# Modelling of stand alone micro-grid using hybrid energy storage: Solar – Battery - Hydrogen

- Department of Energy Engineering -

Thesis Report

HYTEC4-1010
Foivos Maniatis
Ignacio Javier Moreno Roche
Pedro Miguel Fernandes de Carvalho

Aalborg University
Department of Energy Engineering

Here you can write something about which tools and software you have used for typesetting the document, running simulations and creating figures. If you do not know what to write, either leave this page blank or have a look at the colophon in some of your books.

**Electronics and IT**
Aalborg University
http://www.aau.dk

**AALBORG UNIVERSITY**
STUDENT REPORT

**Title:**
Modelling of stand alone micro-grid using hybrid energy storage Solar – Battery - Hydrogen

**Theme:**
Scientific Theme

**Project Period:**
Spring Semester 2023

**Project Group:**
HYTEC4-1010

**Participant(s):**
Foivos Maniatis
Ignacio Javier Moreno Roche
Pedro Miguel Fernandes De Carvalho

**Supervisor(s):**
Torsten Berning
Samuel Simon Araya

**Copies:** 1

**Page Numbers:** 121

**Date of Completion:**
June 2, 2023

**Abstract:**

This project is the final semester master thesis for the HYTEC specialisation in the Energy engineering department. The aim of this project is the development of an automated tool for assessing the economic feasibility of hybrid hydrogen microgrids based on solar power.
For this purpose, an interface was generated capable of generating consumption data for different commercial and residential applications. An artificial intelligence algorithm for ensuring the size of the components adequate for maintaining the energy demand and plot the economics for 20 years of operation. Finally, the energy flows in the system are optimised, simulating the real operation of a microgid with hybrid energy storage.

# Contents

# Summary

The present master thesis presents a comprehensive investigation of the sizing , simulation, energy flow optimisation and economic feasibility of hybrid energy storage hydrogen microgrids powered by solar energy. The study begins by establishing the motivation and the problem formulation that underpin this research, setting the stage for a detailed exploration into the current state of hydrogen microgrid technology.

The research meticulously examines essential model components such as Photovoltaic panels (PV), batteries, electrolyzers, hydrogen storage, and fuel cells, aiming to encapsulate the latest advancements and technologies in the field. Subsequently, the study elucidates the assumptions for the system modelling and delineates the methodology adopted for the corresponding calculations. This involves an in-depth review of solar irradiation libraries, load curves, fuel cell and electrolyzer models, as well as economic models. In addition, the computational methods for the sizing and energy flow optimisation are also examined in order to identify the most suitable algorithm for each method.

The study then presents the results derived from system sizing and the evaluation of economic indicators. It juxtaposes four cases to discern the most economically viable and sustainable options for hybrid hydrogen microgrids. Those cases being a communal facility in Kenya, a house in the Netherlands, a resort in the Maldives and a data centre in Greece. Progressing further, the research compares non-optimised and optimised energy management systems, delineating energy flows within the system and examining the performance outcomes of the optimisation process.

Finally, the thesis concludes by synthesising the key findings and offering closing remarks. Potential areas for additional study and a comprehensive discussion on the implications of the research are also provided. The present work explores the intersection of renewable energy, AI, and economic viability, contributing significantly to the understanding and application of hybrid hydrogen microgrids.

# Preface

This master thesis is the cumulation project for the HYTEC specialisation in the Energy Engineering department. Presented is the exploration of the economic feasibility of hybrid hydrogen microgrids powered by solar energy.

In the face of growing energy demands and increasing environmental concerns, the development of reliable and sustainable energy systems has become a global priority. Harnessing solar energy, coupled with hydrogen technology for energy storage, provides a promising approach. To advance this technology, a comprehensive understanding of its economic viability is crucial.

In this research, we propose an automated tool that assesses the economic feasibility of hybrid hydrogen microgrids. The tool features a novel interface for generating consumption data across commercial and residential sectors, and an AI-based algorithm for determining optimal component sizing and predicting the economics for a 20-year operation period. Lastly, the thesis elucidates an energy flow optimisation strategy within the system, mirroring a real-world microgrid operation.

It is the author's sincere hope that this work contributes to the field of energy engineering, particularly in the understanding and application of hybrid hydrogen microgrids. Hereby we attach the GITHUB repository containing the code that was developed for this project.

URL: `https://github.com/Heraclitus2/Heraclitus2`

We would like to express our gratitude to Torsten Berning and Samuel Simon Araya, whose guidance and insights have been instrumental in the execution of this project.

Aalborg University, June 2, 2023

Foivos Maniatis
<fmania21@student.aau.dk>

Ignacio Javier Moreno Roche
<imoren21@student.aau.dk>

Pedro Miguel Fernandes de Carvalho
<pferna21@student.aau.dk>

# Chapter 1

# Introduction

## 1.1 Background and motivation

In recent years, there has been an ever growing pursuit of energy independence, both for residential and industrial applications. The main reason behind this is the ability to avoid both sudden and continuous increases in electricity prices, along with gas for heating, and the regulations and restrictions that may come to energy usage due to different circumstances.

Some factors, like the Russia-Ukraine conflict that began in 2022, prove that these changes can come unexpectedly, with great impact in the everyday life of families and businesses all over the world, namely in Europe in the case of the aforementioned conflict. These events with global impacts can create power and resource shortages that not only affect prices, but also availability, with certain governments and organizations threatening to limit the allowed electricity consumption and gas usage [86]. These energy price changes can be seen in graph 1.1, where 100 represents the index value for oil, coal, gas and electricity on the 23rd of February, 2022.

Furthermore, even though countries are making an effort to move the global energy grid towards the use of renewable energy sources, a great share of the total energy use is still non-renewable and extremely polluting. A great example of this is Denmark, which, despite being on the forefront of energy development and a big advocate for renewable energy sources, still uses a large percentage of oil, coal and natural gas on their energy supply, as can be seen in figure 1.2.

Considering this, a lot more energy users are beginning to look into transitioning into more environmental friendly energy sources, not only for the obvious benefits that brings to the world, but also in order to be eligible to receive some of the financial assistance that governments all over the developed world are beginning to offer to businesses and individuals willing to invest in less polluting technologies.

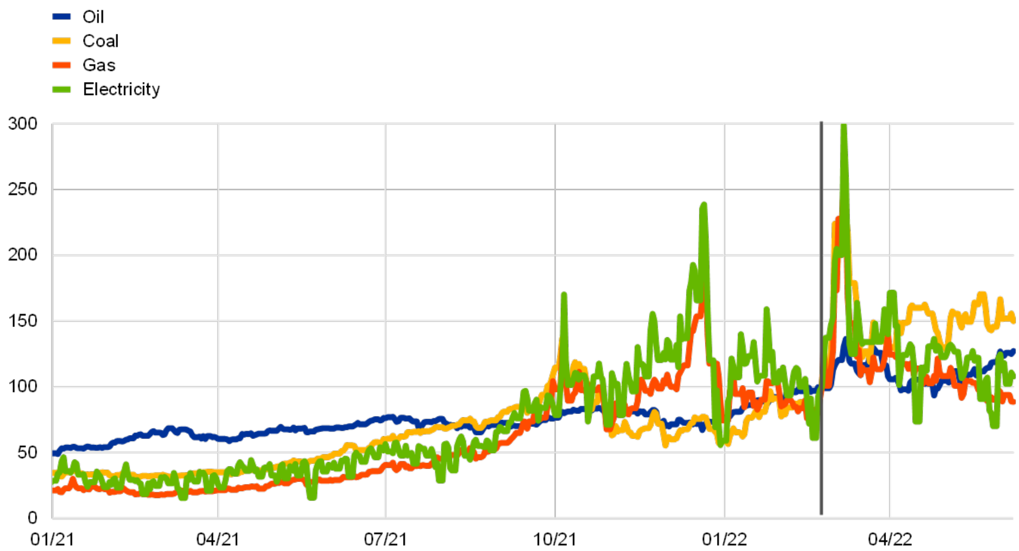



**Figure 1.1:** Changes in energy prices with the Russia-Ukraine conflict [41]

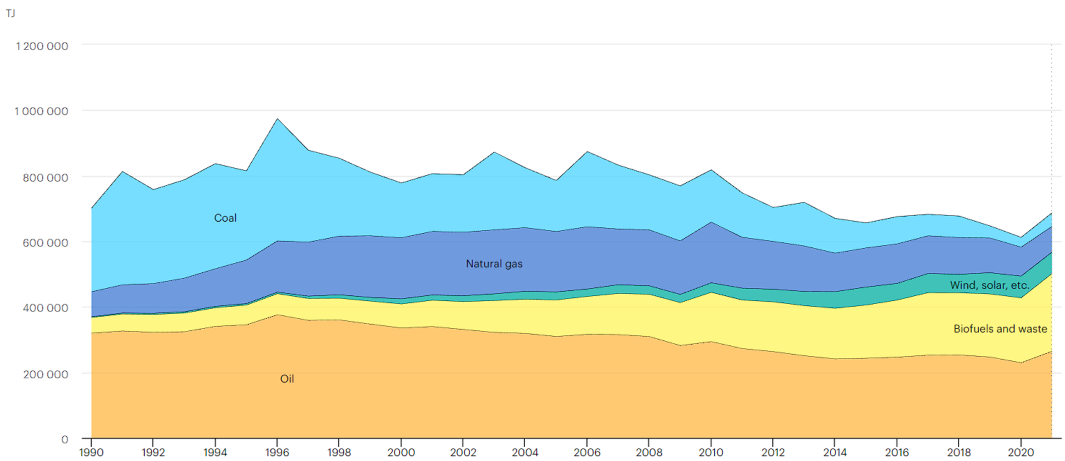



**Figure 1.2:** Total energy supply (TES) by source, Denmark 1990-2021 [69]

Combining both the needs previously mentioned, the solution that can be studied and applied are hybrid micro-grids. These consist of extending a solar grid(consisting of a solar panel and battery) with further storage in the form of hydrogen. To accomplish this objective, the excess energy not used by the micro-grid once the battery is fully charged, is diverted into an electrolyzer in order to produce hydrogen gas that is then stored in a storage tank. This hydrogen is then used by a fuel cell to produce electricity whenever it is needed by the system. The hydrogen components of this micro-grid are now under great development and its use is fast increasing, making its application a great investment in the future.

By applying solar/hydrogen hybrid micro-grids, the direct emissions of energy production are zero, and with the correct sizing of its components, its possible to achieve full energy independence. However, in order to obtain this accurate sizing, full consideration of the energy needs of the application, the solar irradiation on the location selected, efficiencies of the different components at different work conditions and many more factors need to be taken into account, while also trying to minimize the cost of such a system.

## 1.2 Problem formulation

As the world shifts towards renewable energy sources, it's becoming increasingly important to design efficient and cost-effective microgrids that can provide reliable power to small communities and remote areas. However, current renewable microgrid technologies are limited by their short-term energy storage capabilities.

The hybrid microgrid comprises of solar, battery, hydrogen, and fuel cell technologies, to create a sustainable and long-lasting energy solution for small-scale microgrids. This project aims to develop a software that simulates the operation of such microgrids. By modeling and simulating a hybrid energy storage system that includes hydrogen, this software will provide valuable insights into the feasibility and economic viability of using hydrogen as a long-term storage solution for solar microgrids

Furthermore, it will enable researchers and engineers to optimize the design and operation of such systems, thereby facilitating the wider adoption of renewable energy technologies and helping to mitigate the impacts of climate change.

## 1.3 Project objective

In this project, a software will be developed from the ground up, using the programming language *Python*, with the purpose of sizing a hybrid micro-grid while minimizing its costs. In order for the code to run the desired simulation and provide results for the sizing of the components, it will require user inputs. The type of application (residential, commercial, industrial, etc.), previous or expected average energy consumption and peak power used by said application, and the location are used as inputs. The location will allow the extraction of the solar irradiation information from an online Application Programming Interface (API), while the average energy consumption and peak power will allow for the energy load profile to be approximated.

Following this first step, an initial guess for the sizing of the components of the hybrid micro-grid can be made. Using the values from this initial guess, an hourly simulation of the Energy Management System (EMS) can be made, in order to verify the given sizing achieves the technical needs of the micro-grid, in terms of obtaining energy from the environment using solar panels, and storing it with the use of both a lithium-ion battery and a hydrogen system consisting of electrolyzer, hydrogen storage and fuel cell. This simulation will also allow the calculation of the degradation of the components and necessary replacement of said components, giving the total cost of the energy system after the 20 simulated years. Considering that the goal of the sizing made by this software is to minimize the total cost of the components while maintaining the technical validity of the energy system, the simulation will be run again with different sizings of the micro-grid, to obtain the most cost efficient, functioning energy system for each application.

When the optimal solution is obtained, financial indicators can be calculated and included with the financial metrics already calculated in the simulation to provide a complete economical overview of the micro-grid given.

## 1.4 Project outline

The structure of the project is as follows:

In Chapter 1 the background and motivation are layed out, as well as the problem formulation.

Chapter 2 presents the state of the art regarding hydrogen microgrids and the selected modeled components. Those components are Photovoltaic panels (PV), batteries, electrolyzer, hydrogen storage and fuel cells.

Chapter 3 presents the assumptions used for the modeling and methodology of the calculations. Included are the solar irradiation libraries, load curves, fuel cell and electrolyzer models and economic models. In addition, the algorithms used for the sizing and the energy management system are presented. The selected case studies are also presented in Chapter 3.

Chapter 4 presents the results of the system sizing and the economic indicators. A comparison between the 4 selected cases is presented with a discussion about the performance.

Chapter 5 presents the comparison between the non optimised energy management system and the optimised energy management system. Included is a presentation of the energy flows and a discussion of the performance.

Finally, Chapter 6 concludes the project. Concluding remarks, additional study and discussion are presented.

# Chapter 2

# State of the art

In this chapter the different technologies simulated in the report are outlined. Firstly, the components of the microgrid and the underlying mechanisms are outlined, starting with solar irradiation and battery storage. Secondly, hydrogen components are described in detail, including different fuel cell and electrolyzer technologies, followed by hydrogen storage. Lastly, the implementation of microgrids using hydrogen storage and the methods for designing and simulating said systems is presented.

## 2.1 Microgrids and components

Microgrids have quickly grown in popularity over the years due to the rise in renewable energy technologies, and easier accessibility to said technologies, which can be seen, for example, in the large decrease in price of power originated via photovoltaic panels in comparison to other technologies, as represented in figure 2.1. This has lead to the implementation of solar and wind power micro grids in housing and industrial applications, allowing for off-grid living and energy storage. Due to the limitations when it comes to energy storage in batteries, hybrid microgrids have started to surface as a great alternative, offering the storage of energy in the form of hydrogen.

The use of micro grids presents several advantages over the use of regular grid energy. The first advantage is the complete independence from the centralized power grid, which protects the users from sudden increases in prices, power outages and power limits introduced by centralized authorities, whether it is to limit consumption for environmental reasons, or for natural/human catastrophes. At the same time, moving from a centralized process of power generation and storage to independent and local power production reduces the great power losses that come from the electricity being pushed over longer distances from the large scale power plants to the end user. This process of energy transmission can even

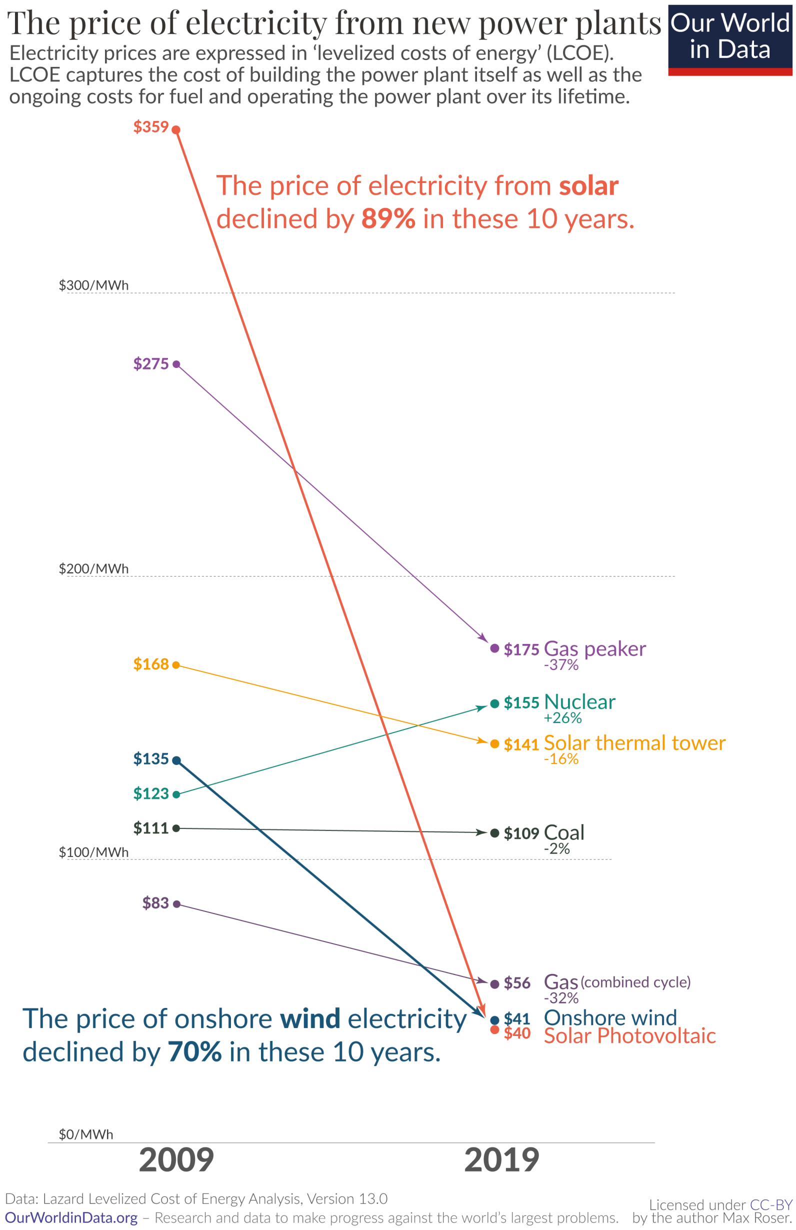



**Figure 2.1:** Variation in cost of electricity [95]

have energy losses of up to 10%, which in the grand scheme of things represents an enormous loss of energy [23]. Finally, even though energy grids around the world, and specially in the developed countries, are starting to move towards integrating more and more renewable energies as their power sources, a large scale of energy gets produced from non renewable, highly polluting energy sources. As such, users interested in reducing their carbon footprint and reducing their impact on the environment may find the prospect of having a fully renewable micro grid highly enticing.

A good example of the value of microgrids can be seen in the United States of America, where natural disasters are usually followed by wide and prolonged power outages, which lead to not only major inconveniences for the user, but also enormous economical losses. Microgrids could assist in protecting the economy and the users from these power outages [83].

Staying with the USA as an example, during the year 2022 in Texas households saw electricity prices go up by 50% when compared to the same period of the previous year. This was due to shortages in natural gas because of the Russia-Ukraine war, which lead to Texas having to export a lot of their natural gas production [111]. With price fluctuations this extreme and unexpected, grid independence can prove to be an invaluable asset provided by microgrids.

When it comes to what constitutes the power grid, it depends on what type of micro grid is considered. Independently, there are still three different groups that constitute a micro grid: firstly, the applications that consume the power. Secondly, the source of power generation that allows the micro grid to be off grid and decentralized, along with the energy storage. Lastly, the energy management system that controls the energy usage and functioning of the micro grid [37].

The application varies depending on the case study, it can be a house, a village or a factory. The main sources of power in renewable microgrids are solar energy through photovoltaic panels and wind power using wind turbines. As for the energy storage, for solar and wind power micro grids, usually batteries are used for storing power. However, these are very limited, so hybrid systems started being implemented, with alternative storage systems. One of these storage alternatives is the use of hydrogen, in which there is still a battery present for short term energy storage and peak shaving, but now there is an electrolyzer that uses the excess energy produced by the solar panels/wind turbines to produce hydrogen which is then stored in an hydrogen tank. When there are energy requirements that cannot be met by the solar panels or the battery, a fuel cell uses the hydrogen stored and transforms it to electricity.

## 2.2 Photovoltaic panels and battery

### 2.2.1 Solar irradiation

Solar irradiation analysis is essential for the proper design, study and sizing of an hybrid microgrid. The hours at which its available and power that it is capable of providing are key to not only the sizing of the solar panels, but the entire system as well, with limitations in area for solar panels possibly even leading to a technical limitation of the microgrid.

Irradiation is the process of receiving electromagnetic radiation from a source, and it represents the energy per unit of area of surface. In the specific case of solar irradiation, the source is the sun, and the recipient is judged as the earth, with a particular location being chosen dependant on where the microgrid is. There are 3 different components of solar irradiation [17], and the total coming from these components:

- Direct Solar Irradiance: radiation that travels from the sun to the surface of the earth unaffected by obstacles, without scattering or absorption. It is dependent on the atmosphere and the angle of incidence.
- Diffuse Solar Irradiance: resultant of the diffusion as the radiation crosses the atmosphere and the electromagnetic waves are scattered. Relatively uniform across the sky and with no specific direction.
- Reflected Solar Irradiance: radiation reflected on other surfaces that eventually affects the surface being considered.

The sum of these 3 components that are represented in figure 2.2 is the global solar irradiance, and is what is considered when studying the microgrid.

As mentioned before, the calculation of the irradiation is essential in order to properly size the photovoltaic system that powers the entire energy grid, with the patterns and profile of solar irradiation throughout the year being very important to control the rest of the microgrid.

### 2.2.2 Solar panels

The purpose of a solar panel is to use solar irradiation to obtain electricity, by using semi conductive materials, such as silicon that allow for the photovoltaic effect to happen. This photovoltaic effect happens when light shines upon a semiconductor, causing the electrons to be freed from their atoms and therefore generating a flow of electricity. Since this effect can only happen with the photons (fundamental particles of light and electromagnetic radiation) that are adsorbed by the semi conductor, the material surface of the solar panel is built with the goal of making the front surface of the cell more receptive to dislodged electrons, so the flow of electricity happens towards the surface [34].

As mentioned in the previous paragraph, an example of semiconductor is silicon, which is the component chosen to make the majority of solar panels in the market. This chemical element is the second most present in the Earth's crust, however, obtaining it in order to produce solar panels is still a difficult process due to silicon being bound to other elements in its natural state, meaning it requires a thorough process of separation in order to obtain it in its purest form. This

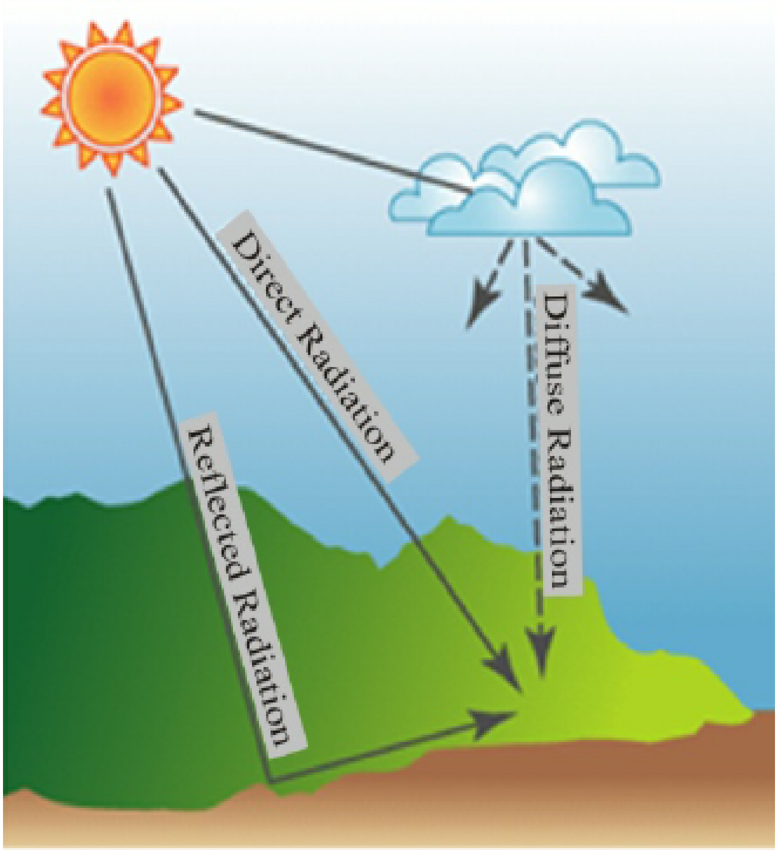



**Figure 2.2:** The three types of solar radiation [17]

separation process is where most of the cost of silicon production accrues from [105].

Having obtained the silicon, the way it is implemented can vary depending on its crystal structure, where it can be monocrystalline or polycristalline. Polycristalline panels have the great advantage of being significantly cheaper than monocrystalline panels. However, this price difference has steadily decreased over the years, making the technical differences between these crystal arrangements all the more relevant. Since monocrystalline panels have higher efficiency values, when this is taken into account along with the smaller price difference, this technology ends up being the most cost-effective. Besides this, when it comes to off-grid micro-grid usage, the solar panels provide all the energy the system will use, while having area limitations depending on the application (for example, the size of the roof on a house). As such, the extra efficiency that monocrystalline solar panels bring to the table when compared to polycristalline represent a valuable asset in several applications [105].

Another important factor that affects the efficiency and power output of photovoltaic panels is temperature. Even though increase in temperature affects the current output of the solar system positively, there is a larger scale linear decrease in the voltage output, leading to lower power output [79]. This can be seen in figure 2.3, where the higher the temperature, the lower the power the solar panel can provide.

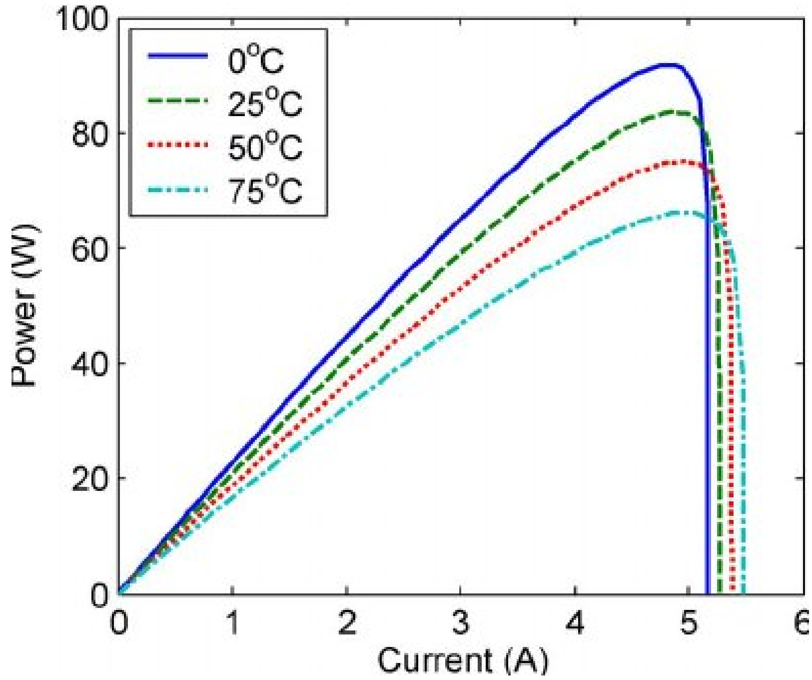



**Figure 2.3:** Variation in power output of photovoltaic panels with temperature [124]

### 2.2.3 Battery

Batteries are one of the most prevalent methods for storing energy nowadays. From smartphones and headphones to cars and stationary applications, batteries are found everywhere. The efficient storage of energy in batteries as chemical energy and their ability to convert this energy into electrical energy with a fast time response, make them crucial devices in facilitating the green transition. By charging during high renewable electricity production and discharging when production drops, as it does suddenly for renewables, batteries are decisive in the decarbonization of the energy markets [70]. On the other hand, batteries are not the perfect solution. They require big quantities of raw materials and degradation diminishes their capacity to store energy. In the next paragraphs, the two main battery technologies will be presented and a brief overview of the promising flow batteries will be provided.

#### Lead-Acid batteries

This was the first genre of battery to be developed and brought to the public, dating as far back as the 1850's, which explains why it's present in the most applications out of all the battery types and it's easiness to obtain, being cheap and widely available. It's limitations in the current delivery lead to a poor energy/weight ratio when compared to other technologies, adding to its low energy storage, regular maintenance requirements and the smallest lifespan due to the very limited ammount of charge/discharge cycles it can handle [18].

#### Li-ion batteries

As opposed to the aforementioned Lead-Acid batteries, Lithium-Ion batteries are very recent and have seen enormous development in recent years due to shear amount of applications they have, from portable electronics to electric cars. This particular battery technology is extremely suitable for these applications due to its very high energy density, the possibility to be fully discharged without completely ruining further performance, quick recharge speeds and big lifespans due to its high tolerance in terms of charge/discharge cycles. However, since it's still early in its development cycle and its a relatively recent technology, with high demand in very big markets, it has very high prices when compared to the Lead-Acid batteries [18].

#### Flow batteries

This technology is extremely early in its development, and it consists of a rechargeable battery that uses two liquid electrolytes to store energy. The electrolytes then flow through a cell stack in order to electrochemically release or adsorb energy, therefore charging or discharging the battery. This cell stack the electrolytes flow through lead to one of the advantages of this technology, which is its ability to scale, allowing for increase in the stack in order to have higher power output, and increase in the size of the electrolyte tanks to have higher energy storage. Besides this, they have a longer lifespan and the materials used in its production are safer and more environmentally friendly than materials present in other technologies, like lithium. This new technology also has its disadvantages however, such as lower efficiencies, in the range of 50% to 80%, higher costs due to being early in its development and having low availability, and finally its low energy density. This last disadvantage can pose as an advantage in microgrid applications, as low energy density excludes the use of this technology in mobile applications, therefore reducing demand and possibly reducing prices and increasing availability

[104][106].

## 2.3 Hydrogen

With the rise in usage and availability of renewable energies, the search for new and improved ways of storing energy has increased. This is due to the innate inconsistency and unreliability of renewable energy sources, which are heavily dependant on nature and weather conditions in particular. In the particular case of the energy source used for the microgrid in question in this report, solar power varies with many circumstances. Firstly, the night and day cycle immediately excludes, on average, half of the hours in a day for power generation. Adding to this, there's the seasonal factor, which leads to higher power generation in the summer when compared to winter, with a gradient in between them. Finally, the weather can heavily affect the power generation of photovoltaic panels, with a series of cloudy days being able to completely undermine the power generation of a solar energy system in a microgrid [102].

To combat the aforementioned inconsistency and unreliability, batteries of different technologies and sizes were used, but those quickly proved to be very limited and incapable of handling several applications. As such, one alternative storage method that began being studied and used is energy storage in the form of hydrogen. This element has several advantages that make it very favorable for usage in the energy storage industry:

- Hydrogen is the most abundant element in the universe and, namely, in the Earth's atmosphere;
- It has a very high energy density, meaning it holds an incredible energy to weight ratio, having nearly 3 times more energy content than gasoline for the same mass [116];
- Hydrogen has a variety of production methods, from natural gas and biomass, to water electrolysis. The versatility of hydrogen expands to its storage methods, ranging from compressed gas to cryogenic liquid, and also to its applications, being used to power fuel cell vehicles, as energy storage or industrial processes [116];
- Finally, hydrogen possesses a much higher capability of storing energy for a longer period of time without losing its energy potential, making it a possible solution for large scale energy storage, as can be the case in a hybrid microgrid [116].

Although hydrogen possesses characteristics that make it very favorable for energy storage, it also comes with its disadvantages:

- Even though it has very high energy density, it has extremely low mass density, so even though it holds great amounts of energy for a low mass, that same mass takes up a very large volume, making its storage a bigger problem. This volumetric density problem also affects transportation, while adding to the large volume needs the problem of high flammability [116];
- The use of hydrogen as a form of energy storage represents large losses due to the efficiency of the several technologies involved, with the efficiency of the electrolyzer, compressor and fuel cell leading to a round trip efficiency of approximately 35% [116];
- Hydrogen as a form of energy storage and all the technology related to its transformation to and from energy are still in early stages of development, making it expensive at the time of writing, with expectations of prices lowering with further development and with the technology becoming more widespread and available [116];
- Safety wise, hydrogen presents several concerns, whether they are its high flammability or the very high pressures at which it is frequently stored, bringing forth a risk of explosion. These problems are, again, heavily present both in storage and transportation [116].

For the purpose of the application in question in this report, the hydrogen can be produced through the use of electrolysis, using the energy obtained through the photovoltaic panels.

There are 3 types of hydrogen depending on the way its produced. Grey, blue and green hydrogen [33]. The majority of hydrogen produced at the time of writing is grey hydrogen, where hydrocarbons are processed with the use of Steam Methane Reforming, partial oxidation or Autothermal reforming. However, despite being the most economically advantageous option, this production method emits high quantities of carbon emissions, making it environmentally unfriendly. The use of carbon capture can make this production method more sustainable, while also allowing for other usages, for example combining the hydrogen with this carbon in order to produce methanol. The production of hydrogen with this inclusion of carbon capture, that can lead to a reduction of up to 90% in the carbon emissions, is called blue hydrogen (IEA. The future of hydrogen. 2019.). Lastly, the green hydrogen, which will be the focus of study in this particular case, is produced using electrolysis. This method consists in using the electricity deriving from renewable energy sources, namely solar power in the case of the hybrid microgrids studied in this report, to induce an electrochemical process that separates the oxygen and hydrogen present in the water. This method produces zero direct carbon dioxide emissions and currently represents approximately 2% of the global hydrogen production [66].

The sustainability aspect of green hydrogen production has made it a serious focus of research and development, with several pushes and incentives being made, not only by companies but also government bodies. This has lead to great enough security in the technology that allows for commercially available products in the market, creating the current goal of making an hydrogen system more affordable and readily available, with the example of the US Department of Energy, which seeks to reduce the cost of clean hydrogen by 80% to 1$ per kilogram by 2030 [31]. This process can be compared to that of photovoltaic systems, that over the years greatly reduced their cost and became a more widespread technology.

### 2.3.1 Electrolysis

As mentioned before, electrolysis consists of the process of using electrical energy to split water molecules into $H_2$ and $O_2$. This is done with the assistance of two electrically charged electrodes, called anode (positively charged) and cathode (negatively charged), and an electrolyte material, which varies depending on the type of electrolyzer in question. These components, while in contact with an acidic water solution or pure water lead to the aforementioned decomposition [61]. There are several types of electrolyzers, however 3 are vastly more developed and currently available in the market, and can be seen in figure 2.4.

- Alkaline Electrolysis Cell (AEC)
- Proton Exchange Membrane Electrolysis Cell (PEMEC)
- Solid Oxide Electrolysis Cell (SOEC)

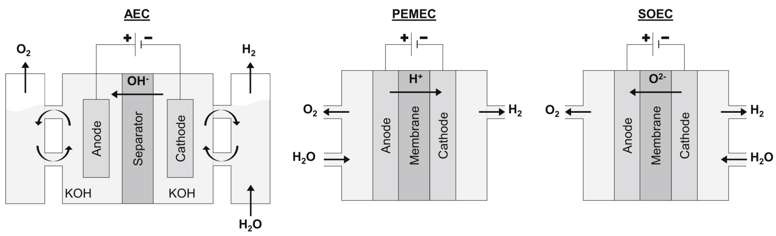



**Figure 2.4:** Conceptual set-up of three electrolysis cell technologies [101]

#### Alkaline Electrolysis Cell

This type of electrolyzer is the oldest, dating back to the 1920s. As such, it's also the simplest, consisting of the two aforementioned electrodes submerged in a liquid electrolyte solution, such as potassium hydroxide (KOH) and sodium hydroxide

(NaOH), although potassium hydroxide tends to be the preferable choice, as it has higher solubility in water, therefore creating a more conductive electrolyte solution. Besides this, KOH also has the advantage of leading to lower operating temperatures and longer service life compared to its alternative solution [67].

In this type of electrolyzer setup, the water molecules ($H_2O$) enter at the cathode and go through a reduction process, where they react with 2 electrons, producing an hydrogen molecule ($H_2$) and 2 negatively charged hydroxide ions ($2OH^-$). This reaction can be seen more clearly in equation 2.1[11].

$$2H_2O_{(l)} + 2e^- \rightarrow H_{(}g) + 2_{OH}(aq)^- \tag{2.1}$$

The hydrogen product that was desired can then exit without leaving the same side of the electrolyzer, while the hydroxide ions go through the separator between the electrodes, reaching the anode. The separator must allow for the transport of these ions, while preventing the crossover of hydrogen and oxygen gases, ideally having good chemical resistance, high porosity and low electrical conductivity. Asbestos was a great example of the correct type of material, was one of the most commonly used materials for this purpose. However, due to health and safety concerns, alternatives revolving around polymer-based separators are starting to become more common, such as polytetrafluoroethylene (PTFE), polyethylene, or polypropylene [11].

After passing through the separator, the hydroxide ions go through a reaction at the anode known as oxidation, where the excess electrons are released from the ions, leading to the formation of water and oxygen molecules. This reaction is better described in equation 2.2[11].

$$Anode : 2OH^- \rightarrow 0.5O_2(g) + H_2O(l) + 2e^- \tag{2.2}$$

The resulting oxygen and water can then be taken from the anode side of the electrolyzer, finalizing the electrolysis process.

This electrolysis method comes with the advantage of having availability in the market, due to being an older technology, while having lower costs due to the non-noble materials it uses for electrodes, such as Nickel, and having high durability, allowing for up to 30 years of usage. Despite all this, the alkaline electrolyzer cell presents some disadvantages that make non eligible for use in a hybrid microgrid, such as limitations when it comes to the current density, which can be limiting in applications that require a higher power electrolyzer, it has a low operating pressure, which creates the need for either a larger storage tank or a compressor, which leads both to more expenses and higher energy losses, and finally, and most importantly, the alkaline technology has severe limitations when working with intermittent power supplies, which does not go along well with the inconsistency and unreliability of renewable energies [101][11].

#### Proton Exchange Membrane Electrolysis Cell

This type of technology is more recent that the Alkaline Electrolysis Cell, having surfaced in the 1960s. This type of cell is much more structurally complex, including two plates that em compass the membrane electrode assembly, which includes two gas diffusion layers, two electrodes (the anode and cathode) and a solid polymer electrolyte, which allows for the separation of the gases, the conduction of the protons and the electrical insulation of the electrodes [101].

