Journal of
open research software

Varner, J F et al. 2019 WPTherml: A Python Package for the Design of Materials for Harnessing Heat. *Journal of Open Research Software,* 7: 28. DOI: https://doi.org/10.5334/jors.271

SOFTWARE METAPAPER

WPTherml: A Python Package for the Design of Materials for Harnessing Heat

James F. Varner, Noor Eldabagh, Derek Volta, Reem Eldabagh and Jonathan J. Foley IV
Department of Chemistry, William Paterson University, Wayne, NJ, US
Corresponding author: Jonathan J. Foley IV (foleyj10@wpunj.edu)

WPTherml is a Python package for the design of materials with tailored optical and thermal properties for the vast number of energy applications where control of absorption and emission of radiation, or conversion of heat to radiation or vice versa, is paramount. The optical properties are treated within classical electrodynamics via the Transfer Matrix Method which rigorously solves Maxwell's equations for layered isotropic media. A flexible multilayer class connects rigorous electrodynamics properties to figures of merit for a variety of thermal applications, and facilitates extensions to other applications for greater reuse potential. WPTherml can be accessed at https://github.com/FoleyLab/wptherml.

Keywords: Nanophotonics; Solar Thermophotovoltaics; Passive Cooling; Radiative Cooling; Computational Electrodynamics; Concentrated Solar Power; Transfer Matrix Method

(1) Overview

Introduction

In 2017 over sixty five percent of energy produced was not useable and released as waste heat [1]; consequently, there are a large number of opportunities to develop technologies which can mitigate waste heat associated with energy production, or which can be used to convert or reclaim waste heat for useful purposes. Nano-scale materials, including multi-layered nanostructures among others, with tailored optical and thermal emission properties are centrally important for many of these envisioned technologies, which include solar thermophotovoltaics (STPV) [2–4], radiative cooling devices [5], and highly efficient incandescent lighting [6] that minimize IR losses.

WPTherml (**W**illiam **P**aterson University's tool for **Th**ermal **E**nergy and **R**adiation management with **M**ulti-**L**ayer nanostructures), is a computational engine for materials which can be leveraged for these and other technologies where the control of optical and/or thermal radiation properties is paramount. Specifically, WPTherml can be utilized to simulate optical properties (reflectivity, transmissivity, absorptivity, and emissivity) of nanostructures made from layered isotropic media; thermal radiation spectra of these nanostructures can be derived from these optical quantities in conjunction with Planck's blackbody radiation law. With these spectral quantities in hand, figures of merit for the design of such multi-layer nanostructures for applications such as STPV, passive radiative cooling, concentrated solar power, incandescent lighting, and plasmonics can be computed. Thus, the WPTherml package serves as an engine for computing spectral quantities, and for relating these spectral quantities to relevant figures of merit. Central to the WPTherml package is a *multilayer* class; computation of spectral quantities and figures of merit are performed by methods of the *multilayer* class and are stored as attributes. Spectral quantities are computed within the framework of classical electrodynamics, specifically utilizing the transfer matrix method [7], which is implemented in a library called *tmm.* The required inputs for the transfer matrix method include the geometry (thickness) and refractive index of each layer in the multi-layer nanostructure, so the *multilayer* class has an associated method for parsing user input in an intuitive format to specify the geometry and material composition of a multi-layer structure. Refractive index data for a number of common materials is stored and accessed within a data library called *datalib.* This library also provides access to relevant data for the computation of various figures of merit, for example, spectral response functions of several popular photovoltaic materials (see **Figure 1** for an illustration of the spectral response of InGaAsSb), the AM1.5 solar spectrum, the photopic luminosity function, and the atmospheric transmissivity spectrum can all be accessed using the functionality of *datalib.*

The core of WPTherml is the transfer matrix method, implemented in the *tmm* library, which is a classical electrodynamics approach that relates the incident electromagnetic field upon a multi-layer structure to the reflected and transmitted electromagnetic fields, thereby permitting computation of reflection, transmission, and absorption of light by the multilayer:

$$\begin{pmatrix} 1 \\ r \end{pmatrix} = \begin{pmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{pmatrix} \begin{pmatrix} t \\ 0 \end{pmatrix}, \tag{1}$$

where $M_{i,j}$ are the elements of the transfer matrix, r is the reflection amplitude,

$$r = \frac{M_{2,1}}{M_{1,1}} \tag{2}$$

and t is the transmission amplitude,

$$t = \frac{1}{M_{1,1}}. \tag{3}$$

The reflection and transmission amplitudes may be used to compute the reflectivity, transmissivity, absorptivity, and emissivity of the structure:

$$R = r^* r, \tag{4}$$

$$T = t^* t \frac{n_L \cos(\theta_L)}{n_1 \cos(\theta_1)}, \tag{5}$$

$$A \equiv \epsilon = 1 - R - T, \tag{6}$$

where R denotes the reflectivity, T the transmissivity, A the absorptivity, and ε is the emissivity. In equations (4) and (5), r^* and t^* denote the complex conjugate of the reflection and transmission amplitudes defined in Eq. (2) and (3), respectively. The transmissivity (T) depends on the angle of incidence of the light beam into the multi-layer structure (θ_1) as well as the angle of refraction into the terminal semi-infinite layer (θ_L) as well as the refractive index of the semi-infinite incident material (n_1) and semi-infinite terminal material (n_L). Because the incident and terminal layers are taken to have an infinite extent, they should not be light-absorbing, meaning their refractive indices should be pure real numbers. It should be noted that all quantities above depend on wavelength, and the explicit wavelength dependence of the elements of the transfer matrix have been defined in detail [8]. The core transfer matrix library has been utilized by one of the authors to model thermal emission of multi-layer planar nanostructures [4], and the optical properties of the 2D material known as Borophene [9].

