The Shell
Manual
version 1.00

S. Heester
L.J.A. Koster
December 12, 2025

Contents

Chapter 1

Introduction

The Shell provides a powerful graphical interface around SIMsalabim, to perform drift-diffusion simulations for semiconductor devices. The aim of The Shell is to simplify running complex numerical simulations by providing a user-friendly interface accessible directly through a web browser. It enables users to define detailed device structures, set material and optical properties, run simulations, and visualize and analyze results without the need to modify or interact directly with the underlying code.

The Shell is built on the Streamlit framework, and leverages the pySIMsalabim package to interact with SIMsalabim and retrieve the proper functionality. It supports a wide range of experimental techniques such as steady state and transient current-voltage (JV) sweeps, external quantum efficiency (EQE), impedance spectroscopy (IS), intensity modulated photocurrent spectroscopy (IMPS), and capacitance-voltage (CV) profiling. In addition to this, it provides for a selection of these direct comparison of simulated to experimental data.

Notice: The Shell does not yet support all features and functionality that SIMsalabim has to offer. If one wants to use the full functionality and capabilities of SIMsalabim, with the highest level of customizability, download and run the project on your machine as described in the SIMsalabim Project readme. For a full overview of the functionality of SIMsalabim, refer to the SIMsalabim manual

Chapter 2

Installation

There are two ways of using The Shell:

- It is deployed online and can be accessed through the browser through www.simsalabim-online.com, without the need for installation.
- It can be run on a local machine for full control over resources and functionality, by following the steps below.

2.1 Requirements

Before setting up The-Shell, ensure your system meets the following requirements:

- **Operating system:** Linux or MacOS
  - The Shell does not natively support running on Windows, and some modifications or adaptations by the user might be needed to achieve this.
- **Python version:** >= 3.10
  - Dependent Python packages will be automatically installed, and a list can be found in section 2.3.

2.2 Running The Shell

2.2.1 Cloning the repository

Download the source files from the Github repository:

```
git clone git@github.com:kostergroup/SIMsalabim-The-Shell.git
```

This creates a local folder containing the entire project, from which The Shell will be started.

2.2.2 Running the installation script

The repository includes `run_The_Shell_local.sh`, which automates environment setup, package installation, SIMsalabim/pySIMsalabim compilation, and starting the web application. Execute it with:

```
bash run_The_Shell_local.sh
```

This script does all the setup to run The Shell, a more detailed description of some of the sub steps is listed in the next section.

2.3 Installation steps

- `python3 -m venv venv`

  To run and manage The Shell, we use a lightweight Python virtual environment (venv). This ensures an isolated, clean environment is used, to avoid any issues with dependencies or managing environments. If the virtual environment already exists in the folder, this one is used instead. As an alternative, a Pipenv can also be used.

- `source venv_TheShell/bin/activate`

  This will activate the virtual environment for the current session.

- `pip install -r requirements.txt`

  Installs the required Python package dependencies in the virtual environment. The dependent packages are:

  - streamlit
  - pandas
  - matplotlib
  - numpy
  - scipy

- `python3 get_pySIMsalabim.py`

  Get the latest version of pySIMsalabim from the repository. It is recommended to always get the latest software release, to ensure compatability and have access to all functionality.

- `python3 get_SIMsalabim.py`

  Get the latest version of SIMsalabim from the repository. If the *free Pascal compiler* is installed on the system, SIMsalabim is compiled on the spot. If not, the packaged releases are retrieved. Refer to SIMsalabim for more details on compilation.

- `streamlit run SIMsalabim.py`

  This will start the virtual environment and local server to start the application in the default browser at `http://localhost:8501`. If a different port is required, this can be changed in `.streamlit/config.toml`

Chapter 3

Simulation workflow

Upon launching, the GUI displays multiple tabs for the different experiments in the navigation bar. By choosing a tab, the user will be brought to experimental page. Here a device can be defined, which by default is a typical perovskite solar cell. For more details on the device setup and individual parameters, refer to the SIMsalabim manual. User files such as refractive indices, trap state distributions, and complete devices can be uploaded here as well. All pages except for Steady State JV, have a separate input section for defining experiment specific parameters.

After changes to the device have been made, the parameters must be saved. This ensure that all parameters are correctly written to their respective files needed for the simulation. These parameters will also be shared among the different experiments, it does thus not matter from which experiment a device is defined. Upon saving, an energy band diagram of the device is shown as reference for quick validation. When running the simulation, input verification and error handling and propagation are performed. Upon success, navigate to the Simulation results tab to view the results.

The result pages shows the output specific for the just run experiment, from JV curves and detailed carrier density and recombination profiles, to Nyquist and Bode spectra. Here the results can also be downloaded, for further use or processing.

The currently supported experiments are:

- Steady-state JV sweep
- External Quantum Efficiency
- Transient JV sweep

- Impedance spectroscopy
- Intensity modulated photocurrent spectroscopy
- Capacitance Voltage profiling

For more details on the experiments and their implementation, refer to the pySIMsalabim package.

Chapter 4

Troubleshooting

- **GUI does not launch:** Check Python version and virtual environment activation
- **Simulation fails or returns an error:** Check what the error message exactly specifies. We propagate the error handling from SIMsalabim for any errors that are a direct result of the simulation. In these cases, a clear message is returned on what the issue was, for more details refer to the SIMsalabim manual.
- **File does not upload:** Uploaded files can only be in `TXT` file format, and one at a time, except for experimental JV curves for a transient JV sweep or n,k files.
- **No results are displayed:** Ensure that a simulation has been run and that it completed successfully.
- **The simulation timed out:** By default the maximum time a simulation can run is 5 minutes, which should be well enough for most simulations. If a longer time is required, the `timeout` parameter in the User Interface section must be changed.

If issues or bugs are encountered, open an issue on the Github repository or contact us. Also check the documentation of SIMsalabim, for details on the drift-diffusion model. Provide as much information as possible, including the exact configuration of your device, and steps to reproduce the bug or issue.