The operation of this type of electrolysis cell begins with the water being fed to the anode side of the cell, where a bi-polar titanium plate is in place. This material is chosen due to its strength and resistance to corrosion, but other material can be used, although leading to more corrosion and a lower lifespan of the cell, while reducing costs.

The water then goes through the titanium mesh and gets in contact with the anode, where a reaction takes place, taking the water molecules and transforming them into oxygen molecules and hydrogen ions, along with a set of 2 electrons. This reaction is described in equation **??**[20].

The oxygen molecules are then released through the same side of the electrolyzer where the water was introduced. The remaining products of the equation go through the membrane and reach the cathode, where another reaction takes place combining the electrons with the hydrogen ions, forming hydrogen molecules. These molecules then pass through a carbon mesh, which requires less corrosion resistance, as the reaction that takes place on this side of the membrane is less corrosive, and are expelled through the cathode side of the electrolyzer. The cathode side reaction is described in equation 2.3.1[20].

$$2H^+ + 2e^- \rightarrow H_2(g) \tag{2.3}$$

Just like the previous electrolysis technology, Proton Exchange Membrane electrolysis comes with its advantages and disadvantages. Immediately, it is more fitting for hybrid microgrid applications as it has a fast response time, being able to quickly respond to the demand. However, it functions with lower energy losses and less degradation if it is constantly fed power, even at lower values, since on-off cycles can become very demanding on its lifespan, which can be problematic with irregular feed power from renewable power sources. It also requires substantially less space than the former technology, although this can be considered negligible in a lot of the cases, since most, if not all, microgrid applications are stationary and the space requirements aren't demanding enough for the change in volume of the fuel cells to be relevant. However, in the cases where space is restricted, PEMEC technology takes up 100 times less area than AEC. The efficiencies are comparable between both technologies, however, in the case of Proton Exchange Membrane, it comes at a cost, with the use of metals such as platinum, iridium, ruthenium and others as catalyst significantly raising prices [20][11].

Within the realm of studies of PEMEC, there is a very similar technology called Anion Exchange Membrane that takes the basis of the Proton Exchange Membrane technology but uses an alkaline solid polymeric membrane instead, heavily reducing the prices by not using the expensive metals previously mentioned. This newly developed technology also shows great promise by adding very good efficiency values and directly compressed hydrogen to its reduction in price [49].

#### Solid Oxide Electrolysis Cell

This type of electrolysis generates hydrogen in a manner different from the previously mentioned technologies, and that begins with the conditions created by the temperature at which Solid Oxide Electrolyzers function. Functioning at temperatures between 650 and 1000ºC compared to the 50-90ºC of PEMEC and AEC, the Solid Oxide Electrolyzer works with water vapor instead of liquid. This steam is fed to the cathode, where a reduction happens, leading to the formation of hydrogen molecules and oxygen ions. The hydrogen is released through the cathode side again, while the oxygen ions go through a ceramic material that serves as electrolyte, reaching the anode side where they oxidise forming oxygen molecules. These two chemical reactions can be seen in equations 2.4 and 2.5[73][11][101].

$$H_2O(g) + 2e^- \rightarrow H_2(g) + 0^{2-}(aq) \tag{2.4}$$

$$20^{2-}(aq) \rightarrow O_2(g) + 4e^- \tag{2.5}$$

This electrolysis technology comes with its advantages and disadvantages, as did the previous ones. For starters, Solid Oxide Electrolyzers can reach a theoretical efficiency of up to 90%, however, this is only if heat from other sources is reused in order to help the electrolyzer reach the extremely high temperatures necessary, otherwise, efficiency revolves around values of 50-60% . Another theoretical advantage that is setback by the high temperature needs of the electrolysis is the durability of the solid oxide electrolyzer. Its components are highly durable and resistant, however, due to working under extreme thermal stress for very long periods of time, there are a lot of degradation problems still left to be studied and solved. These same highly durable and resistant materials, specialized in functioning under such straining thermal conditions lead to a high capital cost of the electrolyzer. To add to this, SOEs have very high start up times in order to achieve the temperatures of 600-1000ºC in order to function, which makes them unable to respond in sudden changes in the demand, which can be a big obstacle in hybrid microgrid usage, namely less consistent demand patterns. The very high temperatures and extreme complexity of the system can also make this technology harder to implement, with a lot of applications not being able to accommodate this type of electrolyzer and its temperature values. The last disadvantage is its limited hydrogen purity, with values of around 98-99%, which although high, can present a problem in applications that require higher purity values.

### 2.3.2 Fuel cell

A fuel cell uses a fuel to produce electricity and heat, thus functioning as the reverse of an electrolyzer. As such, the anode is now negatively charged and the cathode positively charged. In an hydrogen fuel cell, the electricity flow derives from the natural tendency of hydrogen to combine with other elements, in the case of fuel cells its oxygen, forming water. Just like in the electrolyzers, there's also an electrolyte material separating anode and cathode.

There are several types of hydrogen fuel cells, with different working conditions and different advantages and disadvantages. These will be mentioned in the following paragraphs.

#### Alkaline Fuel Cells(AFCs)

Just like the Alkaline Electrolyzers, these are some of the oldest fuel cell technology, meaning its also one of the most widely available and reasonably priced. It also keeps its parity with its electrolyzer equivalent by having a similar process but inverted. As such, hydrogen is fed at the anode, with oxygen or air being fed at the cathode. The hydrogen is then oxidized and electrons and hydroxide ions are the result, The electrons flow through an external circuit, generating the desired electricity flow, while the ions flow through the electrolyte material, reaching the oxygen in the cathode and forming water. These fuel cells have a very high electrical efficiency (50-60%) and use low cost materials. At the same time, they have longer start up times which can be problematic while trying to follow the demand of an hybrid microgrid, are very sensitive to the amount of $CO_2$ and other impurities in the air fed into the cathode and can be victim of its high operating temperature and pressure [108].

#### Proton Exchange Membrane Fuel Cells(PEMFCs)

These fuel cells have a similar system to its electrolyzer counter part, but working in the different direction, with the hydrogen being fed into the anode and oxygen into the cathode. The electrons are then separated and flow towards the cathode, generating electric current, while the hydrogen ions are allowed to travel through the membrane, reuniting with the oxygen and electrodes on the cathode side and forming water. These fuel cells work at relatively low temperatures (80ºC), have a higher power density, can have a small form factor and have a very fast start-up time, which is ideal for hybrid microgrids. However, they are very sensitive to impurities in the fuel and use very expensive materials, such as platinum [108].

#### Solid Oxide Fuel Cells(SOFCs)

Yet again, the Solid Oxide fuel cell works in a similar, yet opposite way compared to its electrolyzer counter part. Considering this, the hydrogen is, again, fed into the anode side with oxygen in the cathode side, with the oxygen being oxidized again and electrons flowing to generate an electric current. The ions combine with the oxygen, and form water both at the anode and the cathode. This technology is often used in large scale applications, with high electrical efficiency of approximately 60%, with these values being able to rise if the heat is reused on other applications. It can also have a very long lifespan, being vulnerable however to thermal stress and cracking problems, while its very high operating temperature (1000ºC) requires longer start up times and more insulation, with these temperatures also requiring very expensive ceramic materials [108][50].

#### Phosphoric Acid Fuel Cells(PAFCs)

AS the name indicates, this type of fuel cell uses phosphoric acid as its electrolyte. Just like in the previous technologies, hydrogen gas is fed into the anode, while air or oxygen is fed into the cathode. The catalyst in the anode helps oxidize the hydrogen, with the electrons following an external circuit and generating an electric current, and the hydrogen ions flowing through the electrolyte to the cathode, where they will react with the oxygen to form water. This technology has a decently high efficiency of 40-50%, which is still lower than some other fuel cells, and has a lifespan similar to solid oxide fuel cells. However, it functions at temperatures double the ones needed by the Proton Exchange Membrane Fuel Cell, implying longer start up times, it uses expensive materials, such as platinum, like the PEMFC, and has a low power density, requiring larger form factor cells in order to produce the same power [108][50].

#### Molten Carbonate Fuel Cells (MCFCs)

In this type of fuel cell, not only is hydrogen fed into the cathode, but also carbon monoxide or carbon dioxide. At this nickel composed electrode, oxidation occurs, releasing electrons that then follow, yet again, an external circuit. The remainder of the components combine to form water, while the remaining ions flow through the electrolyte, which is an alloy made out of lithium, sodium and potassium carbonates, into the cathode side. There they combined with the inflow of oxygen and form water, along with other carbon by products such as carbon dioxide. Just like SOFC, this technology works at very high temperatures (650ºC) and presents high electrical efficiencies of around 60%, which can be increased with heat reintegration, while having a lifespan similar to the aforementioned technology. However, and keeping the similarities, the high temperature also requires an longer start up time, expensive materials for the construction and leads to a susceptibility to corrosion and degradation over time [108][50].

### 2.3.3 Hydrogen storage

Due to the aforementioned characteristics of hydrogen, its storage is very particular. Since it has very high energy density, but very low volumetric density, it requires compression to higher pressures in order to be able to store high amounts of energy within reasonable volumes in gas form, or otherwise to be stored in an alternative manner. The other two storage methods are in cryogenic liquid form, or solid state storage [57][22].

Cryogenic storage consists of storing hydrogen in liquid form, by lowering its temperatures to extremely low values (-253ºC). Hydrogen in this form has a much higher volumetric density, lowering the volume needs. However, this storage method comes with difficulties related to its extreme low temperature needs. The materials and technology have very high costs, maintaining such low temperature values requires very high energy consumption, reducing the round-trip efficiency of the hydrogen system even more, and it comes with safety concerns related to leaks [57].

solid-state storage is another promising storage method that has been gaining traction recently. It consists in storing the hydrogen with the assistance of a solid-state material, namely metal hydrides, chemical hydrides and carbon-based materials. The metal hydrides represent high storage capacity and variety in the form they can present themselves in, from powder to plates. Chemical hydrides, have the favorable property of being stable at room temperature, with the condition of releasing hydrogen upon heating, which can represent complications in several applications. Finally, carbon-based materials, for example graphene, have started being considered as an option due to their high ability to store hydrogen. In general, these materials present a more efficient storage method, but the technology is still very under developed, expensive and difficult to produce [57].

Lastly, there is compressed hydrogen, which consists in compressing the hydrogen into higher pressures, typically between 200 and 700 bar, and storing it in tanks. This method allows for higher volumetric energy densities the higher the pressure is, its low maintenance, it only requires energy usage in the compression part of the gas, it has a lot of preexisting technology already and compression is already a very developed field, reducing the costs and increasing availability. However, this method also has limited capacity, due to a maximum in the pressure that can be achieved while keeping the hydrogen a gas and with the current technology, which then also leads to safety concerns, since the pressure wants to be maximized in order to achieve higher energy densities, but the higher the pressure, the higher the safety risks. Despite all this, this was the chosen method for the hybrid microgrid, as it is the only one that can be applicable due to safety issues, prices, availability, etc [57][22][43][94].

The energy densities associated which each pressure and liquefied hydrogen are in figure 2.5.

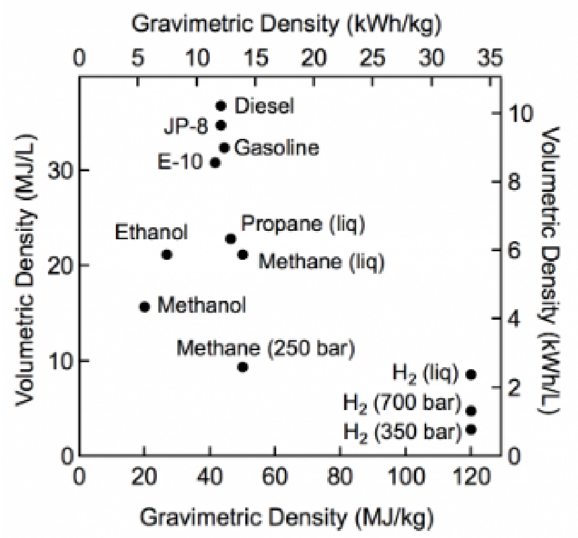



**Figure 2.5:** Energy density hydrogen [39]

## 2.4 Hydrogen microgrids

Energy storage will be a key issue in the future of decentralized and renewable electrical production., such as grid stability and energy storage. The intermittent nature of solar and wind energy presents inherent challenges for power systems. Solar and wind production plants hardly ever match their production with the load demand. A system which is capable of releasing and absorbing energy rapidly is needed when integrating these renewable assets in any grid. Commonly, in a country's national grid a Hydropower plants, Natural Gas Plants, Combined Heat and Power Plants or Energy Storage Systems (ESS) provide these necessary ancillary services to the grid.

As mentioned above, ESSs play a vital role in a grid or microgrid. Today, different technologies are available for this purpose, however depending on their capacity, response time and additional case-specific factors the optimal solution usually is a combination of them. In figure 2.6, the most reliable or promising

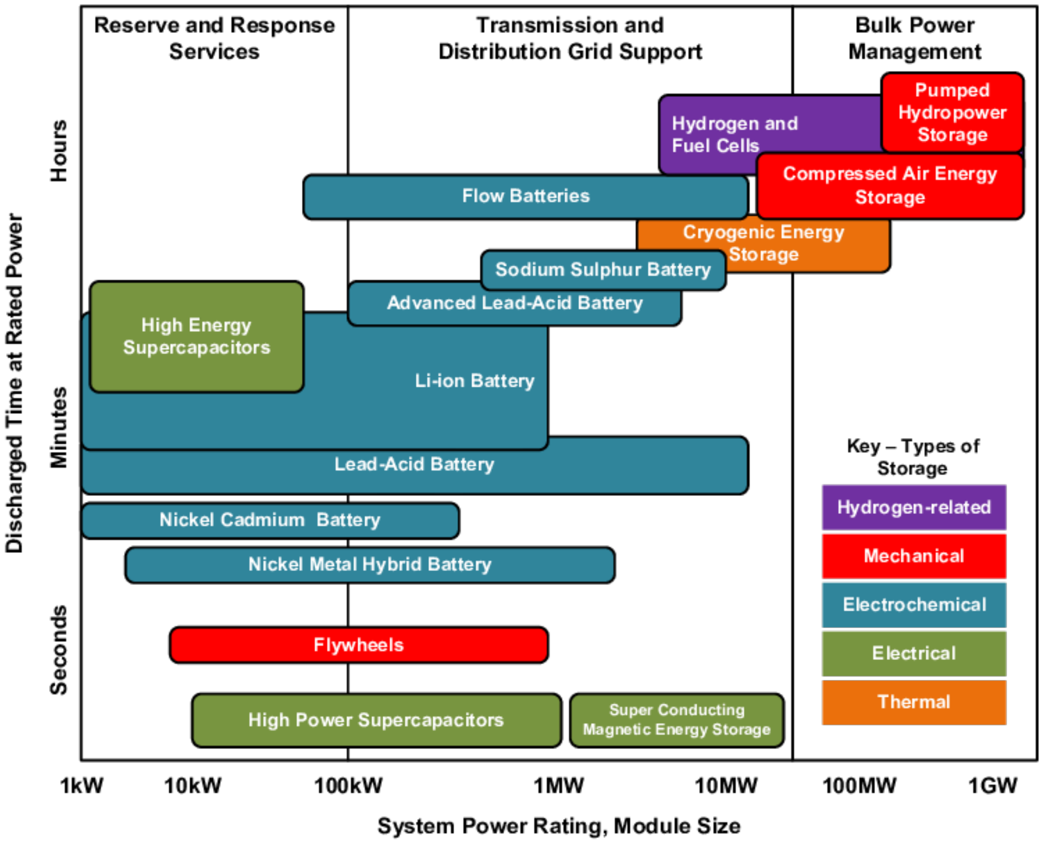



**Figure 2.6:** Typical discharge time and power ratings for energy storage technologies

energy storage solutions are depicted. The lower-left part of the graph is occupied by technologies meant mainly for power quality while the upper-right side is filled energy management solutions.

Hydrogen storage stands as a good alternative that targets the problems that cannot be target by batteries nor pump hydro or compressed gas storage. Batteries require a big quantity of materials, sometimes scarce and expensive to extract from Earth's crust. This issue is also encountered in fuel cells and electrolyzers, e.g. Iridium, which can become a bottleneck for the PEM technology [7]. However, both devices are power-rated, meaning that they are not meant to store energy. In contrast, they just perform an energy conversion, from chemical energy to electricity or vice versa. The quantity and cost of batteries for powering an average European house for over a week are prohibited. Nonetheless, they are commonly included in hydrogen microgrids, due to their fast response time and high round-trip efficiency. On the other side, pump hydro or compressed gas storage requires a specific type of geography in order to be economically viable. [78]

Hydrogen is a highly promising energy carrier with inmense potential. The long-range freigh intensive transport sector is looking into biofuels or hydrogen power vehicles. Electrification in this case is not a good approach since batteries have currently a higher specific energy. [2] Hydrogen also poses as an alternative for stationary energy storage, especially in remote communities, as it is the focus of this report. Fuel cells and electrolyzers will provide back-up power in off-grid island microgrid configurations, which will make it possible for small communities to have access to electricity all year round, every hour of the day. In the case of grid-connected micro or mini-grids, hydrogen storage can provide valuable ancillary services and price arbitrage. This will reduce electricity prices and give higher reliability to the main grid.

Hydrogen investment is increasing rapidly, yet, there is a necessity to complete the proposed projects to finally demonstrate the possibilities of hydrogen and attract more capital. In this report, we aim to provide a useful software solution that can be used by different professionals and decision-makers. [71]

### 2.4.1 Prior research

In this section, a review of several relevant papers on hybrid hydrogen microgrids is presented. These papers are focused on the minimization of cost via optimal sizing, enhancement of lifetime, and the reduction of wasted power. Factors that will enable a faster microgrid deployment and realisation. The microgrid studies explored hereby are not grid-connected, mainly targeting rural areas or big hydrogen production plants located in deserts for instance. Different algorithms are used in order to get the lowest cost, without forgetting the environmental perspective.

Monforti et al. [84] presented 4 different strategies for an HS (Hybrid System). Only hydrogen, only batteries and two hybrid operation modes, one with hydrogen priority and one with battery priority. In this work, the different strategies are compared in terms of Loss of Load (kWh, %) and Over Production (kWh, %), accounting for the amount of energy not delivered to the load and the energy wasted, respectively. The hybrid system outperformed the other two in Loss of Load, determining that hybridisation is always beneficial to increase energy security. The battery priority strategy was able to follow the load better than the hydrogen preferent one, which is logical, as fuel cells have slower response times than batteries. On the other side, the amount of energy loss was reduced when hydrogen devices were chosen over the battery. This paper is based on a previously sized microgrid, and the optimization is not taking into account the degradation of the components.

G.Bruni et al. [13] discuss the effect of sizing and energy management strategy on the efficiency of the overall system. The energy management used follows a voltage control, the highest voltage is the one delivering the power. The system discussed in this paper is a Fuel cell-battery-PV system, so it is not a closed system, hydrogen has to be shipped to the location from a production site. When the system is unable to follow the demand, a diesel generator supplies the deficit. The authors sized the system primarily according to the following:

- Solar Panels deliver 150% of the consumption of the house studied.
- FC power is equal to the maximum allowed consumption in an hour.
- The battery is able to provide energy for a day and a half.

Similarly to the conclusions from [85], the authors state that a microgrid without long-term energy storage has a lower renewable energy exploitation. The main contribution of this work is the sizing study of the components. 3-dimensional graphs depicting a sensitivity analysis of the size of the components and plotting the efficiency or percentage of fossil fuels use of the microgrid.

Torreglosa et al. [114] developed a Hierarchical energy management system for a stand-alone microgrid in Matlab - Simulink environment. Their model decides where the power should be allocated evaluating the SOC of the battery, the operation mode (Charge or Discharge), and the H2 tank level. 10 different states are described. The lifetime estimation of the components is done directly assigning a lifetime to each of the components. This work includes a remarkable simulation at a timescale of seconds. The so-called Slave control uses PI controllers to balance the bus voltage following the reference powers given by the master control. The demand signal is modified introducing noise to simulate a real demand curve.

Garcia et al. [96] presented an energy management system for an islandic hybrid microgrid including solar and wind generating assets. The system is able to store energy in a battery and a hydrogen tank. The modeling technique is based on a fuzzy logic. First, a charge power limit is defined. This value sets the cutoff power which provides a value to a variable that decides whether it is cheaper to power a battery, electrolyzer, or fuel cell. Afterward, the fuzzy variable takes the mentioned variable and the state of the microgrid to decide the amount of power awarded to each component. The authors claim to save a 13º% of energy compared to EMS which only use system states to allocate the power.

In this other piece of work done by Torreglosa et al. [113] sizing and EMS is executed. Despite the sizing method is not provided in the paper, the authors lead to this other source developed by Casteñaneda et al. [15]. The work from Torreglosa et al. uses Linear Programming to minimize the objective function, which is based on the cost per hour of operating each component. A degradation approach is included in their model, decreasing the performance of the components according to their usage. Thus, the authors claim that they can better estimate the replacements of the components, increasing the accuracy of their results. Fuel cell and electrolyzer manufacturers commonlly provide a warranty of the working hours of their devices within allowable values of voltage and/or efficiency. In this paper, the authors calculated the equivalent working hours of the fuel cell as follows:

$$H_{fc} = \frac{P_{fc}}{P_{fc}^{nom}} + 3 * K_{on} * \frac{Cycle_{fc}}{Cycle_{fc}^{nom}} \tag{2.6}$$

The method to obtain the hourly cost of power per component is done as follows. It is presented for the battery, but the process is similarly done for the fuel cell and the electrolyzer.

$$C_{bat} = \frac{NPC_{bat}}{E_{bat,cycle} * L_{bat,cycles} * SPWF} + \frac{NPC_{bat,O\&M}}{P_{bat,max} * SPWF} \tag{2.7}$$

Their innovative strategy decreased by more than 250% the power lost and reduced the number of electrolyzers used from 4 to 3 in a 25-year span.

The sizing procedures are explained in [15]. Here the authors discuss 4 different procedures in which a microgrid could be sized:

1. **Basic technical sizing:** This method sized the Solar farm by balancing the generation and demand. The fuel cell was chosen in order to provide the demand peak power alone. The electrolyzer was sized to absorb the maximum excess power, which is equal to the peak generating power minus the minimum load. And the storage systems, battery and hydrogen tank were able to cover the demand of a complete 24-hour day and 8 hours respectively

2. **Simulink Design Optimization:** Latyn Hypercube and Genetic Algorithm are used in this study. This method is entirely technical and therefore it does not account for any economic metrics.

3. **HOMER:** Homer is a software tool developed by NREL, capable of designing power systems including multiple energy sources and storage options. HOMER optimizes the stand-alone microgrid by running hourly simulations and ordering them by economic criteria [38].

4. **iHOGA:** This is a software developed by Unizar, University of Zaragoza, Spain. The tool makes use of 2 genetic algorithms to minimize the sizing first and optimize the EMS for each proposed size [115].

In summary, most of the papers make use of a pre-defined energy management strategy for their simulations. Several strategies are exposed depending on the decision mechanism, declaration of states, and priorities. Including the degradation in terms of usage, stands as a more precise option to calculate the loss of performance of the components. Different sizing methods are also explored throughout these studies as well as sensitivity analysis. However, papers are hardly comparable one to another due to the different sizes, locations and demand curves. There is not a unique solution that will work everywhere, but the goal is to find among the available technology the best combination that provides low-cost energy benefiting the exploitation of the accessible resources.

### 2.4.2 Early adoption

Although hydrogen adoption is barely visible around the majority of us, there are several cases of hydrogen-powered houses around the globe.

The first commercial Hydrogen House was established in Pennington, New Jersey, US. The system was developed by the engineer Mike Stricki in 2015 and it includes a 40 kWp Solar installation and a 20 kW backup power plant. The house will include a charging station in the future, for electrical or hydrogen cars [98].

Another engineer, Hans Olof Nilsson, developed his own hydrogen house in South-west Sweden. He proved that it is possible to be completely off-grid even with the cold and dark Swedish winters. 20 kWp of solar power generates all the electricity needed by the house. 144 kWh of lead-acid battery energy storage provides backup power during the night or to charge the electric vehicle driven by the engineer. A 9 kW electrolyzer converts the excess electricity into hydrogen during the summer, which is consumed by a 5 kW fuel cell, whose main task is to recharge the batteries. A compressor is used in order to pump and increase the pressure of the hydrogen coming out from the electrolyzer. The house is 160 $m^2$ [24].

Greater energy demanding matters have also showcased the reliability of hydrogen powered generators. In 2021, the company Hydrologiq successfully powered the public transport hub at Boomtown Fair, a festival with over 65000 attendees. The generator recharged the electric vehicle fleet during 11 days with a 100 kVA fuel cell generator. 6

# Chapter 3

# Methodology

In this chapter, the methodology of the modeling of the components is outlined. Starting with the case studies considered to test the code. The energy system is outlined for the calculations regarding solar panels, battery and hydrogen components, including the features calculated and considered in the code, such as economics. Finally, the two optimisation methods chosen for the sizing and the energy management of the system are outlined. All the modeling was done using *Python* programming language, with the additional packages specified in the appendix A.1.1.

## 3.1 Selection of test studies

For the purposes of this study, specific test cases had to be chosen in order to not only determine the situations in which hybrid microgrids are most viable, but also to fully test the software developed. As such, 3 different cases were chosen with a wide range of applications, locations and power consumption. These are presented as follows:

- Village in Kenya
- House in the Netherlands
- Resort in the Maldives
- Data center in Greece.

The reasoning behind this case selection and what their results could bring to this study are presented in the following section.

### 3.1.1 Selection criteria, discussion

The first basis for the selection was the need for 3 different scales, both in dimension and in consumption, while also including an application that wouldn't be residential. As such, the first and most straight forward choice was an house in isolation, with residential load curves and lower consumption values, in the order of 3000 kW/year, as per the Germany average [29]. Following that, a real case was taken, worked in association with *United Village*, of a village's common area in Kenya, composed of fridges and washing machines, and consuming in the vicinity of 8000 kW/year.

The larger cases were selected based on information found and realistic cases. First, a particular case from with focus on bringing renewable energies and energetic independence to islands in the Maldives was studied.The specific case in hand is that of *Park Hyatt Resort*, located in the Gaafu Alifu Atoll, with a peak photovoltaic power of 616 $kW_p$, toned down to 10% of the value given by the source in order to have a wider range of test cases [109]. An overview of this resort can be seen in figure **??**.

The final case was chosen with the purpose of being one of the largest scale applications existent, with the final decision falling on a data center. The location was chosen based on the planned construction of a data center in Crete, Greece, by Digital Realty's Lamda Hellix, and a total capacity of 6.5 MW [26]. As for obtaining more numbers around data centers, the existing case of Foulum Data Center, by Apple, and located in Viborg, was taken into consideration. The numbers that were taken about this data center were the size, at 166.000 square meters, the 42 MW solar farm that powers it, the 700GWh per year consumption and the investment needed of 850 million € to build it [122]. All these numbers can be used to calculate the proportions of the energy needs and costs that go into a data center's energy system.

## 3.2 Microgrid features

The components of the microgrid are structured and modeled linearly. The commencement of the calculations and the initial values considered are solar irradiation or global horizontal irradiation (ghi), ambient temperature and the load curve of the electricity depending on the application. Afterwards, the energy flows can be approximated and the hourly calculations for the lifetime of the project calculated.

The energy flows are based on consumption that comes from the load curve and production from solar panels. The energy is then transferred to charge a lithium-ion battery. The battery is responsible for following the load and also powering the electrolyzer. The electrolyzer is powered when excess energy is provided to the battery and produces hydrogen gas, which, depending on the application, passes

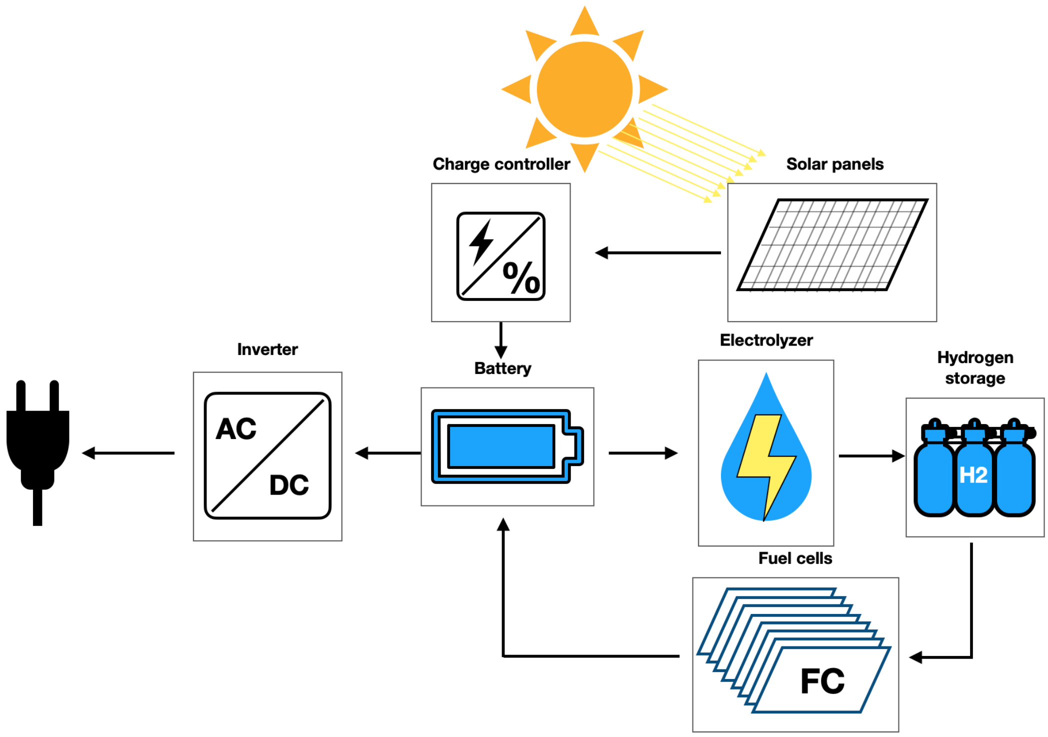



**Figure 3.1:** Components of residential microgrid with energy flows using solar power as input battery and hydrogen storage

through a compressor or is channeled directly to storage tanks. Depending on the demand, supply, energy balance and state of charge (SOC) of the battery, the fuel cell is powered and its energy output charges the battery. A simple schematic of the energy system is illustrated in figure 3.1.

### 3.2.1 Calculation and modeling tools

In this subsection the calculations and modeling tools utilised to simulate the component operation, interactions, sizing and optimisation will be discussed. The computational framework is based on a combination of mathematical equations, data analysis tools, empirical correlations, optimisation algorithms and financial analysis. The implementation of the aforementioned framework was conducted using the *Python* programming language, in *Jupyter Notebooks* [74] and *Visual Studio Code* as the primary development environments.

Python in the last years has played the role of lingua franca amongst software developers and data analysts [119]. Data analysis is integral to modeling and optimisation processes, by facilitating the importation or creation, structure, organisation, manipulation, visualisation and export of data. Arguably the most popular tool for data analysis, and the one used in for this thesis project, is the *Pandas Dataframes* library [82]. The *Pandas* library provides an efficient and stable structure for interacting with large datasets. In the context of microgrid modeling, the Pandas library enables the efficient management of time-series data, an integral part in the modeling. Solar irradiance, load curves and ambient temperature, which are the essential inputs for the simulation and optimisation of the system, are all time-series data. Moreover, the comparison and visualisation of the simulation results is facilitated by the *Pandas* library, thus assisting in the identification of patterns and underlying trends that facilitate more efficient and cost-effective microgrid configurations.

The optimisation is carried out in two layers: the first layer is facilitated by the use of genetic algorithms, which are a type of evolutionary optimisation that mimic processes of natural selection in order to identify optimal solutions in a vast solution space [58]. In these algorithms, some initial candidate solutions are initialised and then evaluated based on a fitness function, with the best performing solutions being combined with each other and mutated and new solutions are generated in successive iterations, until an optimal combination of solutions is identified [53]. This approach avoids the algorithm solutions of being trapped in local optima [27], identifying high quality solutions and thus making sure that the search for the best microgrid configuration is robust and efficient. Genetic algorithms have been applied to a multitude of problem domains of microgrid optimisation, such as optimizing distribution of energy resources, sizing and placement [19], energy management [126] and the design of multi energy carrier and storage systems [9]. Leveraging the adaptability of genetic algorithms and based on the demonstrated performance in solving complex, multi-objective and non linear optimisation problems in the aforementioned studies, they were implemented in the microgrid sizing for the present thesis report.

The second layer of the optimisation was carried out for the simulation of the energy management system (EMS). For this stage, linear programming methods were utilised for the optimal allocation and management of the energy resources between the different components of the microgrid [123], while taking into account the constraints and objectives associated with the system, outlined in section 3.4. Linear programming is an optimisation method that can identify solutions to problems by minimising or maximizing a linear objective function, which is subjected to linear equality or inequality constraints [5]. Microgrid energy scheduling, renewable energy source integration and load balancing are problems that can be handled particularly well with linear programming techniques [55]. The implementation of the linear programming methods was done using the *Pyomo* library [54], an open-source optimisation package that provides a flexible and customisable platform for the formulation and solution of optimisation problems, enabling the efficient modeling and management of the energy flows in the microgrid. The precise optimisation strategy and EMS algorithms are presented in section 3.4.

### 3.2.2 Solar irradiation

Solar irradiation data is the most essential input for the simulation of the microgrid performance. In the calculations for the project, solar irradiation data is acquired using the library PVlib [59] and specifically, PVGIS [16]. The data provided from PVGIS are global horizontal irradiation (GHI), direct normal irradiation (DNI), diffuse solar irradiance (DSI) and ambient air temperature. GHI data is calculated by summing the direct and diffuse solar irradiation. The inputs for the PVlib call are latitude and longitude, the tilt and azimuth of the solar panels are calculated and optimised automatically. The solar irradiation data, as well as the temperature data are hourly and obtained for a 12 year time-span, between 2005 and 2016. A PVlib call is presented in the Appendix A.7.

In order to ensure a more conservative approach to the microgrid design, the worst case scenario in regards to solar irradiation is calculated, selecting the year with the lowest average GHI in the available data range. The average GHI is also calculated, then the two values are structured in a 20 year dataset. The script also processes the loaded solar data by eliminating the data for the leap days (February 29th). The datasets are structured in a format where the average year is replicated 4 times, followed by the worst case, always in terms of GHI values. Then the 5 year dataframe is replicated 4 times to form a 20-year blueprint that is used for sizing and financial assessment of the system.

Finally, the solar panel power output is calculated by multiplying the available solar area with the GHI and the panel efficiency, which is 22% [77]. The temperature losses and gains were also accounted for, as mentioned in 2.2.2 solar panel current is heavily dependent on temperature. For the calculations, the temperature of the solar panel surface when receiving solar irradiation, was assumed to be 20 degrees higher than the ambient temperature [48]. The temperature of the solar panel surface was then calculated in relation to the nominal operating cell temperature (NOCT), which was 42 degrees, and an adjustment of 0.03% per degree of performance loss or gain was adjusted. Both the solar panel efficiency and the temperature characteristics were calculated according to the LG NeOn bifacial solar panel [77].

The equation for the solar power $P_{solar}$ input is outlined in 3.2.2. Where $GHI$ is the global horizontal irradiation, $\eta$ the solar panel efficiency, $T_{diff}$ the difference between the solar panel surface temperature and the nominal operating cell temperature, $\alpha$ is 0.33 %/C°[77].

$$P_{solar} = GHI \times \eta \times A - (T_{diff} \times \alpha \times P_{solar}) \tag{3.1}$$

### 3.2.3 Interface

The interface developed for this report is designed to aid in the collection of solar data and the creation of the load curve for each of the selected applications. The Interface centered around AAU campus is illustrated in figure 3.2.

On the right side of the interface is an interactive map from which coordinates can be selected. On the left side there are two checkboxes for selecting the type of project, either residential or commercial. The residential option when expanded is illustrated in figure 3.3. The options are the area of the house, the yearly electricity consumption in kWh, the option of electric vehicle charging and the coordinates. In figure 3.4 all the inputs are specified, the specified numbers are the ones used for the calculations and modeling of the residential application in this report. The values are then passed on a python dictionary. The calculations following are outlined in sections 3.2.4 and 3.2.2.

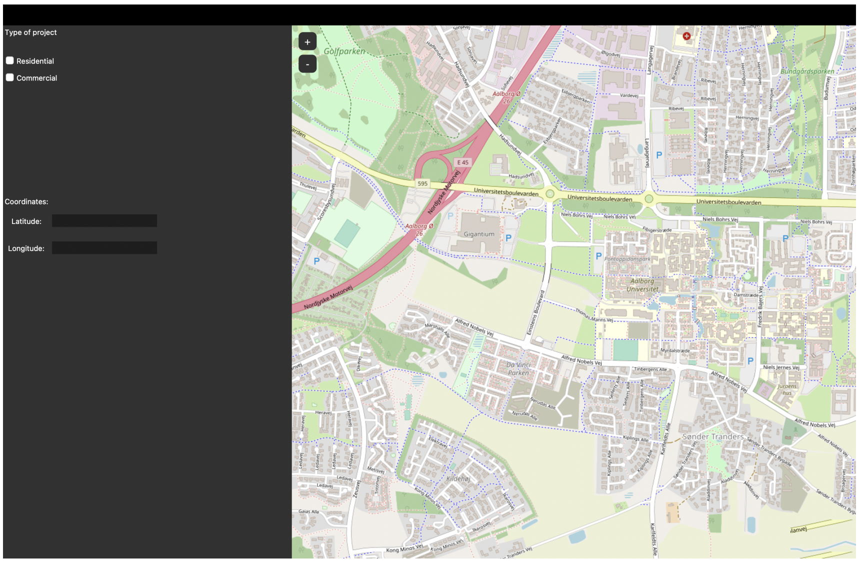



**Figure 3.2:** Graphical User interface centered at Aalborg University east campus

### 3.2.4 Load curve

The load curve is the other fundamental aspect for the understanding of the microgrid energy balance and simulating and optimising the operation.

These curves represent the energy demand of an electrical system over a specified period of time and for repeated timesteps, capturing the energy usage patterns and temporal fluctuations [28]. Each different load curve for the test cases selected in this report contains crucial information for the design, sizing, energy management, operation and cost of the microgrids. Such parameters are peak demand, to-

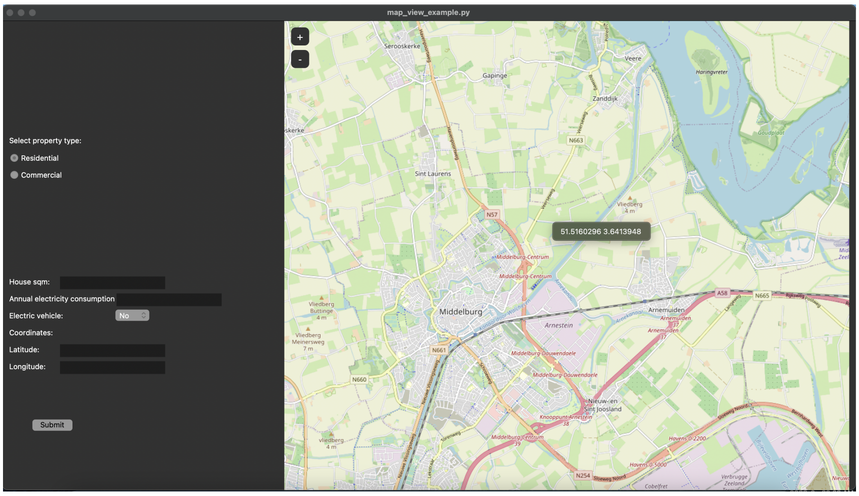



**Figure 3.3:** Graphical use interface for selecting location of the residential installation on the map

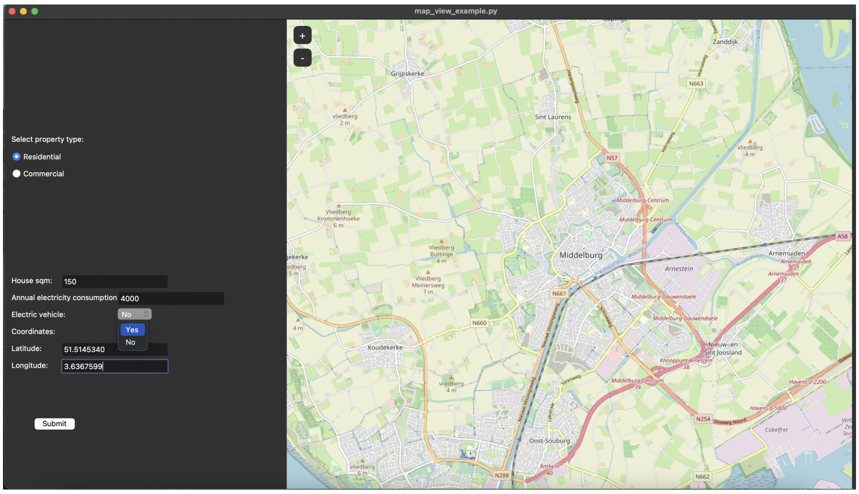



**Figure 3.4:** Graphical use interface for selecting location, yearly consumption set to 4000 kwh and electric vehicle

tal energy consumption, synchronisation with renewable energy production, load diversity and demand cite flexibility.

For this comprehensive analysis, energy consumption and load curves for four different scenarios were evaluated: a communal facility for a village in Kenya, a house in the Netherlands, a hotel in the Maldives and a data center in Greece. The test cases are structured according to electricity consumption. Further description regarding the selection of the test studies and details are outlined in section 3.1.

- **Residential load curve for house in the Netherlands**

  For the reference test case of the house in the Netherlands, the location and yearly energy consumption are specified as outlined in section 3.2.3 and solar data are loaded as outlined in 3.2.2. The energy demand including the heating and cooling load depend on a variety of factors and mainly the physical parameters of the building, the climate and level of solar gain. From the input values of the GUI, the floor area of the building is used to calculate the heating and cooling demand. The floor area of the building is then used to calculate the wall and roof areas. Using the floor area, the length and width of the house are calculated as the square root of the total area and the wall area as two times the product of the length and height of the walls, for a height of 3 meters. The assumption taken here is that the house is a square and one of the sides has significant solar exposure and one does not, meaning a symmetrical North-South orientation. The roof area is the same as the floor area.

Then the U-value of the building's walls and roof are defined according to the Danish Building Research Institute [25]. Those values being 0.3 $W/m^2K$ for the walls and 0.2 $W/m^2K$ for the roof. The U-value is a measure of heat transmission through a building's parts, the lower the U-value, the better the insulation of the building. Each building part has a different U-value. The orientation, shading and reflectivity values are defined for the adjustment of the solar gains. The solar gains for the walls are calculated for each orientation and then summed. The equations for calculating the solar gains for walls and roof are outlined in equations 3.2.4 and 3.2.4 respectively.

Solar gains for walls:

$$\begin{aligned}\text{SolarGain}_{\text{walls}} =& \frac{GHI}{1000} \times \text{OrientationFactor}_{\text{walls}} \\ & \times \text{ShadingFactor}_{\text{walls}} \times \text{Reflectivity}_{\text{walls}} \times \text{WallArea}\end{aligned} \tag{3.2}$$

Solar gains for roof:

$$\begin{aligned}\text{SolarGain}_{\text{roof}} =& \frac{GHI}{1000} \times \text{OrientationFactor}_{\text{roof}} \\ & \times \text{ShadingFactor}_{\text{roof}} \times \text{Reflectivity}_{\text{roof}} \times \text{RoofArea}\end{aligned} \tag{3.3}$$

Heating degree days (HDD) and Cooling degree days (CDD) are then calculated. HDD and CDD are climatological metrics in the energy analysis of a building, quantifying the demand for heating and cooling loads respectively. HDD quantify the difference, in degrees, and the time span, in days, that air temperature is lower than a specific base temperature. The base temperature used was 18 degrees Celsius [25]. Conversely, CDD is the equivalent measurement when the air temperature is higher than the same base temperature

[56]. The difference between air temperature and base temperature is added to the total CDD and in case it is below zero, zero is added. Both those processes, integrate the deviation of the temperature from the base temperature over time, providing a total measure of heating or cooling need over a specified period of time.

Heating Degree Days (HDD):

$$HDD = max(BaseTemperature - Temp_{air}, 0) \tag{3.4}$$

Cooling Degree Days (CDD):

$$CDD = max(Temp_{air} - BaseTemperature, 0) \tag{3.5}$$

The overall heating and cooling demands of the building are calculated by a set of equations that include the previously defined U-values, the calculated HDD and CDD, solar gains and electricity consumption. Heating and cooling demands are calculated for walls and roof respectively. The equations for calculating the heating and cooling demand for walls and roof are outlined in 3.2.4, 3.2.4 and 3.2.4, 3.2.4 respectively and equations 3.2.4 and 3.2.4 for the total amounts. The calculated heating and cooling are time series data. The time series are compared so that heating and cooling loads are not present at the same time. The heating and cooling loads are then added to the electricity demand for the house.

The heating demand for the walls and roof are calculated using:
For walls:

$$\begin{aligned}\text{HeatingDemand}_{\text{walls}} = \text{UValue}_{\text{walls}} \times \text{WallArea} \times \text{HDD} - \text{SolarGain}_{\text{walls}} \\ - 0.2 \times \text{ElectricityConsumption}\end{aligned} \tag{3.6}$$

For roof:

$$HeatingDemand_{roof} = UValue_{roof} \times RoofArea \times HDD - SolarGain_{roof} \tag{3.7}$$

The cooling demand for the walls and roof are calculated using:
For walls:

$$\begin{aligned}CoolingDemand_{walls} = UValue_{walls} \times WallArea \\ \times CDD + SolarGain_{walls} \\ + 0.2 \times ElectricityConsumption\end{aligned} \tag{3.8}$$

For roof:

$$CoolingDemand_{roof} = UValue_{roof} \times RoofArea \times CDD + SolarGain_{roof} \quad (3.9)$$

Total heating and cooling demands:

$$\text{Heating Demand} = \text{Heating Demand}_{\text{walls}} + \text{Heating Demand}_{\text{roof}} \quad (3.10)$$

$$\text{Cooling Demand} = \text{Cooling Demand}_{\text{walls}} + \text{Cooling Demand}_{\text{roof}} \quad (3.11)$$

The electricity load curve for the residential application is shaped based on the publication titled "Households' hourly electricity consumption and peak demand in Denmark" [4]. Which provides hourly coefficients of consumption for each hour of each month based on a study of the load profile in 50,000 Danish households.

The load curve coefficients were passed to a csv file and then were multiplied by the average hourly electricity consumption for the user input as it is described in 3.2.3. The coefficients represent a fraction of the total daily electricity consumption assigned to each hour of the day and each month. After processing, the data are shaped in an hourly load for one year. The Danish residential load curve [4] was compared with load curves from electricity consumption from all Europe and USA [36], Italy [3] and Japan [121] and a striking resemblance in the shape was observed. Therefore, the Danish coefficients [4] were used for all the residential applications. Moreover, the code checks whether the residence owns an electric vehicle (EV) and calculates the hourly energy demand for charging. The yearly electricity consumption required for EV charging is set to 3000 $kWh$ [4]. Random $+-5\%$ noise is introduced to the load curve and the electric car charging in order to make the data more realistic. The code then adjusts each individual power point to maintain the overall energy consumption constant by introducing a scaling factor. The code for generating the electricity consumption and EV charging is presented in the appendix A.8.

The final output for 20 years is illustrated in 3.5 and for two years in 3.6. The shape of a typical daily residential load curve for a winter day including heating and cooling demands and EV charging is illustrated in 3.7 and for a summer day in 3.8.

- **Load curve for communal facility in Kenyan village**

  The communal facility in Kenya was modeled for a real working case study. For this case study a communal washing and refrigeration facility near Nairobi

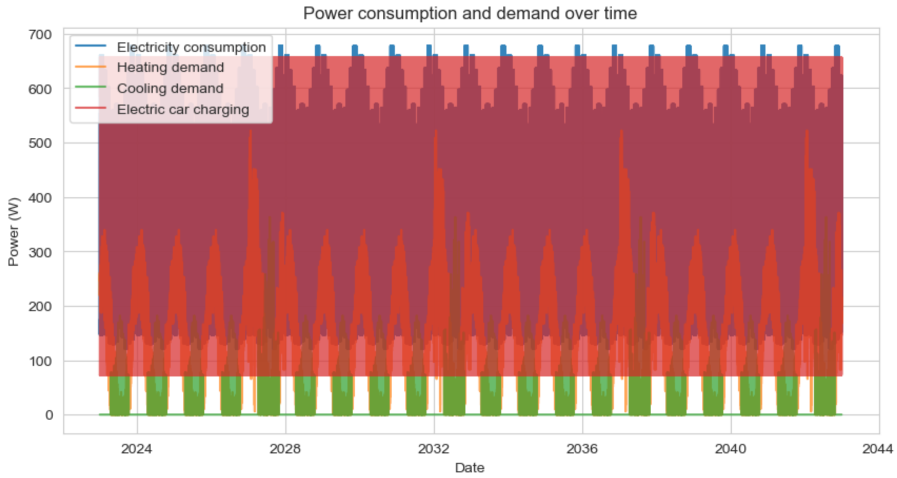



**Figure 3.5:** 20 year load curve including EV charging selected for house in the Netherlands, including heating and cooling demands

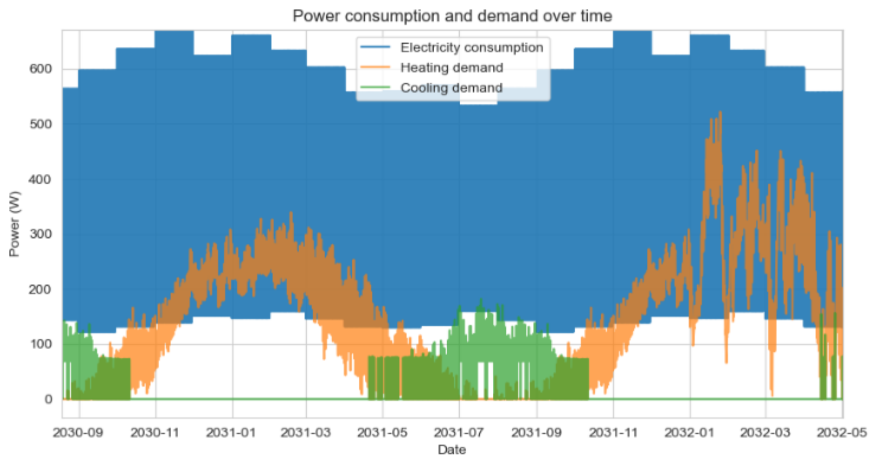



**Figure 3.6:** Yearly variation in the heating and cooling demand as well as the electricity consumption

was considered. The local community comprises of 100 people who are cattle farmers, so they are in need of two refrigeration units and two washing machines.

For the load curve, a baseline consumption of 200 Watts was set for miscellaneous consumption for lighting, phone charging and standby power of devices. It is presumed that there are two refrigeration units, each with an additional consumption of 200 Watts, thus the base consumption is updated accordingly. The consumption of refrigeration units is assumed as steady [4]. There are two washing machines also considered, the energy consumption pattern follows a washing cycle consumption pattern [127], assuming that

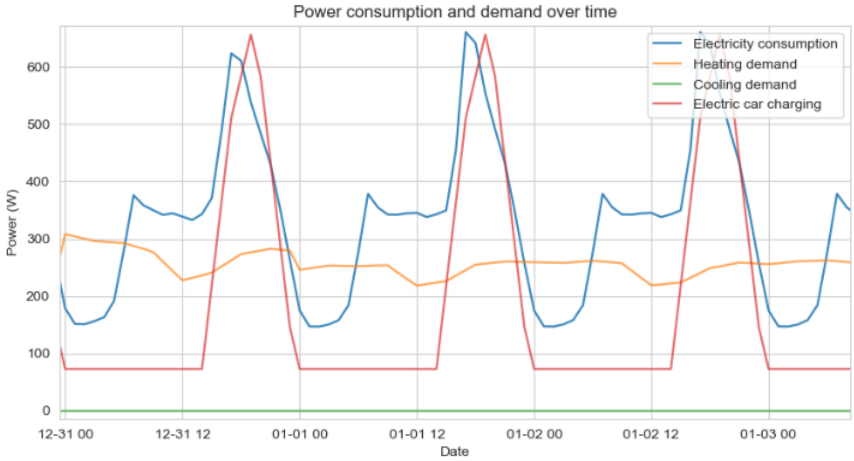



**Figure 3.7:** 72 hour load curve for house in the Netherlands during winter, with visible heating demand and EV charging

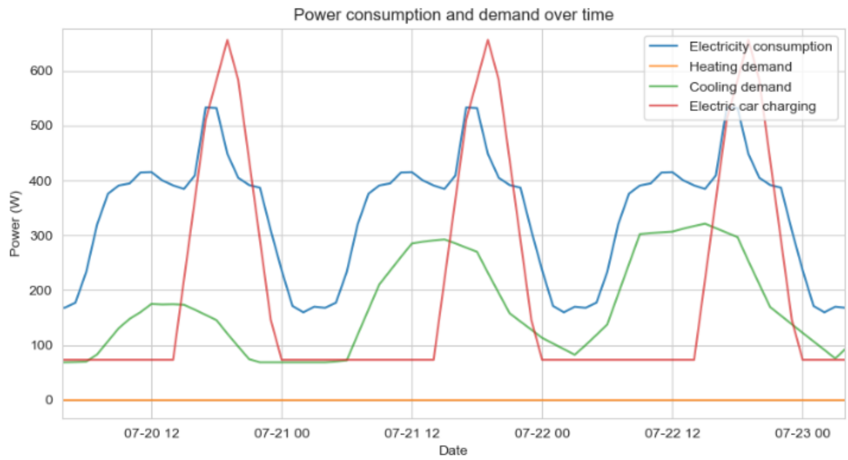



**Figure 3.8:** 72 hour load curve for house in the Netherlands in the summer, with visible cooling demand and EV charging

only one can run at peak power at a time, and that they operate between 9 am and 2pm. The code adds up this load and then multiplies it to create an hourly load curve. The code for generating the load curve is presented in A.9 and a 48 hour load curve is illustrated in figure 3.9.

- **Load curve for a resort in the Maldives**

  Regarding the resort in the Maldives, the load profile is based on an annual electricity consumption. The consumption was obtained from the yearly sustainability report for the Park Hyatt resort in Gaafu Alifi Atol [60]. The annual electricity consumption is reduced to 10% of the total, to make the case

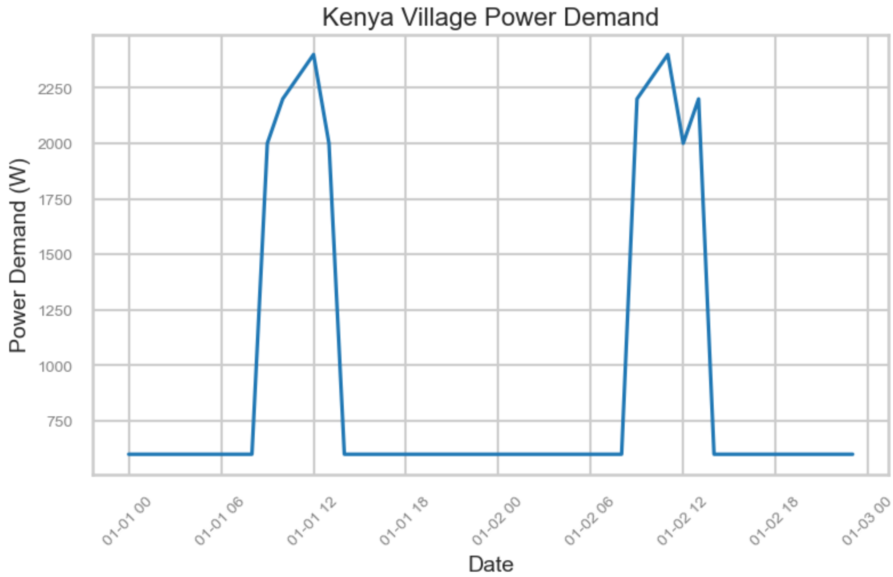



**Figure 3.9:** 48h load curve for communal facility in Kenyan village, peaks during the day indicate washing machine washing cycle and steady load 2 refrigeration units

studies more varied, regarding energy demand.

The load profile is based on the typical hourly coefficients of consumption for a hotel in the Maldives [52]. The coefficients are multiplied by the hourly average electricity consumption. Random noise of $+-5\%$ is introduced to the load curve representing the inherent variability in real-world energy consumption patterns. Seasonal variations in the load profile are also introduced, adjusting the monthly electricity consumption. The adjustments are based on the yearly tourism trends in the Maldives [80]. The total energy is then normalized after the monthly adjustments so the total energy consumed over a year remains constant. The monthly variation and yearly load curve are illustrated in figure 3.10 and a 48 hour load curve is illustrated in figure 3.11.

- **Load curve for data centre in Greece**

  Regarding the load profile of the data centre, the load is based on the peak consumption. The load profile is drawn upon the expected performance of a zero emission data centre located in Chania, Greece which is anticipated to be operational by 2025 [32].

  For this study the peak power is defined as $6.5MW$ [32]. The peak demand is used as a benchmark for the calculations of power consumption. An array of coefficients is defined, reflecting the hourly power demand during different times of day. The coefficients are higher during daytime [51]. Random noise of $+-5\%$ is added to introduce a degree of realism and reflect some of the

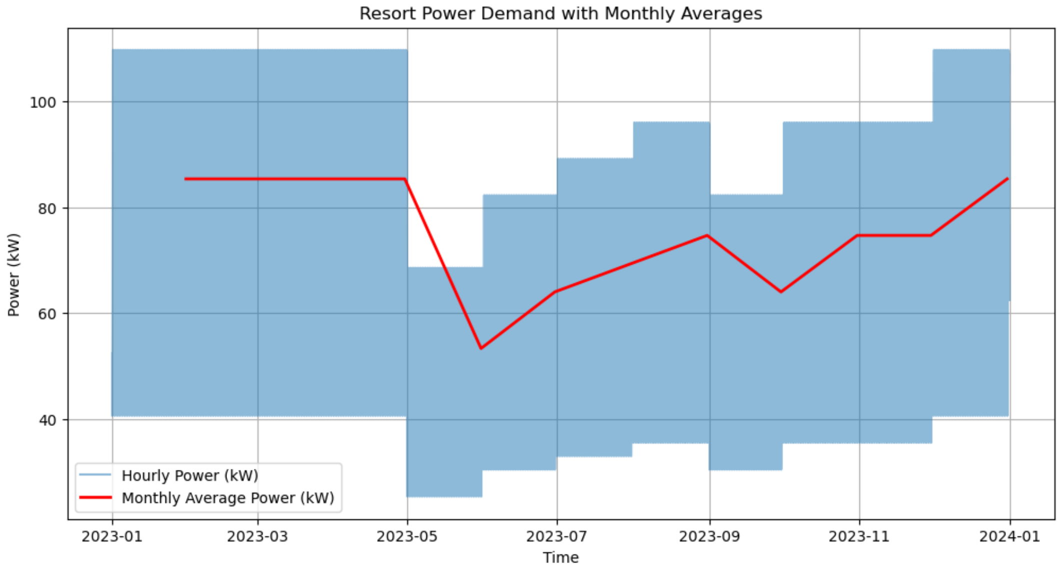



**Figure 3.10:** Yearly Resort load curve with monthly average electricity consumption calculated by number of tourists visiting the Maldives

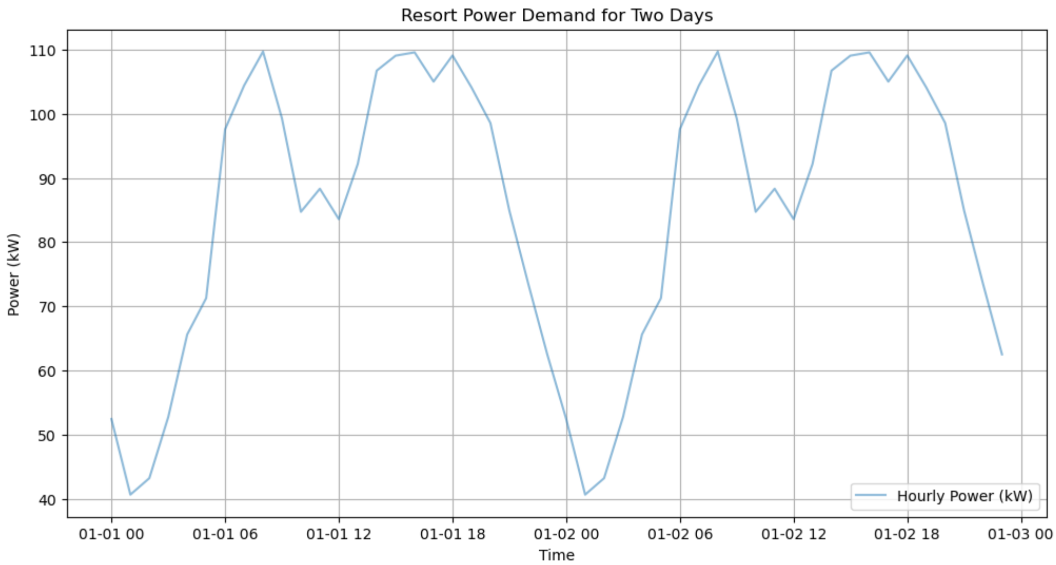



**Figure 3.11:** 48 h resort load curve on the highest consumption month

randomness of real world systems. Finally, some seasonal variation is introduced by reducing the demand by 15% for the months of July and August. The external temperature and conditions for heating or cooling are not taken into account [51]. The code for creating the load curve is presented in the appendix A.10 and a typical 48 hour load curve is illustrated in figure 3.12

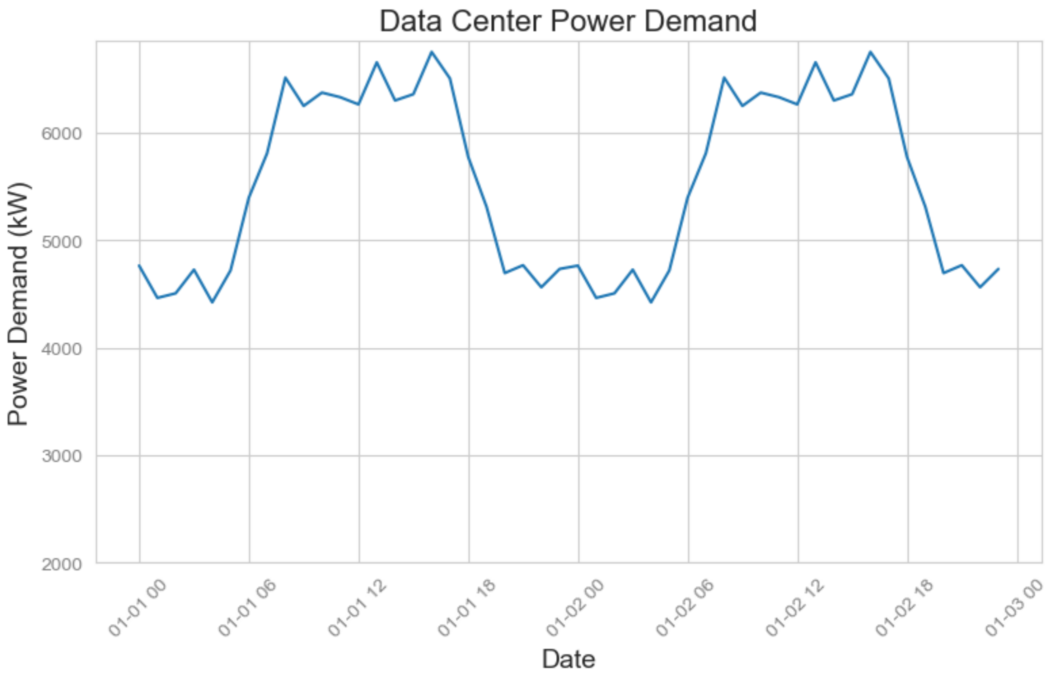



**Figure 3.12:** 48 h data centre load curve

### 3.2.5 Power of components

When it comes to electrolyzer and fuel cell efficiency analysis, computational modeling plays a role of pivotal importance between practical real word applications and fundamental scientific principles. Modern solutions have leveraged the power of computational tools in order to conduct complex simulations and the design of robust systems [99].

The computational model implemented for this thesis focuses on the analysis of the electrolyzer and fuel cell efficiencies through the power output. The model is highly adaptable in the calculation of the efficiency based on the power output for a wide range of power ratings and calculating the efficiency of the stacks for fuel cells and electrolyzers respectively.

- Electrolyzer model

  The model parameters are derived mainly from physical properties and operating conditions of a proton exchange membrane electrolyzer. Said input parameters are the effective area per cell, temperature, pressure at the anode and cathode, number of cells and number of stacks. The activation losses and ohmic losses are modeled in order to derive the cell voltage as output.

  The main governing mechanism behind the model for electrolyzer efficiency analysis is the enthalpy of water formation. In the code, the first implementation of electrochemical mechanism is the Nersnt equation. This equation is used to calculate the open circuit voltage $V_{oc}$, which is the voltage produced, when no current is passing through the cell. Included in the model are also the computation of the activation overpotentials at the anode and cathode ($V_{act}^{an}$ and $V_{act}^{cat}$ respectively). Said parameters are calculated using the Butler-Volmer equation, which uses the exchange current densities at the anode and cathode $i_0^{an}, i_0^{cat}$, charge transfer coefficients ($\alpha_{an}, \alpha_{cat}$), and the activation energies ($Ea_{an}, Ea_{cat}$). Ohmic losses ($V_{ohm}$) are also computed due to the resistance to the flow of ions through the electrolyte and the electrode and interconnections resistance [14]. Those losses exhibit a linear increase with current density. The polarization curve is then generated by the code.

Further, the Faradaic efficiency is computed by the model. The ratio of the actual hydrogen production rate compared to the theoretical rate, given the current passing through the cell is represented by this parameter. Lastly, the first and second law efficiencies are computed, resulting in the overall performance metric of the electrolyzer[75]. The code for the electrolyzer efficiency is presented in the appendix A.12,A.13, A.14, A.15.

It should be noted that there are some simplifying assumptions made by this model, such as the omission of concentration overpotentials and neglecting the change in partial pressure of the reactant across the cell. Despite those assumptions, the tool can be used in system level optimisation and control. Further refinement of the model could include, accounting for thermal effects and considering the effect of degradation.

- Fuel cell model

  The model used for the fuel cell, similar to the electrolyzer model, leveraging principles discussed in [7]. The model includes several fundamental constants, operating conditions, component properties and parameters, influencing the performance of the fuel cell. Those parameters include, temperature, the universal gas constant, Faraday's constant, the internal current density and the fuel crossover among others.

  The key feature of both models is the scalable design, mirroring real world PEM systems where the stack size is varied depending on different power demands. The model establishes an array of current densities, and then utilises this array in calculating the voltage of the fuel cell. The voltage is calculated as a function of current density using the Nernst equation. Other potential losses, relationship between cell potential and current are illustrated. The generated power is then determined as the product of the voltage and the current.

  The functionality of the model is extended to calculating the rate of hydrogen consumption. This enables the evaluation of fuel efficiency as a function of current density. The model calculates the fuel cell efficiency as the ratio of the produced power to the power of the hydrogen that is consumed.

Both models are used in the estimation of the efficiency and the power output of the electrolyzer and fuel cell compared to their maximum power rating. This comparison takes into account the power needed from the load curve depending of the equivalent application. The result of the fuel cell efficiency model for a $5kW$ fuel cell is illustrated in figure 3.13. A comparison with figure 3.14 which is taken from a manual for a $5kW$ fuel cell system by the company PowerCell [1] reveals a similarity in the shape between the theoretical output of the model, and a real world application.

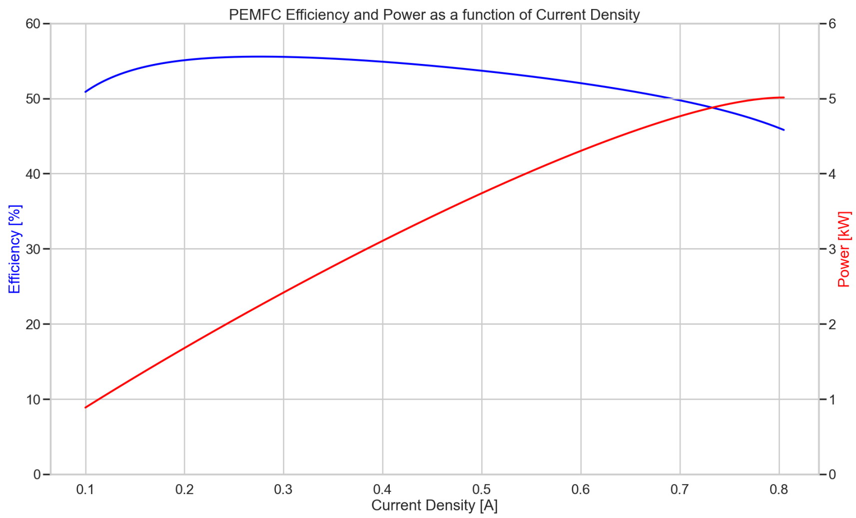



**Figure 3.13:** Efficiency and power of model used for the thesis

### 3.2.6 Economic metrics

In the forthcoming section, the economic metrics used in the microgrid system will be introduced. Those critical metrics are the Capital Expenditure (CAPEX), Operational Expenditure (OPEX), Levelised Cost of Energy (LCOE) and Total Cost of Ownership (TCO). Comprehensive understanding of those metrics is key to the evaluation of the financial feasibility, viability and performance of those micro-grids.

- Capital Expenditure (CAPEX)

  Capital Expenditure refers to the initial investment cost in association with the acquisition, installation, and commissioning of a certain set of assets such

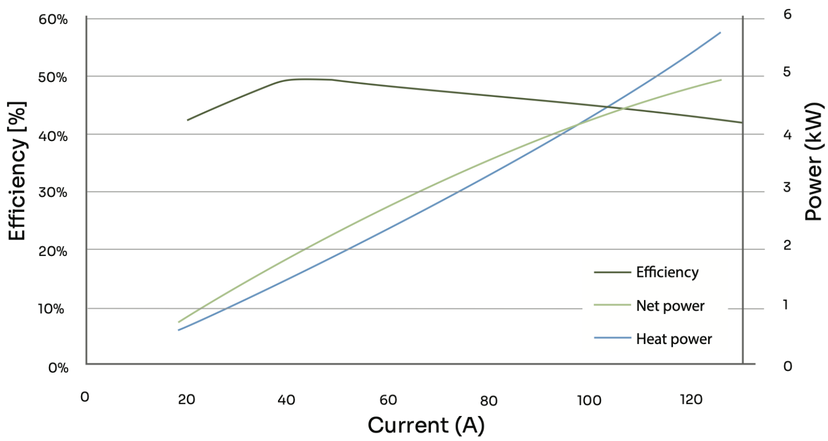



**Figure 3.14:** Efficiency and power form the company PowerCell [1]

as energy systems of microgrids. The cost associated can include the cost of the equipment itself, any necessary site preparations, logistics and transportation or training required to operate the equipment [12]. In the context of microgrids researched in this project, the CAPEX represents the initial cost associated with solar panels, battery systems, electrolyzers, hydrogen storage including compression, fuel cells and all remaining supplementary components. Accurate calculation and evaluation of the CAPEX is essential in order to analyse the implementation of the microgrid and also the assessment of the energy flows optimisation outlined in 3.4.1 and 3.4.2.

- Operational Expenditure (OPEX)

  Operational Expenditure is the ongoing cost required for the operation and maintenance of the equipment associated with the system throughout its lifetime. This cost can include any routine maintenance costs, utilities, consumables such as fuel, operational personnel costs and unforeseen expenses such as repairs or faults [103]. In the context of hydrogen microgrids, the OPEX includes routine maintenance of all the aforementioned equipment or also financing.

- Levelized Cost of Energy (LCOE)

  The Levelized Cost of Energy is an indication of the average cost per unit of energy produced by the system over its operational lifespan. In order to calculate the LCOE, the total cost (including CAPEX and OPEX) is divided by the total energy output of the system over its lifespan. The LCOE is a very valuable, widely used and comprehensive metric for comparing the total cost efficiency of different systems, technologies or strategies [10].

- Total Cost of Ownership (TCO)

  The Total Cost of Ownership offers an overview of all the costs associated with the installation and operation of an asset over its operational lifetime. The TCO includes both CAPEX and OPEX and may also include the depreciation of assets, inflation and opportunity cost of capital [125].

- Payback period

  The Payback period signifies the time needed for an investment to reach a break even point. This concept is very important in determining the feasibility of investments, particularly in capital - intensive sectors such as energy production. For the present study the payback period will be calculated when comparing the system to the operation of a diesel generator.

The aforementioned metrics will form the foundation of the economic evaluation of the hydrogen microgrid outlined in 4.1.2.

### 3.2.7 Economics of scale

When it comes to the economic evaluation of hydrogen microgrids after the work of Vivas et al. [120] provides a comprehensive study of hydrogen based EMS systems for hydrogen microgrids. This study includes the review of 80 publications, 19 of which are including economic decision factors in the objective of optimisation.

Central to the exploration of those studies is the financial evaluation of these microgrids. In the work of Torreglosa et al. [114], Garcia et al. [45] and Trivino et al. [46], a fixed real world system is modeled using either the cost of acquisition for each component per $kW$ or the Net present cost as an optimisation parameter. In the work of Valverde et al [117] an efficiency - cost map is defined based on weather conditions and fixed component prices. The cost values in those publications although robust in their formulation, do not sufficiently encapsulate the dynamism and nuances involved at real world microgrid economic planning.

In stark contrast, the present report deviates from the static approach and embarks on an innovative approach of using scaled cost values. The cost values are depending on the power output of the components, reflecting the price variations and intricacies that are dependent on economies of scale and efficiency increases. This approach although not widely adopted in existing literature, will provide a more realistic evaluation of microgrids.

Even more than the cost scaling of the components, the present study is trying to evaluate the installed cost of the components, rather than the manufacturing cost. In order to make this approximation, the Anual Technology Baselines from the National Renewable Energy Laboratory were used [89], [87], [88], [92], [91], [90] which are 2022 updates on the work of Augustine et al. [6].

To form the basis of the installed cost breakdown of all the system components, the cost for residential scale batteries, commercial scale batteries, residential scale PV and commercial scale PV illustrated in figures 3.15, 3.16, 3.17 and 3.18 were used. The cost of the components and BOP represents 12.4%, 21.1%, 26% and 56.5% of the overall installed cost respectively. Based on those numbers, a coefficient of increase of 3 on the manufacturing cost is applied to all the components. The justification for this value is that the average value for the aforementioned percentages is 30%.

The cost of the installed system components that will be studied is the cost of the electrolyzer per $kW$, cost of fuel cell per $kW$, cost of battery storage per $kW$, solar panels per $m^2$ and cost of hydrogen storage per $kg$. All prices are in US dollars $ Following is a description of each economic model and the illustrated cost scaling curves.

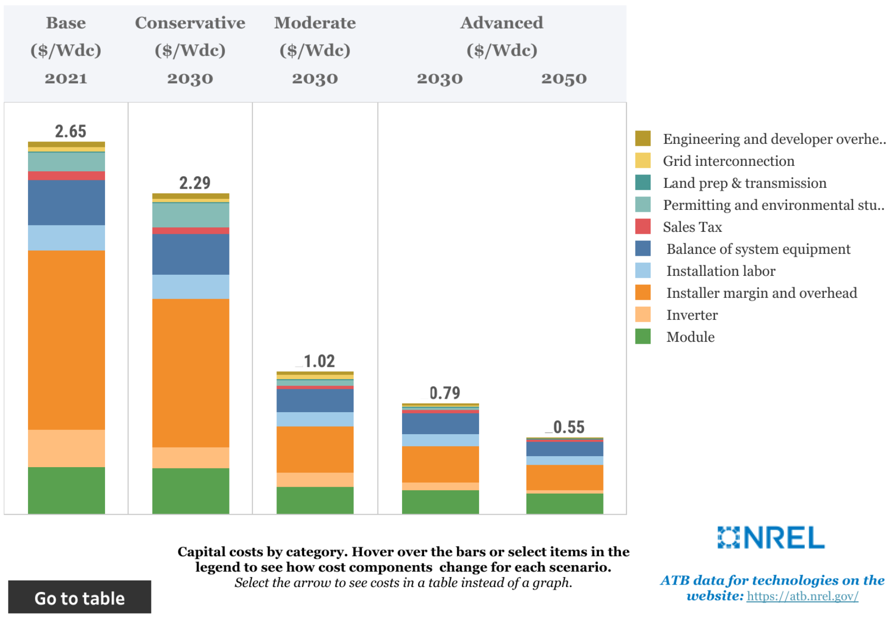



**Figure 3.15:** Residential scale PV component cost breakdown [90]

- PV prices

  Regarding the analysis of solar panel prices across a range of power rating, data form the National Renewable Energy Laboratory were used. NREL provides detailed cost breakdowns and cost projections of PV systems based on application.

  Those applications are, residential, commercial and utility scale [90] [88] [92]. Residential applications are defined as applications that have a power output

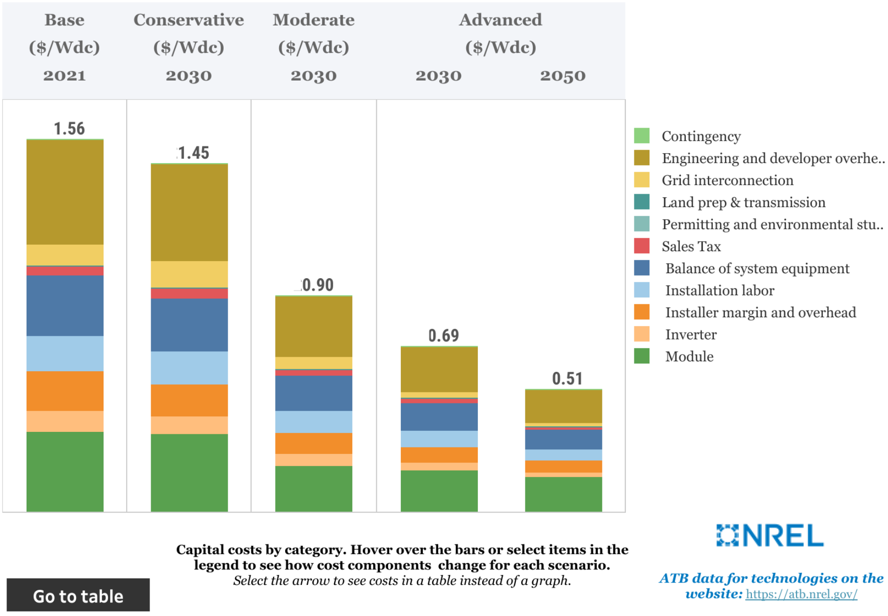



**Figure 3.16:** Commercial scale PV component cost breakdown [88]

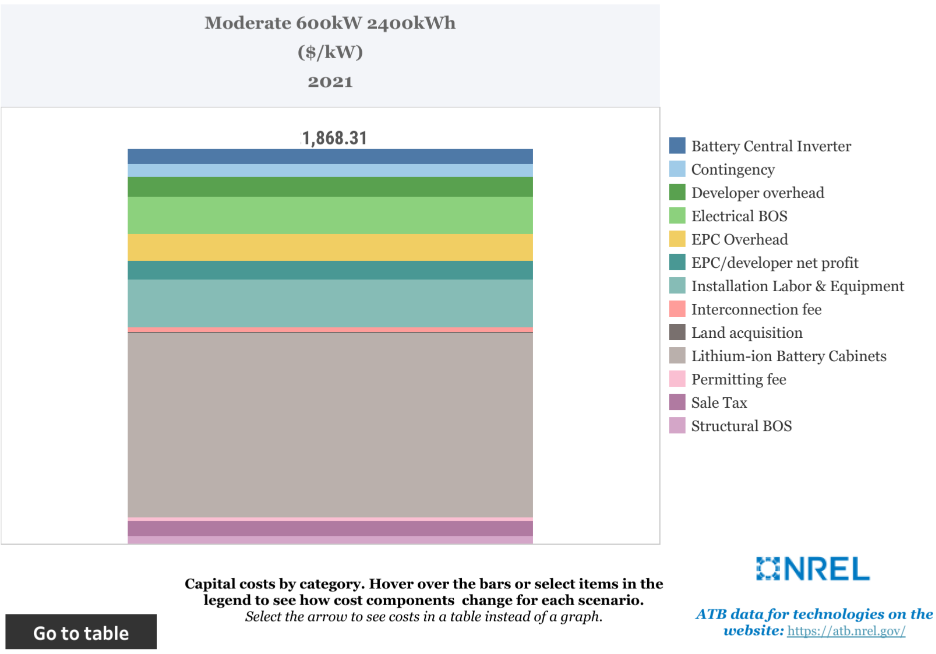



**Figure 3.17:** Commercial scale battery component cost breakdown [87]

less than $100kWp$, between $100kWp$ and $1MWp$ are the commercial applications and above $1MWp$ the system is considered utility scale [92].

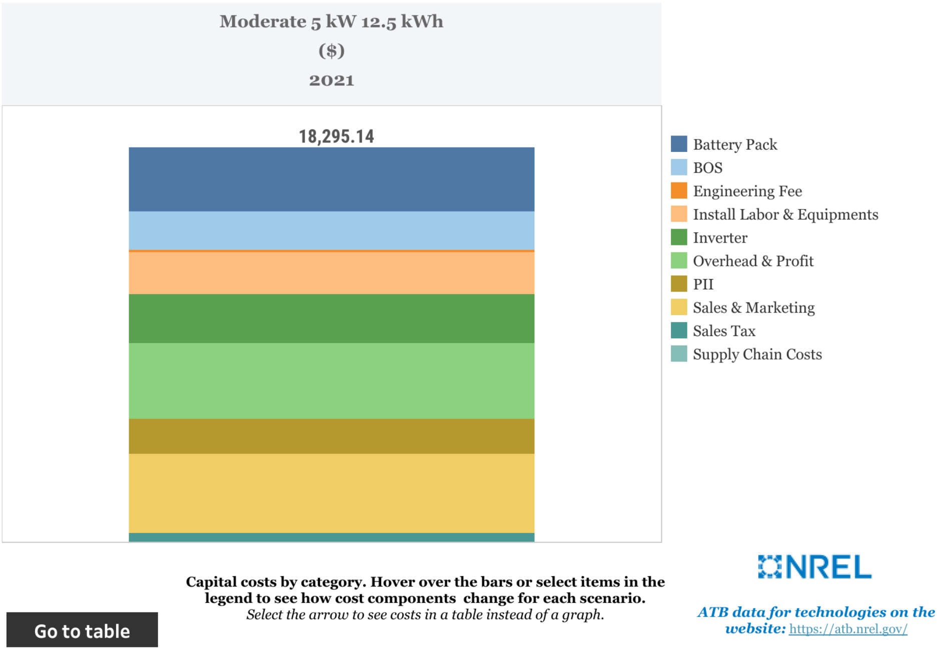



**Figure 3.18:** Residential scale battery component cost breakdown [89]

In order to extrapolate the cost per $kWp$, two lists were defined and a dataframe was created based on those lists. The CAPEX was taken form each of the application price from NREL [90] [88] [92] and the scale from the aforementioned power ratings. The values used for the power rating were the base case for 2021 as those are presented in figures 3.15 and 3.16 and from [92] for utility scale PV. The two lists were then used to make a dataframe, and the script then conducts a linear interpolation between the defined data points. For the interpolation $60,000$ points were used for the values between between $0.01 kWp$ and $10 MWp$.

The data visualisation of the dataframe is done using a logarithmic x-axis. Distinguishing the different power ratings and by extension application area using a different color scheme. The resulting illustration is presented in figure3.19. The Python script for generating the data and graph is presented in A.16.

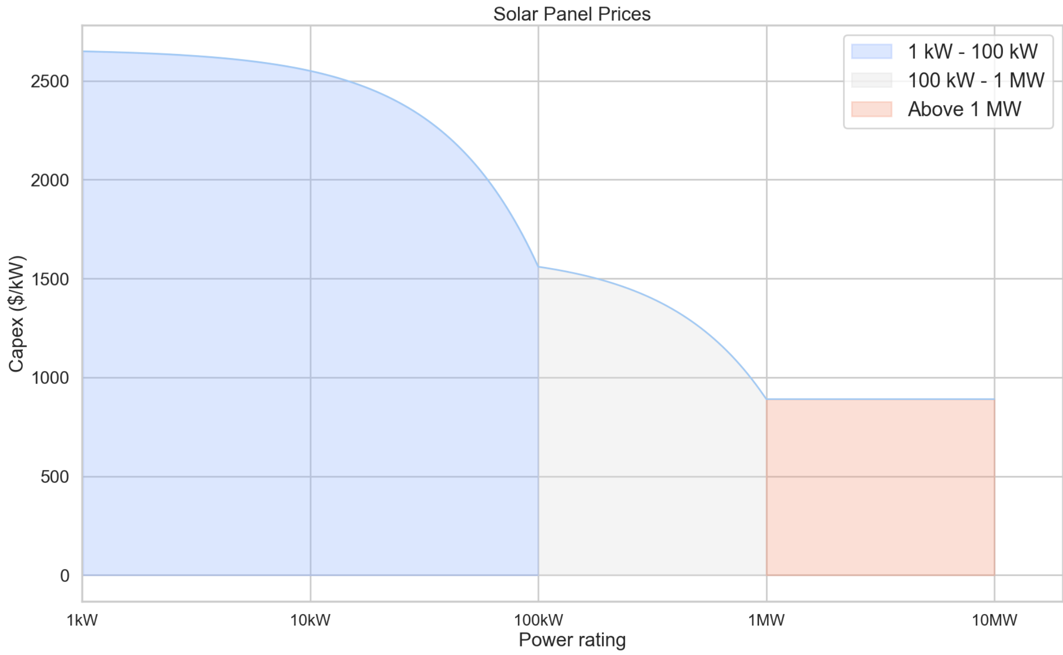



**Figure 3.19:** Solar panel price scaling in a logarithmic x-axis

- Battery prices

  Similar to the cost analysis for PV, battery cost analysis across different capacities used the data for cost breakdown from NREL. The main difference is that battery prices are give as a function of both power rating in $kW$ and capacity in $kWh$.

  For the different scales between residential, commercial and utility scale, the cost for the lowest reported capacity divided by the capacity was used in order to calculate the CAPEX per $kWh$. The power rating was neglected in the calculations. Those prices for residential and commercial are illustrated in figures 3.18 and 3.17 and the utility scale cost is in [91].

  For the data pre-processing, two lists are defined, one representing the battery capacity in $kWh$ and one for the CAPEX. These lists then make up a dataframe, the CAPEX and capacity are taken from the NREL data [89], [87], [91]. The values used for the residential case were the values presented in figure 3.18 divided by the battery capacity. Regarding the commercial and utility scale batteries, the price for 1 hour of autonomy was used divided by the assigned cost as presented in [87] and [91]. A linear interpolation was then performed in the dataframe using $50,000$ points for the values between between $0.01kWh$ and $90MWh$.

  The data visualisation of the dataframe is done using a logarithmic x-axis.

Distinguishing the different capacity ratings and by extension application area using a different color scheme. The resulting illustration is presented in figure3.20. The Python script for generating the data and graph is presented in A.17.

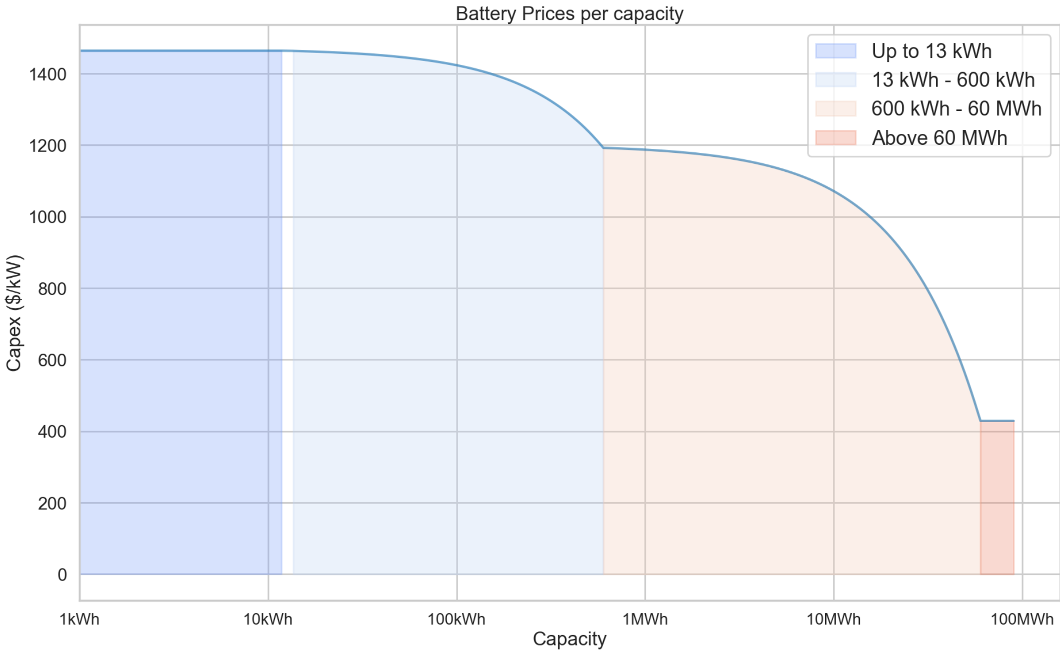



**Figure 3.20:** Battery price scaling in a logarithmic x-axis

- Electrolyzer prices

  The cost analysis of electrolyzers draws data from the prices listed online for Enapter AEM electrolyzers [62], NREL report on electrolyzer manufacturing cost analysis [81] and a presentation made for the company Hydrogenics [112]. From the two latter publications, only prices for PEM electrolysis were used. All prices were multiplied by 3.

  The first three data points are prices for Enapter electrolyzers from the website hyfindr.com [62]. The total price of the electrolyzer was divided by the power of the electrolyzer. A significant cost decrease in the price per $kW$ can be observed even for such small power rating differences. The price for $200kW$ was taken from the report of NREL [81]. The price indicated for $1MW$ was the same between the NREL report and Thomas et al [112]. Prices for $5MW$ and $20MW$ are taken from the report of Thomas et al [112].

  The specified datapoints were passed into two lists , one for price per $kW$ and another for power rating in $kW$. The two lists were used to form a dataframe. Linear interpolation was performed using 50000 points for the values between $0.01 kW$ and $20 MW$.

The data visualisation of the dataframe is done using a logarithmic x-axis. Distinguishing the different power ratings using a different color scheme. The resulting illustration is presented in figure3.21. The Python script for generating the data and graph is presented in A.18.

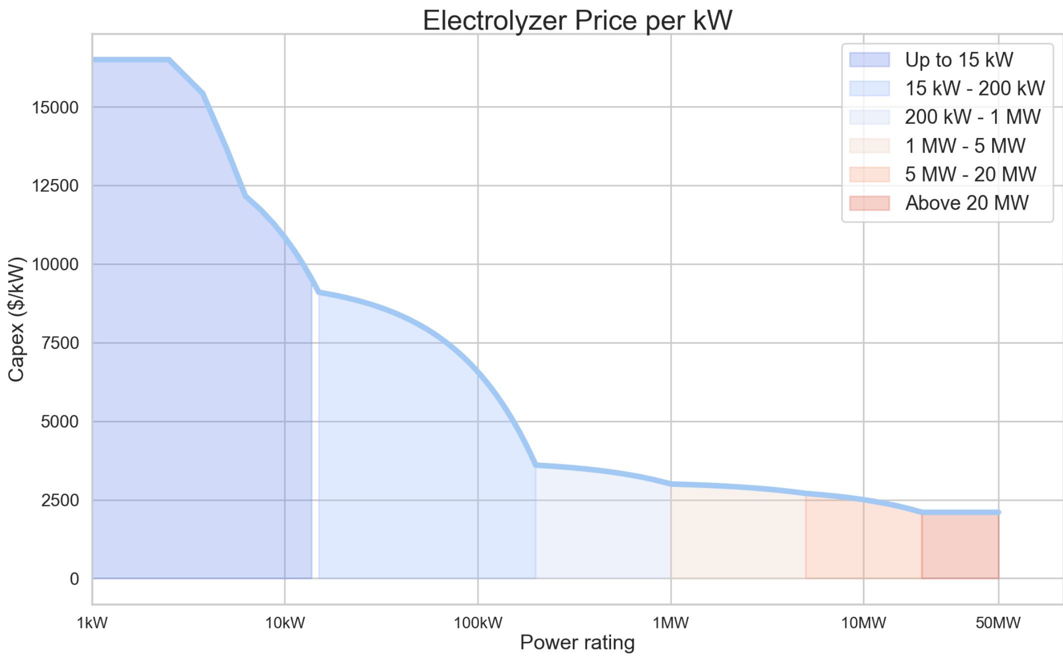



**Figure 3.21:** Electrolyzer price scaling in a logarithmic x-axis

- Fuel cell prices

  The cost analysis for fuel cell cost scaling uses data for PEM fuel cells. The sources used were the 2020 research report of Hydrogen Europe [63], US DOE manufacturing cost analysis for 5 and 10 kw PEMFC [68] and the publication of Cigolotti et al. [21]. All prices were multiplied by 3.

  The first data-points are taken from the US DOE report for $5 - 10 kW$ fuel cells [68] and the report 2020 report of Hydrogen Europe [63]. The further prices for 50 and 500 $kW$ are also from the same report. The cost breakdown for PEMFCs was compared between the two aforementioned reports and [21].

  The specified price and power rating points were assigned to two lists that were used to create a dataframe. Linear interpolation was performed using 50000 points for the values between $0.01 kW$ and $20 MW$.

  The resulting illustration is presented in figure3.22. The Python script for generating the data and graph is presented in A.19.

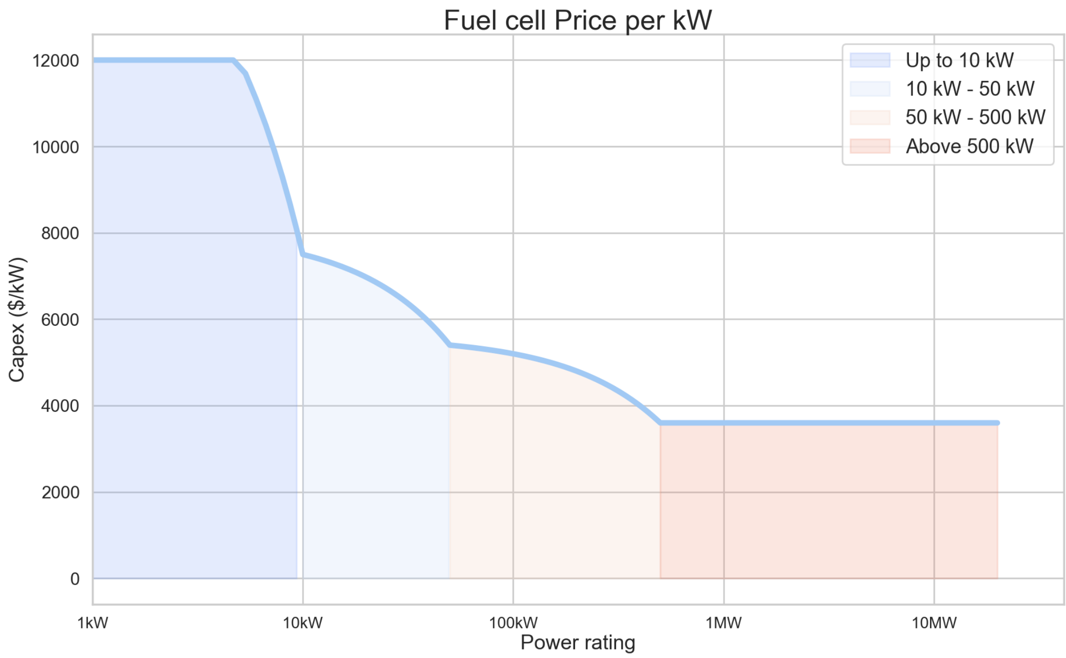



**Figure 3.22:** Fuel cell price scaling in a logarithmic x-axis

- Hydrogen storage prices

  The cost analysis for hydrogen storage cost scaling used prices from industrial and academic sources. For prices bellow $74.8kg$ of hydrogen the price of a Type IV storage tank from the company Steelhead Composites[107] was used from an invoice for a $5kg$ hydrogen tank [64]. The price specified was $50usd/kWh$, or $1650usd/kg$. Prices between $75kg$ and $3t$ prices from Osti.gov [72] and the DOE [65] were compared, where the price for compressed hydrogen storage was $12.8usd/kWh$ or $1267usd/kg$. For hydrogen mass above 3 tons, numbers from the work of Elberry et al [35] were used, according to whom, large scale salt cavern hydrogen storage cost is currently at $2usd/kWh$ or 198 usd per $kg$. All prices were multiplied by 3. The calculations performed for the results of the report were all using the prices per kWh, but for illustration purposes, mass units were used.

  The specified price and storage capacity points were assigned to two lists that were used to create a data frame. Linear interpolation was performed using 50000 points for the values between $1kg$ and $3000t$.

  The resulting illustration is presented in figure3.23. The Python script for generating the data and graph is presented in A.20.

- Diesel generator prices

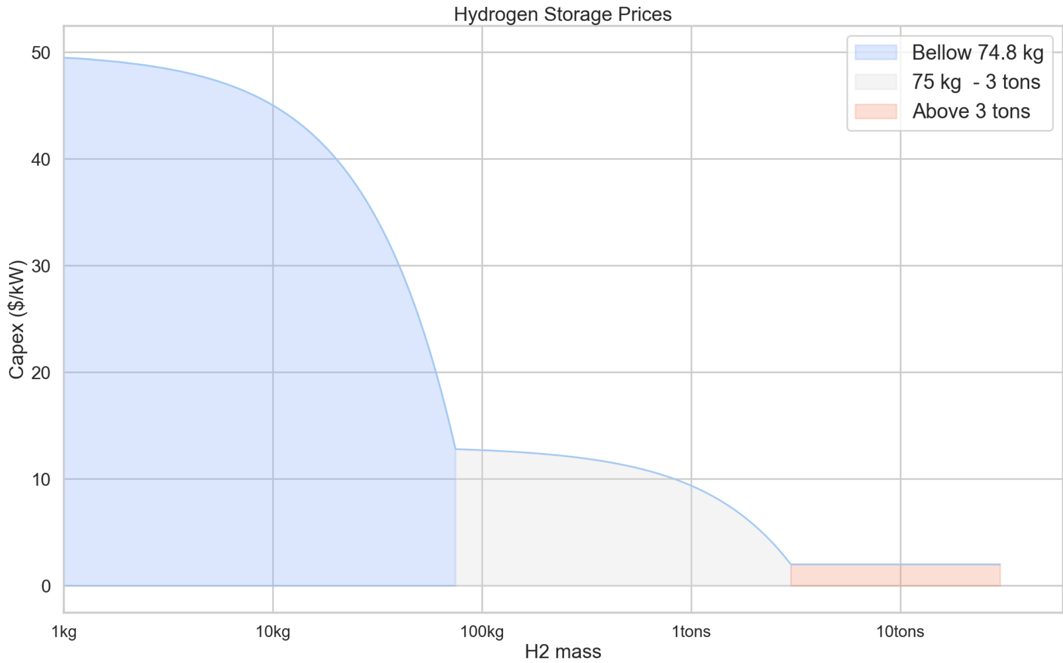



**Figure 3.23:** Hydrogen storage price scaling in a logarithmic x-axis

The cost analysis of the diesel generators takes into account one extra dimension of cost scaling. The scaling is not only done for the CAPEX per power capacity but also for the OPEX per power capacity. The prices for the opex scaling include factors such as the increase in generator efficiency with size and lower maintenance. In the OPEX, the cost of the fuel is also included.

For the creation of the cost scaling curve, different diesel generators with different power ratings were chosen on the website Alibaba [30]. The prices and efficiencies were passed on a table and then the cost per power rating on different points was approximated. The specified price and power rating points were assigned to two lists that were used to create a dataframe. Linear interpolation was performed using 50000 points for the values between $0.1kW$ and $100MW$. All prices were multiplied by 3.

The resulting illustration is presented in figure3.24. The Python script for generating the data and graph is presented in A.21 and A.22.

All the interpolated prices were passed on a python dictionary that could approximate, depending on a given power point or capacity input, the exact price for that point. The python code for generating the dictionary is presented in A.23 and A.24. In figure 3.25 the output of a dictionary call for the price of a $1kW$ electrolyzer, $1kW$ fuel cell, $15kWh$ of battery, $25m^2$ of solar panels, $1kg$ of hydrogen storage and the sum of all the components is presented.

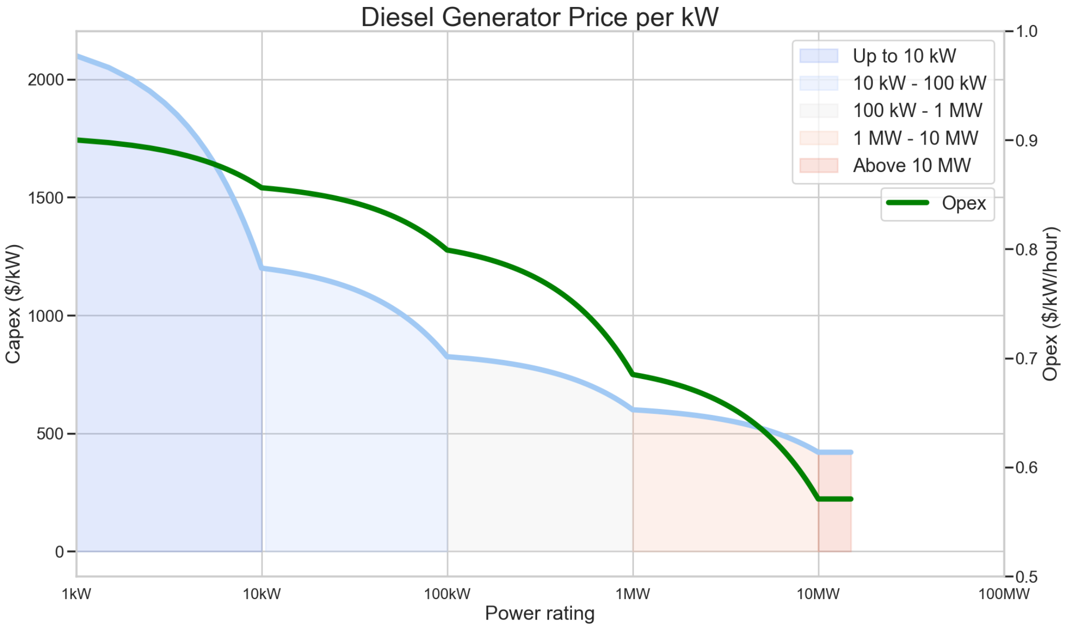
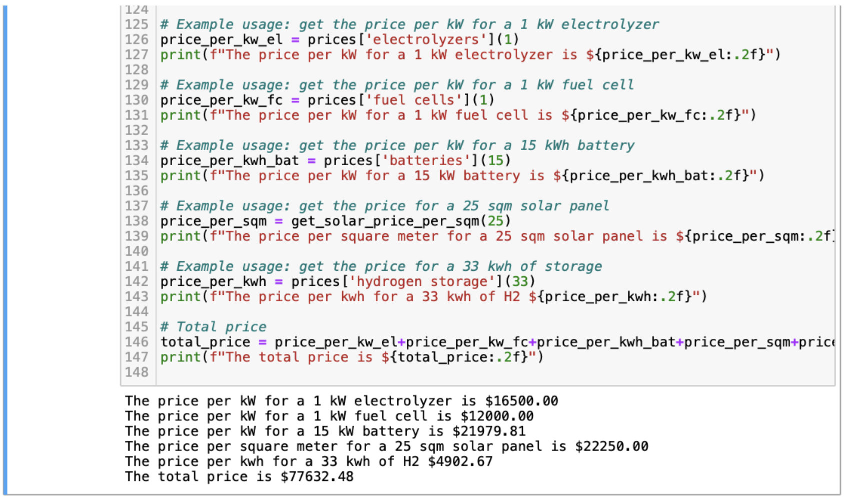



**Figure 3.24:** Diesel generator price scaling in a logarithmic x-axis, OPEX on a secondary axis including fuel cost

The diesel CAPEX was calculated as the instantaneous maximum power output plus 20% of the load curve for each of the cases. The OPEX was calculated from specifying each power point on the OPEX curve for the load curve of each of the cases. The cost was then calculated in hourly OPEX cost and added up to calculate the 20 year diesel TCO for each of the cases.



```
# Example usage: get the price per kW for a 1 kW electrolyzer
price_per_kw_el = prices['electrolyzers'](1)
print(f"The price per kW for a 1 kW electrolyzer is ${price_per_kw_el:.2f}")

# Example usage: get the price per kW for a 1 kW fuel cell
price_per_kw_fc = prices['fuel cells'](1)
print(f"The price per kW for a 1 kW fuel cell is ${price_per_kw_fc:.2f}")

# Example usage: get the price per kW for a 15 kWh battery
price_per_kwh_bat = prices['batteries'](15)
print(f"The price per kW for a 15 kW battery is ${price_per_kwh_bat:.2f}")

# Example usage: get the price for a 25 sqm solar panel
price_per_sqm = get_solar_price_per_sqm(25)
print(f"The price per square meter for a 25 sqm solar panel is ${price_per_sqm:.2f

# Example usage: get the price for a 33 kwh of storage
price_per_kwh = prices['hydrogen storage'](33)
print(f"The price per kwh for a 33 kwh of H2 ${price_per_kwh:.2f}")

# Total price
total_price = price_per_kw_el+price_per_kw_fc+price_per_kwh_bat+price_per_sqm+price
print(f"The total price is ${total_price:.2f}")

The price per kW for a 1 kW electrolyzer is $16500.00
The price per kW for a 1 kW fuel cell is $12000.00
The price per kW for a 15 kW battery is $21979.81
The price per square meter for a 25 sqm solar panel is $22250.00
The price per kwh for a 33 kwh of H2 $4902.67
The total price is $77632.48
```

**Figure 3.25:** Outputs of the python code for cost for all the components

## 3.3 Energy management system

The initial modeling of the energy flows of the hydrogen microgrid was based on the best solutions found in the literature [114] [44]. To enhance the system's overall efficiency and reduce the size of the components, priority was given to the utilisation of batteries. Hydrogen is therefore used as a backup to power the system in those periods when solar power and battery energy storage are incapable of serving the demand. A hysteresis band was implemented, to prevent a possible frequent switching on and off of the fuel cell and electrolyzer. The battery SOC was constrained to a minimum of 40% and a maximum of 90%. Occasionally, the battery may fall above or below these limits. When this situation is encountered, the algorithm will rapidly concentrate on restoring the battery to the recommended levels of SOC[85] [114]. The hysteresis band is consequently activated when the battery SOC is below or above the defined minimum or maximum SOC and it is not deactivated until the battery reaches a preset SOC, which in this case is 60% in case the battery was discharged and 80% in the case battery was overcharged.

The decision-making process is illustrated in figure 3.26. The EMS is presented in a logical algorithmic view that has a specific setpoint output for each component. The presented algorithm was used in the sizing of the system components making sure that the power demand is met and cost is minimised.

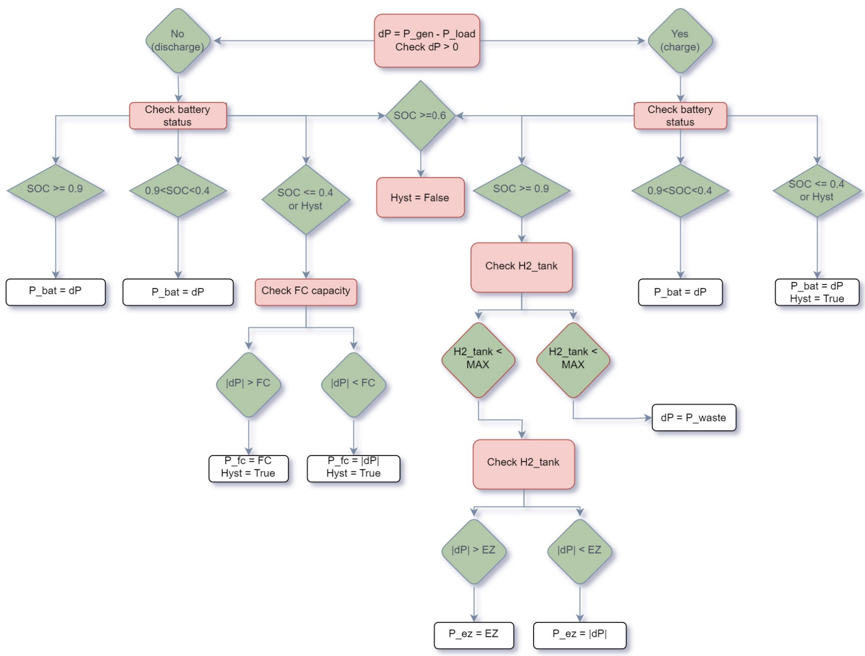



**Figure 3.26:** Energy management system algorithm based on the battery priority strategy for hydrogen microgrid

### 3.3.1 Performance indicators

In order to assess the performance of the energy management systems, key performance indicators (KPIs) need to be chosen.

- Wasted power

  This relates to the energy being generated by the solar panels that cannot be used due to the battery being charged at its max power while the electrolyzer is also fully powered, or hydrogen tank is full. The smaller the value of wasted energy, the better the results.

- Electrolyzer and fuel cell switches

  The electrolyzer and fuel cell switches represent On-Off cycles of electrolyzer and fuel cell. The goal is to minimize them, as they represent higher degradation on the components [113]. The difference between shutdown and idle power is not considered in the calculations.

- Average SOC of battery and hydrogen tank

  The goal is to compare what the average state of charge of both the battery and the hydrogen tank are between both energy management systems.

- Lifespan of components

  Comparison between the management of lifespans of the components between both energy management systems.

## 3.4 Optimisation methods

The two techniques chosen to perform the optimisation in this thesis were a Genetic Algorithm, for sizing purposes and a Linear Programming optimisation, which optimized the energy management through the components.

### 3.4.1 Genetic Algorithm

A genetic algorithm is a population-based stochastic search optimization, which replicates the natural phenomenon of survival of the fittest. Stochastic systems, are non-deterministic since their outcome directly depends on randomness. As mentioned in 3.2.1, GA is broadly used for non-linear problems and problems which might be intractable to formulate, as no mathematical relationships need to be built to find a valid solution. The main advantage of GAs is that they explores a vast search space, making it easier to escape from local minimum solutions. [42]

```
ga = GeneticAlgorithm(
    n_variables = 5,
    n_individuals = 30,
    n_iterations = 1000,
    transformation_type = 'mutation', #transformation
    mutation_type = 'shrink',
    crossover_type = 'uniform',
    gene_selection_type = 'ranking',
    n_selected_individuals = 10,
    fitness_func = fitness_func
    )
```

**Figure 3.27:** Genetic Algorithm settings

### Implementation of GA in hydrogen microgrid sizing

1. **Initialization**

   To initiate the functioning of a Genetic Algorithm, a random population needs to be generated. Each "individual" within the population contains a series of genes. In this case, since the optimization is carried out to size solar PV panels, a battery, a fuel cell, an electrolyzer and a hydrogen tank, this number is set to 5. The preset settings for the optimisation can be seen in figure 3.27.

   The number of individuals indicates the size of the population per iteration. This value multiplied by the number of iterations corresponds to the sizes which are evaluated by the algorithm. The number of iterations is basically the number of different generations that will be simulated. The transformation type is set to mutation, as this is the common procedure to induce new genes in the population. And the number of selected individuals accounts for the individuals that are chosen to pass down their genes. The remainder parameters that can be observed in figure 3.27 are explained in the current section.

   It is beneficial to start with a population that is in the vicinity of the optimal solution. Hence, the algorithm will require fewer iterations to find the desired sizing.

   The initialization of the population is done in the following manner:

   - Solar Panel area that would generate 120% of the consumption needs.
   - Battery size that could assume the load demand for 24 hours by itself.
   - The fuel cell has the capacity to meet the peak power demand

- The electrolyzer is capable of absorbing the peak power generated by the solar panels previously considered.
- The hydrogen tank can provide the fuel cell with enough hydrogen to power the system for 4 days by itself.
- From those genes, a population of 30 individuals is generated with a relative standard deviation or coefficient of variation (CV) of 40%, which represents a moderate to high variability.

2. **Evaluation**

   The primary population undergoes evaluation in the fitness function. This function assesses each individual and grants a singular qualification to each and every one. In the case studied, the aim is to minimize the size of the components, therefore the best qualifications are given to those individuals with a lower result. The way individuals are assessed is explained thereupon.

   - Each individual or possible optimal solution goes through the algorithm described in section 3.3.
   - A variable "count" is defined. This variable adds 1 each time the Battery SOC is above 100% or below 10%. Also when the hydrogen tank is empty or has a negative value.

   The result returned by this function follows this rule:

$$f(x) = \begin{cases} 10 \cdot 2^{(1 - \frac{count}{1000})}, & \text{if } count \geq 0 \\ 1/result & \text{otherwise} \end{cases}$$

   With "result" being the scalar multiplication of the Price vector and the solution vector. The Price vector contains the values defined in section 3.2.7, whereas the solution vector is each one of the sizes in the current iteration.

   This way of granting a score to each individual improves the performance of the GA, as even the unfeasible solutions are ranked ascending in quality. The best individual from each iteration is appended and stored as a list in Python.

3. **Gene selection**

   The selection of the genes is done with a ranking function, which transfers the best fits among the population to the next generation. In this work, the best 10 individuals from each iteration can pass their genes to the next generation.

   To cover also for the uncommon event in which an unfit individual is able to procreate, a 20% randomness is included in the ranking process. Thus, 20% of the time any individual from the population, without condition, will be selected and pass its genes.

4. **Crossover**

   The selected individuals are grouped randomly by pairs 15 times. Each pair represents a parent of two children from the following generation. Some of the individuals can be left out at this point since the process chooses the parents randomly. Additionally, some parents will produce children with different partners. Crossover is done uniformly. Meaning that each dominant gene is selected from each of the parents randomly. By the end of this process, a new population of 30 individuals will be generated.

5. **Mutation**

   According to [42], there are two types of mutation, Random and Shrink. Random means that a gene is substituted by a randomly generated variable. Shrink consists of adding a value to the gene from a uniform, normal, or any other kind of distribution.

   20% of the new population experiences a mutation, which consists of an alteration of their genes. An individual can also be chosen twice and experience two mutations, however, this is very unlikely. The gene is altered by adding or subtracting from the gene (both cases have the same probability of occurring) a mutation coefficient times the gene. The mutation coefficient is set to 30%, being the maximum allowed mutation to occur for a gene. A random number from 0 to 1 in steps of 0.1 creates different possible mutation deviations from the original gene, as presented in equation 3.12.

$$Individual_x[y] = Individual_x[y] \pm mutation_{coef} \times \frac{Rand(0,10)}{10} \times Individual_x[y] \tag{3.12}$$

   After executing the amendments mentioned, steps 2-5 are repeated for a specific number of iterations set by the user. Convergence criteria can be added by setting a maximum amount of iterations to be run without changes or establishing a minimum change of the best score stored so far.

   Before jumping back to step 2 (the fitness function), limits are set for each gene. For instance, none of the components can be below 0, therefore if after a mutation the value falls below the limit, a function brings back the gene to 0. Similarly, the user can set a minimum or maximum value for each component, and the gene is therefore constrained. Solar panels are a clear example of this, as the property where the microgrid is installed can have a limited amount of area available for the PV panels.

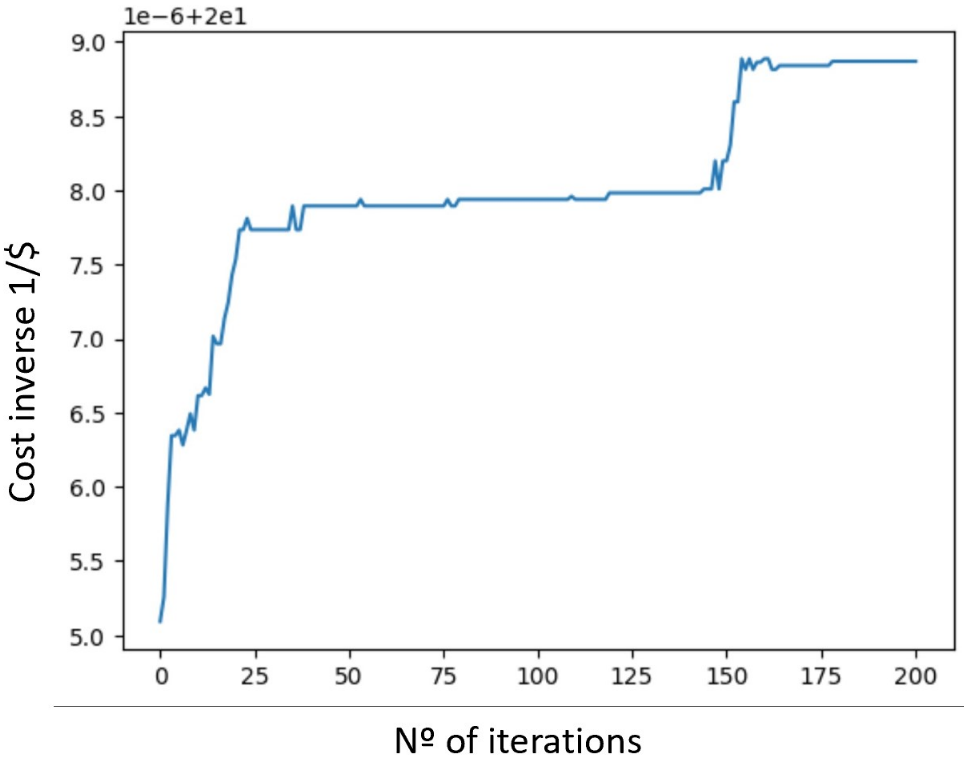



**Figure 3.28:** Genetic Algorithm evolution

6. **Results**

   After the last iteration, it is necessary to transform the results back to the real value, the summation of the cost of each component. An inverse function of the modification in 2 is applied.

$$f(x) = 20 + \frac{1}{result} \tag{3.13}$$

$$f^{-1}(x) = \frac{1}{result - 20} \tag{3.14}$$

   Figure 3.28, shows the evolution of the simulation per iteration. The y-axis reflects the inverse of the cost of the system, as it is modified according to equation 2.

### 3.4.2 PYOMO Linear Programming

There are several optimization paths that can be followed when managing an hybrid microgrid: there can be simply an insurance that demand is followed, a technical optimization that guarantees that the system functions efficiently in its engineering variables, an economic optimization that tries to reduce costs as much as possible and a techno-economic optimization that combines the engineering efficiency with the minimization of the costs. For the purposes of this project, the last optimization method was the one used. In paper *A review of energy management strategies for renewable hybrid energy systems with hydrogen backup*, a great overview of all these optimization methods and their applications is presented. With 80 reports being summarized according to the type of hybrid system, depth of analysis, optimization objectives and constraints [120]. Although all of them present valuable information for the development of the EMS optimization intended, 3 of them were object of further analysis, as they resemble the goal of this project the most:

- *Optimized operation combining costs, efficiency and lifetime of a hybrid renewable energy system with energy storage by battery and hydrogen in grid-connected applications*[47];
- *Definition, analysis and experimental investigation of operation modes in hydrogen-renewable-based power plants incorporating hybrid energy storage*[118];
- *Control based on techno-economic optimization of renewable hybrid energy system for stand-alone applications*[113].

In these attempts, optimizing an energy management system for a hybrid microgrid, there were 4 goals: to ensure demand, minimize costs, maximize lifespans and improve performance. These same goals were the final objective for the optimization process developed for this project. Similarly, the constraints used in this project are in some ways parallel to those presented in the aforementioned reports, such as power balance, SOC of battery and hydrogen tank.

Having established some of the basis studied, the optimization process used for the purpose of this report can now be described. The first decision to be made is the software in which the optimization will be done. As the rest of the coding and calculations previously mentioned in this report are made using the *Python* programming language and the *Pandas Library*, it makes sense to continue using the same software to allow for better integration between each stage of the hybrid microgrid calculations. As such, *Python Optimization Modeling Objects* (*PYOMO*) was used in order to have an optimization tool in Python. *PYOMO* is a *Python*-based open-source software package that supports a diverse set of optimization capabilities for formulating, solving, and analyzing optimization models. By default, it uses *GNU Linear Programming Kit* (*GLPK*), a package intended for solving large-scale linear programming (LP), mixed integer programming (MIP), and other related problems. In the case of this project, only linear programming was used, with the solver behaving like a black box, receiving the functions, constraints and values given, and providing the solution for each time frame.

With the software and solver chosen, the code could now be developed. Firstly, the load curve is obtained as previously explained, along with the generation from the photovoltaic system and the prices of each component based on the sizing earlier obtained through the genetic algorithm. Both the load and the irradiation data are then separated into "chunks", dividing the 20 years worth of data into smaller time sets, depending on the definition, in order for the solver to be able to compute the results in a reasonable amount of time. As such, instead of running the solver for a 1 year problem, the solver would instead run for a daily problem, on 365 iterations, as an example. Having all the data necessary, the optimization using *Pyomo* can then be done.

Initially, all the values must be included inside the *Pyomo* function. With this in mind, a dictionary is created in order to be used as input for the function. After this step, the function is created and a *model.T* is generated that indicates to the solver how many steps are in the time frame that has to be solved. Inside this function, both parameters and variables are created. Parameters are fixed values throughout the application of the linear optimization and through time, such as the cost of the components, which require a initialization with a specific value. For this initialization, the previously created dictionary that was input to the function is used. An example of this parameter can be seen in figure 3.29. Parameters are also used to input the previously obtained load and generation values for each moment in time. Variables on the other hand do not have any previously attributed values, and are the unknowns that the system is solving for. Examples for the definition of a variable that can assume general values, and a variable used as a binary can be seen in figures 3.30 and 3.31.

```
model.battery_capacity = Param(initialize = settings['battery_capacity'])
```

**Figure 3.29:** Parameter set in *PYOMO* function

```
model.SOC_Tank = Var(model.T, domain = NonNegativeReals, bounds = (0,model.tank_capacity))
```

**Figure 3.30:** General variable set in *PYOMO* function

```
model.EZ_switch = Var(model.T, within = Binary, bounds = (0,1))
```

**Figure 3.31:** Binary variable set in *PYOMO* function

Having all the parameters (capacity of battery and hydrogen tank, maximum power of battery, fuel cell and electrolyzer, cost of components, generation and demand) and variables (SOC of battery and hydrogen tank, current power of battery, fuel cell and electrolyzer, and some smaller binary variables to help the solver) defined, the objective function and constraints are established in order for the optimization problem to be solved.

Firstly, the objective function is set. Since the sizing of the components is made separately with the use of the genetic algorithm, the only cost optimization that can be done with the EMS is by maximizing the lifespan of components, therefore minimizing re-acquisition costs. As such, a cost function was used as the objective function, composed of 4 different costs. The cost of each component is calculated by multiplying the cost of the component by the percentage of lifespan used per hour. In the case of electrolyzer and fuel cell, on/off cycles also consume an extra 3 hours of lifespan [113]. The first is the battery cost per hour, represented in equation 3.15.

$$Bat_{CostperHour} = Bat_{Cost} \times (\frac{(Bat_{ch} + Bat_{dch})}{Bat_{cap} \times Bat_{upw} \times Bat_{warr}} + \frac{(1 - Min_{SOC}) \times 0.25}{Bat_{warr}} + \frac{Max_{SOC} \times 0.25}{Bat_{warr}} \tag{3.15}$$

The cost per hour is in function of the total battery cost ($Bat_{Cost}$), the total charge and discharge of the battery during the hour ($Bat_{ch}$ and $Bat_{dch}$), the battery capacity ($Bat_{cap}$), the battery's usable power ($Bat_{upw}$) and finally the warranty of the battery in cycles ($Bat_{warr}$) [113]. The $Min_{SOC}$ is dependant on a constraint that makes it 0 when the battery is below 0.2 SOC, attributing a cost to the battery being below this value, so the model avoids it, but allows in case it is necessary energy wise. While the $Max_{SOC}$ follows the same principle, but for a maximum SOC value before there is extra degradation [8]. Following the battery cost per hour, the equation for the fuel cell cost per hour can be seen in equation 3.16.

$$FC_{CostperHour} = FC_{Cost} \times \frac{FC_{ON} + 3 \times FC_{Switch}}{FCwarr} \tag{3.16}$$

The cost per hour of the fuel cell depends on the total cost of the fuel cell ($FC_{Cost}$), a binary indicating if the fuel cell was on in that hour ($FC_{Power}$), a switch that represents an On-Off cycle $FC_{Switch}$ and the warranty of the fuel cell in hours of usage ($FC_{warr}$) [113]. Finally, the equation for the electrolyzer cost per hour is in equation 3.17.

$$EZ_{CostperHour} = EZ_{Cost} \times \frac{EZ_{ON} + 3 \times EZ_{Switch}}{EZwarr} \tag{3.17}$$

Just like in the case of the fuel cell, the cost per hour of electrolyzer depends on the total cost of the electrolyzer ($EZ_{Cost}$), a binary indicating if the electrolyzer was on in that hour ($EZ_{Power}$), a switch that represents an On-Off cycle $EZ_{ON}$ and the warranty of the electrolyzer in hours of usage ($EZ_{warr}$) [113]. Combining all of these equations, the total cost/objective function is represented in equation 3.18.

$$Bat_{CostperHour} + FC_{CostperHour} + EZ_{CostperHour} + Waste_{Power} \times Waste_{Cost} \tag{3.18}$$

The component costs per hour are included, along with the waste power, that is introduced so that it is also being minimized in the objective function, with the $Waste_{Cost}$ being equal to the average cost of electricity in a country or the cost of diesel, depending on the location. The definition of the cost function in the code is represented in figure 3.32.

```python
def objective_rule(model):
        return  sum((model.cost_battery * (model.P_Bat_ch[t] + model.P_Bat_dch[t])/(model.battery_capacity*model.usable_power*model.warranty_cycle)
                +  (model.cost_fc * (model.FC_ON[t])/(model.warranty_hours_fc))
                + (model.cost_ez * (model.EZ_ON[t])/(model.warranty_hours_ez))
                + (model.cost_ez * 2 * (model.EZ_switch[t] /(model.warranty_hours_ez)))
                + (model.cost_fc * 2 * (model.FC_switch[t] /(model.warranty_hours_fc)))
                + (model.cost_battery *(1-model.Min_SOC[t])*0.25/(model.warranty_cycle))
                + (model.cost_battery * model.Max_SOC[t] * 0.25/(model.warranty_cycle))
                + (model.P_waste[t] * 0.001) for t in model.T)
model.Objective = Objective(rule=objective_rule, sense=minimize)
```

**Figure 3.32:** Objective function set in *PYOMO* function

The objective function minimized the cost, by trying to achieve the maximum lifespan of the components. Having set this objective function, the constraints must now be set. The first constraint exists to ensure that demand is followed. As such, equation 3.19 is set.

$$\begin{aligned} Demand = -\frac{Bat_{ch}}{Bat_{eff}} + (Bat_{dch} \times Bat_{eff}) \\ + FC_{Power} - EZ_{Power} + Generation \\ \times Degradation_{PV} + Waste_{Power} \end{aligned} \tag{3.19}$$

The parameters that were not defined previously are the demand from the application in question, battery efficiency ($Bat_{eff}$), the generation and the degradation of the photovoltaic system. This constraint equals the demand to the power drained from the battery, the energy generated with the solar panels and the power from the fuel cell, while it takes into account the energy that goes into the electrolyzer and the waste power. The following constraint defines the SOC of the battery, as demonstrated in equation 3.20.

$$Bat_{SOC}[t] = Bat_{SOC}[t-1] + (Bat_{ch}[t] - Bat_{dch}[t]) \tag{3.20}$$

The time defining "[t]" exists in the previously mentioned equations, however it was not represented in order to avoid a cluster of too much information and too many variables. However, for this constraint, it is necessary, as the SOC of the battery is calculated taking the SOC in the previous moment in time and adding the charged power and subtracting the discharged power. It should also be noted that the SOC in the first moment of the solution has to be an input, as the equation cannot run for moment [t-1] at instant 0. This constraint is represented in figure 3.33, both to demonstrate how the first time step is defined, but also to present how constraints are defined in general.

```
def SOC_BAT_rule(model,t):
        if t == model.T.first():
                return model.SOC_B[t] == model.battery_init + model.P_Bat_ch[t] - model.P_Bat_dch[t]
        else:
                return  model.SOC_B[t] == model.SOC_B[t-1] + (model.P_Bat_ch[t] - model.P_Bat_dch[t]
model.SOC_Bat = Constraint(model.T, rule = SOC_BAT_rule)
```

**Figure 3.33:** Constraint set in *PYOMO* function

With this constraint in mind, and keeping the topic on the battery, another two constraints were included in order to guarantee that the battery cannot charge and discharge at the same time. This is represented in equations 3.21 and 3.22.

$$Bat_{dch}[t] <= (1 - Binary[t]) \times Bat_{MaxDischarge} \tag{3.21}$$

$$Bat_{ch}[t] <= Binary[t] \times Bat_{MaxCharge} \tag{3.22}$$

In these equations, a binary is introduced, along with the max charge and discharge capacity of the battery. The purpose of this binary is that, when it is 1, it only allows for the battery to charge, and when it is 0 it only allows for the battery to discharge. It also gives space for both to be 0, with the less than or equal to the max charge/discharge rate. The last battery constraints are presented in equations 3.23 and 3.24.

$$Bat_{SOC} >= Min_{SOC} \times Bat_{cap} \times Min_{cap} \tag{3.23}$$

$$Bat_{SOC} <= Bat_{cap} \times Max_{cap} + Max_{SOC} \times Bat_{cap} \times (1 - Max_{cap}) \tag{3.24}$$

These constraints take the battery capacity and multiply it by a specific state of charge, from which the battery gets more degradation upon usage. In this case, there is both a 0.2 minimum capacity, and in the case of that constraint, the binary $Min_{SOC}$ becomes 0 when the battery is below this threshold, therefore assuming an additional cost in the objective function. As for the maximum, if the battery goes over the limit of 0.8, the binary becomes 1 and assumes, yet again, a cost [8]. It should be noted that both of these are soft limits, meaning the battery can surpass these values, but the optimization tries to avoid so in order to reduce costs.

With the battery side fully defined, the hydrogen system now needs its constraints regarding the SOC of the hydrogen tank. This constraint can be seen in equation 3.25.

$$Tank_{SOC}[t] = Tank_{SOC}[t-1] + (EZ_{Power} \times EZ_{eff} \times Compressor_{eff}) - \frac{FC_{Power}}{FC_{eff}} \tag{3.25}$$

Just like in the battery SOC, the SOC of the hydrogen tank each a moment in time is dependant on its status on the previous time period. Taking that previous time period, the tank charges with electrolyzer power, while taking into account the efficiency of the electrolyzer itself and the compressor necessary to get the hydrogen pressure to the values of the tank. It also discharges the tank with the fuel cell and once again takes into account its efficiency. Similarly to the battery, the SOC of the tank in the first moment of time has to be an input as there is no possibility for the solver to solve for [t-1] at time = 0. After these components are constrained, the electrolyzer and fuel cell's functioning can be managed. Firstly, the constraint that relates the binary representing the On status in an hour and the power used by the electrolyzer and fuel cell is represented in equations 3.26 and 3.27 respectively.

$$EZ_{Power} <= EZ_{ON} \times EZ_{MaxPower} \tag{3.26}$$

$$FC_{Power} <= FC_{ON} \times FC_{MaxPower} \tag{3.27}$$

In these equations, the power supplied by either the electrolyzer or fuel cell are dependant on a binary ($EZ_{ON}$ and $FC_{ON}$) and the maximum power that each can supply. At the same time, logistically it does not make sense for the electrolyzer and fuel cell to be able to work simultaneously, as either generation is in excess, so it can follow demand and provide power to the electrolyzer, or there is not enough generation to follow demand and not enough battery power, and therefore the fuel cell would need to follow the demand and the electrolyzer would remain in idle. As such, a constraint was designed to emulate this behaviour, and it is represented in equation 3.28.

$$FC_{Power} <= (1 - EZ_{ON}) \times FCMaxPower \tag{3.28}$$

This constraint makes it so that when the electrolyzer is on, and therefore the binary is 1, the fuel cell power has to be smaller or equal to 0, and the fuel cell power can only assume zero or positive values, hence the fuel cell power being zero when the electrolyzer is on. The final constraints in the realm of the electrolyzer and fuel cell are the ones used to define the switch, that allows a cost to be attributed to On-Off cycles. For this purpose, a binary called "Help" was defined using equations 3.29 and 3.30.

$$Help_{EZ}[t] <= EZ_{ON}[t-1] \tag{3.29}$$

$$Help_{FC}[t] <= FC_{ON}[t-1] \tag{3.30}$$

So these binaries assume the value zero if the component was off in the time step before, or can assume one or zero if the component was on in the hour before. It should also be noted that at time step zero (t = 0), these binaries are hard coded to assume the value zero, as the model would not be able to calculate a value for $EZ_{ON}$ and $FC_{ON}$ at t = -1. These binary values can now be used in the definition of the switches, as represented in equations 3.31 and 3.32.

$$Switch_{EZ}[t] = EZ_{ON}[t] - EZ_{ON}[t-1] + Help_{EZ}[t] \tag{3.31}$$

$$Switch_{FC}[t] = FC_{ON}[t] - FC_{ON}[t-1] + Help_{FC}[t] \tag{3.32}$$

Both switches depend on the state of the electrolyzer at the current time stamp and the previous one. At the same time, the help variable exists to regulate for the fact that if the electrolyzer is off in the current hour, but on in the previous hour, the switch would return a value of -1 which is not supported by the binary quality of the variable. As such, the constraint help compensates so that the value -1 is not returned. At timestamp zero, the components $EZ_{ON}[t-1]$ and $FC_{ON}[t-1]$ are replaced by input values in the function. The last constraints are concerning the waste power.

$$Waste_{Power} <= Irradiance_{Max} \times Waste_{ON} \tag{3.33}$$

Firstly, the boundary is set for the waste power in equation 3.33. In this constraint, the waste power is maximized by the max value of irradiance within the set data of solar irradiation on the application, and it is multiplied by a binary that indicates if there is waste power or not. This binary is relevant for the following two equations 3.34 and 3.35.

$$Bat_{dch} <= (1 - Waste_{ON}) \times Bat_{MaxDischarge} \tag{3.34}$$

$$FC_{Power} <= (1 - Waste_{ON}) \times FC_{MaxPower} \tag{3.35}$$

These two constraints make it so that there cannot be waste power while the fuel cell is working or the battery is discharging. This makes logistical sense because if there is a demand, and enough generation, this generation should be following the demand, instead of wasting power and forcing either the battery or the fuel cell to compensate.

Last step of this function is to setup the outputs, allowing for the user to access SOC of both battery and tank, power of electrolyzer and fuel cell, demand and photovoltaic generation, at all time steps.

This function is now ready to be used in a *for* loop. The selection of a *for* loop was made in order to solve all the chunks sequentially, while also updating certain values of the energy system within the loop. As such, inside the loop, the function was ran, and the results of each solution of the function were used to both update battery and hydrogen tank values for the next solution, but also store the value of the electrolyzer and fuel cell binaries, in order for the optimization code to know the value of the switches.

Besides the aforementioned functionalities within the *for* loop, the degradation was also calculated in this section of the code. Firstly, the degradation of the maximum power capacity of the battery was calculated using the number of cycles warranted by *AutoX* and *LG*, who warrant 6000 cycles with a final remaining capacity of 60% [100][76]. As such, as the *for* loop was ran, when enough time frames were solved to complete a month, the total charge and discharge energy of the battery was taken. The number of cycles were then calculated considering the maximum capacity of the battery, and, for each cycle, a percentage of $\frac{0.04}{6000}$ was deducted from the battery capacity.

Following the battery, the degradation of the electrolyzer and fuel cell were also calculated. For the calculation of the electrolyzer, 2% degradation per 8000 hours was considered [97], while for the fuel cell 0.4% per 1000 hours was taken into account [116]. Using these values, the amount of hours of use for each period of a month was taken and the degradation on these components introduced.

Finally, the degradation of the photovoltaic panels simply takes the flat degradation of 3.6% at the end of 25 years of warranty, and linearly distributes it every month [40].

In order to allow for longer simulations if desired, when the degradation reaches the final value expected, whether it is 60% for the battery or 80% for the fuel cell and electrolyzer, the degradation is reset in order to represent the acquisition of a new component.

# Chapter 4

# Sizing of microgrid and economic analysis results

In this chapter, the specific sizing of all the components of a hydrogen microgrid is presented. The sizing process is executed by the Genetic Algorithm, described in section 3.4.1. The code finds the best fit by simulating the 20-year energy flows of multiple microgrid sizing possibilities. The selection of the best solution is based only on cost.

## 4.1 Sizing results

The results in table 4.1 display significant variations, primarily due to the diverse loads, the shape of the demand curve, and the sole reliance on solar irradiation for energy provision.

| | **Kenya (FACILITY)** | **Netherlands (HOUSE)** | **Maldives (HOTEL)** | **Greece (DATA CENTER)** |
|---|---|---|---|---|
| **PV panels** | 30 $m^2$ | 69 $m^2$ | 3,820 $m^2$ | 299,964 $m^2$ |
| **Battery** | 24.2 kWh | 24.5 kWh | 1,979 kWh | 125 MWh |
| **Fuel cell** | 1.6 kW | 3.3 kW | 167 kW | 5,575 kW |
| **Electrolyzer** | 1.3 kW | 4.2 kW | 418 kW | 4,896 kW |
| **H2 tank** | 14.4 kg | 163.4 kg | 3.2 t | 247 t |
| **Consumption (yearly)** | 8,221 kWh | 8,753 kWh | 1,288 MWh | 55,874 MWh |

**Table 4.1:** Final sizing for each case study

In both the Kenyan facility and the private property in the Netherlands, there are constraints on the solar panel area, limited to a maximum of 30 $m^2$ and 69 $m^2$, respectively. One expected takeout deduced from these results is that the seasonality of solar irradiance in the Netherlands compared to Kenya accentuates the need for long-term hydrogen energy storage to endure the challenging winter conditions. Since the utilisation of hydrogen is larger, thus the fuel cell and electrolyzer are sized accordingly.

In regards to the Hotel in the Maldives and the Data Center in Greece, the numbers are one order of magnitude larger compared to the previously discussed. They also differ greatly between them, so it is not easy to extract conclusions here. It is worth mentioning that in these two last cases, the solar panel area was not restricted, thus the energy wasted by these two systems is expected to be larger. The result for the solar area in the two cases appears as a large number but for the sake of comparison, the largest solar farm in Greece has an area of solar panels of 1.36 $million\ m^2$ [93] and in Denmark 3.74 $million\ m^2$ [110].

In these cases, it is very convenient to have a hydrogen off-taker, such as another resort or a refueling station. Hydrogen can also be used in a green hydrogen grid balancing plant to provide power and ancillary services to the power grid, in case there is a connection in the surroundings of the microgrid.

### 4.1.1 Component specification and cost

In table 4.1, a complete breakdown of the price of the components is presented. For Kenya, the complete installation cost of the microgrid totals 163,369 $. For the Netherlands, the price rises up to 409,283 $, which is more than double for a similar yearly energy consumption. The main driver of the cost difference is the hydrogen tank, which accounts for nearly 50% of the total cost of the house in the european country. On the other side, moving to the more energy-demanding microgrids, the cost is mainly driven by solar panels and batteries.

To provide a better understanding of this table, pie charts showing the share of the cost for each component are illustrated in figure 4.2. For Greece and Maldives, solar panels and battery account for as much as 71% and 84.6% of the total price respectively. The hydrogen side of the microgrid is, in these cases, not the primary factor influencing the price, while it is for the smaller cases. The fact that the PV panels are not a constraint also influences the results, but the economies of scale and primarily the cheaper hydrogen storage for big quantities of gas are the main factors behind the cost reduction.

It can also be seen how in those places closer to the equator, Kenya and the Maldives, the requirement for hydrogen storage is less as the sun irradiation is not affected largely by seasonality.

It is also worth noting how the price is influenced by the fuel cell and the

| | KENYA (LAUNDRY) | NETHERLANDS (HOUSE) | MALDIVES (HOTEL ) | GREECE (DATA CENTER) |
|---|---|---|---|---|
| PV panels | 26,700 $ | 61,510 $ | 3,399,800 $ | 266,967,960 $ |
| Battery | 35,560 $ | 44,269 $ | 3,677,751 $ | 182,927,740 $ |
| Fuel cell | 19,152 $ | 39,456 $ | 823,644 $ | 20,070,000 $ |
| Electrolyzer | 20,790 $ | 62,158 $ | 1,436,457 $ | 13,257,388 $ |
| H2 tank | 61,167 $ | 201,890 $ | 632,472 $ | 48,989,446 $ |
| TOTAL | 163,369 $ | 409,283 $ | 9,970,124 $ | 532,212,534 $ |

**Figure 4.1:** Breakdown of prices for each case study

electrolyzer for the Data Center in Greece, less than 7% of the total.

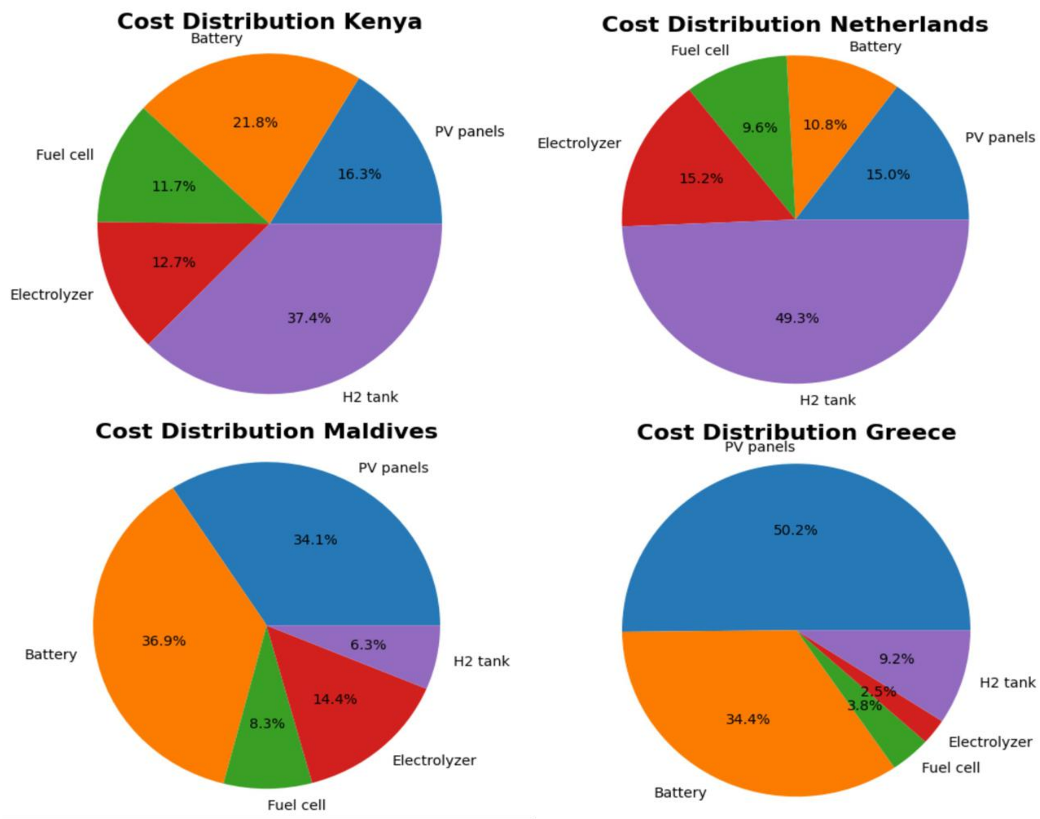



**Figure 4.2:** Comparison of component Cost Breakdown for the cases studied

### 4.1.2 Economic analysis

Hereby, the economic results of the different microgrid designs are presented in figure 4.3. The CAPEX is described in the previous subsection 4.1.1. The OPEX is calculated as a percentage of the CAPEX [63],[112],[90],[89], for the microgrid case and as the summation of the maintenance and fuel cost for its diesel counterpart. The TCO and LCOE are the metrics used to compare the hydrogen microgrid, a diesel generator and the national power grid.

In all cases, the price of electricity from the electrical grid is much more affordable than either of the other two alternatives. Therefore, it does not make much economical sense to build a microgrid wherever the national grid is already present. In remote locations, a deeper feasibility study could be done to discern which option is economically more viable.

Looking at the Break even column from figure 4.3, it can be observed that all cases are above 20 years from the moment when it becomes preferable to construct a hydrogen microgrid, cost-wise. The fact that all cases are above 20 means that in any case, it is more affordable to produce electricity via diesel than by constructing a hydrogen microgrid since the expected lifetime of the microgrid is 20 years. However, the final decision on which system to build is influenced by factors beyond just monetary considerations. $CO_2$ emissions, deforestation or energy ownership are elements that lean the decision towards a microgrid.

The Kenyan facility and the data center in Greece are the cases where a microgrid is closer to being economically viable. Additionally, the possible commercialization of hydrogen which is not considered in this study, would highly influence the results shown in table 4.3.

| Case | CAPEX | OPEX (yearly) | TCO (20y) | TCO (20y) Diesel | TCO (20y) Grid | Payback period | LCOE (H2) | LCOE (Diesel) | Price grid |
|---|---|---|---|---|---|---|---|---|---|
| Kenya | 163,369$ | 15,97$ | 195,309$ | 153,752$ | 21,211$ | 26.1 years | 1.187 $/kWh | 0.973 $/kWh | 0.129 $/kWh |
| Netherlands | 409,184$ | 40,64$ | 490,464 $ | 170,698 $ | 84,912$ | 104 years | 2.33 $/kWh | 0.975 $/kWh | 0.485 $/kWh |
| Maldives | 9,970,124$ | 199,482 $ | 13,959,764 $ | 10,804,798 $ | 1,951,092 $ | 30 years | 1.077 $/kWh | 0.830 $/kWh | 0.150 $/kWh |
| Data centre | 560,751,188 $ | 4,325,102$ | 618,714,574 $ | 603,713,250 $ | 212,602,071$ | 20.59 years | 0.648 $/kWh | 0.641 $/kWh | 0.226 $/kWh |

**Figure 4.3:** Economic metrics for each case study

# Chapter 5

# Energy management optimisation results

In this chapter, the results of the regular EMS using the Genetic Algorithm are compared to the optimized PYOMO results. Both energy flows and the performance indicators are presented and discussed. For the reference case of showcasing the energy flows in the system, the case for the house of the Netherlands was chosen due to the yearly seasonality and also the familiarity with the order of magnitude of the results.

## 5.1 Energy and hydrogen flows

Firstly, the energy and hydrogen flows of both the Genetic Algorithm and the PYOMO are used. Both codes give us the hourly flow of energy in and out of the battery, the electrolyzer and fuel cell powers, which lead to variation in the battery and hydrogen tank levels, and generation and demand characteristics.

The first graphs presented show the full energy flows over the 20 year period simulated by both the Genetic Algorithm and the PYOMO optimization in figures 5.1 and 5.2, respectively.

In both these graphics, the periodicity of the hydrogen tank SOC can be seen, with it being refilled to the maximum every summer and used every winter. At the same time, the effects of the worst case scenario for irradiation present every 5 years can also be observed, with the hydrogen tank failing to completely refill during the summer in these cases, for both simulation softwares. It is also apparent that both the fuel cell and battery charge are maximized by the PYOMO software, working at values much closer to both max fuel cell power and max discharge power respectively.

In order to further see the fluctuations in the flows of energy and hydrogen, the first graphs are amplified and excerpts of 4 days for both the Genetic Algo-

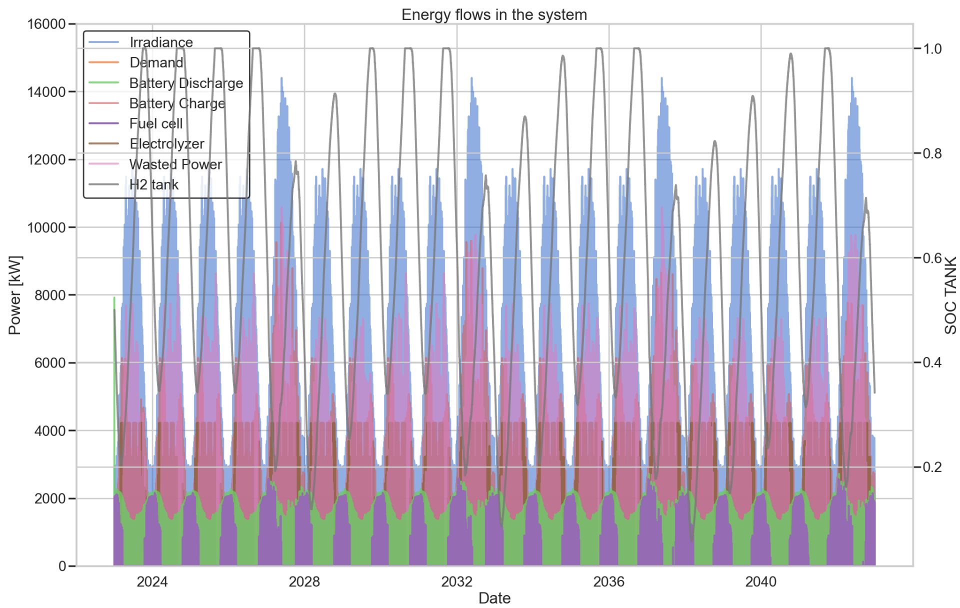



**Figure 5.1:** Energy flow values for the entire Genetic Algorithm simulation of a 20 year energy system of a house in the Netherlands

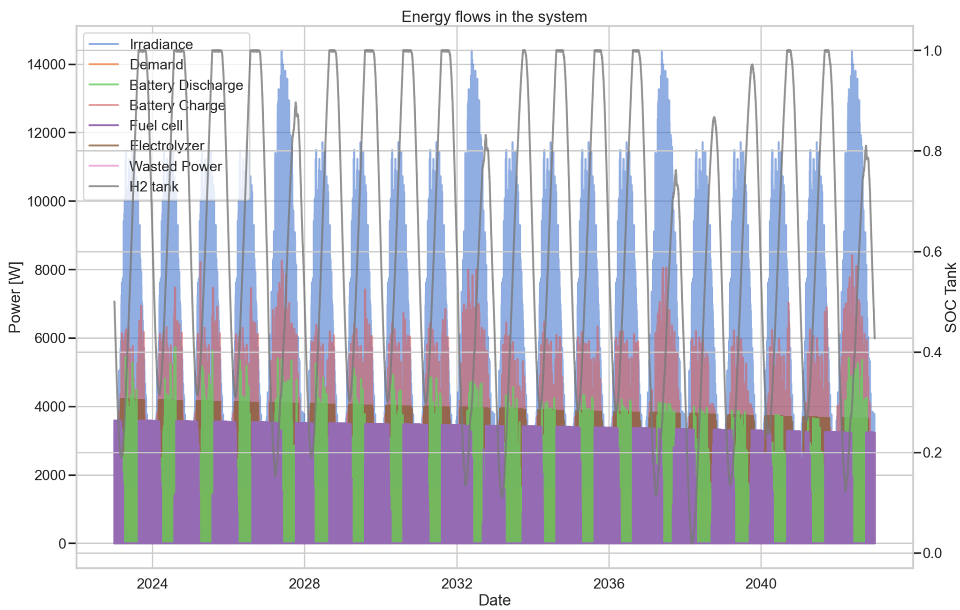



**Figure 5.2:** Energy flow values for the entire PYOMO simulation of a 20 year energy system of a house in the Netherlands

rithm and the PYOMO are taken. These timeframes are then inspected for both an average irradiation year and a worst case scenario for irradiation. Energy and hydrogen flows for the house in the Netherlands, with the first case showing values for an average year of irradiation in figure 5.3 and the second case presenting the worst case irradiation in figure 5.4, both using Genetic Algorithm.