In Equation (6), an important relation known as Kirchoff's law [2–4] is invoked, which equates the absorption spectrum of a multi-layer structure to its emissivity spectrum. The emissivity spectrum plays a key role in shaping the thermal emission spectrum of a structure at a given temperature,

$$TE(\lambda, T_{ml}) = \rho(\lambda, T_{ml}) \epsilon(\lambda), \tag{7}$$

where $TE(\lambda, T_{ml})$ denotes the thermal emission, and

$$\rho(\lambda, T_{ml}) = \frac{2hc^2}{\lambda^5} \frac{1}{\exp\left(\frac{hc}{\lambda k_B T_{ml}}\right) - 1} \tag{8}$$

is Planck's blackbody law, where T_{ml} denotes the temperature of the multi-layer structure, λ denotes the wavelength of light, h is Planck's constant, c is the speed of light, and k_B is Boltzmann's constant. In the following, we will illustrate the particular application of WPTherml to solar thermophotovoltaics (STPV), where one tailors the thermal emission spectrum of an object through its emissivity such that it is dramatically different from an ideal blackbody emitter, which has an emissivity of 1 for all wavelengths. Following that discussion, we will briefly summarize several other applications for which WPTherml has been specifically designed to address.

Example application to STPV

Overview of application

Management of thermal emission as an energy source, as is done in STPV, may have many beneficial impacts in multiple markets including, but not limited to, clean energy, military use, and the consumer market [3]. Here we illustrate the use of WPTherml to simulate the thermal emission of a multi-layer nanostructure optimized for performance as an STPV emitter; this structure is illustrated in **Figure 1(a)**. Briefly, the objective is to tune the thermal emission of a material so that it is well matched to the spectral response of a photovoltaic (PV) material. In **Figure 1(b)**, we illustrate the thermal emission spectrum of the structure when it is heated to 1700 K and overlay it with the spectral response function of an InGaAsSb PV cell, which is commonly used in STPV applications [4], along with the blackbody spectrum at 1700 K. Importantly, the thermal emission of the structure illustrated in **Figure 1(a)** strongly departs from the blackbody spectrum, and has excellent overlap with the spectral response of InGaAsSb. To further illustrate the suitability of this structure for STPV applications, we plot the product of the thermal emission of this structure with the spectral response of InGaAsSb in **Figure 1(c)**. It is in fact the ratio of the area under this curve, relative to the total area under the thermal emission curve, which plays a critical role in the overall conversion efficiency in a given STPV device (see [8] for more details). An example of code used to generate these figures in WPTherml is included below **Figure 1**.

In the above code example, an instance of the *multilayer* class called *fig_1* is created following user input defined in the dictionary called *structure*. The physical structure that this instance models consists of several dielectric layers (aluminium oxide (Al_2O_3), titanium dioxide (TiO_2), and glass (SiO_2)), and an optically-thick tungsten (W) layer, which are defined in the '*Material_List*' key of the *structure* dictionary. Upon instantiation, the optical and thermal emission properties of this structure at 1700 K (specified by '*Temperature*') are computed for 1000 wavelengths between 300 and 6000 nm (specified in '*Lambda_List*') and stored as attributes of *fig_1*. Additionally, because the key '*STPV_EMIT*' is set to the value 1, several STPV figures of merit for this structure, including the spectral efficiency, power density, and the TPV efficiency, will be computed at instantiation and stored as the attributes *fig_1.spectral_efficiency_val*, *fig_1.power_density_val*, *fig_1.tpv_efficiency_val*, respectively. The mathematical details of these figures of merit are discussed in references [4, 8].

After instantiation, a modification to one of the layers is made whereby the layer with index 1 (originally 20 nm thick Al_2O_3) is changed to an alloy of Al_2O_3 and W with a

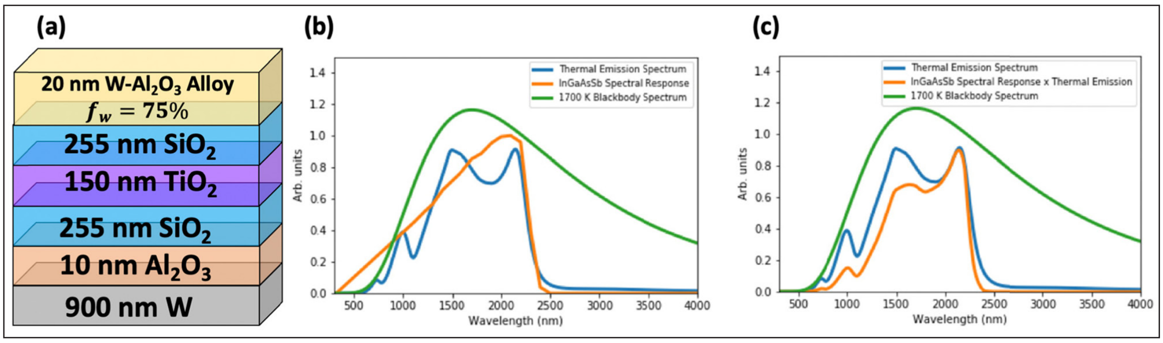

Figure 1: (a) Example of a multi-layer structure that exhibits selective thermal emission that is nearly optimal for the spectral response of InGaAsSb PVs. **(b)** The spectral response (which has SI units of A/W) of InGaAsSb plotted against the thermal emission spectrum of the structure illustrated in (a) and the blackbody spectrum at 1700 K; these thermal emission spectra are divided by an arbitrary number ($5 \cdot 10^{10}$) to aid visualization against the spectral response function. **(c)** Plot of the product of the spectral response function of InGaAsSb with the thermal emission of the structure in (a) plotted against the thermal emission itself; these two functions are the integrands that give rise to the short circuit current and the incident power in a STPV system, respectively. See reference [8] for a more detailed discussion of these quantities and their relation to the STPV efficiency.

