/\mu} - e^{-\lambda_\mu}) \]
\[ + \frac{J}{\lambda_\mu + 1} (e^{\lambda_\mu} - e^{-\tau/\mu}) + \sigma_1 (1 - e^{-\tau/\mu}) \]
\[ + \sigma_2 (\mu e^{-\tau/\mu} + \tau - \mu). \]  

(11)

and can also be found in the get_thermal_1d_gfluxi module. The functions \( G, H, K, I, \sigma_1, \sigma_2, \sigma_3 \), and \( \sigma_2 \) have been computed for the hemispheric-mean approximation in Toon et al. (1989) and are used in our calculation as well. Solving Equations (10) and (11) also require boundary conditions for the diffuse flux at the top and bottom of the atmosphere. These boundary conditions are set using the thermal blackbody intensities at the top and bottom of the atmosphere using,

\[ B_{\text{top}} = (1 - e^{-\tau/\mu}) B(T_{\text{top}}) \]
\[ B_{\text{bot}} = B(T_{\text{bot}}) + \mu_1 B(T_{\text{bot}} - 1) \]
\[ + \mu_2 B(T_{\text{bot}} - 2) \]  

(12)

where \( B(T) \) represents the blackbody function, \( T_{\text{top}} \) is the temperature of the topmost atmospheric layer, \( T_{\text{bot}} \) is the temperature of the bottom most atmospheric layer, and \( \tau \) is the optical depth of the bottom most atmospheric layer. \( \mu_1 \) is assumed to be 0.5 due to the hemispheric-mean approximation. Note that the bottom boundary condition is valid only for gas-giant atmospheres where the highest pressure grid point corresponds to the end of the user-defined grid and does not correspond to a “surface.” PICASO 3.0 does have the option to swap boundary conditions that are pertinent to the hard surfaces needed for terrestrial atmospheres, but it is not relevant for this work on gas giants, and as such, we do not discuss it here. \( \tau' \) is given by

\[ \tau' = \tau_{\text{top}} \frac{P_{\text{top}}}{(P_{\text{top+1}} - P_{\text{top}})} \]  

(13)

This expression, along with the top boundary condition in Equation (12), captures the downward thermal flux arising from the part of the atmosphere that has pressure less than the minimum pressure in the used atmospheric pressure grid. This formulation prevents artificial cooling of the topmost atmospheric layer.

Equations (10) and (11) can be used to calculate the incidence angle \( \mu \) dependent upward and downward \( I^+_n(\mu) \) and \( I^-_n(\mu) \) intensity field in each atmospheric layer. But in order to use these radiative intensities ultimately for the convergence criteria in Equation (6), they need to be integrated with Equation (8) to calculate the direction independent upward and downward fluxes\( F^+_\nu \) and \( F^-_\nu \). In order to compute these disk-averaged, layer fluxes, the upward and downward thermal intensity for each atmospheric layer is calculated using Equations (10) and (11) at five incident angles. The cosine of these incident angles \( \mu \) is determined using the Gaussian-quadrature method with five points (Abramowitz & Stegun 1970). The choice of five Gauss points is the default setting in the code, but other choices are also available in the get_angular_1d module. Table 1 shows the five default values of \( \mu \) and the corresponding Gauss weights used for this Gaussian-quadrature integration. The intensities at different incident angles are used

\(^5\)There is a typo in Equation (12) of Toon et al. (1989), which incorrectly swaps in \( S^+_\nu \) for \( S^-_\nu \) in the negative partial flux.
to compute the integral in Equation (8) with the Gaussian-quadrature integration technique using the weights in Table 1. The module get\_thermal\_1d\_gflux uses the formulation described above to calculate the wavelength-dependent upward and downward thermal fluxes at the edges of each atmospheric layer.

For the radiative transfer of the reflected stellar light, we use the quadrature approximation. The calculation is performed for a single incidence angle of the stellar radiation beam (60°, μ0 = 0.5) unlike the five Gauss point method of the thermal counterpart. Under the quadrature approximation, γ1 is 0.5/3(2 - ωo(1 + g)), γ2 is ωo√3(1 - g)/2, γ3 is (1 - √3 gμ0)/2, and γ4 is 1 - γ3. These functions can be found in the get\_reflected\_1d\_gflux module. The source functions Sγ and Sγ for this component are

\[
S^+ = \gamma_3 \pi F_0 \omega_0 e^{-(\tau + \tau_c)/\mu_0} \\
S^- = \gamma_4 \pi F_0 \omega_0 e^{-(\tau - \tau_c)/\mu_0}
\]

(14)

where τc is the cumulative optical depth of all of the atmospheric layers above the layer of calculation, and τ is the optical depth of the layer itself. F0 here represents the stellar flux incident on the top of the atmosphere. F is interpolated from the PHOENIX grid of models (Allard et al. 2012) available as a part of the PySynPhot (STScI Development Team 2013) package. Using these functions, Equation (9) can be solved for the upward and downward fluxes for the nth atmospheric layer, as has been shown in Toon et al. (1989),

\[
F_{\nu,n}^+ = k_{1n} e^{\lambda_{1n}} + \Gamma_{n}^{+} k_{2n} e^{-\lambda_{2n}} + C_{n}^{+} \\
F_{\nu,n}^- = \Gamma_{n}^{+} k_{1n} e^{\lambda_{1n}} + k_{2n} e^{-\lambda_{2n}} + C_{n}^{-}
\]

(15)

where the quantities Γn, k1n, k2n, λn, Cn+, and Cn− are defined in Toon et al. (1989; Equations 21–24 therein). Equation (15) is solved in the get\_reflected\_1d\_gflux module. However, an additional term needs to be added to the downward fluxes solved from Equation (15) in this formulation: this term is:

\[
F_{\nu,n}^- = F_{\nu,n}^+ + \mu_0 F_0 e^{-\tau_c}/\mu_0.
\]

(16)

Like the thermal component, complete solutions of Equation (15) also require boundary conditions on the diffuse flux at the top and bottom of the atmosphere. We use the boundary conditions

\[
F_{\text{top, st}} = 0 \\
F_{\text{bot, st}} = R_b \mu_0 F_0 e^{-\tau_{c, bot}/\mu_0}
\]

(17)

where F_{\text{top, st}} and F_{\text{bot, st}} are the diffuse flux at the top and bottom of the atmosphere, R_b denotes the reflectivity of the bottom surface, and τ_{c, bot} is the cumulative optical depth of the deepest layer. The reflectivity of the bottom surface is typically assumed to be 1% in the model, but users should check that the bottom surface is not optically thin. This is further discussed in Section 4. This formulation is used by the get\_reflected\_1d\_gflux module to calculate the wavelength-dependent upward and downward reflected stellar light fluxes at the edges of each atmospheric layer. But the calculation of both the thermal and reflected stellar light fluxes following the procedure described above requires another important parameter—the optical depth τ. The implemented τ calculation procedure in this model is described below.

Generally, two approaches are used for calculating the optical depth τ: (1) the line-by-line approach (e.g., Gandhi & Madhusudhan 2017; Burrows et al. 2008), or (2) the correlated-k approach (e.g., Marley & McKay 1999; Fortney et al. 2008; Malik et al. 2017). This model uses a correlated-k opacity approach for computationally efficient inclusion of gaseous opacities. In the correlated-k approach, each wavelength bin is represented by eight distinct Gaussian-quadrature points (gi), each with its associated weight (wi). Table 2 shows the Gauss points and the Gauss weights used for the eight-point correlated-k approach in the model. Each of these Gauss points also has an optical depth τi associated with it. These “Gauss” points should not be confused with the “Gauss” points used to calculate the disk-integrated fluxes. This means that the radiative transfer equations for each wavelength bin must be computed eight times, once for each value of optical depth τi corresponding to the ith Gauss point. This calculation is done by calling get\_thermal\_1d\_gfluxi and get\_reflected\_1d\_gfluxi modules eight times by the climate module. The eight thermal and radiative upward and downward fluxes are then added together following

\[
F_{\nu,n}^{\pm}(\lambda) = \sum_{i=1}^{8} w_i F_{i,n}^{\pm}(\lambda, \tau_i,n)
\]

(18)

where F_{i,n}^{\pm} is the thermal/reflected light flux at the nth atmospheric layer calculated with the optical depth τ_{i,n} corresponding to the ith Gauss point. Our calculation of

| Table 1 | Gauss Points and Weights Used for Gaussian-quadrature Integration of Thermal Flux Over Different Angles |
|---------|-----------------------------------------------------------------------------------------------------------------|
| θ(deg)  | µ = cos(θ)                                                                                                      | Gauss Weight |
| 84.345  | 0.09853                                                                                                           | 0.015747     |
| 72.270  | 0.30453                                                                                                           | 0.073908     |
| 55.804  | 0.56202                                                                                                           | 0.146386     |
| 36.680  | 0.80198                                                                                                           | 0.167174     |
| 16.221  | 0.96019                                                                                                           | 0.096781     |

| Table 2 | Gauss Points and Weights Used for Calculation of the Correlated-k Opacities |
|---------|------------------------------------------------------------------------------|
| Gauss Points (gi) | Gauss Weights (wi) |
| 0.065960251992824  | 0.165231051440291  |
| 0.313509004297193  | 0.309768948559709  |
| 0.636490995702807  | 0.309768948559709  |
| 0.884039748007175  | 0.165231051440291  |
| 0.953471592210149  | 0.00869637112843658 |
| 0.966500473910379  | 0.0163036288715637 |
| 0.983499526089621  | 0.0163036288715637 |
| 0.996528407789851  | 0.00869637112843658 |
correlated-k opacities is described in more details in Section 2.1.4. This summation with Equation (18) within the climate module ultimately produces thermal/reflected wavelength-dependent upward and downward fluxes in each atmospheric layer. These fluxes can now be used to calculate the convergence criteria in Equation (6). The net thermal flux for the nth layer can be simply obtained with

\[ F^n_{\text{thermal,net}} = \int_\lambda (F^n_{\lambda,\text{up}} - F^n_{\lambda,\text{down}}) d\lambda \]

\[ F^n_{\text{stellar,net}} = \int_\lambda (F^n_{n,\lambda,\text{st,up}} - F^n_{n,\lambda,\text{st,down}}) d\lambda \]

where \( F^n_{\lambda,\text{up}} \) represents the upward thermal flux from the nth layer within the wavelength bin between \( \lambda \) and \( \lambda + d\lambda \), and \( F^n_{\lambda,\text{down}} \) is the downward thermal flux from the same layer within the same wavelength bin. \( F^n_{n,\lambda,\text{st,up}} \) also represents the same quantity for the reflected light component. The total net radiative flux in the nth atmospheric layer can be calculated using

\[ F^n = r_{tb} F^n_{\text{thermal,net}} + r_{st} F^n_{\text{stellar,net}} \]

where \( r_{tb} \) is the contribution factor of the thermal radiation to the net flux and is generally fixed at 1, and \( r_{st} \) is the contribution factor of the stellar radiation to the net flux. Nonzero values of \( r_{st} \) are only relevant when the external irradiation on the atmosphere is nonzero. In the scenario when a user is computing a planet-wide average \( T(P) \) profile, the stellar irradiation is contributing to 50% (one hemisphere) of the planet and as a result \( r_{st} = 0.5 \). If instead the goal is to compute a night-side average atmospheric state, \( r_{st} \) is set to be 0. On the other extreme, to compute the dayside atmospheric state of a tidally locked planet, \( r_{st} \) should be set at 1.

This full net flux, \( F^n \), is the same as \( 4\pi \int_H dH d\nu \) in Equation (6). Therefore, \( F^n \) can be used to check if the convergence criteria (Equation (6)) is satisfied at all of the radiative layers of the atmosphere. This check is done in the “Converged?” box of the flow chart shown in Figure 1. Now that we have described the radiative transfer within our model, we will discuss other blocks of the model flow chart shown in Figure 1 starting with atmospheric chemistry.

2.1.3. Equilibrium Chemistry

As shown in Figure 1, in order to compute the radiative fluxes for checking the convergence criteria, as outlined in the previous Section 2.1.2, we first need a method to provide the chemical state of the atmosphere. This is required because the chemistry dictates the optical depth (\( \tau \)) required for the radiative transfer calculations. In the simplest case, we determine the chemical state of the atmosphere using the \( T(P) \) profile of the atmosphere, atmospheric metallicity, and C/O ratio assuming chemical equilibrium. The atmospheric metallicity (\( M/H \)) is defined as the ratio of the abundances of all heavy elements to hydrogen abundance in the atmosphere. Our climate model uses a precalculated grid of molecular abundances on a pressure versus temperature versus [\( M/H \)] versus C/O grid. This chemistry grid is computed using the thermochemical equilibrium models presented in Gordon & McBride (1994), Lodders (1999, 2002), and Visscher et al. (2006, 2010) and using protosolar elemental abundances from Lodders (2010). The grid includes 73 temperature points between 75 and 4000 K and 20 pressure points logarithmically spaced between \( 10^{-6} \) and 3000 bars. This corresponds to a precomputed grid with 1460 grid points. The molecular abundances included in this grid are \( H_2, H, H+, H_2^+, H_3^+, H_4^+, He, H_2O, CH_4, CO, CO_2, OCS, HCN, C_2H_2, C_2H_6, C_2H_4, \) NH₃, N₂, PH₃, H₂S, SiO, TiO, VO, Fe, FeH, MgH, CrH, Na, K, Rb, Cs, Li, LiOH, LiH, LiCl, and e-. The metallicities included in the grid are: [\( M/H \)] (relative to solar) = \(-1, -0.75, -0.5, -0.3, -0.25, 0.0, 0.5, 0.7, 1.0, 1.5, 1.7, \) and 2. The C/O ratios included in the grid are: C/O (relative to solar) = \(0.25, 0.5, 1.0, 1.5, 2, \) and 2.5. In this way, solar values are \([M/H] = 0 \) and C/O = 1. In order to change the C/O ratio for a given atmospheric metallicity, both the elemental abundances ratios, C/H and O/H, are increased/decreased slightly while maintaining a constant (C+O)/H. The version of PICOSS, described in this work, does not allow us to change other elemental ratios (e.g., S/H or N/H). Future releases will include this flexibility.

Figure 2 shows the precomputed volume mixing ratios of four major atmospheric gases, \( CH_4, CO, H_2O, \) and \( NH_3 \) for a solar mixture of chemical elements. Converged brown dwarf \( T(P) \) profiles with log(\( g \)) = 5 and multiple \( T_{\text{eff}} \) values between 300 and 2300 K are also overplotted in the pressure–temperature space to depict parts of the parameter space relevant for objects with different \( T_{\text{eff}} \). There are a few important features that strongly influence the climates of substellar atmospheres that are worth noting. First, there is a sharp drop in \( H_2O \) vapor abundance at \( T < 200 \) K, which can be seen in the lower-left panel Figure 2. This drop in \( H_2O \) vapor abundance is caused by the condensation of \( H_2O \) into cloud particles. A similar condensation effect can also be seen in the case of \( NH_3 \) at \( T < 100 \) K. Condensation-induced changes in the gaseous abundances are included in the precomputed grid of gaseous abundances. Second, the most abundant carbon-bearing gas in the atmosphere also changes from CO to \( CH_4 \) between temperatures of \( \sim 1000 \) and 1200 K. The top-right panel of Figure 2 shows that CO is the major carbon-bearing gas above \( \sim 1200 \) K, but at temperatures cooler than 1200 K, CO abundance decreases rapidly, and \( CH_4 \) abundance rises. This change can be understood by the net chemical reaction responsible for inter-conversion between CO and \( CH_4 \) under chemical equilibrium,

\[ CH_4 + H_2O \leftrightarrow CO + 3H_2. \] (21)

This reaction is favored in the forward direction for higher temperatures than \( \sim 1200 \) K, and as result, CO is more dominant for such temperatures. The reverse reaction dominates for temperatures lower than \( \sim 1200 \) K, and \( CH_4 \) becomes the dominant C-bearing gas there. Many other similar interesting trends have been explored in the literature for several gases (e.g., Lodders 2002; Moses et al. 2005; Visscher et al. 2006, 2010; Moses et al. 2011, 2013). These trends can be directly explored from the available chemical grid within the model, but we move on to another challenge of using a precomputed chemistry grid—a limited number of grid points.

Due to the finite number of grid points (1460 points), chemical abundances cannot be calculated exactly at any pressure–temperature point of interest. Therefore, each time the chemistry routine premix_atmosphere is called, the climate model uses an interpolation scheme to compute abundances at each pressure–temperature layer point. This interpolation is performed by the module chem_interp. The 2D interpolation scheme used relies on finding the four surrounding grid points.
namely $T_{\text{low}}(P_{\text{low}})$, $T_{\text{low}}(P_{\text{high}})$, $T_{\text{high}}(P_{\text{low}})$, and $T_{\text{high}}(P_{\text{high}})$. Then the abundances in these surrounding points represented by $x_{\text{m}i\text{low,low}}$, $x_{\text{m}i\text{low,high}}$, $x_{\text{m}i\text{high,low}}$, and $x_{\text{m}i\text{high,high}}$ are used to interpolate the abundance of each species in the $T(P)$ point. This interpolation is done using,

$$
\ln \xi_i(T, P) = (1 - t_{\text{int}})(1 - p_{\text{int}}) \ln \xi_{\text{m,low,low}}^i
+ t_{\text{int}}(1 - p_{\text{int}}) \ln \xi_{\text{m,high,low}}^i + t_{\text{int}}p_{\text{int}} \ln \xi_{\text{m,high,high}}^i
+ (1 - t_{\text{int}})p_{\text{int}} \ln \xi_{\text{m,low,high}}^i
$$

where $t_{\text{int}}$ and $p_{\text{int}}$ are given by

$$
t_{\text{int}} = \frac{1/T - 1/T_{\text{low}}}{1/T_{\text{high}} - 1/T_{\text{low}}}$$

$$
p_{\text{int}} = \frac{\ln P - \ln P_{\text{low}}}{\ln P_{\text{high}} - \ln P_{\text{low}}}$$

This interpolation can be found in the chem_interp module. In the case where the $T(P)$ moves outside the edges of this chemistry grid, a linear interpolation scheme is instead adopted within the range 50–5200 K. This allows for some flexibility in the iteration scheme. However, beyond these values, the model is not valid. Note that choices in interpolation routines cause discrepancies in resulting abundances. For example, only using two nearest neighbors causes instabilities in chemical profiles. The interpolation chosen here was tested in multiple chemical regimes to ensure stability. After computing the atmospheric chemistry, the next step in Figure 1 is the computation of the atmospheric opacities, which we describe next.

### 2.1.4. Premixed Opacities

The chemical structure of the atmosphere dictates the optical depth ($\tau(\lambda)$) of each atmospheric layer, which is necessary for solving the radiative transfer equations described in Section 2.1.2. This calculation of atmospheric optical depths occurs in the “Opacity” block of the model flow chart in Figure 1. As previously stated, we use the correlated-k approach to handle the molecular opacities in this model. An alternative approach is the line-by-line method, which is more precise but, comparatively speaking, more computationally expensive than the correlated-k approach at low resolution, and over large wavelength ranges. For completeness, we describe
Table 3

| Molecule | Reference |
|----------|-----------|
| CH$_4$   | Yurchenko et al. (2013); Yurchenko & Tennyson (2014) |
| CO       | Rothman et al. (2010); McCall et al. (2017); Li et al. (2015) |
| CO$_2$   | Huang et al. (2014) |
| CrH      | Burrows et al. (2002) |
| Fe       | Ryabchikova et al. (2015); O’Brian et al. (1991); Fuhr et al. (1988); Bard et al. (1991); Bard & Kock (1994) |
| FeH      | Dalick et al. (2003); Hargreaves et al. (2010) |
| H$_2$    | McCall et al. (2017) |
| H$_2$O   | Poliansky et al. (2018) |
| H$_2$S   | Azzam et al. (2016) |
| HCN      | Harris et al. (2006); Barber et al. (2014); Gordon et al. (2022) |
| LiCl     | Bittner & Bernath (2018) |
| LiF      | Bittner & Bernath (2018) |
| LiH      | Coppola et al. (2011) |
| MgH      | Yadin et al. (2012); Gharib-Nezhad et al. (2013) assembled in Gharib-Nezhad et al. (2021) |
| N$_2$    | Rothman et al. (2015) |
| NH$_3$   | Yurchenko et al. (2011); Wińiewski et al. (2016) |
| OCS      | McCall et al. (2017) |
| PH$_3$   | Sousa-Silva et al. (2014) |
| SiO      | Barton et al. (2013) |
| TiO      | McKemmish et al. (2019) assembled in Gharib-Nezhad et al. (2021) |
| VO       | McKemmish et al. (2016) assembled in Gharib-Nezhad et al. (2021) |
| Li,Na,K  | Ryabchikova et al. (2015); Allard et al. (2007a, 2007b, 2016, 2019) |
| Rb,Cs    | |

Note. Correlated-K opacities are available at Lupu et al. (2021) and detailed in Marley et al. (2021).

the general methodology of our approach here. However, the data were originally computed by Lupu et al. (2021), available on Zenodo, and detailed in Marley et al. (2021). We include a table of the main reference data used to compute the premixed opacities in Table 3.

In the correlated-k approach, the relevant wavelength range for the radiative transfer is first divided to 196 wavelength bins. These carefully chosen wavelength bins are shown in Figure 3 and chosen to reflect the approximate spectral energy distributions of planetary atmospheres. The molecular opacity $\kappa$ in any one of these wavelength bins has numerous individual molecular lines, as shown in the left panel of Figure 4. The cumulative distribution function (CDF) of the opacity can be defined as $G(\kappa_0) = N(\kappa \leq \kappa_0)$ where $N$ denotes the total number of instances when the condition $\kappa \leq \kappa_0$ is satisfied within the wavelength bin. The CDF $G(\kappa)$ of the opacity within each wavelength bin is computed. This function, $G(\kappa)$, is then inverted to obtain the “k-distribution.” An example k-distribution for the opacity window that is shown in the left panel of Figure 4 is shown in the right panel of Figure 4. After computing a k-distribution for each of the 196 wavelength bins, the eight Gauss points $(g_i)$ shown in Table 2 are used to represent each distribution. Generally, the slopes of k-distributions are shallow and slowly changing at lower Gauss point values $(G)$ between 0 and 0.9. Then, the slope steepens rapidly between $G = 0.9$–1, as can be seen in Figure 4. To capture the complex shape of the k-distribution with just eight Gauss points, a double-Gauss method is used for the integration. The first set of four Gauss points is shown in the right panel of Figure 4 with black points. These four Gauss points sparsely sample most of the k-distribution between $G = 0$–0.95, whereas the last four Gauss points, shown with red points in Figure 4, sample the small, but rapidly changing, part of the k-distribution between $G = 0.95$–1. The first and second set Gauss points and weights for this double-Gauss point method can be generated from sample points of the generally used Gauss–Legendre quadrature (Abramowitz & Stegun 1970) using,

$$
g_{i,1} = \left( g_i + 1 \right) / 2$$

$$
w_{i,1} = f w_i / 2$$

$$
g_{i,2} = f + (1 - f) \left( g_i + 1 \right) / 2$$

$$
w_{i,2} = (1 - f) w_i / 2$$

(25)

where $g_{i,1}$ and $w_{i,1}$ represent the first set of our Gauss points (shown with black points in Figure 4), $g_{i,2}$ and $w_{i,2}$ represent the second set (shown with red points in Figure 4), $g_i$ and $w_i$ are the sampling points for Gauss–Legendre quadrature of some order (four in our case) defined within the interval [-1,1], and $f$ is the adjustable parameter, which sets the division in the values of $G$ that will be sampled by the first and second set of Gauss points.

For example, if $f = 0.8$, then $g_{i,1}$ will sample values between 0 and 0.8 while the rest will be sampled by $g_{i,2}$. For our purpose, this $f$ is set at 0.95, which is a reasonable choice based on the typical shape of k-distributions as shown in Figure 4.

This double-Gauss method for calculating correlated-k-coefficients has a large impact on the time and computational efficiency of the code as it reduces the number of Gauss points required for radiative transfer calculations but still maintains sufficient accuracy required for the model. It has been shown that the radiative transfer with double-Gauss method with eight Gauss points is as accurate as a normal set of 20 Gauss points (Michael Line, private communication; detailed in Marley et al. 2021). The choice of the wavelength bins for the correlated-k opacities are also crucial for the model and are described next.

The 196 wavelength bins used span 0.2–227 $\mu$m to capture the general spectral energy distribution of planetary atmospheres. However, these wavelength bins are not all of equal wavelength width. These wavelength bins are shown in Figure 3 where three blackbody curves corresponding to temperatures of 300, 1000, and 2000 K are also shown for comparison. The bins are narrow between 0.5 and 10 $\mu$m to capture the large number of molecular rovibrational bands in this range. At the tail ends of the blackbody distributions, larger bins help boost computational speed and maintain sufficient precision in radiative transfer calculations required for the application cases of this model.

As our model uses a precomputed chemistry grid during iterations on the $T(P)$ profile, the k-coefficients are also precomputed on the same pressure–temperature–metallicity–C/O ratio grid. For interactively $T-P$ values, the
molecular opacity is interpolated using the same formalism as has been described for the gas abundances in Section 2.1.3. Figure 5 shows heat maps of precomputed Planck function and abundance-weighted molecular cross sections in the same pressure–temperature grid as Figure 2. For each \( P - T \) point in the grid, the abundance-weighted molecular cross sections are integrated over all wavelengths using the Planck function (corresponding to the temperature \( T \)) as the integrating kernel (see Equation (2) in Freedman et al. 2014). Each panel corresponds to cross sections at different Gauss points, which we use in our models. Planck mean cross sections at only the first four Gauss points are shown here in the four panels. Dashed black lines marked on the cross-sectional maps depict different converged \( T(P) \) profiles of brown dwarfs from Marley et al. (2021) with \( T_{\text{eff}} \) between 300 and 2300 K and \( \log(g) = 5 \). As the order of the Gauss points increases in Figure 5, the cross sections become higher as higher-order Gauss points trace higher-opacity parts of the \( k \)-distribution. Achieving atmospheric convergence in regions of the \( P - T \) space where the cross section changes rapidly with small changes in temperature or pressure can be difficult. Such an “opacity cliff” can be seen at \( T \) values between \( \sim 900 \) and \( 1700 \) K in Figure 5. This cliff appears in the Planck mean cross sections due to the overlap of the peak of the Planck function at 900–1700 K with large H\(_2\)O opacity bands between 1.5 and 3 \( \mu \)m. These “opacity cliffs” were also seen in Freedman et al. (2008, 2014) with Rosseland-mean opacities. We discuss the effect of these cliffs on model convergence in more detail in Section 4.3.

In addition to molecular opacities, collision induced absorption (CIA) of H\(_2\)-H\(_2\), H\(_2\)-H, H\(_2\)-He, H\(_2\)-N\(_2\), H\(_2\)-CH\(_4\), and continuum opacities such as H-bf, and H-ff are also accounted for. These are included separately from our correlated-\( k \) table. The CIA opacities are precalculated between temperatures of 75 and 7000 K with 1000 wavelength bins and then interpolated to the correct \( P - T \) combination with a spline. Note that even though the CIA temperatures are computed up to 7000 K, the temperature valid range of our model is still limited by the 1460 opacity grid.

We note that the precomputed opacity grid used with this model is known to be incomplete with regards to atomic and ionic opacities, which are particularly important at the high-temperature and low-pressure parts of the atmosphere (e.g., Hoeijmakers et al. 2018, 2020). This limits, to some extent, the current code’s ability to treat the “ultra-hot Jupiters” \( (T_{\text{eq}} \gtrsim 2200 \text{ K}) \). In a future update, we will add these opacities to the correlated-\( k \) tables.

### 2.1.5. Disequilibrium Chemistry

A significant addition to this model is the capability to treat vertical-mixing-induced disequilibrium chemistry self-consistently within the RCE framework. This is an optional part of the model, as shown in Figure 1. In 1D models, vertical mixing is often parameterized as a diffusive process, which is described by the eddy diffusion coefficient, \( K_{zz} \) (Allen et al. 1981). The mixing timescales of all relevant gases in each atmospheric layer are given by

\[
\tau_{\text{mix}} = \frac{H^2}{K_{zz}}
\]

where \( H \) is the local scale height of that atmospheric layer. All thermochemical reactions within the atmosphere like CO \( \leftrightarrow \) CH\(_4\) and NH\(_3\) \( \leftrightarrow \) N\(_2\) also proceed with a characteristic timescale, \( \tau_{\text{chem}} \). Zahnle & Marley (2014) parameterized the \( \tau_{\text{chem}} \) of several such gases using 1D chemical kinetics models. The parameterized \( \tau_{\text{chem}} \) values from Zahnle & Marley (2014) and Visscher et al. (2006) are given by

\[
\begin{align*}
\tau_{\text{CO,CH}_4\text{H}_2O} &= 3 	imes 10^{-6} \exp\left(\frac{42000K}{T}\right) \\
\tau_{\text{NH}_3\text{N}_2} &= 10^{-7} \exp\left(\frac{52000K}{T}\right) \\
\tau_{\text{CO}_2} &= 10^{-10} \sqrt{\frac{P_{\text{bar}}}{T}} \exp\left(\frac{38000K}{T}\right) \\
\tau_{\text{HCN}} &= 1.5 \times 10^{-4} \frac{P_{\text{bar}}^{0.7}}{[\text{OH}]} \exp\left(\frac{36000K}{T}\right) \\
\tau_{\text{PH}_3} &= 1.9 \times 10^{12} \exp\left(\frac{6013.6K}{T}\right)
\end{align*}
\]
where $P_{\text{bar}}$ is the atmospheric pressure in bars, $T$ is the temperature, and $m$ is the atmospheric metallicity relative to solar metallicity. $t_{\text{chem}}$ is generally short at high-pressure high-temperature regions of the atmosphere. As the atmosphere gets colder at lower pressures, the $t_{\text{chem}}$ rises exponentially and becomes large. The volume mixing ratios of CO, CH$_4$, NH$_3$, CO$_2$, H$_2$O, and HCN are expected to follow equilibrium chemistry at pressures where $t_{\text{mix}} \gg t_{\text{chem}}$, which happens in the deeper atmosphere. But as the atmosphere becomes colder with lowering pressure, $t_{\text{chem}}$ may exceed $t_{\text{mix}}$. The pressure at which this occurs is called the “quench pressure.” At pressures less than the “quench pressure,” gases are expected to depart from chemical equilibrium, and their mixing ratios become constant.

In this model, we include these disequilibrium chemistry effects for the eight gas species shown in Equation (27): H$_2$O, CH$_4$, CO, CO$_2$, NH$_3$, N$_2$, HCN, and PH$_3$. The quench pressure is determined using the parameterized $t_{\text{chem}}$ in Equation (27). The abundance then follows the equilibrium chemistry at pressures more than the quench level, and it is held constant at pressures less than the quench level.

The uncertain parameter in the disequilibrium modeling framework is the eddy diffusion parameter, $K_{zz}$ (Fortney et al. 2020; Phillips et al. 2020; Karalidi et al. 2021). Therefore, this model is flexible in assumptions regarding $K_{zz}$. Currently, there are two user-options for defining $K_{zz}$ within a model run: (1) a fully user-defined $K_{zz}$ value, which is either constant or variable throughout the height of the atmosphere, but that does not change during the iterative climate solution (as was done in Phillips et al. 2020 and Karalidi et al. 2021), and (2) a model-predicted $K_{zz}$, which is calculated from the $T(P)$ profiles in the convective zones, using mixing length theory (Giersch & Conrath 1985), and in the radiative zones, using parameterizations (e.g., Moses et al. 2021). In the second case, along with the $T(P)$ profile, the $K_{zz}$ will also change simultaneously with the iterations in the model. The $K_{zz}$ in the convective zone can be calculated with mixing length theory and is given by Giersch & Conrath (1985),

$$K_{zz} = \frac{H}{3} \left( \frac{L}{H} \right)^{4/3} \left( \frac{\mu R F}{\mu \rho_0 c_p} \right)^{1/3}$$

(28)

where $H$ is the local scale height of the atmosphere, $L$ is the turbulent mixing length, $R$ is the universal gas constant, $\mu$ is the mean molecular weight of the atmosphere, $\rho_0$ is the atmospheric density, $c_p$ is the atmospheric specific heat at constant pressure, and $F$ is the convective heat flux. The convective heat flux can be calculated by the difference between the net thermal radiative flux and $\sigma T_{\text{eff}}^4$ within the convective zones of the atmosphere. Therefore, $\sigma T_{\text{eff}}^4$ is the maximum allowed value of $F$ in this framework if the energy transport within the convective atmosphere is assumed to be completely convective.

While various parameterizations of $K_{zz}$ in the radiative zones of substellar atmospheres have been discussed in the literature (Parmentier et al. 2013; Wang et al. 2015; Zhang & Showman 2018; Tan 2022), we have included the parameterization from Moses et al. (2021) as a starting point. Further parameterizations of $K_{zz}$ can easily be swapped in, in the future.

The Moses et al. (2021) radiative zone $K_{zz}$ is given by

$$K_{zz} = \frac{5 \times 10^8}{\sqrt{P_{\text{bar}}}} \left( \frac{H}{620 \text{km}} \right) \left( \frac{T_{\text{eff}}}{1450 \text{K}} \right)^4$$

(29)

where $P_{\text{bar}}$ is the pressure of the radiative level in bars, and $H$ is the atmospheric scale height. Both Equations (28) and (29) can be found in the get_kzz function. As the $T(P)$ profile iterates toward the converged solution, the quench levels of various gases change as well. This means that the abundances of quenched species will depart from chemical equilibrium, and the precomputed k-coefficient tables described in Section 2.1.4 are no longer valid. The methodology to remix the k-coefficients with updated abundances and recompute resulting optical depth calculation during the convergence process, is referred to as “on-the-fly” mixing, and is described in Section 2.1.6.
2.1.6. Mixing Opacities “On-the-fly”

With disequilibrium chemistry, the atmospheric chemistry depends on the quench pressures, which again depend on the \( T(P) \) profile of the atmosphere. Therefore, because the chemistry of the atmosphere cannot be predetermined, the atmospheric opacities also need to be calculated “on-the-fly.”

We mix the correlated-k opacities of individual gases using the methodology of Amundsen et al. (2017) called the resort-rebin technique. Currently, in this model, we focus on the quenching of CO, CH\(_4\), H\(_2\)O, NH\(_3\), CO\(_2\), N\(_2\), HCN, and PH\(_3\). However, the major opacity sources among these gases are mainly CO, CH\(_4\), H\(_2\)O, and NH\(_3\). Meaning, the contribution of N\(_2\), HCN, and PH\(_3\) on the total gas opacity is negligible for the relatively small departures from the chemical equilibrium with \( \log(M/H) \leq 2.5 \) explored in this analysis. Therefore, we mix the correlated-k opacities of CO, CH\(_4\), H\(_2\)O, and NH\(_3\) with the correlated-k opacities of all of the other sets of background gases that follow equilibrium chemistry as the volume mixing ratio of these gases evolves due to quenching with the \( T(P) \) profile of the atmosphere. Should the motivation to include more gases in the “on-the-fly” methodology present itself in future observations, these gases can be included in a future code release.

One of the drawbacks of “on-the-fly mixing” outlined in Amundsen et al. (2017) is the dependence of the accuracy of the technique on the spectral resolution of the correlated-k opacities. This effect has been well explored in Karalidi et al. (2021). Specifically, it was found that the 196 wavelength bins, traditionally used in the chemical equilibrium version of EGP and this new Python version, are not sufficiently high to use with the resort-rebin technique. Karalidi et al. (2021) found that 661 wavelength bins in the correlated-k opacities are required to counteract the inaccuracies in the resort-rebin technique. Therefore, when disequilibrium calculations are requested by the user, PICASO automatically switches to 661 wavelength bins.

With the opacities and mixing routines described, the final module, shown in Figure 1, is the computation of the convective zones, which we outline in Section 2.1.7.

---

Figure 5. The top-left panel shows the heat map of the Planck-weighted molecular opacity in the pressure–temperature k-coefficient grid used in the model with equilibrium chemistry described in the paper. Dashed black lines marked on the cross-sectional map depict different converged \( T(P) \) profiles of brown dwarfs from (Marley et al. 2021) with \( T_{\text{eff}} \) between 500 K and 2300 K and \( \log(g) = 5 \). This grid is used for interpolation of opacities during iteration of the atmospheric state in the code. The same for the 3rd, 5th, and 7th Gauss points are also shown in the top-right, bottom-left, and bottom-right panels.
Figure 6. A heat map of the adiabatic lapse rate as a function of pressure and temperature is shown. This grid is used to determine the lapse rate in atmospheric layers unstable against convection.

2.1.7. Convective Zones

As the $T(P)$ profile iterates toward radiative equilibrium, parts of the atmosphere will become unstable against convection, and energy transport will be expected to occur via convection instead of radiation. These parts of the atmosphere are forced to follow the local adiabat. Figure 6 shows a heat map of the adiabatic lapse rate, $\frac{d\ln T}{d\ln P}$, used in PICASO. This grid is precomputed assuming a solar H–He mixture with He mass fraction of $Y = 0.28$. This includes $H_2 \rightarrow 2H$ dissociation and a detailed accounting of the molecular vibrational and rotational levels.\(^5\) The calculations for this grid were done by Didier Saumon as described in Marley et al. (2021). The precomputed grid has 53 temperature points between 10 and 3981 K and 26 pressure points between $10^{-2}$ and 10^3 bars. The lapse rate of parts of the profile, which become unstable against convection, is interpolated from this grid. For every atmospheric layer, the lapse rate, $\frac{d\ln T}{d\ln P}$, is first calculated from the $T(P)$ profile. The grid of adiabatic lapse rates shown in Figure 6 is then used to interpolate the local adiabatic lapse rate for each atmospheric layer. This interpolation is done using 2D interpolation, similar to the technique described in Section 2.1.3.

If the ratio between the lapse rate obtained from the $T(P)$ profile and the interpolated lapse rate, $\nabla$, is greater than $\sim 1$ (numerically set to 0.98) in any of the layers, then these layers are considered convective and are forced to follow the interpolated local adiabatic lapse rate according to the following equation:

$$T_i^{i+1} = \exp\left(T_i^{i+1} + \nabla \ln \left( \frac{P_i^{i+1}}{P_i^i} \right) \right)$$  \hspace{1cm} (30)

where $T^i$ and $P^i$ are the temperature and pressure of the $i$th convective layer, and $T^{i+1}$ and $P^{i+1}$ are the temperature and pressure of the $(i+1)$th convective layer starting from below the radiative-convective boundary layer. We should note that this convective adjustment is done for one layer at a time followed by an iteration of the entire $T(P)$ as this adjustment for each layer, in principle, would lead to temperature changes in all of the other layers. This approach used in this model is different from forcing all of the layers, which are unstable against convection, to be convective at once. The $\text{t\_start}$ module implements this convective adjustment of the $T(P)$ profile. This methodology is used to develop and grow convective zones in the atmosphere with the $\text{find\_strat}$ module during the iterations of the model. Note that this module is designed to always form or grow convective zones and not shrink or remove them. This is because once a particular atmospheric layer is made convective, it is no longer iterated to achieve radiative equilibrium, as it is a convective layer. So once a layer is found to be convective, it just responds and changes according to the iterations of all of the other radiative layers. We have validated this method by comparing lapse rates with other models (e.g., Goyal et al. 2020; Phillips et al. 2020) for several brown dwarfs or giant planets with single or multiple convective zones. We discuss the important implications of this for the user in Section 4.

2.2. Iteration Scheme and Model Convergence

Each required and optional physical component of the model shown in Figure 1 has been described separately in Section 2.1. But as our model is iterative in nature, a description of the iterative scheme of the model along with its convergence criteria is a necessary component. We provide a brief outline of the model iterative scheme for both the chemical equilibrium and disequilibrium runs separately below.

2.2.1. Iterative Scheme for Chemical Equilibrium

This model aims to achieve RCE using the Newton-Rhapson iterative scheme. As a first step, the chemistry and the correlated-k opacities of the atmosphere for the first given guess $T(P)$ profile are calculated as detailed in Sections 2.1.3 and 2.1.4. Then, the methodology described in Section 2.1.2 is used to compute the net radiative fluxes in all atmospheric layers. The temperature of each layer is then perturbed by a small arbitrary $dT$, historically set at 0.01% of the current layer temperature, while keeping the other layer temperatures fixed. The radiative fluxes in all layers are then recomputed after this perturbation. These two sets of radiative fluxes are used to compute the Jacobian $A_{ij}$, which quantifies the change in radiative flux in layer $i$ due to perturbed temperature in layer $j$. The Jacobian is given by (e.g., Hubeny 2017)

$$A_{ij} = \frac{F_{ij}^{\delta T} - F_{ij}}{\delta T_j}$$  \hspace{1cm} (31)

where $F_{ij}$ is the net radiative flux in layer $i$ before the perturbation, and $F_{ij}^{\delta T}$ is the perturbed net radiative flux in layer $i$ due to a change in temperature $\delta T_j$ in layer $j$. The ultimate temperature correction, $\delta T$, needed to ensure radiative

\(^5\) The raw grid can be found in GitHub file specific_heat_p_adiabat_grad.json.
equilibrium (in the radiative zones) is then solved using
\[ A \delta T = \sigma T^4_a - F(T(P)) \]
(32)
where \( A \) is the Jacobian, and \( F(T(P)) \) are the net radiative fluxes in each layer with the current atmospheric state. This process continues iteratively until a tolerance of maximum allowed radiative flux difference from \( \sigma T^4_a \) is reached in all of the radiative layers of the atmosphere. This convergence criteria can be expressed as
\[ \left| \frac{F_n^a - \sigma T^4_a}{\sigma T^4_a} \right| \leq \epsilon \]
(33)
where \( \epsilon \) denotes the tolerance parameter, set at \( 1 \times 10^{-3} \) in our model. This equation is a numerical version of Equation (6). The model is also considered to be converged if the maximum fractional temperature correction among all of the radiative layers \( |\delta T/T| \), calculated from Equation (32), during any iteration is smaller than \( \epsilon \). It is important to note that the model satisfies the convergence criteria in Equation (33) in the mid-point of each atmospheric layer while the temperature at the edges of each atmospheric layer (levels) is iterated. For example, if the optical depths at the bottom and top (edges) of the \( i \)th atmospheric layer are \( \tau_i \) and \( \tau_{i-1} \), then the model aims to satisfy Equation (33) in this layer for an optical depth of \( (\tau_i + \tau_{i-1})/2 \). But, the iteration of the temperature profile is done to the temperature at the bottom and top (or edges) of this atmospheric layer. This is important for the stability of the iterative process.

As the iteration progresses, the stability of each atmospheric layer against convective mixing is also checked using the technique described in Section 2.1.7. If layers unstable against convection are found, the temperature of these convectively unstable layers is forced to a H2-He gas mixture adiabat from the grid shown in Figure 6. Once the convergence criteria of Equation (33) is met, the model run stops and it produces the outputs outlined by the green boxes in Figure 1. Outputs such as the converged \( T(P) \) profile, chemical abundances, and outgoing radiative fluxes in the 196 wavelength grid are always returned to the user after model converges. These outputs can further be optionally used for calculating various observables such as transmission, emission, or reflected light spectra with PICASO.

For a deeper understanding of the iteration scheme of this model, three atmospheric iteration steps for a brown dwarf with \( T_{\text{eff}} \) of 1000 K and \( \log(g) = 5 \) are shown in Figure 7, and iterations for an irradiated planet at a distance of 0.1 au from a Sun-like star with \( T_{\text{int}} \) of 300 K and \( \log(g) = 3.4 \) are shown in Figure 8. In Figures 7 and 8, each row corresponds to the atmospheric state in a certain iteration of the model. The first column shows the \( T(P) \) profile of the atmosphere in each iterative step of this model with the red solid line, whereas the \textit{SONORA} \( T(P) \) profile for this case is shown as a black dashed line. The second column shows the lapse rate of the \( T(P) \) profile as a function of pressure in each iteration with the solid blue line, and the adiabatic lapse rate is shown with a black solid line. The third column shows the volume mixing ratio profiles of four gases, H2O, CH4, NH3, and CO, as a function of pressure, and the fourth column shows the emergent thermal spectrum from the top of the atmosphere. In Figure 7, the first row shows a simple initial isothermal guess profile of 500 K, and the last row shows the final converged solution. For our initial guess, the bottom four layers of the atmosphere have been assumed to be convective. The initial thermal emission spectrum resembles a blackbody because the atmosphere is isothermal. As the model iterates, the isothermal \( T(P) \) profile is perturbed to reach an atmospheric state such that Equation (33) is satisfied for all radiative layers of the atmosphere. In the iteration shown in the middle row, the bottom of the atmosphere becomes unstable against convection, and therefore in the final solution, the lapse rate in that part of the atmosphere is forced to follow the local adiabatic lapse rate. As the model iterates toward the converged solution, the \( T(P) \) profile becomes more complex and leads to a redistribution of flux in each wavelength bin. This leads to molecular absorption lines becoming more evident in the thermal spectrum with each iteration. The final converged solution matches well with the \textit{SONORA BOBCAT} grid of models (shown as black dashed line). For the irradiated planet convergence shown in Figure 8, the initial guess is an isothermal profile with \( T = 700 \) K, and the iterations continue until Equation (33) is satisfied.

A second helpful method of visualizing a converged run is to look at the contribution of radiative and convective fluxes as compared to the target flux in brown dwarfs given by \( \sigma T_{\text{eff}}^4 \) or the internal heat flux for planets given by \( \sigma T_{\text{int}}^4 \). Figure 9 shows these layer-by-layer radiative and convective fluxes for converged models of a representative brown dwarf (left panel), warm Jupiter (middle panel), and hot Jupiter (right panel). The net radiative flux in each layer is shown with the orange shaded region. The shaded blue region depicts the convective flux at each layer. The red dashed line shows the target flux for the brown dwarf, \( \sigma T_{\text{eff}}^4 \). This converged brown dwarf model has two convective and radiative zones, as can be seen from its \( T(P) \) profile shown in black. As a result, the convective flux (blue) peaks at the location of these convective zones. The net radiative flux decays rapidly at the deeper convective zone, and convection carries the majority of the energy in these deep convective layers. The proof that this is a converged model and Equation (33) is satisfied lies in the fact that the sum of the net radiative and convective fluxes is equal to the target flux at all atmospheric layers. For the planet case where there is an additional energy flux from the host star, we show the additional energy with black hatched shading (second and third columns of Figure 9). This incident flux is downward, compared to the thermal radiative and convective fluxes, which is upward. Therefore, the upper atmosphere needs larger quantities of upward thermal radiative fluxes to balance this downward stellar flux. Ultimately, the goal is to maintain a summed flux of \( \sigma T_{\text{int}}^4 \) (red dashed line) at all of the atmospheric layers of the planet. Therefore, the upper atmosphere of irradiated planets, with a given \( T_{\text{int}} \), is pushed toward hotter temperatures when compared to a brown dwarf with a comparable \( T_{\text{eff}} \).

### 2.2.2. Iteration Scheme for Chemical Disequilibrium

The primary difference in the iterative scheme of the disequilibrium model is that it uses a two-step convergence method. With the user-specified input parameters, the code first uses the basic chemical equilibrium model (Section 2.2.1) to converge to a chemical equilibrium atmospheric solution. This atmospheric solution is then used as the initial guess to the disequilibrium chemistry model in the second step of the iterative process. Of particular importance is the need
to recalculate the convective zones when disequilibrium chemistry is turned on. This is important because recent work has shown that there are major differences in the location, extent, and number of convective zones between the equilibrium chemistry and disequilibrium chemistry atmospheric solutions (Mukherjee et al. 2022). The general procedure is to partially reset the convective zones before moving to disequilibrium solver. First, any upper convective zones are removed. Next, the upper boundary of the deepest convective zone is reset to slightly higher pressures (usually ~5 levels in a 91-level atmosphere from $10^{-4}$ to 200 bars). This enables the convective zones in the presence of vertical mixing to be recalculated, because we cannot assume there will be minor perturbations from the equilibrium chemistry solutions.

In order to highlight the convergence of a brown dwarf disequilibrium chemistry model, a few atmospheric iterations are shown in Figure 10 for a brown dwarf with $T_{\text{eff}}$ of 700 K and log(g) of 5. The figure is similar to Figure 7 except here the third column now shows the mixing (black dashed line) and chemical timescales (solid colored lines) as a function of pressure, and the fourth column shows the volume mixing ratio profiles of various gases. The first row shows the atmospheric state reached after the first step of the convergence process, where a converged

Figure 7. Three steps in the iteration toward a converged solution for a brown dwarf with $T_{\text{eff}}$ of 1000 K and log(g) of 5 are shown here in each row. The first column shows the iteration of the $T(P)$ profile in red, with the SONORA BOBCAT model shown in black dashed. The second columns show the lapse rate from the $T(P)$ profiles in blue and the local adiabatic lapse rate in black. The third column shows the iterations in the volume mixing ratios of various gases. The fourth column shows the thermal spectrum emergent from the atmosphere of the object in each iteration. The animated version of this figure shows the evolution of the $T(P)$ profile in the top-left panel, lapse-rate profiles in the top-right panel, chemical abundances in the bottom-left panel, and the emergent spectrum in the bottom-right panel. The animation runs for 13 s.

(An animation of this figure is available.)
chemical equilibrium solution is achieved. The last row shows the final converged solution with disequilibrium chemistry. The chemical equilibrium solution from the SONORA grid is shown with the black dashed line in the first column. As the model iterates, chemical equilibrium $T(P)$ profile is perturbed to reach an atmospheric state where the $T(P)$ profile is visibly colder than the chemical equilibrium solution. This is a result of different atmospheric optical depths as a result of quenching of several gases in the deeper atmosphere. The chemical equilibrium solution has a single deep convective zone for this brown dwarf. However, in the iteration shown in the middle row and last row, a second convective zone also develops when disequilibrium chemistry is treated self-consistently. This shows that chemical disequilibrium also impacts the location and number of convective zones in the atmosphere. As a second convective zone develops in the middle and third rows, a decrease in mixing timescale can also be seen in that pressure range of the convective zone because convective zones are assumed to be much more efficient in mixing and thus have higher $K_z$ and lower $t_{mix}$ than radiative zones. Comparing the fourth column in the first and last rows shows that mixing causes a several-orders-of-magnitude change in the abundance of CO in the upper atmosphere when compared to chemical equilibrium solutions. With these examples, we conclude our detailed discussion on the methodology of the model and move on to the model benchmarking analysis.
3. Benchmarking Analysis

We have benchmarked this model in four ways in order to ensure all of the described methodology is functioning as expected. In order to benchmark the version of the model with chemical equilibrium, we compare PICASO 3.0 with atmospheric models from two different grids—the SONORA BOBCAT grid from Marley et al. (2021) and the ATMO 2020 chemical equilibrium model grid from Phillips et al. (2020). Figure 11 shows the comparison between the three models. The models described in this paper are shown in red, the BOBCAT models are shown with a black dashed line, and the blue dotted-dashed line shows the models from Phillips et al. (2020). The left columns show a comparison between the T(P) profiles of the three models. The right columns show a comparison between the lapse rate of the profiles. Included in the benchmarking is four different log(T) = 5 brown dwarf models with Teff values of 600, 700, 800, and 1000 K, which covers significant chemical transitions of carbon species. PICASO 3.0 and the models from Marley et al. (2021) match well with T(P) profile differences smaller than 0.1%. The lapse rates from this model also match excellently with those from Marley et al. (2021). Given PIAO’s heritage in the EGP model used to compute the grid in Marley et al. (2021), this is validation that the update to Python from Fortran did not introduce numerical issues. The disagreements between PICASO 3.0 and the chemical equilibrium models from the ATMO 2020 grid are typically smaller than 1% in the deeper atmosphere below 10 mbars. At pressures less than 10 mbars, the differences between the two models are between 1% and 5%. These differences are considered minor given the independent code setup, opacity calculations, and chemistry routines. For example, models presented in Phillips et al. (2020) use the “on-the-fly” opacity mixing method described in Section 2.1.6 even when modeling atmospheres with chemical equilibrium, whereas PICASO 3.0 uses interpolations of premixed opacities for iterations of chemical equilibrium models. Differences such as these along with differences in opacity line lists used by the two models could account for the 1%–5% differences in T(P) profiles in the upper atmosphere.

We also benchmark the equilibrium chemistry version of the code by comparing models of irradiated exoplanets calculated using PICASO 3.0 with the published ATMO models presented in Goyal et al. (2020). We use the RCTE models for WASP-25 b (Enoch et al. 2011; Southworth et al. 2014) presented in Goyal et al. (2020) to perform this benchmarking. WASP-25 b is a hot Saturn with an estimated equilibrium temperature of ~1210 K (assuming 0 albedo; Enoch et al. 2011; Southworth et al. 2014). We use the same system parameters for WASP-25 b as are used in the models presented in Goyal et al. (2020) and detailed in the appendix section of their article. Figure 12 shows the comparison between the two models at subsolar and solar metallicities with a supersolar C/O ratio. The models generally agree for both the metallicities. For the subsolar metallicity models, the agreement is better than 1% across all of the pressures. For the solar metallicity models, the maximum disagreement between the models is ~3%. These minor differences were also found between the PICASO 3.0 and ATMO 2020 models for brown dwarfs. As described above, these minor differences can be attributed to different opacities and numerical methodologies used.

In order to benchmark the version of the model with disequilibrium chemistry, we have done two types of comparisons. First, we set Kzz = 0 in our disequilibrium chemistry model and compare to the results of our equilibrium chemistry model from the SONORA grid. Second, we benchmark our disequilibrium chemistry model with results from the ATMO 2020 grid. In the first test, if Kzz is assumed to be zero, tmax becomes infinitely large according to Equation (26). Therefore, in principle, none of the gases will be quenched, and their volume mixing ratios will follow equilibrium chemistry throughout the atmosphere. Therefore, a disequilibrium chemistry model run with Kzz = 0 must produce the same result as the equilibrium chemistry models. Of particular importance, this test checks if
our “on-the-fly” mixing routines, and double-iterative routine, are performing as expected. Figure 13 shows the benchmarking between our disequilibrium chemistry model with $K_{zz} = 0$ and the equilibrium chemistry models from the SONORA BOBCAT grid. The left column shows the comparison of the $T(P)$ profiles between the two models, and the right column shows the comparison of the lapse rates between the two models. Each row from the top to bottom shows log(g) = 5.0 brown dwarf models with $T_{\text{eff}}$ of 600, 700, 800, and 1000 K. The two models match with differences $\leq 1\%$ level. The small $\leq 1\%$ deviations are due to the slight inaccuracies in the “on-the-fly” mixing technique (Amundsen et al. 2017), which is sensitive to the number of wavenumber bins, versus the preweighted technique (see Section 2.1.6).

In the second test, we directly benchmark our disequilibrium chemistry model with disequilibrium chemistry models from Phillips et al. (2020). Disequilibrium chemistry models in Phillips et al. (2020) assume a constant-with-pressure $K_{zz}$ for the atmospheres. Therefore, we use a constant $K_{zz}$ in our benchmarking test and show the results in Figure 14. For a brown dwarf with $T_{\text{eff}}$ of 700 K and log(g) of 5.5, the top-left panel in Figure 14 shows: (1) the differences that arise in the $T(P)$ profile when disequilibrium is turned on for both the models from Phillips et al. (2020) and PICASO 3.0, and (2) the differences that arise between Phillips et al. (2020) and PICASO 3.0. Both the equilibrium chemistry and disequilibrium chemistry models show excellent matches among themselves, which is an excellent benchmarking demonstration.

Figure 10. Three steps in the iteration toward a converged solution with disequilibrium chemistry for a brown dwarf with $T_{\text{eff}}$ of 700 K and log(g) of 5.25 is shown here in each row. The first column shows the iteration of the $T(P)$ profile with the red line, and the SONORA BOBCAT chemical equilibrium model is shown by the black dashed. The second column shows the lapse rate from the $T(P)$ profiles in blue and the local adiabatic lapse rate in black. The third column shows the chemical timescales of various gases as a function of pressure with solid lines of different colors. The pressure-dependent mixing timescale is shown in the third column with the black dashed line. The fourth column shows the iterations of the volume mixing ratios of various gases. The animated version of this figure shows the evolution of the $T(P)$ profile in the top-left panel, lapse-rate profiles in the top-right panel, chemical abundances in the bottom-left panel, and the mixing and chemical timescale profiles in the bottom-right panel. The animation runs for 9 s.

(An animation of this figure is available.)
of our methods and codes. Additionally, this also shows the impact of disequilibrium chemistry on the $T(P)$ profiles of this representative brown dwarfs, which results in $\sim 50$ K colder $T(P)$ profiles compared to equilibrium chemistry models.

The rest of the panels in Figure 14 show comparisons between various disequilibrium chemistry $T(P)$ profiles with different $K_{zz}$ and gravity values produced with our model with the grid of models from Phillips et al. (2020). Our
disequilibrium models generally agree well within 5% levels in the deeper atmosphere with models from Phillips et al. (2020). However, the disagreements are higher in the much lower-pressure upper atmosphere, which was also the case for the equilibrium chemistry model comparisons. The small disagreements between the models are mainly due to two reasons: (1) Phillips et al. (2020) use chemical kinetics models for calculation of quenched abundances whereas we use the “quench-time” approximation (Section 2.1.5) to do so, and (2) differences between the accuracy of the “on-the-fly” mixing method used between the two models due to differences in number of wavelength bins used. Phillips et al. (2020) used 32 wavelength bins between 0.2 and 322 μm (shown in Table 4 of Amundsen et al. 2014) to iterate their T(P) profiles, whereas we use 661 wavelength bins in a similar wavelength range for models, which include disequilibrium chemistry. These disagreements show the uncertainty in state-of-the-art models, which perhaps can be explained in the future with comparisons with high signal-to-noise data.

4. Modeling Recommendations

The code is generally well behaved across the parameter space of interest for brown dwarfs and exoplanets. However, iterative schemes are sometimes notoriously tricky to converge. These difficulties can arise due to a variety of reasons like large changes in atmospheric chemistry or opacities resulting from a small iterative step in the T(P) profile. Iterative steps causing the models to step out of the domains of tabulated opacities or chemistry also can create convergence issues. Therefore, here we outline recommendations regarding the use of code such that users can get meaningful and accurate results. These recommendations are implemented in the publicly available code tutorials available via the PICASO documentation page.7

4.1. Choosing Model Pressure Grid

We recommend using typically 50–90 atmospheric pressure layers, corresponding to 51–91 pressure levels. In general, a higher number of pressure layers increases the computational time required for convergence substantially, and a lower number of layers makes the atmospheric grid too coarse for an accurate calculation. The maximum and minimum values of the pressure grid are also ultimately chosen by the user. While the choice of the minimum pressure can be made somewhat arbitrarily, we recommend that the model not be run at pressures lower than 10⁻⁶. This is the lowest pressure for which chemistry and opacities are computed (shown in Figures 2 and 5). Another uncertainty from the double-Gauss correlated-k approach can arise from running the model at very low pressures (≤10⁻⁴ bars). As the molecular lines become very narrow at such low pressures, most of the opacities reside at very high values of G(κ) (e.g., between 0.995 and 1.0) in the k-distribution. Therefore, these opacities are not taken into account even with the double-Gauss method. This can make the radiative transfer at such low pressures inaccurate.

The maximum pressure of the atmospheric model needs to be chosen carefully in the case of giant planet atmospheres and brown dwarfs. If the maximum pressure of the atmospheric model is too low, then the model can become transparent to the deepest atmospheric layer, especially in wavelengths with low gaseous opacities (e.g., optical wavelengths ≤1 μm). This would result in an inaccurate T(P) profile. A good way to check if the profile was run with low maximum pressure is to plot the wavelength-dependent brightness temperature of the converged model along with the converged temperature of the deepest atmospheric layer. If the brightness temperature at any wavelength matches the brightness temperature associated with the lower boundary condition defined in Equation (12), then the pressure grid is not well suited for the calculation. However, if the brightness temperature is smaller than this bottom temperature at all wavelengths, then the pressure grid is well suited for the calculation. In the first scenario, the pressure grid needs to be extended to higher pressures. In case of model runs with atmospheric thermal inversion, the value of the wavelength-dependent optical depth can also be inspected to make sure that the bottom of the atmospheric model is optically thick throughout the wavelength range. The problem with this approach to choose a pressure grid is that it can only be done after running the model once. A more practical

Figure 12. T(P) profile computed using PICASO 3.0 for WASP-25 b is shown with the red solid line compared with the ATMO models presented in Goyal et al. (2020) with dashed black lines. The left panel shows the comparison for subsolar metallicity of 0.1 × solar whereas the right panel shows comparisons between solar metallicity models. The same system parameters were used to compute both the models, and they agree with each other with less than 3% differences.
alternative to this approach is to find a comparable model from the **SONORA BOBCAT** (Marley et al. 2021)\(^8\) to the temperature–gravity parameter space wanted and use the maximum pressure corresponding to that model. Also, the highest pressure for which chemistry and opacities are computed (shown in Figures 2 and 5) is 3000 bars. Therefore, runs with pressure grids extending to higher pressures than this will be inaccurate, as they would require linear extrapolation of both gas abundances and opacities.

---

\(^8\) A direct link to the data set on Zenodo is [https://doi.org/10.5281/zenodo.5063476](https://doi.org/10.5281/zenodo.5063476).
When computing disequilibrium chemistry runs, the user must take another aspect into consideration while choosing the maximum pressure of the atmospheric pressure grid: the quench levels of various gases. If the maximum pressure of the atmospheric grid is too low such that gases are expected to quench at deeper parts of the atmosphere than the pressure grid, then the model run will be incorrect. Therefore, users must check the quench levels of various gases after performing a disequilibrium chemistry calculation with the model. If the quench levels of any of the gases are at the last pressure level, then the user needs to increase the maximum pressure of the run to get a correct converged solution.

4.2. Choosing the Initial Guess $T(P)$ Profile

Climate models that leverage an iterative scheme require a first guess of the $T(P)$ profile. This guess $T(P)$ profile is then iterated to reach the converged solution. Even though we used isothermal $T(P)$ profiles as our first guess in both Figures 7 and 8, it is usually preferable to start with an initial guess that is close to the expected solution. Using simple profiles like isothermal profiles can lead to a significant increase in run times and, in the worst cases, also lead to solutions that do not converge. A far better alternative to isothermal profiles is publicly available model grids like SONORA BOBCAT or the
Phillips et al. (2020) models. For exoplanets, we recommend using parameterized $T(P)$ profiles from, e.g., Guillot (2010), as a first guess. Functionalities to browse these profiles directly are already available in PICASO, via the guillot.pt() and sonora() functions.9

Another first guess required to run the model is the extent of the deepest convective zone. Specifically, the user needs to specify a guess of the pressure level of the radiative-convective boundary in the atmosphere that the code will then modify using methods described in Section 2.1.7. Even though this guess is modified within the iteration, we recommend always setting the bottom three to five levels to be convective first. If the user sets too many layers to be convective, the converged solutions may be inaccurate because the model is only equipped to grow and merge convective zones, not shrink them. Therefore, the best practice is to start with three to five bottom convective levels. If the code runs into convergence issues after doing so, we recommended increasing this number slowly to check if it helps converge the solution.

4.3. Convergence

In most cloud-free cases, the code will converge. However, there are a few known cases where extra steps are needed to reach a converged solution. For equilibrium chemistry brown dwarf models with $T_{\text{eff}}$ between 1500 and 1700 K, an oscillating behavior is seen in the iterations of the $T(P)$ profile. In these cases, the $T(P)$ profile oscillates between two profiles with a constant temperature offset. This behavior is not unexpected and is caused by the sharp “cliff” in gaseous opacity at these effective temperatures. This can be seen in the cross-sectional maps in Figure 5. All four panels in Figure 5 show a sharp change in gaseous opacity around the 900–1700 K temperature range. The overlap of this cliff with the converged brown dwarf $T(P)$ profile at $T_{\text{eff}} = 1700$ K can also be seen in Figure 5. As already mentioned in Section 2.1.4, this sharp opacity “cliff” makes the convergence difficult around this temperature range. In order to overcome this, we recommend performing multiple runs for the same object by recycling the unconverged final $T(P)$ profile of each run as an initial guess profile for the next run until a converged solution is reached.

A similar convergence issue also occurs for objects with $T_{\text{eff}} < 300$ K. The bottom-left panel of Figure 2 shows how H$_2$O abundance in the vapor phase falls drastically due to H$_2$O condensation at such low temperatures. This sharp drop in the H$_2$O abundance causes the $T(P)$ profiles of these cold objects at the low-pressure regions to oscillate between multiple possible solutions. This behavior can be seen in the upper parts of the $T_{\text{eff}} = 300$ K $T(P)$ profile in Figure 5. We recommend two solutions to this problem: (1) either set the minimum pressure of the model pressure grid to no less than 1 mbar, or (2) use a SONORA profile as an initial guess and recycle the unconverged profile multiple times (∼4 times) to reach a converged state. The first recommendation helps the user exclude the part of the $T(P)$ profile that causes the instability in the convergence. As these objects are cold, the excluded pressure region is extremely cold (<120 K), which means that there is practically no contribution to the observables (e.g., total emergent flux).

9 Also see various tutorials, e.g., Download and Query from Sonora Profile Grid, and Accessing the pressure-temperature profile parameterizations.

5. Calculating Observables

After using the RCE model to calculate the atmospheric structure of a brown dwarf or an exoplanet, the user may want to compute various observables. PICASO has been updated to include the capability of calculating the 1D and 3D thermal emission spectrum, the 1D transmission spectrum, the 1D and 3D reflection spectroscopy, and phase curves. With a 1D $T(P)$ profile, users will likely want to produce 1D higher-resolution thermal emission spectra, reflection spectra, and/or transmission spectra of the planet or brown dwarf. These calculated observables can be directly compared with observational data.

Figure 15 shows an example of this with an exoplanet. The exoplanet has been assumed to be a Jupiter-mass and -sized planet with $T_{\text{int}} = 300$ K at a distance of 0.1 au around a Sun-like star. The RCE model described in this work has been used to calculate the planet-wide average atmospheric structure of this hypothetical planet once with chemical equilibrium and then again with chemical disequilibrium. The top-left panel in Figure 15 shows the $T(P)$ profile of this planet in each case. It is clear that self-consistent treatment of disequilibrium chemistry also impacts the $T(P)$ profile of irradiated planets compared to equilibrium chemistry models. We have then used the 1D radiative transfer routines of PICASO to compute the reflection albedo spectrum of this exoplanet. This albedo spectrum has been shown in the top-right panel of Figure 15. The planet is not very reflective after 0.5 μm, after which brightness from Rayleigh scattering tapers off. The transmission spectra of the planet is also modeled with PICASO and is shown in the bottom-left panel of Figure 15. This panel shows that the presence of disequilibrium chemistry can be clearly detected for a hot Jupiter like this with its transmission spectra between 3 and 5 μm. The thermal spectra are shown in the bottom-right panel, and the presence of disequilibrium chemistry in hot Jupiters can also be easily detected with their thermal spectra in the M band, or in various NIRSpec and/or NIRCam JWST modes. Note that for this illustrative example, we have used the planet-wide average computed $T(P)$ profile for all of these viewing geometries. However, a more self-consistent method to compute these viewing geometries would be to compute a dayside averaged profile for use in the zero-phase reflection and thermal spectrum, and a planet-wide average for the transmission spectrum.

PICASO has also been equipped with simple modules to couple the atmospheric models produced from this code with the evolutionary models from Marley et al. (2021) to calculate absolute Vega magnitudes of the modeled object in any filter of choice. The atmospheric model presented in this paper used with these radiative transfer tool are already available in PICASO and can be immensely helpful in modeling or planning observations using models.

