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

Instruction Manual

Compiled by Austin M. Kay & Maura Fitzsimons

Under the supervision of

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

Version Created:

Monday 19th December, 2022

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

Llywodraeth Cymru
Welsh Government

Cronfa Datblygu
Rhanbarthol Ewrop
European Regional
Development Fund

Sêr SAM

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 a guide to installing the text editor Notepad++ in Section 1.3, which we recommend for quickly saving and loading data. Finally, an introductory guide to Jupyter notebooks is given in Section 1.4.

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.

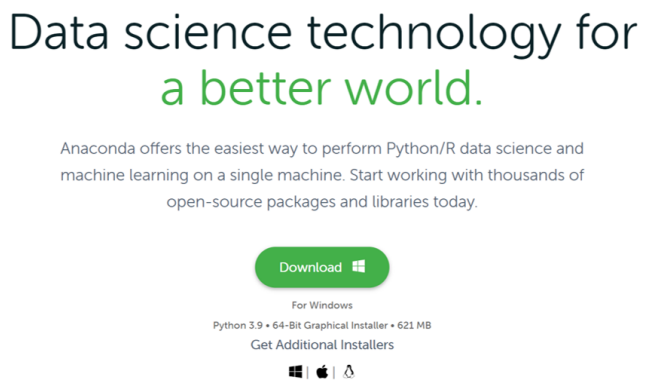

Figure 1.1: The website that the Anaconda installer can be downloaded from for Windows.

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

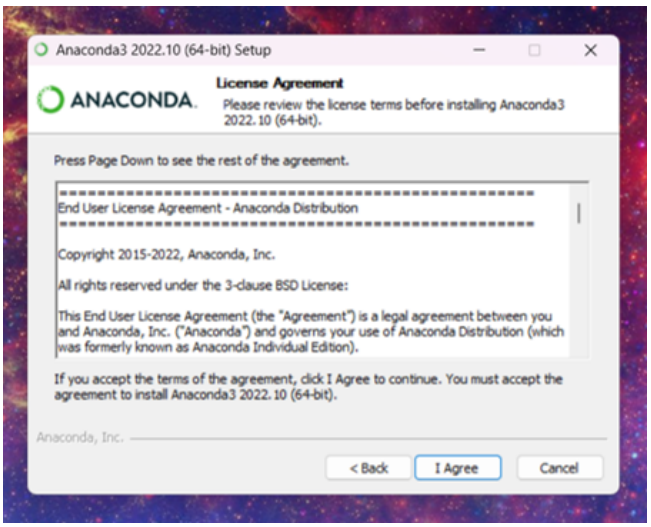

Figure 1.3: Agree to the license agreement.

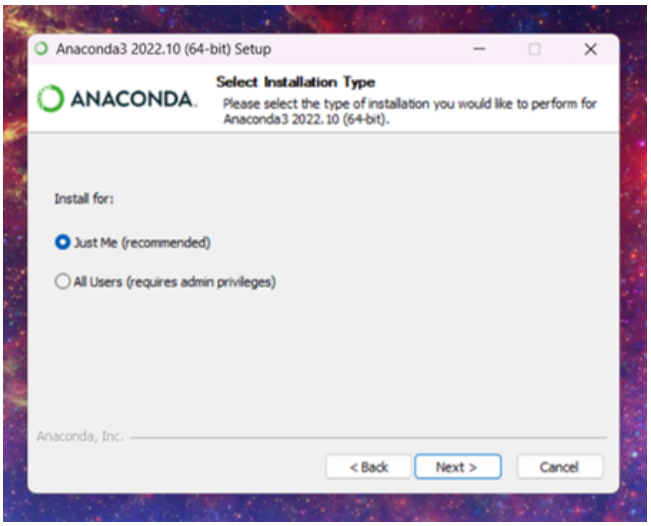

Figure 1.4: Choose whether to install Anaconda for all users or just yourself.