Example Code for STPV application

```
from wptherml.wpml import multilayer
from matplotlib import pyplot as plt
from wptherml.datalib import datalib

structure = {
    ‘Temperature’: 1700,
    ‘Material_List’: [‘Air’, ‘Al2O3’, ‘SiO2’, ‘TiO2’, ‘SiO2’, ‘Al2O3’, ‘W’, ‘Air’],
    ‘Thickness_List’: [0, 20e-9, 255e-9, 150e-9, 255e-9, 10e-9, 900e-9, 0],
    ‘Lambda_List’: [300e-9, 6000e-9, 1000],
    ‘STPV_EMIT’: 1
    }
### create instance of multilayer called fig_1
fig_1 = multilayer(structure)
### make layer 1 an alloy of alumina and W where volume fraction
### of W is 0.75 and RI of Al2O3 is taken to be
### 1.66 + 0i for all wavelengths - this will modify multilayer instance “fig_1”
fig_1.layer_alloy(1, 0.75, 1.66+0j, ‘W’, ‘MaxwellGarnett’)
### compute optical properties (reflectivity, transmissivity, emissivity) of modified structure
fig_1.fresnel()
### compute thermal emission of modified structure
fig_1.thermal_emission()

### import spectral response data of InGaAsSb PV from datalib
Spec = datalib.SR_InGaAsSb(fig_1.lambda_array)


### plot scaled thermal emission of multilayer along with spectral response function and blackbody
### spectrum at 1700 K
plt.plot(1e9*fig_1.lambda_array, fig_1.thermal_emission_array/ 5e10, label = “Thermal Emission
Spectrum”, linewidth = 3)
plt.plot(1e9*fig_1.lambda_array, Spec, label = “InGaAsSb Spectral Response”, linewidth = 3)
plt.plot(1e9*fig_1.lambda_array, fig_1.BBs/5e10, label = “1700 K Blackbody Spectrum”,linewidth = 3)
plt.ylim(0, 1.5)
plt.xlim(300,4000)
plt.xlabel(“Wavelength (nm)”)
plt.ylabel(“Arb. units”)
plt.legend(loc = ‘best’, fontsize = “small”)
plt.show()

### plot thermal emission * spectral response function, which relates directly to short
### circuit current of the TPV device
plt.plot(1e9*fig_1.lambda_array, fig_1.thermal_emission_array/ 5e10, label = “Thermal Emission
Spectrum”, linewidth = 3)
plt.plot(1e9*fig_1.lambda_array, Spec*(fig_1.thermal_emission_array/5e10), label = “InGaAsSb Spectral
Response x Thermal Emission”, linewidth = 3)
plt.plot(1e9*fig_1.lambda_array, fig_1.BBs/5e10, label = “1700 K Blackbody Spectrum”, linewidth = 3)
plt.ylim(0, 1.5)
plt.xlim(300,4000)
plt.xlabel(“Wavelength (nm)”)
plt.ylabel(“Arb. units”)
plt.legend(loc = ‘best’, fontsize = “small”)
plt.show()
```

volume fraction of 75% W; note that in this example, we specify a static refractive index of 1.66+0i for the Al_2O_3 layer when making the alloy. Note also that currently this modification **must** be made after instantiation because a separate method (*layer_alloy(layer_index, volume_fraction, material_1, material_2, model)*) must be invoked to model a particular layer as an alloy between two different materials; the original specification of layer 1 as the material Al_2O_3 is arbitrary as any material could have been used as a placeholder for the layer that would be modified to become an alloy. The model used to define the wavelength-dependent refractive index of this alloy is known as the Maxwell-Garnett effective medium theory [4], specified by the argument "*MaxwellGarnett*" (an alternative model known as the Bruggeman model [4] can be invoked with the argument "*Bruggeman*"). Once this modification is made, the optical properties (reflectivity, transmissivity, emissivity) are recomputed by calling the *fresnel()* method, and the thermal emission spectrum is recomputed with these updated quantities by calling the *thermal_emission()* method.

Also illustrated in this example is how our data library *datalib* can be used to extract relevant data (either stored or computed) including the spectral response function of an InGaAsSb PV cell (*datalib.SR_InGaAsSb(array_of_wavelengths)*) and the blackbody spectrum at a specific temperature (*datalib.BB(array_of_wavelengths, temperature)*). The remaining code in the example represents standard usage of the python library *matplotlib*, for which extensive documentation already exists [10]. A full list of methods and attributes associated with the *multilayer* class can be found in the WPTherml README documentation [11].

Brief Summary of Additional Applications

The control of optical and thermal emission properties of materials is critical for a number of applications, and here we briefly summarize some of these applications for which WPTherml provides unique capabilities.

Incandescent light bulbs with efficiencies approaching or exceeding that of LED-based lights can be realized by tailoring the thermal emission of filament materials [6]. In particular, the thermal emission should be high in the visible and low in the IR at standard operating temperatures (~2500 K). The key to this property is to tailor the emissivity of the filament material to have strong overlap with the photopic luminosity function, which describes the sensitivity of the human eye to visible light [6]. A figure of merit for incandescent bulbs, known as the luminous efficiency, is related to the convolution of the thermal emission and the photopic luminosity function [6, 8], and this figure of merit can be computed using the method *luminous_efficiency()* of the *multilayer* class (see [11] for more details).

Daytime radiative cooling is an application where a material is used to passively cool a space through a net negative radiative power flux from the material's surface, thereby eliminating or mitigating the need for the use of air conditioning [5]. The key properties of these materials are that they are highly reflective in the solar spectrum and highly emissive in mid-IR wavelengths, specifically wavelengths where the atmosphere is highly transparent (~8–14 microns). Hence, these materials minimize the positive (warming) radiative power flux associated with solar absorption and absorption of thermal radiation from the atmosphere, and maximize the negative (cooling) radiative power flux associated with the material's thermal emission at its operating temperature. These power fluxes are computed by the method *cooling_power()* in the *multilayer* class. An illustration of the emissivity of such a radiative cooling structure inspired by reference [5] is shown in **Figure 2**, in addition to the example code below which will generate these figures, a further illustration can be found in a Jupyter notebook that ships with our package [12].

