A Jupyter-Based Photovoltaic External Quantum Efficiency Spectrum-Analysing Tool

Manual

Compiled by Austin M. Kay & Maura E. Fitzsimons

Under the supervision of

Dr. Oskar J. Sandberg, Prof. Ardalan Armin, Dr. Gregory Burwell, & Prof. Paul Meredith

Version Created:

Friday 19th May, 2023

Sustainable Advanced Materials (Sêr SAM)
Swansea University
Swansea
SA2 8PP
United Kingdom

Contents

List of Figures

Chapter 1

Prerequisites

1.1 Introduction

Python is a robust, general-purpose programming language that can be used in a variety of applications. Python code can be written and executed at the command-line, or can be written in integrated development environments (IDEs) such as "Spyder" and "Jupyter". Each of these IDEs has its own benefits; we have chosen to use Jupyter as it allows us to alternate between Python code and LaTeX-style text.

We have prepared this document to help anyone navigate our Jupyter tool's graphical user interface (GUI), so that they can analyse photovoltaic external quantum efficiency data, or simulate device performance under different illumination conditions. This document serves as a step-by-step walk-through; from installing the necessary applications, to opening the script and using the GUI. No prior experience with Python, or any other programming language for that matter, is required.

In this remainder of this chapter, we provide instructions on how to install the prerequisite tools and applications for anyone who might not have them. This starts with the Python distribution Anaconda in Section 1.2, which can be used to open a Jupyter notebook as well as other types of IDEs. Following this comes an introductory guide to Jupyter notebooks in Section 1.3.

1.2 Installing Anaconda

The Python distribution Anaconda contains several different Python-based IDEs, where code can be independently written in each of them; to install it on Windows, we follow this guide. The guides for other operating systems can be found on this website.

To begin, the Anaconda installer is downloaded from this website, which is illustrated in Figure 1.1. After some time, the download completes and the installer should open as seen in Figure 1.2. After clicking "Next >" , agree to the license agreement (Figure 1.3), choose whether to install Anaconda for all users or just yourself (Figure 1.4), and specify where to install Anaconda (Figure 1.5). Following this, choose the option to "Register Anaconda3 as my default Python 3.9" as shown in Figure 1.6, then begin the installation process.

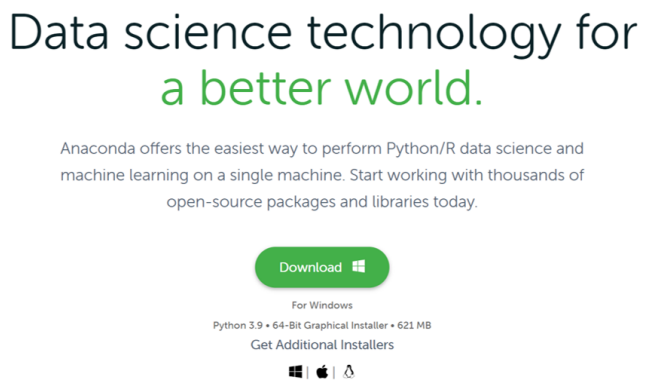


Figure 1.1: The website that the Anaconda installer can be downloaded from for Windows.

Figure 1.2: The first page of the Anaconda installation process.

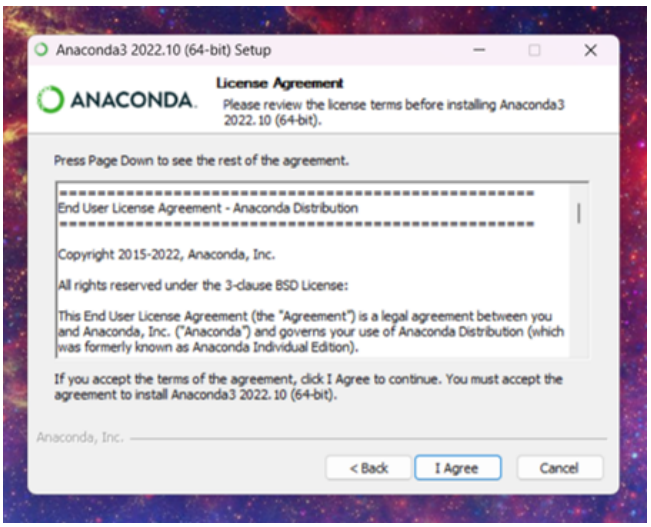


Figure 1.3: Agree to the license agreement.

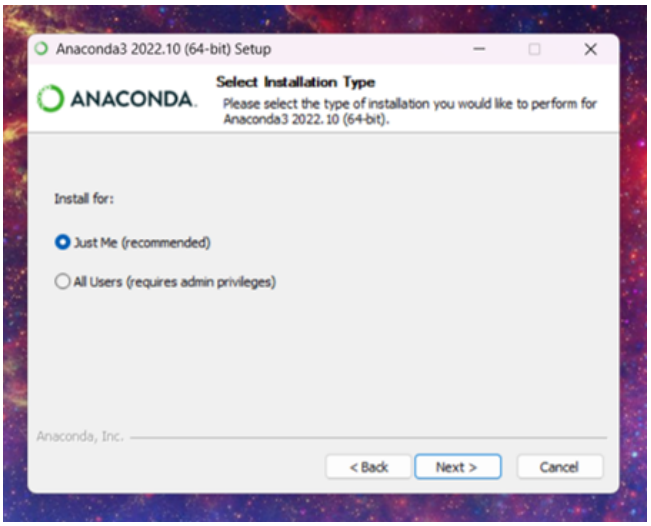


Figure 1.4: Choose whether to install Anaconda for all users or just yourself.

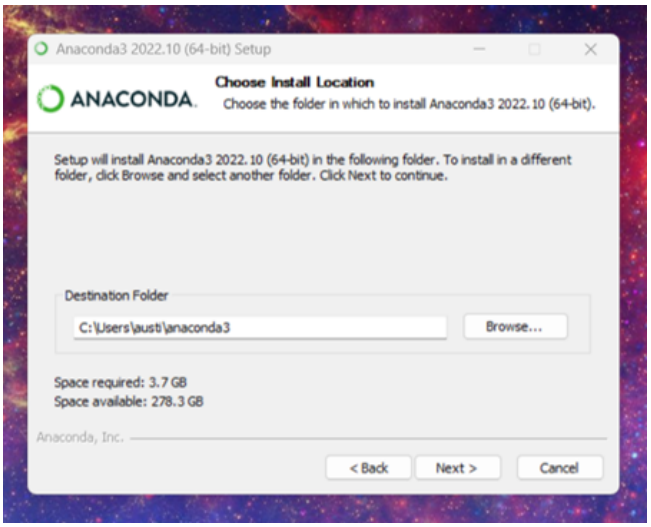


Figure 1.5: Specify Anaconda's installation location.