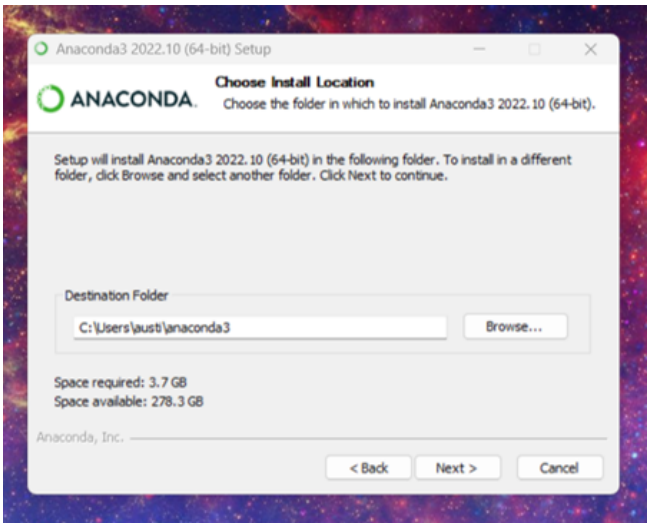

Figure 1.5: Specify Anaconda's installation location.

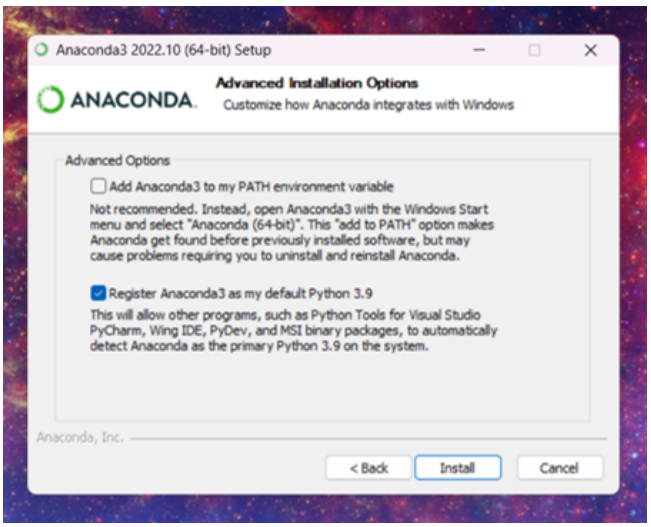

Figure 1.6: Register Anaconda as the default Python 3.9 application.

1.3 Notepad++ (Recommended)

Come back to this...

1.4 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)".