The optical properties of thin metal films has been the subject of intensive investigation owing to their ability

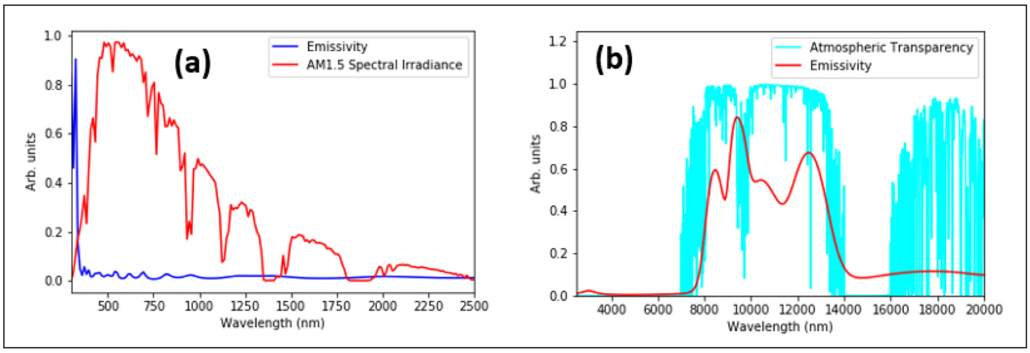

Figure 2: (a) Visible absorptivity/emissivity (blue line) of a multi-layered structure optimized for radiative cooling plotted alongside the AM1.5 solar spectrum, illustrating that this structure minimizes absorption of solar power; the AM1.5 spectrum is divided by an arbitrary number ($1.4 \cdot 10^9$) to aid visualization. **(b)** IR emissivity (red line) of the multi-layered structure optimized for radiative cooling plotted alongside the transmissivity of the earth's atmospheres, illustrating that this structure maximizes ambient thermal radiation and minimizes absorption of thermal radiation from the atmosphere. This simulated structure was inspired by reference [5], and code to reproduce these figures in WPTherml can be found in the *Example Code for radiative cooling application* section of this manuscript and in a Jupyter notebook that ships with our package [12].

Example Code for radiative cooling application

```
from wptherml.wpml import multilayer
from matplotlib import pyplot as plt
from wptherml.datalib import datalib
structure = {
   ‘Material_List’: [‘Air’, ‘SiO2’, ‘HfO2’, ‘SiO2’, ‘HfO2’, ‘SiO2’, ‘HfO2’, ‘SiO2’, ‘Ag’, ‘Air’],
   ‘Thickness_List’: [0, 230e-9, 485e-9, 688e-9, 13e-9, 73e-9, 34e-9, 54e-9, 200e-9, 0],
   ‘Lambda_List’: [300e-9, 60000e-9, 5000],
   ‘EXPLICIT_ANGLE’: 1,
   ‘COOLING’: 1

   }
### create instance of multilayer class called cool_ml
cool_ml = multilayer(structure)
### get AM1.5 spectra and atmospheric transmissivity data from datalib
AM = datalib.AM(cool_ml.lambda_array)
T_atm = datalib.ATData(cool_ml.lambda_array)
### plot emissivity of cool_ml against AM1.5 spectrum in the visible range
### to demonstrate the high solar reflectivity of cool_ml
plt.plot(cool_ml.lambda_array*1e9, cool_ml.emissivity_array, ‘blue’, label=’Emissivity’)
plt.plot(cool_ml.lambda_array*1e9, AM/(1.4*1e9), ‘red’, label=’AM1.5 Spectral Irradiance’)
plt.xlabel(“Wavelength (nm)”)
plt.ylabel(“Arb. units”)
plt.legend(loc = ‘best’)
plt.xlim(300,2500)
plt.show()
### plot the emissivity of cool_ml against the atmospheric transmissivity in the IR
### to demonstrate the selective thermal emissivity of cool_ml
plt.plot(cool_ml.lambda_array*1e9, T_atm, ‘cyan’, label=’Atmospheric Transparency’)
plt.plot(cool_ml.lambda_array*1e9, cool_ml.emissivity_array, ‘red’, label=’Emissivity’)
plt.xlabel(“Wavelength (nm)”)
plt.ylabel(“Arb. units”)
plt.legend(loc = ‘best’)
plt.xlim(2500,20000)
plt.ylim(0,1.25)
plt.show()
```

to support optical resonances known as surface plasmon polaritons [13–15], and perfectly absorbing modes [16–18]. We have included the capability to analyze multilayer structures for both surface plasmon polariton and perfectly absorbing modes through the *find_spp()* and *find_pa()* methods of the *multilayer* class, respectively. Mathematical details of these modes can be found in references [8, 13, 16].

Finally, to aid in the user's intuition for the relation between optical properties, thermal emission, and appearance of a structure, we have implemented methods to render the color of a structure based on its reflectance (*ambient_color()*) or based on its thermal emission (*thermal_color()*). The details behind these methods are discussed in reference [8].

Description of alternative packages

There are several open-source software packages that implement the transfer matrix method to simulate the optical response (for example, reflectivity, transmissivity, absorptivity, emissivity) of multi-layer nanostructures; a partial list includes several python-based transfer matrix method packages [19–23], a cython implementation [24], and a Matlab implementation [25]. Related electrodynamics methods, specifically the S-matrix method, have also been implemented in open-source software packages [26–28]. The authors are currently unaware of any alternative software package that leverages such rigorous electrodynamics techniques for the use in thermal radiation applications; consequently, WPTherml provides a unique and useful capability for a large community of scientists and engineers who aim to design materials and technologies for harnessing heat.