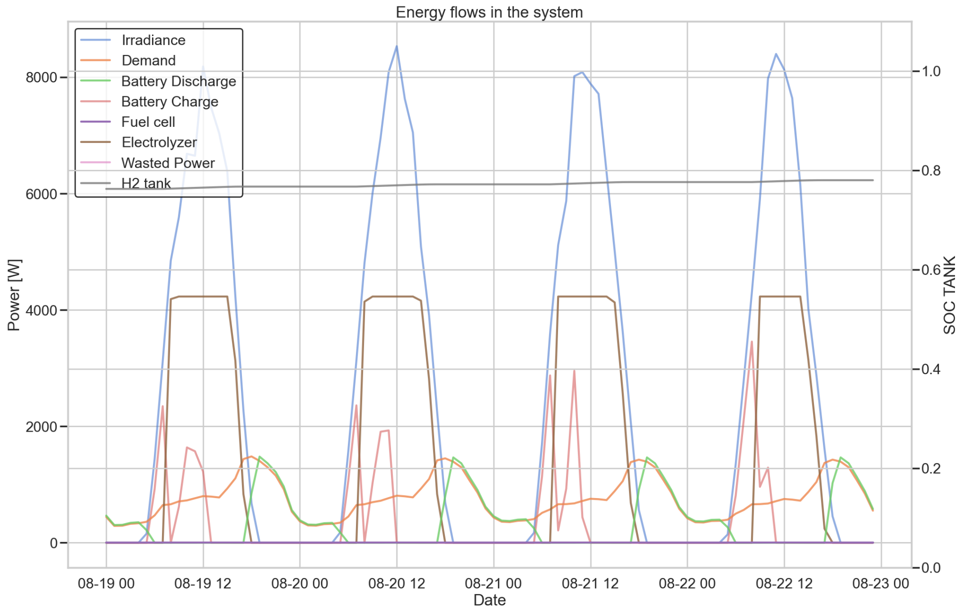



**Figure 5.3:** Energy flow values over 4 days, in the summer of an average year, for the Genetic Algorithm simulation of a 20 year energy system of a house in the Netherlands

In these images, the impact of the worst year of irradiation is very visible when compared to an average irradiation year, with some days showing irradiation levels that require the microgrid to use the hydrogen system, even in the summer. The same can be seen for the PYOMO, in figures 5.5 and 5.6.

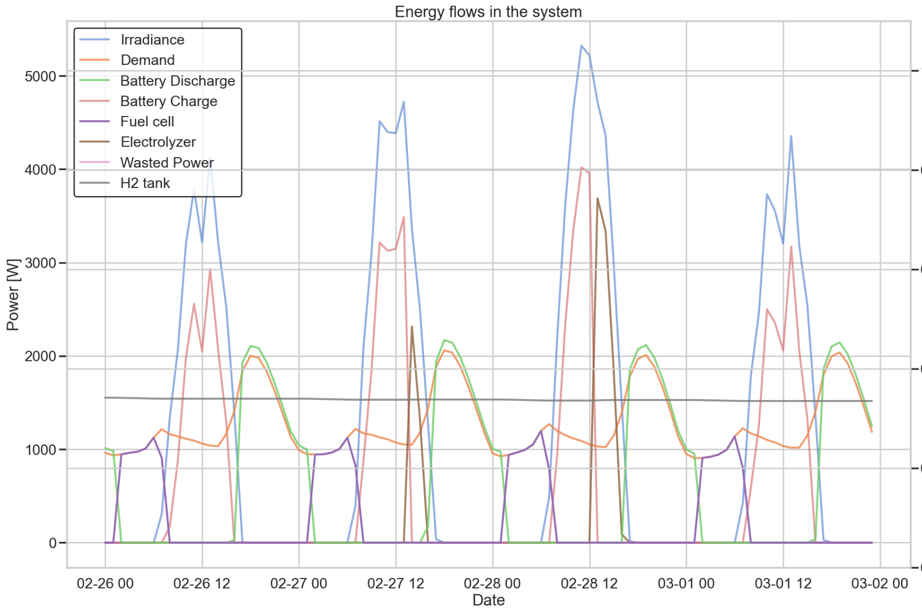



**Figure 5.4:** Energy flow values over 4 days, in the summer of the worst year, for the Genetic Algorithm simulation of a 20 year energy system of a house in the Netherlands

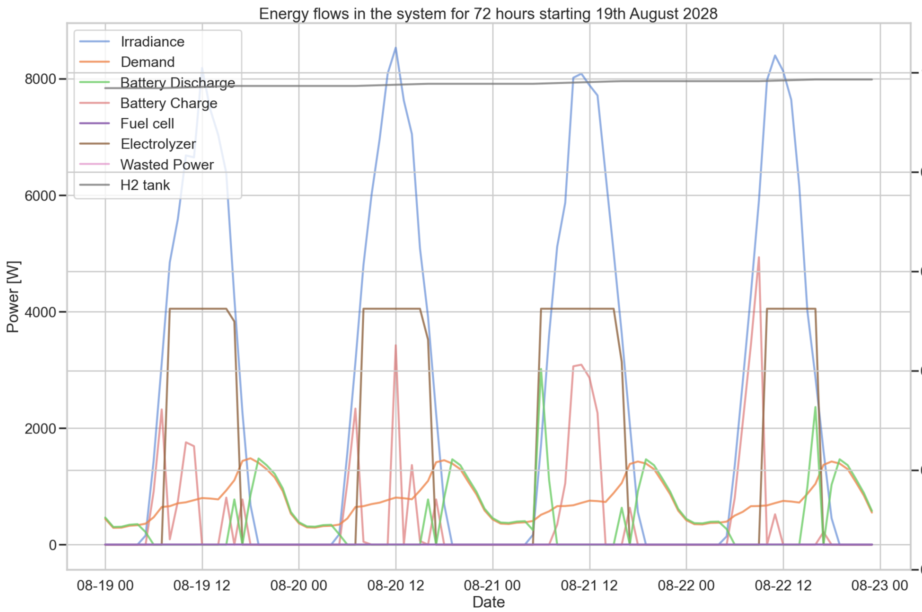



**Figure 5.5:** Energy flow values over 4 days, in the winter of an average year, for the PYOMO simulation of a 20 year energy system of a house in the Netherlands

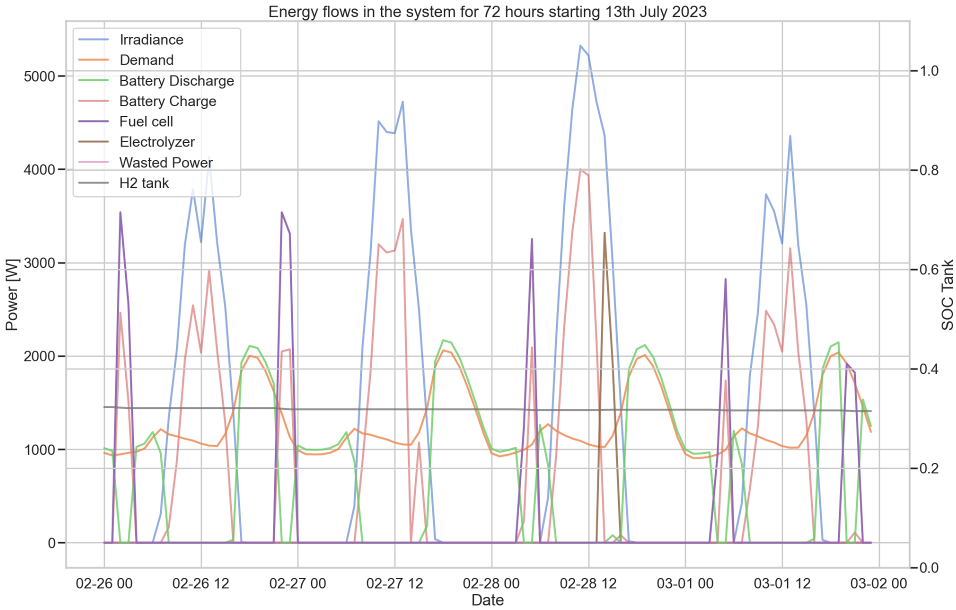



**Figure 5.6:** Energy flow values over 4 days, in the summer of the worst year, for the PYOMO simulation of a 20 year energy system of a house in the Netherlands

## 5.2 Performance indicators

The first performance indicator to evaluate is the wasted power. As such, a graph can be seen in figure 5.7 representing the waste power as it gets summed up over time.

This figure shows that the PYOMO optimization did a great job at reducing the wasted power, by having a difference of nearly 20000 kW in power lost. Considering the non optimized system waste upwards close to 80000 kW, this represents a reduction of approximately 25%. Further comparison of the waste power numbers is represented in table **??**.

| | Percentage of Power Loss (%) | Average Wasted Power (W) | Probability of Wasted Power (%) |
|---|---|---|---|
| Genetic Algorithm | 48.38 | 483.5 | 14.71 |
| PYOMO | 37.82 | 377.9 | 13.24 |

In the table, it can be seen that there is a significant reduction in the percentage of waste power lost, when compared with the total demand power (percentage of power lost) . Along with this, the average wasted power per day is also much lower, however the probability of wasted power, which is the number days in which there is wasted power over the total number of the days, is very similar. This means that, although the average value of the power wasted is lower in the PYOMO, the

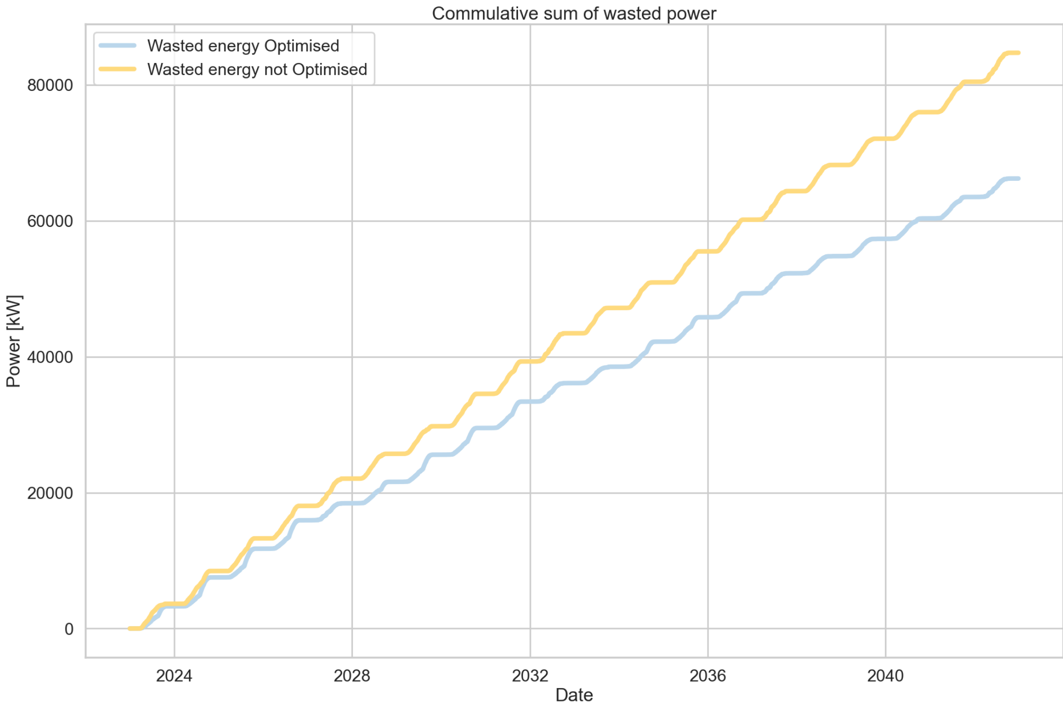



**Figure 5.7:** Comparison of total wasted power, between the Genetic Algorithm, and the optimization using PYOMO

number of days in which there is waste is very similar between both softwares. Also, although reduced, the waste power using the PYOMO optimization is still very high, and as such it should be analyzed why this wasted power exists. In order to better do this analysis, heat maps representing the wasted power per month through the years in the Genetic Algorithm and the PYOMO optimization are presented in figure 5.8.

These graphs better show the distribution of the wasted power over time. In both softwares, waste power is present in a period of the year from April to September, with the Genetic Algorithm presenting higher peaks of waste power in the month of August. The source of this waste power can also be explained by the time frame in which it occurs. As these months present higher irradiation values, there will be high influxes of energy from the photovoltaic system. As the battery and the electrolyzer both try to use this power influx, they are both limited by their maximum charge capacity and electrolyzer power, respectively. As such, if the irradiation power is higher than the sum of the maximum charge capacity and electrolyzer power, there will be waste power. At the same time, there is also a limitation created by the battery and hydrogen tank reaching maximum SOC. If this happens, the system will again accrue in having wasted power.

With the waste power comparison made, the use of the fuel cell and electrolyzer

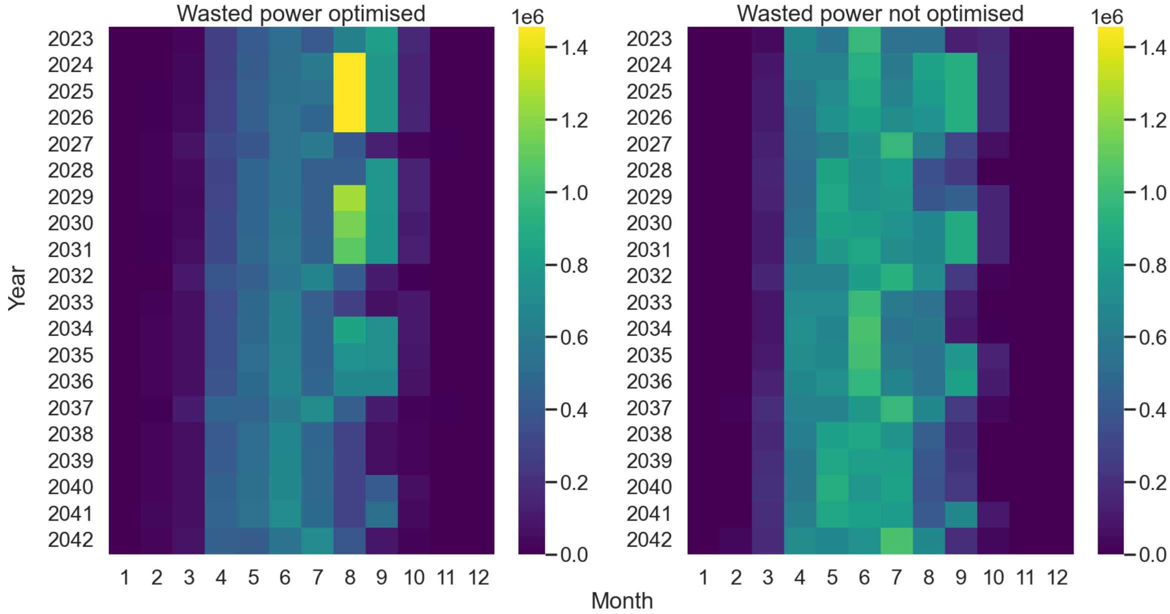



**Figure 5.8:** Heat map of the wasted power per month over the years in the Netherlands, using Genetic Algorithm on the left, and PYOMO optimization on the right

can now be analyzed. For this purpose, the number of hours that both pieces of equipment are utilized is represented in figure 5.9.

This figure shows that, even though the electrolyzer usage is very similar between both softwares, the PYOMO optimization clearly edges out when it comes to hours of run time of fuel cell. This reduction by approximately 15000 hours of fuel cell usage represents a 40% reduction when compared to the Genetic Algorithm, extremely favoring the lifespan of the component. However, when it comes to the amount of On-Off cycles of both the electrolyzer and fuel cell, the management systems performed very differently, as can be seen in figure 5.10.

In this figure, it can be clearly seen that the PYOMO optimization results present a significantly higher value of switches, with each switch representing an On-Off cycle. The reasoning behind this will be analyzed at the end of this section, while encompassing all the results at hand.

Having gone through fuel cell and electrolyzer usage, the average SOC of both battery and hydrogen tank are a factor of interest when analyzing how both softwares performed. These results are present in figure 5.11 and 5.12.

These graphs show that the PYOMO optimization steadily maintained a lower value of the battery SOC over the months, which is explained by the constraint in the code that introduces a cost to the battery working above 80% charge capacity due to extra degradation, with the case being the same below 20%. The impacts of this coding choice will be further analyzed again at the end of this section. As

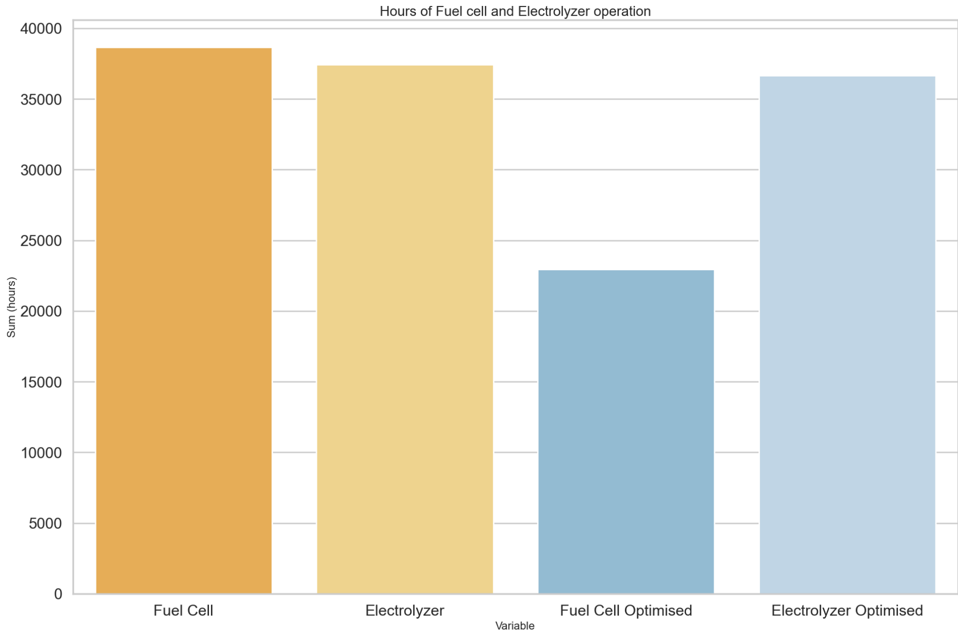



**Figure 5.9:** Comparison of hours of power of both fuel cell and electrolyzer, between the Genetic Algorithm, and the optimization using PYOMO

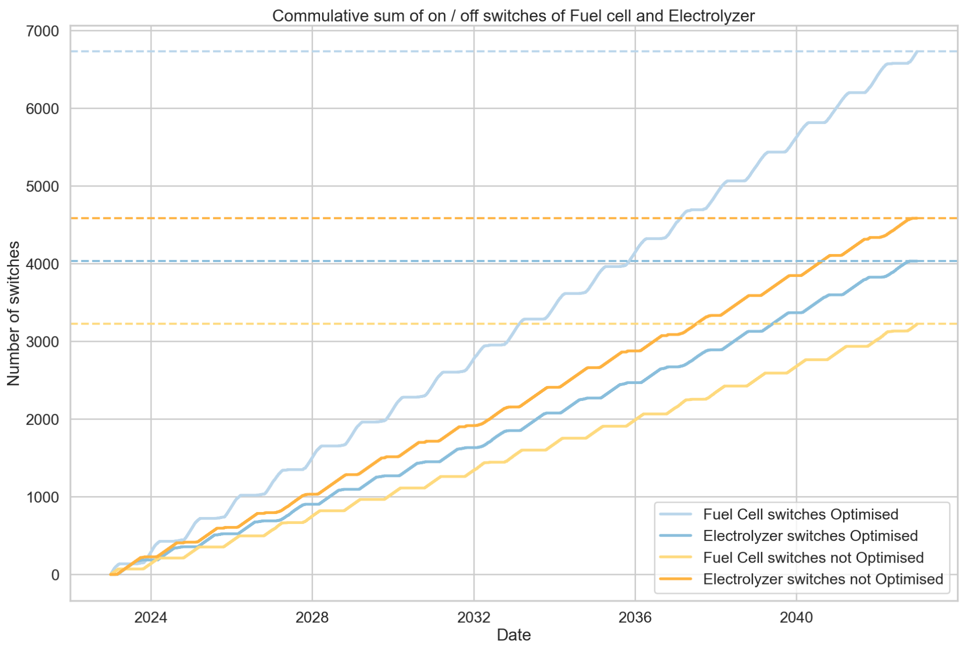



**Figure 5.10:** Comparison of On-Off cycles of both fuel cell and electrolyzer, between the Genetic Algorithm, and the optimization using PYOMO

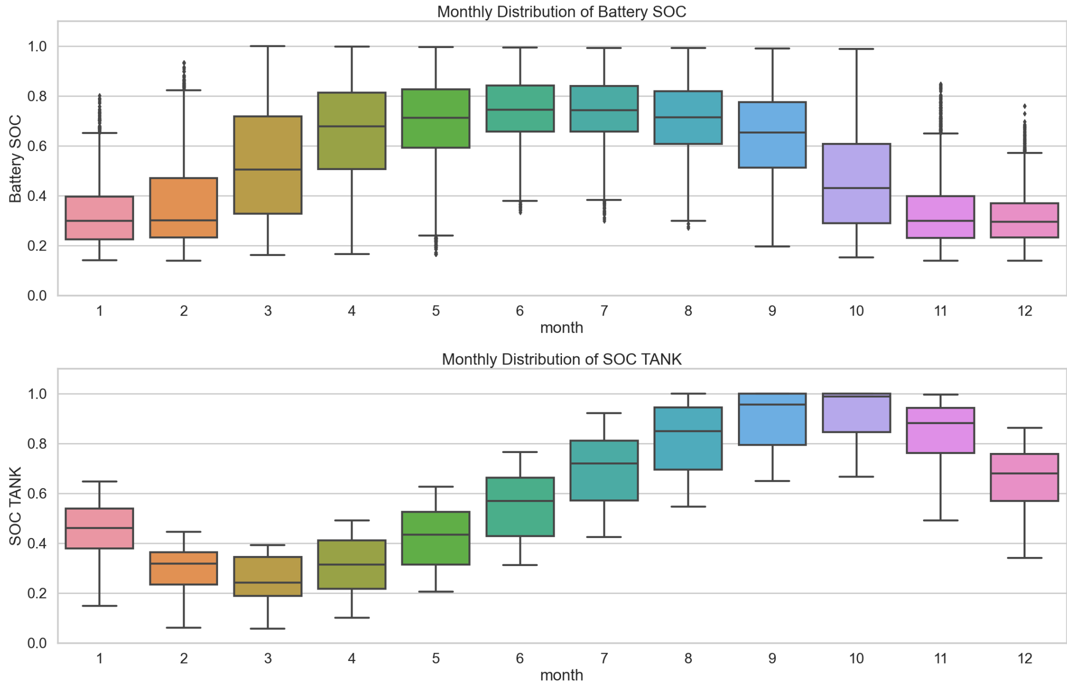



**Figure 5.11:** Average state of charge of the battery and hydrogen tank, over a 20 year span, using Genetic Algorithm

for the SOC of the hydrogen tank, it has very similar results for both cases, with PYOMO only showing slightly smaller fluctuation in the values.

The final analysis to be made on the performance of the PYOMO is on the effect of degradation on the lifespan and capacity of components. This is observed in figure 5.13.

In this graph, it can be seen that there is a progressive and gradual reduction in the maximum capacity of the components, with the electrolyzer and fuel cell losing 500W and 300 W of maximum power respectively. However, the 20 years of degradation using the lifespan optimization is not enough for the two components to reach below the 20% degradation necessary for their replacement. This is not the case for the battery however, which reaches the 40% degradation necessary for its replacement around the 19 year mark, as can be seen by the battery capacity resetting to its initial value.

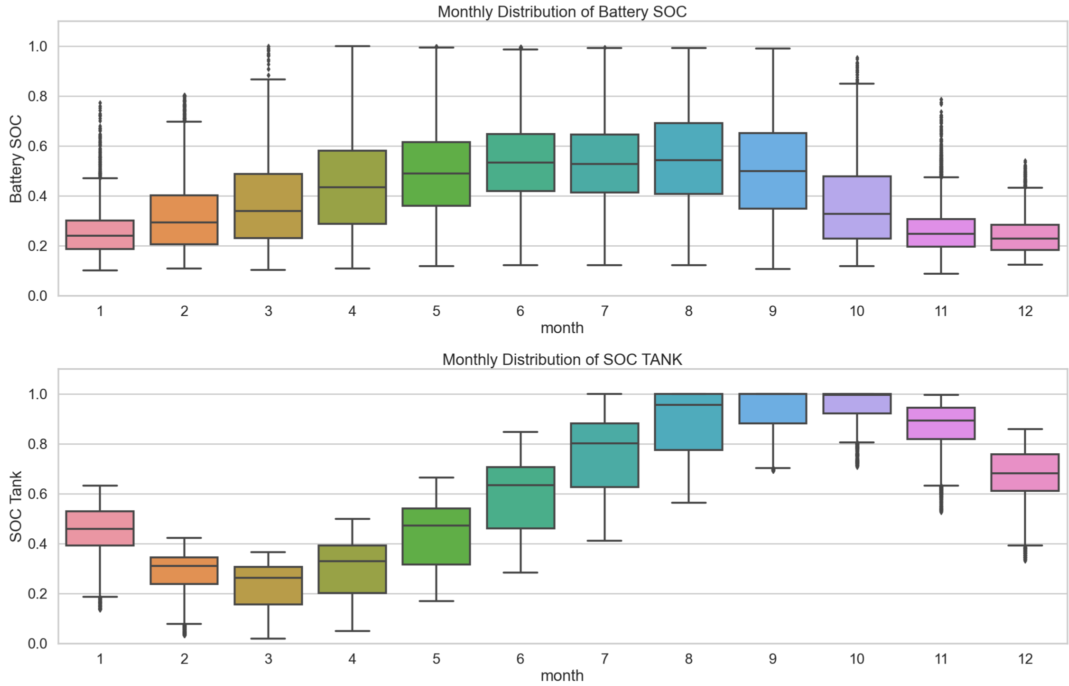



**Figure 5.12:** Average state of charge of the battery and hydrogen tank, over a 20 year span, using PYOMO optimization

## 5.3 Discussion

With each piece of information, an analysis of the results in their complete capacity can be made, encompassing all the information obtained. The analysis to be made is mainly pertaining the performance indicators, as they are the ones who show the decisions the code made when optimizing the system.

The most unexpected result, and the one who goes against the fact that the PYOMO presents an optimization of the system, is the number of switches. However, there is an explanation to why the PYOMO decided to have so many On-Off cycles when compared to the Genetic Algorithm. This is due to both something by design, and something that can be improved in the future. The reason why, by design, the PYOMO optimization is introducing more switches, is because of the battery SOC having a cost to going below 20% and over 80%. This leads to the management system utilizing the electrolyzer more times in order to avoid the extra cost of surpassing the 80% mark, but doing it without running the electrolyzer on a constant manner. The same can be said about the fuel cell, where the system is turning on the fuel cell when necessary in order to avoid the battery going below 20% SOC, as mathematically this is economically the correct decision. However, this is where the future improvement can be accomplished. The reason why the


**Figure 5.13:** Graphical representation of the maximum capacity of components over time

system is deciding to increase the number of On-Off cycles over extra degradation of the battery by being below 20% or over 80% is because the cost that is being given to this extra battery degradation is higher than the cost of extra switches of the fuel cell and electrolyzer. However, this might not be the case, since even though the battery working in ranges below 20% and over 80% is prejudicial to its lifespan, it does not present degradation values that significantly higher than its regular usage within 20% and 80%. As such, more investigation would be necessary to evaluate the true cost of utilizing the battery at each level of charge, in order to more accurately evaluate whether the system should avoid these SOC's or not.

Besides this, the PYOMO optimization could be further optimized in order to allow for the chunks of time it uses for each optimization to be higher than the current value of 10 hours. Higher chunks of time could allow for better evaluation of future circumstances and maybe a continuous use of both electrolyzer or fuel cell, while avoiding the costly State of Charge over 80% or under 20%. The currently used chunks of time of only 10 hours, do not allow for the system optimize according to an entire day worth of irradiation and usage, and as such, the simple accomplishment of achieving optimization for bigger chunks of time could achieve much better results.

Besides this possible problem related to the switches, it is clear to see that the PYOMO does achieve its goal of optimizing the way in which the system works both in number of hours of fuel cell run time, the average battery State of Charge, that is within the values at which it has lower degradation and especially the wasted power. This last component has an attributed cost equal to the power of electricity in the cost function of the PYOMO, which leads it to attempt reducing it, something not done by the genetic algorithm.

# Chapter 6

# Conclusion and future study

In this chapter the concluding remarks based on the results will be discussed as well as comprehensive suggestions for further exploration using the foundations set be the present report. The software developed for this thesis will be open sourced in a GITHUB repository given below:
`https://github.com/Heraclitus2/Heraclitus2`

## 6.1 Conclusion

After a comprehensive and thorough study on the different power requirements and economic metrics for the test cases, the results were presented in the form of sizing, simulation and optimisation. Those were conducted for all the proposed stand alone hybrid storage microgrids, showcasing the flexibility of the developed software.

After the selection of the 4 cases and the energy requirement criteria, the sizing of the energy system was conducted. The basis for the sizing is that the load is followed in an islanded, off grid configuration. Meaning, complete autonomy all year without any external energy supply, an assumption somewhat realistic for a laundry facility in Kenya or a house in Europe, but out of grasp for a data centre. The load profiles of each of these locations and applications was successfully approximated, with hourly and monthly variations according to expected usage. These designed load profiles were taken, along with the irradiation data, and used as input in the developed Genetic Algorithm.

The Genetic Algorithm managed to provide the sizing results of the microgrid in question. This was possible by utilizing the solar power received by the application and its demand, while also taking into account efficiencies of components and their power limitations. The system was then sized by minimizing the cost, while following a battery prioritization strategy when it comes to its energy management. Utilizing the calculation methods explained in section 3.4.1, the software achieved convergence and provided valid numbers for the sizing of the full hybrid microgrid.

The sizing of the system was the basis for the economic results. Upon comparison of the economic metrics, the effects of economies of scale and decrease of prices with increasing size of components are evident. The data centre exhibited the lowest payback period when compared to the cost of a diesel system. Also the data centre microgrid has the lowest LCOE of all the cases. The residential application exhibited the worst performance in regards to LCOE and payback period. This can be attributed to the presence of EV charging in the load curve which had a peak outside of the solar input time frame. This made it so that the hydrogen system had to be oversized in order to supply this application.

No configuration was able to closely compete with the grid. However,this poses an unfair comparison because there are aspects that justify the applicability of a hybrid storage microgrid which can not be quantified in economic metrics. One of the advantages of this is the independence from the energy grid. This is extremely valuable when considering sudden increases in price, power outages due to natural disasters or human error, and government enforced limitations on the grid. Besides this, there's also an environmentally friendly advantage to moving towards an hybrid microgrid, as this represents zero $CO_2$ emissions.

The values obtained through the Genetic Algorithm were then supplied to the PYOMO optimization, where 20 years of energy management were simulated. This software successfully followed the demand, while minimizing degradation. This allowed for an increase in the lifespan in the components, reducing re aquisition costs, and allowing for a longer longevity of the microgrid.

## 6.2 Future study

Expansion and future outlook of the study conducted for this report could include further development in each of the different levels of the report. Those levels being, general software development, economics, sizing of components and energy management system, as well as the major assumptions made in the beginning of the report. One point of interest would be the exploration of the energy modeling and economic evaluation of the system using the same conditions and assumptions, but the microgrid not having a 100 % islanded configuration and being off grid, but rather be grid connected. Research objectives could be the introduction of electricity pricing and carbon credits in the EMS and energy trading algorithms. Another addition to the energy system could also be the inclusion of wind power as a primary energy source or supplementary to PV power.

In the general software development domain, the expansion of the study could include more online APIS other than just solar irradiance data. Those could include land usage cost or the aforementioned wind speed data. Also the calculation of the available solar area for each project, a value already present in the energy system constraints. In the interface domain, the output of the code, like graphs and simulations also may be presented after the selection and value input as those where outlined in section 3.2.3.

Further development and study when it comes to the economics could include the projected decrease in prices of the system components. Prices for PVs and batteries are already outlined in section 3.2.7 for the year 2030, so comparison of the system CAPEX and all other economic metrics could be studied for future investments. Moreover, a further addition based on this premise could be the inclusion of the price decrease projections in the cost of replacement of components. Another economic comparison that could take place could be the one between different electrolyzer and fuel cell technologies, since the prices used for the present study were only accounting for PEM fuel cells, and PEM and AEM electrolyzers. Finally, no variation in electricity or diesel price was assumed on this report as well as no effect of inflation on the economic metrics.

The sizing of components in the study conducted in this report did not include the wasted power and it was concluded that it made the system oversized, as it produced and stored more energy than the system could use in some cases, such as Kenya. It is difficult to decide which solution is better for this kind of system since they are not designed to maximize producion, but to provide electricity in a reliable manner. A weighted value for the Power wasted could be included in the fitness function decision using the current price of electricity produced in the country can be used for this purpose. However, some people might argue that for such systems price is the only determining factor.

Optimally, the PYOMO optimization would be linked to the GA searching method, providing a doubled-optimized solution. The GA can also be further optimised by testing different mutation rates, crossover options and ranking procedures.

The platform and test cases assumed in this project could be used in order to validate other energy management system strategies. Those could include hydrogen system priority in order to minimize the wasted power or battery priority in order to minimise the cost.

Also, as mentioned in section 5.3, there are improvements to be made to the PYOMO optimization. Firstly, work should be done to progress towards being able to use the Linear Programming solver for larger "chunks" of time. To achieve this, it would be possible to allow the solver to use a higher portion of the user's computational power to process the calculations necessary. Besides this, there could also be introduced a tolerance, to allow the solver to reach values close enough to the final solution, without requiring the extra computational power to achieve the exact solution. The final solution possible to optimize the time required to solve the problem on higher "chunks" of time is to better define the constraints, so that the function isn't over constrained. This optimization that allows for larger time frames to be simulated is essential to allow for the proper sizing of the microgrid to be simulated. This is due to the fact that the proper sizing would reduce size of fuel cell, allowing for a more continuous working state of this component. However, this would require the energy management system to evaluate that it requires fuel cell power to help compensate the battery discharge in advance, as the fuel cell would not be able to follow the demand on its own.

Besides the previous problem with the optimization, there is also a possibility of further improving the costs given to each property in the cost function. The main costs that need future reviewing and study are the cost of charging/discharging the battery at extra degradation SOC's and the cost of wasted power. The arbitrary cost given to using the battery at below 20% and above 80% state of charge was a quarter of a lifetime cycle. However, this value was merely given to attribute a cost to this state so that the PYOMO solver would avoid these extra degradation states. A more accurate cost wasn't found in the research made, with a proper weighing of the cost of the extra degradation when compared with appropriate battery usage needing to be calculated. With this improvement to the code being made, a prioritization of charging the battery over using the electrolyzer could lead to a reduction in the number of On-Off cycles this component is subjected to. At the same time, a sensitivity analysis of the cost of waste power could lead to finding the ideal value to optimize the system. Despite this, the results of the energy and hydrogen flows indicate that the waste power is, for the most part, properly managed, and derives from over sized components.

# Bibliography


[1] PowerCell Sweden AB. *PG System*. Available at: `https://press.powercell.se/en/pg-system-5`. 2023.

[2] International Renewable Energy Agency. *Electricity storage and renewables: Costs and markets to 2030*. Tech. rep. International Renewable Energy Agency, 2017.

[3] Anna Alberini et al. "Hot weather and residential hourly electricity demand in Italy". In: (2023).

[4] Frits Møller Andersena et al. *Households' hourly electricity consumption and peak demand in Denmark*. 2017.

[5] Jasbir Arora. *Introduction to Optimum Design*. 4th ed. Academic Press, 2016. ISBN: 9780128008065.

[6] Chad Augustine and Nate Blair. *Storage Futures Study: Storage Technology Modeling Input Data Report*. Tech. rep. National Renewable Energy Laboratory, 2020.

[7] F. Barbir. *PEM Fuel Cells: Theory and Practice*. Elsevier, 2012.

[8] Battery University. *BU-808: How to Prolong Lithium-based Batteries*. `https://batteryuniversity.com/article/bu-808-how-to-prolong-lithium-based-batteries`. No publication year provided.

[9] Jef Beerten, Jorn Willems, and Johan Driesen. "Energy storage in multi-energy carrier microgrids". In: *IET Renewable Power Generation* 8.1 (2014), pp. 43–50.

[10] Subhes C. Bhattacharyya. *Energy Economics: Concepts, Issues, Markets, and Governance*. Springer, 2014.

[11] Jorn Brauns and Thomas Turek. *Alkaline Water Electrolysis Powered by Renewable Energy: A review*. 2020.

[12] Richard A Brealey, Stewart C Myers, and Franklin Allen. *Principles of Corporate Finance*. McGraw-Hill, 2003.

[13] G. Bruni, S. Cordiner, and V. Mulone. "Domestic distributed power generation: Effect of sizing and energy management strategy on the environmental efficiency of a photovoltaic-battery-fuel cell system". In: *International Journal of Hydrogen Energy* 39.19 (2014), pp. 10152–10164.

[14] M. Carmo et al. "A comprehensive review on PEM water electrolysis". In: *International Journal of Hydrogen Energy* 38.12 (2013), pp. 4901–4934.

[15] Manuel Castañeda et al. "Sizing optimization, dynamic modeling and energy management strategies of a stand-alone PV/hydrogen/battery-based hybrid system". In: *International Journal of Hydrogen Energy* 38.10 (2013), pp. 3830–3845.

[16] European Commission Joint Research Centre. *Photovoltaic Geographical Information System (PVGIS)*. 2001–2021. URL: https://re.jrc.ec.europa.eu/pvg_tools/en/tools.html.

[17] Christos Chalkias, Antigoni Faka, and Kleomenis Kalogeropoulos. "Assessment of the Direct Sun-Light on Rural Road Network through Solar Radiation Analysis Using GIS". In: (2013). Copyright © 2013 Christos Chalkias et al. This is an open access article distributed under the Creative Commons Attribution License. URL: https://www.researchgate.net/publication/239847020_Assessment_of_the_Direct_Sun-Light_on_Rural_Road_Network_through_Solar_Radiation_Analysis_Using_GIS.