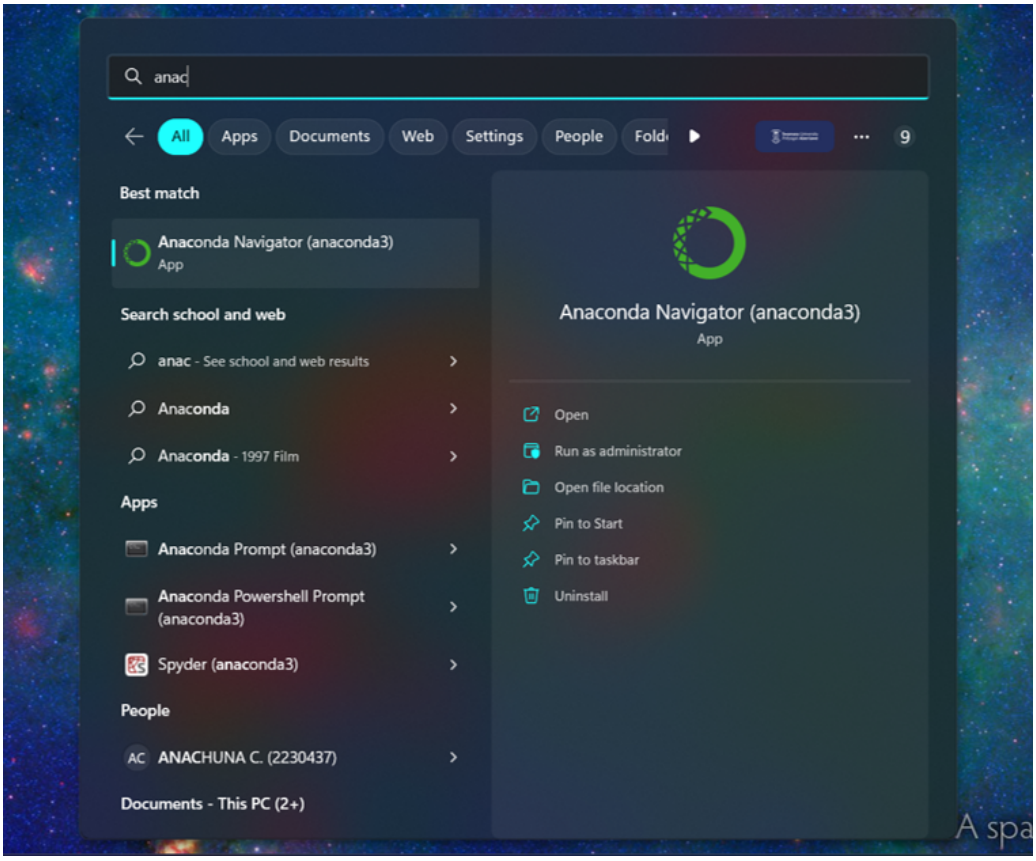

Figure 1.7: Opening Anaconda using the search bar.

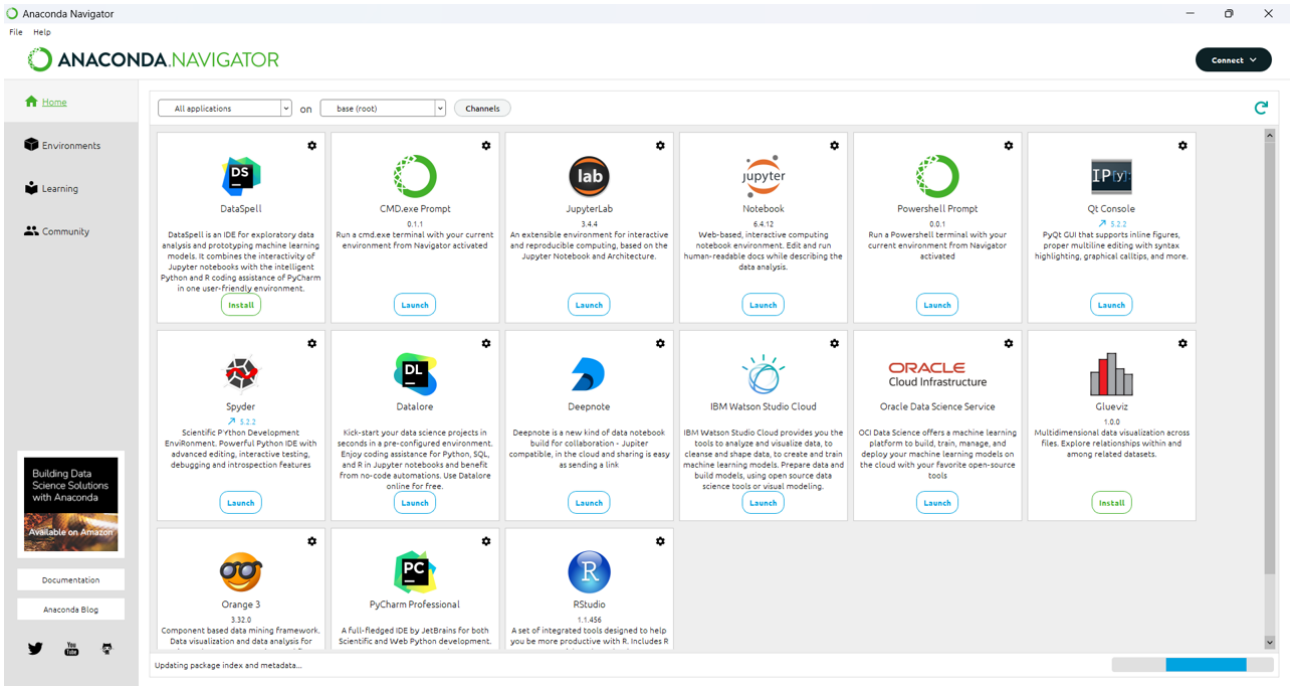

Figure 1.8: The Anaconda application, where different types of IDEs can be opened.

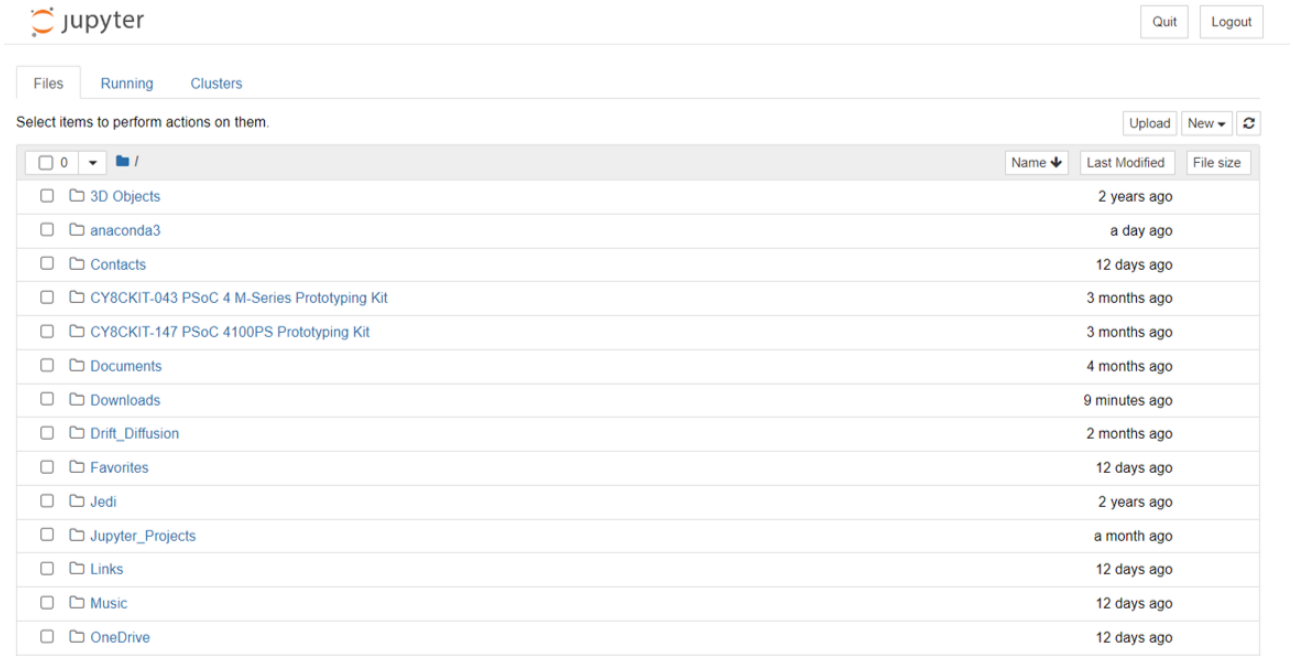

Figure 1.9: A new instance of Jupyter in the web browser.

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).

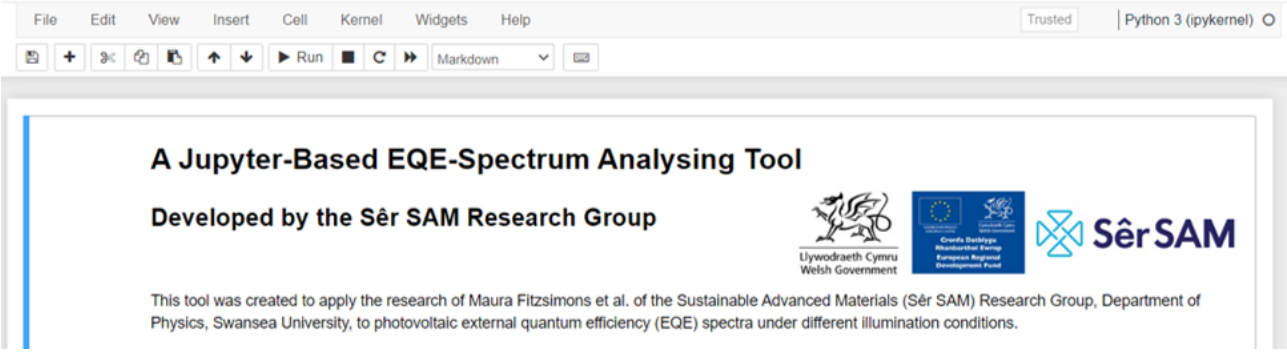

Figure 2.1: Opening the Jupyter-based EQE_{PV} spectrum-analysing tool.

In the GUI (shown in Figure 2.2), you will find two interfaces, one for performing simulations, the other for analysing EQE_{PV} spectra; the first of these interfaces is explored in this chapter, whereas the second is explored in the next.

The Simulation Interface is broken into two panels; one for controlling the overall input parameters (like temperature, or non-radiative loss model), and one for running the simulations. The overall inputs are explored further in Section 2.1,

2.1 Overall Inputs

2.2 Optical Gap-Dependent Simulations

2.3 Intensity-Dependent Simulations

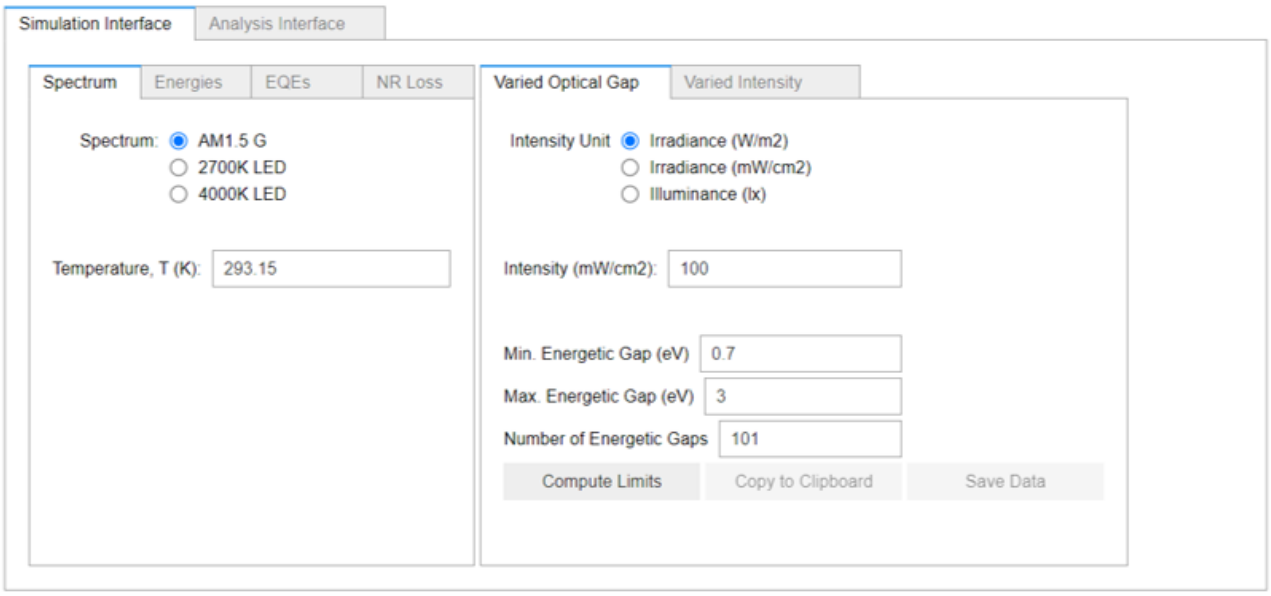

Figure 2.2: The graphical user interface, opened up on the simulation panel.

Chapter 3

Analysis Interface