Implementation and architecture

WPTherml is written in pure Python 3 with minimal dependencies including numpy, scipy, and matplotlib. WPTherml has been tested with Anaconda Python 3.6 and 3.7 running on Windows 10, Mac OS Sierra, Mac OS Mojave, and Fedora 22. WPTherml includes a *multilayer* class, several function libraries (*tmm*, *numlib*, *stpvlib*, *coolinglib*, *colorlib*, and *lightlib*), and a data library (*datalib*). The function libraries contain the functions that perform calculations to extract spectral information and figures of merit for the various applications supported by WPTherml as follows:

- *tmm* leverages the transfer matrix method to compute optical properties (reflectivity, transmissivity, absorptivity, emissivity) of multi-layered nanostructures.
- *numlib* provides a simple function for numerical quadrature, which is used to compute a variety of figures of merit.
- *stpvlib* uses optical properties and thermal emission to compute figures of merit relevant to solar

thermophotovoltaic, concentrated solar, and standard photovoltaic applications.

- *coolinglib* uses optical properties and thermal emission to compute figures of merit relevant to radiative cooling.
- *colorlib* uses optical properties and thermal emission to render color of structures, and to classify a spectrum as giving rise to one of the 7 colors in the ROYGBIV spectrum.
- *lightlib* uses optical properties and thermal emission to compute figures of merit relevant for incandescent light sources.
- *datalib* provides data to support various calculations, including:
    - Refractive index as a function of wavelength for an extensive number of common materials; this data is required by the transfer matrix method for computing optical properties.
    - Spectral Response functions for several common photovoltaic materials; this data is required for computing STPV and standard PV efficiencies.
    - AM1.5 spectral data; this data is required for traditional PV efficiency calculations, concentrated solar absorption calculations, and radiative cooling applications.
    - Atmospheric transmission spectrum; this data is required for radiative cooling applications.
    - Spectral response functions for the human eye; this data is required for rendering the colors of structures and for computing the luminous efficiency of incandescent light sources.

The *multilayer* class provides an intuitive framework for connecting user input to the functionality of WPTherml. The user specifies information about the materials, geometry (thickness), and calculation type to the multilayer class through a dictionary whose keys are descriptive of their meaning with respect to the class. The following code illustrates a simple example where a dictionary called *‘structure’* is defined such that when passed to the multilayer class, it specifies the 3-layer structure illustrated in **Figure 3** below:

It is important to note the ordering convention used for the ‘*Material_List*’ and ‘*Thickness_List*’ keys, whose values are lists of strings and floats that specify the materials and thicknesses of each layer in the multi-layer structure, respectively. In particular, these lists are both ordered starting from the side upon which light is incident in a typical reflection experiment and ending in the material into which light would finally be transmitted in the same experiment. Also of note is that both reflected light and thermally emitted light emanate from the top side of structure. All optical and thermal emission spectra will be computed in the range 400–800 nm, with resolution of (800–400)/1000 nm as specified by the value of ‘*Lambda_List*’. A full list of supported keys that can be parsed by the *multilayer* class is provided in the WPTherml README documentation [11].

A schematic of the flow of data between the user, the *multilayer* class, and the data and function libraries is illustrated in **Figure 4** below.

The multilayer class should provide a convenient mechanism for extension of the package to include additional applications, which might require different manipulations of the Fresnel quantities stored as the attributes *self.reflectivity_array, self.emissivity_array, self.transmissivity_array*, or the thermal emission stored as the attribute *self.thermal_emission_array*. Additionally, extension may be made to include different classes of structures (non-planar structures, for example, where the same attributes *self.reflectivity_array,* etc, would be computed by a different method than the transfer matrix method). The typical workflow to extend the capabilities of the package could include

1. Identifying any new properties that will be computed by the extension and adding appropriate attributes to the multilayer class
2. Adding one or more functions to the libraries (*stpvlib,* etc) that manipulate the Fresnel and/or thermal emission quantities as required to compute the new desired property or properties
3. Adding one or more multilayer methods to call the new library functions and store the resulting data in new or existing multilayer attributes as appropriate.

```
structure = {
    # Specify the materials that define the structure here!
    # Note terminal layers are semi-infinte and are air
    # Terminal layers do not necessarily need to be air, but they must be non-absorbing
    ‘Material_List’: [‘Air’, ‘SiO2’, ‘TiO2’, ‘Au’, ‘Air’],
    # Specify the thickness of each layer here!
    # Terminal layers are always given a thickness of 0
    ‘Thickness_List’: [0, 100e-9, 50e-9, 20e-9, 0],
    # Specify the wavelengths over which the optical properties will be computed:
    # here 400e-9 meters (400 nm) is the shortest wavelength
    # 800e-9 meters (800 nm) is the longest wavelength
    # and the optical properties will be computed at 1000 different wavelengths in between
    # 400 – 800 nm
    ‘Lambda_List’: [400e-9, 800e-9, 1000],
    }
### This creates an instance of the multilayer class called coated_au_film.
### By default, this instance will have attributes including the reflectivity,
### transmissivity, and emissivity spectrum at 1000 wavelengths between 400 and 800 nm
### stored as coated_au_film.reflectivity_array, coated_au_film.transmissivity_array, and
### coated_au_film.emissivity_array, respectively
coated_au_film = multilayer(structure)
```

