Modelling of Global Levelized Cost of Hydrogen under Use of an Open-Source Modelling Environment

Master's Thesis for the Attainment of the Degree

M.Sc. Energy and Process Technology

at the School of Engineering and Design at the Technical University of Munich

Supervised by: Dr.-Ing. Serafin von Roon
M.Sc. Tapio Schmidt-Achert
Forschungsstelle für Energiewirtschaft e.V. (FfE e.V.)
M.Sc. Patrick Buchenberg
Lehrstuhl für Erneuerbare und Nachhaltige Energiesysteme (ENS)

Submitted by: Miguel Ángel Martínez Pérez

Submitted: 28.09.2022 in Munich

Abstract

Germany's target to become climate-neutral by 2045 creates high expectations for a green hydrogen economy as a strategic path in the energy transition. Still, the questions about where the hydrogen will be produced and at what cost must be answered. This thesis helps answer these questions, focusing on developing an electrolysis-based hydrogen production model worldwide. The employed methodology starts with selecting a python-based software, PyPSA in this case. This software is used to minimize the levelized cost of hydrogen (LCOH) for a defined electrolysis-based hydrogen production system by optimizing its size, using linear optimization. The model is implemented using status quo techno-economic parameters together with the country risk premiums, which characterize the economic risk of the different countries.

The worldwide model results show that battery storage system is still too expensive to bring an advantage to the system. Furthermore, the hybrid PV-wind configuration reduces the LCOH taking certain regions of the world into a more economically competitive position. Finally, the country risk premiums can shape the LCOH distribution worldwide, excluding countries with low economic attractiveness from an investor perspective, despite having high renewable energy potential.

Kurzfassung

Das Ziel Deutschlands, bis 2045 klimaneutral zu werden, weckt hohe Erwartungen an eine grüne Wasserstoffwirtschaft als strategischen Pfad der Energiewende. Dennoch müssen die Fragen beantwortet werden, wo der Wasserstoff produziert werden soll und zu welchen Kosten. Die vorliegende Arbeit trägt zur Beantwortung dieser Fragen bei und konzentriert sich auf die Entwicklung eines Modells für die weltweite Wasserstoffproduktion auf Elektrolysebasis. Die angewandte Methodik beginnt mit der Auswahl einer python-basierten Software, in diesem Fall PyPSA. Diese Software wird verwendet, um die Wasserstoffgestehungskosten (LCOH) für ein definiertes elektrolysebasiertes Wasserstoffproduktionssystem zu minimieren, indem dessen Größe mithilfe der linearen Optimierung optimiert wird. Das Modell wird unter Verwendung von techno-ökonomischen Status-quo-Parametern zusammen mit den „Country Risk Premiums“ implementiert, die das wirtschaftliche Risiko der verschiedenen Länder charakterisieren.

Die weltweiten Modellergebnisse zeigen, dass das Batteriespeichersystem immer noch zu teuer ist, um einen Vorteil für das System zu bringen. Darüber hinaus reduziert die hybride PV-Wind-Konfiguration die LCOH, was bestimmte Regionen der Welt in eine wirtschaftlich wettbewerbsfähigere Position bringt. Schließlich können die „Country Risk Premiums“ die LCOH-Verteilung weltweit beeinflussen und Länder ausschließen, die trotz ihres hohen Potenzials an erneuerbaren Energien aus Sicht der Investoren wirtschaftlich wenig attraktiv sind.

Statement of Academic Integrity

I,

Last name:	Martínez Pérez
First name:	Miguel Ángel
ID No.:	03737010

Hereby confirm that the attached thesis,

Modelling of Global Levelized Cost of Hydrogen under Use of an Open-Source Modelling Environment

Was written independently by me without use of any sources or aids beyond those cited, and all passages and ideas taken from other sources are indicated in the text and given the corresponding citation.

I confirm to respect the „Code of Conduct for Safeguarding Good Academic Practice and Procedures in Cases of Academic Misconduct at Technische Universität München, 2015", as can be read on the website of the Equal Opportunity Office of TUM.

Tools provided by the chair and its staff, such as models or programmes, are also listed. These tools are property of the institute or of the individual staff members. I will not use them for any work beyond the attached thesis or make them available to third parties.

I agree to the further use of my work and its results (including programmes produced and methods used for research and instructional purposes).

I have not previously submitted this thesis for academic credit.

Munich, 28.09 2022

Martínez Pérez, Miguel Ángel

List of contents

List of figures

List of tables

1 Introduction

Hydrogen is expected to be a key component for the energy transition and, therefore, for the decarbonization of the European and global energy system [1]. Germany is also supporting a hydrogen economy as a key strategy to fulfill the climate neutrality targets set by the European Union by 2050 and set in the German Federal Climate Change Act (Klimaschutzgesetz) by 2045. Furthermore, as Germany has limited potential for national green hydrogen, the government plans to import high amounts of hydrogen [2].

Due to the national hydrogen strategies, new questions concerning green hydrogen have been raised. One of these questions regards the worldwide regions with higher hydrogen production potential and the costs for its production, which is addressed in this thesis.

1.1 Motivation

The motivation for this thesis lies in two different activities. First, the national hydrogen strategies create a need, to know which countries or regions in the world will be able to produce the cheapest green hydrogen.

Secondly, this thesis is written in the FfE Munich (Forschungsstelle für Energiewirtschaft), whose main activity is the analysis of today's energy system and how this energy system might develop in the future. For this goal, they developed the simulation model ISAaR (Integriertes Simulationsmodell zur Anlageneinsatz- und Ausbauplanung mit Regionalisiserung). The ISAaR is a linear optimization model that describes the European energy system mathematically, and one of the questions that addresses, is the role of hydrogen in the future energy system.

The ISAaR already considers green hydrogen in its mathematical model, as shown in the Figure 1-1, however, the accuracy of the model may be improved by implementing a more consistent cost potential curve for hydrogen as input for the ISAaR, i.e. the correlation between the amount of hydrogen and its cost. A series of steps are needed to obtain the cost potential curves for the different countries or regions. The first one is the optimization of the hydrogen production costs using weather data, which is tackled in this thesis.

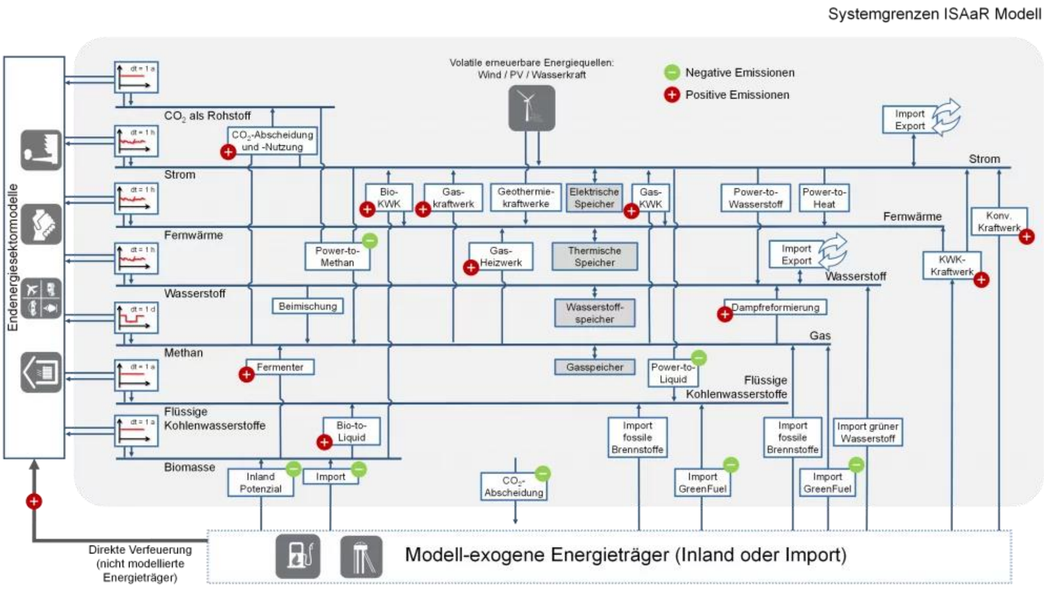


Figure 1-1 System boundaries, energy carrier rails and sector-coupling conversion technologies in ISAaR [3]

This cost potential curve of hydrogen includes two different costs. On the one hand, the production costs of hydrogen represented by the levelized cost of hydrogen (LCOH), and on the other hand the import costs, which is how much does it cost to import the produced hydrogen from the production site to the consumption site, in this case Germany in Europe, for the ISAaR model.

The scope of this thesis considers the production cost of hydrogen, which is the levelized cost of hydrogen (LCOH), leaving the land use potential analysis and import costs for future studies.

To model the levelized cost of hydrogen (LCOH), an open-source environment based on linear optimization will be used.

1.2 Research Questions

The main goal of this thesis is to model the levelized cost of hydrogen (LCOH) in a global scale considering different techno-economic criteria. To achieve this goal the following research questions are proposed:

1. Which Open-Source environment is appropriate to model the Levelized Cost of Hydrogen (LCOH) production worldwide?
2. What are the effects of considering a hybrid wind-solar system and battery storage system on the Levelized Cost of Hydrogen (LCOH)?
3. Which are the effects of Country Risk Premium (CRP) on the Levelized Cost of Hydrogen (LCOH)?

1.3 Thesis Structure

Firstly, a theoretical framework concerning economic criteria regarding the production of hydrogen is summarized, focusing mainly on the levelized cost of hydrogen (LCOH) and on the country risk premiums. Following, the electrolysis-based hydrogen production system is explained as well as the main ideas about modelling energy systems and linear programming.

Then an analysis of the existing models for the levelized cost of hydrogen (LCOH) is encapsulated in a matrix, with the main assumptions regarding geographical scope and technologies used for the hydrogen production.

Afterwards the followed methodology is presented including the electrolysis-based hydrogen production system to be modelled, the explanation of the used qualitative assessment for the selection of the open-source software and finally all the criteria specific needed for the development of the model. These criteria involve the identification of input data, the development of a database structure for all inputs and outputs, the program code and processing of all the inputs and finally the selection of calculation scenarios.

To finish, the presentation of the results answering the research questions followed by the conclusions and a small outlook for this master's thesis.

2 Theoretical Framework

This chapter contains three main parts. First, the fundaments on specific hydrogen economy techno-economic criteria, followed by the electrolysis-based hydrogen production system, and finally, a short allusion to energy systems modelling.

2.1 Techno-economic criteria

This techno-economic criteria chapter refers to the characterization of the hydrogen production cost.

2.1.1 Levelized Cost of Hydrogen (LCOH)

The concept levelized cost of hydrogen is derived from the long existing concept levelized cost of energy (LCOE). The LCOE represents the sum of investment and operational cost of the power plant producing the energy throughout its lifetime, divided by the total energy produced in its lifetime [4, 5]. In the same line, the levelized cost of hydrogen (LCOH) represents the sum of all the investment (CAPEX) and operational (OPEX) costs for the components involved in the production of hydrogen divided by the amount of hydrogen produced throughout its lifetime. Just as the LCOE allows the comparison of different alternative power plants for the electricity production, the LCOH is a key concept for assessing different hydrogen production systems [4–6].

$$LCOH = \frac{Total\ CAPEX\ and\ OPEX\ costs\ through\ lifetime}{Hydrogen\ produced\ through\ lifetime}\ in\ \frac{€}{kg_{H_2}}$$

The LCOH is expressed in €/kg_{H2}. The components to be considered in the calculation of the LCOH, include the systems for the production, transmission and storage of electricity, heat, and hydrogen. Additional systems such as water pumping systems, desalination plants or other infrastructures should be also included.

2.1.2 Annuity Factor (AnF) and interest rate (r)

An alternative way to calculate the LCOH is to consider the total costs and produced hydrogen over a year instead of over its lifetime. To achieve that, the CAPEX costs should be annualized through an annuity factor (AnF).

The annuity factor is normally used from an investing perspective to calculate the present value of a cash flow series, with a specific interest rate r, also known as a discount rate or expected return rate, and with a number of years n, during which the annual cash flow payment C is received [7–9].

$$Present\ Value = C \cdot AnF = C * \sum_{n=1} \frac{1}{(1+r)^n} = C \cdot \frac{(1+r)^n - 1}{r * (1+r)^n}$$

From a company cost perspective, for example to invest in a project, the annuity factor is used to annualize the current total investment cost (CAPEX), which means distributing the complete CAPEX in equal annual payments through the lifetime of the project or system. In this case the, the above mentioned "Present Value" would represent the current total investment cost and the C would represent the annualized cost to pay during the life of the system, n [10].

$$Total\ CAPEX = C \cdot AnF = C * \sum_{n=1}^{n} \frac{1}{(1+r)^n} = C \cdot \frac{(1+r)^n - 1}{r * (1+r)^n}$$

Then, clearing the C,

$$Annualized\ cost = C = \frac{Total\ CAPEX}{AnF}$$

Being again,

$$AnF = \frac{(1-(1+r)^n)}{r} = \frac{(1+r)^n - 1}{r * (1+r)^n}$$

Applying the annuity factor to the LCOH formula must be done carefully, given that each component i of the hydrogen production system has a different lifetime and even maybe different interest rate.

$$LCOH = \frac{\sum_{i=1}^{i} (\frac{Total\ CAPEX_i}{AnF_i} + annual\ OPEX_i)}{Hydrogen\ produced\ annually} \quad in \quad \frac{€}{kg_{H_2}}$$

Generally, the OPEX is given as a percentage of the total CAPEX, which is why the equation can also be expressed as:

$$LCOH = \frac{\sum_{i=1}^{i} Total\ CAPEX_i \cdot (\frac{1}{AnF_i} + OPEX_i(\%))}{Hydrogen\ produced\ annually} \quad in \quad \frac{€}{kg_{H_2}}$$

While the lifetime n of the system or component is direct to know, the interest rate r per period is more difficult to understand [8].