[18] Dolphin Charger. *Lithium, GEL or AGM battery: which type of battery for which use?* 2021. URL: https://www.dolphin-charger.com/news/lithium-gel-agm-battery.

[19] Xiaogang Chen, Song Gao, and Chongqing Kang. "Genetic algorithm based approach for optimal planning of distributed energy resources". In: *Applied Energy* 184 (2016), pp. 1140–1151.

[20] Greig Chisholm, Tingting Zhao, and Leroy Cronin. "24 - Hydrogen from water electrolysis". In: *Storing Energy (Second Edition)*. Ed. by Trevor M. Letcher. Second Edition. Elsevier, 2022, pp. 559–591. ISBN: 978-0-12-824510-1. DOI: https://doi.org/10.1016/B978-0-12-824510-1.00015-5. URL: https://www.sciencedirect.com/science/article/pii/B9780128245101000155.

[21] Viviana Cigolotti, Matteo Genovese, and Petronilla Fragiacomo. "Comprehensive Review on Fuel Cell Technology for Stationary Applications as Sustainable and Efficient Poly-Generation Energy Systems". In: ().

[22] McKinsey Company. *Hydrogen Insights A perspective on hydrogen investment, market development and cost competitiveness*. 2021.

[23] Council of European Energy Regulators. *CEER Report on Power Losses*. 2017.

[24] Author or Creator's Name. *Title of the Video*. 2023. URL: https://www.ogv.energy/play/462.

[25] SBI for the Danish Energy Agency. *Cost-optimal Levels of Minimum Energy Performance Requirements in the Danish Building Regulations*. Accessed: 16th May 2023. 2018. URL: https://ens.dk/sites/ens.dk/files/Energibesparelser/bilag_3a_cost-optimal_levels_of_minimum_energy_performance_requirements_in_the_danish_building_regulations_-_sbi_2018.pdf.

[26] Datacenter Dynamics. *Digital Realty's Lamda Hellix to Build Data Center in Crete*. https://www.datacenterdynamics.com/en/news/digital-realtys-lamda-hellix-to-build-data-center-in-crete/. Accessed: May 19, 2023.

[27] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[28] Nikhil Deshmukh and Bala Natarajan Mohan. *Microgrids: Architectures and Control*. John Wiley & Sons, 2014.

[29] Destatis - Federal Statistical Office. *Electricity Consumption by Households*. https://www.destatis.de/EN/Themes/Society-Environment/Environment/Material-Energy-Flows/Tables/electricity-consumption-households.html. Accessed: May 19, 2023.

[30] *Diesel Generator*. Accessed: 2023-05-16. 2023. URL: https://www.alibaba.com/showroom/diesel-generator.html.

[31] DOE. *Hydrogen Shot*. 2021. URL: https://www.energy.gov/eere/fuelcells/hydrogen-shot.

[32] Data Center Dynamics. *Digital Realty's Lamda Hellix to build data center in Crete*. 2023. URL: https://www.datacenterdynamics.com/en/news/digital-realtys-lamda-hellix-to-build-data-center-in-crete/.

[33] Federal Ministry for Economic Affairs and Energy Public Relations Division. *The National Hydrogen Strategy*. 2020.

[34] eia.gov. *Photovoltaics and electricity*. 2022. URL: https://www.eia.gov/energyexplained/solar/photovoltaics-and-electricity.php.

[35] Ahmed M. Elberry et al. "Large-scale compressed hydrogen storage as part of renewable electricity storage systems". In: (2021).

[36] *Electricity Map*. Accessed: 2023-05-16. 2023. URL: https://app.electricitymaps.com/zone.

[37] Department of Energy. *How Microgrids Work*. 2014. URL: https://www.energy.gov/articles/how-microgrids-work.

[38] HOMER Energy. *HOMER Pro*. Accessed: 2023. URL: https://www.homerenergy.com/products/pro/index.html.

[39] Office of Energy Efficiency and Renewable Energy. *Hydrogen Storage*. 2022. URL: https://www.energy.gov/eere/fuelcells/hydrogen-storage.

[40] Europe Solar Store. *LG NeON 2 Bifacial 430-435-440W N2T-E6 Datasheet*. https://www.europe-solarstore.com/download/lg/LG_NeON_2_Bicacial_430-435-440W_N2T-E6_Datasheet.pdf. Accessed: May 19, 2023.

[41] European Central Bank. *Economic Bulletin: Focus on the Economic Outlook*. 2022. URL: https://www.ecb.europa.eu/pub/economic-bulletin/focus/2022/html/ecb.ebbox202204_01~68ef3c3dc6.en.html.

[42] Ander Fernandez. *Genetic Algorithm in Python*. URL: https://anderfernandez.com/en/blog/genetic-algorithm-in-python/.

[43] V. Fierro G. Sdanghi G. Maranzana A. Celzard. *Review of the current technologies and performances of hydrogen compression for stationary and automotive applications*. 2021.

[44] Felipe Ignacio Gallardo et al. "A Techno-Economic Analysis of solar hydrogen production by electrolysis in the north of Chile and the case of exportation from Atacama Desert to Japan". In: (2020).

[45] Pablo García et al. "Optimal energy management system for stand-alone wind turbine/photovoltaic/hydrogen/battery hybrid system with supervisory control based on fuzzy logic". In: ().

[46] Pablo García-Triviño et al. "Long-term optimization based on PSO of a grid-connected renewable energy/battery/hydrogen hybrid system". In: ().

[47] Pablo García-Triviño et al. "Optimized operation combining costs, efficiency and lifetime of a hybrid renewable energy system with energy storage by battery and hydrogen in grid-connected applications". In: *International Journal of Hydrogen Energy* 41.48 (2016), pp. 23132–23144. ISSN: 0360-3199. DOI: https://doi.org/10.1016/j.ijhydene.2016.09.140. URL: https://www.sciencedirect.com/science/article/pii/S0360319915319340.

[48] M. Gargiulo et al. "Analysis of thermal behavior of photovoltaic panels under different operating conditions". In: *Renewable Energy* 141 (2019), pp. 491–500. DOI: 10.1016/j.renene.2019.03.025.

[49] Irene Gatto et al. "Optimal operating conditions evaluation of an anion-exchange-membrane electrolyzer based on FUMASEP® FAA3-50 membrane". In: *International Journal of Hydrogen Energy* (2022). ISSN: 0360-3199. DOI: https://doi.org/10.1016/j.ijhydene.2022.04.176. URL: https://www.sciencedirect.com/science/article/pii/S0360319922017487.

[50] GENCELL. *COMPARING FUEL CELL TECHNOLOGIES*. 2022. URL: https://www.gencellenergy.com/news/comparing-fuel-cell-technologies/.

[51] Ghatikar et al. "Demand Response and Open Automated Demand Response Opportunities for Data Centers". In: *California Energy Commission. Publication number: CEC-500-2012-002* (2010).

[52] Global Islands Network. *The Maldives: a sustainable energy future*. 2008. URL: http://www.globalislands.net/userfiles/Maldives4.pdf.

[53] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[54] William E. Hart et al. *Pyomo-Optimization Modeling in Python*. Vol. 67. Springer, 2017.

[55] Nikos Hatziargyriou. *Microgrids: Architectures and Control*. John Wiley & Sons, 2014.

[56] Refrigerating American Society of Heating and Air-Conditioning Engineers. *ASHRAE Handbook: Fundamentals*. American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2009.

[57] I.A. Hassan Haitham S. Ramadan Mohamed A. Saleh Daniel Hissel. *Hydrogen storage technologies for stationary and mobile applications: Review, analysis and perspectives*. 2021.

[58] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.

[59] William F. Holmgren, Clifford W. Hansen, and Mark A. Mikofski. "pvlib python: a python package for modeling solar energy systems". In: *Journal of Open Source Software* 3.29 (2018), p. 884. DOI: 10.21105/joss.00884. URL: https://doi.org/10.21105/joss.00884.

[60] Hyatt. *Sustainability Management Plan*. 2021. URL: https://www.hyatt.com/content/dam/hotel/propertysites/assets/park/mldph/documents/en_us/home/MLDPHSustainabilityManagementPlan.pdf.

[61] Hydrogen and Fuel Cell Technologies Office. *Hydrogen Production: Electrolysis*. 2022. URL: https://www.energy.gov/eere/fuelcells/hydrogen-production-electrolysis.

[62] *Hydrogen Electrolyzers*. 2023. URL: https://hyfindr.com/?post_type=product&s=enapter (visited on 05/16/2023).

[63] Hydrogen Europe. *Strategic Research and Innovation Agenda*. Tech. rep. 2020, pp. 119–122.

[64] *Hydrogen Storage*. 2023. URL: https://wiki.opensourceecology.org/wiki/Hydrogen_Storage (visited on 05/16/2023).

[65] *Hydrogen Storage*. 2023. URL: https://www.energy.gov/eere/fuelcells/hydrogen-storage (visited on 05/16/2023).

[66] IEA. *The future of hydrogen*. 2019.

[67] Cummins Inc. *ELECTROLYZERS 101: WHAT THEY ARE, HOW THEY WORK AND WHERE THEY FIT IN A GREEN ECONOMY*. 2020. URL: https://www.cummins.com/news/2020/11/16/electrolyzers-101-what-they-are-how-they-work-and-where-they-fit-green-economy.

[68] Battelle Memorial Institute. *Manufacturing Cost Analysis of PEM Fuel Cell Systems for 5- and 10-kW Backup Power Applications*. Tech. rep. DOE Contract No. DE-EE0005250. Golden, CO: U.S. Department of Energy Golden Field Office, 2016.

[69] International Energy Agency. *Denmark*. No publication year provided. URL: https://www.iea.org/countries/denmark.

[70] International Renewable Energy Agency. *Energy Storage Costs*. No publication year provided. URL: https://www.irena.org/Energy-Transition/Technology/Energy-storage-costs.

[71] IRENA and CPI. *Global landscape of renewable energy finance, 2023*. Tech. rep. Abu Dhabi: International Renewable Energy Agency, 2023.

[72] Brian D. James et al. *Final Report: Hydrogen Storage System Cost Analysis*. Tech. rep. 2016.

[73] Stefan Loos Christian Immanuel Bernacker Thomas Weißgarber Lars Rontzsch Hamish Andrew Miller a Karel Bouzek Jaromir Hnat and Jochen Meier-Haackd. *Green hydrogen from anion exchange membrane water electrolysis: a review of recent developments in critical materials and operating conditions*. 2020. URL: https://https://pubs.rsc.org/en/content/articlepdf/2020/se/c9se01240k.

[74] Thomas Kluyver et al. "Jupyter Notebooks – a publishing format for reproducible computational workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. 2016, pp. 87 –90.

[75] J. Larminie and A. Dicks. *Fuel Cell Systems Explained*. John Wiley & Sons, 2003.

[76] LG Chem. *LG Chem RESU Datasheet*. https://www.europe-solarstore.com/download/lgchem/LG_Chem_RESU_datasheet.pdf. No publication year provided.

[77] *LG NeON 2 Bifacial 430-435-440W N2T-E6 Datasheet*. URL: https://www.europe-solarstore.com/download/lg/LG_NeON_2_Bicacial_430-435-440W_N2T-E6_Datasheet.pdf.

[78] Eduardo Lopez Gonzalez et al. "Energy evaluation of a solar hydrogen storage facility: Comparison with other electrical energy storage technologies". In: *International Journal of Hydrogen Energy* 45.57 (2020), pp. 31128–31142.

[79] Mahindra Teqo. *Impact of Temperature on Solar Panels Efficiency*. https://www.mahindrateqo.com/blog/impact-of-temperature-on-solar-panels-efficiency/. Accessed: May 19, 2023.

[80] Maldives Financial Review. *Maldives Tourism Outlook for 2021*. 2021. URL: https://mfr.mv/travel-and-tourism/maldives-tourism-outlook-for-2021.

[81] Ahmad Mayyas et al. *Manufacturing Cost Analysis for Proton Exchange Membrane Water Electrolyzers*. Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States), 2013.

[82] Wes McKinney et al. "Data structures for statistical computing in python". In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. 2010, pp. 51–56.

[83] Microgrid Knowledge. *Here's Where We Really Need Microgrids in the US*. No publication year provided. URL: https://www.microgridknowledge.com/google-news-feed/article/11436842/heres-where-we-really-need-microgrids-in-the-us.

[84] Andrea Monforti Ferrario et al. "A model-based parametric and optimal sizing of a battery/hydrogen storage of a real hybrid microgrid supplying a residential load: Towards island operation". In: *Applied Energy* 230 (2018), pp. 1234–1249.

[85] Andrea Monforti Ferrario et al. "Hydrogen vs. Battery in the Long-term Operation. A Comparative Between Energy Management Strategies for Hybrid Renewable Microgrids". In: *Energies* 11.10 (2018), p. 2575.

[86] Author's name. *France to order air-conditioned shops to keep doors shut in energy-saving drive*. 2022. URL: https://www.theguardian.com/world/2022/jul/25/france-air-conditioned-shops-doors-shut-energy-saving.

[87] National Renewable Energy Laboratory. *2022 ATB Commercial Battery Storage*. Accessed: 2023-05-16. 2022. URL: https://atb.nrel.gov/electricity/2022/commercial_battery_storage.

[88] National Renewable Energy Laboratory. *2022 ATB Commercial PV*. Accessed: 2023-05-16. 2022. URL: https://atb.nrel.gov/electricity/2022/commercial_pv.

[89] National Renewable Energy Laboratory. *2022 ATB Residential Battery Storage*. Accessed: 2023-05-16. 2022. URL: https://atb.nrel.gov/electricity/2022/residential_battery_storage.

[90] National Renewable Energy Laboratory. *2022 ATB Residential PV*. Accessed: 2023-05-16. 2022. URL: https://atb.nrel.gov/electricity/2022/residential_pv.

[91] National Renewable Energy Laboratory. *2022 ATB Utility-Scale Battery Storage*. Accessed: 2023-05-16. 2022. URL: https://atb.nrel.gov/electricity/2022/utility_battery_storage.

[92] National Renewable Energy Laboratory. *2022 ATB Utility-Scale PV*. Accessed: 2023-05-16. 2022. URL: https://atb.nrel.gov/electricity/2022/utility-scale_pv.

[93] Balkan Green Energy News. *PPC Renewables picks Terna to build biggest solar power plant in Greece*. 2023. URL: https://balkangreenenergynews.com/ppc-renewables-picks-terna-to-build-biggest-solar-power-plant-in-greece/#:~:text=The\%20largest\%20photovoltaic\%20plant\%20in,facility\%20is\%20located\%20in\%20Kozani.

[94] Ubong Eduok Omar Faye Jerzy Szpunar. *A critical review on the current technologies for the generation, storage, and transportation of hydrogen*. 2022.

[95] OurWorldInData.org. *The price decline of electricity from renewable sources*. 2020. URL: https://ourworldindata.org/cheap-renewables-growth.

[96] Garcia Pablo et al. "Optimal energy management system for standalone wind turbine/photovoltaic/hydrogen/battery hybrid system with supervisory control based on fuzzy logic". In: (2013).

[97] ITM Power. *Increase in Hydrogen Production Rate*. https://www.itm-power.com/images/MediaCentre/ITM_Power_Technical_Highlight_Hydrogen_Production_Rate.pdf. Published: August 2017.

[98] Hydrogen House Project. *Hydrogen House Project*. Accessed: 2023. URL: https://hydrogenhouseproject.org/index.html.

[99] K. Promislow et al. "Computational modelling of polymer electrolyte membrane (PEM) fuel cells: Challenges and opportunities". In: *International Journal of Hydrogen Energy* (2007). URL: https://www.researchgate.net/publication/222823639_Computational_modelling_of_polymer_electrolyte_membrane_PEM_fuel_cells_Challenges_and_opportunities.

[100] Rentech Energy Solutions. *Lithium Batteries Limited Product Warranties*. https://rentech.co.za/wp-content/uploads/2022/03/Lithium-Batteries-Limited-Product-Warranties-v1.3.pdf. No publication year provided.

[101] Oliver Schmidt et al. "Future cost and performance of water electrolysis: An expert elicitation study". In: *International Journal of Hydrogen Energy* 42.52 (2017), pp. 30470–30492. ISSN: 0360-3199. DOI: https://doi.org/10.1016/j.ijhydene.2017.10.045. URL: https://www.sciencedirect.com/science/article/pii/S0360319917339435.

[102] Scientific American. *Renewable Energy Intermittency Explained: Challenges, Solutions, and Opportunities*. No publication year provided. URL: https://blogs.scientificamerican.com/plugged-in/renewable-energy-intermittency-explained-challenges-solutions-and-opportunities/.

[103] Jae K. Shim and Joel G. Siegel. *Budgeting Basics and Beyond*. Wiley, 2000.

[104] Maria Skyllas-Kazacos et al. "Progress in flow battery research and development". In: *Journal of The Electrochemical Society* 158.8 (2011), R55–R79.

[105] solarpowerworld.com. *What are solar panels made of?* 2013. URL: https://www.solarpowerworldonline.com/2013/05/what-are-solar-panels-made-of/.

[106] Grigorii L. Soloveichik. "Flow batteries: current status and trends". In: *Chemical Reviews* 115.20 (2015), pp. 11533–11558.

[107] *Steelhead Composites*. 2023. URL: https://steelheadcomposites.com (visited on 05/16/2023).

[108] Bengt Sundén. "Chapter 8 - Fuel cell types - overview". In: *Hydrogen, Batteries and Fuel Cells*. Ed. by Bengt Sundén. Academic Press, 2019, pp. 123–144. ISBN: 978-0-12-816950-6. DOI: https://doi.org/10.1016/B978-0-12-816950-6.00008-7. URL: https://www.sciencedirect.com/science/article/pii/B9780128169506000087.

[109] Swimsol. *Park Hyatt Roof Solar*. https://swimsol.com/solar-projects/park-hyatt-roofsolar-swimsol/. Accessed: May 19, 2023.

[110] Power Technology. *Power Plant Profile: Harre Solar Farm, Denmark*. 2023. URL: https://www.power-technology.com/marketdata/power-plant-profile-harre-solar-farm-denmark/#:~:text=Harre\%20Solar\%20Farm\%20is\%20a,the\%20project\%20is\%20currently\%20active.

[111] Texas Tribune. *Texas Legislature passes energy bills aimed at boosting natural gas exports*. No publication year provided. URL: https://www.texastribune.org/2022/07/05/texas-energy-bills-natural-gas-export/.

[112] Denis Thomas. "COST REDUCTION POTENTIAL FOR ELECTROLYSER TECHNOLOGY". In: 2018.

[113] Juan P. Torreglosa et al. "Control based on techno-economic optimization of renewable hybrid energy system for stand-alone applications". In: *Expert Systems with Applications* 51 (2016), pp. 59–75.

[114] Juan Pablo Torreglosa et al. "Hierarchical energy management system for stand-alone hybrid system based on generation costs and cascade control". In: *Renewable Energy* 112 (2017), pp. 163–174.

[115] University of Zaragoza. *IHOGA - Institute of Hydrogeology and Environmental Geology*. Accessed: 2023. URL: https://ihoga.unizar.es/.

[116] U.S. Department of Energy. *Hydrogen Storage*. No publication year provided. URL: https://www.energy.gov/eere/fuelcells/hydrogen-storage#text=Hydrogen.

[117] L. Valverde et al. "Definition, analysis and experimental investigation of operation modes in hydrogen-renewable-based power plants incorporating hybrid energy storage". In: (2016).

[118] L. Valverde et al. "Definition, analysis and experimental investigation of operation modes in hydrogen-renewable-based power plants incorporating hybrid energy storage". In: *International Journal of Hydrogen Energy* 44.3 (2019), pp. 1618–1633.

[119] Jake VanderPlas. *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media, Inc., 2016.

[120] F.J. Vivas et al. "A review of energy management strategies for renewable hybrid energy systems with hydrogen backup". In: *TEP-192, Departamento de Ingeniería Electrónica, de Sistemas Informáticos y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Huelva, Spain* (2023).

[121] Annette Werth, Kenji Tanaka, and Others. "Evaluation of centralized and distributed microgrid topologies and comparison to Open Energy Systems (OES)". In: (2015). DOI: 10.1109/EEEIC.2015.7165211.

[122] Wikipedia. *Foulum Data Center*. https://en.wikipedia.org/wiki/Foulum_Data_Center. Accessed: May 19, 2023.

[123] H. Paul Williams. *Model building in mathematical programming*. John Wiley & Sons, 2013.

[124] Weidong Xiao, Nathan Ozog, and W.G. Dunford. "Topology Study of Photovoltaic Interface for Maximum Power Point Tracking". In: *IEEE Transactions on Industrial Electronics* 54 (3 2007), pp. 1696–1704. DOI: 10.1109/TIE.2007.894732. URL: https://ieeexplore.ieee.org/document/4280267.

[125] Aya Zahran. *Total cost of ownership: a key concept in strategic cost management decisions*. Center for Advanced Purchasing Studies, 1998.

[126] Xiaolin Zhang et al. "Energy management and optimization of microgrids using genetic algorithms". In: *Energy Procedia* 103 (2016), pp. 227–232.

[127] B Zhou et al. "A novel approach for electric load curve holistic modelling and simulation". In: (2015). URL: https://www.researchgate.net/publication/280756503_A_novel_approach_for_electric_load_curve_holistic_modelling_and_simulation.

# Appendix A

# Appendix

## A.1 Software tools

```
In [1]:
import pvlib
import pandas as pd
from pvlib import clearsky, atmosphere, solarposition
from pvlib.location import Location
import folium
import matplotlib.pyplot as plt
import matplotlib
import datetime
import plotly.express as px
import numpy as np
import tkinter as tk
from tkinter import ttk
from pytz import timezone
import plotly.graph_objs as go
import math
import seaborn as sns
```

**Figure A.1:** Imported python packages used for the modeling of the system

### A.1.1 Interface

### A.1.2 Modeling

### A.1.3 Energy management system

### A.1.4 Economics

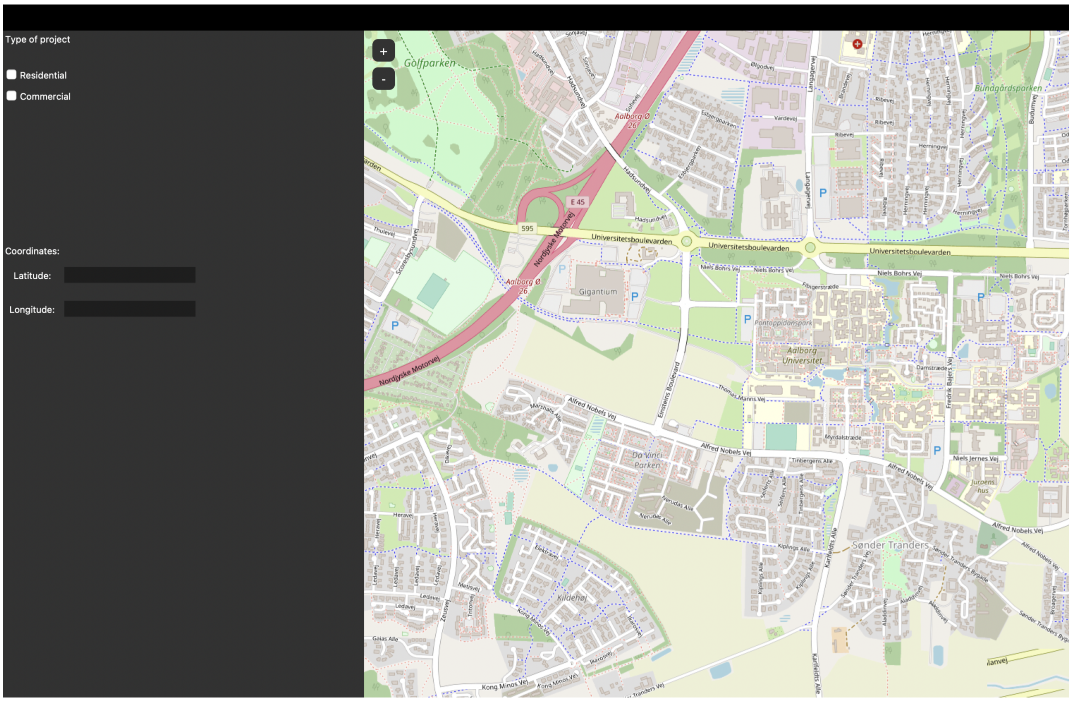



**Figure A.2:** Graphical User interface centered at Aalborg University east campus

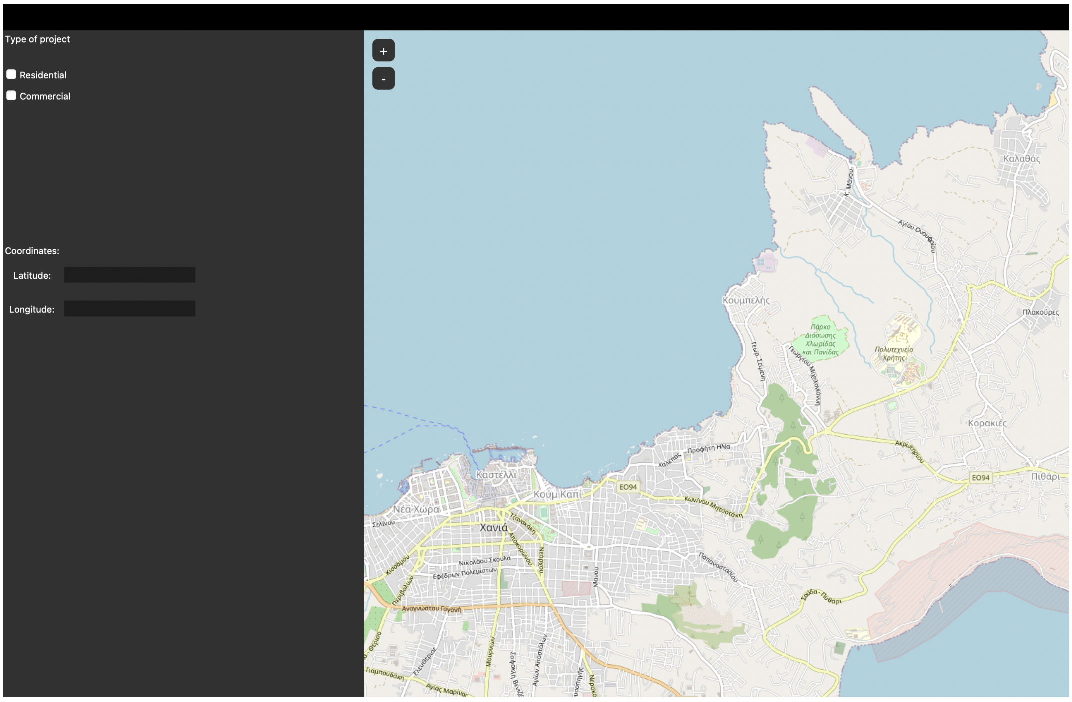



**Figure A.3:** Graphical User interface centered at Chania Greece

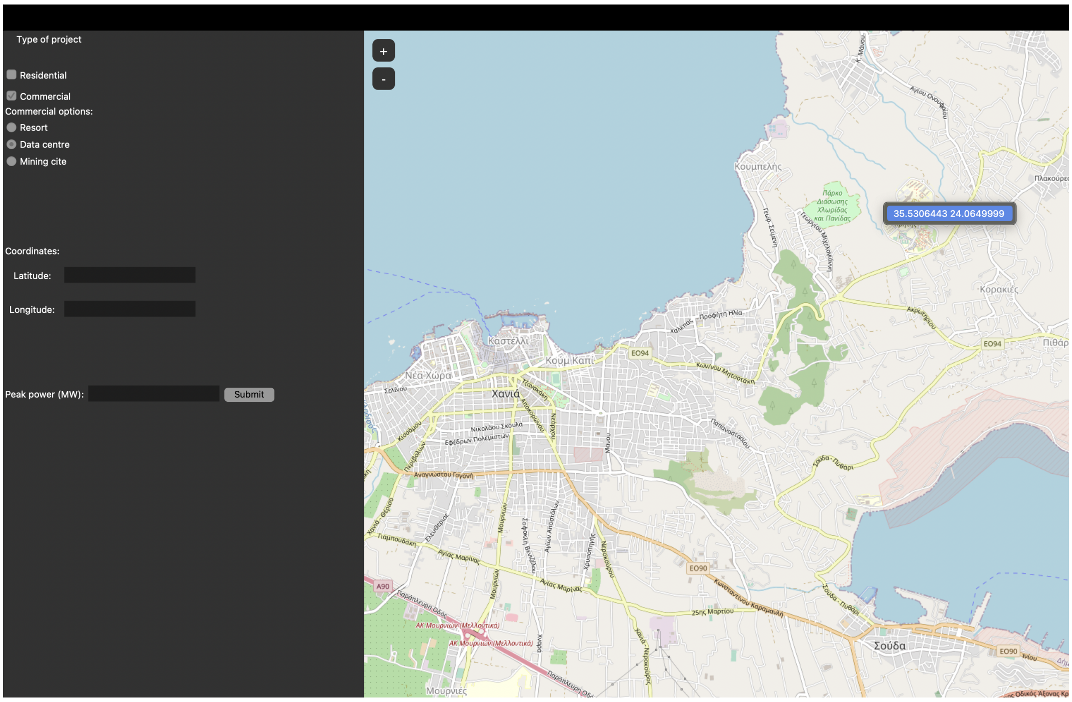



**Figure A.4:** Graphical User interface with expanded commercial option and selected coordinates

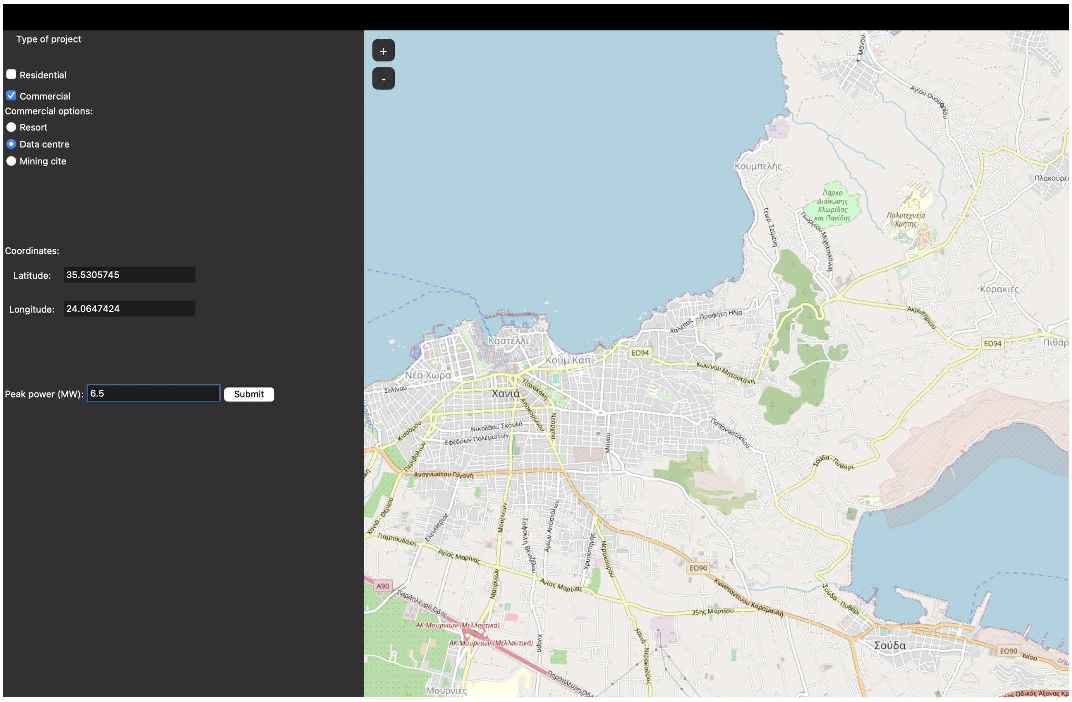



**Figure A.5:** Graphical User interface with expanded commercial option and selected coordinates for 6.5 MW data centre

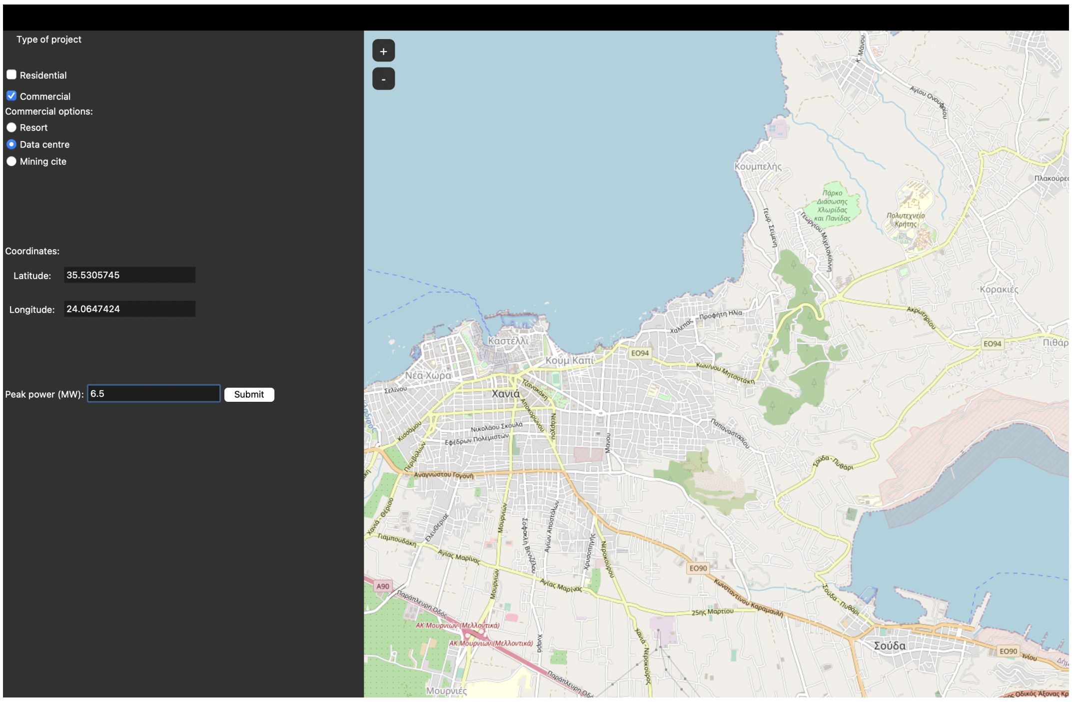



**Figure A.6:** Graphical User interface with expanded commercial option and selected coordinates for 6.5 MW data centre

In [5]:

```python
#Set location to Nairobi, Kenya & reorganize df
df0, meta, inputs = pvlib.iotools.get_pvgis_hourly(
    latitude = -1.29, longitude= 36.82, start=2010, end=2015) #Getting pvgis data from website
df0
```

Out[5]:

| time | poa_direct | poa_sky_diffuse | poa_ground_diffuse | solar_elevation | temp_air | wind_speed | Int |
|---|---|---|---|---|---|---|---|
| 2010-01-01 00:05:00+00:00 | 0.0 | 0.0 | 0.0 | 0.00 | 17.41 | 1.70 | 0 |
| 2010-01-01 01:05:00+00:00 | 0.0 | 0.0 | 0.0 | 0.00 | 16.86 | 1.56 | 0 |
| 2010-01-01 02:05:00+00:00 | 0.0 | 0.0 | 0.0 | 0.00 | 16.30 | 1.43 | 0 |
| 2010-01-01 03:05:00+00:00 | 0.0 | 0.0 | 0.0 | 0.00 | 15.75 | 1.30 | 0 |
| 2010-01-01 04:05:00+00:00 | 28.0 | 51.0 | 0.0 | 7.34 | 16.51 | 1.73 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 2015-12-31 19:05:00+00:00 | 0.0 | 0.0 | 0.0 | 0.00 | 19.21 | 2.40 | 0 |
| 2015-12-31 20:05:00+00:00 | 0.0 | 0.0 | 0.0 | 0.00 | 18.35 | 2.06 | 0 |
| 2015-12-31 21:05:00+00:00 | 0.0 | 0.0 | 0.0 | 0.00 | 17.48 | 1.72 | 0 |
| 2015-12-31 22:05:00+00:00 | 0.0 | 0.0 | 0.0 | 0.00 | 17.48 | 1.72 | 0 |
| 2015-12-31 23:05:00+00:00 | 0.0 | 0.0 | 0.0 | 0.00 | 17.48 | 1.72 | 0 |

52584 rows × 7 columns

**Figure A.7:** Call for solar irradiation data for Nairobi, Kenya using PVlib and PVGIS

```python
dfw = pd.read_csv('/Users/foivosmaniatis/Documents/Usefull CSVs/Book3.csv', delimiter=' ',header=1)

dfw[dfw.columns[-1]] = dfw[dfw.columns[-1]].shift(1)
dfw.iloc[0,len(dfw.columns)-1] = 'January'
dfw.iloc[2,len(dfw.columns)-1] = 'March'
dfw = dfw.set_index(dfw[dfw.columns[-1]])
del dfw[dfw.columns[-1]]
del dfw[dfw.columns[-1]]
#dfw = dfw.T
# dfw.rename(index={'24':'0'},inplace=True)
# dfw['time'] = dfw.index
# dfw['time'] = dfw['time'].astype(int)



#dates = pd.date_range(start='2023-01-01 00:00:00', end='2023-12-31 23:00:00', freq='H')
#df = pd.DataFrame({'Month': dates.month, 'Time of Day': dates.strftime('%H:%M')}, index=dates)
#df['time'] = df.index.hour

df_coefficients_long = pd.melt(dfw, var_name='month', value_name='coefficient')
df_coefficients_long['hour'] = df_coefficients_long.index % 24
df_coefficients_long

# Define the load profiles for each month
load_profiles = {
    1: dfw.iloc[0].to_numpy(),
    2: dfw.iloc[1].to_numpy(),
    3: dfw.iloc[2].to_numpy(),
    4: dfw.iloc[3].to_numpy(),
    5: dfw.iloc[4].to_numpy(),
    6: dfw.iloc[5].to_numpy(),
    7: dfw.iloc[6].to_numpy(),
    8: dfw.iloc[7].to_numpy(),
    9: dfw.iloc[8].to_numpy(),
    10: dfw.iloc[9].to_numpy(),
    11: dfw.iloc[10].to_numpy(),
    12: dfw.iloc[11].to_numpy()
}

# Loop over each month and fit the corresponding load profile to the data
for month in range(1, 13):
    # Get the load profile for this month
    load_profile = load_profiles[month]

    # Extract the consumption data for this month
    month_mask = df.index.month == month
    consumption_month = df.loc[month_mask, 'Electricity consumption (Wh)'].values

    # Repeat the load profile to cover the entire month
    load_profile_month = np.tile(load_profile, int(np.ceil(len(consumption_month) / 24)))
    load_profile_month = load_profile_month[:len(consumption_month)]

    # Scale the load profile to match the total energy consumption of the original data
    total_consumption_month = consumption_month.sum()
    total_profile_month = load_profile_month.sum()
    scaling_factor_month = total_consumption_month / total_profile_month
    load_profile_month_scaled = load_profile_month * scaling_factor_month

    # Add noise to the data
    noise_month = np.random.normal(0, 0.05, len(consumption_month))
    consumption_month_residential = load_profile_month_scaled + noise_month

    # Replace the consumption data for this month with the fitted data
    df.loc[month_mask, 'Electricity consumption (Wh)'] = consumption_month_residential


if global_values['electric_car']=='Yes':
    df['Electric car charging'] = 3000*1000/(365*24)
    # define the coefficients for electric car charging
    ev_coefs = [0.1, 0.1, 0.1, 0.1, 0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1, 0.3, 0.5, 0.7, 0.8, 0.9, 0.8,
    # normalize the coefficients so they add up to 1.0
    ev_coefs = np.array(ev_coefs)
    ev_coefs /= np.sum(ev_coefs)

    # set the desired total amount of energy to transform
    total_energy = 3000*1000/(365*24)  # replace with your desired total energy in kWh

    # multiply the normalized coefficients by the scaling factor to get the new coefficients
    ev_coefs = ev_coefs * (total_energy / 24)

    # create a pandas dataframe to store the coefficients
    df_ev_coefs = pd.DataFrame({'coeff': ev_coefs})
    df_ev_coefs = pd.concat([df_ev_coefs] * 7300, ignore_index=True)
    df_ev_coefs = df_ev_coefs.set_index(df.index)
    df = df.join(df_ev_coefs)
    df['Electric car charging'] = df['Electric car charging']*df['coeff']
    df=df.drop(columns=['coeff'])
    df['Total energy demand (Wh):'] = df['Electricity consumption (Wh)'] + df['Heating demand (Wh):'] + df['Co
    df_house = df
    %matplotlib notebook
```

**Figure A.8:** Generation of residential load curve for electricity consumption and EV charging

```python
def Consumption(data):
    data['Consumption'] = 200  #Baseload
    #fridge_number = input("How many refrigeration units are in the pod? ")
    fridge_number = 2
    base_consumption = (data['Consumption']+ fridge_number*200)
    data['Consumption'] = base_consumption
    #washing_machine_number = input("How many washing machines are in the pod? ")
    washing_machine_number = 2
    #Washing machine load curve fitting for the hours between 9 and 16
    list_int = [2000, 2200, 2300, 2400]
    # Define the time range
    start_time = datetime.time(9)
    end_time = datetime.time(14)
    # Create a boolean mask based on the time range
    mask = (data.index.time >= start_time) & (data.index.time < end_time)


    # Apply the sequence of values only to the defined time range
    data.loc[mask,'new_column'] = np.tile(list_int, len(df[mask])//len(list_int) + 1)[:len(data[mask])]
    data.loc[(data.index.time >= datetime.time(9)) & (data.index.time < datetime.time(20)),'Consumption'] = da

    # Use .where() method to fill the missing values with 2000 when the index is '2020-01-01 09:00:00+03:00'
    data['Consumption'] = data['Consumption'].where(data.index != '2020-01-01 09:00:00+03:00', 2000)

    # Now use .fillna() method to fill the missing values with the base_load in other cases
    data['Consumption'].fillna(base_consumption, inplace=True)


    del data['new_column']

    return(data)
```

**Figure A.9:** Generation of load curve for communal facility in Kenyan village

In [1]:

```python
import pandas as pd
import numpy as np

#peak power
p_peak = 6500

# define the coefficients for data center load
coefficients = [0.7, 0.7, 0.7, 0.7, 0.7, 0.7, 0.8, 0.9, 1.0, 1.0, 1.0, 1.0,
                1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.9, 0.8, 0.7, 0.7, 0.7, 0.7]

# create a pandas dataframe to store the coefficients
df_data_center = pd.DataFrame({'Power (kW)': np.array(coefficients) * p_peak})

# set the index to be hourly for a year
df_data_center.index = pd.date_range('2023-01-01', freq='H', periods=24)

# set power to peak between 9am and 5pm
df_data_center.loc[(df_data_center.index.hour >= 9) & (df_data_center.index.hour < 17), 'Power (kW)'] = p_peak

# add random noise between -5% to +5%
noise = np.random.uniform(-0.05, 0.05, size=len(df_data_center))
df_data_center['Power (kW)'] = df_data_center['Power (kW)'] * (1 + noise)

# multiply the dataframe by 365*20 to create a 20 year dataframe
df_data_center = pd.concat([df_data_center]*365, ignore_index=True)

# set the index to be yearly
df_data_center.index = pd.date_range('2023-01-01', freq='H', periods=len(df_data_center))

# reduce demand by 15% for July and August
df_data_center.loc[df_data_center.index.month.isin([7, 8]), 'Power (kW)'] *= 0.85

print(df_data_center['Power (kW)'])
```

```
2023-01-01 00:00:00    4630.793240
2023-01-01 01:00:00    4562.921830
2023-01-01 02:00:00    4436.192531
2023-01-01 03:00:00    4738.011220
2023-01-01 04:00:00    4635.994090
                          ...
2023-12-31 19:00:00    5032.434644
2023-12-31 20:00:00    4524.467108
2023-12-31 21:00:00    4385.684410
2023-12-31 22:00:00    4373.970985
2023-12-31 23:00:00    4503.360278
Freq: H, Name: Power (kW), Length: 8760, dtype: float64
```

**Figure A.10:** Generation of load curve for data centre

In [2]:

```python
import numpy as np
##FUEL CELL MODEL
#parametres
#This model takes common values for PEMFC, Area, number of cells can change to account for the amount of Power
R=8.3144598 #[kPa*L/mol/K] Universal gas constant
T=343.15 #[K] Inlet temperature
F=96485.33 #[s*A/mol] Faraday constant
i_n = 2   #internal current density & fuel crossover
R_0=30e-6 #%ohms]
i_0 = 0.067  #[ma/cm2]exchange current density
i_L0 = 900 #[mA/cm2]Limiting current density
A=0.05
B=0.06
T_0=283.15 #[K] Ambient temperature
P_0=1 #[bar] Ambient pressure
Cp_air=1.003 #[J/kg/K]
M_O2=32e-3 #kg/mol molar mass of O2
lamb = 2 #stoichiometric ratio
M_air=28.97e-3 #[kg/mol] Molar mass of air
M_water=18.016e-3 #[kg/mol] Molar mass of water
RH_in=60 #[%] Relative humidity
M_H2=2.02 #%[kg/mol]
LHV_H2=33.3 #%[kWh/kg]
HHV_H2= 39.4 #%[kWh/kg]
A_cell=260 #[cm2]
Power_FC = input("What is the power of the chosen FC in kW? ")
Power_FC = int(Power_FC)
N=8*Power_FC #Number of cells


i_range = np.arange(0,895)  #[mA/cm2]
log_x = np.vectorize(np.log)
V_0 = ( 1.229 - (i_range + i_n)*R_0 - A*log_x((i_range + i_n)/i_0) + B*log_x(1-((i_range + i_n)/i_L0)) ) #[V]
Power_graph = N*A_cell/1000000*i_range*V_0     #[kW]

#Only plot until max power point is achieved
max_power_position = np.argmax(Power_graph)
max_power = Power_graph[max_power_position]
Power_graph = Power_graph[:max_power_position]

i_range = i_range[:max_power_position]
H2_cons_curve=((i_range*A_cell*N)/(2*1000*F) + (i_0*500*A_cell/1000*N/(2*F)))*(2.02e-3)*(LHV_H2*3600) #%[kW of

eff_FC=Power_graph/H2_cons_curve;

def find_nearest(a, a0):
    "Element in nd array `a` closest to the scalar value `a0`"
    idx = np.abs(a - a0).argmin()
    return  idx
```

```
What is the power of the chosen FC in kW? 1
```

**Figure A.11:** Fuel cell efficiency

In [1]:

```
##Electrolyzer MODEL

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import math


i = np.zeros(16)
N_cells = 1 # "EL stack number of cells" 136 cells = 3MW ->
A = 1250.0 # "Effective area per cell (unit = "cm2")"
T = 70 + 273.15 # "Stack temperature (unit = "K")"
P_an = 1.013e5 # "Anode pressure (unit = "Pa")"
P_cat = 40e5 # "FC stack height (unit = "Pa")"
#D_w = 1.28e-10 # "Diffusive coefficient of water in the membrane (unit = "m2/s")"
T_std = 25 + 273.15 # "Standard temperature (unit = "K")"
#K_darcy = 1.58e-18 # "Membrane permeability to water (unit = "m2")"
rho_w = 1000 # "Density of water (unit = "kg/m3")"
I_el = 1000 #Intensity crossing the stack (example, we pass a vector to plot)
N_stacks = 3 #3 stacks of 1 MW each
i = (1000 / A) * I_el #Current density
P_electricity = 0.02 #euro/kWh
Price_EL = 918000/3 #Price set by IRENA

Cp_stack = 1100 # FC stack specific heat capacity (unit = "J/(kg.K)")
Antoine_N = 8.07131 - 1730.63 / (233.426 + T - 273.15) # Parameter for antoine eq. (P_sat_water) mmHg ºc
P_sat_w = 10 ** Antoine_N * 133.322 # "Saturated pressure of water [Pa]"
F = 96485 # "Faraday constant A*s/mol"
R = 8.3144 # "Universal Gas Constant
mu_w = 0.0011 # "Water viscosity"
lamb_da = 22 # Membrane water constent (22 for wet membrane)
delta_mem = 0.000194 # "Membrane thickness (unit = "m")

sigma_mem = (0.005139 * 22 - 0.00326) * math.exp(1268 * (1 / 303 - 1 / T)) # "Membrane conductivity (S/cm)";
P_O2 = P_an - P_sat_w # "Anode exchange current density";
P_H2 = P_cat - P_sat_w # "Cathode exchange current density";

#Exchange current parameters
Mm_an = 1e-3 # Anode catalyst loading [g/cm2]
Mm_cat = 0.3e-3 # Cathode catalyst loading [g/cm2]
rho_Pt = 21.45 #Platinum (cathode) density [g/cm3]
rho_Ir = 22.56 #Iridium (anode) density [g/cm3]
dM_an = 2.9e-7 #Cristallite diameter anode [nm]
dM_cat = 2.7e-7 #Crystallite diameter cathode [nm]
phi_1 = 0.75 #Fraction of metal catalyst in contact with the ionomer
Ea_cat = 4.3 #Activation energy cathode [kJ/mol]
Ea_an = 76 # Activation energy anode [kJ/mol]
i_0_std_an = 7.407e-12 #  [A/cm2]
i_0_std_cat = 1e-3 #  [A/cm2]

#Charge transfer coefficients
alfa_an = 1.2 # Anodic charge transfer coefficient
alfa_cat = 0.6 # Cathodic charge transfer coefficient

```

**Figure A.12:** Electrolyzer efficiency

```python

# Roughness factor
gamma_M_cat = phi_1*Mm_cat*6/(rho_Pt*dM_cat) #(ratio between electrochemically active area and the geometric a
gamma_M_an = phi_1*Mm_an*6/(rho_Ir*dM_an) # (ratio between electrochemically active area and the geometric are

#Anode and cathode exchange current density (T dependance)
i_0_an = gamma_M_an*i_0_std_an*math.exp(-Ea_an*1000/R*((1/T)-(1/T_std)))
i_0_cat = gamma_M_cat*i_0_std_cat*math.exp(-Ea_cat*1000/R*((1/T)-(1/T_std)))

#CALCULUS OF EO, according to Dick Fuel Cell book
DG0 = 237200; #Gibbs free energy
T = 70 + 273.15 # "Stack temperature (unit = "K")"
z=2; #number of electrons per reaction
V0 = DG0/(z*F) #Reversible voltage
R_ohmic = delta_mem * 100 / (A * sigma_mem) # FC stack cell ohmic resistance (V)
#Open circuit voltage (Temperature dependant)
V_oc = V0 - 0.9e-3 * (T - T_std) + R * T / (F * 2) * math.log((P_O2 / P_an) ** 0.5 * (P_H2 / P_cat)) #Check fo
#Activation losses voltage
V_act = math.asinh(I_el/A/(i_0_an*2)) * R * T / (F * alfa_an) + math.asinh(I_el/A/(i_0_cat*2)) * R * T / (F *
V_act_an = math.asinh(I_el/A/(i_0_an*2)) * R * T / (F * alfa_an)
V_act_cat = math.asinh(I_el/A/(i_0_cat*2)) * R * T / (F * alfa_cat)
#Ohmic losses
V_ohm = R_ohmic * I_el
#Cell voltage
V = V_oc + V_act + V_ohm

eta_act_an = np.zeros(4000)
eta_act_cat = np.zeros(4000)
eta_ohm = np.zeros(4000)
V_vector = np.zeros(4000) #Vector that will collect all the voltages at different powers
V_matrix = np.zeros((4000,3)) #Matrix that will separate the previous vector in 3 columns, 1 for each T.

T = 343
#Delta_mem depends on T
Antoine_N = 8.07131 - 1730.63 / (233.426 + T - 273.15) # Parameter for antoine eq. (P_sat_water) mmHg ºc
P_sat_w = 10 ** Antoine_N * 133.322 # "Saturated pressure of water [Pa]"
P_O2 = P_an - P_sat_w # "Anode exchange current density [Pa]";
P_H2 = P_cat - P_sat_w # "Cathode exchange current density [Pa]";

    #Temperature dependant anode exchange current [mA/cm2]
i_0_an = gamma_M_an*i_0_std_an*math.exp(-Ea_an*1000/R*((1/T)-(1/T_std)))
i_0_cat = gamma_M_cat*i_0_std_cat*math.exp(-Ea_cat*1000/R*((1/T)-(1/T_std)))
    # Temperature dependant membrane conductivity
sigma_mem = (0.005139 * 22 - 0.00326) * math.exp(1268 * (1 / 303 - 1 / T)) # "Membrane conductivity (S/cm)";

R_ohmic = delta_mem * 100 / (A * sigma_mem) # FC stack cell ohmic resistance (V)
V_oc = V0 - 0.9e-3 * (T - T_std) + R * T / (F * 2) * np.log((P_O2 / P_an) ** 0.5 * (P_H2 / P_cat)) # Open circ
for e in range(0,4000):  # This goes over 0 - 4A/cm2 and gets one value of voltage for each current density. e
        V_act = np.log(e/A/(i_0_an*2) + ((e/A/(i_0_an*2))**2+1)**0.5) * R * T / (F * alfa_an) + np.log((e/A/(i
        V_act_an = np.log(e/A/(i_0_an*2) + ((e/A/(i_0_an*2))**2+1)**0.5) * R * T / (F * alfa_an)
        V_act_cat = np.log((e/A/(i_0_cat*2)) + ((e/A/(i_0_cat*2))**2+1)**0.5) * R * T / (F * alfa_cat)
        V_ohm = R_ohmic * e
        V = V_oc + V_act + V_ohm
        V_vector[e] = V
        eta_act_an[e] = V_act_an
```

**Figure A.13:** Electrolyzer efficiency

```
        eta_act_an[e] = V_act_an
        eta_act_cat[e] = V_act_cat
        eta_ohm[e] = V_ohm
# Creating the Voltage matrix.



I_el = np.linspace(start = 0, stop = 4000, num = 4000) #Intensity across the stack (A)
i_el = I_el*1000/A

I_el_data = np.array([400. ,500.,750.,1000.,1250.,1500.,1750.,2000.,2250.,2500.,2750.,3000.,3250.,3500.,3750.]
V_data = np.array([1.586, 1.600, 1.638, 1.669,1.710, 1.748, 1.783,1.814,1.845, 1.869,1.903, 1.934, 1.966, 1.99
plt.scatter(I_el_data, V_data, label = 'Experiment 70ºC')



V_70 = V_vector


plt.plot(I_el, V_70, label = 'Model 70ºC')

plt.xlabel('Current [A]')
plt.ylabel('Cell voltage [V]')
plt.legend()
plt.show()


#Declare vector for powers

P_used_70 = np.zeros(4000)


I_el = np.linspace(start = 1, stop = 4000, num = 4000) # Intensity across the stack (A)
#Calculate Power used by the stack

P_used_70 = V_vector * I_el *N_cells

Vth = 1.481 #Thermo-neutral voltage
#Declare faradaic efficiency vector
eta_F = np.zeros((4000,1))

#Calculate faradaic efficiency
k_H2 = 6.6e-13 * math.exp(-21030/(R*T))
k_O2 = k_H2 / 2
iH2x = 2 * F * k_H2 / delta_mem * P_cat
iO2x = 2 * F * k_O2 / delta_mem * P_an
eta_F = 1 - ((iH2x + iO2x)/ (I_el*1000/A))
eta_F[eta_F < 0] = 0.00000000000001 #Very low efficiency of faraday, to avoid division by 0

#Second law efficiency

eta_V_70 = 100*eta_F * V0/V_vector# consider HHV
```

**Figure A.14:** Electrolyzer efficiency

```

#First law efficiency

eta_VH_70 = 100*eta_F * Vth/V_vector # consider HHV

#Only plot until max power point is achieved
max_eff_position = np.argmax(eta_VH_70)
max_eff = eta_VH_70[max_eff_position]
eta_VH_70 = eta_VH_70[max_eff_position:]
P_used_70 = P_used_70[max_eff_position:]
i_el = i_el[max_eff_position:]

#Plotting the curves
plt.plot(i_el, eta_VH_70, color="orange")
plt.xlabel('Current density [mA/cm2]')
plt.ylabel('Efficiency [%]')
plt.legend()
plt.text(300, 98, '$\u03B5_{\u0394H}$', fontsize = 15)
plt.text(300, 80, '$\u03B5_{\u0394G}$', fontsize = 15)
plt.axis([0, 3250, 50, 110])
plt.show()


print(P_used_70)
```

**Figure A.15:** Electrolyzer efficiency

### Solar panel cost

In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Define the given data points
power_solar = [0.01, 1, 100, 1000, 10000]
capex_solar = [2650, 2650, 1560, 890, 890]

# Create a linear interpolation function
df_solar = pd.DataFrame({'Power': power_solar, 'Capex': capex_solar})

# Evaluate the interpolation function at a set of points
interpolated_power_solar = np.linspace(min(power_solar), max(power_solar), 60000)
interpolated_capex_solar = np.interp(interpolated_power_solar, df_solar.Power, df_s

# Create a new dataframe with the interpolated data
df_interpolated_solar = pd.DataFrame({'Power': interpolated_power_solar, 'Capex': i

# Set seaborn style and context
sns.set_style('whitegrid')
sns.set_context('poster')

# Create a figure and axis object
fig, ax = plt.subplots(figsize=(22, 13))

# Plot the data
sns.lineplot(data=df_interpolated_solar, x='Power', y='Capex', linewidth=2)

# Set the axis labels and title
ax.set_xlabel('Power rating')
ax.set_ylabel('Capex ($/kW)')
ax.set_title('Solar Panel Prices')

# Define a color palette with seaborn
color_palette = sns.color_palette("coolwarm", 3)  # 5 is the number of different ca

# Add shaded regions with labels
ax.fill_between(df_interpolated_solar['Power'], df_interpolated_solar['Capex'], 0,
ax.fill_between(df_interpolated_solar['Power'], df_interpolated_solar['Capex'], 0,
ax.fill_between(df_interpolated_solar['Power'], df_interpolated_solar['Capex'], 0,

# Show the legend
ax.legend(fontsize=25)

# Set x-axis to logarithmic scale and start from 0.01
ax.set_xscale('log')
ax.set_xlim(left=1)

# Define your custom ticks
x_ticks = [1, 10, 100, 1000, 10000]

# Generate your custom labels
x_labels = [str(x)+'kW' if x < 1000 else str(int(x/1000))+'MW' for x in x_ticks]

# Set the ticks and labels
ax.set_xticks(x_ticks)
ax.set_xticklabels(x_labels, fontsize=20)

# save the plot with the axis labels
#fig.savefig('Solar Panel Prices.png', bbox_inches='tight')

# Show the plot
plt.show()

```

**Figure A.16:** Solar panel cost scaling

**Battery cost**

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Define the given data points
power_battery = [0.01, 12.5, 600, 60000]
capex_battery = [1463.6, 1463.6, 1192, 428.5]

# Create a linear interpolation function
df_battery = pd.DataFrame({'Power': power_battery, 'Capex': capex_battery})

# Evaluate the interpolation function at a set of points
interpolated_power = np.linspace(1,90000, 50000)
interpolated_capex = np.interp(interpolated_power, df_battery.Power, df_battery.Cap

# Create a new dataframe with the interpolated data
df_interpolated = pd.DataFrame({'Power': interpolated_power, 'Capex': interpolated_

# Set seaborn style and context
sns.set_style('whitegrid')
sns.set_context('poster')

# Create a figure and axis object
fig, ax = plt.subplots(figsize=(22,13))

# Plot the data
sns.lineplot(data=df_interpolated, x='Power', y='Capex', linewidth=2)

# Set the axis labels and title
ax.set_xlabel('Capacity')
ax.set_ylabel('Capex ($/kW)')
ax.set_title('Battery Prices per capacity')


# Define a color palette with seaborn
color_palette = sns.color_palette("coolwarm", 4)  # 5 is the number of different ca

# Add shaded regions with labels
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.3, c
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.3, c
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.3, c
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.3, c

# Show the legend
ax.legend(fontsize=25)

# Set x-axis to logarithmic scale and start from 1
ax.set_xscale('log')
ax.set_xlim(left=1)

# Define your custom ticks
x_ticks = [1, 10, 100, 1000, 10000, 100000]

# Generate your custom labels
x_labels = [str(x)+'kWh' if x < 1000 else str(int(x/1000))+'MWh' for x in x_ticks]

# Set the ticks and labels
ax.set_xticks(x_ticks)
ax.set_xticklabels(x_labels, fontsize=20)

# save the plot with the axis labels
#fig.savefig('Battery Prices per capacity.png', bbox_inches='tight')

# Show the plot
plt.show()
```

**Figure A.17:** Battery cost scaling

### Electrolyzer cost scaling

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# create the data
# Define the given data points
power = [0.01,3, 6, 15, 200, 1000, 5000, 20000]
capex = [5500, 5500, 4083, 3033, 1200, 1000, 900, 700]

capex = [val * 3 for val in capex]

# Create a linear interpolation function
df = pd.DataFrame({'Power': power, 'Capex': capex})

# Evaluate the interpolation function at a set of points
interpolated_power = np.linspace(0.01, 50000, 40000)
interpolated_capex = np.interp(interpolated_power, df.Power, df.Capex)

# Create a new dataframe with the interpolated data
df_interpolated = pd.DataFrame({'Power': interpolated_power, 'Capex': interpolated_c

# Print the interpolated dataframe
print(df_interpolated)

# create a figure and axis object
sns.set_style('whitegrid')
sns.set_context('poster')
sns.set_palette('pastel')
fig, ax = plt.subplots(figsize=(22,13))

# plot the data
sns.lineplot(data=df_interpolated,x='Power', y='Capex',linewidth=7,zorder=2)

# set the axis labels and title
ax.set_xlabel('Power rating',fontsize=26)
ax.set_ylabel('Capex ($/kW)',fontsize=26)
ax.set_title('Electrolyzer Price per kW',fontsize=35)

# Define a color palette with seaborn
color_palette = sns.color_palette("coolwarm", 6)  # 6 is the number of different cat

# Use the colors from the palette in the fill_between method
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.3, co
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.3, co
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.3, co
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.3, co
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.3, co
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.3, co

# Set x-axis to logarithmic scale and start from 1
ax.set_xscale('log')
ax.set_xlim(left=1)


# Define your custom ticks
x_ticks = [1, 10, 100, 1000, 10000, 50000]

# Generate your custom labels
x_labels = [str(x)+'kW' if x < 1000 else str(int(x/1000))+'MW' for x in x_ticks]

# Set the ticks and labels
ax.set_xticks(x_ticks)
ax.set_xticklabels(x_labels, fontsize=20)

# Show the legend
ax.legend(fontsize=25)

# save the plot with the axis labels
#fig.savefig('Electrolyzer Price per kW.png', bbox_inches='tight')

# display the graph
plt.tight_layout()
plt.show()
```

**Figure A.18:** Electrolyzer cost scaling

### Fuel cell cost scaling

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Define the given data points
power = [0.01, 5, 10, 50, 500]
capex = [4000, 4000, 2500, 1800, 1200]

capex = [val * 3 for val in capex]

# Create a linear interpolation function
df = pd.DataFrame({'Power': power, 'Capex': capex})

# Evaluate the interpolation function at a set of points
interpolated_power = np.linspace(0.01, 20000, 30000)
interpolated_capex = np.interp(interpolated_power, df.Power, df.Capex)

# Create a new dataframe with the interpolated data
df_interpolated = pd.DataFrame({'Power': interpolated_power, 'Capex': interpolated_

# Create a figure and axis object
sns.set_style('whitegrid')
sns.set_context('poster')
sns.set_palette('pastel')
fig, ax = plt.subplots(figsize=(22,13))

# Plot the data
sns.lineplot(data=df_interpolated, x='Power', y='Capex', linewidth=7, zorder=2)

# Set the axis labels and title
ax.set_xlabel('Power rating', fontsize=26)
ax.set_ylabel('Capex ($/kW)', fontsize=26)
ax.set_title('Fuel cell Price per kW', fontsize=35)

# Define a color palette with seaborn
color_palette = sns.color_palette("coolwarm", 4)  # 5 is the number of different ca

# Add shaded regions with labels
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.2, c
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.2, c
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.2, c
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.2, c

# Set x-axis to logarithmic scale and start from 1
ax.set_xscale('log')
ax.set_xlim(left=1)

# Define your custom ticks
x_ticks = [1, 10, 100, 1000, 10000]

# Generate your custom labels
x_labels = [str(x)+'kW' if x < 1000 else str(int(x/1000))+'MW' for x in x_ticks]

# Set the ticks and labels
ax.set_xticks(x_ticks)
ax.set_xticklabels(x_labels, fontsize=20)

# Show the legend
ax.legend(fontsize=25)

# save the plot with the axis labels
#fig.savefig('Fuel cell Price per kW.png', bbox_inches='tight')

# Show the plot
plt.tight_layout()
plt.show()

```

**Figure A.19:** Fuel cell cost scaling

### Hydrogen storage

In [5]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Define the given data points
power_storage = [1, 2491, 100000,1000000]
capex_storage = [50, 12.8, 2,2]

power_storage = [val/33.3 for val in power_storage]
print(power_storage)

# Create a linear interpolation function
df_storage = pd.DataFrame({'Power': power_storage, 'Capex': capex_storage})

# Evaluate the interpolation function at a set of points
interpolated_power_storage = np.linspace(min(power_storage), max(power_storage), 500
interpolated_capex_storage = np.interp(interpolated_power_storage, df_storage.Power,

# Create a new dataframe with the interpolated data
df_interpolated_storage = pd.DataFrame({'Power': interpolated_power_storage, 'Capex'

# Set seaborn style and context
sns.set_style('whitegrid')
sns.set_context('poster')

# Create a figure and axis object
fig, ax = plt.subplots(figsize=(22, 13))

# Plot the data
sns.lineplot(data=df_interpolated_storage, x='Power', y='Capex', linewidth=2)

# Set the axis labels and title
ax.set_xlabel('H2 mass')
ax.set_ylabel('Capex ($/kW)')
ax.set_title('Hydrogen Storage Prices')

# Define a color palette with seaborn
color_palette = sns.color_palette("coolwarm", 3)  # 4 is the number of different cat

# Add shaded regions with labels
ax.fill_between(df_interpolated_storage['Power'], df_interpolated_storage['Capex'],
ax.fill_between(df_interpolated_storage['Power'], df_interpolated_storage['Capex'],
ax.fill_between(df_interpolated_storage['Power'], df_interpolated_storage['Capex'],

# Show the legend
ax.legend(fontsize=25)

# Set x-axis to logarithmic scale and start from 0.01
ax.set_xscale('log')

ax.set_xlim(left=10)

# Set x-axis to logarithmic scale and start from 0.01
ax.set_xscale('log')
ax.set_xlim(left=1)

# Define your custom ticks
x_ticks = [1, 10, 100, 1000, 10000]

# Generate your custom labels
x_labels = [str(x)+'kg' if x < 1000 else str(int(x/1000))+'tons' for x in x_ticks]

# Set the ticks and labels
ax.set_xticks(x_ticks)
ax.set_xticklabels(x_labels, fontsize=20)

# save the plot with the axis labels
#fig.savefig('Hydrogen Storage Prices.png', bbox_inches='tight')

# Show the plot
plt.show()
```

**Figure A.20:** Hydrogen storage cost scaling

**Diesel generator cost**

In [12]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Define the given data points
power = [1, 10, 100, 1000, 10000]
capex = [700, 400, 275, 200, 140]

capex = [val * 4 for val in capex]

# Create a linear interpolation function
df = pd.DataFrame({'Power': power, 'Capex': capex})

# Evaluate the interpolation function at a set of points
interpolated_power = np.linspace(0.01, 15000, 30000)
interpolated_capex = np.interp(interpolated_power, df.Power, df.Capex)

# Create a new dataframe with the interpolated data
df_interpolated = pd.DataFrame({'Power': interpolated_power, 'Capex': interpolated_

# Create a figure and axis object
sns.set_style('whitegrid')
sns.set_context('poster')
sns.set_palette('pastel')
fig, ax = plt.subplots(figsize=(22,13))

# Plot the data
sns.lineplot(data=df_interpolated, x='Power', y='Capex', linewidth=7, zorder=2)

# Set the axis labels and title
ax.set_xlabel('Power rating', fontsize=26)
ax.set_ylabel('Capex ($/kW)', fontsize=26)
ax.set_title('Diesel Generator Price per kW', fontsize=35)

# Define a color palette with seaborn
color_palette = sns.color_palette("coolwarm", 5)

# Add shaded regions with labels
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.2, c
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.2, c
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.2, c
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.2, c
ax.fill_between(df_interpolated['Power'], df_interpolated['Capex'], 0, alpha=0.2, c

# Set x-axis to logarithmic scale and start from 1
ax.set_xscale('log')
ax.set_xlim(left=1)

# Define your custom ticks
x_ticks = [1, 10, 100, 1000, 10000, 100000]

# Generate your custom labels
x_labels = [str(x)+'kW' if x < 1000 else str(int(x/1000))+'MW' for x in x_ticks]

# Set the ticks and labels
ax.set_xticks(x_ticks)
ax.set_xticklabels(x_labels, fontsize=20)

# Define the given data points for OPEX
power_opex = [1, 10, 100, 1000, 10000]
opex = [7884, 7500, 7000, 6000, 5000]

opex = [val / 8760 for val in opex]

# Create a linear interpolation function for OPEX
df_opex = pd.DataFrame({'Power': power_opex, 'Opex': opex})

# Evaluate the interpolation function at a set of points
interpolated_opex = np.interp(interpolated_power, df_opex.Power, df_opex.Opex)

# Add the OPEX data to the interpolated dataframe
df_interpolated['Opex'] = interpolated_opex
```

**Figure A.21:** Diesel generator cost scaling

```
# Plot the OPEX data on a secondary y-axis
ax2 = ax.twinx()
sns.lineplot(data=df_interpolated, x='Power', y='Opex', ax=ax2, linewidth=7, zorder=

# Set the axis labels and title for the OPEX plot
ax2.set_ylabel('Opex ($/kW/hour)', fontsize=26)
ax2.grid(False)

# Show the legend
ax.legend(fontsize=25)
from matplotlib.lines import Line2D

line = Line2D([0], [0], color='green', linewidth=7, linestyle='-')
ax2.legend([line], ['Opex'], fontsize=25, bbox_to_anchor=(1, 0.73))

# Set the range for the secondary y-axis
ax2.set_ylim([0.5, 1])

# save the plot with the axis labels
#fig.savefig('Diesel Generator Prices.png', bbox_inches='tight')

# Show the plot
plt.show()
```

**Figure A.22:** Diesel generator cost scaling

```
In [7]:
import pandas as pd
import numpy as np

#Fuel cell pricing
# Define the given data points
power_fc = [0.01, 5, 10, 50, 500]
capex_fc = [4000, 4000, 2500, 1800, 1200]

# Multiply the fuel cell capex values by 3
capex_fc = [val * 3 for val in capex_fc]

# Create a linear interpolation function
df_fc = pd.DataFrame({'Power': power_fc, 'Capex': capex_fc})
interpolation_func = np.interp

# Create a function that returns the price per kW for a given power
def get_fuel_cell_price(power_fc):
        # Use the interpolation function to evaluate the capex at the given power
        capex_at_power = interpolation_func(power_fc, df_fc.Power, df_fc.Capex)

        # Calculate the price per kW by dividing the capex by the power
        price_per_kw = capex_at_power * power_fc

        return price_per_kw


#Electrolyzer pricing
# Define the given data points
power_el = [0.01, 3, 6, 15, 200, 1000, 5000, 20000]
capex_el = [5500, 5500, 4083, 3033, 1200, 1000, 900, 700]

# Multiply the electrolyzer capex values by 3
capex_el = [val * 3 for val in capex_el]

# Create a linear interpolation function
df_el = pd.DataFrame({'Power': power_el, 'Capex': capex_el})
interpolation_func = np.interp

def get_electrolyzer_price(power_el):
        # Use the interpolation function to evaluate the capex at the given power
        capex_at_power = interpolation_func(power_el, df_el.Power, df_el.Capex)

        # Calculate the price per kW by dividing the capex by the power
        price_per_kw = capex_at_power * power_el

        return price_per_kw

# Battery pricing
# Define the given data points
power_battery = [0.01, 12.5, 600, 60000]
capex_battery = [1463.6, 1463.6, 1868, 1454]

# Create a linear interpolation function
df_battery = pd.DataFrame({'Power': power_battery, 'Capex': capex_battery})
interpolation_func = np.interp


# Create a function that returns the price per kW for a given power
def get_battery_price(power_bat):
    # Use the interpolation function to evaluate the capex at the given power
    capex_at_power = interpolation_func(power_bat, df_battery.Power, df_battery.Ca

    # Calculate the price per kW by dividing the capex by the power
    price_per_kw = capex_at_power * power_bat

    return price_per_kw



# Solar pricing
# Define the given data points
power_solar = [0.01, 1, 100, 1000, 10000]
capex_solar = [2650, 2650, 1560, 890, 890]
efficiency = 0.2

area_solar = [(val*efficiency) for val in capex_fc]

```

**Figure A.23:** Total cost scaling

```python


# Create a linear interpolation function
df_solar = pd.DataFrame({'Power': power_solar, 'Capex': capex_solar})
interpolation_func = np.interp

# Define the efficiency of the solar panel

# Create a function that returns the price per square meter for a given area
def get_solar_price_per_sqm(area):
    power = area*efficiency*1000
    # Use the interpolation function to evaluate the capex at the calculated power
    capex_at_power = interpolation_func(power, df_solar.Power, df_solar.Capex)

    # Calculate the price per watt by multiplying the capex with the area
    price_per_sqm = capex_at_power * area

    return price_per_sqm


# Hydrogen storage pricing
power_storage = [0.01, 1, 2491, 100000] #https://www.pnnl.gov/sites/default/files/r
capex_storage = [50, 50, 12.8, 2]

# Multiply the storage capex values by 3
capex_storage = [val * 3 for val in capex_storage]

df_storage = pd.DataFrame({'Power': power_storage, 'Capex': capex_storage})
interpolation_func = np.interp

# Create a function that returns the price per kWh for a given power
def get_storage_price(power_storage):
    # Use the interpolation function to evaluate the capex at the given power
    capex_at_power = interpolation_func(power_storage, df_storage.Power, df_storage
    # Calculate the price per kWh by dividing the capex by the power
    price_per_kwh = capex_at_power * power_storage
    return price_per_kwh


# Create a dictionary of prices for batteries, electrolyzers, and fuel cells
prices = {
    'batteries': get_battery_price,   # interpolated values based on power
    'electrolyzers': get_electrolyzer_price,   # interpolated values based on powe
    'fuel cells': get_fuel_cell_price,   # interpolated values based on power
    'solar panels': get_solar_price_per_sqm,   # interpolated values based on pow
    'hydrogen storage': get_storage_price,   # interpolated values based on power
}

# Example usage: get the price per kW for a 1 kW electrolyzer
price_per_kw_el = prices['electrolyzers'](1)
print(f"The price per kW for a 1 kW electrolyzer is ${price_per_kw_el:.2f}")

# Example usage: get the price per kW for a 1 kW fuel cell
price_per_kw_fc = prices['fuel cells'](1)
print(f"The price per kW for a 1 kW fuel cell is ${price_per_kw_fc:.2f}")

# Example usage: get the price per kW for a 15 kWh battery
price_per_kwh_bat = prices['batteries'](15)
print(f"The price per kW for a 15 kW battery is ${price_per_kwh_bat:.2f}")

# Example usage: get the price for a 25 sqm solar panel
price_per_sqm = get_solar_price_per_sqm(25)
print(f"The price per square meter for a 25 sqm solar panel is ${price_per_sqm:.2f

# Example usage: get the price for a 33 kwh of storage
price_per_kwh = prices['hydrogen storage'](33)
print(f"The price per kwh for a 33 kwh of H2 ${price_per_kwh:.2f}")

# Total price
total_price = price_per_kw_el+price_per_kw_fc+price_per_kwh_bat+price_per_sqm+price
print(f"The total price is ${total_price:.2f}")

```

```
The price per kW for a 1 kW electrolyzer is $16500.00
The price per kW for a 1 kW fuel cell is $12000.00
The price per kW for a 15 kW battery is $21979.81
The price per square meter for a 25 sqm solar panel is $22250.00
The price per kwh for a 33 kwh of H2 $4902.67
The total price is $77632.48
```

**Figure A.24:** Total cost scaling