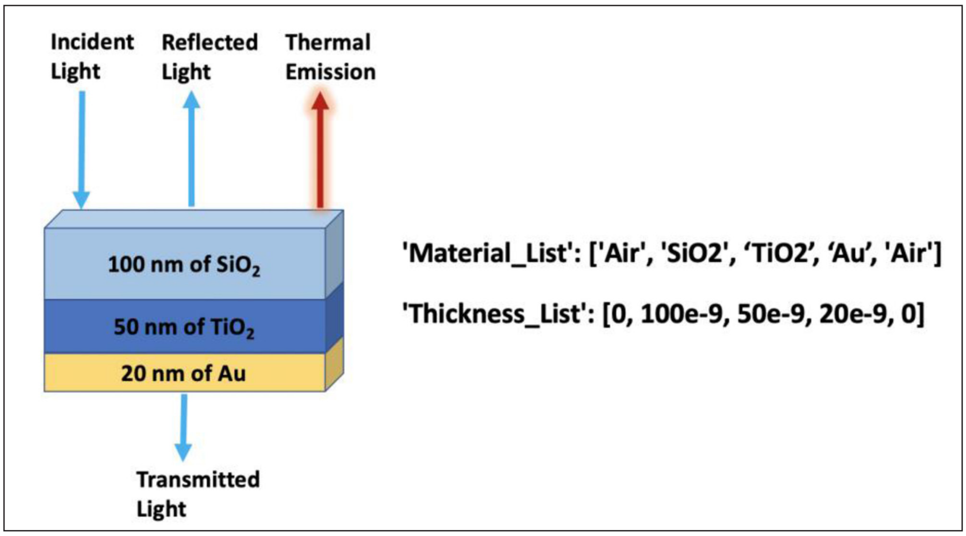

Figure 3: Schematic of the three-layer structure as defined by the 'structure' dictionary above, illustrating the convention for the direction of incoming, reflected, transmitted, and emitted light with respect to the order in which materials and thicknesses are specified by the user.

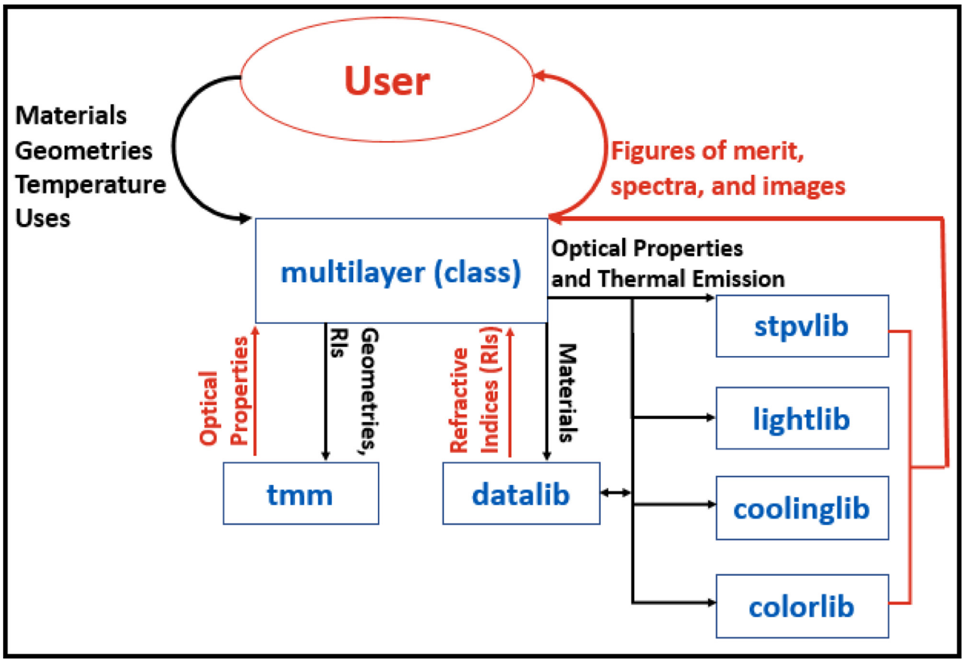

Figure 4: Schematic illustrating the flow of data between the user, the multilayer class, and the data and function libraries of WPTherml. More details can be found in the WPTherml README documentation [11].

Quality control

Unit testing has been performed for all documented methods of the multilayer class, and the user can verify upon installation of the cloned repository by executing the following command from the /test directory:

```
Python3 – m pytest test.py
```

Additionally, several validation examples can be investigated by the user for comparison to results from peer-reviewed literature on several thermal applications. The *tmm* library was validated against the existing python transfer matrix package developed by S. J. Byrnes [19, 20], and calculations using WPTherml that are directly comparable to this package can be run by the user through the Jupyter notebook entitled 'Validate_Fresnel' that ships with our package [29]. This calculation also serves as a validation for the *stpvlib* library against results reported in reference [4]. The user may also run a test of WPTherml that checks the functionality of *coolinglib* through the Jupyter notebook entitled 'Validate_Cooling' that ships with our package [30].

References for *datalib*

Electrodynamics calculations rely upon refractive index data for materials; currently *datalib* parses datasets for the following materials (wptherml keywords for each material are given in parenthesis, references for the datasets in brackets): silver ("Ag") [32, 33], gold ("Au") [32, 34], aluminium ("Al") [35], aluminium nitride ("AlN") [4, 36], aluminium oxide ("Al_2O_3") [4], hafnium oxide ("HfO_2") [4], palladium ("Pd") [37], platinum ("Pt") [37], rhenium ("Re") [37], rhodium ("Rh") [37], ruthenium ("Ru") [37], silicon dioxide ("SiO_2") [4, 38, 39], silicon ("Si") [40], titanium nitride ("TiN") [4], titanium dioxide ("TiO_2") [41], and tungsten ("W") [37]. Additionally, several of the figures of merit implemented depend upon external data, including spectral response functions of GaSb [42] and InGaAsSb [43] photovoltaic cells, the AM1.5 solar spectrum [44], the transmissivity of the atmosphere [45], and the photopic luminosity function [46].

(2) Availability

Operating system

WPTherml has been tested on MacOS Sierra, MacOS Mojave, Windows 10, and Fedora 22. Since WPTherml is written in Python 3, it should run on any system on which Python 3 and the WPTherml dependencies run.

Programming language

WPTherml is written in Python 3 and has been tested with Anaconda Python (3.6 and 3.7).

Additional system requirements

The total size of the WPTherml package source code and supporting data stored in *datalib* is approximately 50 MB; typical execution should not pose any significant memory demands.