Let us consider the common case where a company wants to start a project. In that case, it is important to differentiate between the company to which the project belongs, and the investors that will provide the capital for the project, normally banks or stockholders. From the company's perspective, the interest rate is an interest cost and from the investor's perspective, it is the minimum rate of return required to invest in the project.

Usually, the weighted average cost of capital (WACC) is used as this interest cost or expected rate of return, as it represents the average costs that the company has to pay for debt and equity holders [9]. Then the WACC provides, in a single value, the minimum profit required by the investors (banks and stakeholders) to be willing to borrow the capital, which corresponds with company's capital cost [11].

The capital costs are then divided into equity capital and debt capital. The equity capital represents the cost of money funded by the company stakeholders, which is the minimum rate of return expected to take the risk of investing. On the other hand, debt capital represents the money funded by loans, usually from banks. Each type of capital cost is weighted depending on the company's capital structure so that they contribute differently to the WACC [11]:

$$WACC = \frac{E}{V} * R_E + \frac{D}{V} * R_D * (1 - T)$$

Where:

- E/V is the share of equity capital
- D/V is the share of debt capital
- R_E is the cost of equity calculated usually by the Capital Asset Pricing Model (CAPM), from investors
- R_D is the cost of debt, which is the interest rate of the obtained loan
- T represents the tax rate to consider the tax-deductible share of the loan

- **Capital Asset Pricing Model (CAPM)**

The CAPM is used to calculate the cost of equity, R_E, for a specific company in a specific market [8]. The model has different variants but the simplest one considers the risk-free rate and the premium risk given by the specific market and the sensitivity of the company or investment to the market. Another common consideration in the CAPM is the country risk premium (CRP), which represents an additional risk associated to the country where the investment is being considered [8, 12].

The simplest CAPM formula is:

$$R_E = Rf + \beta * (Rm - Rf) + CRP$$

Where:

- R_E is the cost of equity
- R_f is the risk-free rate
- R_m is the expected market return
- (R_m – R_f) is the market risk premium
- β represents the sensitivity of the company or investment to the market
- CRP is the country risk premium

As a simplification is to consider that the capital costs come completely from costs of equity, which means that the invested money is completely funded by stakeholders and there are no loans.

Therefore, the WACC would be equal to the cost of equity.

$$WACC = r = R_E = \ Rf + \beta * (Rm - Rf) + CRP = WACC' + CRP$$

The equity risk premium represents the excess risk over the risk-free rate in a specific market. The equity risk premium cost can be then deduced from the above WACC equation [13]:

$$Equity\ Risk\ Premium = \ WACC - R_f = \ \beta * (Rm - Rf) + CRP$$

2.1.3 Country Risk Premium (CRP)

The country risk premiums take into account the additional risk of investing in a specific country in comparison to another country [14]. Professor Damodara from the Stern School of Business from the New York University (NYU) has developed extensive studies for the country risk premiums and also provides this country risk premium data. This data can be found in Appendix I: Country Risk Premiums (January 5, 2022) [15, 16]. The country risk premiums are based on default risk for the specific countries, an additional risk in case the country stops paying its international debt. An essential question to approach here is why the country risk premiums are needed, and to answer the existing types of investment risks must be understood.

2.1.3.1 Assessment of investment risks

Two types of investment risks are identified, unsystematic or diversifiable risks and systematic or non-diversifiable risks [8].

As its name suggests, the diversifiable risk can be eliminated through diversification, which means that investing in a large number of projects or in different markets or countries will reduce the risk to zero. However, as this risk is eliminable, it will not add any premium rate of return to the risk-free rate. Therefore, diversifiable risks cannot add a risk premium [8].

On the other hand, the non-diversifiable risks cannot be eliminated, which means that the company will depend on the overall economy, which may be affected by unpredictable events. To assume the non-diversifiable risks, the investors will demand a higher profit which translates into a risk premium [8].

2.1.3.2 Is country risk diversifiable or non-diversifiable?

Now the question is if the country risk premium is a diversifiable or non-diversifiable risk. This question is critically analyzed with arguments for and against it. Globalization is a key argument against a country risk premium because the possibility of investing in multiple countries will, in the end, diversify the risk. Therefore, just considering this, the country risk should be diversifiable. Nevertheless, the approach should also be considered more carefully, firstly considering if the average investor has truly access or absence of impartiality for investing in multiple countries and secondly if there are positive return correlations between countries. A positive return correlation means that investing in one country will positively affect other countries' investments. These correlations have been intensively studied in the last decades and appear to be of increasing importance. [17].

The country risk may look diversifiable at first sight, and in part it is, but always there will be a residual country risk that is non-diversifiable. Although there might be other approaches to include this risk, a country risk premium represents a direct and realistic one [14].

2.1.3.3 Calculation of the country risk premium

Despite globalization being present in the current worldwide economy, there is no standard approach for calculating the country risk premiums. The Stern School of Business at New York University uses two different calculation approaches for the country risk premiums. These approaches have similarities, but they deviate from the data sources. Both quantify the long-term country risk first, which is later corrected by summing up the short-term country risks [18]. The following table summarizes each approach, alongside its advantages and drawbacks.

Table 2-1 Country risk premium calculation approaches considered by the Stern School of Business (NYU) [16]

	First Approach	Second Approach
Quantify long-term country risk	Estimate the bond default spread. Using local currency sovereign rating.	Default spreads from sovereign Credit Default Swap (CDS) market.
Correct the previous adding short-term country risk	Equity market volatility	Equity market volatility
Advantage	Default spread characterizes the market risk better than rating agencies (rating agencies consider additional criteria) Default spreads are dynamic	Sovereign CDS reflect current situation of default risk. Dynamic
Downsides	Correlation between qualitative (rating) and quantitative (bond default) indicators not perfect. Ratings depend on private financial agencies. Not regularly updated	Sovereign CDS include additional risks not related with country risks. No sovereign CDS for every country.

<u>First approach for the calculation of the country risk premiums</u>

This first method quantifies the long-term risk using the local currency sovereign rating to estimate the bond default spreads. Later, the effect of the short-term country risk is added through the equity market volatility [16].

- **Local currency sovereign rating**

The local currency sovereign rating is assessed by agencies such as Moody's or Standard & Poor's to determine if a country can meet its financial responsibilities. This assessment considers political, economic, financial, legal and fiscality factors, among others. The sovereign rating cannot be directly used for the calculation of the risk premium. However, it must be correlated to obtain how much a country deviates from the best rated countries through the bond default spread.

One of the most known country ratings is Moody's rating. It results from an extensive rating process of financial obligations implemented by the private financial service company Moody's Investors Service. This rating can be divided into short-term and long-term rating [19].

The short-term ratings characterize the ability of a financial body to return all short-term obligations. On the other hand, the long-term ratings denote the credit risk for fixed financial obligations with a maturity year of at least a year. That means how likely will the financial obligation be fulfilled. Ultimately the long-term ratings represent a qualitative credit risk of an investment [19].

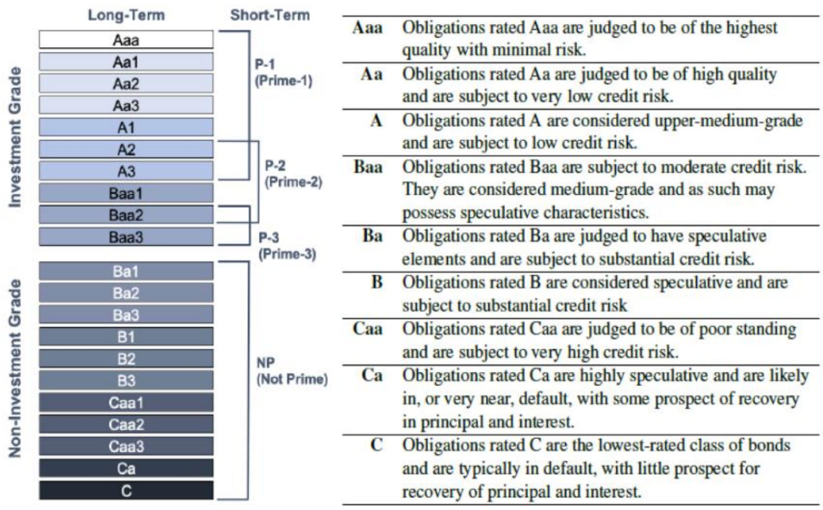


Figure 2-1 Moody's global long-term rating scale. Source: Moody's Investor Service

Moody's Corporation is not the only rating agency existing on the market, Standard & Poor's (S&P) is another important financial rating agency. The downside about these ratings is that their update depends completely on the rating agency, and therefore they are not often updated. These agencies consider factors such as [18]:

- Political risk
- Economic structure
- Economic growth prospects
- Fiscal flexibility
- General government debt burden
- Offshore and contingent liabilities

- Monetary flexibility
- External liquidity
- External debt burden

These agencies also differentiate between local and foreign currency ratings:

- Foreign currency rating. Due to the current globalized economy, many countries have been developing their economies by borrowing money from other countries. These loans are defined in foreign currency. The inability of the borrowing country to pay back to its issuer in the foreign currency results in default. This kind of default is common, and it is rated [18].
- Local currency rating. More complicated to comprehend are the local currency ratings. These assess the inability of a country to pay its local debt, which means how likely a local currency default can occur. It is easy to think that local currency default can be avoided just by printing more money, given that the country has control over the local currency. However, there are some reasons why local currency defaults are preferred over printing more money. Two of these reasons are:
    - There is a shared currency between countries (such as the euro), so printing money is limited for the benefit of the common currency system [18].
    - There is a trade-off between default and currency devaluation. Currency devaluation implies prestige loss, political instability, inflation, and economic recession. One key factor to decide between default or printing more money is how deep the debt of local companies is dependent on foreign currency. If the corporations have a high rate of foreign currency debt, the devaluation of the local currency could ruin their finances and in that case is the default preferable [18].

For calculating the country risk premiums, the Stern School of Business at the NYU uses Moody's long-term ratings for the local currency [18]. As these ratings denote a qualitative credit risk and not a quantitative one, there is the need to quantify these country ratings, accomplished with the bond default spreads.

- **Bond default spread**

The bond default spread is the difference between the interest rate of a country specific bond and the interest rate "risk-free" bond, being both bonds issued for the same period of time and in the currency corresponding to the “risk-free” bond to make them comparable [18].

The “risk-free” bond also entails risk, but the goal here is to set a reference bond (preferably one with a powerful and accessible currency such as dollar or euro) to which other countries can be compared. The default spread serves as a direct way to characterize the long-term country risk premium, however not every country issues bonds in foreign currency. Therefore, an alternative way is needed to characterize the country risk premiums for other countries [18].

This alternative consists of identifying Moody’s local currency ratings for the countries with existing default spreads, then calculating the average default spread and correlating this result with the countries with the same sovereign Moody’s rating, also the ones with existing default spreads [16, 20].

In case a country has neither foreign currency bonds nor sovereign ratings, this approach cannot be utilized.

$$Bond\ Def.Spread = Int.rate\ bond_{ctry\ x,t\ years,curr\ a} - Int.rate\ bond_{risk-free\ ctry,t\ years,curr\ a}$$

$$Volatibility\ fator = \frac{\sigma_{country\ x}}{\sigma_{risk-free\ country}}$$

- **Equity Market Volatility**

Once the long-term risk is quantified through the sovereign ratings and the bond default spread, the short-term risks are to be considered through a volatility factor. To address this, the equity markets (stock markets) are compared. The parameter used as the multiplication factor is the ratio of the standard deviation for the country’s equity market divided by the standard deviation of the reference equity market (reference used in the bond default spread calculation) [18].

$$Volatibility\ fator = \frac{\sigma_{country\ x}}{\sigma_{risk-free\ country}}$$

The first calculation approach can be summarized in the following equation.

$$CRP_1 = Bond\ Def.Spread\ (estimated\ through\ Moody's\ rating) * Volatility\ factor$$

Second approach for the calculation of the country risk premiums

The second quantification method is quite similar to the first, but it uses Credit Default Swap (CDS) spreads instead of bond default spreads [16, 18].

