Off-Grid Systems Modelling with MicroGridsPy

Hands-on 2

GitHub repository for source code:

SESAM-Polimi/MicroGridsPy-SESAM: MicroGridsPy - SESAM-PoliMi (github.com)

MicroGridsPy is an open-source project, currently under active development, check out the detailed online documentation for usage guidance and updates:

https://microgridspy-documentation.readthedocs.io/en/latest/

Learning outcomes

By the end of this exercise, you will learn how to:

1) Interact with the MicroGridsPy User Interface

2) Explore the default configuration

3) Complete the first RUN

Important requirement

Ensure that MicroGridsPy is correctly installed and functioning within the dedicated conda environment. It will be also required a stable internet connection. Additionally, it is suggested (though not strictly required) to have Gurobi installed for improved performance of the model optimization. Refer to the installation procedure and guidance provided in Hands-on 1 for detailed instructions.

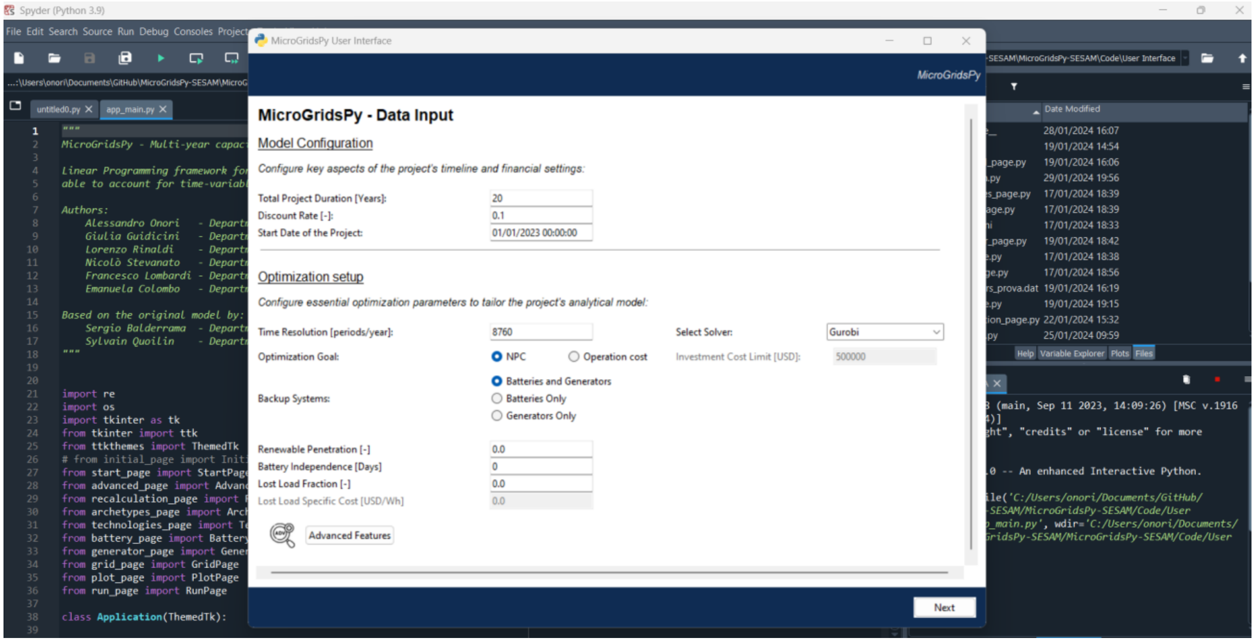
Initial Configuration Page

Launch the Application: Recap

Let’s briefly recap how to launch the user interface after having correctly creating the MicroGridsPy environment using conda and downloading the MicroGridsPy folder.

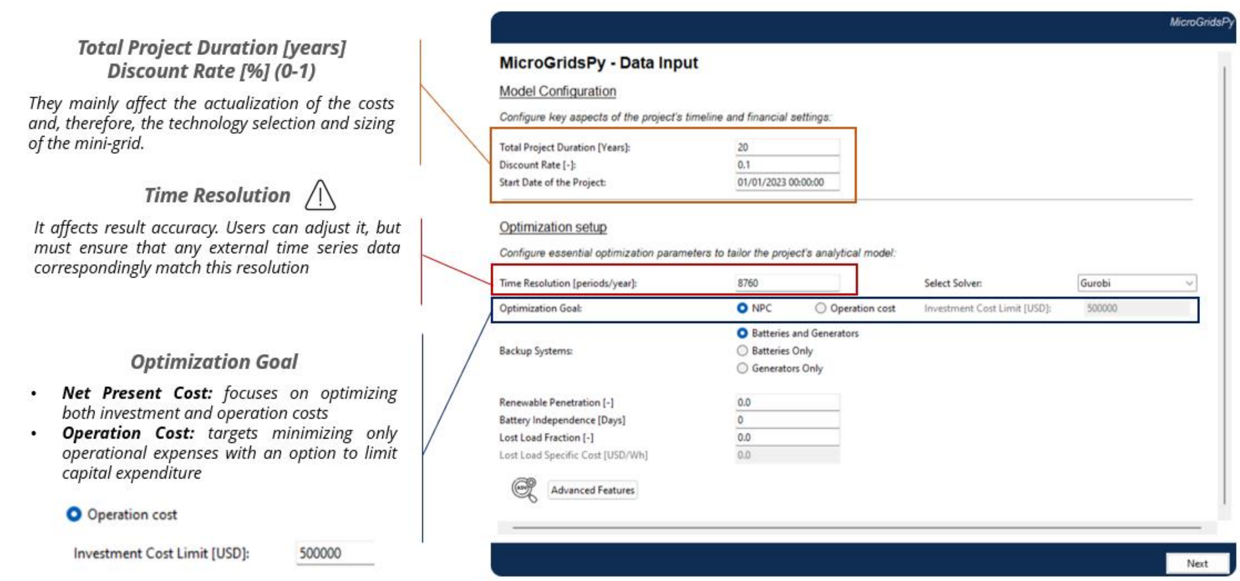
- Launch Spyder Using Anaconda. To launch Spyder, an integrated development environment (IDE) for Python, users can proceed this in two ways:
    1. Using Anaconda Prompt: Open the Anaconda Prompt, activate the mgpy environment (if not already activated), type the command *spyder* and press enter to open Spyder.
    2. Using Anaconda Navigator: Launch Anaconda Navigator, activate the mgpy environment from the "Environments" tab. Then, in Anaconda Navigator's "Home" tab, find "Spyder" in the list of available applications and click "Launch" to open Spyder.
- Launch the Interface within Spyder: Once Spyder is open, follow these steps to launch the MicroGridsPy interface:
    1. Check if the default layout is suitable; if not, set it from the "View" button.
    2. Locate the MicroGridsPy working folder using the Spyder interface's "File" menu.
    3. Open the project folder by double-clicking on it.
    4. Navigate to the "Code/User Interface" folder within the project folder.
    5. Look for the app_main.py file and double-click on it to open it.
    6. Run app_main.py by pressing F5 or using the "Run" button in Spyder's toolbar.

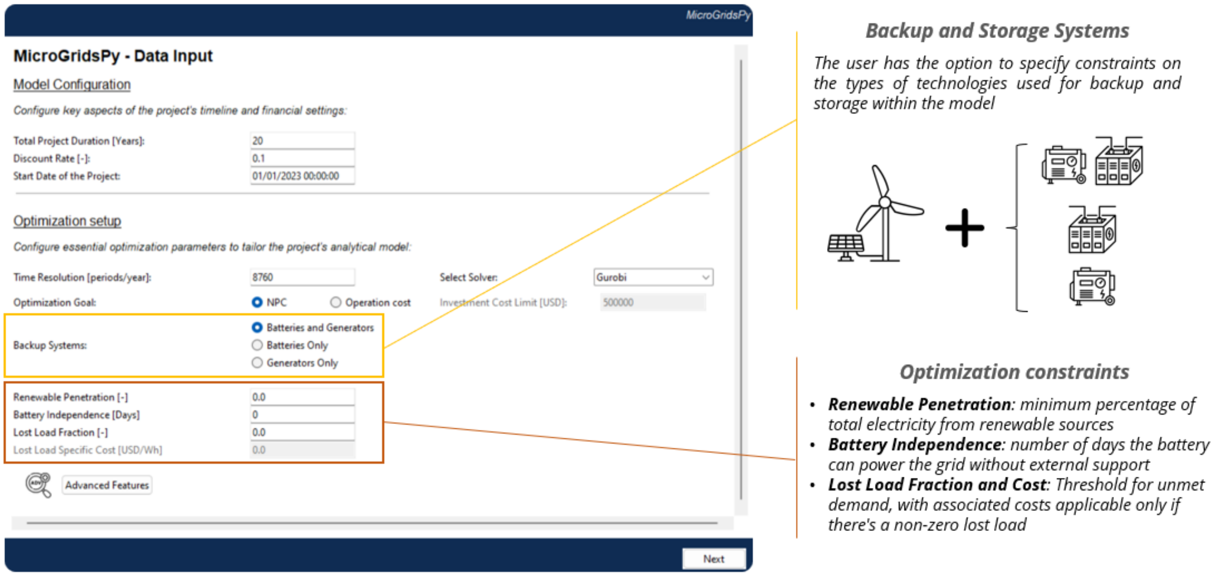
Upon completing these steps, the MicroGridsPy interface will launch within Spyder resulting in this situation:



Setting Configuration Parameters

Begin your data input journey by specifying fundamental parameters for your mini-grid project. This initial page serves as the central hub for configuring parameters and functionalities including duration, resolution, optimization goal, specific constraints and more. Here below a summary of the basic configuration parameters:



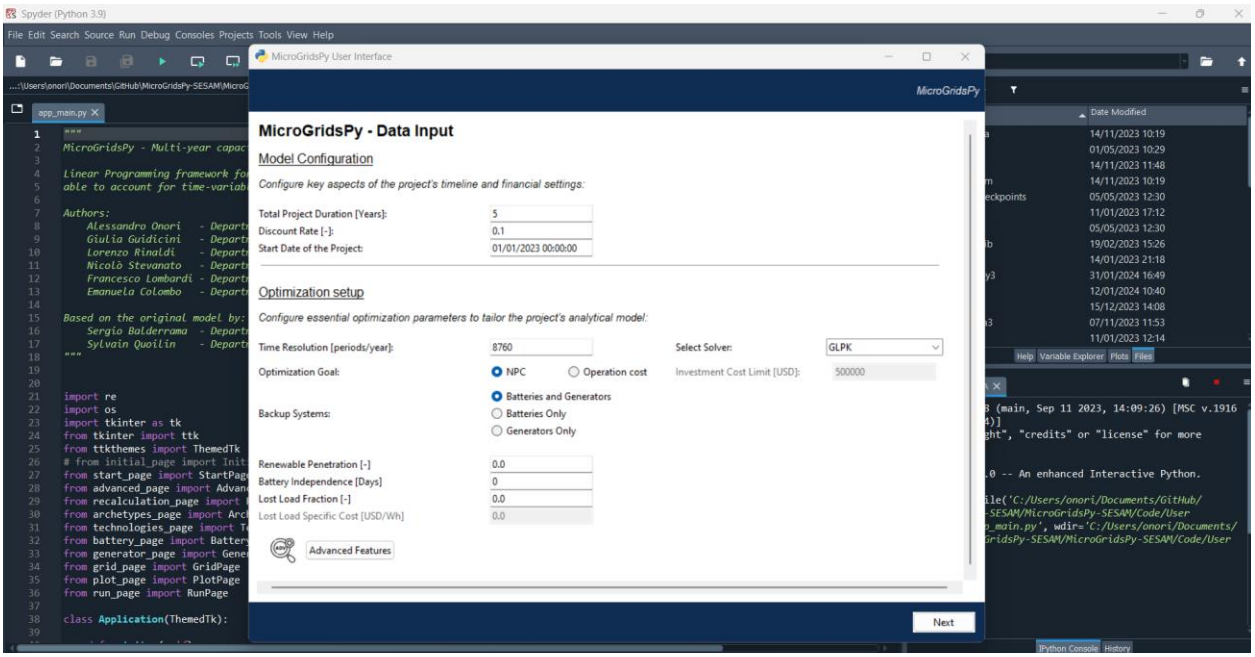


For the purpose of this hands-on exercise, our objective is to familiarize ourselves with the MicroGridsPy interface and its fundamental functionalities by testing the default configuration parameters. These default settings provide us with a comprehensive overview of the user interface and the construction of mini-grid models. The default configuration defines a project spanning a 20-year time horizon, utilizing a 10% discount rate, and operating at an hourly resolution. The primary optimization goal is to minimize net present costs (NPC). In terms of backup systems, the model will consider both batteries and a generator. Additionally, there are no additional constraints imposed, such as minimum renewable penetration or requirements for battery independence. This approach will help us gain a broad understanding of how to work with the interface and build microgrid models effectively.

Note

If you have Gurobi installed with an activated license, you can choose to leave Gurobi as the default solver for your MicroGridsPy simulations. This solver is known for its capabilities and speed, which can significantly expedite the optimization process.

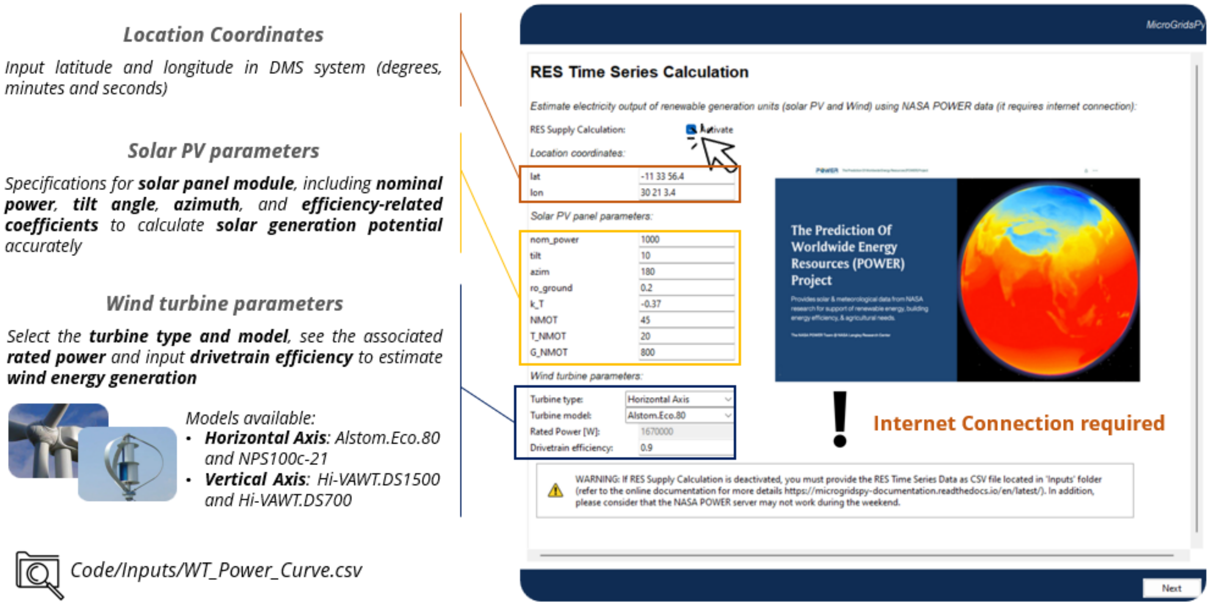
However, if you don't have a Gurobi license, you can opt to use the open-source free-to-use solver option, "GLPK," but making a specific adjustment. Please select "GLPK" from the solver dropdown menu and set the time horizon to 5 years. While this configuration may result in slight variations in the results (though not significantly), it substantially reduces the computational effort and the time required for solving. You can expect the solving time to be approximately 45-60 minutes (depending on your system computational performace), striking a favorable balance between result accuracy and computational efficiency.



As this is the default configuration, we will not delve into advanced features at this stage. Simply proceed by clicking "NEXT" and accepting the default settings. Please note that the application will display a confirmation message to ensure your intention, as it will not be possible to return and modify the model configuration at a later stage.

RES Time Series Calculation Page

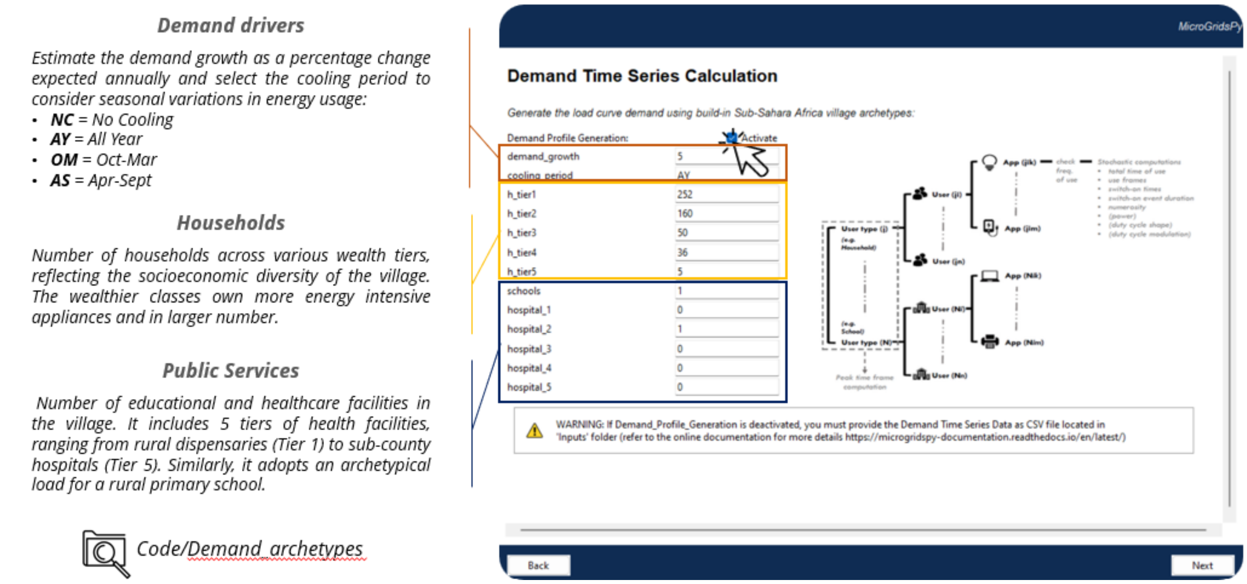
The RES (Renewable Energy Sources) estimation features within MicroGridsPy provide a convenient and comprehensive way to estimate renewable energy potential using NASA Power Project database. By simply checking the "activate" check button, all relevant parameters will be enabled, allowing users to tailor their RES estimation. The layout offers a scrollable area for a comprehensive list of parameters, including latitude, longitude, time zone, and turbine information. Custom fonts and tooltips enhance the user experience, making it a user-friendly interface for setting up required parameters. In the default configuration, the system is set to evaluate a specific location in Zambia, taking advantage of its solar-rich environment. The configuration includes a 1 kW solar PV module with standard values for parameters such as tilt angle, ground reflectivity (ro_ground), and operating temperature. Additionally, the horizontal axis Alstom.Eco.80 wind turbine, rated at 1670 kW, is integrated into the assessment.



Press “Next” to move forward.

Demand Time Series Calculation Page

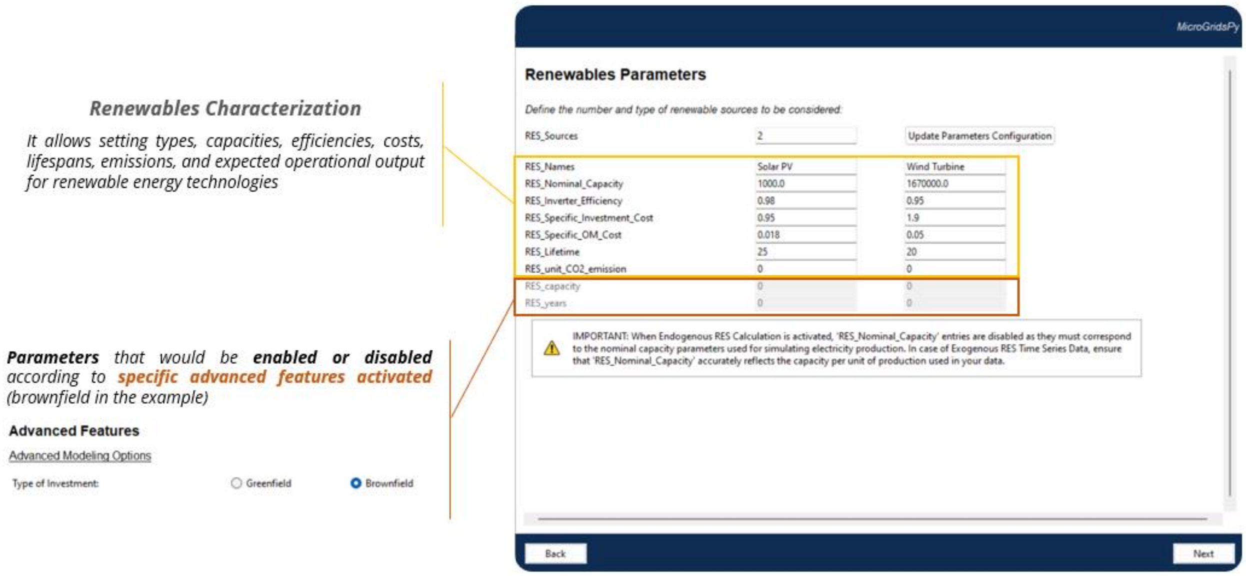
The Demand Time Series calculation feature offers a powerful tool for modeling load demand in a Sub-Saharan village. By activating the "activated" checkbox, users gain access to a range of parameters related to drivers, households, and public services to simulate households demand profiles by means of built-in archetypes referring to rural villages in Sub-Saharan Africa at different latitudes. Relevant parameters are composed of different types of end-users like households according to the wealth tier (i.e., from 1 to 5), hospitals with the same wealth scale and schools. The possibility for demand growth and specific cooling period are also integrated within this feature.



In the default configuration, the simulation focuses on a village comprising 503 households, distributed across various tiers. Additionally, the model accounts for 1 school and 1 health facility, both subject to a 5% annual growth rate. Press “Next” to move forward.

Renewables Parameters Page

Configure the available renewable energy sources and their techno-economic properties. The page defines default parameters and their initial values for renewable energy sources (RES), manages user input for RES parameters with validation and updates, creates labels, entry fields, and tooltips for RES parameters.



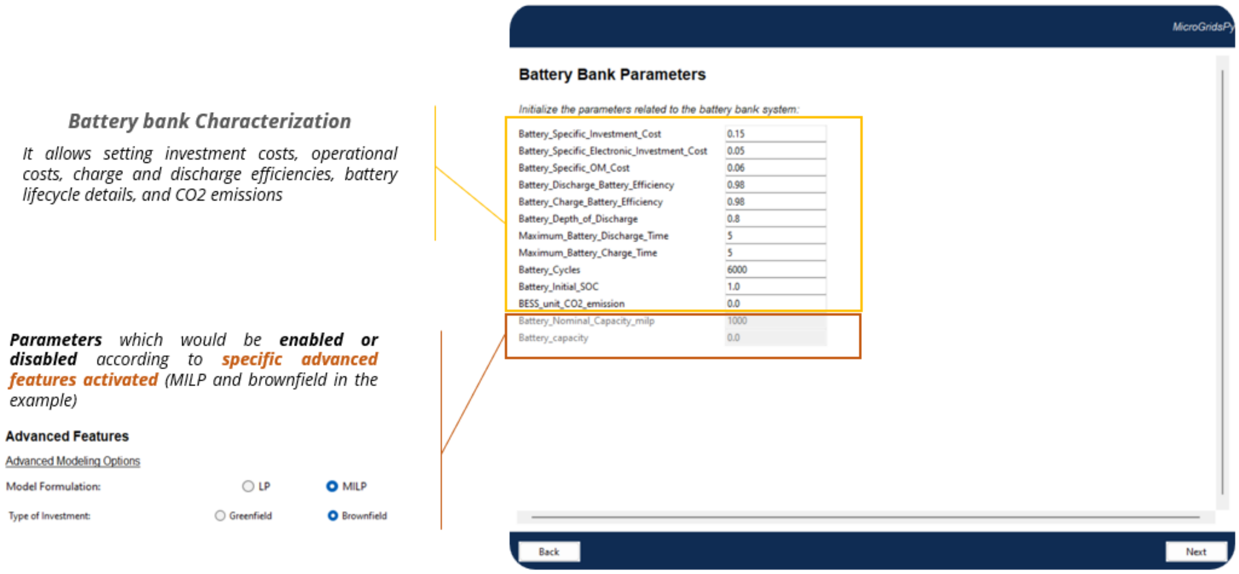
In MicroGridsPy renewables are considered as a general technology characterized by a set of parameters. Users have the freedom to create different "types" of renewables, and this flexibility is entirely arbitrary, allowing for customization based on project needs. Notably, within the interface, users can add an unlimited number of entries for a single parameter, and they have the liberty to define the names of these renewables as they see fit.

However, it's important to note that if users opt to utilize the built-in RES time series estimation, the model will primarily consider solar and wind as the available renewable sources. Therefore, in this scenario, users can define parameters for up to two renewable sources: one for solar and another for wind, as reflected in the interface.

In the default configuration, we simulate average data for a photovoltaic (PV) module and a wind turbine. It's essential to mention that the cost values provided in this default setting are purely exemplative and do not represent realistic cost figures. These default parameters serve as a starting point for users to explore and customize renewable technologies within the MicroGridsPy interface. Press “Next” to move forward.

Battery Bank Parameters Page

Set up battery-related parameters. It provides robust input validation, tooltips for parameter descriptions, and conditional parameter handling, ensuring data accuracy and usability.

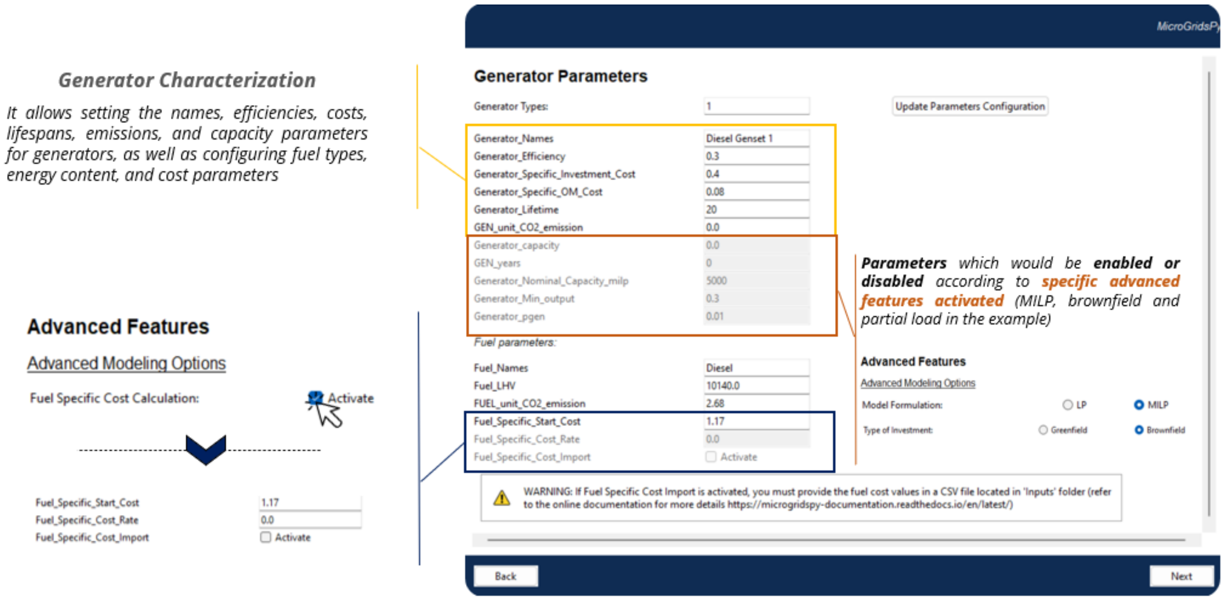


The Battery Bank Parameters page in MicroGridsPy offers users the capability to define parameters for a single type of battery bank unlike renewables, which can have arbitrary types. In the default configuration, the provided values serve as exemplative representations, which could correspond to a lead-acid battery type.

These default parameters provide a starting point for users to work with a standard battery type within the MicroGridsPy interface. Press "Next" to move forward.

Generators Parameters Page

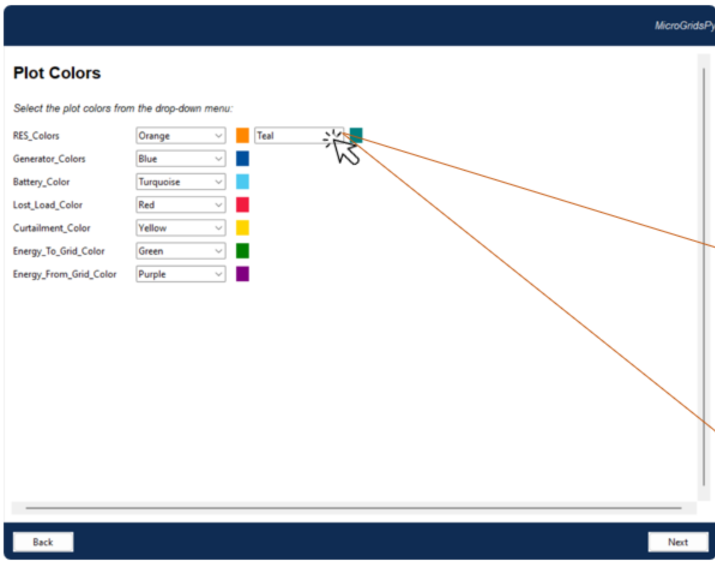
Define generator types and characteristics. Similarly to renewables, the page defines default parameters and their initial values but the user can add new entries for different types of generators. It offers strong input validation, parameter description tooltips, and conditional parameter management.

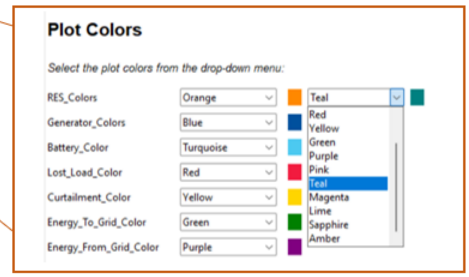


Users can create distinct generator types, specifying different fuels, capacities, and other relevant parameters to accurately simulate their scenarios. This flexibility allows users to model a wide range of generator configurations to suit their specific project requirements. In the default configuration, exemplative values are provided, which can be seen as representative of a classic diesel generator. Press “Next” to move forward.

Plot Colors Page

Visualize the color codes for data visualization by means of a dynamic color legend. These parameters are used for the aesthetic aspects of model outputs, assigning colors to different energy sources, storage options, and other model components for visual representation in plots and charts.

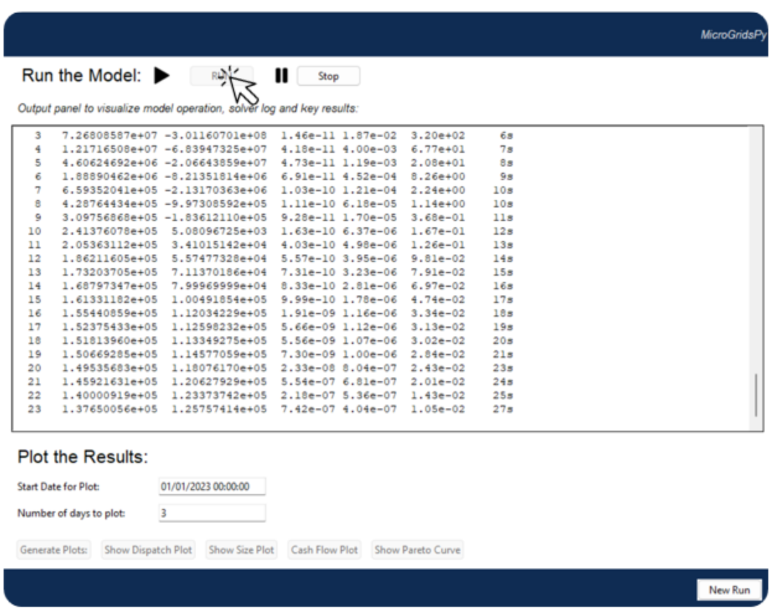




Users can **select colors** from a **dropdown menu** to represent different components such as renewable energy sources (RES), generators, batteries, lost load, curtailment, and energy exchanged with the grid, enhancing the **clarity and distinction of various data sets** in graphical representations

Running the model

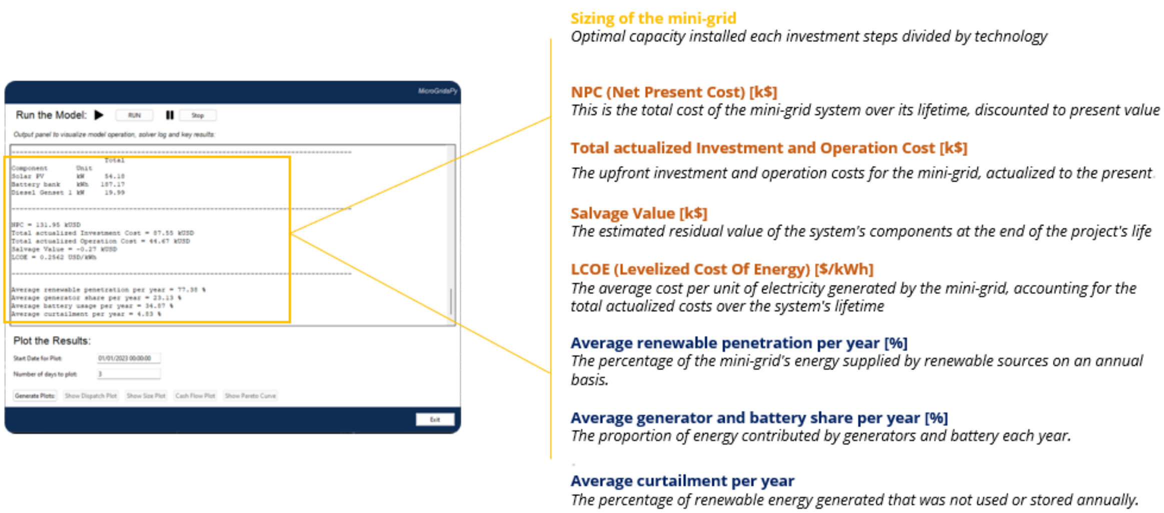
Finally, save your input data and initiate the optimization process. It includes the following functionalities: updating and displaying output messages in a text widget, showing dispatch, size, and cash flow plots in separate windows, generating plots based on user inputs and running a model in a separate thread, displaying progress messages and results.



Upon initiating the model with the 'Run' button, users can **monitor the solver's log** and the **system's operational actions** in **real-time**. This output panel provides a **live feed** of the **model's processing data** displaying also potential errors messages.

Key Results

Key results include a breakdown of the system components and various performance metrics. It outlines the installation capacities of solar panels, batteries, and generators, and presents key financial indicators such as net present cost, total investment and operation costs, salvage value, and the levelized cost of energy. It also provides percentages for renewable energy usage, generator and battery contributions, and unused renewable energy.

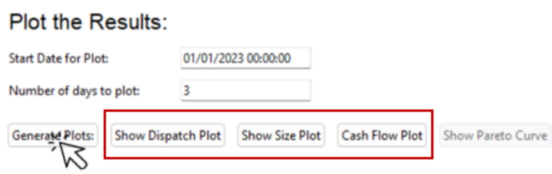


Explore and take note of the key results.

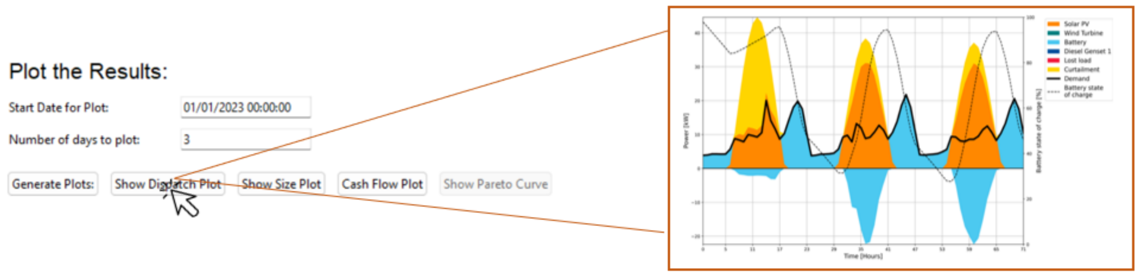
Visualizing Plots

Additionally, the interface offers functionalities to visualize the data through graphs based on selected parameters.

Displaying plots



Users can create visual **plots of model outputs** by setting a date range and choosing from various plot types. The buttons **generate graphs for energy dispatch, system sizing and financial analysis**, providing a comprehensive view of the model's performance over the selected timeframe.



The model not only computes various financial and operational metrics but also saves all results in the "Code/Results" folder for record-keeping and further analysis. For visual representations, updated plots are stored in the "Code/Results/Plots" folder, allowing for a graphical overview of the system's performance. It is important to note the date on these files, as not all graphs are newly created after each model run. Some plots might be from previous simulations with different configurations, and users should be mindful to differentiate between current and older data to ensure accurate interpretations.

Save the dispatch plot for later.

Once you have reviewed and saved the necessary data, you can close the application by clicking the 'X' at the right top corner.

Well done on completing the hands-on 2!