Dependencies

WPTherml depends on numpy, scipy, and matplotlib.

The code was tested with

- numpy version 1.14.3
- scipy version 1.1.0
- matplotlib version 2.2.2

List of contributors

James F. Varner; Software development, planning, and testing.
Noor Eldabagh; Software development and testing.
Derek Volta; Software development and testing.
Reem Eldabagh; Software development and testing.
Jonathan J. Foley IV; Project conception, software development, planning, and testing.

Software location

Archive (e.g. institutional repository, general repository) (required – please see instructions on journal website for depositing archive copy of software in a suitable repository)

Name: SourceForge
Persistent identifier: https://sourceforge.net/projects/wptherml
Licence: GPL 3.0
Publisher: Jonathan Foley
Version published: v1.0.13 beta
Date published: 03/22/2019

Code repository

Name: GitHub
Identifier: https://github.com/FoleyLab/wptherml
Licence: GPL 3.0
Date published: 03/04/2019

Language

Python 3

(3) Reuse potential

WPTherml was developed to support the simulation and design of a practically endless number of structures for multiple applications where optical and thermal emission properties are paramount. Such applications include, but are not limited to, solar thermophotovoltaics [2–4, 18], radiative cooling [5], incandescent lighting [6], transparent conductors [8], plasmonic structures [13–15, 17], perfect absorbers [16], thermoelectrics [2], and concentrated solar power [2]. The package was also developed to be easily extensible to support new applications as the developers or users see fit, as discussed in the *Implementation and Architecture* section. Several potential applications are illustrated in Jupyter notebooks that ship with the package, and the developers plan to continue to add and refine these examples over time. Users seeking support for additional features can communicate with us through the issues page of the GitHub repository [31], or may contact the corresponding author by email at foleyj10@wpunj.edu.

Acknowledgements

The authors wish to acknowledge support from the College of Science and Health and the Department of Chemistry at William Paterson University, specifically Dean Kenneth Wolf, Dean Venkat Sharma, Prof. Bhanu Chauhan, and Dr. Mukesh Sahni, for their vision and persistence in realizing the computational infrastructure that allowed the development of WPTherml. J. F. V. acknowledges support from the Louis Stokes Alliance for Minority Participation, and J. J. F. acknowledges support from the ART program at William Paterson University, and from the CFR Minigrant program.

Competing Interests

The authors have no competing interests to declare.

References

1. https://flowcharts.llnl.gov/commodities/energy.
2. **Gupta, M C, Ungaro, C, Foley, J J, IV** and **Gray, S K** 2018 Optical nanostructures design, fabrication, and applications for solar/thermal energy conversion. *Solar Energy*, 165: 100–114. DOI: https://doi.org/10.1016/j.solener.2018.01.010
3. **Zhou, Z, Sakr, E, Sun, Y** and **Bermel, P** 2016 Solar thermophotovoltaics: reshaping the solar spectrum. *Nanophotonics*, 5: 1–21. DOI: https://doi.org/10.1515/nanoph-2016-0011
4. **Jeon, N, Hernandez, J J, Rosenmann, D, Gray, S K, Martinson, A B F** and **Foley, J J, IV** 2018 Pareto Optimal Spectrally Selective Emitters for Thermophotovoltaics

via Weak Absorber Critical Coupling. *Adv. Energy Mater*, 8: 1801035. DOI: https://doi.org/10.1002/aenm.201801035