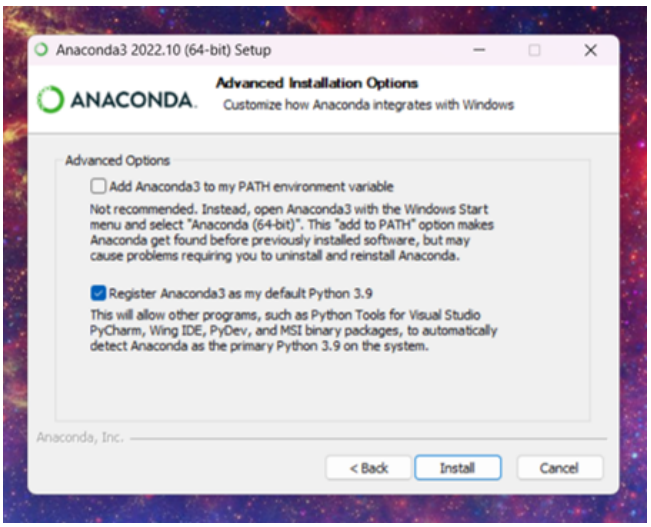


Figure 1.6: Register Anaconda as the default Python 3.9 application.

1.3 Jupyter Notebooks

When Anaconda has been installed as described in Section 1.2, the Anaconda application can be opened by looking it up in the search bar, as shown in Figure 1.7. After opening the application, update it if notified to. The Jupyter IDE can now be launched, resulting in a new window like Figure 1.9 popping up in your web browser. To open an existing interactive Jupyter notebook, navigate from your home directory to wherever the notebook is saved. To create a new Jupyter notebook, select "New" to open up a drop-down menu, then select "Python 3 (ipykernel)".

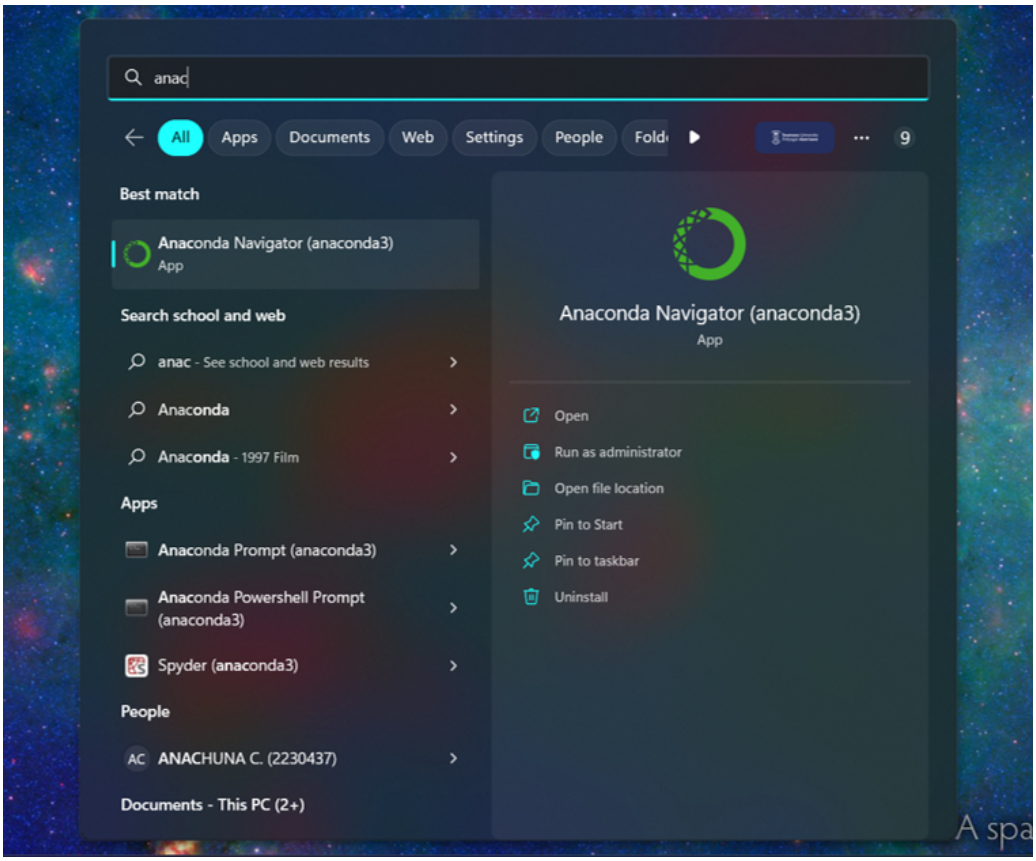


Figure 1.7: Opening Anaconda using the search bar.

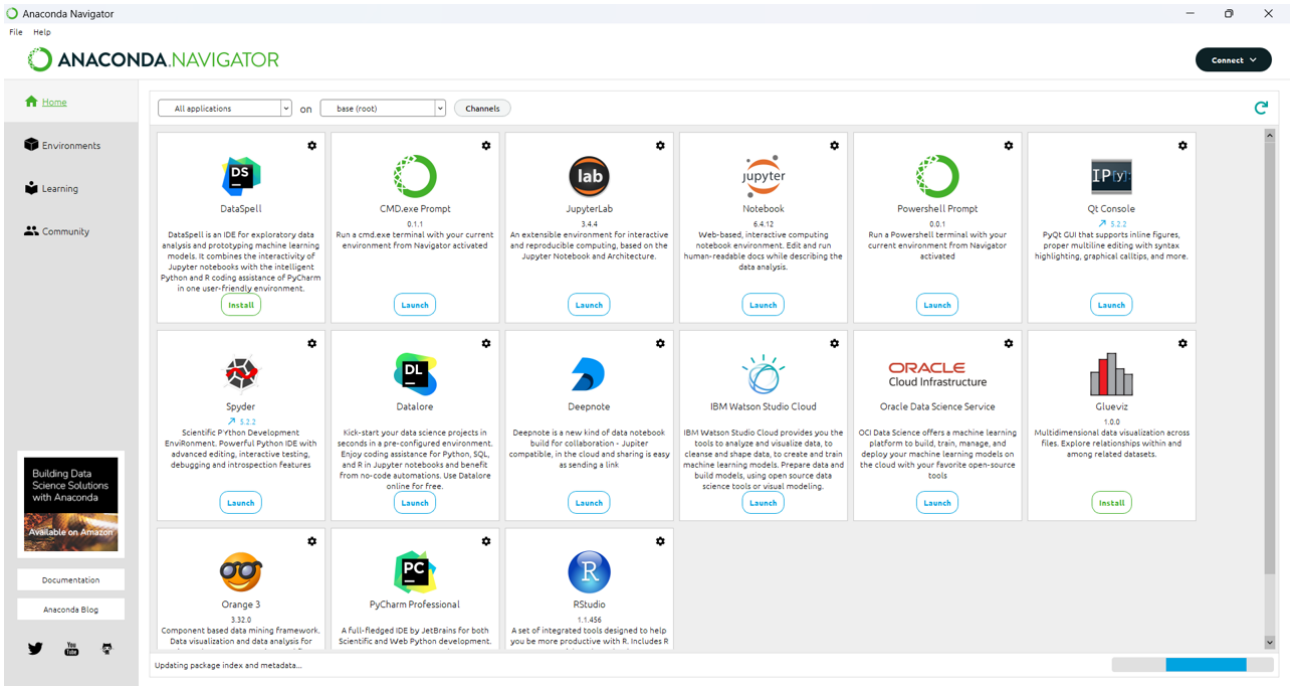


Figure 1.8: The Anaconda application, where different types of IDEs can be opened.

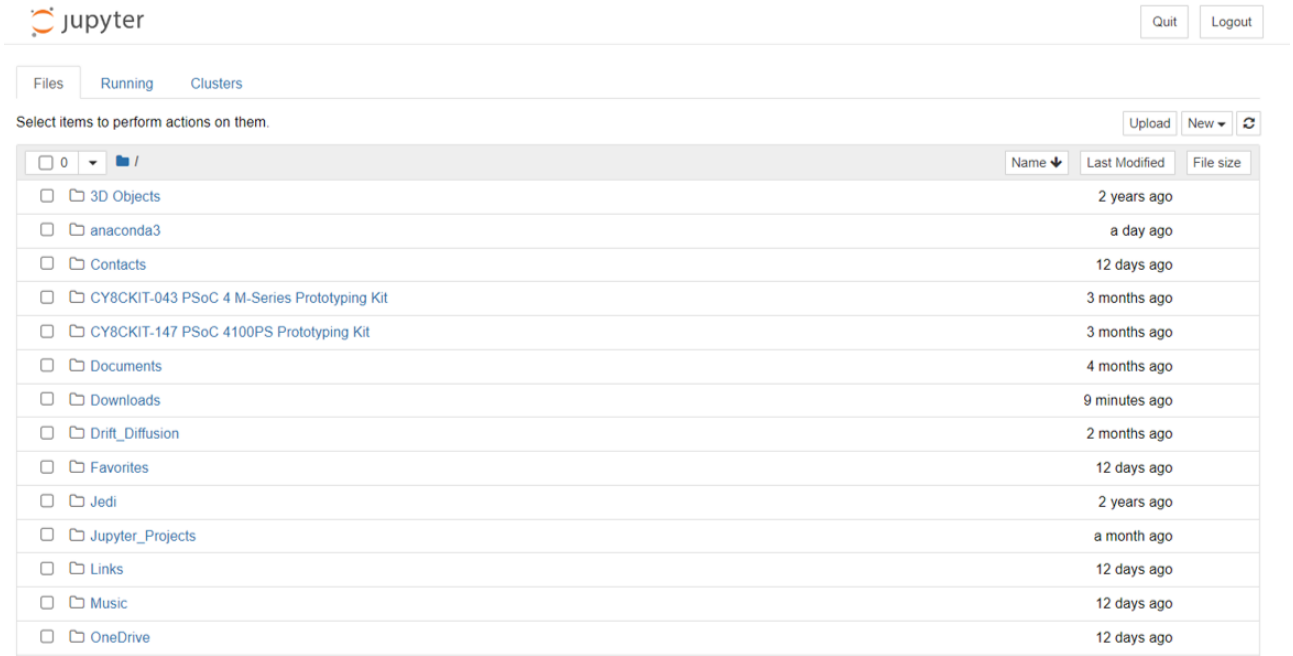


Figure 1.9: A new instance of Jupyter in the web browser.

Chapter 2

Simulation Interface

After navigating to the directory with the Jupyter-based EQE_{PV} spectrum-analysing tool, open up the script. You should find a page like the one in Figure 2.1. To initialise the script, select "Cell" from the top bar, then select "Run all". All the code in the script will be run, and the script will jump to the GUI (which will be generated by the code after a few moments).

Figure 2.1: Opening the Jupyter-based EQE_{PV} spectrum-analysing tool.

In the GUI (shown in Figure 2.2), you will find two interfaces (highlighted by green box), one for simulating figures of merit under a simulated EQE_{PV} spectrum (Simulation Interface), the other for analysing imported EQE_{PV} spectra (Analysis Interface); the first of these interfaces is explored in this chapter, whereas the second is explored in the next.

The Simulation Interface, which is the interface the GUI opens on by default, is broken into two panels; one for controlling the overall input parameters like temperature or the non-radiative loss model being used (indicated by **(a)** in Figure 2.2), and one for running optical gap-dependent or intensity-dependent simulations (indicated by **(b)** in Figure 2.2). The input parameter interface and the simulation-running interface are explored in more detail in Section 2.1 and Section 2.2, respectively.

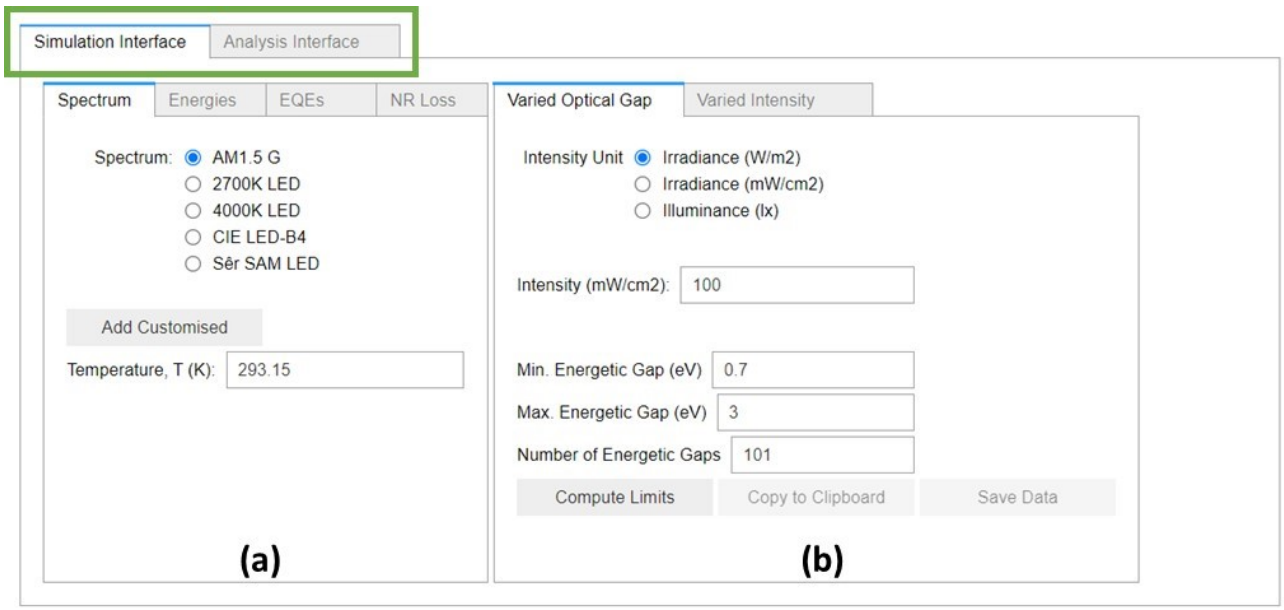


Figure 2.2: The graphical user interface, opened up on the simulation panel.

2.1 Input Parameters

In Figure 2.3, the four panels of the simulation input parameter interface are shown. The first of these (panel **(a)**) is used to control the spectrum with which the simulation is being conducted (e.g., AM1.5 G) and the temperature at which the simulation is being conducted (default is 293.15 K). By pressing the "Add Customised Button", a new pane will open up in this interface (see Figure 2.4), wherein two or more spectra may be superimposed by checking their boxes, and adding their relative contributions (i.e., their contribution to the total intensity of the light source); these contributions can be anything, with 1:1 meaning the relative intensity of both spectra is equal.

In panel **(b)** of Figure 2.3, the parameters defining the photon energies used to simulate the $\mathrm{EQE_{PV}}$ spectrum, interpolate the photon fluxes, and, ultimately, evaluate the figures-of-merit are specified. The maximum photon energy (in electronvolts) must be larger than the minimum photon energy, and the number of photon energies must be an integer.

There were three different models for $\mathrm{EQE_{PV}}$ spectra considered in this work. As shown in panel **(c)** of Figure 2.3, these models are the step function model, the Urbach tail model, and the Exciton absorption model. The Urbach tail model is applicable to inorganic semiconductors (like amorphous silicon Urbach energy $E_\mathrm{U} \sim 40\,\mathrm{meV}$) and perovskites (with Urbach energies on the order of 15 meV). The inputs that follow the model type start with "Above-Gap EQE", which is used in all models to specify the above-gap photovoltaic quantum efficiency. If a step-function model for $\mathrm{EQE_{PV}}$ is assumed, the second option will be"Below-Gap EQE", which specifies the photovoltaic quantum efficiency below the gap (and defaults to zero). If the Urbach tail model is selected, the second input becomes the Urbach energy (with an option to set it equal to the thermal energy, kT, with k being the Boltzmann constant and T being the temperature). If the Exciton Absorption model is selected, the second input will be the energetic disorder (σ_s, which describes the density of exciton states).

Shown in panel **(d)** of Figure 2.3 is the final panel of the input parameter interface. This specifies

the non-radiative loss type used in the simulation. A variety of empirical non-radiative loss models are provided, including the quadratic model based on the work of Ullbrich et al.. As one might expect, the radiative limit assumes no open-circuit voltage loss. Multiple loss types can be selected (the simulations will be conducted for both, then stored as separate curves).

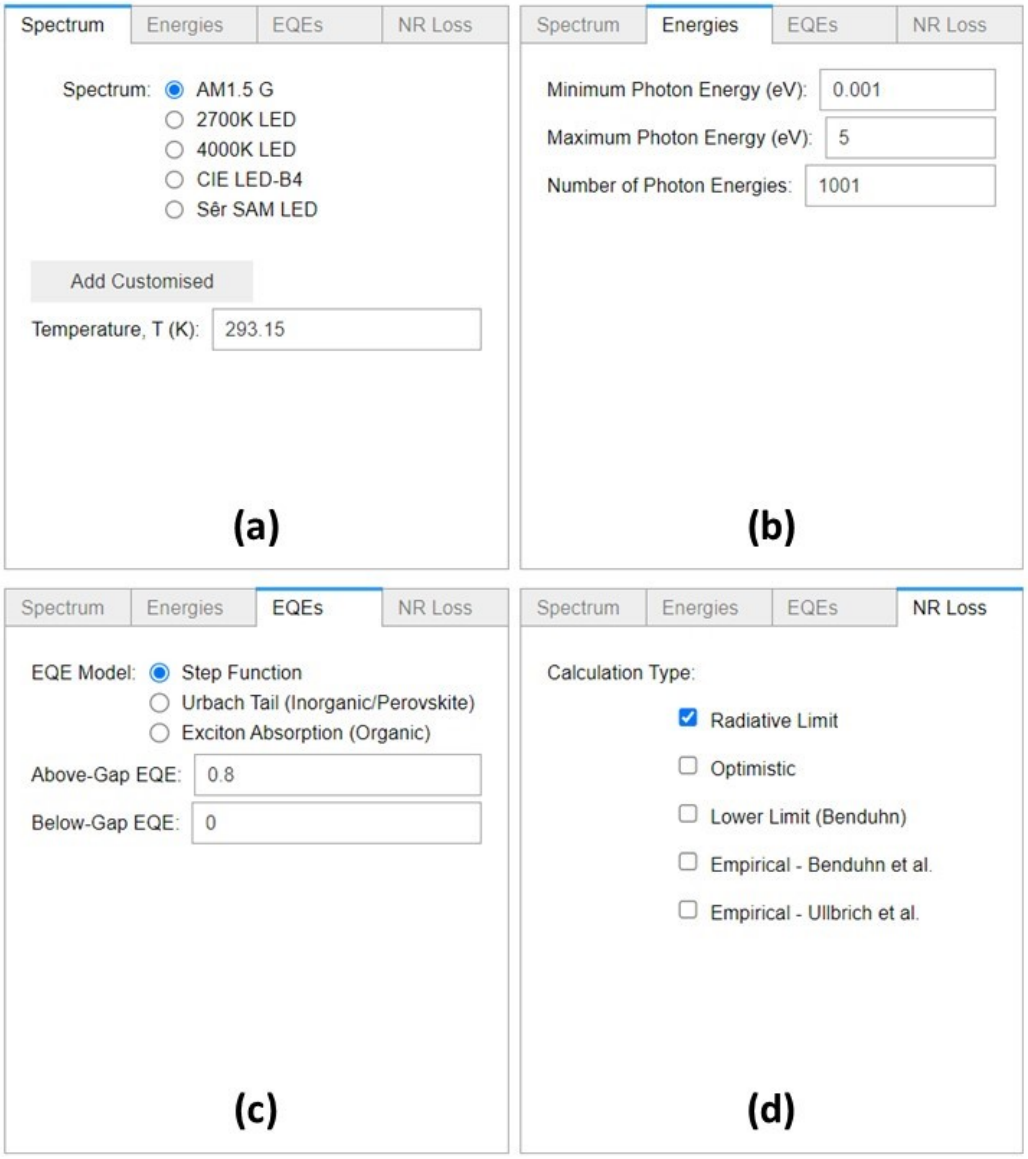


Figure 2.3: **(a)**-**(d)** The panels of the simulation input parameter interface, needed for estimating figures-of-merit using a simulated $\mathrm{EQE_{PV}}$ spectrum.

2.2 Simulation-Running Interface

The simulation-running interface takes all the parameter inputs and does one of two things; it either simulates against the optical gap at a fixed light intensity, or it simulates against the light intensity at a particular optical gap. The former interface is illustrated in **(a)** of Figure 2.5, whereas the latter is illustrated in **(b)**. The process is essentially the same for both: the varying parameters should be specified, "Compute Limits" should be pressed, then the simulations will be conducted and the graphs presented (see Figure 2.6). Therein, the graphs can be toggled between using the radio buttons in

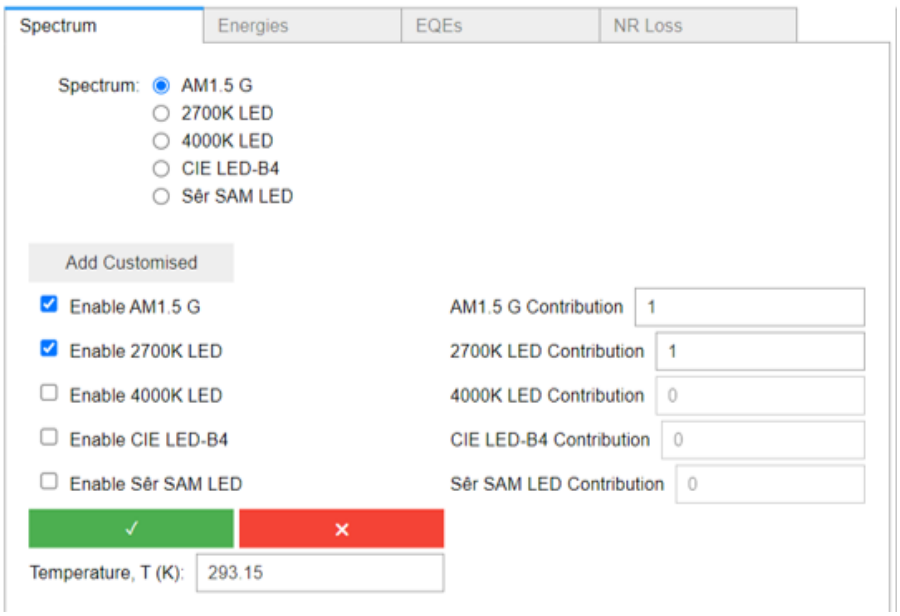


Figure 2.4: The spectrum superimposing interface, where AM1.5G and 2700K LED are to be superimposed at a 1:1 ratio. Clicking the green tick will add the spectrum to the list of available spectra (at the top of the image), whereas clicking the red cross will cancel the process.

the blue box. Also indicated by the green box is the readout, which gives (for optical gap-dependent simulations) the best-performing optical gap and the maximum power conversion efficiency. Alternatively, for intensity dependent simulations, if "Find and Use Best Gap" is enabled, the readout will be the optical gap that gives the maximum power conversion efficiency.

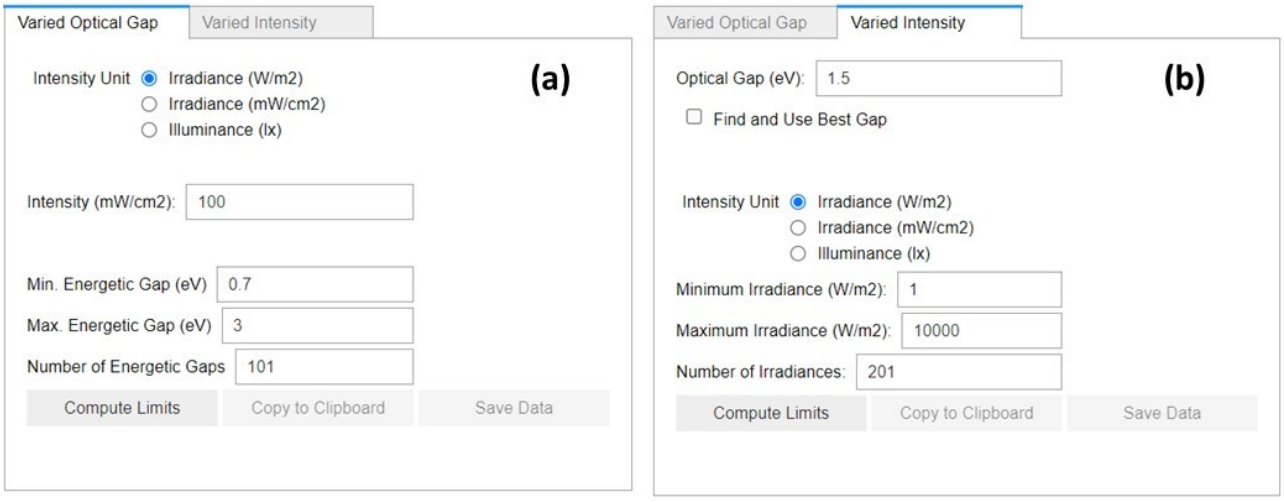


Figure 2.5: The simulation-running interfaces for **(a)** optical gap-dependent simulations and **(b)** light intensity-dependent simulations.

Following a simulation, the user can either copy the data to the clipboard by pressing "Copy to Clipboard", or save the data to a text file by pressing "Save Data". Alternatively, the basic Python plots can be saved directly by right-clicking them.

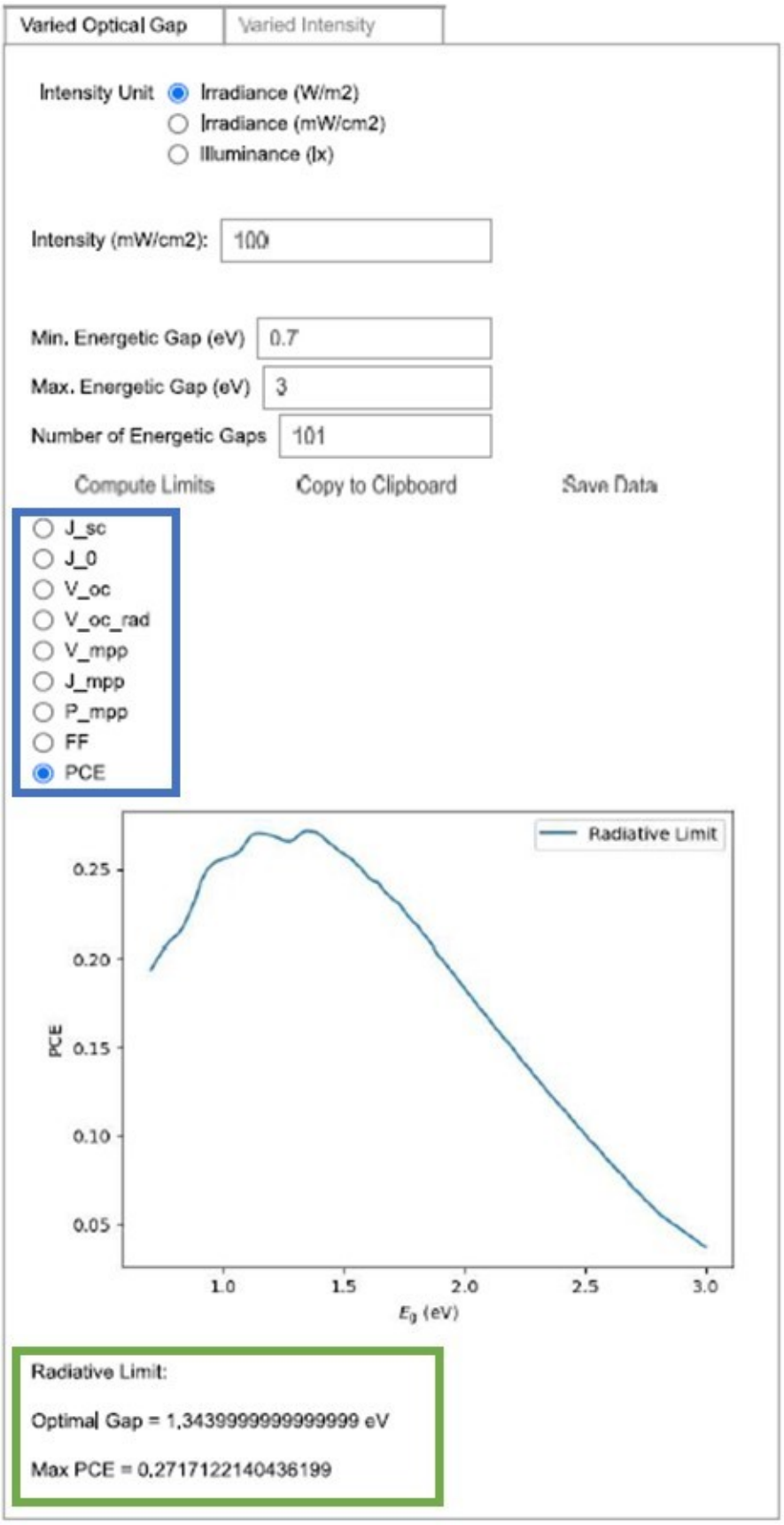


Figure 2.6: Simulation panel after conducting a simulation. The blue box highlights the radio button widgets that can be used to change between graphs, whereas the green box indicates the output values according to the non-radiative loss model.

Chapter 3

Individual Analysis Interface

After navigating to the Analysis Interface and pressing the "Add EQE Spectrum" button, a user should be presented with a text box and an accompanying drop-down list with all the EQE_{PV} filenames available in the 'EQE_Spectra' directory, as illustrated in Figure 3.1. The list of available spectra can be reduced by typing a letter or two from the filename. The spectrum-adding process will not proceed until a valid filename has been entered.

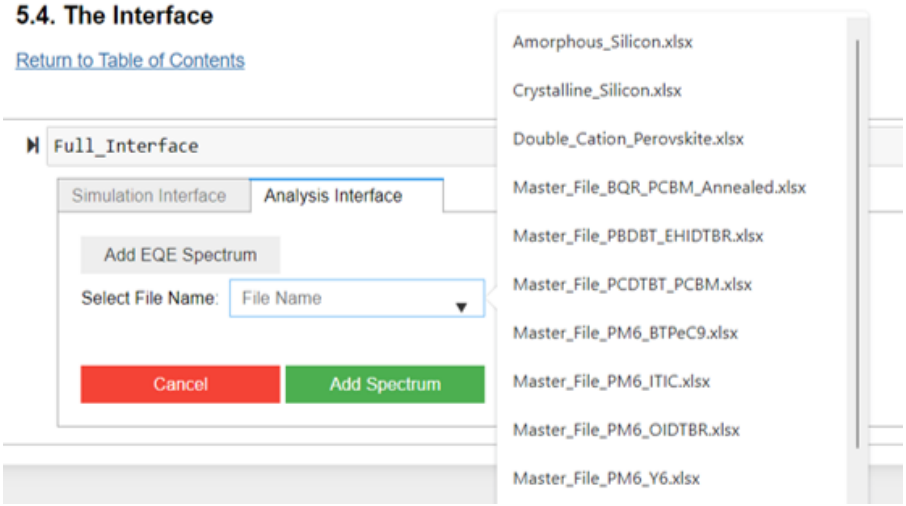


Figure 3.1: The first step of importing an EQE_{PV} spectrum into the computational tool: entering a filename.

Entering a valid filename will trigger a preliminary loading of the data, where a preview of the data should be provided (see Figure 3.2). Under this preliminary view, which columns are independent variables and which are dependent variables can be specified. Any given dependent variable column will use the nearest independent variable column on its left. This allows data sets with alternating λ and EQE_{PV} data to be loaded into the tool. Moreover, which columns are included can be altered using the checkboxes - a file containing many columns can be narrowed down to one data set if one wishes. Furthermore, EQE_{PV} can be in % or unitless, as long as the user ticks the correct box. Finally, the column labels can be altered; this, in turn, alters the eventual tab names in the spectrum-analysing interface.

When the user is ready to import the spectrum, the green "Add Spectrum" must be clicked,

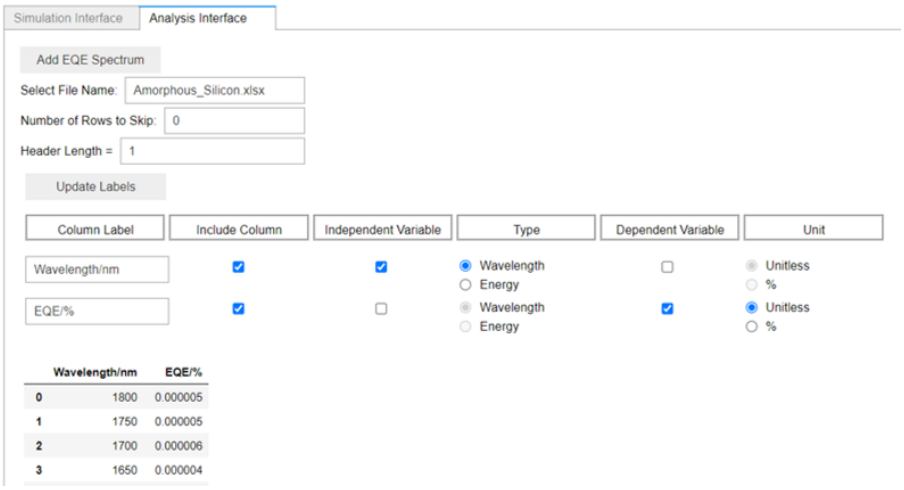


Figure 3.2: The second step of importing an $\mathrm{EQE_{PV}}$ spectrum into the computational tool: specifying dependent and independent variables, and units.

resulting in a tab being added to the Analysis Interface, as shown in Figure 3.3. The left-hand portion of this tab functions as an extremely bare-bones version of the input parameter panel (see Section 2.1), whereas the right-hand panel functions exactly the same as the simulation-running tab in Section 2.2. The only addition with the former is that, instead of non-radiative loss models, the non-radiative loss, the electroluminescent external quantum efficiency, and the open-circuit voltage are used to determine non-radiative losses. To get the radiative limit, enter a non-radiative loss of zero. On the other hand, the only difference with the simulation-running panel in the right-hand side of Figure 3.3 is the addition of the $\mathrm{EQE_{PV}}$ spectrum, which is added to the other graphs.

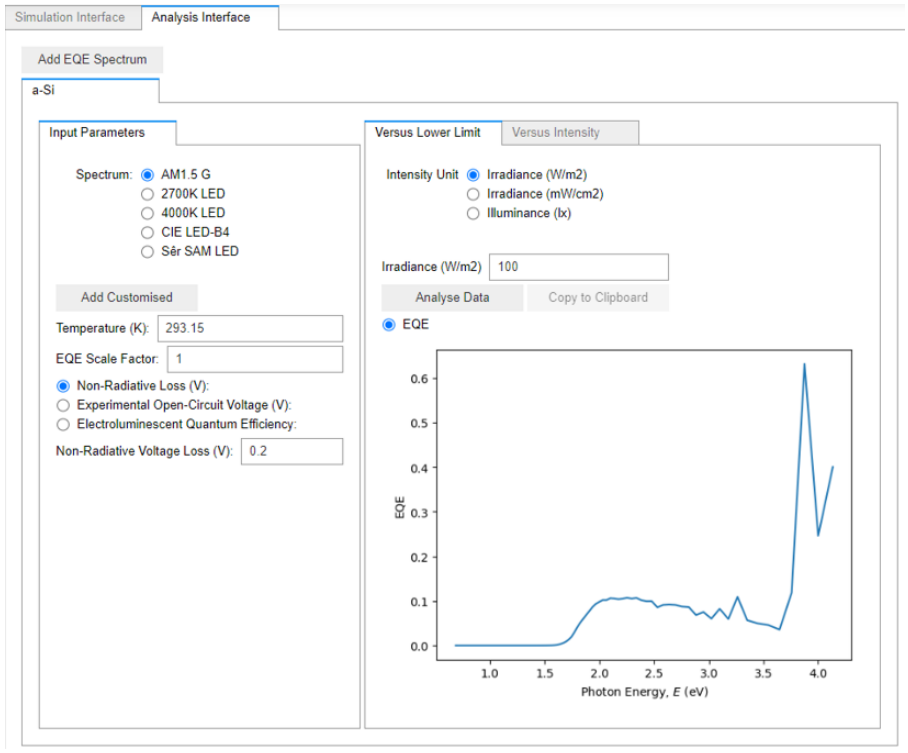


Figure 3.3: The second step of importing an $\mathrm{EQE_{PV}}$ spectrum into the computational tool: specifying dependent and independent variables, and units.

Chapter 4

Bulk Analysis Interface

The interface for analysing data in bulk is illustrated in Figure 4.1. This relatively simple interface lets the user select a spectrum type on the top left-hand side (e.g., AM1.5 G, or LED-B4). It also allows the user add a customised spectrum (i.e., superimposing multiple spectra). Under the selected spectrum at a specified light intensity (unit can be selected using the "Intensity Unit" widget) and temperature, the stored data can be analysed in bulk by pressing the "Analyse Data" button on the bottom right. Following this, the results will be presented in the graph on the upper left (alongside the Shockley-Queisser limit of that spectrum), and the option to copy the results to the clipboard will become available.

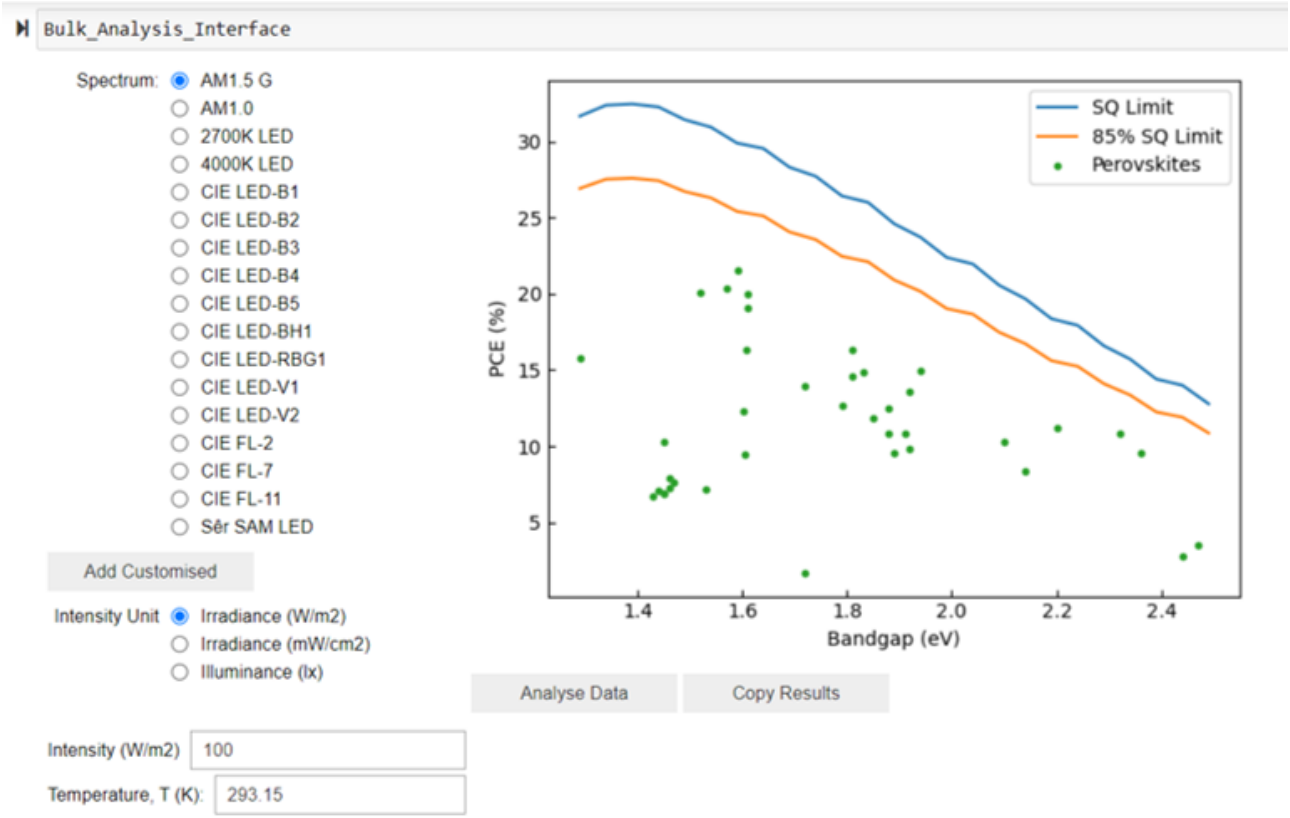


Figure 4.1: The bulk analysis interface, through which a spectrum can be selected, an intensity value and temperature can be specified, the simulations can be run (with data outputted on the graph), and the results can be copied.

Chapter 5

Suggestions for User Customisation

If you would like to customise the Jupyter tool further, for a few different scenarios we now provide a few suggestions that might be helpful.

5.1 Adding EQE Spectra

Additional photovoltaic external quantum efficiency spectra can be added to the tool's data base by simply saving the data to a '.xlsx' Excel file, with the independent variable (be it photon energy or wavelength) in the first column, and the $\mathrm{EQE_{PV}}$ dependent variable stored in the second column, as shown in Figure 5.1. To avoid errors when importing the data, we recommend that the data is stored with the header being no more than one row (i.e., title and unit together).

Once an $\mathrm{EQE_{PV}}$ spectrum has been saved to the 'EQE_Spectra' directory, re-initialising the computational tool will add the spectrum to the list of available spectra in the experimental spectrum-analysing interface.

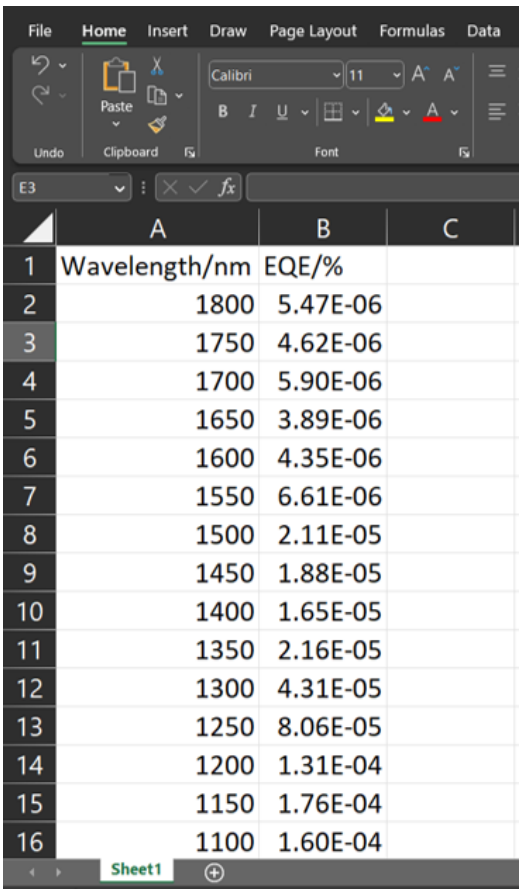
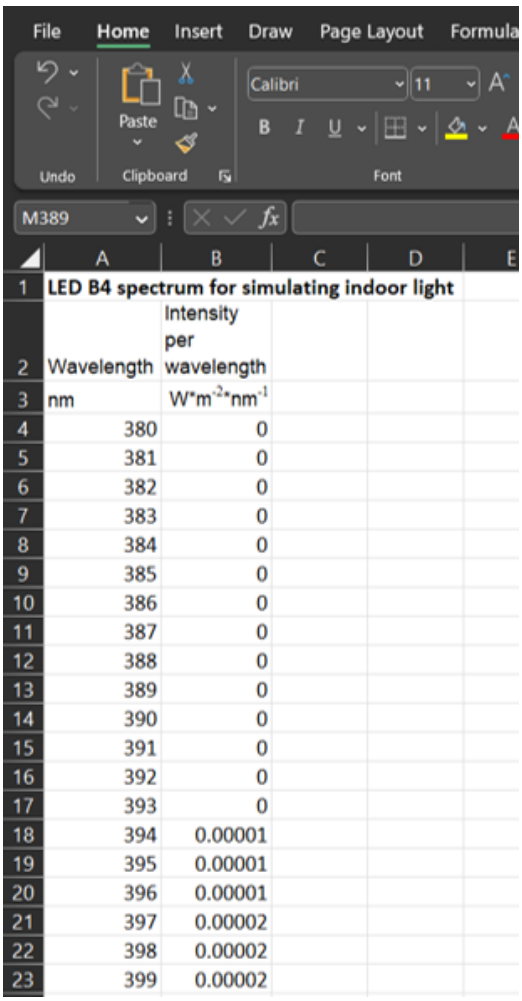


Figure 5.1: An example '.xlsx' Excel file, with photon wavelengths being stored in the first column, photovoltaic external quantum efficiency measurements being stored in the second, and a header that is one row in length.

5.2 Adding Photon Flux Spectra

The basic version of the computational tool includes a handful of photon flux spectra, including AM1.5G, 2700K LED, and CIE LED-B4. Additional spectra can be added to the tool by simply adding them as new sheets to the 'Spectra.xlsx' Excel file in the 'Spectra' directory. The added spectrum should be inserted as shown in Figure 5.2, with the first row being reserved for the spectrum name, the second row containing the column titles, and the third row being the units. The spectrum must be in terms of wavelength, with the intensities quoted in units of $\mathrm{W\,m^{-2}\,nm^{-1}}$.

Once the spectrum has been added to the Excel file, re-initialising the script will add it as an in-built option. The lux conversion factors, the total irradiances, and more of the like, will readily be computed as the script runs. Your added spectrum should appear as a new option among original spectra.



	A	B	C	D	E
1	**LED B4 spectrum for simulating indoor light**				
2	Wavelength	Intensity per wavelength			
3	nm	$W*m^{-2}*nm^{-1}$			
4	380	0			
5	381	0			
6	382	0			
7	383	0			
8	384	0			
9	385	0			
10	386	0			
11	387	0			
12	388	0			
13	389	0			
14	390	0			
15	391	0			
16	392	0			
17	393	0			
18	394	0.00001			
19	395	0.00001			
20	396	0.00001			
21	397	0.00002			
22	398	0.00002			
23	399	0.00002			

Figure 5.2: Photon flux spectrum stored in a sheet of the 'Spectra.xlsx' Excel file, with three rows reserved for the spectrum name, column headers, and units, respectively. Spectra are expected to be inserted as a function of the wavelength, and should be in units of $\mathrm{W\,m^{-2}\,nm^{-1}}$.

5.3 Altering the Code

The computational tool was tailor-made for our group's work on indoor photovoltaics. However, as much effort as possible has been made to keep the code scalable and customisable, such that future investigations may be conducted using it. We encourage users to alter and improve the code to whatever purpose they deem fit, if they'd like to. We are also open to discussions and recommendations/suggestions.