6. Future Improvements

Currently, our publicly released model has the capability to model noncloudy brown dwarfs and exoplanets with equilibrium chemistry and disequilibrium chemistry. However, several future improvements of the model are needed to enhance its capability to capture a larger parameter space. Here we briefly discuss these needed future improvements. We note that all PICASO development is public, and open to community involvement.
6.1. Clouds

Our iterative climate model currently does not include the capability to treat atmospheric clouds self-consistently. The EGP code already includes condensation clouds with the Ackerman & Marley (2001) model and has been used to model cloudy L-dwarf atmospheres (e.g., Cushing et al. 2008; Stephens et al. 2009). Clouds can become important opacity sources in brown dwarf and exoplanetary atmospheres due to their strong scattering tendencies. This also has large effects on the $T(P)$ profile of these atmospheres (Morley et al. 2012, 2014). We plan to couple this model with the Python-based cloud model VIRGA (Batalha et al. 2020). Recently, VIRGA has been updated to include variable sedimentation efficiency ($f_{\rm sed}$) with height (Rooney et al. 2022). We will couple this updated VIRGA model with our model so that our models can be used to model hotter L-dwarfs, which are generally assumed to be cloudy and also much colder Y-dwarfs with H$_2$O clouds. This improvement will also enable us to apply our model to cloudy exoplanets and brown dwarfs.

6.2. 1D Chemical Kinetics Model

We use the quench-time approximation to model the effects of vertical mixing on the chemical abundance profiles in this model. This approximation is based on the parameterized mixing timescales from Zahnle & Marley (2014) outlined in Section 2.1.5. However, more robust 1D chemical kinetics models of treating disequilibrium chemistry caused by vertical mixing are now open-sourced like the VULCAN model (Tsai et al. 2017, 2021). We plan to couple our Python model with the VULCAN chemical kinetics model to increase the flexibility of our model. This will make our disequilibrium chemistry models more robust. Moreover, VULCAN also includes the capability to treat stellar irradiation induced photochemistry. This will lead to a significant improvement for applicability of our model to irradiated exoplanets where photochemistry can largely impact atmospheric chemistry, especially at lower pressures. This update will allow us to explore the impact of photochemistry on the atmospheric structure of exoplanets. We have already made progress for this coupling but still need to test and benchmark several aspects of the coupled code before this update is publicly available for use.

6.3. Time Evolution Version

Recently, Mayorga et al. (2021) and Robinson & Marley (2014) enhanced the EGP model to EGP+ by including a time-stepping version of the code, which can model the dynamic temporal response of the atmosphere to time-varying physical conditions like stellar irradiation changing with time. This is especially relevant to planets with highly eccentric orbits as the stellar irradiation can change by a large amount within one...
7. Conclusions and Summary

We presented a new open-source Python-based 1D RCE model as part of the PICASO package. This code derives its heritage from the EGP code developed by Marley & McKay (1999) based upon a Titan atmospheric model developed by McKay et al. (1989). The EGP code has been used to model brown dwarf and exoplanetary atmospheres for almost three decades now. The model is applicable to H-dominated atmospheres of both brown dwarfs and irradiated exoplanets. The model includes the capability to do calculations with both equilibrium chemistry and disequilibrium chemistry due to vertical mixing. The model includes options to use $K_{zz}$ values constant with height while performing disequilibrium chemistry runs. We have also included the capability to use self-consistent prescriptions of $K_{zz}$ within the model where the $K_{zz}$ will also iterate along with the $T(P)$ profile and atmospheric chemistry to ultimately reach a converged atmospheric state.

We have benchmarked this model with publicly available models from the SONORA BOBCAT grid and also with results from an independent model used by Phillips et al. (2020) to produce the ATMO 2020 atmospheric grid. For irradiated planets, the PICASO 3.0 models were benchmarked against the ATMO grid presented in Goyal et al. (2020), using the hot Saturn WASP-25 b as a test case. The chemical equilibrium version of the model was benchmarked both with models from the SONORA grid, ATMO 2020, and the ATMO models for WASP-25 b. The chemical disequilibrium models were benchmarked against models from the disequilibrium chemistry atmospheric models from the ATMO 2020 grid of models. This benchmarking analysis showed excellent agreement with PICASO.

Our model is open-source and publicly available for the community (we include several in-text links to code throughout this manuscript). Additionally, we outlined many recommendations in this work for proper usage of the model. This includes recommendations on choosing the atmospheric pressure grid for a science object, choosing the initial guess $T(P)$ profile for a science object and also the various parts of the parameter space where the model is known to face convergence issues. We also included ways of resolving these convergence issues.

We also released tutorials to apply this model for various science cases with the code. We plan to actively develop this model further to include clouds, couple it with 1D chemical kinetics codes for better robustness, and also include time-dependent effects like variable stellar irradiation within our model.

S.M. thanks the UC Regents Fellowship award for supporting him in this work. N.E.B. acknowledges support from NASA Astrophysics Division. N.E.B. and J.J.F. acknowledge support from NASA’s Interdisciplinary Consortia for Astrobiology Research (NNH19ZDA001N-ICAR) under award No. 19-ICAR19-2-0041. J.J.F. and M.S.M. acknowledge the support of NASA XRP grant 80NSSC19K0446. We thank the anonymous referee for helping us to improve the manuscript significantly.

Software: PICASO (Batalha et al. 2019, 2022), pandas (McKinney 2010), NumPy (Walt et al. 2011), IPython (Pérez & Granger 2007), Jupyter (Kluyver et al. 2016), matplotlib (Hunter 2007).

References

Abramowitz, M., & Stegun, I. A. 1970, Handbook of mathematical functions : with formulas, graphs, and mathematical tables (Washington, D.C.: U.S. Dept. of Commerce, National Bureau of Standards)

Ackerman, A. S., & Marley, M. S. 2001, ApJ, 556, 872

Allard, F., Allard, N. F., Homeier, D., et al. 2007a, A&A, 474, L21

Allard, F., Homeier, D., & Freytag, B. 2012, KZP, 370, 2765

Allard, N. F., Kielkopf, J. F., & Allard, F. 2007b, EPJD, 44, 507

Allard, N. F., Spiegelman, F., & Kielkopf, J. F. 2016, A&A, 589, A21

Allard, N. F., Spiegelman, F., Leininger, T., & Molliere, P. 2019, A&A, 628, A120

Allen, M., Yung, Y. L., & Waters, J. W. 1981, IGRA, 86, 3617

Amundsen, D. S., Baraffe, I., Tremblin, P., et al. 2014, A&A, 564, A59

Antonello, D. S., Tremblin, P., Manners, J., Baraffe, I., & Mayne, N. J. 2017, A&A, 598, A97

Arcangeli, J., Desert, J.-M., Line, M. R., et al. 2018, ApJl, 855, L30

Azzam, A. A. A., Tennyson, J., Yurchenko, S. N., & Naumenko, O. V. 2016, MNRAS, 460, 4063

Barber, R. J., Strange, J. K., Hill, C., et al. 2014, MNRAS, 437, 1828

Bard, A., A., & Kock, M. 1991, A&A, 248, 315

Bard, A., & Kock, M. 1994, A&A, 282, 1014

Barnum, T. S., Hauschildt, P. H., & Allard, F. 2001, ApJ, 556, 885

Barton, E. J., Yurchenko, S. N., & Tennyson, J. 2013, MNRAS, 434, 1469

Batalha, N., Rooney, C., & Mukherjee, S. 2020, natashabatalha/iriga: Initial Release, Version: v0.0, Zenodo, doi:10.5281/zenodo.3759888

Batalha, N., Rooney, C., Mukherjee, S., et al. 2022, natashabatalha/picasso: Release 3.0, Version: v3.0, Zenodo, doi:10.5281/zenodo.7254818

Batalha, N. E., Marley, M. S., Lewis, N. K., & Fortney, J. J. 2019, ApJ, 878, 70

Baudino, J. L., Bézard, B., Boccaletti, A., et al. 2015, A&A, 582, A83

Bell, K. 1990, JPhB, 13, 1859

Bell, K. L., & Berrington, K. A. 1987, JPhB, 20, 801

Bittner, D. M., & Bernath, P. F. 2018, VizieR On-line Data Catalog: J/ApJS/236/46

Binningham, B., Marley, M. S., Line, M. R., et al. 2017, MNRAS, 470, 1177

Binningham, B., Faherty, J. K., Gonzales, E. C., et al. 2021, MNRAS, 506, 1944

Burrows, A., Budaj, J., & Hubeny, I. 2008, ApJ, 678, 1436

Burrows, A., Ram, R. S., Bernath, P., Sharp, C. M., & Milsom, J. A. 2002, ApJ, 577, 986

Burrows, A., Marley, M., Hubbard, W. B., et al. 1997, ApJ, 491, 856

Burrows, A., Sudarsky, D., & Hubeny, I. 2006, ApJ, 640, 1063

Chabrier, G., Baraffe, I., Allard, F., & Hauschildt, P. 2000, ApJ, 542, 464

Coppola, C. M., Lodi, L., & Tennyson, J. 2011, MNRAS, 414, 487

Cushing, M. C., Marley, M. S., Saumon, D., et al. 2008, ApJ, 678, 1372

Deline, A., Hooton, M. J., Lendl, M., et al. 2022, A&A, 659, A74

Dulick, M., Bauschlicher, C. W., Jr., Burrows, A., et al. 2003, ApJ, 594, 651

Enoch, B., Cameron, A. C., Anderson, D. R., et al. 2011, MNRAS, 410, 1631

Fortney, J. J., Lodders, K., Marley, M. S., & Freedman, R. S. 2018, ApJ, 468, 1419

Fortney, J. J., Marley, M. S., & Barnes, J. W. 2007, ApJ, 659, 1661

Fortney, J. J., Marley, M. S., Lodders, K., Saumon, D., & Freedman, R. 2005, ApJ, 627, L69

Fortney, J. J., Visscher, C., Marley, M. S., et al. 2020, AJ, 160, 288

Fossati, L., Young, M. E., Shulyak, D., et al. 2021, A&A, 653, A52

Freedman, R. S., Lustig-Yaeger, J., Fortney, J. J., et al. 2014, ApJS, 214, 25

Freedman, R. S., Marley, M. S., & Lodders, K. 2008, ApJS, 174, 504

Fuhr, J. R., Martin, G. A., & Wiese, W. L. 1988, JPCRD, 17, 5