5. **Raman, A P, Anoma, M A, Zhu, L, Rephaeli, E** and **Fan, S** 2014 Passive radiative cooling below ambient air temperature under direct sunlight. *Nature*, 515: 540–544. DOI: https://doi.org/10.1038/nature13883
6. **Ilic, O, Bermel, P, Chen, G, Joannopoulos, J D, Celanovic, I** and **Soljačić, M** 2016 Tailoring High Temperature Radiation and the Ressurection of the Incandescent Source. *Nat. Nanotechnol*, 11: 320–324. DOI: https://doi.org/10.1038/nnano.2015.309
7. **Yeh, P** 1988 Optical Waves in Layered Media. New York: Wiley.
8. https://github.com/FoleyLab/wptherml/blob/master/documentation/Equations.pdf.
9. **Adamska, L, Sadasivam, S, Foley, J J, IV, Darancet, P** and **Sharifzadeh, S** 2018 First-principles investigation of Borophene as a monolayer transparent conductor. *J. Phys. Chem. C*, 122: 4037–4045. DOI: https://doi.org/10.1021/acs.jpcc.7b10197
10. https://matplotlib.org/contents.html.
11. https://foleylab.github.io/wptherml/.
12. https://github.com/FoleyLab/wptherml/blob/master/Validate_Cooling.ipynb.
13. **Raether, H** 1988 *Surface Plasmons on Smooth and Rough Surfaces and on Gratings*. Verlag: Springer. DOI: https://doi.org/10.1007/BFb0048317
14. **Maier, S M** 2007 *Plasmonics: Fundamentals and applications*. Springer. DOI: https://doi.org/10.1007/0-387-37825-1
15. **Foley, J J, IV, McMahon, J M, Schatz, G C, Harutyunyan, H, Wiederrecht, G P** and **Gray, S K** 2014 Inhomogeneous surface plasmon polaritons. *ACS Photon*, 1: 739–745. DOI: https://doi.org/10.1021/ph500172f
16. **Kats, M A, Blachard, R, Genevet, P** and **Capasso, F** 2013 Nanometre optical coatings based on strong interference effects in highly absorbing media. *Nat. Mat*, 12: 20–24. DOI: https://doi.org/10.1038/nmat3443
17. **Foley, J J, IV, Harutyunyan, H, Rosenmann, D, Divan, R, Wiederrcht, G P** and **Gray, S K** 2015 When are Surface Plasmon Polaritons Excited in the Kretschmann-Raether Configuration? *Sci. Rep*, 5: 9929. DOI: https://doi.org/10.1038/srep09929
18. **Foley, J J, IV, Ungaro, C, Sun, K, Gupta, M C** and **Gray, S K** 2015 Design of emitter structures based on resonant perfect absorption for thermophotovoltaic applications, 23: A1373–A1387. DOI: https://doi.org/10.1364/OE.23.0A1373
19. **Byrnes, S J** 2018 Multilayer Optical Calculations. https://arxiv.org/abs/1603.02720.
20. https://pypi.org/project/tmm/.
21. **Yuffa, A J** and **Scales, J A** 2012 Object-oriented electrodynamic S-matrix code with modern applications. *J. Comp. Phys*, 20: 4823–4835. DOI: https://doi.org/10.1016/j.jcp.2012.03.018
22. https://pypi.org/project/openTMM/.
23. https://kitchenknif.github.io/PyTMM/.
24. https://github.com/ovidiopr/tmmnlay.
25. https://www.mathworks.com/matlabcentral/fileexchange/47637-transmittance-and-reflectance-spectra-of-multilayered-dielectric-stack-using-transfer-matrix-method.
26. **Liu, V** and **Fan, S** 2012 S4: A free electromagnetic solver for layered periodic structures. *Comp. Phys. Comm*, 183: 2233–2244. DOI: https://doi.org/10.1016/j.cpc.2012.04.026
27. https://web.stanford.edu/group/fan/S4/.
28. https://nanohub.org/resources/s4sim/about.
29. https://github.com/FoleyLab/wptherml/blob/master/Validate_Fresnel.ipynb.
30. https://github.com/FoleyLab/wptherml/blob/master/Validate_Cooling.ipynb.
31. https://github.com/FoleyLab/wptherml/issues.
32. **Johnson, P B** and **Christy, R W** 1972 Optical Constants of noble metals. *Phys. Rev. B*, 6: 4370. DOI: https://doi.org/10.1103/PhysRevB.6.4370
33. **Yang, H U, D'Archangel, J, Sundheimer, M L, Tucker, E, Boreman, G D** and **Raschke, M B** 2015 Optical dielectric function of silver. *Phys. Rev. B*, 91: 235137. DOI: https://doi.org/10.1103/PhysRevB.91.235137
34. **Olmon, R L, Slovick, B, Johnson, T W, Shelton, D, Oh, S-H, Boreman, G D** and **Raschke, M B** 2012 Optical dielectric function of gold. *Phys. Rev. B*, 86: 235147. DOI: https://doi.org/10.1103/PhysRevB.86.235147
35. **Rakić, A D** 1995 Algorithm for the determination of intrinsic optical constants of metal films: application to aluminium. *Appl. Opt*, 34: 4755–4767. DOI: https://doi.org/10.1364/AO.34.004755
36. **Kischkat, J, Peters, S, Gruska, B, Semtsiv, M, Chashnikova, M, Klinkmüller, M, Fedosenko, O, Machulik, S, Aleksandrova, A, Monastyrskyi, G, Flores, Y** and **Masselink, W T** 2012 Mid-infrared optical properties of thin films of aluminum oxide, titanium dioxide, silicon dioxide, aluminum nitride, and silicon nitride. *Appl. Opt*, 51: 6789–6798. DOI: https://doi.org/10.1364/AO.51.006789
37. **Palik, E D** 1998 Handbook of optical constants of solids. Academic Press.
38. **Malitson, I H** 1965 Interspecimen comparison of the refractive index of fused silica. *J. Opt. Soc. Am*, 55: 1205–1208. DOI: https://doi.org/10.1364/JOSA.55.001205
39. **Popova, S, Tolstykh, T** and **Vorobev, V** 1972 Optical characteristics of amorphous quartz in the 1400–200 cm-1 region. *Opt. Spectrosc*, 33: 444–445.
40. **Schinke, C, Peest, P C, Schmidt, J, Brendel, R, Bothe, K, Vogt, M R, Kröger, I, Winter, S, Schirmacher, A, Lim, S, Nguyen, H T** and **MacDonald, D** 2015 Uncertainty analysis for the coefficient of band-to-band absorption of crystalline silicon. *AIP Advances*, 5: 67168. DOI: https://doi.org/10.1063/1.4923379
41. **Siefke, T, Kroker, S, Pfeiffer, K, Puffky, O, Dietrich, K, Franta, D, Ohlídal, I, Szeghalmi, A, Kley, E-B** and **Tünnermann, A** 2016 Materials pushing the

application limits of wire grid polarizers further into the deep ultraviolet spectral range. *Adv. Opt. Mater*, 4: 1780–1786. DOI: https://doi.org/10.1002/adom.201600250

42. **Dashiell, M,** et al. 2005 "Quaternary InGaAsSb Thermophotovoltaic Diode Technology". *Technical Report*. DOI: https://doi.org/10.1063/1.1841919
43. **Martin, D** and **Algora, C.** 2004 "Temperature-dependent GaSb material parameters for reliable thermophotovoltaic cell modelling". *Semicond. Sci. Technol*, 19: 1040–1052. DOI: https://doi.org/10.1088/0268-1242/19/8/015
44. ASTM G173-03 Reference Spectra Derived from SMARTS v. 2.9.2. https://www.nrel.gov/grid/solar-resource/smarts.html.
45. IR Transmissivity data from https://www.gemini.edu/sciops/telescopes-and-sites/observing-condition-constraints/ir-transmission-spectra.
46. Photopic luminosity function data from http://www.cvrl.org/lumindex.htm.

How to cite this article: Varner, J F, Eldabagh, N, Volta, D, Eldabagh, R and Foley, J J, IV 2019 WPTherml: A Python Package for the Design of Materials for Harnessing Heat. *Journal of Open Research Software,* 7: 28. DOI: https://doi.org/10.5334/jors.271

Submitted: 22 March 2019 **Accepted:** 09 August 2019 **Published:** 19 August 2019

]u[*Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS