# The VSPEC Collection: A suite of utilities to model spectroscopic phase curves of 3D exoplanet atmospheres in the presence of stellar variability.

Ted M Johnson [a,b,c,d,e,*], Cameron Kelahan [a,b,c], Avi M. Mandell [c], Ashraf Dhahbi [a,b,c], Tobi Hammond [a,b,c], Thomas Barclay [c,f], Veselin B. Kostov [c], Geronimo L. Villanueva [c]

[a] *Southeastern Universities Research Association, 1201 New York Avenue NW, Suite 430, Washington, DC, 20055, USA*
[b] *Center for Research and Exploration in Space Science and Technology, NASA Goddard Space Flight Center, 8800 Greenbelt Road, Greenbelt, MD, 20771, USA*
[c] *NASA Goddard Space Flight Center, 8800 Greenbelt Road, Greenbelt, MD, 20771, USA*
[d] *Nevada Center for Astrophysics, University of Nevada, Las Vegas, 4505 South Maryland Parkway, Las Vegas, NV, 89154, USA*
[e] *Department of Physics and Astronomy, University of Nevada, Las Vegas, 4505 South Maryland Parkway, Las Vegas, NV, 89154, USA*
[f] *University of Maryland, Baltimore County, 1000 Hilltop Cir, Baltimore, MD, 21250, USA*

Abstract

We present the Variable Star PhasE Curve (VSPEC) Collection, a set of Python packages for simulating combined-light spectroscopic observations of 3-dimensional exoplanet atmospheres in the presence of stellar variability and inhomogeneity. VSPEC uses the Planetary Spectrum Generator's Global Emission Spectra (PSG/GlobES) application along with a custom-built multi-component time-variable stellar model based on a user-defined grid of stellar photosphere models to produce spectroscopic light curves of the planet-host system. VSPEC can be a useful tool for modeling observations of exoplanets in transiting geometries (primary transit, secondary eclipse) as well as orbital phase curve measurements, and is built in a modular and flexible configuration for easy adaptability to new stellar and planetary model inputs. We additionally present a set of codes developed alongside the core VSPEC modules, including the stellar surface model generator vspec-vsm, the stellar spectral grid interpolation code GridPolator, and a Python interface for PSG, libpypsg.

Keywords: Exoplanet Atmospheres, Stellar Variability, Spectroscopic Phase Curve, Open Source, Planetary Spectrum Generator

1. Introduction

In the era of high-sensitivity transiting exoplanet characterization missions such as the Hubble Space Telescope (HST), the James Webb Space Telescope (JWST), and the future Atmospheric Remote-sensing Infrared Exoplanet Large-survey (ARIEL), spectral analysis of exoplanet atmospheres is increasingly sensitive to contamination due to stellar inhomogeneities (e.g. spots, granulation) and stochastic stellar variability (e.g. flares). As an example, recent analysis of the JWST/NIRSpec transit of GJ 486b by Moran et al. (2023) exposed a degeneracy between atmospheric absorption by water and water-rich spots on the stellar surface, and similar effects were seen in the transit spectrum of LHS 1140b (Cadieux et al., 2024) and many other recent transit measurements with JWST (e.g., Lim et al., 2023; May et al., 2023; Fournier-Tondreau et al., 2024). This "transit light source effect" (TLS, Rackham et al., 2018, see also Apai et al. (2018); Barclay et al. (2021); Garcia et al. (2022); Barclay et al. (2023)) occurs when the region of the stellar surface occulted by the transiting planet is not representative of the disk-integrated spectrum.

Stellar contamination may also cause errors in the extraction of thermal phase curve spectroscopy if the stellar spectrum changes significantly over the period of the planet's orbit; if the variations are non-linear, but only linear interpolation between subsequent secondary eclipses is used to remove the stellar contribution, the resulting planetary flux measurements will not be exclusively representative of the planet's emission. This will be particularly important for observations of planets around more variable stars such as active M-stars or later-type evolved host stars.

To adequately prepare for future observations and future missions, it will be necessary to demonstrate methods to mitigate these effects. This task requires a flexible tool that combines models of exoplanet atmospheres and stellar variability in a robust way. In this paper we present the VSPEC (Variable Star PhasE Curve[1]) Collection, a collection of open-source Python 3 packages to simulate observations of exoplanet systems with variable host stars. The VSPEC[2] package itself is merely an interface that pulls together a variable star surface model vspec-vsm[3], a

---

*Corresponding Author
Email address:* `ted.johnson@unlv.edu` (Ted M Johnson)

[1] `https://github.com/orgs/VSPEC-collab/repositories`
[2] `https://github.com/VSPEC-collab/VSPEC`
[3] `https://github.com/VSPEC-collab/vspec-vsm`

stellar spectra grid interpolation code Gridpolator[4], and a Python interface for the Planetary Spectrum Generator (PSG, Villanueva et al., 2018), `libpypsg`[5]. All of these codes together allow `VSPEC` to account for 3D planetary atmospheres, time-resolved effects of both planet and star, multiple geometries including transit and secondary eclipse, and realistic noise modeling.

In this paper we will describe the science motivation behind all of these codes, the details of each package and module in the `VSPEC` collection, and demonstrate examples of their use. Section 2 describes `VSPEC` and goes into some detail about setup, use, and the code's internals. In Section 3 we describe `libpypsg` as an interface for the Planetary Spectrum Generator. Section 4 describes `VSPEC`'s stellar model, `vspec-vsm`. We discuss the spectral model interpolation code GridPolator in Section 5. Section 6 is dedicated to the simple climate model that `VSPEC` uses to produce realistic phase curves in the absence of a user-supplied output file from a more sophisticated Global Circulation Model (GCM). Section 7 demonstrates a few simple `VSPEC` use cases. Finally, in Section 8 we will discuss the future of the code and issues it might have.

2. Using `VSPEC`

The

The typical flow of `VSPEC`'s code is broken into three shown in Figure 1, as well as described in the following sections. It is generally broken into four parts: i) read in the configuration, ii) compute planetary spectra, and iii) compute stellar spectra. See the `VSPEC` documentation page, iv) analyze, with steps (i) and (iv) varying widely by use case.

Most users of `VSPEC` will interact exclusively with two classes: `VSPEC.ObservationModel` and `VSPEC.PhaseAnalyzer`. The bulk of the work is done by the `ObservationModel` class, as it reads in a user's inputs, runs the model, and writes output files. `VSPEC` produces many output files, which themselves can be read easily using tools such as Astropy (Astropy Collaboration et al., 2013, 2018, 2022). However, the `PhaseAnalyzer` class makes it easy to read them as an ensemble, as well as quickly pull out portions of the data that are of interest.

Section 2.1 describes initializing the `ObservationModel`. Section 2.2 describes computing the planetary phase curve using the `ObservationModel.build_planet()` method. The next step, calling `ObservationModel.build_spectra()`, recalls the data produced by PSG previously, but combines it with a spectral model of a variable star, scaling reflected light and noise appropriately. This step, along with detailed descriptions of various substeps, is described in Sections 2.3, 2.4, 2.5, and 2.6.

Use of the `PhaseAnalyzer` object is not described in detail as it mostly acts as a container, but Section 7 includes many examples of its use. Additionally, the VSPEC documentation page[6] for contains detailed API documentation.

2.1. Configuring `VSPEC`

`VSPEC` configurations are designed to minimize human errors by providing two equivalent formats. The first, a file written in YAML, is optimized for human readability. For example, the `system` section, which describes the relationship between the observed planetary system and the observer, could be written as shown in Listing 1.

```
1  system:
2      distance: 12.4 pc
3      inclination: 89.7 deg
4      phase_of_periastron: 0 deg
```
Listing 1: `VSPEC` YAML Configuration

Note that the units of each of these parameters is included in a human-readable way. Internally, `VSPEC` casts each of these parameters to an Astropy `Quantity` instance, so that the user can input any desired value and unit combination so long as the physical type is correct – i.e. it would be equivalent to write `3.83e17 m` as the value for the `distance` parameter.

The second input method available is to directly initialize the Python object that `VSPEC` uses internally to store its model parameters. The top-level object, `InternalParameters`, is structured to mirror the YAML input file, with one argument per YAML section. The same configuration shown in Listing 1 would be written in Python as shown in Listing 2.

```
1  from astropy import units as u
2  from VSPEC.params import *
3  params = InternalParameters(
4      ... # Other arguments
5      system=SystemParameters(
6          distance=12.4*u.pc,
7          inclination=89.7*u.deg,
8          phase_of_periastron=0*u.deg
9      )
10 )
```
Listing 2: `VSPEC` Python configuration

This is convenient for producing configurations programmatically, for example for producing a grid of model phase curves.

However the user decides to provide model parameters, they are read into the main `VSPEC` object: the `ObservationModel`. This is the class that runs the model and handles files. Upon initialization a local working directory (by convention `.vspec/`) is created to store simulation outputs and intermediate files.

2.2. Planetary Phase Curve

`VSPEC` generates a phase curve when the `ObservationModel.build_planet()` method is called. It

---

[4] `https://github.com/VSPEC-collab/GridPolator`
[5] `https://github.com/tedjohnson12/libpypsg`

[6] https://vspec-collab.github.io/VSPEC/0.6.0/index.html`https://vspec-collab.github.io/VSPEC/0.6.0/index.html`
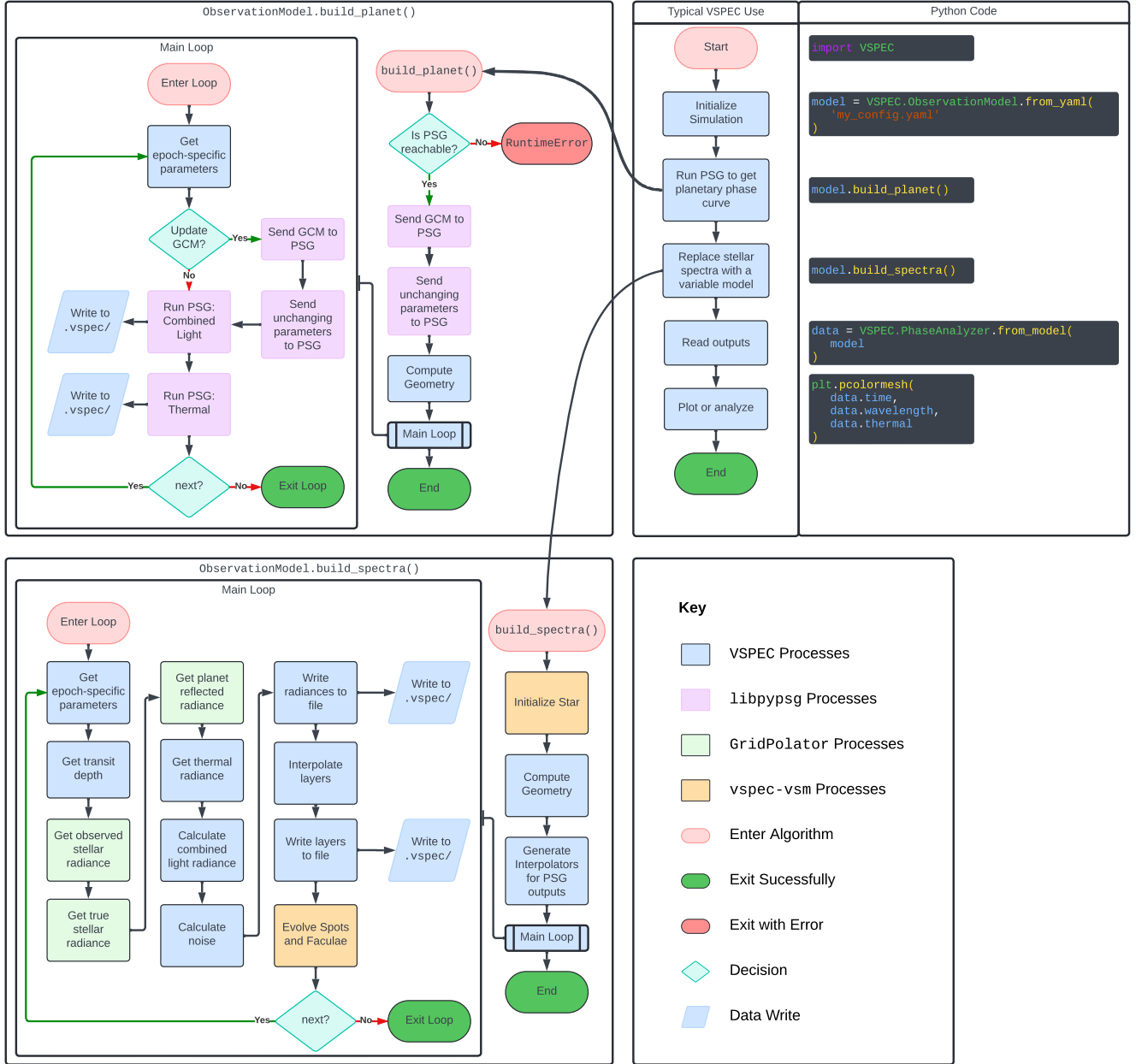
Figure 1: Flowchart diagram outlining the typical workflow of a VSPEC simulation. Upper right: Processes representing the main VSPEC stages alongside Python code snippets. Nearly all VSPEC use cases involve the code `model.build_planet()`; `model.build_spectra()`, though the input file and the particulars of the analysis can be very different. Upper left: `build_planet()` algorithm. This portion of the code is responsible for interfacing with PSG (via `libpypsg`, see Section 3) to generate a phase curve. The decision labeled "Update GCM?" evaluates to true when a time-dependent climate model is specified in the input file, or when the PSG configuration returned by the latest API call is longer than 1500 lines, indicating that PSG should be reset. For convenience, the PSG GlobES application appends new configuration files to those from previous API calls, but stops reading after 2000 lines for security reasons. Therefore, the internal configuration file must be reset on occasion. Lower left: `build_spectra()` algorithm. This portion of the code is responsible for adding stellar variability into our results. We use the outputs from the `build_planet()` step to create interpolator instances (implemented as `GridPolator.Gridspectra` objects, see Section 5) for the time- and wavelength-dependent values of relevant quantities (e.g., thermal emission flux, transit depth, various types of noise). We then step through each epoch in our time series and evaluate the variable star model (using `vspec-vsm`, see Section 4). We replace the stellar spectrum from PSG with that of our model, scaling the noise appropriately. Finally, we evolve the star to the time of the next observation. Lower right: Diagram key. Note that, in general, all the steps marked as `GridPolator` processes also involve `vspec-vsm`.

does this though a series of API calls to the Planetary Spectrum Generator (PSG Villanueva et al., 2018), using `libpypsg` (see Section 3) to interface with the API. Initially, configurations are sent that give PSG information about the system that does not change with time – that includes the 3D Global Circulation Model (GCM) and instrument parameters. `VSPEC` then enters the main loop, iterating through each observation epoch (e.g. integration or combination of integrations) and making a pair of API calls: one to get a combined light spectrum and one to get the thermal flux only. Results from PSG in each epoch are stored locally in the `.fits` format.

2.2.1. Phase Sampling

Making API calls to PSG can be computationally expensive, especially when trying to resolve small structures in the climate model. In order to reduce unnecessary calls to PSG, the temporal sampling of the planetary spectrum is independent of the stellar model/output sampling. When computing the planetary flux in the output files, the raw PSG output is interpolated and averaged over each time step using the trapezoid integration rule. By default one spectrum is computed with PSG at each interface separating time steps; this eases the calculation of edge effects between steps.

2.3. Stellar light curve

The last step is to replace the stellar spectra used by PSG with `VSPEC`'s stellar model by calling the `ObservationModel.build_spectra()` method. In this step `VSPEC` acts as a bridge between the stellar surface model `vspec-vsm` (which models spatial changes to the star with time, see Section 4) and a grid of pre-computed stellar spectra (from the GridPolator package; see Section 5). The star is initialized and allowed to evolve for a user-specified amount of time in order for spots and faculae to approach growth-decay equilibrium (this is also called the "burn in" phase). The star is then evolved at the output cadence, generating a set of key-value pairs at each epoch that describe the surface temperature and coverage fractions of the portion of the stellar disk visible to the observer. These $T_{\rm eff}$s and coverage fractions are fed into GridPolator to produce composite stellar spectra.

2.4. Reflected light

`VSPEC` then produces a composite spectrum of the portion of the star visible to the planet to accurately incorporate the time-dependent stellar flux into the reflected-light spectrum. To compute the total reflected light in the output files, `VSPEC` first divides the reflected light flux from PSG by the PSG stellar model to obtain the apparent albedo (I/F). This albedo is then multiplied by the composite stellar spectrum (facing the planet) to obtain the total reflected flux.

2.5. Simulated Noise

Alongside the planetary radiance files, PSG returns a file giving the analytic noise estimate for each epoch broken down by source. We assume that the total noise is the quadrature sum of its constituent parts and that the `source` column (i.e. photon noise) is the only one affected by replacing one stellar spectrum with the other. This means the photon noise can be calculated:

$$N_{\rm photon} = N_{\rm photon,\ PSG} \sqrt{\frac{F_{*,\ \rm VSPEC}}{F_{*,\ \rm PSG}}} \qquad (1)$$

where F_* is the stellar flux. The total noise associated with the integration is then

$$N_{\rm tot} = \sqrt{N_{\rm photon}^2 + N_{\rm detector}^2 + N_{\rm telescope}^2 + N_{\rm background}} \qquad (2)$$

2.6. Transit and Eclipse

Users must be careful when using `VSPEC` to simulate transit and eclipse measurements because of the vast timescale differences between those events and the typical phase variations of a planet – a sparsely sampled phase curve that happens to include one epoch of transit will not adequately sample the light curve accurately during a transit or eclipse because the interpolator knows no better than to assume a linear interpolation scheme. It is recommended that users use a high cadence for observations of transits and eclipses so that each event will be sampled by more than one point.

In the case of a transiting geometry, `VSPEC` computes the spectrum of the occulted portion of the star in order to properly simulate the transit light source effect (TLS Rackham et al., 2018). This spectrum is the flux blocked by the planet in the case that it is a solid occulting circle, and therefore produces a flat transmission spectrum. The effect of atmospheric transmission is added by comparing the PSG-computed effective radius to a purely geometric calculation of such a "bare-rock" transit depth:

$$F_\lambda = F_{\rm rock,\lambda}\, F_{\rm PSG,\lambda} \left(\frac{R_*}{R_p}\right)^2 \qquad (3)$$

where F_λ is the flux of the variable star surface occulted by the planet with the atmosphere considered. $F_{\rm rock,\lambda}$ is the flux of the surface occulted by a bare rock with the planet's radius, $F_{\rm PSG,\lambda}$ is the flux computed by PSG to be lost due to occultation and transmission, and R_* and R_p are the stellar and planetary radii, respectively.

In the case of a total eclipse of the planet, the planetary thermal and reflected contributions are set to zero flux. However, in the case of a partial eclipse (large impact parameter), `VSPEC` computes the fraction of the planet that is visible to the observer and reduces both thermal and reflected flux accordingly. In this case `VSPEC` treats the planet homogeneously; however, future work could utilize PSG/GlobES's ability to return a hypercube of spatially resolved spectra in order to construct planetary spectra only from regions of the planet that are not eclipsed.

3. Interfacing with the Planetary Spectrum Generator via `libpypsg`

PSG[7] (Villanueva et al., 2018) is a powerful radiative transfer tool that is ubiquitous in the exoplanet and solar system atmosphere fields. Either through its web graphical interface, its public API, or the local version available through Docker, PSG allows users to simulate observations of planets, comets, and moons with a variety of geometries and realistic noise models.

```python
import libpypsg
from astropy import units as u
from netCDF4 import Dataset

# This next line only needs to be run
#    once per user.
#    These settings are
#    saved to `~/.libpypsg/settings.json`
libpypsg.settings.save_setting(
    api_key='my_api_key',
    url='https://psg.gsfc.nasa.gov/api.php'
)

gcm_data = Dataset('path/to/gcm.nc')

cfg = libpypsg.cfg.PyConfig(
    target=libpypsg.cfg.Target(
        object='Exoplanet',
        name='proxima-cen-b'
        ...
    ),
    geometry=libpypsg.cfg.Observatory(...),
    generator=libpypsg.cfg.Generator(...),
    telescope=libpypsg.cfg.SingleTelescope(...)
    noise=libpypsg.cfg.CCD(...)
    gcm=libpypsg.globes.waccm_to_pygcm(
        gcm_data, itime=0,
        molecules=['H2O','CO2'],
        aerosols=['Water','WaterIce'],
        mean_molecular_mass=48.0)
)

caller = libpypsg.APICall(
    cfg=cfg, app='globes',
    output_type='rad'
)
response = caller()
print(response.rad['Wave/freq','proxima-cen-b'])
```
Listing 3: `libpypsg` Example Input

VSPEC uses a Python interface for the PSG API system called `libpypsg`, which is a general stand-alone library built for any Python-based API call to PSG. `libpypsg` draws inspiration from object-relational mapping in frameworks such as Django[8] to encode a PSG configuration file as a native Python object. Fields of a `libpypsg` data model represent one or more lines of a configuration file, and the two representations are interchangable without loss of information. Upon creation of this configuration as a `PyConfig` object, a user can create an `APICall` instance which handles sending a request to PSG via Python's

`requests`[9] package. This dedicated caller is useful because it reads a user's settings (e.g. an API key or the URL of the PSG API endpoint) from a file on their hard drive. This makes it much simpler to publish code that utilizes PSG because 1) it eliminates the danger of accidentally publishing an API key, and 2) it means that users can run the same code regardless of whether they have a local installation of PSG.

```
Wave/freq  proxima-cen-b
    um       W / (um m2)
----------- -------------
        1.0   1.81314e-20
       1.02    3.1926e-20
     1.0404   4.95983e-19
       ...          ...
16.97557767   7.54221e-20
17.31508922   7.49258e-20
17.66139101   7.35507e-20
Length = 146 rows
```
Listing 4: `libpypsg` Example Output

A powerful feature of `libpypsg` is that it provides a universal interface for 3D atmosphere models and PSG's Global Emission Spectra (GlobES) application (Fauchez et al. 2024, in prep; see also Kofman et al. 2024, accepted) GlobES expects 3D data to be presented in a binary data format, and converting a model to this format is non-trivial. `libpypsg`'s `PyGCM` class provides a native Python representation of GlobES input, and includes built-in methods for converting popular climate models like ExoCAM (Wolf et al., 2022), the Whole Atmosphere Community Climate Model (WACCM, Marsh et al., 2013), and ExoPlaSim (Paradise et al., 2022) to the correct format.

Listing 3 shows a minimal example that uses `libpypsg` to make a call to PSG/GlobES, with the output shown in Listing 4. Here `libpypsg` converts a GCM output from the netCDF format to a `PyGCM`, which, along with data model classes like `Target` and `Observatory`, is used as an argument to construct a `PyConfig` class instance. This configuration object is then passed to an `APICall` instance. When the API call is made, the contents of the `PyConfig` are included in the body of the POST request in the `file` field. PSG reads this the same as it would if we had written those contents to a file `config.txt` and passed the `--data-urlencode file@config.txt` option to `cURL`, as are the instructions in the PSG handbook (Villanueva et al., 2022, pp. 162-163). More details on use of this library including its API and examples can be found on the `libpypsg` documentation page[10].

4. Stellar Variability Model

The VSPEC stellar model is designed in a modular fashion to allow for both simple and complex behaviors. The

---

model is available as the Python package `vspec-vsm`, which is based on the earlier stellar spot model `spotty` (Barclay et al., 2021). It is spectral-model agnostic, meaning that it relies on another package (GridPolator, see Section 5) to compute the actual variable spectra. Instead, the surface model takes a pair of latitude and longitude points as input and returns a dictionary $\{T_{\mathrm{eff}} : f_{T_{\mathrm{eff}}}\}$ where the keys T_{eff} are the effective temperatures of each surface component and the values $f_{T_{\mathrm{eff}}}$ are the fraction of the visible stellar disk with that effective temperature.

The main model class is the `Star` object, which describes bulk properties (e.g. radius, photospheric T_{eff}, limb darkening parameters) in addition to acting as a container for objects representing sources of variability such as `StarSpot`, `Facula`, `StelarFlare`, and Granulation, which are described below. For a complete description of the code, including examples, see the `vspec-vsm` documentation page[11].

4.1. Stellar Surface Grid

Sources of variability that exist as major structures on the stellar surface (i.e. spots and faculae) are encoded as deviations from the photospheric effective temperature on a grid of points representing the stellar surface. Each time the fractions of each surface component are calculated, the grid is initialized with constant T_{eff}. The code then iterates through each spot and facula, replacing portions of the grid with the effective temperatures that make up each structure.

The simplest grid to build is one that is rectangular in latitude and longitude, and in which the area that each point represents decreases towards the poles. However, this can lead to unnecessary spatial oversampling at the poles, and may lead to poor resolution at the equator to keep the total number of points from being excessively high. To mitigate these problems, `vspec-vsm` supports a "grid" generated using a Fibbonacci spiral to create a near-isotropic distribution of points. This is especially useful when simulating transits, giving optimal sampling at the stellar equator. It is also useful for resolving small structures on the surface, as the number of points needed for a given minimum resolution is greatly reduced.

4.2. Spots

Our star spot model is nearly entirely based on observations of the Sun. Sunspots can be resolved and are well-studied, whereas spots on other stars (especially non-solar-type stars) can only be observed indirectly. We therefore designed our spot model to mimic the behavior of sunspots but with parameterized values for spot temperature and lifetime that can be matched to observations of other stellar types and ages. On the Sun, spots have two regions shown in Figure 2: the dark, central umbra and the lighter,
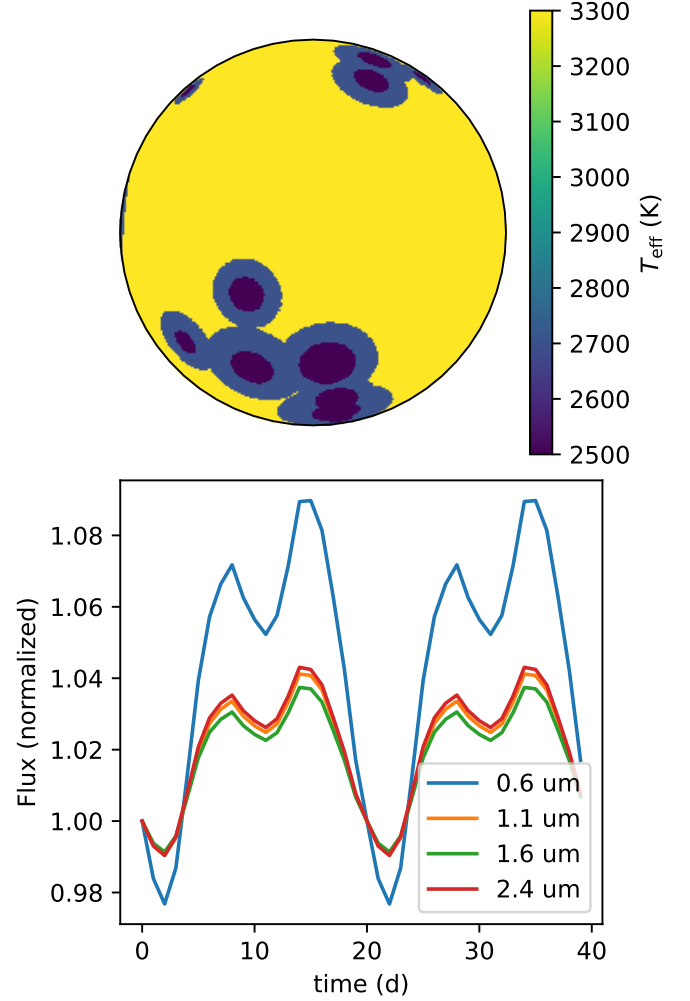

Figure 2: **Top**: The surface map of a star with spots. The star has a photospheric temperature of 3300K, a penumbra temperature of 2700K, and an umbra temperature of 2500K. **Bottom**: An example light curve of a spotted star. Flux variations are due to varying spot coverage as a function of sub-observer point. Two rotational periods are shown.

---

[11] `https://vspec-vsm.readthedocs.io/en/latest/`

surrounding penumbra (a detailed review of sunspot behavior, including sizes and lifetimes, is described in Solanki (2003)). The spot area as a function of time is

$$A(t) = \left\{ \begin{array}{ll} A_0 \exp{((t-t_0)/\tau)}, & \text{if } t \leq t_0 \\ A_0 - W(t-t_0), & \text{if } t > t_0 \end{array} \right\} \quad (4)$$

where A_0 is the maximum area reached, t_0 is the time of the maximum, τ is the exponential growth rate, and W is the linear decay rate.

In addition to modeling the variability produced by a population of spots, `vspec-vsm` contains utilities to produce those populations given some description. These parameters include the average spot area (lognormally distributed, Bogdan et al., 1988), T_{eff}, growth and decay rates, and the method for distributing spots on the surface. These `SpotGenerator` objects are responsible for evolving the spot population as the star ages. The number of spots created in some interval Δt is:

$$N(\Delta t) = \frac{4\pi R_*^2 f_{\text{spot}}}{A_{\text{mean}}} \frac{\Delta t}{T} \frac{\Delta t}{\tau_{\text{spot}}} \quad (5)$$

where R_* is the stellar radius, f_{spot} is the fraction of the surface covered by spots in growth-decay equilibrium, A_{mean} is the lifetime-averaged mean spot area, and $T \tau_{\text{spot}}$ is the spot lifetime, defined as:

$$T\tau_{\text{spot}} = \frac{A_0}{W} - \frac{1}{\tau} \ln\left(\frac{A_{\text{min}}}{A_0} \right) \quad (6)$$

where A_{min} is the area given to each spot when it is initialized. There are also two modes to distribute spots on the star's surface. The first distributes them isotropically; the second, modeled after analysis of spots on the Sun by Mandal et al. (2017), concentrates the spot latitude around $\pm 15°$.

4.3. Faculae

Faculae are magnetically-generated regions of the solar surface that usually appear as bright points near the limb; we employ the "hot wall" model (Spruit, 1976) where faculae are described as three-dimensional pores in the stellar surface with a hot, bright wall and a cool, dark floor, as shown in Figure 3. Their three-dimensional structure causes faculae's observational properties to change depending on their angle from disk-center. Close to the limb, the hot wall is visible to the observer, and faculae appear as bright points; near the center, however, the cool floor is exposed and faculae appear dark. To consider this effect in the faculae light curve, we compute the fraction of each facula's projected area – the area on the disk it would occupy as seen by the observer – that is occupied by the hot wall versus the cool floor. Mathematically, this is no different from finding the area of the intersection of two overlapping ellipses, and we use this conceptualization to calculate the contributions of each portion, as described below.

As shown in Figure 3, peering through the top of the pore the observer can see a portion of the hot wall and a portion of the cool floor. Consider an ellipse centered at the origin with semimajor axis R and semiminor axis μR, oriented so that the semimajor axis points along the y-axis. Now consider a second, identical ellipse, displaced along the x-axis by some amount $0 < d < \mu R$. The intersection of these two ellipses is the portion of the cool floor that is visible, and the area of the second ellipse outside the intersection is the portion of the hot wall that the observer sees. We are not interested in the absolute areas of each of these regions, only how they relate to the total ellipse area $\pi \mu R^2$, and so we can generalize to the case that $\mu = 1$ (i.e. circles). This problem is solved for arbitrary circles by Weisstein (2004); the details of their derivation are not relevant to this text, but we use their result to create a function $f(d, R)$ that computes the fraction of a circle with radius R and center $(d, 0)$ that is inside the unit circle. For a facula with radius r, wall height h, and angle from disk center θ, we find that the effective area of the cool floor is:

$$\frac{A_{\text{floor}}}{A_{\text{total}}} = f\left(\frac{h}{r} \tan\theta, 1 \right) \quad (7)$$

and for the hot wall it is:

$$\frac{A_{\text{wall}}}{A_{\text{total}}} = 1 - f\left(\frac{h}{r} \tan\theta, 1 \right) \quad (8)$$

According to studies of solar faculae (Topka et al., 1997), faculae temperatures (of both the floor and wall) are dependent on the facula radius, while depth appears to be constant. They also find that the smallest faculae have no visible floor, and that even at disk center they appear as bright points. We parameterize the floor temperature to be

$$\Delta T_{\text{eff,floor}} = \left\{ \begin{array}{ll} \text{Not visible,} & \text{if } r < r_{\text{min}} \\ m_{\text{floor}}(r - r_{\text{min}}) + \Delta T_{\text{eff,floor,0}}, & \text{if } r \geq r_{\text{min}} \end{array} \right\} \quad (9)$$

where r is the radius of the facula, r_{min} is the minimum radius where the floor is visible, $\Delta T_{\text{eff,floor,0}}$ is the difference between the floor and photosphere T_{eff} at r_{min}, and m_{floor} is the slope of the relationship with units of [temperature] [length]$^{-1}$. Similarly, the wall temperature is parameterized as

$$\Delta T_{\text{eff,wall}} = m_{\text{wall}} r + \Delta T_{\text{eff,wall,0}} \quad (10)$$

where $\Delta T_{\text{eff,wall,0}}$ is the temperature of a zero-radius facula and m_{wall} is the slope of the radius-temperature relationship with units of [temperature] [length]$^{-1}$. These relationships can be defined by the user, for example setting $m = 0$ for constant temperatures.

Facula lifetimes are defined as the time it takes its radius to decay by e^{-2}. Because faculae grow and decay exponentially at the same rate, each facula spends one lifetime with a radius greater than e^{-1} of its maximum. Each facula is born and dies at a radius of e^{-2} of its maximum, effectively

existing in the code for two lifetimes. Hovis-Afflerbach and Pesnell (2022) suggest a typical facula lifetime to be on the order of 6 hours, with a distribution that resembles a Poisson function; however we choose a log-normal distribution for both lifetime and maximum radius because it does not allow for these values to be 0. We choose to correlate lifetime and radius so that they are determined by a normalized random distribution - for each new facula, a value is randomly drawn and determines both the facula lifetime and maximum radius.

4.4. Flares

Flares are an important source of stellar variability on short timescales. `vspec-vsm`'s flare light curve model is based on the `xoflares` package (Barclay and Emilygilbert, 2020), which itself is based on empirically derived light curve shapes from Davenport (2016). However, `xoflares` is designed to fit light curves in a single spectral band, so we add additional parameters to extend it to a multiwavelength light curve. We completely describe a flare using its temperature, total energy, full width at half maximum (FWHM), and the time of its peak. We model a flare as a hot, optically thin region above the photosphere that produces a blackbody spectrum. In this model, the temperature is constant, and the sharp rise and fall seen in the light curve is caused by the region's rapidly changing area. This simplification allows the total flare energy E to be used as a normalization factor:

$$\int_{-\infty}^{\infty} A \, dt = \frac{E}{\sigma T^4} \tag{11}$$

where A is the time-dependent area of the flare region, T is the constant flare temperature, and σ is the Stefan-Boltzmann constant. This allows flare light curves to be produced with a fixed bolometric luminosity. Because the temperature is fixed, the relevant quantity for modeling the flux of a flare in a given integration is the integrated time-area $\int A \, dt$. These quantities are computed numerically and fed to `VSPEC` along with T to produce spectra with the appropriate absolute flux.

Before simulating an observation, `VSPEC` asks `vspec-vsm` to pre-compute a population of flares based on a power-law frequency-energy relationship based on Kepler and TESS light curves (Gao et al., 2022):

$$\log \left(f/[\text{day}] \right) = \beta + \alpha \, \log \left(E/[\text{erg}] \right) \tag{12}$$

where f is the frequency of flares with energies $\geq E$. We compute the number of expected flares over a time duration and determine the number to create N by a random Poisson draw. We then generate N flares with energies determined by the quantile function:

$$E = E_{\min}(1 - X)^{1/\alpha} \tag{13}$$

where $E_{\min}$ is the minimum considered flare energy and X is a random number on the interval $[0, 1]$. We set the
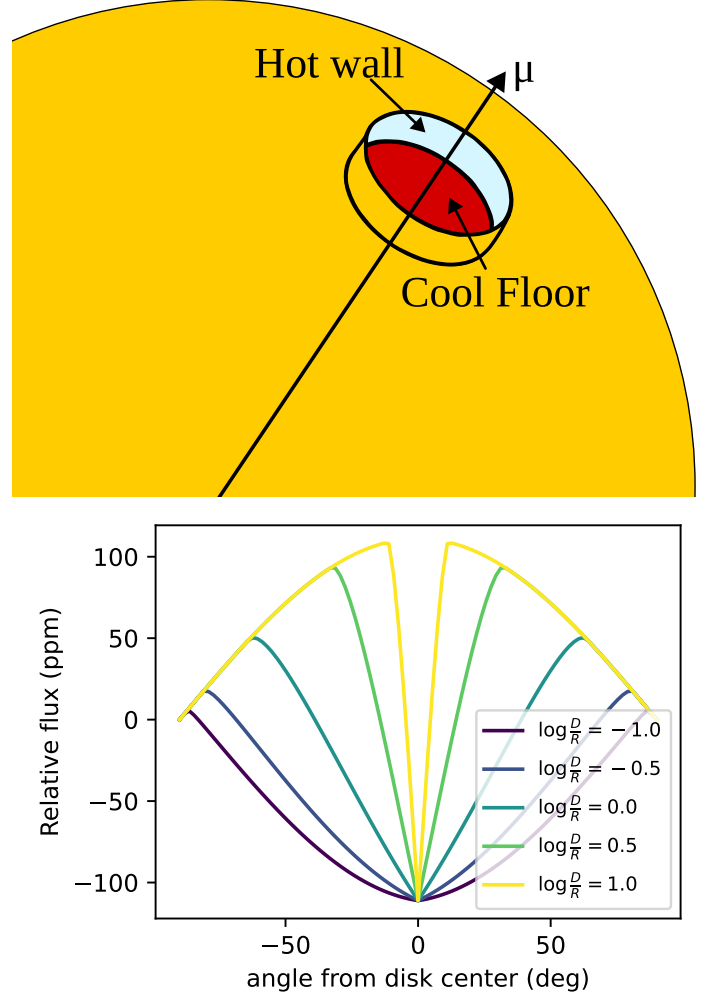

Figure 3: "Hot Wall" model of faculae. Faculae structure causes their contrast to be dependent on their distance from the center of the disk. **Top**: Depiction of a facula on the limb of a star. The hot wall is exposed to the observer causing the pore to appear bright. At disk center, the cool floor is most visible. **Bottom**: The effects of depression depth and viewing angle on facula brightness. The 3D structure of faculae is most apparent when radius $\sim$ depth. A toy flux model was used to demonstrate the shape of these curves, but in practice their magnitudes depend on stellar spectral models.
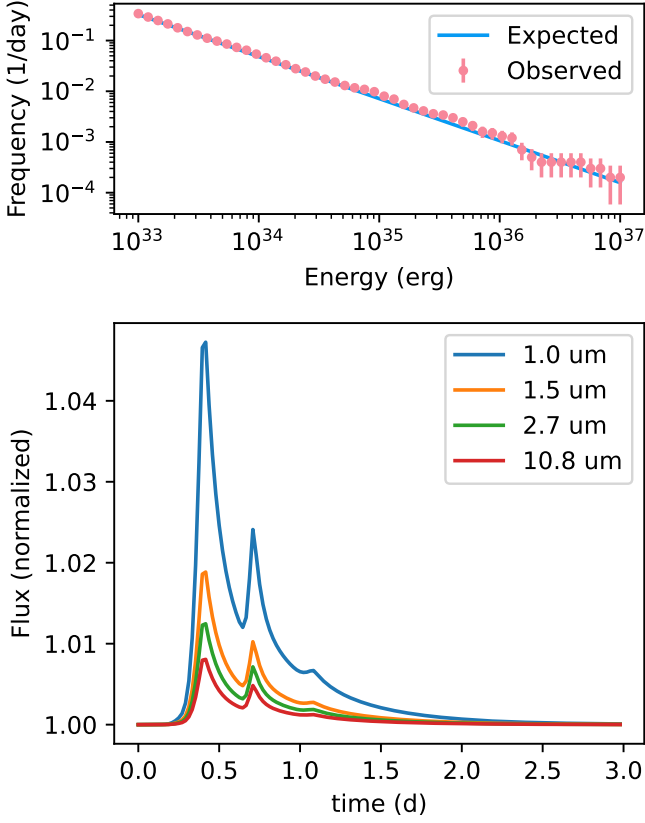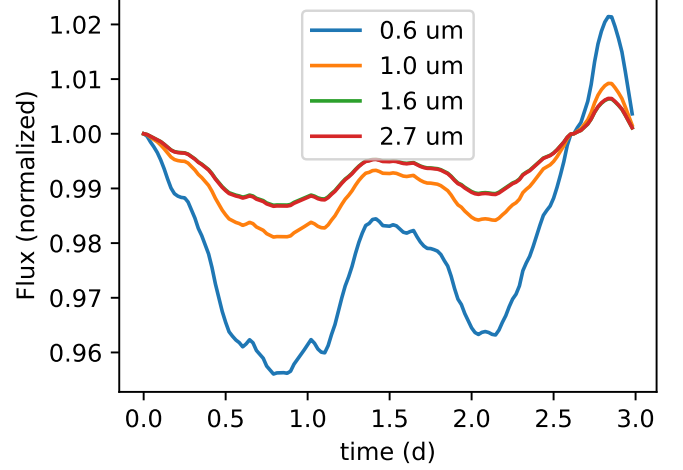
8

Figure 5: Effect of granulation on the light curve of a star. The parameters chosen in this simulation mean that, on average, 10% of the stellar surface is covered by the cool, inter-granule region with a temperature 300K lower than the surrounding photosphere, and that this coverage varies by about 1% on a timescale of 6 hours.

by a Gaussian process (GP) using the TINYGP package (Foreman-Mackey et al., 2024) following the methodology of Gordon et al. (2020). The GP uses a custom kernel function based on a power spectrum (Anderson et al., 1990; Kallinger et al., 2014) to produce random changes in the granulation coverage.

4.6. Transits & Limb Darkening

In the case of a transiting geometry, vspec-vsm is responsible for computing the properties of the portion of the stellar disk that is occulted. It does this by first projecting every visible point (i.e. those for which $\mu > 0$, μ is the cosine of the angle from disk-center) orthographically onto an x-y plane where the stellar disk is represented by the unit circle. The planet, in this projection, is represented by a circle centered at $(a/R_* \sin\phi,\ a/R_* \cos\phi \cos i)$ with a radius of R_p/R_*, for semi-major axis a, stellar radius R_*, planetary radius R_p, phase past midtransit ϕ, and inclination i. It then iterates through each point on the surface, computing the distance in projected coordinates to the center of the occulting disk. Also, for each point, an effective radius is calculated in the projected coordinates

$$r_{\text{eff}} = 2\mu\sqrt{\Omega/(4\pi\,[\text{sr}])} \qquad (14)$$

where Ω is the solid angle of each point. Any point for which the distance is less than $R_p/R_* + 2r_{\text{eff}}$ from the occulting disk's center is flagged as possibly interacting with the planet. The majority of these points are completely covered by the planet, and those that are less than $R_p/R_* - 2r_{\text{eff}}$ from the center of the occultation are immediately marked as such. The remaining points – those near the edge of the occultation – are subdivided into a
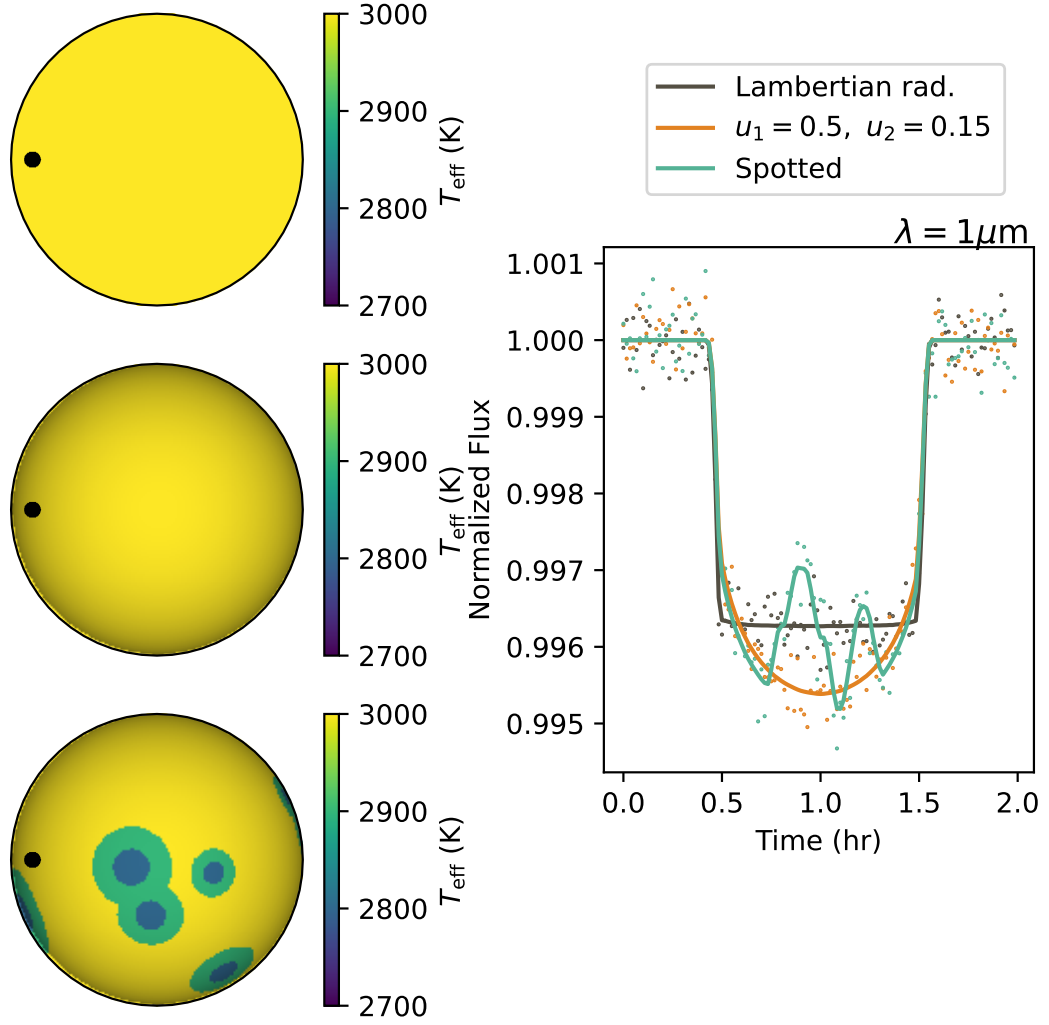

Figure 4: **Top**: Generated flare frequencies compared to those expected from Gao et al. (2022), generated using a 10,000 day simulation. The y-axis shows the frequency of flares with energies greater than or equal to the value of the x-axis. Error bars based on the square root of the number of observed flares as this is a Poisson process. **Bottom**: light curve of a star flaring with the same power-law slope as the top panel, but the intercept (β) has been increased by 0.3 for visual effect. The mean flare temperature is 9000 K and the mean FWHM is 3 hours.

default values $\alpha = -0.829$, $\beta = 26.87$ from Gao et al. (2022), but these can be adjusted by the user. Figure 4 shows our simulation results compared to their power-law values.

4.5. Granulation

Granulation is a source of stellar variability that arises from convection near the surface of the star. The result is a stellar surface that is not constant in temperature and that changes on very short timescales. Hydrodynamic simulations of stellar atmospheres (e.g. Magic and Asplund, 2014) describe hot granules of rising gas surrounded by cooler, sinking regions with a temperature a few percent lower than the nominal value.

We model granulation as a global process, and its effects are computed after the effects of spots and faculae. Of the remaining "quiet" photosphere (i.e. the regions not covered in spots of faculae), a fraction is computed to be part of the cool inter-granule surface; the surface coverage of the cool region at any given time is computed

Figure 6: **Right**: Transit light curves at 1 μm for a Lambertian (i.e. no limb darkening) star, one with significant limb darkening ($u_1 = 0.5$, $u_2 = 0.15$, Espinoza and Jordán, 2015), and one with that same limb darkening law, but with the addition of spots. In each case a 1 R_e planet transits a 0.12 $R_\odot$ ($T_{\text{eff}} = 3000$ K) star with an observation cadence of 1 minute. Scattered points show the observation with simulated noise, considering a single wavelength channel from JWST/NIRSpec G395H binned to a resolving power of 270 and a target distance of 5 pc. **Left**: Stellar surface maps including translucent masks to show limb darkening and transit. The Lambertian star (top) does not exhibit any limb darkening, and its disk has constant surface brightness. However, the limb-darkened stars (middle and bottom) are much brighter at disk-center than near the limb. The dark circle on the leftmost limb of each star is the region occulted by the planet when it is 0.7° from mid-transit. In the spotted star (bottom) the planet crosses over several spots as it moves from left to right. Note in the light curves that while no spot is covered, the transit depth is larger, and the effective radius of the planet appears higher. However, when a spot is crossed, the transit depth decreases dramatically.

10

100×100 point meshgrid. A 2D numerical integral is performed over this grid to compute the fraction of each point that is inside the occulting disk.

When computing the disk-integrated flux (including flux removed via a transit) the weight of each point is determined by the geometric projected area as well as a quadratic limb darkening law (see Espinoza and Jordán, 2015). We treat limb darkening as a weight factor on the contribution of each point to the disk-integrated spectrum that depends on μ. Points that are part of spots and faculae are treated the same as the photospheric points – their respective weights are increased if they lie near the center and decreased if they lie on the limb. This treatment allows the user to specify two parameters u_1 and u_2 that describe the flux variation as a function of angle from disk-center. A caveat of this method is that there is no dependence on wavelength. A better treatment would vary the temperature of the stellar model as a function of μ; we simply apply a gray scaling factor to the flux. Figure 6 compares transit light curves for two different sets of limb darkening parameters.
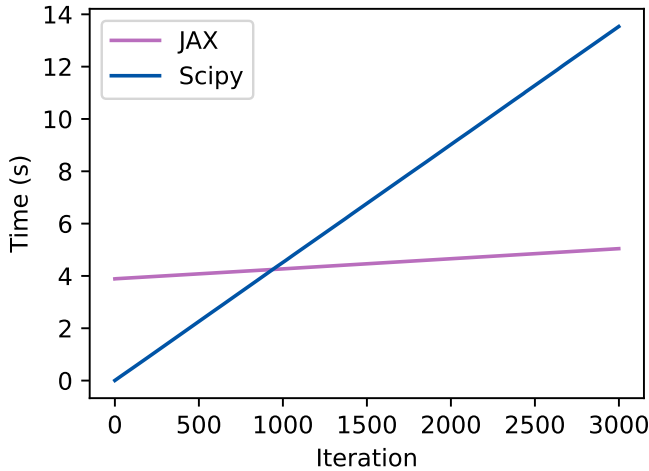
5. Stellar Spectral Grids

Figure 7: Speed comparison between the JAX and SciPy implementations of the GridPolator interpolation backend. Both interpolators use the builtin `VSPEC` grid, with T_{eff} ranging from 2800 to 3300K, and wavelengths from 5-12 μm with $R = 100$. SciPy is most efficient for a small number of evaluations, with a constant cost per call. For a large number of evaluations, however, the speed gained via JAX's JIT compilation offsets the cost of performing that compilation.

`VSPEC` utilizes a library of pre-computed stellar models for producing its multi-component composite stellar spectrum; detailed stellar spectral models are computationally expensive to generate, so it is more efficient to interpolate over a discretized grid. This interpolation is handled by the VSPEC-related Python package GridPolator, whose main class `GridSpectra` acts as a wrapper for the SciPy (Virtanen et al., 2020) or JAX (Bradbury et al.,

2018) `RegularGridInterpolator` with a specialized API. This setup accomodates spectral grids of arbitrary dimension, and treats the wavelength axis independently of the other parameters so that each wavelength channel is independent. When a `Gridspectra` instance's `evaluate()` method is called, it creates a `RegularGridInterpolator` instance for each wavelength channel, evaluates that interpolator, and then resamples the returned spectrum at the requested wavelength points. In the case that the JAX implementation is used, the portion of `evaluate()` that does the interpolation is just-in-time (JIT) compiled to dramatically improve performance on subsequent calls. Figure 7 compares the computational cost of JAX with that of SciPy.

GridPolator grids can be built arbitrarily by the user, but the package comes with a built-in custom grid of `PHOENIX` stellar models (Allard et al., 1994; Hauschildt et al., 1999; Husser et al., 2013) created specifically for `VSPEC`. Each spectrum spans from 0.1 to 20 microns in steps of 5×10^{-6} microns. The stellar atmospheres used to create the spectra have solar metallicity and a $\log g$ of 5.0, with effective temperatures ranging from 2300K to 3900K in steps of 100K. This is the only grid currently supported by `VSPEC`, with support for additional grids planned in future releases. When the grid is initialized, GridPolator looks for the relevant data files on the user's local system. If they are not present, then the files are downloaded and stored in `~/.gridpolator/grids/`.

Within this suite of codes, GridPolator has the added responsibility of binning spectra and producing wavelength axes that match those output by PSG. Wavelength sampling is done using a constant resolving power $R = \lambda/\Delta\lambda$. The binning algorithm takes a pair of high-resolution flux and wavelength arrays and resamples the flux to a lower-resolution wavelength axis. PSG specifies the central wavelength of each channel, and so this algorithm defines bin edges to be $\lambda_{\text{cen}} \pm \lambda_{\text{cen}}/R$.

6. Simplified Climate Model

`VSPEC` allows users to run simulations with any climate model they desire (so long as it can be converted into the `libpypsg.PyGCM` format) and even comes with utilities to work with PSG/GlobES binary files, ExoCAM (Wolf et al., 2022), WACCM (Marsh et al., 2013), and ExoPlaSim (Paradise et al., 2022) without any custom code from the user.

However, as a full global circulation model can take days to months to converge, there are many cases where a "quick and dirty" 3D climate model may be sufficient. Built into `VSPEC` is such a "faux" climate model – one that does not contain any of the circulation physics of the above sophisticated codes, but instead constructs a 3D planetary atmosphere in a fraction of a second from just a few parameters. Such a model can also be advantageous because it is extremely predictable; a feature in the phase curve can easily be traced back to the model parameters that cause it.
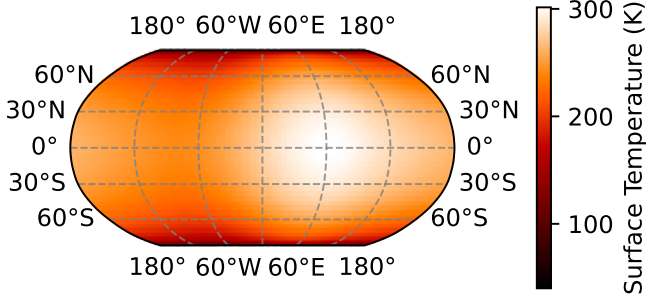
11

Figure 8: Example of a surface temperature map created by the VSPEC GCM, using values for the solar flux equivalent to Earth, a Bond albedo of 0.3, and $\epsilon = 2\pi$. Notice that the hottest point is offset from the sub-stellar point due to thermal inertia.

The VSPEC GCM centers around a 2D temperature map of the planet's surface. Based on Cowan and Agol (2011), this map is determined exclusively by the incident flux from the star (parameterized by the host T_{eff} and radius, and the planet's semimajor axis and Bond albedo) and the unitless thermal redistribution efficiency ϵ. This efficiency is 0 for a planet that re-radiates instantly and $\gg 1$ for an atmosphere that efficiently redistributes heat to its night side. Figure 8 shows a temperature map example for an Earth-like planet with $\epsilon = 2\pi$. These temperature maps are validated by the energy balance of the planet; VSPEC computes the incident and emergent flux across the planet and issues a warning if they are not in agreement.

To compute the surface temperature we first compute the temperature at the equator as a function of longitude. This is done by integrating Cowan and Agol (2011) Equation (10).

$$\frac{d\tilde{T}}{d\Phi} = \frac{1}{\epsilon}(\max(\cos\Phi, 0) - \tilde{T}^4) \qquad (15)$$

Where $\tilde{T}$ is the dimensionless temperature defined as T/T_0, or the temperature normalized by a local fiducial value. Φ is the angle from the substellar point, so we numerically integrate it from $-\pi$ to π using one of several different schemes depending on the value of ϵ. For $0 < \epsilon < 1$ we adopt a reflecting initial value problem (IVP) scheme, where we set $\tilde{T} = 1$ at $\Phi = \pi$, integrate to $-\pi$, then use the last value as the initial value for an integration from $-\pi$ to π. Note that for extremely small values of ϵ (e.g. 10^{-6}), the integration is slow, and will fail if $\epsilon = 0$; we recommended a lower limit of $\epsilon = 10^{-4}$. When $1 \leq \epsilon < 10$ we solve the integral using a boundary value problem (BVP) scheme. For each of these two methods we employ the SciPy (Virtanen et al., 2020) implementation: solve_ivp(), employing a 5th order Runga-Kutta

scheme (Dormand and Prince, 1980) and solve_bvp(), based on an earlier MATLAB implementation (Kierzenka and Shampine, 2001), respectively. When $\epsilon \geq 10$ it is more efficient to use the analytic approximation given by Equation (A2) in Cowan and Agol (2011). In Figure 9 we show results for several different values of ϵ.

6.1. Extending to 3 Dimensions

The treatment by Cowan and Agol (2011) includes no mixing in the latitudinal direction, nor a vertical temperature profile. To allow for variations in pressure and inclination, we augment their treatment with several simple approximations for latitudinal and vertical temperature variations. However, it should be noted that these assumptions produce a discontinuity at the poles, which is non-physical. A more physically-motivated model of latitudinal mixing (e.g., including a diffusion parameter in the heat transport equation) would avoid this caveat, but would also be more expensive to produce.

We add a parameter α that is defined by the ratio between the polar temperature and the equatorial temperature at any given longitude. We define $f(\lambda; \alpha)$ to be the temperature as a function of latitude λ at an arbitrary longitude; therefore if the surface is in equilibrium, the energy balance requires the outgoing energy flux to be independent of α:

$$E_{out} = \sigma \int_0^{\pi/2} f(\lambda; \alpha)^4 \cos\lambda \, d\lambda \qquad (16)$$

where E_{out} is the outbound energy flux of an infinitesimal sliver of differential longitude. In the case of $\alpha = 0$ (i.e. no mixing) we can treat each latitude value independently, so the equilibrium temperature is

$$f(\lambda; 0) = f(0; 0) \cos^{1/4}\lambda \qquad (17)$$

where $f(0; 0)$ is the previously computed temperature at the equator (see Figure 9). Plugging this into Equation (16), we see

$$\begin{aligned} E_{out} &= \sigma \int_0^{\pi/2} (f(0; 0) \cos^{1/4}\lambda)^4 \cos\lambda \, d\lambda \\ &= \sigma f^4(0; 0) \int_0^{\pi/2} \cos^2\lambda \, d\lambda \\ &= \frac{\pi}{4}\sigma f^4(0; 0) \end{aligned} \qquad (18)$$

If instead we let $f(\lambda; 1)$ be a constant function where the equator and pole have equal temperatures, then

$$\begin{aligned} E_{out} &= \sigma \int_0^{\pi/2} f(\lambda; 1)^4 \cos\lambda \, d\lambda \\ &= \sigma f^4(0; 1) \int_0^{\pi/2} \cos\lambda \, d\lambda \\ &= \sigma f^4(0; 1) \end{aligned} \qquad (19)$$

We calculate $f(0, 1)$ by setting Equations (18 & 19) equal to find

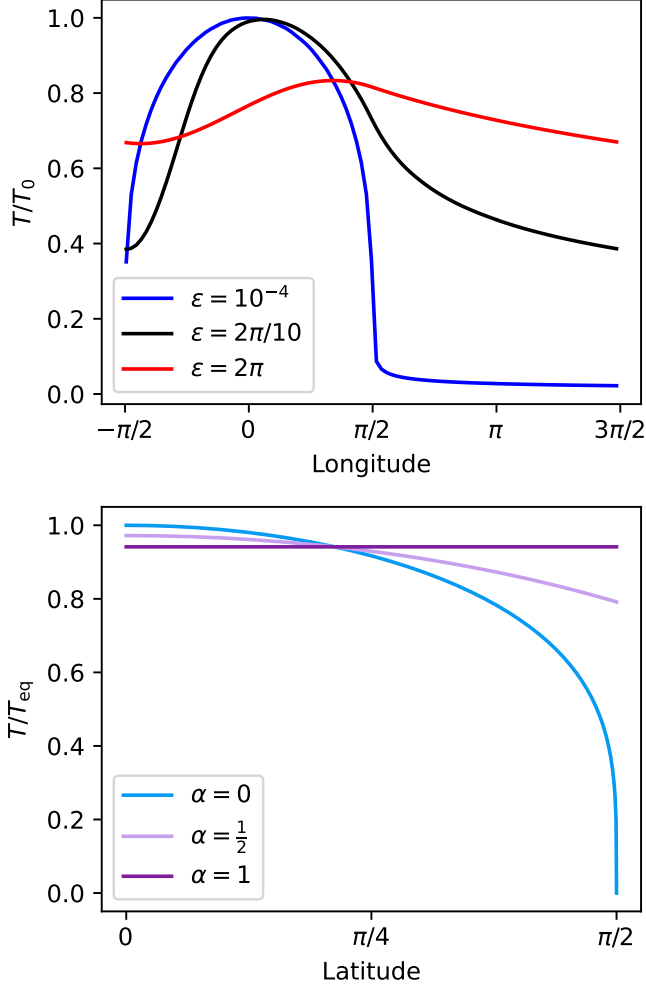$$f(0; 1) = \left(\frac{\pi}{4}\right)^{1/4} f(0; 0) \approx 0.94 \, f(0; 0) \qquad (20)$$
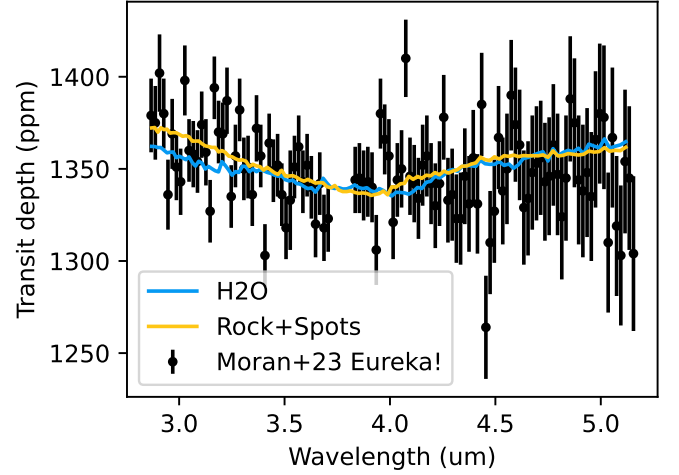
Figure 10: The results from our spotted transit experiment. We find that the addition of spots to a star's surface can produce spectral features in what would otherwise be a flat transit spectrum due to a lack of atmosphere or other mitigating issues such as planetary aerosols. These features mimic those that would be produced due to molecular absorption in a planetary atmosphere. This is in agreement with the results of Moran et al. (2023).

We can now write a general parameterization for the temperature as a function of latitude

$$f(\lambda; \alpha) = f(0; 0) \left(\alpha \frac{\pi}{4} + (1 - \alpha) \cos \lambda \right)^{1/4} \tag{21}$$

Combining Equation (21) with the numerically-computed equatorial temperatures yields a 2D temperature map that covers the surface of the planet. We construct a 3D atmosphere on top of this surface assuming the atmosphere is adiabatic with index γ. The user supplies both the adiabatic index and the surface pressure P_{surf}. The temperature at some altitude z is given by

$$T_z = T_{\text{surf}} \left(\frac{P_z}{P_{\text{surf}}} \right)^{1 - \frac{1}{\gamma}} \tag{22}$$

Prescribing the temperature profile via Equation (22) is, of course, an oversimplification. The profile does not consider the energy transport between layers of the atmosphere, and will violate the energy balance of the planet. We emphasize that this model is meant to approximate an atmosphere quickly, not be a self-consistent climate model.

Finally, we populate our atmosphere with molecules. The user specifies the species and abundance, and the atmosphere is filled accordingly with a constant molecular suite.

Figure 9: **Top**: Reproduction of Cowan and Agol (2011) Figure 1 using the methods described in the text. These curves show the dimensionless temperature at the equator of a planet as a function of angle from substellar point Φ for thermal inertia ϵ. Note that the $\epsilon = 0$ curve in the original figure is replaced with $\epsilon = 10^{-4}$ (see explanation in the text), and we do not include additional curves for mid-latitudes as in the original figure. As ϵ increases, heat is moved from the dayside onto the nightside and the temperature maximum shifts away from the substellar point. **Bottom**: Relative surface temperature as a function of latitude λ for varying latitudinal mixing parameter α. A value of 1 means the equatorial temperature, as computed in the upper panel. Without mixing ($\alpha = 0$), heat is concentrated at the equator. However, as α increases, the equator cools and heat is redistributed towards the poles.

7. Examples

7.1. Transit of a spotted star

In this section we will simulate the transit of a atmosphere-less occulting disk across a spotted star to

demonstrate how an inhomogeneous stellar surface can lead to a false atmospheric signal – a phenomenon known as the "transit light source effect" (Rackham et al., 2018).

During primary transit of a planet with an atmosphere, the atmosphere is illuminated by the portion of the star directly behind the planet. Ideally, the depth of the transit should be measured with respect to the spectrum of the region of the stellar surface covered by the planet. However, the disk-integrated stellar spectrum (as measured before an after the transit) is used as the baseline. In the case that the stellar surface is not homogeneous, the transit signal is modulated by the uncorrected stellar inhomogeneities. In this example, we will reproduce the TLS results by Moran et al. (2023), who contended with this degeneracy with JWST/NIRSpec observations during a transit of super-Earth GJ 486b.

Whenever possible, we draw parameter values from Moran et al. (2023) and the NExScI rchive[12]. We use the JWST NIRSpec/G395H instrument setup from PSG with an observing time of 3.53 hours with an 8 minute cadence to match Moran et al. (2023), but set $R = 200$ to account for the resolution of their reduction. For simplicity, we do not include limb darkening or stellar rotation.

We performed two `VSPEC` simulations based on the two scenarios described in Moran et al. (2023):

1. 1 bar H_2O isothermal atmosphere, no spots.
2. No atmosphere, 2.5% coverage by 2700 K spots

In Case 1 we expect to see an absorption spectrum characteristic of H_2O. Case 2 is more complicated, though in the absence of spots we would expect a flat spectrum. Note also that we disabled H_2O-H_2O collision-induced absorption (CIA) for these simulations because the effect was not considered in the original analysis of GJ 486b. This is a simplified and highly controlled experiment, so there is no need to use standard transit light curve fitting techniques. We are able to compute the transit depth as $(f_{pre} - f_{tr})/f_{pre}$, where f_{pre} is the total spectrum before the transit, at an epoch chosen by visual inspection, and f_{tr} is the total spectrum during the transit, again chosen by visual inspection. A light curve fitting code would be needed if we had introduced stellar rotation or limb darkening.

We find that, as expected, the addition of spots changes the transit depth in a way that depends on wavelength – mimicking light that would be lost to absorbers in a planetary atmosphere. Figure 10 shows us that, in this case, we observe false absorption signals; this is due to a temporary increase in the spotted fraction of the surface relative to the quiet photosphere. However, if a spot was blocked by the planet we might see similar features inverted to mimic emission.

7.2. Phase Curve

Analysis of JWST MIRI-LRS phase curves of GJ 1214b by Kempton et al. (2023) did not consider stellar con-
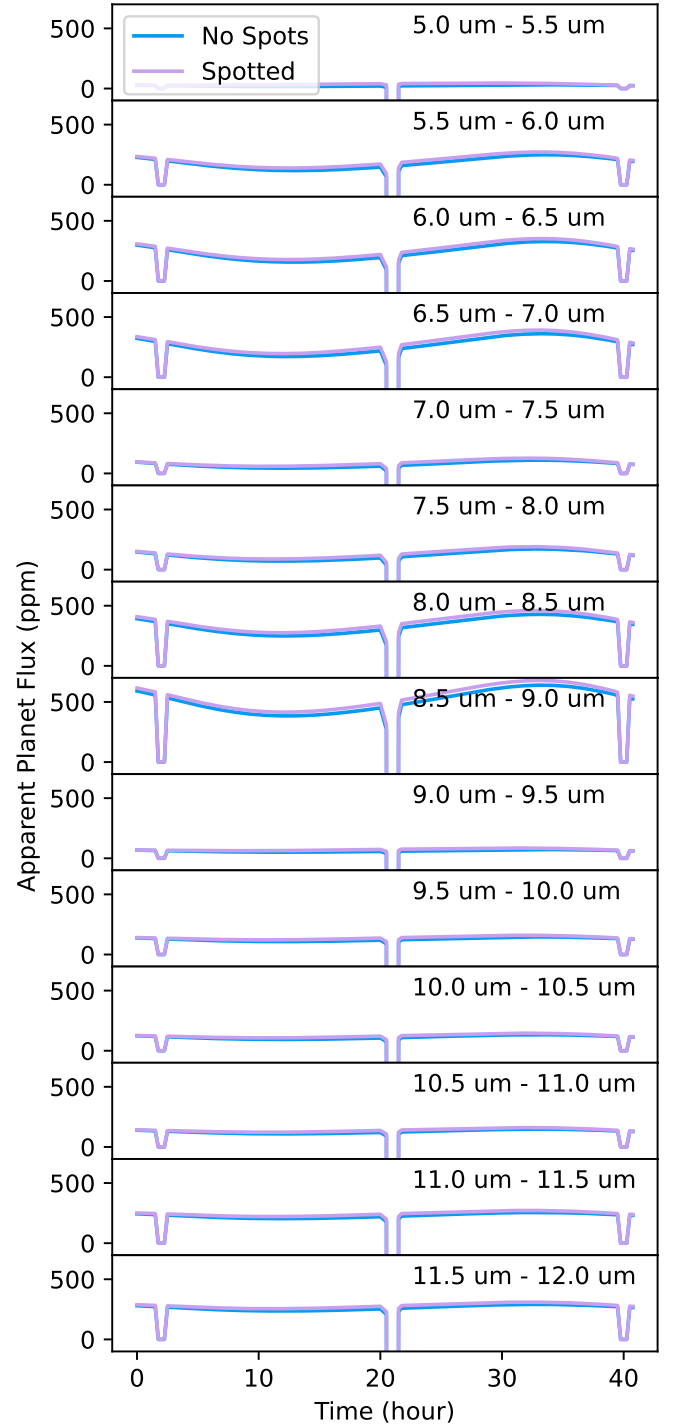
Figure 11: A simulation of a MIRI phase curve for GJ 1214b, binned by wavelength; compare to real data published by Kempton et al. (2023)(Extended Data Figure 1). While the addition of stellar variability does have a visible effect on the light curve, it is much smaller than the thermal emission produced by the exoplanet.

tamination because of the slow rotation rate of the host star (approximately $1/80^{\text{th}}$ the orbital frequency, Cloutier et al., 2021). In this example, we simulate these observations using `VSPEC` and determine whether this is a reasonable assumption.

We initialized `VSPEC` to simulate an observation of the system using the same instrument setup and observation parameters as Kempton et al. (2023). We used stellar and planetary parameters from the NExScI Exoplanet Archive[13]. The VSPEC GCM (see Section 6) used has a thermal inertia of $\epsilon = 6$ and a 1 bar CO_2 atmosphere, appropriate for a high-metallicity CO_2-rich atmosphere scenario.

We initialized two stellar models: one with nothing more than a 3250 K surface, and another with 20% coverage by 2700 K spots. After running the `VSPEC` simulations, we binned the simulated data into 0.5 μm wavelength channels and normalized each channel to the two eclipses, similar to standard data analysis techniques; this assumes that the star behaves linearly between eclipse measurements and therefore attributes any deviation to the planet.

We show in Figure 11 that the stellar properties assumed in this example do not generate enough contamination to significantly change the observed phase curve of GJ 1214b. At worst (i.e. near transit), the observed spectrum varies by 50 ppm due to stellar contamination – compared to the 700 ppm relative flux of the planet at $8.5\mu m$.

Similar studies could be done to assess the potential for stellar contamination in other exoplanetary systems in order to judge fitness as a target; in particular, faster stellar rotation should produce much larger contamination signals that may approach or dwarf the planetary signal.

8. Conclusion

We present a suite of codes to model observations of exoplanets with 3D atmospheres orbiting variable stars. The main package, `VSPEC`, produces combined-light phase curves by interfacing with an ensemble of support packages whose responsibilities range from grid interpolation to interfacing with PSG. These codes are all open-source and available to the public on GitHub and the Python Package Index (PyPI).

While all of these codes are stable, development is ongoing. Future work on `VSPEC` will focus on improving performance so that it can be used to produce phase-curve grids for retrievals. In general, a phase curve retrieval will not require the stellar portion of VSPEC's output, and if there is no need for both thermal and reflected spectra, then only one PSG calculation is required per epoch. This could reduce `VSPEC`'s computational cost by approximately a factor of 4 purely by skipping unnecessary steps.

A future goal of `vspec-vsm` development should be to create a common API for externally-developed physics-based variable star surface models, so that `VSPEC` can interface with those models as well. GridPolator is currently in the process of adding support for more built-in stellar grids, which are more complex and cover a larger range of spectral types than the current `VSPEC` grid. However, as the size of these grids grows, it becomes less efficient to build the grid prior to each run. Future versions will support caching of grids locally using standard Python object serialization libraries such as `pickle`[14]. This method would allow interpolator-oriented Python objects to be recalled nearly instantly (limited by hard drive read-write speeds) rather than the expensive process of re-initialization.

`libpypsg` has the most to offer the exoplanet community as a stand-alone package, but currently lacks support for much of the data that PSG is capable of producing. Its development so far has focused around the `VSPEC` use case, so future work will center around making it more useful to the majority of PSG users.

9. Acknowledgements

10. Author Contributions

Ted M Johnson: Conceptualization, Methodology, Software, Writing - Original Draft **Cameron Kelahan:**

Conceptualization, Methodology, Software, Writing - Review & Editing **Avi M. Mandell:** Conceptualization, Methodology, Writing - Original Draft, Writing - Review & Editing, Supervision, Project administration, Funding acquisition **Ashraf Dhahbi:** Software **Tobi Hammond:** Software **Thomas Barclay:** Methodology, Software **Veselin B. Kostov:** Methodology **Geronimo L. Villanueva:** Resources, Writing - Review & Editing, Supervision

References

Allard, F., Hauschildt, P.H., Miller, S., Tennyson, J., 1994. The Influence of H 2O Line Blanketing on the Spectra of Cool Dwarf Stars. The Astrophysical Journal 426, L39. doi:10.1086/187334.

Anderson, E.R., Duvall, Jr., T.L., Jefferies, S.M., 1990. Modeling of Solar Oscillation Power Spectra. The Astrophysical Journal 364, 699. doi:10.1086/169452.

Apai, D., Rackham, B.V., Giampapa, M.S., Angerhausen, D., Teske, J., Barstow, J., Carone, L., Cegla, H., Domagal-Goldman, S.D., Espinoza, N., Giles, H., Gully-Santiago, M., Haywood, R., Hu, R., Jordan, A., Kreidberg, L., Line, M., Llama, J., López-Morales, M., Marley, M.S., de Wit, J., 2018. Understanding Stellar Contamination in Exoplanet Transmission Spectra as an Essential Step in Small Planet Characterization. doi:10.48550/arXiv.1803.08708.

Astropy Collaboration, Price-Whelan, A.M., Lim, P.L., Earl, N., Starkman, N., Bradley, L., Shupe, D.L., Patil, A.A., Corrales, L., Brasseur, C.E., Nöthe, M., Donath, A., Tollerud, E., Morris, B.M., Ginsburg, A., Vaher, E., Weaver, B.A., Tocknell, J., Jamieson, W., van Kerkwijk, M.H., Robitaille, T.P., Merry, B., Bachetti, M., Günther, H.M., Aldcroft, T.L., Alvarado-Montes, J.A., Archibald, A.M., Bódi, A., Bapat, S., Barentsen, G., Bazán, J., Biswas, M., Boquien, M., Burke, D.J., Cara, D., Cara, M., Conroy, K.E., Conseil, S., Craig, M.W., Cross, R.M., Cruz, K.L., D'Eugenio, F., Dencheva, N., Devillepoix, H.A.R., Dietrich, J.P., Eigenbrot, A.D., Erben, T., Ferreira, L., Foreman-Mackey, D., Fox, R., Freij, N., Garg, S., Geda, R., Glattly, L., Gondhalekar, Y., Gordon, K.D., Grant, D., Greenfield, P., Groener, A.M., Guest, S., Gurovich, S., Handberg, R., Hart, A., Hatfield-Dodds, Z., Homeier, D., Hosseinzadeh, G., Jenness, T., Jones, C.K., Joseph, P., Kalmbach, J.B., Karamehmetoglu, E., Kałuszyński, M., Kelley, M.S.P., Kern, N., Kerzendorf, W.E., Koch, E.W., Kulumani, S., Lee, A., Ly, C., Ma, Z., MacBride, C., Maljaars, J.M., Muna, D., Murphy, N.A., Norman, H., O'Steen, R., Oman, K.A., Pacifici, C., Pascual, S., Pascual-Granado, J., Patil, R.R., Perren, G.I., Pickering, T.E., Rastogi, T., Roulston, B.R., Ryan, D.F., Rykoff, E.S., Sabater, J., Sakurikar, P., Salgado, J., Sanghi, A., Saunders, N., Savchenko, V., Schwardt, L., Seifert-Eckert, M., Shih, A.Y., Jain, A.S., Shukla, G., Sick, J., Simpson, C., Singanamalla, S., Singer, L.P., Singhal, J., Sinha, M., Sipőcz, B.M., Spitler, L.R., Stansby, D., Streicher, O., Šumak, J., Swinbank, J.D., Taranu, D.S., Tewary, N., Tremblay, G.R., de Val-Borro, M., Van Kooten, S.J., Vasović, Z., Verma, S., de Miranda Cardoso, J.V., Williams, P.K.G., Wilson, T.J., Winkel, B., Wood-Vasey, W.M., Xue, R., Yoachim, P., Zhang, C., Zonca, A., Astropy Project Contributors, 2022. The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package. The Astrophysical Journal 935, 167. doi:10.3847/1538-4357/ac7c74.

Astropy Collaboration, Price-Whelan, A.M., Sipőcz, B.M., Günther, H.M., Lim, P.L., Crawford, S.M., Conseil, S., Shupe, D.L., Craig, M.W., Dencheva, N., Ginsburg, A., VanderPlas, J.T., Bradley, L.D., Pérez-Suárez, D., de Val-Borro, M., Aldcroft, T.L., Cruz, K.L., Robitaille, T.P., Tollerud, E.J., Ardelean, C., Babej, T., Bach, Y.P., Bachetti, M., Bakanov, A.V., Bamford, S.P., Barentsen, G., Barmby, P., Baumbach, A., Berry, K.L., Biscani, F., Boquien, M., Bostroem, K.A., Bouma, L.G., Brammer, G.B., Bray, E.M., Breytenbach, H., Buddelmeijer, H., Burke, D.J.,

Calderone, G., Cano Rodríguez, J.L., Cara, M., Cardoso, J.V.M., Cheedella, S., Copin, Y., Corrales, L., Crichton, D., D'Avella, D., Deil, C., Depagne, É., Dietrich, J.P., Donath, A., Droettboom, M., Earl, N., Erben, T., Fabbro, S., Ferreira, L.A., Finethy, T., Fox, R.T., Garrison, L.H., Gibbons, S.L.J., Goldstein, D.A., Gommers, R., Greco, J.P., Greenfield, P., Groener, A.M., Grollier, F., Hagen, A., Hirst, P., Homeier, D., Horton, A.J., Hosseinzadeh, G., Hu, L., Hunkeler, J.S., Ivezić, Ž., Jain, A., Jenness, T., Kanarek, G., Kendrew, S., Kern, N.S., Kerzendorf, W.E., Khvalko, A., King, J., Kirkby, D., Kulkarni, A.M., Kumar, A., Lee, A., Lenz, D., Littlefair, S.P., Ma, Z., Macleod, D.M., Mastropietro, M., McCully, C., Montagnac, S., Morris, B.M., Mueller, M., Mumford, S.J., Muna, D., Murphy, N.A., Nelson, S., Nguyen, G.H., Ninan, J.P., Nöthe, M., Ogaz, S., Oh, S., Parejko, J.K., Parley, N., Pascual, S., Patil, R., Patil, A.A., Plunkett, A.L., Prochaska, J.X., Rastogi, T., Reddy Janga, V., Sabater, J., Sakurikar, P., Seifert, M., Sherbert, L.E., Sherwood-Taylor, H., Shih, A.Y., Sick, J., Silbiger, M.T., Singanamalla, S., Singer, L.P., Sladen, P.H., Sooley, K.A., Sornarajah, S., Streicher, O., Teuben, P., Thomas, S.W., Tremblay, G.R., Turner, J.E.H., Terrón, V., van Kerkwijk, M.H., de la Vega, A., Watkins, L.L., Weaver, B.A., Whitmore, J.B., Woillez, J., Zabalza, V., Astropy Contributors, 2018. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. The Astronomical Journal 156, 123. doi:10.3847/1538-3881/aabc4f.

Astropy Collaboration, Robitaille, T.P., Tollerud, E.J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A.M., Kerzendorf, W.E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M.M., Nair, P.H., Unther, H.M., Deil, C., Woillez, J., Conseil, S., Kramer, R., Turner, J.E.H., Singer, L., Fox, R., Weaver, B.A., Zabalza, V., Edwards, Z.I., Azalee Bostroem, K., Burke, D.J., Casey, A.R., Crawford, S.M., Dencheva, N., Ely, J., Jenness, T., Labrie, K., Lim, P.L., Pierfederici, F., Pontzen, A., Ptak, A., Refsdal, B., Servillat, M., Streicher, O., 2013. Astropy: A community Python package for astronomy. Astronomy and Astrophysics 558, A33. doi:10.1051/0004-6361/201322068.

Barclay, T., Emilygilbert, 2020. Mrtommyb/xoflares 0.2.1. Zenodo doi:10.5281/zenodo.4156285.

Barclay, T., Kostov, V.B., Colón, K.D., Quintana, E.V., Schlieder, J.E., Louie, D.R., Gilbert, E.A., Mullally, S.E., 2021. Stellar Surface Inhomogeneities as a Potential Source of the Atmospheric Signal Detected in the K2-18b Transmission Spectrum. The Astronomical Journal 162, 300. doi:10.3847/1538-3881/ac2824.

Barclay, T., Sheppard, K.B., Latouf, N., Mandell, A.M., Quintana, E.V., Gilbert, E.A., Liuzzi, G., Villanueva, G.L., Arney, G., Brande, J., Colón, K.D., Covone, G., Crossfield, I.J.M., Damiano, M., Domagal-Goldman, S.D., Fauchez, T.J., Fiscale, S., Gallo, F., Hedges, C.L., Hu, R., Kite, E.S., Koll, D., Kopparapu, R.K., Kostov, V.B., Kreidberg, L., Lopez, E.D., Mang, J., Morley, C.V., Mullally, F., Mullally, S.E., Pidhorodetska, D., Schlieder, J.E., Vega, L.D., Youngblood, A., Zieba, S., 2023. The transmission spectrum of the potentially rocky planet L 98-59 c. doi:10.48550/arXiv.2301.10866.

Bogdan, T.J., Gilman, P.A., Lerche, I., Howard, R., 1988. Distribution of Sunspot Umbral Areas: 1917–1982. The Astrophysical Journal 327, 451. doi:10.1086/166206.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., Vander{P}las, J., Wanderman-{M}ilne, S., Zhang, Q., 2018. {JAX}: Composable transformations of {P}ython+{N}um{P}y programs.

Cadieux, C., Doyon, R., MacDonald, R.J., Turbet, M., Artigau, É., Lim, O., Radica, M., Fauchez, T.J., Salhi, S., Dang, L., Albert, L., Coulombe, L.P., Cowan, N.B., Lafrenière, D., L'Heureux, A., Piaulet-Ghorayeb, C., Benneke, B., Cloutier, R., Charnay, B., Cook, N.J., Fournier-Tondreau, M., Plotnykov, M., Valencia, D., 2024. Transmission Spectroscopy of the Habitable Zone Exoplanet LHS 1140 b with JWST/NIRISS. The Astrophysical Journal 970, L2. doi:10.3847/2041-8213/ad5afa.

Cloutier, R., Charbonneau, D., Deming, D., Bonfils, X., Astudillo-Defru, N., 2021. A More Precise Mass for GJ 1214 b and the

Frequency of Multiplanet Systems Around Mid-M Dwarfs. The Astronomical Journal 162, 174. doi:10.3847/1538-3881/ac1584.

Cowan, N.B., Agol, E., 2011. A Model for Thermal Phase Variations of Circular and Eccentric Exoplanets. The Astrophysical Journal 726, 82. doi:10.1088/0004-637X/726/2/82.

Davenport, J.R.A., 2016. The Kepler Catalog of Stellar Flares. The Astrophysical Journal 829, 23. doi:10.3847/0004-637X/829/1/23.

Dormand, J., Prince, P., 1980. A family of embedded Runge-Kutta formulae. Journal of Computational and Applied Mathematics 6, 19–26. doi:10.1016/0771-050X(80)90013-3.

Espinoza, N., Jordán, A., 2015. Limb darkening and exoplanets: Testing stellar model atmospheres and identifying biases in transit parameters. Monthly Notices of the Royal Astronomical Society 450, 1879–1899. doi:10.1093/mnras/stv744.

Fauchez, T., Villanueva, G.L., Kofman, V., Kopparapu, R.K., Suissa, G., 2024. From General Circulation Models (GCMs) to Exoplanet Spectra with the Global Emission Spectra (GlobES). Prep .

Foreman-Mackey, D., Yu, W., Yadav, S., Becker, M.R., Caplar, N., Huppenkothen, D., Killestein, T., Tronsgaard, R., Rashid, T., Schmerler, S., 2024. Dfm/tinygp: The tiniest of Gaussian Process libraries. Zenodo. doi:10.5281/ZENODO.6389737.

Fournier-Tondreau, M., MacDonald, R.J., Radica, M., Lafrenière, D., Welbanks, L., Piaulet, C., Coulombe, L.P., Allart, R., Morel, K., Artigau, É., Albert, L., Lim, O., Doyon, R., Benneke, B., Rowe, J.F., Darveau-Bernier, A., Cowan, N.B., Lewis, N.K., Cook, N.J., Flagg, L., Genest, F., Pelletier, S., Johnstone, D., Dang, L., Kaltenegger, L., Taylor, J., Turner, J.D., 2024. Near-infrared transmission spectroscopy of HAT-P-18 b with NIRISS: Disentangling planetary and stellar features in the era of JWST. Monthly Notices of the Royal Astronomical Society 528, 3354–3377. doi:10.1093/mnras/stad3813.

Gao, D.Y., Liu, H.G., Yang, M., Zhou, J.L., 2022. Correcting Stellar Flare Frequency Distributions Detected by TESS and Kepler. The Astronomical Journal 164, 213. doi:10.3847/1538-3881/ac937e.

Garcia, L.J., Moran, S.E., Rackham, B.V., Wakeford, H.R., Gillon, M., de Wit, J., Lewis, N.K., 2022. HST/WFC3 transmission spectroscopy of the cold rocky planet TRAPPIST-1h. Astronomy and Astrophysics 665, A19. doi:10.1051/0004-6361/202142603.

Gordon, T.A., Agol, E., Foreman-Mackey, D., 2020. A Fast, Two-dimensional Gaussian Process Method Based on Celerite: Applications to Transiting Exoplanet Discovery and Characterization. The Astronomical Journal 160, 240. doi:10.3847/1538-3881/abbc16.

Harris, C.R., Millman, K.J., Van Der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., Van Kerkwijk, M.H., Brett, M., Haldane, A., Del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. doi:10.1038/s41586-020-2649-2.

Hauschildt, P.H., Allard, F., Baron, E., 1999. The NextGen Model Atmosphere Grid for 3000<=Teff<=10,000 K. The Astrophysical Journal 512, 377–385. doi:10.1086/306745.

Hovis-Afflerbach, B., Pesnell, W.D., 2022. Two New Methods for Counting and Tracking the Evolution of Polar Faculae. Solar Physics 297, 48. doi:10.1007/s11207-022-01977-8.

Hunter, J.D., 2007. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering 9, 90–95. doi:10.1109/MCSE.2007.55.

Husser, T.O., Wende-von Berg, S., Dreizler, S., Homeier, D., Reiners, A., Barman, T., Hauschildt, P.H., 2013. A new extensive library of PHOENIX stellar atmospheres and synthetic spectra. Astronomy and Astrophysics 553, A6. doi:10.1051/0004-6361/201219058.

Kallinger, T., De Ridder, J., Hekker, S., Mathur, S., Mosser, B., Gruberbauer, M., García, R.A., Karoff, C., Ballot, J., 2014. The connection between stellar granulation and oscillation as seen by the Kepler mission. Astronomy and Astrophysics 570, A41. doi:10.1051/0004-6361/201424313.

Kempton, E.M.R., Zhang, M., Bean, J.L., Steinrueck, M.E., Piette, A.A.A., Parmentier, V., Malsky, I., Roman, M.T., Rauscher, E., Gao, P., Bell, T.J., Xue, Q., Taylor, J., Savel, A.B., Arnold, K.E.,

Nixon, M.C., Stevenson, K.B., Mansfield, M., Kendrew, S., Zieba, S., Ducrot, E., Dyrek, A., Lagage, P.O., Stassun, K.G., Henry, G.W., Barman, T., Lupu, R., Malik, M., Kataria, T., Ih, J., Fu, G., Welbanks, L., McGill, P., 2023. A reflective, metal-rich atmosphere for GJ 1214b from its JWST phase curve. Nature 620, 67–71. doi:10.1038/s41586-023-06159-5.

Kierzenka, J., Shampine, L.F., 2001. A BVP solver based on residual control and the Maltab PSE. ACM Transactions on Mathematical Software 27, 299–316. doi:10.1145/502800.502801.

Kofman, V., Villanueva, G.L., Fauchez, T., Mandell, A.M., Johnson, T.M., Payne, A., Latouf, N., Kelkar, S., 2024. The pale blue dot: Using the Planetary Spectrum Generator to simulate signals from hyper realistic exo-Earths. Accepted .

Lim, O., Benneke, B., Doyon, R., MacDonald, R.J., Piaulet, C., Artigau, É., Coulombe, L.P., Radica, M., L'Heureux, A., Albert, L., Rackham, B.V., de Wit, J., Salhi, S., Roy, P.A., Flagg, L., Fournier-Tondreau, M., Taylor, J., Cook, N.J., Lafrenière, D., Cowan, N.B., Kaltenegger, L., Rowe, J.F., Espinoza, N., Dang, L., Darveau-Bernier, A., 2023. Atmospheric Reconnaissance of TRAPPIST-1 b with JWST/NIRISS: Evidence for Strong Stellar Contamination in the Transmission Spectra. The Astrophysical Journal 955, L22. doi:10.3847/2041-8213/acf7c4.

Magic, Z., Asplund, M., 2014. The Stagger-grid: A grid of 3D stellar atmosphere models - VI. Surface appearance of stellar granulation. arXiv:1405.7628.

Mandal, S., Karak, B.B., Banerjee, D., 2017. Latitude Distribution of Sunspots: Analysis Using Sunspot Data and a Dynamo Model. The Astrophysical Journal 851, 70. doi:10.3847/1538-4357/aa97dc.

Marsh, D.R., Mills, M.J., Kinnison, D.E., Lamarque, J.F., Calvo, N., Polvani, L.M., 2013. Climate Change from 1850 to 2005 Simulated in CESM1(WACCM). Journal of Climate 26, 7372–7391. doi:10.1175/JCLI-D-12-00558.1.

May, E.M., MacDonald, R.J., Bennett, K.A., Moran, S.E., Wakeford, H.R., Peacock, S., Lustig-Yaeger, J., Highland, A.N., Stevenson, K.B., Sing, D.K., Mayorga, L.C., Batalha, N.E., Kirk, J., López-Morales, M., Valenti, J.A., Alam, M.K., Alderson, L., Fu, G., Gonzalez-Quiles, J., Lothringer, J.D., Rustamkulov, Z., Sotzen, K.S., 2023. Double Trouble: Two Transits of the Super-Earth GJ 1132 b Observed with JWST NIRSpec G395H. The Astrophysical Journal 959, L9. doi:10.3847/2041-8213/ad054f.

Met Office, 2010/2015. Cartopy: A cartographic python library with a Matplotlib interface.

Moran, S.E., Stevenson, K.B., Sing, D.K., MacDonald, R.J., Kirk, J., Lustig-Yaeger, J., Peacock, S., Mayorga, L.C., Bennett, K.A., López-Morales, M., May, E.M., Rustamkulov, Z., Valenti, J.A., Adams Redai, J.I., Alam, M.K., Batalha, N.E., Fu, G., Gonzalez-Quiles, J., Highland, A.N., Kruse, E., Lothringer, J.D., Ortiz Ceballos, K.N., Sotzen, K.S., Wakeford, H.R., 2023. High Tide or Riptide on the Cosmic Shoreline? A Water-rich Atmosphere or Stellar Contamination for the Warm Super-Earth GJ 486b from JWST Observations. The Astrophysical Journal 948, L11. doi:10.3847/2041-8213/accb9c.

Paradise, A., Macdonald, E., Menou, K., Lee, C., Fan, B.L., 2022. ExoPlaSim: Extending the Planet Simulator for exoplanets. Monthly Notices of the Royal Astronomical Society 511, 3272–3303. doi:10.1093/mnras/stac172.

Rackham, B.V., Apai, D., Giampapa, M.S., 2018. The Transit Light Source Effect: False Spectral Features and Incorrect Densities for M-dwarf Transiting Planets. The Astrophysical Journal 853, 122. doi:10.3847/1538-4357/aaa08c.

Renaud, J.P., Lopez, E., Brande, J., Cruz-Arce, C.E., Kelahan, C., Susemiehl, N., Cristy, D., Hostetter, C., Moore, M.D., Patel, A., Mandell, A.M., 2022. The Exoplanet Modeling and Analysis Center at NASA Goddard. Research Notes of the American Astronomical Society 6, 185. doi:10.3847/2515-5172/ac9060.

Solanki, S.K., 2003. Sunspots: An overview. Astronomy and Astrophysics Review 11, 153–286. doi:10.1007/s00159-003-0018-4.

Spruit, H.C., 1976. Pressure equilibrium and energy balance of small photospheric fluxtubes. Solar Physics 50, 269–295. doi:10.1007/BF00155292.

Topka, K.P., Tarbell, T.D., Title, A.M., 1997. Properties of the Smallest Solar Magnetic Elements. II. Observations versus Hot Wall Models of Faculae. The Astrophysical Journal 484, 479–486. doi:`10.1086/304295`.

Villanueva, G.L., Liuzzi, G., Faggi, S., Protopapa, S., Kofman, V., Fauchez, T., Stone, S.W., Mandell, A.M., 2022. Fundamentals of the Planetary Spectrum Generator.

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

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Van Der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., Van Mulbregt, P., SciPy 1.0 Contributors, Vijaykumar, A., Bardelli, A.P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C.N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D.A., Hagen, D.R., Pasechnik, D.V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G.A., Ingold, G.L., Allen, G.E., Lee, G.R., Audren, H., Probst, I., Dietrich, J.P., Silterra, J., Webber, J.T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J.L., De Miranda Cardoso, J.V., Reimer, J., Harrington, J., Rodríguez, J.L.C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N.J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P.A., Lee, P., McGibbon, R.T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T.J., Robitaille, T.P., Spura, T., Jones, T.R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y.O., Vázquez-Baeza, Y., 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods 17, 261–272. doi:`10.1038/s41592-019-0686-2`.

Weisstein, E.W., 2004. Circle-Circle Intersection. https://mathworld.wolfram.com/Circle-CircleIntersection.html.

Wolf, E.T., Kopparapu, R., Haqq-Misra, J., Fauchez, T.J., 2022. ExoCAM: A 3D Climate Model for Exoplanet Atmospheres. The Planetary Science Journal 3, 7. doi:`10.3847/PSJ/ac3f3d`.