- **Credit Default Swap market (CDS)**

The CDS market was created in 1994 and has evolved in the last three decades. This market prices the default risk of different financial bodies so that the buyer can be protected in case of default [21].

The CDS market is divided into different branches, being the corporate CDS the biggest one, second the bank CDS and lastly, concerning the country risk, the sovereign CDS. The sovereign CDS denotes the default risk of a nation for not being able to pay back its government bonds [16, 21].

The weakness in the sovereign CDS market is its dependence on other risks unrelated to the country default risk, such as liquidity and market narrowness, which make them more volatile. The strength of sovereign CDS is that they are regularly updated. On the other hand, ratings are more reliable but depend on a rating agency to be updated, so they respond slower to changes in specific markets and countries [18, 21].

The CDS default spread calculation is analog to the bond default spread calculation.

$$CDS\ Def.Spread = CDS_{ctry\ x} - CDS_{risk-free\ ctry}$$

The volatility factor is identical to the first approach. The second calculation approach for the country risk premiums quantification is summarized in the following equation.

$$CRP_2 = CDS\ Def.Spread + Volatility\ factor$$

The CDS Market does not include most countries, which is one of the reasons for using the first method for the country risk premiums quantification. Also, the sovereign CDS market is quite volatile in terms of the narrowness and liquidity of the market. Any changes in the market parties or liquidity, independent from the country risk, will be reflected in the sovereign CDS rates [20].

2.1.3.4 Effect of the base rate on the CRP

The base rate is the interest rate defined by a central bank and charged to a commercial bank for borrowing money [22], consequently, the base rate is related to the debt capital; see 2.1.2.

The country risk premiums are usually introduced in the Capital Assessment Pricing Model (CAPM) as an additional risk premium and therefore considered part of the equity capital [12]; see also 2.1.2. For this reason, the base rate and the country risk premium are independent and the increase or decrease of one does not affect the other or vice versa.

Just to give an example, if the base rate increases, the WACC of the company will also increase, but it would increase equally for every country, as the country risk premium is not affected. In the model assumptions, the WACC is simplified, considering only the equity capital, this assumption can be understood as if the company’s capital comes completely from stakeholders and not from bank loans. Therefore, no debt capital is considered in the calculations and the base rate would not affect the annualized costs of the components.

2.2 Electrolysis-based hydrogen production system

The following chapter briefly describes the electrolysis-based hydrogen production system and its main components.

2.2.1 Overview of the system

The production of electrolysis-based hydrogen is becoming more economically viable thanks to the increase of electricity produced by renewable energies and the cost reduction of electrolyzers [23].

A green hydrogen production system is integrated with different systems, from producing electricity to storing the produced hydrogen. These systems can be grouped by its energy carrier plus water systems:

- Electricity: components for the production, storage, and transmission of electricity.
- Heat: components for the production, storage, and transmission of heat.
- Hydrogen: systems for the production, storage, and transport of hydrogen.
- Water: components for the provision of clean water.

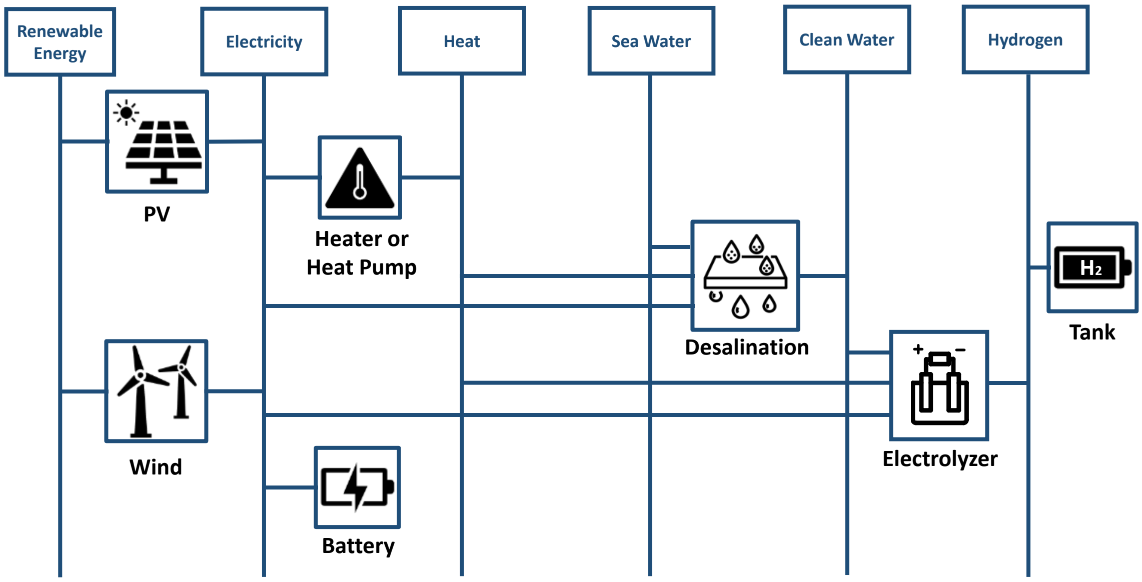


Figure 2-2 Components in an electrolysis-based hydrogen production system

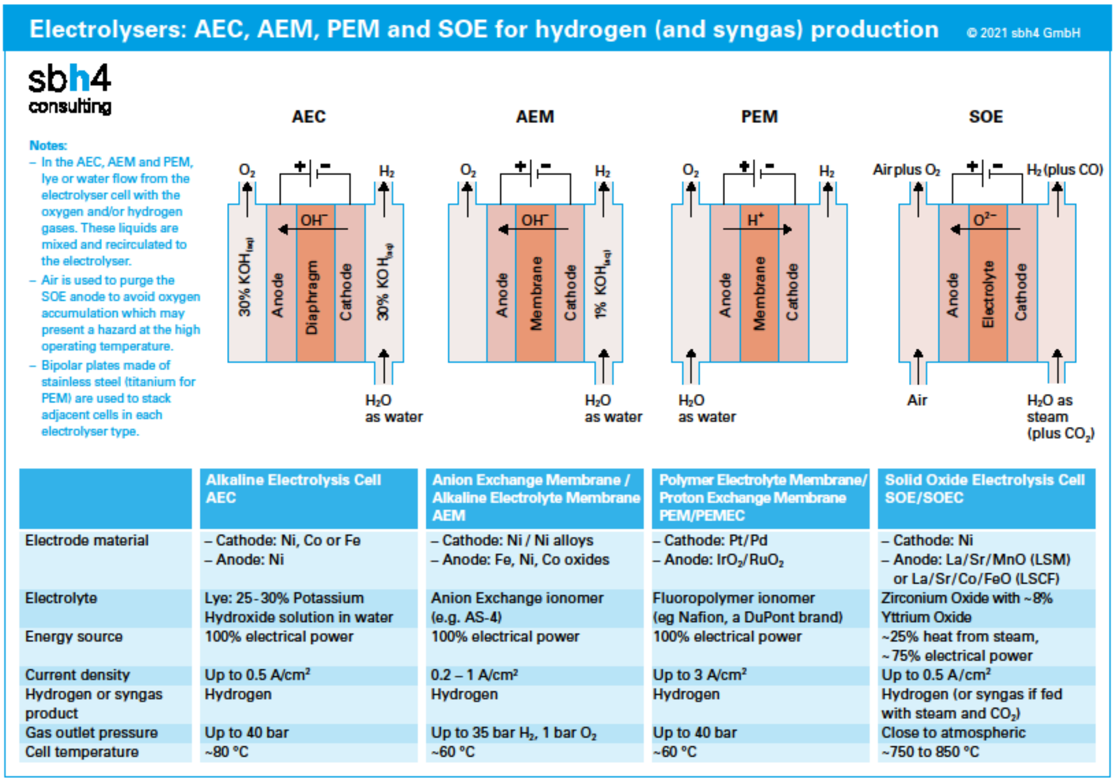
Figure 2-2 shows the main systems in the production of electrolysis-based hydrogen, needing electricity, water, and heat. There are more alternatives to produce electricity or heat, coming from other renewable energy systems. In addition, the desalination system is only needed in case no fresh water is available or it is scarce and only suitable in locations close to the sea or with a saltwater source.

2.2.2 Types of electrolyzers

The most important component in this system is the electrolyzer, where the hydrogen is produced. While all kind of electrolyzers need water and electricity, the amount of heat needed depends on the type of electrolyzer and its operation. Even some electrolyzers may be operated without adding any additional heat [24]. The four main existing hydrogen electrolyzers are:

- Alkaline Electrolyzers (AEL)
- Proton Exchange Membrane Electrolyzer (PEM)
- Solid Oxide Electrolyzer Cell (SOEC)
- Anion Exchange Electrolyzer (AEM)

These electrolyzers have different construction approaches, operational processes, efficiencies, and maturity levels. The electrolyzers technologies are constantly evolving at different levels (stack or cell level and whole system level), aiming to increase the systems' efficiencies and scalabilities, while optimizing the amount of materials used and extend their lifetime [23]. Figure 2-3 shows the main technical specifications for the four electrolyzers mentioned above.



	Alkaline Electrolysis Cell AEC	Anion Exchange Membrane / Alkaline Electrolyte Membrane AEM	Polymer Electrolyte Membrane/ Proton Exchange Membrane PEM/PEMEC	Solid Oxide Electrolysis Cell SOE/SOEC
Electrode material	– Cathode: Ni, Co or Fe – Anode: Ni	– Cathode: Ni / Ni alloys – Anode: Fe, Ni, Co oxides	– Cathode: Pt/Pd – Anode: IrO_2/RuO_2	– Cathode: Ni – Anode: La/Sr/MnO (LSM) or La/Sr/Co/FeO (LSCF)
Electrolyte	Lye: 25-30% Potassium Hydroxide solution in water	Anion Exchange ionomer (e.g. AS-4)	Fluoropolymer ionomer (eg Nafion, a DuPont brand)	Zirconium Oxide with ~8% Yttrium Oxide
Energy source	100% electrical power	100% electrical power	100% electrical power	~25% heat from steam, ~75% electrical power
Current density	Up to 0.5 A/cm²	0.2 – 1 A/cm²	Up to 3 A/cm²	Up to 0.5 A/cm²
Hydrogen or syngas product	Hydrogen	Hydrogen	Hydrogen	Hydrogen (or syngas if fed with steam and CO_2)
Gas outlet pressure	Up to 40 bar	Up to 35 bar H_2, 1 bar O_2	Up to 40 bar	Close to atmospheric
Cell temperature	~80 °C	~60 °C	~60 °C	~750 to 850 °C

Figure 2-3 AEC, AEM, PEM and SOEC electrolyzers main characteristics [24]

A compressor is also needed to facilitate the storage of the produced hydrogen, given its low volumetric density. Therefore, after the electrolysis of hydrogen, a compressor should be installed. Also, the electrolyzer itself can electrochemically compress the hydrogen, which is more energetically efficient than compressing the hydrogen in conventional ways, that is, with a mechanical compressor [25]. The hydrogen can also be stored in other physical states or transformed into other energy carriers.

2.2.3 Storage of hydrogen

Besides the electrolysis-based hydrogen production, storing the produced hydrogen is also a key challenge for the further development of the hydrogen economy. Here are presented the main methods for the storage of hydrogen [26]:

- **Physical-based storage of hydrogen**
  - Compressed gas: great volumes needed
  - Liquified: losses due to boil-off of the hydrogen
  - Cryo-compressed: takes advantages from the two previous storage methods
- **Adsorption-based storage of hydrogen**
- **Chemical-based storage of hydrogen**

2.2.4 Synthetic fuels and ammonia as alternative energy carriers

An alternative to hydrogen storage is to transform the produced hydrogen in other energy carriers, such as synthetic fuels and ammonia [27].

- **Synthetic fuels:** use the Fischer-Tropsch process with a green carbon source to transform the hydrogen in fuels for aviation, methane, or other hydrocarbons.
- **Ammonia:** use the Harber process to transform the hydrogen into ammonia, which can be used as fuel for shipping and plays a key role in the production of fertilizers.

The transformation of hydrogen into synthetic fuels and ammonia could stimulate the development of the hydrogen economy, contribute to the energy transition with carbon neutral fuels and reduce the greenhouse emissions in specific industries such as the ammonia industry [27, 28].

2.3 Modelling of energy systems

The energy system models analyze the processes involving the production, transport, or supply of energy, among others. The models consider different scopes, spatial resolution and temporal resolution, and scope. These models can be a small part of a sub-sector in a specific region, such as gas supply in Germany. They can also comprise the combination of sectors at an international level, for example, the production, demand, and consumption of electricity from all kinds of energy sources at the European level [29, 30].

