

Off-Grid Systems Modelling with MicroGridsPy

Hands-on 3

GitHub repository for source code:

[SESAM-Polimi/MicroGridsPy-SESAM: MicroGridsPy - SESAM-PoliMi \(github.com\)](https://github.com/SESAM-Polimi/MicroGridsPy-SESAM:MicroGridsPy-SESAM-PoliMi)

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/>

References:

[1] N. Stevanato, R. Mereu, A. Brasco, A. Muumbo, A. Onori and E. Colombo, Access to Electricity in Informal Settlements: Literature Review and Load Curve Estimation, IEEE PES/IAS PowerAfrica, Marrakech, Morocco, 2023, pp. 1-5, doi: 10.1109/PowerAfrica57932.2023.10363327.

Learning outcomes

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

- 1) Explore the MicroGridsPy User Interface functionalities
- 2) Understand how to initialize and input load demand in MicroGridsPy
- 3) Understand how load demand affects results

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.

Introduction

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:

Hands-on objective

In this hands-on session, we will delve into the Load Demand page of MicroGridsPy to master its interface through two key activities: simulating endogenous demand and providing exogenous demand data. Our goal is to understand how to manipulate and analyze load demand within the software effectively.

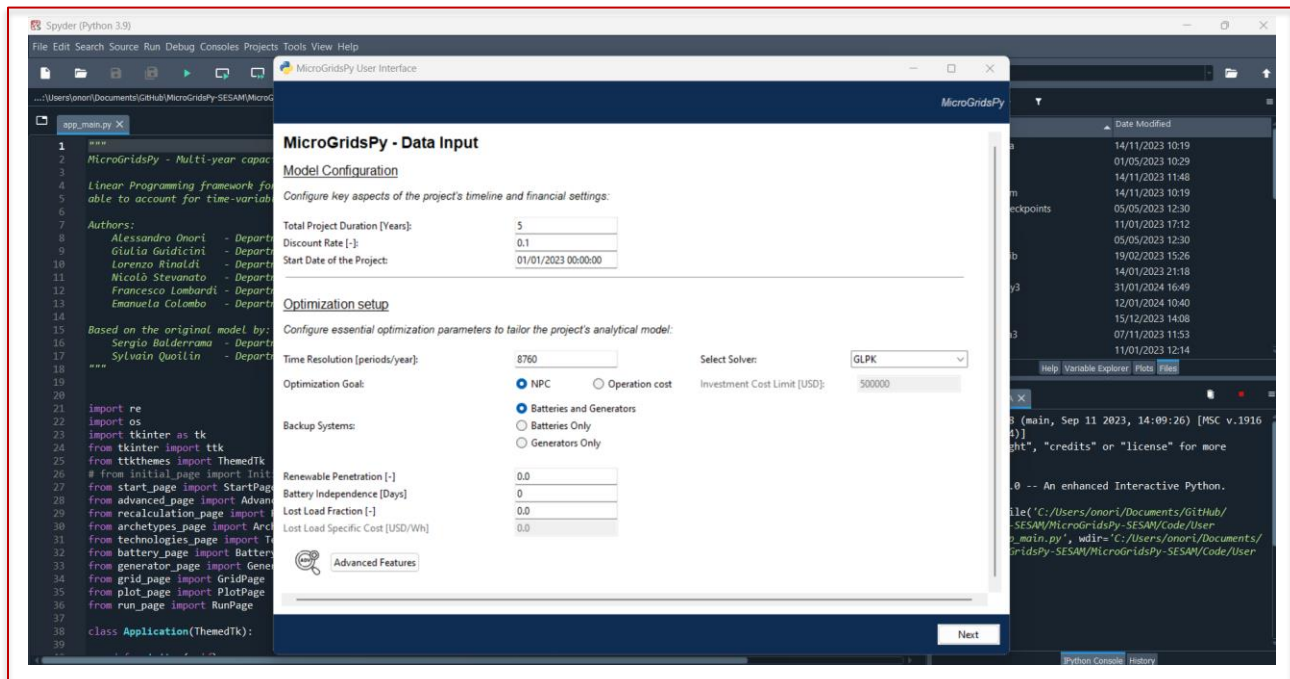
Default configuration

To recap, the default configuration we'll work with includes a 20-year model horizon, hourly operation, a 10% discount rate, and a focus on minimizing net present costs. The model is allowed to consider both batteries as storage system and generators as backup system and there are no additional constraints implemented. The exemplative case study focuses on Sub-Saharan village in Zambia with an annual growth rate of 5%, leveraging NASA Power Project data for resource assessment (specifically solar and wind). Technology characterization includes photovoltaic modules, wind turbines, battery banks, and diesel generator with exemplary but not realistic parameters.

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 performance), striking a favorable balance between result accuracy and computational efficiency.



Endogenous Demand Time Series

The Demand Time Series calculation feature offers a powerful tool for modeling load demand in a Sub-Saharan village. By activating the "activate" checkbox, users can simulate demand profiles and built-in archetypes referring to rural villages in Sub-Saharan Africa at different latitudes. These 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.

Demand drivers

Estimate the demand growth as a percentage change expected annually and select the cooling period to consider seasonal variations in energy usage:


- **NC** = No Cooling
- **AY** = All Year
- **OM** = Oct-Mar
- **AS** = Apr-Sept

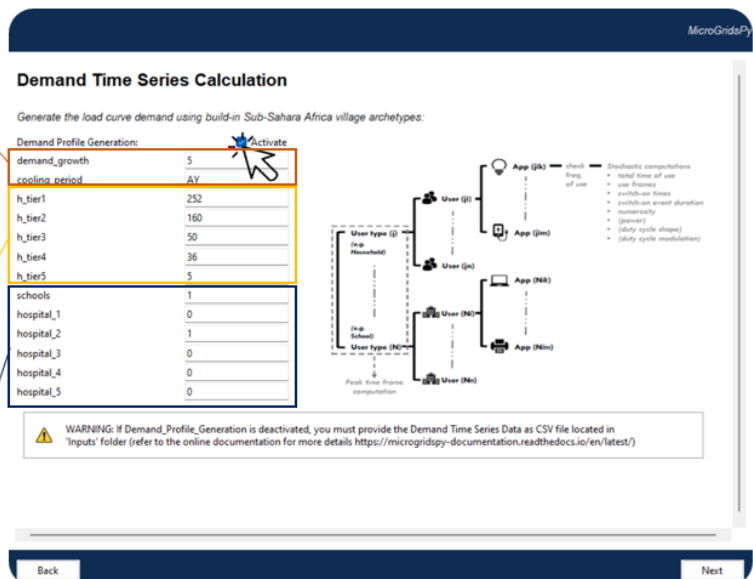
Households

Number of households across various wealth tiers, reflecting the socioeconomic diversity of the village. The wealthier classes own more energy intensive appliances and in larger number.

Public Services

Number of educational and healthcare facilities in the village. It includes 5 tiers of health facilities, ranging from rural dispensaries (Tier 1) to sub-county hospitals (Tier 5). Similarly, it adopts an archetypical load for a rural primary school.

 [Code/Demand_archetypes](#)



Demand Time Series Calculation

Generate the load curve demand using built-in Sub-Sahara Africa village archetypes:

Demand Profile Generation: ☒ **Activate**

demand_growth	5
cooling_period	AY
h_tier1	252
h_tier2	160
h_tier3	50
h_tier4	36
h_tier5	5
schools	1
hospital_1	0
hospital_2	1
hospital_3	0
hospital_4	0
hospital_5	0

WARNING: If Demand_Profile_Generation is deactivated, you must provide the Demand Time Series Data as CSV file located in 'Inputs' folder (refer to the online documentation for more details <https://microgridpy-documentation.readthedocs.io/en/latest/>)

Back Next

All load demand archetypes, including 100 for households, 5 for the health center, and 1 for a rural school, are accessible in the dedicated folder located at [Code/Demand_archetypes](#).

Exercise 1

In this exercise, you will simulate a different load demand scenario in MicroGridsPy to master the usage of the interface and understand the impact of load demand on results. Your task is to configure the model for a community with 2200 households without electricity in a rural area near Ilembula (Tanzania). The households can be divided into 4 groups based on their appliances:

- 700 households in Tier 1
- 1000 households in Tier 2

- 245 households in Tier 3
- 255 households in Tier 4

Additionally, simulate demand for 5 primary rural schools, and for 4 health facilities (Tier 2). The annual population growth rate is estimated at 5.5% and cooling period can be assumed as AY (all years).

Approximate location coordinates (GPS system): -8.896247572521382, 34.58168323510706 where the first -8.896247572521382 represents the latitude, and 34.58168323510706 represents the longitude. In DHS system: -8° 53' 46.968" , 34° 34' 53.8".

Once set the described (exemplative) scenario, run the model while keeping other parameters that are NOT affected by this new scenario (not necessary all) unchanged. Refer to the previous hands-on guides for detailed steps on running the model.

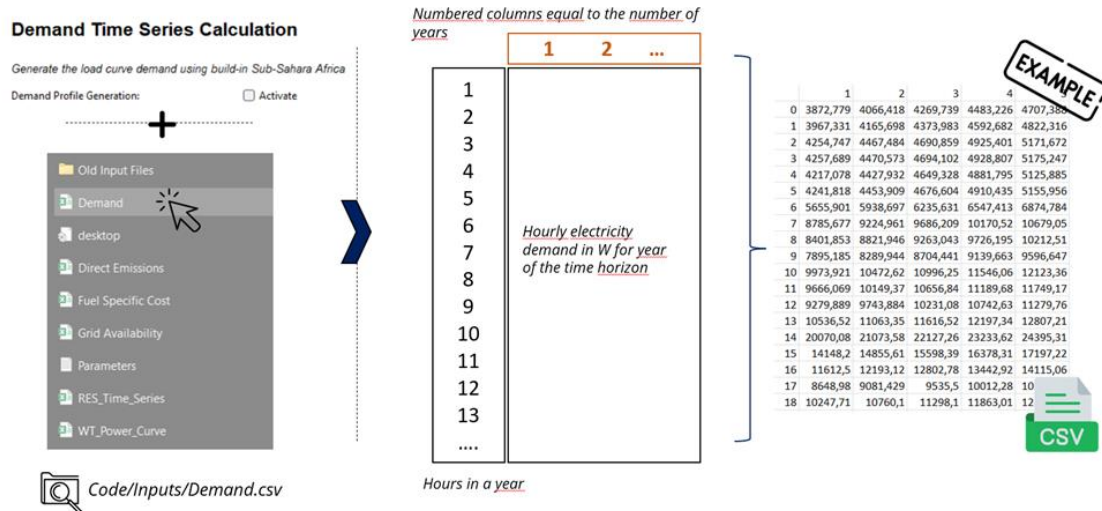
After the optimization process, take note of key results and the relevant plots generated by MicroGridsPy.

Exogenous Demand Time Series

In MicroGridsPy, load demand can be assessed through two approaches. Endogenous estimation leverages built-in archetypes reflecting typical energy usage patterns in Sub-Saharan Africa, automating the generation of load profiles within the software. On the other hand, exogenous demand allows users to input their own specific demand data, offering the flexibility to model energy needs based on external or pre-existing datasets.

To provide exogenous demand in MicroGridsPy:

- Begin by ensuring that the 'Demand Profile Generation' check box for demand estimation in the user interface is NOT activated.
- Locate the Demand.csv file within the Code/Inputs directory of your MicroGridsPy folder.
- Edit this CSV file, adhering to the structure shown in the picture below: numbered columns for each year of the time horizon, and hourly demand values in watts for each row corresponding to the hours in a year. After editing the file with your specific demand data, save the changes.



- Return to the MicroGridsPy interface and proceed by clicking 'Next' to continue with the simulation using your provided demand data.

Exercise 2

In the "Materials" folder you can find a "Demand.csv" file. It describes the load demand (households and commercials) over 20 years of an informal settlement in Nairobi (Kenya) without legal electricity connection [1]. Use that file to provide exogenous load demand time series data as described above. The approximate location coordinates are: $-1^{\circ} 18' 41.92''$, $36^{\circ} 51' 54.72''$.

Note

Please be aware that the filenames within the Code folder of MicroGridsPy, especially input files like 'Demand.csv', must remain unchanged. The software's code references these files by name, and altering them can disrupt the ability to correctly access and process the necessary data. This is a crucial aspect of the model's structure for accurate and efficient functioning.

Once set the described (exemplative) scenario, run the model while keeping other parameters that are NOT affected by this new scenario (not necessary all) unchanged. Refer to the previous hands-on guides for detailed steps on running the model.

After the optimization process, take note of key results and the relevant plots generated by MicroGridsPy.

Managing csv files

When working with CSV (Comma-Separated Values) files, you may need to properly format or manipulate the data in a spreadsheet application like Microsoft Excel, or using a simple text editor.

Modifying CSV File for Decimal Formats

Problem: Different regions use different decimal separators (e.g., some use a comma ,, others use a period .). This can cause issues when importing data into certain programs.

Solution with Text Editor:

1. Open the CSV file in a text editor that supports "find and replace" (e.g., Notepad++, Sublime Text).
2. Use the "find and replace" feature:
 - Find "," and replace it with "." if you need to change the decimal separator from a comma to a period. Alternatively, find "." and replace it with "," for the opposite
3. Save the file after making the changes.

Additional Tip

If the file Demand.csv requires dots for float numbers, ensure that your regional settings in Excel or your text editor match this requirement. For instance, if your Excel is set to use a period as the decimal separator, it might incorrectly interpret the data. The "find and replace" method in a text editor can be particularly handy in such situations to ensure the file adheres to the required format.

Remember, when using "find and replace," be cautious as it can unintentionally change data if not used precisely (for example, changing commas in text fields). Always make a backup of your original file before performing such operations.

Well done on completing the hands-on 3!