For this reason, the goals of the models are wide-ranging. On the one hand, they are used to understand the dynamics of existing energy processes. On the other hand, they serve to investigate and explore the energy systems of the future with different scenarios, predict their evolution, and take decisions that will help get to the desired scenario. An example would be modelling the complete European energy system to predict the future demand of the different energy sources and what would be the next needed steps to reduce the dependency on imported oil and gas [29, 30].

The energy system models are simplifications of the reality. Simplification is needed to facilitate the computational performance of the model and to keep the results comprehensible. If the model is too complex, the correlations between the parameters and constraints are difficult to analyze. This is a common trade-off found not only in energy system models but in any kind of model. More complex models are less comprehensive but more realistic, and simpler models are easier to analyze but not close to reality.

The energy system models use simulation and or optimization methods to calculate the different scenarios. One of the most commonly used methods is linear optimization or linear programming [29].

2.3.1 Linear optimization or linear programming

The linear optimization, or linear programming method, consists of minimizing or maximizing an objective function, which represents a measurement for a real-world process or problem. This objective function is a linear equation composed by specific variables, called decision variables, whose value is to be determined. Only a certain combination of values for the decision variables will minimize or maximize the objective function [31, 32].

Additionally, the objective function is subject to a set of constraints that limit the possible values of the decision variables. These constraints are defined as extra equality and inequality linear equations of the decision variables [31, 32]. Non-negativity constraints are common in the different linear models, especially in energy system models; for example, the amount of electricity cannot be negative [32]. Moreover, the constraints should not contradict each other to keep the feasibility of the optimization problem [31].

The three elements of a linear optimization problem in standard form are [29]:

$$
\begin{array}{lll}
Objective\ Function & min\ or\ max\ f(x_j) = c_1x_1 + c_2x_2 + \cdots + c_nx_n & \\
Decision\ Variables & x_j\ , j = 1, 2, \ldots, n. & \\
Constraints & s.t. & x_1 \geq 0 \quad (non\ negativity) \\
 & & x_2 \geq 0 \quad (non\ negativity) \\
 & & a_1x_1 + a_2x_2 + \cdots + a_nx_n \begin{Bmatrix} \leq \\ = \\ \geq \end{Bmatrix} b
\end{array}
$$

The linear optimization problem can also be written in a matrix form [33]:

$$
\begin{array}{ll}
min\ or\ max\ f = C'X & \\
s.t. & AX \leq b \\
 & X \geq 0 \\
where &
\end{array}
$$

$$C = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

The simplex method is one of the most popular methods to solve linear optimization problems, invented in 1947 by G.B. Dantzig [31].

2.3.2 Non-linear optimization or non-linear programming

Sometimes non-linearities are needed to model certain systems. In that case, the problem is no longer linear, and it is called non-linear optimization problem or non-linear programming problem. There are different approaches to solve non-linear optimization problems, which can be found in the literature.

3 Existing Levelized Cost of Hydrogen (LCOH) studies

To understand the meaning of the levelized cost of hydrogen (LCOH), it is useful to understand the levelized cost of electricity (LCOE). The ISE Fraunhofer Institute defines the LCOE as the division of all lifetime investment and operation costs, CAPEX and OPEX respectively, by the total amount of net electricity generated, independently of whether it is a conventional or renewable power plant [4, 5]; see 2.1.1.

In the same way, the levelized cost of hydrogen (LCOH) is calculated through the division of all CAPEX and OPEX costs by the total amount of hydrogen produced throughout the system's life [6]; see 2.1.1.

The number of articles, models, and tools regarding hydrogen potential and its costs has increased in recent years due to the increasing interest in hydrogen as a key component for the energy transition. Due to the high literature amount, a deeper analysis regarding the most relevant optimization models of the levelized cost of hydrogen (LCOH) is carried out.

3.1 Overview of studies regarding levelized cost of hydrogen (LCOH)

An overview of recent articles and studies regarding the levelized cost of hydrogen (LCOH) might be helpful to know the existing considerations and trends in the field.

In

Table 3-1, column 'Key Word' classifies the different scientific contributions into 'Optimization' or 'Analysis'. While both scientific contributions might model the different elements of the hydrogen production system, only the 'Optimization' contributions consider an optimization, minimizing the LCOH. On the other hand, the 'Analysis' contributions might calculate the LCOH, but without optimizing the size of the hydrogen production system.

Table 3-1 Overview of articles and studies regarding levelized cost of hydrogen (LCOH)

Ref.	Scientific Contribution	Type	Year	Key Word	Topics
[34]	PtX Atlas Fraunhofer IEE	Tool	2021	Optimization PV, Wind	Optimization of LCOH and other synthetic fuels globally, considering techno- and socio-economic criteria Import costs.
[35]	EWI - Estimating Long-Term Global Supply Costs for Low-Carbon Hydrogen	Working Paper	2020	Optimization Analysis PV, Wind	Analysis of LCOH for three different production mechanism: from RES, from natural gas plus CCS and pyrolysis in 94 countries. For the RES case a linear optimization is considered.

Ref.	Scientific Contribution	Type	Year	Key Word	Topics
[36]	MDPI - Electrochemical Hydrogen Production Powered by PV/CSP Hybrid Power Plants: A Modelling Approach for Cost Optimal System Design	Article	2021	Optimization PV, Solar Tower, TES	Optimization of LCOH getting the optimum of six variables regarding the installed power of different components in three locations (Germany, Spain, and Morocco). AEL electrolyzer. Global Optimization Tool, MathWorks.
[37]	Methodology for multi-objective optimization of wind turbine/battery/electrolyzer system for decentralized clean hydrogen production	Article	2021	Optimization Wind, Battery	Optimization of multiple parameters with LCOH among them, using iterations and a self-developed script. AEL electrolyzer.
[38]	Hydrogen from renewables: Supply from North Africa to Central Europe as blend in existing pipelines – Potentials and costs	Article	2019	Optimization PV, Wind, Battery	Optimization of LCOH in the North Africa region. PEM electrolyzer.
[39]	Agora-AFRY: No-regret hydrogen: Charting early steps for H_2 infrastructure in Europe	Study	2021	Optimization PV, Wind	Optimization of LCOH in Europe, also considering hybrid configuration and transport costs.
[40]	AFRY's Global Hydrogen Trade Model	Slides	2022	Optimization PV, Wind	Optimization of demand, supply (LCOH) and transport of hydrogen
[41, 42]	HYPAT – Globaler H2 Potenzialatlas	Tool	2021	Optimization PV, Wind	Optimization of LCOH with minimization of costs, under development.
[43]	H2 Atlas Africa	Tool	2021	Analysis PV, Wind	Analysis of hydrogen potential and its LCOH for Africa. Until now only western Africa analyzed.
[44]	AusH2 - Australia's Hydrogen Opportunities Tool	Tool	2021	Analysis PV, Wind, CCS	Analysis of LCOH allowing the user to choose the parameters for the system.
[45]	HyDRA - Hydrogen Demand and Resource Analysis. NREL	Tool	2009	Analysis	Analysis of LCOH without considering PV and Wind costs. For USA. Also, an existing Worksheet 'H2A Hydrogen Production Model'.
[46]	IRENA – World Energy Transitions – Outlook 2022 1.5ºC Pathway	Study (5.3)	2022	Analysis PV, Wind	Analysis of LCOH in 2050 globally differentiating countries investment risk.
[47]	Lazard's Levelized Cost of Hydrogen Analysis v2.0	Report	2021	Analysis	Analysis of LCOH considering only hydrogen, no renewables technologies investment.
[48]	Techno-economic Analysis of Hydrogen Electrolysis from Off-Grid Stand-Alone Photovoltaics Incorporating Uncertainty Analysis	Article	2020	Analysis Monte Carlo PV	Analysis of LCOH with Monte Carlo simulations to identify key drivers of PV powered electrolysis. Optimize PV system.
[49]	Assessment of Hydrogen Production Costs from Electrolysis: United States and Europe	Report	2020	Analysis Monte Carlo PV, Wind	Analysis of LCOH with Monte Carlo simulations considering from a realistic point of view all costs for hydrogen production.
[50]	Levelized Cost of Hydrogen Calculation from Off-Grid Photovoltaic Plants Using Different Methods	Article	2021	Analysis PV	Analysis of LCOH for an existing PV power system with historical, and two different simulated power production data. Interesting ratio Electrolyzer Power/PV Power (oversize factor).
[51]	Levelized costs of energy and hydrogen of wind farms and concentrated photovoltaic thermal systems. A case study in Morocco	Article	2020	Analysis PV, CPV/T	Analysis of LCOH for a given Renewable power in Morocco optimizing the electrolysis capacity of AEL through numerical simulations in C++
[52]	Hydrogen production costs of a polymer electrolyte membrane electrolysis powered by a renewable hybrid system	Article	2021	Analysis CPV/T, Solar Tower	Analysis of LCOH for a given PEM electrolysis capacity in Morocco, optimizing the multiple solar (oversize factor).
[53]	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	Article	2020	Analysis PV, CSP, TES, Grid	Analysis of LCOH for Chile with real and simulated power data, for AEL and PEM. MATLAB/Simulink.

Ref.	Scientific Contribution	Type	Year	Key Word	Topics
[54]	Techno-economic feasibility evaluation of a standalone solar-powered alkaline water electrolyzer considering the influence of battery energy storage system: A Korean case study	Article	2021	Analysis PV, Battery	Analysis of LCOH potential with Monte Carlo simulation in Korea through the modelling of different components for AEL.
[55]	Country-specific cost projections for renewable hydrogen production through off-grid electricity systems	Article	2021	Analysis PV, Wind	Analysis of LCOH for European countries until 2050.
[56]	The Future of Hydrogen - IEA Seizing today's opportunities	Report	2019	Analysis PV, Wind	Analysis of LCOH globally. Figure 14 shows a map with distribution of LCOH considering hybrid of renewables.

3.2 Analysis of existing optimization models of levelized cost of hydrogen (LCOH)

The focus is to analyze deeper the eight optimization models, classified in
Table 3-1 as 'Optimization'. This analysis provides a perspective of the main considerations made in these models for the LOCH optimization.

The analyzed optimization models are:

1. PtX Atlas Fraunhofer IEE
2. Estimating Long-Term Global Supply Costs for Low-Carbon Hydrogen
3. Electrochemical Hydrogen Production Powered by PV/CSP Hybrid Power Plants: A Modelling Approach for Cost Optimal System Design
4. Hydrogen from renewables: Supply from North Africa to Central Europe as blend in existing pipelines – Potentials and costs
5. Methodology for multi-objective optimization of wind turbine/battery/electrolyzer system for decentralized clean hydrogen production
6. Agora-AFRY: No-regret hydrogen: Charting early steps for H_2 infrastructure in Europe
7. AFRY's Global Hydrogen Trade Model
8. HYPAT – Globaler H2 Potenzialatlas

The key aspects to be analyzed are:

- Organization
- End-Product of the model: Gas Hydrogen or other Synthetic Fuels based on Hydrogen
- Geographical scope
- Year of the model
- Modelling environment:

- Grid: On/off-grid
- Electrical Storage
- Renewable energy source to produce hydrogen
- Hybrid configuration PV and wind turbines
- Model criteria

Table 3-2 Matrix of LCOH optimization models for LCOH

Ref.	Model	Organization	Year	Final Product	Geographical Scope	Modelling Environment	Grid	Electrical Storage	Renewable Energy	Hybrid Solar/Wind (K)	Modelling criteria
[34]	**PtX Atlas** Fraunhofer	Fraunhofer IEE	2021	Fischer-Tropsch, CH3OH, SNG (l/g) H2(l/g)	Global. Maximum 30 areas per country	Optimization Model SCOPE ERA5 Data Open-Source: No	Off	Yes	Onshore Wind PV	Yes, Wind limiting factor	Socioeconomical criteria Cost optimization for different 12 technologies Desalination WACC constant SOEC, PEM
[35]	**EWI** - Estimating Long-Term Global Supply Costs for Low-Carbon Hydrogen	Institute of Energy Economics at the University of Cologne (EWI)	2020	H2 gas	94 countries on six continents (except Antarctica)	- Open Source: No	Off	No	Onshore Wind, Offshore Wind, PV	Yes	Linear optimization of RES case (RES, natural gas, pyrolysis), varying the RES to electrolyzer capacity ratio. WACC constant Low ang high temperature electrolyzer
[36]	**MDPI** -Electrochemical Hydrogen Production Powered by PV/CSP Hybrid Power Plants: A Modelling Approach for Cost Optimal System Design	Deutsches Zentrum für Luft- und Raumfahrt & TU Dresden	2021	H2 gas	3 locations in Germany, Spain, Morocco	Global Optimization Tool, MathWorks which is a Pattern Search algorithm Open Source: No	Off	No, thermal energy storage TES	PV, Solar Tower	No	Six optimization variables (power of elements) Price Level index (OECD) WACC constant AEL
[37]	Methodology for **multi-objective** optimization of wind turbine/battery/ electrolyzer system for decentralized clean hydrogen production	Centre de D´eveloppement des Energies Renouvelables (CDER)	2021	H2 Gas	Algeria	Self-developed script Open Source: No	On/ Off	Yes	Onshore Wind	No	Optimization of four objective functions with LCOH among them and for decision variables, using iterations. No WACC considered AEL
[38]	Hydrogen from renewables: Supply **from North Africa** to Central Europe as blend in existing pipelines – Potentials and costs	Hamburg University of Technology (TUHH)	2019	H2 Gas	Algeria, North Africa locations (close to pipelines)	- Open Source: No	Off	Yes (in off-grid)	Wind, PV	Yes, no K, both optimized.	Transport also considered WACC constant PEM

Ref.	Model	Organization	Year	Final Product	Geographical Scope	Modelling Environment	Grid	Electrical Storage	Renewable Energy	Hybrid Solar/Wind (K)	Modelling criteria
[39]	**No-regret hydrogen**: Charting early steps for H_2 infrastructure in Europe	Agora Energiewende and AFRY Management Consulting	2021	H2 gas	Europe	- Open Source: No	Off	No	Wind, PV	Yes, they consider in hybrid that 15% of wind not usable	'No-regret' hydrogen, the hydrogen demand of the industries with a high probability of hydrogen needs. Two scenarios, Blue-Green (SMRCCS + pyrolysis + RES), Fast-Green (only RES) They use hexagonal cells for the geographical division and the transport of the produced and stored hydrogen
[40]	AFRY's Global Hydrogen Trade Model	AFRY	2022	H2 Gas, Ammonia	Global	- Open Source: No	Off	No	Wind, PV	No	Linear optimization in Python, optimizing demand, supply, and transport simultaneously. It considers green and blue hydrogen.
[41, 42]	**HYPAT** – Globaler H2 Potenzialatlas (in progress)	Bundesministerium für Bildung und Forschung BMBF	-	H2 Gas, Synthetic products	Global	-	On/ Off	Yes (in off-grid)	Additional renewable sources	-	Water and environmental concern considered WACC constant

The eight optimization models are deeper analyzed.

1. **Ptx Atlas**

   The Fraunhofer ISE optimizes the LCOH for almost 600 locations globally. It uses the Optimization Model Scope considering 12 different technologies configurations for the cost minimizations. These technologies contain six different fuels combined with two different electrolysis systems. The main system components considered for the optimization are solar PV on open-field, onshore wind, battery storage, electrolyzer (PEM and SOEC), and hydrogen storage. Additionally, heat pumps, electric boiler, heat storage system, desalination, CO2 capture system among other systems are considered depending on the optimized technology option. The optimization model is not open-source, but the parameters data sources considered, are published [34].

2. **Estimating Long-Term Global Supply Costs for Low-Carbon Hydrogen**

   The Institute of Energy Economics at the University of Cologne (EWI) compares the cost of hydrogen production for three different production mechanisms (methane with CCS, pyrolysis, and electrolysis with renewable energy). This comparison is done for 94 countries in six continents. The third case is studied adopting a linear optimization model considering solar PV, onshore wind and offshore wind as renewable energy technologies and low-temperature and high-temperature electrolyzers. The decision variable for this model is the ratio RES-to-electrolyzer, noting that higher capacity factors of the renewable energy sources do not necessarily mean a higher capacity ratio of the electrolyzer [35].

   Additionally, it considers a Learning Rate for the CAPEX costs, so that if the capacity doubles in comparison with a reference capacity installed, the costs reduce a certain percentage, due to the modelling of the learning curve [35].

   The cost optimization CAPEX and OPEX of the electrolyzer and the renewable energy systems for each country, year and technology considered. Also, a specific hydrogen demand is considered as a parameter for each country and year. The optimization equations and constraints are published [35].

3. **Electrochemical Hydrogen Production Powered by PV/CSP Hybrid Power Plants: A Modelling Approach for Cost Optimal System Design:**

   The Institute of Future Fuels in Cologne and the TU Dresden have developed an optimization cost model for a system based on an alkaline electrolyzer (AEL) supplied by photovoltaic, concentrated solar power (CSP) and thermal energy storage. Other components of the off-grid system are a heater and a turbine. The installed powers for the six mentioned components represent the optimization variables of the system. The system is optimized for three different locations in Germany, Spain and Morocco and two different scenarios for the PV and CSP costs, which are current technology costs and technology costs in 10 years [36].

Two assumptions concerning the modelling of the electrolyzer are. First the electrolyzer can only work between 20% and 100% of the nominal power. Secondly, the standby mode of 1% nominal power is considered to avoid cold start of the electrolyzer. A constant interest rate is considered for the calculation of the annualized costs, but no country risk premiums are considered [36].

Nr.	Optimization Variable	Description	Global Optimization Variable Constraints
1	$P_{CSP,Rec}$	CSP molten salt receiver nominal input power (at DNI 900 W/m^2)	0 or: 800 $\leq P_{CSP,Rec} \leq$ 1200 MW
2	$P_{PV,Peak}$	PV peak power (at GHI 1000 W/m^2)	$\leq$1000 MW
3	P_{AEL}	Nominal power of alkaline electrolyser system	$\leq$1000 MW
4	P_{Turb}	Nominal power of steam cycle turbine	0 or: $\geq$50 MW
5	C_{TES}	Capacity of molten salt thermal energy storage	$\leq$8000 MWh
6	$P_{Heater,el}$	Nominal power of electric molten salt heater	$\leq$1000 MW

Figure 3-1 Example of optimization variables of the electrolyzer system and its constraints [36]

4. Methodology for multi-objective optimization of wind turbine/battery/ electrolyzer system for decentralized clean hydrogen production:

The Centre of Renewable Energies Development (CDER) in Algeria proposes a system that produces hydrogen gas, and includes wind turbine, electrolyzer, battery, power converters, and hydrogen tank. The approach is to optimize four objective functions simultaneously with an iterative optimization algorithm to find Pareto-Optimal solutions [37].

- Total Hydrogen Deficit: minimize the demand not covered
- Energy Dump Possibility: minimize the not used produced energy
- Levelized Cost of Hydrogen: minimize
- CO2 emissions avoided: maximize

Some interesting components considered in the model are a DC/DC power converter to protect battery against overcharge and over discharge, as well as an AC/DC rectifier to convert to DC the energy produced at the wind turbine. The battery modelling is detailed and two optimization strategies depending on the wind load are elaborated for the management of the battery system [37].

Additionally, noticeable constraints for the electrolyzer are considered. The input power must be between 20% and 100%, also the Electrolyzer Operation Time (EOT) must be above a minimum limit, and the Start-ups per day (Start) which sets a limit for the number of start-ups within a day. The hydrogen tank is also subjected to interesting constraints [37].

An annual hydrogen demand of 8760 kg/a is set, representing 1kg/h. Nevertheless, meeting the demand is monthly flexible, according to a specific time series [37].

5. **Hydrogen from renewables: Supply from North Africa to Central Europe as blend in existing pipelines – Potentials and costs:**

The Technical University of Hamburg developed a model to minimize the total costs of a green hydrogen system in Algeria. Afterwards the LCOH will be calculated. The components considered are PV, onshore wind, battery storage, electrolyzer and electricity transmission line. The transport of the produced hydrogen to Europe through existing pipelines is also included and therefore the transport costs are included.

The hydrogen demand considered for the optimization is arbitrary set to 1GWh/A.

The study also calculates the technical production potential of green hydrogen by the exclusion of high populated and big slope areas.

Additionally, a room for improvement chapter for the model is presented with different possible future assumptions. Some examples are the reduction of electrolyzer efficiency and other technologies as they age, the possible consideration of electrolyzer overload to take advantage of electricity peaks, the sale of oxygen or the consideration of social and ecological criteria [37].

6. **No-regret hydrogen: Charting early steps for H_2 infrastructure in Europe**

Agora Energiewende together with AFRY Management Consulting have developed an optimization model, called 'Hexamodel' for the calculation of the LCOH in Europe, dividing the continent in hexagon cells [39].

Only the no-regret demand of hydrogen is considered for the optimization, which means that only the demand for the industrial sectors that are certain to be supplied with hydrogen are considered. Future unreliable demand expectations of hydrogen are excluded [39].

For the optimization model two scenarios are considered. First the Blue-Green (SMRCSS + Pyrolysis + RES) and second the Fast-Green (only RES). Additionally, hybrid PV-Wind configurations are regarded with a simple constraint, which is an overlapping factor of 15%, meaning that in case of a hybrid setup, 15% of the wind potential will be excluded. In this model also hydrogen storage and transport are considered [39] but separated from the optimization of the hydrogen production costs.

7. **AFRY's Global Hydrogen Trade Model**

   AFRY has developed an optimization model integrating demand, supply and transport globally using linear optimization. In the supply model, green and blue hydrogen are considered. For green energies PV and Wind are integrated, but no hybrid setup is considered. In addition, no battery or hydrogen tank are included in the optimization model. Other important assumptions are the non-flexibility of the demand [40].

   Worth mentioning is the integration of the demand, supply and transport at a global scale, resulting in interesting outputs concerning the estimated distribution of hydrogen production worldwide and the import-export relationships between countries [40].

8. **HYPAT – Globaler H2 Potenzialatlas**

   The Federal Ministry of Education and Research of Germany is currently developing an optimization model for hydrogen production globally. In the model, wind PV and additional renewable energies are being considered, as well as Direct Air Capture and specific interest rates for the different countries. Also, an potential area analysis, especially with the exclusion of water scarce areas, will be considered [41, 42].

   The project is still under development. Therefore, many details such as PV-wind hybrid setup, battery or hydrogen tank are still to be published.

This academic work research questions focus on the effects of hybrid PV-wind energy configuration, the introduction of a stationary battery system in the system, and the consideration of the Country Risk Premiums. These components or assumptions are further studied for the eight models to see their effect on the sizing and behavior of the hydrogen production system. Additionally, the hydrogen tank represents an interesting component due to its effect on the system’s performance. The modelled components and assumptions for each of the eight analyzed optimization models are summarized in Table 3-3.

Table 3-3 Modelled components in each analyzed LOCH optimization model [34–42]

Model	Optim.	Modelled components											Country dependent costs (CAPEX/CRP)
		Electrolyzer	Wind	PV	Hybrid (PV-wind)	Battery	H2 Tank	Desalination	Heater	CSP	TES	Other: DAC/ET/PW /Turbine	
PtX Atlas Fraunhofer	Linear	x	x	x	x	x	x	x	x		x	x	
EWI - Estimating Long-Term Global Supply Costs for Low-Carbon Hydrogen	Linear	x	x	x	x								x
MDPI - Electrochemical Hydrogen Production Powered by PV/CSP Hybrid Power Plants: A Modelling Approach	Non-linear	x		x					x	x	x	x	
Methodology for **multi-objective** optimization of wind turbine/battery/ electrolyzer system	Non-linear	x	x			x	x					x	
Hydrogen from renewables: Supply **from North Africa** to Central Europe	Linear	x	x	x	x	x						x	
AFRY's Global Hydrogen Trade Model	Linear	x	x	x									
No Regret Hydrogen Agora-Afry	Linear ?	x	x	x	x								
HYPAT – Globaler H2 Potenzialatlas *(in progress)*	Linear ?	x	x	x	?	?	?	x					x
FfE	Linear	x	x	x	x	x	x	x	x				?

Looking at the eight analyzed optimization models (one of them still being developed), only four consider hybrid wind and PV as energy source for the electrolysis of hydrogen. Three of the eight models consider hydrogen storage in tanks, and also three models include electrical battery storage. Only the PtX Atlas optimization model from Fraunhofer considers hybrid wind-PV, battery storage, and hydrogen storage simultaneously. Also, the PtX Atlas is not limited to one location but optimizes different locations worldwide [34].

Regarding the Country Risk Premiums, only the in-progress optimization model HYPAT might consider them in their calculations [42]. The optimization model (EWI) considers country-dependent CAPEX costs [35], which is a similar approach to the Country Risk Premiums.

Table 3-3. shows also the components planned to be included in the model developed in this academic work. It intends to incorporate hybrid PV-wind, battery, hydrogen storage tank, and Country Risk Premiums. In addition, heater and desalination systems supporting the electrolysis are included as possibility.

The results and conclusions of the five optimization models that consider at least one of the three following components (hybrid PV-wind, battery, or hydrogen tank) are further analyzed. See Table 3-4.

Table 3-4 Optimization model conclusions analysis for the hybrid, battery and hydrogen storage components [34, 35, 37–39]

Model	Hybrid (PV-Wind)	Battery	H2 Storage
PtX Atlas Fraunhofer	Locations with hybrid solar-wind systems are frequently found in the optimization results. Not all locations with solar and wind energy potential will introduce hybrid systems.	For H_2 gas no battery is introduced in the optimization results. (Other energy carriers introduce storage)	For H_2 gas no tank is introduced in the optimization results. (Other energy carriers introduce storage)
EWI - Estimating Long-Term Global Supply Costs for Low-Carbon Hydrogen	Hybrid system results in lower LCOH, only when wind and solar potentials are simultaneously high and even then, with low reduction of costs.	N/A	N/A
Methodology for **multi-objective** optimization of wind turbine/battery/ electrolyzer system for decentralized clean hydrogen production ***Note: location in Algeria***	N/A	Different cases of study. The optimization results usually introduce of battery storage, supporting the hydrogen production. However, this model also minimizes electricity curtailment (energy dump). In conclusions mentioned, that if energy dump objective is ignored, the LCOH decreases (due to battery)	Hydrogen tank is introduced in all the optimization results of the case studies
Hydrogen from renewables: Supply **from North Africa** to Central Europe as blend in existing pipelines – Potentials and costs ***Note: location in Algeria***	Two cases: upper cost and a lower cost. Only one of the cases result in a hybrid system, being the introduced share quite small.	Even halving the CAPEX of the battery system, the optimization results do not introduce battery into the system.	N/A
No Regret Hydrogen Agora-Afry	Locations with hybrid PV-wind systems are frequently found in the optimization results.	N/A	Pressurized tanks and salt caverns are considered, having the salt caverns lower CAPEX but limited by location. In conclusions stated that hydrogen storage provides security to cover the demand.

Table 3-4 presents the results and conclusions for the optimization models that integrate at least of the components, hybrid PV-wind simultaneously, battery storage, and hydrogen storage system. The color represents a qualitative assessment of the existing optimization models, characterizing how integrating the different components affects the Levelized Cost of Hydrogen (LCOH).

In the case of the integration of hybrid PV-wind energy, the different optimization models conclude in a neutral or positive assessment. According to the model results, the hybrid configuration will be profitable only in optimal simultaneity of solar and wind conditions.

Regarding the hydrogen tank, normally, the assessment of the different models is neutral or positive, which means that introducing a hydrogen tank generates no change or a small decrease in the LCOH. Here is also important to differentiate between hydrogen tank and cavern storage, being the last case restricted to existing locations but with lower costs. This academic work considers only the hydrogen tank in the model.

Finally, the assessment of the battery storage system in the models is generally negative due to its high investment costs, making electricity curtailment a more profitable option.

This analysis can delineate a point of reference to which the results of the model developed in this academic work can be compared.

4 Methodology and data

The following chapter focuses on the methodology and data used to develop the electrolysis-based hydrogen production model. First, the electrolysis-based hydrogen production system to be modelled is delimited. Then, the input data is identified, and an appropriate database structure is created. Furthermore, the code program is written, and the desired output data and calculation scenarios are defined.

4.1 Definition of the hydrogen production system

The electrolysis-based hydrogen production system modelled is a simplification of the complete system described in 2.2. The following diagram shows the components included in the system to be modelled.

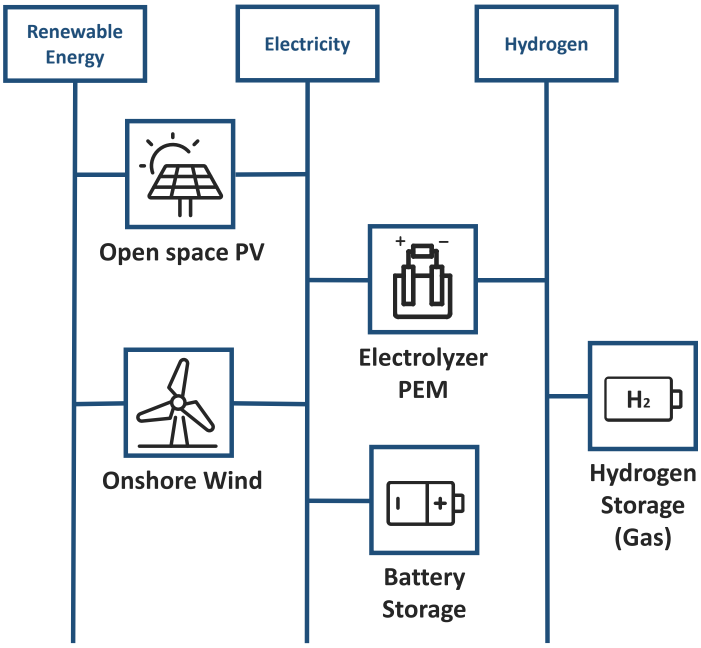


Figure 4-1 Simplified electrolysis-based hydrogen production system

Only open space photovoltaic and onshore wind are included as renewable energy sources for electricity production. Although offshore wind will probably entail a key technology for electrolysis-based hydrogen production, it is not considered in this model.

Modelling the desalination system represents a great interest in the case of coastal regions with water scarcity, but this would have required a certain amount of pre-analysis that can be done in future improvements of the model.

On the other hand, while modelling a heater involves just a high efficiency component for transforming electricity into heat, the amount of heat needed depends on the type of electrolyzer. Some electrolyzers can work with external heat at their reversal voltage (endothermic reaction) or without external heat when they work at its thermoneutral voltage.

In the case of proton exchange electrolyzers (PEM), as in this model, their standard operation requires overpotentials to overcome irreversible losses. These overpotentials entail an exothermic operation [57]; therefore, no extra heat is needed. Also, the viability of recovering the produced waste heat for other applications is being studied [58]. For this reason, modelling a heater or a heat pump is excluded from the model.

A stationary battery system is also modelled in the system. Finally, a hydrogen storage system is included in the model in the form of a tubular accumulator.

4.1.1 Main objective of the model

The above-mentioned electrolysis-based hydrogen production system is modelled, and its size is optimized for several locations and a specific hydrogen demand using linear programming. Therefore, the optimization results include the sizes of the components needed to meet the hydrogen demand (1 kg of hydrogen per hour) while producing the hydrogen at minimum cost.

4.1.2 Assumptions and constraints of the model

Assumptions regarding the number of locations (MERRA-2)

The system size will be optimized for 51677 locations in total. These locations are cells extracted from the MERRA-2 datasets from the NASA Global Modeling and Assimilation Office. MERRA-2 stands for Modern-Era Retrospective Analysis for Research and Application, Version 2. This project has divided the whole world, all land, and sea area, into a grid of 207936 cells, each with an approximate size of 50x50 km [59]. Large amounts of meteorological data have been provided for each of these cells since 1980.

Only inland cells are considered for the development of the model, excluding the Antarctic and North Pole (above latitude .75º 45’). In total, 51677 cells are considered, and the same number of optimizations are carried out. Figure 4-2 shows the world map with the chosen cells to be calculated.

The land use potential analysis regarding environmental protected areas, water scarcity, high-density population, and high steep regions, among others, is excluded for the development of the model. Therefore, no extra exclusion criteria for the cells are included, and no potential analysis for producing hydrogen is done. That means that the results are not meant to tell how much hydrogen can be produced in each cell but to tell how much the hydrogen production would cost if no potential limit is set.

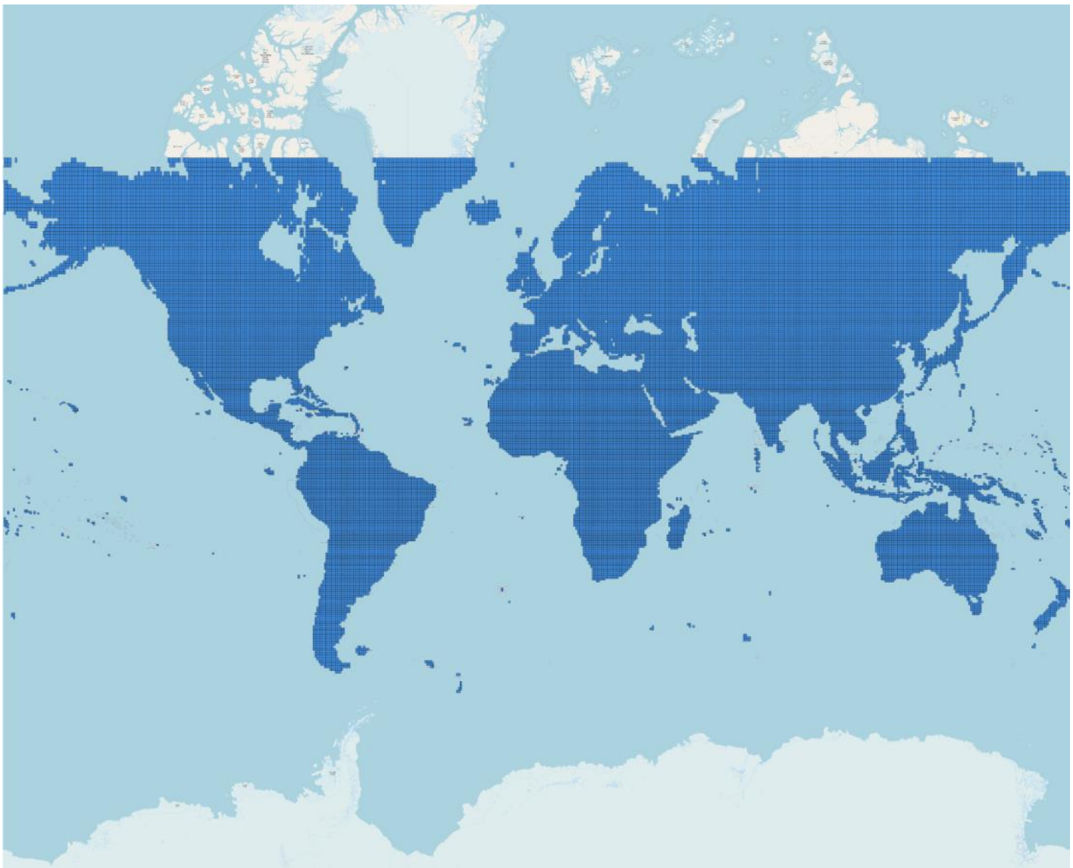


Figure 4-2 World map with MERRA-2 cells division to include in the model

Assumptions regarding hydrogen production system

- Off grid system.
- Components considered in the model as from Figure 4-1:
    - Open space PV
    - Onshore very weak wind turbine (explained in 4.3.2)
    - Onshore very strong wind turbine (explained in 4.3.2)
    - PEM Electrolyzer: the electrolyzer can work between 0% and 100% of nominal power, overloads are excluded. The ramp up of the PEM electrolyzer, either from cold start or warm start, is considered to get to nominal power in a few seconds [60]. The stack exchange after a certain number of operation hours is not included in the model.
    - Hydrogen storage system (tubular accumulator). This storage system is suitable for hydrogen gas, and it is considered as a buffer storage and not

intended for the long-term storage. Nevertheless, its capacity is not limited in the model.

  - Stationary battery storage.

- Hydrogen demand of 1kg$_{H_2}$/h
- Exclusion of components such as offshore wind turbines, compressor, desalination system, heater or heat pump, water costs, and electricity transmission system.

Assumptions regarding general optimization information

- Investment and operation costs, as well as other technical parameters of the components, for the year 2020.
- Weather data of solar irradiation and wind speed for the year 2012 from MERRA-2 database.
- Hourly time resolution, for 2012 equals 8784 h

4.1.3 Linearity of the model's objective function

The size of the system is optimized using linear optimization or linear programming. The objective function to be minimized, is the yearly costs of the system, which is a linear equation. The decision variables are the sizes of the systems to be modelled in the electrolysis-based hydrogen system; see Figure 4-1. Finally, the constraints specified for each component are defined by the employed open-source software that will formulate and solve the optimization problem according to the given input.

The defined demand for the model is 1kg of hydrogen per hour. The bigger the demand, the bigger the size of the optimized system, and consequently higher total costs. However, the LCOH is independent of the considered demand as nor of the following potential limiting constraints are considered in the model:

- No constraints regarding the maximum component size are considered.
- No constraints concerning the potential area analysis are included in the model.

As shown in Figure 4-3Figure 4-4, the total system costs against hydrogen demand have a constant slope until a potential limit is found, which is not the case in the model. For this reason, the LCOH is independent of the considered demand.

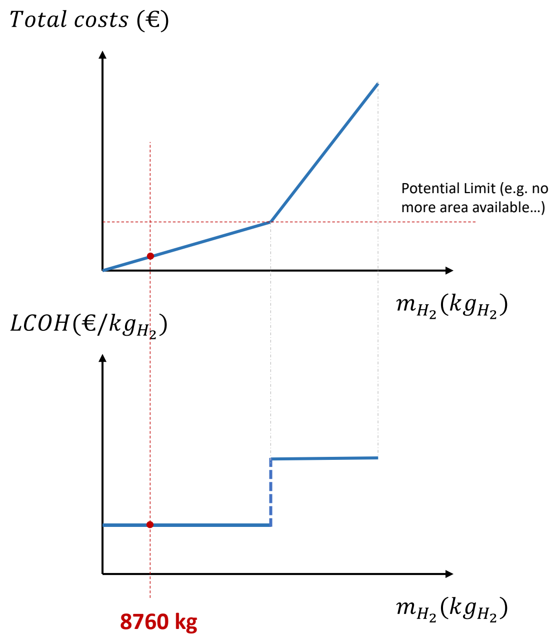


Figure 4-3 Linearity of system costs in the model due to absence of potential limit

4.2 Qualitative assessment of Open-Source software

An Open-Source software is needed to model the electrolysis-based hydrogen production system shown in Figure 4-1. Different open-source environments are evaluated using a qualitative assessment.

4.2.1 Criteria for the qualitative assessment

The assessment of the existing Open-Source linear optimization python-based environments is entirely qualitative and is implemented considering the following criteria:

1. **General criteria:** is it python-based software? Are the software frequently updated and new functions added? Is there an existing community? Are there examples?
2. **Suitable:** can the electrolysis-based hydrogen production system be modelled with the software? Can the components and constraints be introduced? Is the output data meaningful?
3. **Automatable:** given the amount of data, can the optimization be automated for all the locations? Are the input and output files easily created, saved, and loaded? Is the software compatible with tools and solvers that allow the reduction of the optimization time?
4. **Flexible:** if the model is improved in the future, will the system be easy to modify? (Introduce or remove components). Or should the system be built from scratch?

4.2.2 Open-Source software to be assessed

A wide variety of Open-Source software for linear optimization of power systems can be easily found, PyPSA, urbs, PSAT, PYPOWER or DIETERpy, among others. The model is developed in python and a preassessment is conducted to decide which software are firstly excluded and which should be further assessed.

Table 4-1 Preassessment of different open-source software [61–64]

Software	General	Suitable	Automatable	Flexible	Preassessment
PyPSA	✓	✓	✓	✓	Further assessment
urbs	✓	✓	✓	✓	Further assessment
PyPOWER	✓	✓	✗	✗	Excluded
DIETERpy	✗	-	✗	-	Excluded
PSAT	✗	-	-	-	Excluded

- **PyPSA**: further assessment.
- **urbs**: further assessment.
- **PyPOWER**: excluded. Input data must be written in a .py script, which forces the model to be hard coded and therefore uncomfortable to modify [64].
- **DIETERpy**: excluded. Although the software may be interesting, the information provided and the input data structure are not intuitive and few examples are given [63].
- **PSAT**: excluded. For MATLAB instead of Python

Two Open-Source software will be further evaluated, PyPSA, developed by the Technical University of Berlin and urbs, developed by the Technical University of Munich.

4.2.3 PyPSA: Python for Power System Analysis

PyPSA was developed by the Department of Digital Transformation in Energy Systems at the Technical University of Berlin and is used for minimizing cost optimization of power flow systems. It can also be used for modelling other systems (sector coupling) given its adaptability [61].

In PyPSA the system is called a network which contains the different components. There are three fundamental components, bus (node), link (process) and store, which are used to create the other predefined components of the system [61].

Our simple hydrogen production system would be composed by:

- <u>2 buses:</u> Electricity and Hydrogen

- 3 generators: open space PV and onshore very weak wind and onshore very strong wind
- 1 links: PEM Electrolyzer
- 2 storage units: battery storage and hydrogen storage tank
- 1 load: demand

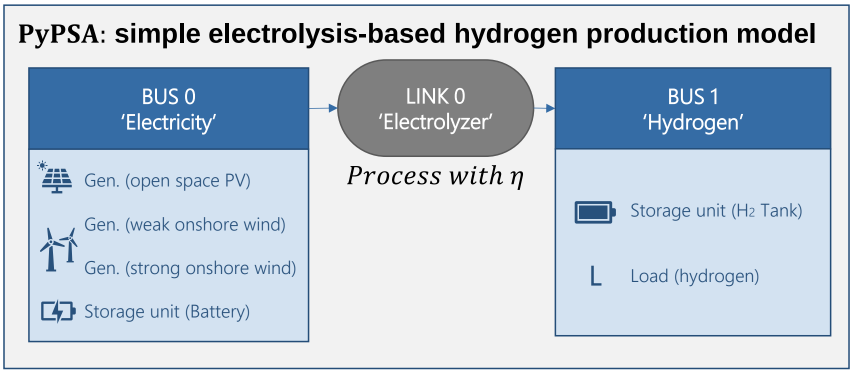


Figure 4-4 Simple PyPSA schema for the hydrogen production system

4.2.4 Urbs: A linear optimization model for distributed energy systems

Urbs was developed by the Chair of Renewable and sustainable Energy Systems at the Technical University of Munich. It is mainly used for the optimal sizing of energy systems, not being restricted to only power energy systems. For that, different commodities can be defined, such as electricity, gas, heat, and hydrogen [62].

A series of model entities allow the model of the energy system. In the case of a simple hydrogen production system, the entities would be:

- 3 commodities:
    - 3 suplm: PV solar and very weak and very strong wind timeseries
    - 1 demand: hydrogen demand
- 4 process:
    - PV panel (solar in, electricity out)
    - Very weak wind turbine (wind in, electricity out)
    - Very strong wind turbine (wind in, electricity out)
    - PEM Electrolyzer (electricity in, hydrogen out)
- 2 storage: Battery storage and hydrogen storage tank

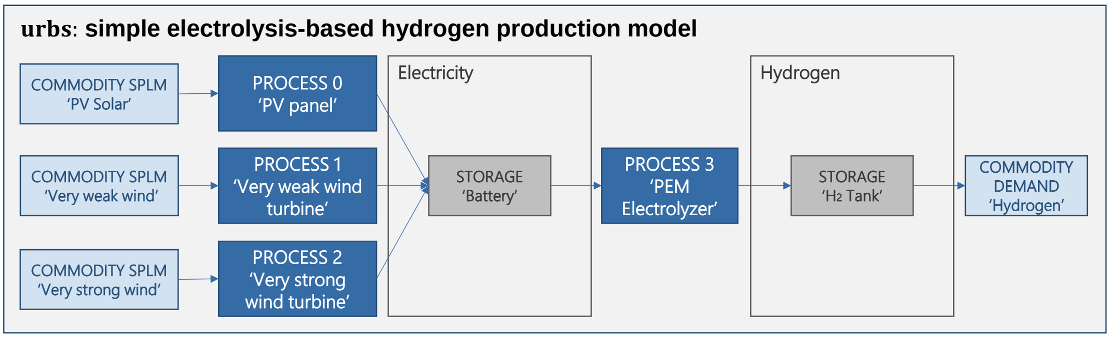


Figure 4-5 Simple urbs schema for the hydrogen production system

4.2.5 PyPSA and urbs comparison

A deeper assessment of PyPSA and urbs open-source software is required to choose the most appropriate one for developing the electrolysis-based hydrogen production model.

Below a comparison table between PyPSA und urbs for the above specified criteria (general, suitability, automatability and flexibility criteria) is presented with the following qualitative assessment:

- Green points: positive rating
- Grey points: neutral rating
- Yellow points: negative rating
- Red points: exclusion criteria. No exclusion points were identified.

Table 4-2 Comparison assessment for the open-source software PyPSA and urbs [61, 62]

	PyPSA	urbs
General	Python Package	Python Environment
	Frequent updates (interesting for Research Institut FfE)	Rare updates
	Community with fast response. Community in Google Groups and GitHub Issues	GitHub issues
	Examples available	Examples available
	Less intuitive	More intuitive
Suitability	System components can be modelled	System components can be modelled
	WACC not directly included	WACC included
	OPEX not directly included	OPEX included
	DSM possible	DSM possible and intuitive
	Shadow prices as output	No shadow prices as output
Automatability	Input: components as csv or DataFrames	Input: components as .xslx
	Input: Fast to modify DataFrame directly	Input: Slower to modify .xlsx directly
	Optimization formulating either with pyomo or without pyomo (faster)	Optimization formulating only with pyomo
	Solvers: GLPK, CPLEX, Gurobi	Solvers: GLPK, CPLEX, Gurobi
Flexibility	Multiple inputs/outputs in links must be defined in advance in code	Multiple inputs/outputs in processes easily defined
	System components can be easily added	System components can be easily added

Although both open-source software are suitable for the implementation of the model and there are no key aspects to exclude one software against the other, PyPSA was selected. This decision was made taken into consideration three arguments in favor of PyPSA which are:

- **PyPSA allows the construction of the optimization model both using pyomo and without using pyomo.** Not using pyomo is stated to be faster since 51677 optimizations must be conducted. Therefore, a faster building of the optimization model represents an advantage for the specific application [61, 62].
- **PyPSA allows the input of the model to be panda DataFrames.** Working with DataFrames may represent an advantage over working with .xlsx files for the model's input (urbs). The DataFrames are usually more efficient in data storage and speed. Especially when iterating, reading DataFrames should be faster than reading .xlsx files to build each one of the 51677 optimization models [61, 62].
- **PyPSA has a greater community to answer questions and the software is frequently updated.** The frequent update of the software ensures the long-term usability of the model and the possibility of adding new functions in the future. From a research institute perspective, as the FfE, this is a relevant aspect to develop the model further.

Therefore, PyPSA was selected as the appropriate open-source tool for the specific development of the electrolysis-based hydrogen production model. This decision answers the first research question: "Which Open-Source environment is appropriate to model the Levelized Cost of Hydrogen (LCOH) production worldwide?".

4.3 Input data

The electrolysis-based hydrogen production model needs different data as input. This data can be classified in three levels, system level, country level and cell level.

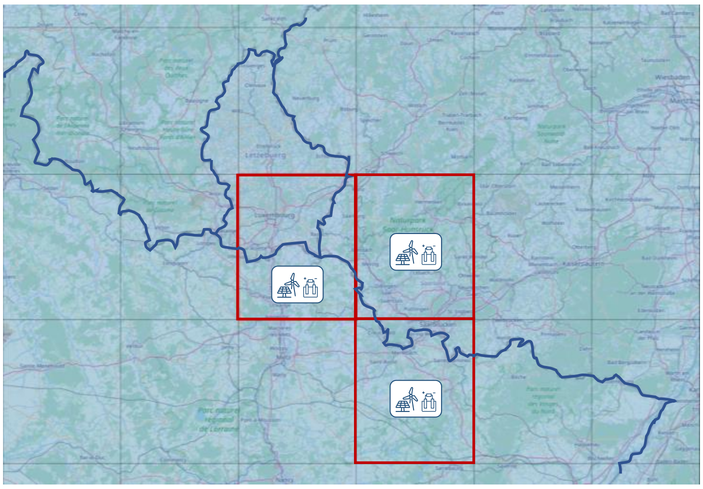


Figure 4-6 Visualization of country, cell, and system level for the data input

Figure 4-6 shows the three classification levels in the Luxembourg, France, German, and Belgium region. The land level is defined by the political borders of the countries, in blue. The cell level is defined by the division grid from MERRA-2 data, in red, and the system level is specific to the modelled electrolysis-based hydrogen production system. To explain the three levels, going from the biggest to the smallest level, first the country level, then the cell level and finally the system level.

Table 4-3 Overview of the input data levels

Data Level	Data
Country Level	Country Risk Premiums
Cell level	Geometrical and geographical MERRA-2 cell data
	PV capacity factors timeseries (derived from MERRA-2 solar irradiation data)
	Wind capacity factors (derived from MERRA-2 wind speed data)
System Level	Model parameters
	Technical parameters
	Economic parameters

4.3.1 Country level data

It is referred to all the data consistent inside one country. The country risk premiums data corresponds to this level, provided by the Stern School of Business at NYU [16] and can be found in the Appendix I: Country Risk Premiums (January 5, 2022). The country risk premium concept and calculation method is explained in 2.1.3. The last update of the country risk premiums and the one used for developing the model is from the January 5, 2022.

4.3.2 Cell level data

The electrolysis-based hydrogen system described in 4.1 is optimized in each of the 51677 cells. The grid data originates from the MERRA-2 database, explained in 4.1.2. The MERRA-2 database provides large amounts of meteorological data for each one of the cells in the grid.

For the development of the model the following MERRA-2 data is used at cell level:

- MERRA-2 cell geometry and geographical location to associate each cell with a single country.
- MERRA-2 cell meteorological data covering solar irradiation and wind speed in hourly resolution

The historical solar irradiation and the wind speed data from MERRA-2 represent interesting datasets for calculating the electricity generation from PV and wind turbines. However, these datasets cannot be used directly and must be transformed into capacity factors to be used directly in the model.

The capacity factors represent the ratio between the actual energy produced by a generator and the maximum energy that can be produced for the same time resolution [65].

$$Capacity\ Factor = \frac{Actual\ Energy\ Output\ [MWh]}{Nomilan\ Power\ [MW] \cdot Time\ Period\ [h]}$$

The most common time resolution is one hour, which results in hourly capacity factors. Looking at the meaning of the capacity factor, the produced energy by the generator for a specific hour can be directly calculated.

$$Output\ Energy\ [MWh] = Capacity\ Factor \cdot Nominal\ Power\ [MW] \cdot Time\ Period\ [h]$$

The FfE has already worked with the MERRA-2 datasets transforming the raw meteorological data into capacity factors with an hourly resolution for specific PV and wind turbines. This capacity factors data is available in the database of the FfE.

- **PV capacity factors available in FfE database**

The irradiation data from MERRA-2 has been used to calculate the hourly capacity factors for open-field PV with specific conditions. The selected open-field PV panels entail a performance factor of 0,804, and the capacity factors are available for different inclinations (10º, 20º, 30º, 40º and 45º) and 16 different cardinal orientations with 22,5º steps taking as 0º reference the south and 90º as west. Also, the capacity factors are calculated for four different weather years, 2012, 2015, 2017 and 2019

- **Wind turbines capacity factors timeseries available in FfE database**

The wind speeds from the MERRA-2 data source are transformed into capacity factors for ten different wind turbines, five offshore wind turbines and five onshore wind turbines. The calculated wind capacity factors already include the cut-in and cut-out speed as well as the corresponding efficiencies for each specific turbine. The capacity factors are calculated for four different weather years, 2012, 2015, 2017 and 2019

Table 4-4 Wind Turbines available for the calculated capacity factors

Ref.	Wind Turbine	On/ Offshore	Classification	Nominal power in MW	Rotor height in m	Rotor diameter in m	Power density in W/m^2
[66]	Siemens SWT-3.6-120	Offshore	Very weak	3,6	100	120	318
[67]	Vestas V164 - 8.0MW	Offshore	Weak	8	140	164	379
[67]	Vestas V164 - 8.0MW	Offshore	Middle	8	105	164	379
[68]	Siemens SWT-3.6-107	Offshore	Strong	3,6	100	107	400
[68]	Siemens SWT-3.6-107	Offshore	Very strong	3,6	80	107	400
[69]	Enercon E-115	Onshore	Very weak	3	140	115,7	285
[70]	Nordex N100	Onshore	Weak	2,5	120	100	318
[71]	Vestas V80 2.0MW	Onshore	Middle	2	100	80	398
[68]	Siemens SWT-3.6-107	Onshore	Strong	3,6	100	107	400
[69]	Enercon E-82 E3	Onshore	Very strong	3,02	80	82	572

For the development of the model, only the onshore wind turbines are considered. The “Classification” column in Table 4-4 follows an intern FfE site classification, being the power density the main classification parameter, followed by the second classification parameter, the rotor height. The wind turbines with the highest power density are classified with the “very strong” rating, which also means they can endure the highest wind speeds and gusts. Intuitively, the taller wind turbines are classified as “weak” or “very weak”, as they cannot endure high wind speeds for stability reasons. For the electrolysis-based hydrogen production system, the classified “very weak” and “weak” wind turbines represent interesting components as they are usually higher, which means more stable and constant wind speeds and, therefore, steadier higher full load hours. These constant wind speeds are advantageous, as they enable the steady operation of the electrolyzer, which is desired to cover the constant hydrogen demand.

4.3.3 System level data

The system level includes all the consistent data inside the electrolysis-based hydrogen production system. These data consider all the techno-economic parameters for the modelled components, as well as specific parameters of the model. Since a hydrogen production system is modelled in each cell, each system shares the same techno-economic and system specific parameters.

As mentioned in 4.1.2, the techno-economic data corresponds to the year 2020, dividing it into technical data and economic data. Below, the model parameters are explained, followed by the technical and economic parameter.

Model parameters data

The specific model parameters are defined in the excel and directly imported to PyPSA to create and calculate the right optimization model. The main model parameters are:

- **snapshots:** a total of 8784 timesteps or snapshots are considered as it is the number of hours that the weather year 2012, a leap year, has.
- **load.p_set:** in the electrolysis-based hydrogen production system, the load corresponds to the demand, and it is set to a specific value for all the timesteps (snapshots). As PyPSA works with power units (MW) [61], in the model the demand is defined as the equivalent power for one kilogram of hydrogen per hour. Knowing that the low heating value of hydrogen is 0,03333 MWh/kg [72]

$$load = 1\frac{kg}{h} \cdot 0{,}03333\frac{MWh}{kg} = 0{,}03333\ MW$$

  Therefore, the system must provide in each timestep 0,03333 MW to fill the demand.
- **p_nom_extendable:** all the components whose size must be optimized, must be defined with the parameter p_nom_extendable set to 'True', to allow the optimization to find the optimum size for each component [61].
- **cyclic_state_of_charge:** if set to True, then the initial state of charge, first timestep, and the final state of charge, last timestep, of the Storage Unit components (hydrogen storage and battery storage) must match as an additional restriction in the optimization model [61].

Technical parameters data

The technical data include technical parameters of the different components modelled in the electrolysis-based hydrogen production system. These parameters are:

- Lifetime

- Efficiency
- Charge Storage Efficiency
- Dispatch Storage Efficiency
- Self-Discharge losses
- Energy Power Ratio

Each one of the technical parameters are specific to one or various technologies. The following table shows the values used in the development of the model.

Table 4-5 Technical parameters of the electrolysis-based hydrogen production system

Ref.	Technology	Lifetime in years	Efficiency	Storage Efficiency	Dispatch Efficiency	Discharge Loss	E/P Ratio
[73]	PVA - Open Area	25	0,804	-	-	-	-
[74]	WEA_very_weak - Enercon E-115	25	-	-	-	-	-
[74]	WEA_weak - Nordex N100	25	-	-	-	-	-
[74]	WEA_middle - Vestas V80	25	-	-	-	-	-
[74]	WEA_very_strong - Enercon E-82 E3	25	-	-	-	-	-
[74]	WEA_strong - Siemens SWT-3,6-107	25	-	-	-	-	-
[75, 76]	H2 Storage Tank (Tubular Accumulator)	30	-	0,975	0,975	0	-
[77, 78]	Battery Storage (Stationary)	10	-	0,950	0,950	0	4
[78, 79]	PEM Electrolyzer	20	0,580	-	-	-	-

The efficiency of the open area PV panels corresponds to the performance factor considered in calculating the capacity factors timeseries obtained by processing the MERRA-2 meteorological irradiation data; see 4.3.2.

The efficiencies of the wind turbines are directly included in the calculation of the capacity factor timeseries, which are calculated from the MERRA-2 wind speed datasets; see 4.3.2.

The energy power ratio is a referred to the stationary battery storage ratio between its energy capacity and the amount of energy that can be supplied in a second, which corresponds to the power. Depending on the battery storage application, a certain energy power ratio or another is preferred. Generally, electric vehicle batteries have an energy power ratio of around one, while stationary battery systems in energy system applications tend to have a higher energy power ratio. Additionally, the energy power ratio represents a key parameter used to compare the costs between battery systems. Usually, batteries with lower energy power ratios are more expensive [77, 80]. In the case of the hydrogen storage tank, the energy power ratio does not represent a common parameter.

The store, dispatch efficiencies, and dispatch losses are characteristic parameters of the battery and the hydrogen storage systems. The FfE considers no discharge losses for either of the storage systems.

Finally, the electrolyzer operation is characterized by its efficiency parameter. The efficiency depends on the type of electrolyzer, and for PEM electrolyzer, 0,58 is considered.

Economic parameters data

The economic data comprehend all the investment specific costs (CAPEX), operational costs (OPEX) given as a percentage of the investment costs, and WACC for the different components of the hydrogen production system.

Table 4-6 Economic parameters of the electrolysis-based hydrogen production system

Ref.	Technology	Investment Cost Power in €/MW	Investment Cost Capacity in €/MWh	Operational Cost Power in % of investment cost	Operational Cost Capacity in % investment cost	WACC
[73]	PVA - Open Area	675551	-	0.03	-	0.035
[81]	WEA_very_weak - Enercon E-115	2034400	-	0.014	-	0.035
[81]	WEA_weak - Nordex N100	1769600	-	0.016	-	0.035
[81]	WEA_middle - Vestas V80	1267200	-	0.023	-	0.035
[81]	WEA_very_strong - Enercon E-82 E3	740000	-	0.039	-	0.035
[81]	WEA_strong - Siemens SWT-3,6-107	934400	-	0.031	-	0.035
[75, 76]	H2 Storage Tank (Tubular Accumulator)	1250	12500	0.02	0.02	0.035
[77]	Battery Storage (Stationary)	513800	133900	0.058	-	0.035
[78]	PEM Electrolyzer	1420000	-	0.02	-	0.035

The costs for the PV panels, wind turbines, and electrolyzers are denoted by their nominal power. The hydrogen and battery storage systems are characterized by both, nominal power, and energy capacity parameters.

The costs for the different wind turbines are calculated based on their technical specifications and certain factors using a SQL script provided by the FfE.

The hydrogen storage system costs are defined by investment and operational costs for its power and capacity. On the other hand, the battery systems costs are represented by the investment cost for power and capacity and only with power operational costs. The WACC is defined at the FfE with the value of 3,5% for all the technologies.

4.4 Database structure and data processing

Structuring the input and output data may facilitate a clear understanding of the model course of action. First, an overview of the input and output data is presented, followed by the database, and excel structures used to import the data into the python model. The database structure to save the output results are also shown.

Later, the processing steps of the different input data are explained.

4.4.1 Overview of input and output data

The python model for the electrolysis-based hydrogen production system takes specific input, identified, and classified in 4.3, in country, cell and system levels. This input must be processed before using it for the optimizations of the hydrogen production systems. The optimizations output must also be processed and saved in the database. An overview of the input and output data interrelations with the model is shown in Figure 4-7.

The input data is completely stored in the FfE database. However, the model imports the data not only from the database but also a small part via excel file for the technical parameters of the components. In the future, it is intended that this data is also directly read from the database.

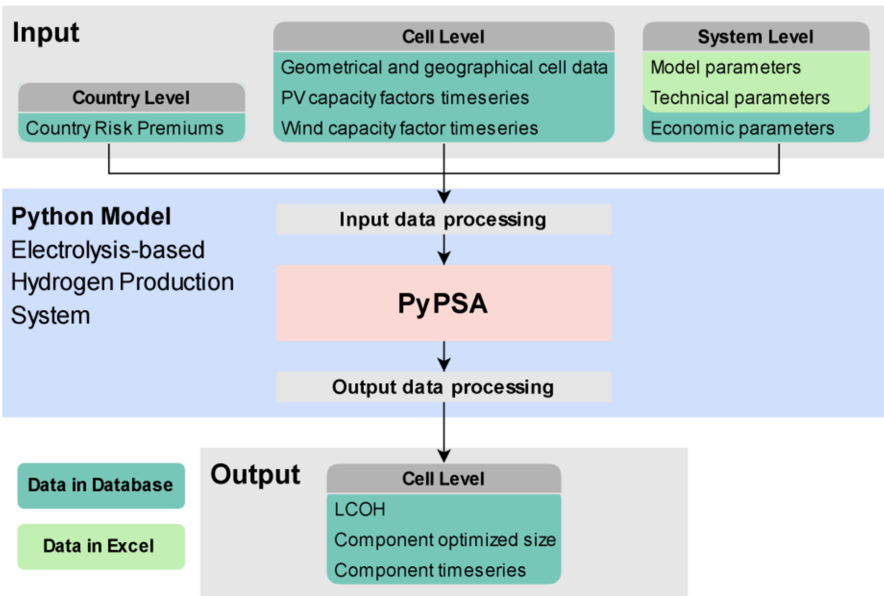


Figure 4-7 Overview of input and output data

Once the input data is introduced in the python model, either via the database or the excel file, this data will be processed before being introduced in PyPSA. PyPSA will then generate the hydrogen production system, formulate the optimization problem with all its constraints, and with an external solver, the problem will be optimized, and the output data obtained. This output data will be sorted, processed, and then directly saved in the database with a specific structure.

4.4.2 Overview of the database structure

As mentioned, all the input data is stored in the FfE database in PostgreSQL, in tables or views starting with the prefixes “t_h2_model_” and “v_h2_model_”, respectively. The difference between tables and views strikes primarily if the data is truly stored and in the presentation of it.

The tables store the data with a vertical structure commonly used in databases with relational tables. These relational tables have few columns and many rows (vertical structure), and the values or parameters inside the columns tend to be identifiers explained in additional associated description tables.

On the other hand, the views do not store data but just read and rearrange data from other existing tables to make it more comprehensive. More comprehensive usually means a greater number of columns and fewer rows (horizontal structure). The values inside the views are not necessarily identifiers or numeric values but also text, giving a direct meaning to the row. Views are useful for understanding the data stored in the relational tables.

Figure 4-8 shows the database structure in the form of an Entity Relationship Diagram (ERD). On the input side, three input data levels can be identified. Firstly, the country risk premiums for the country level, the techno-economic parameters for the system level, and finally, the MERRA-2 for the cell level.

The input data is already processed in PostgreSQL before being introduced in the python model, where it will be further processed. In the case of the output data, the results are processed in the python model and then directly saved in the database.

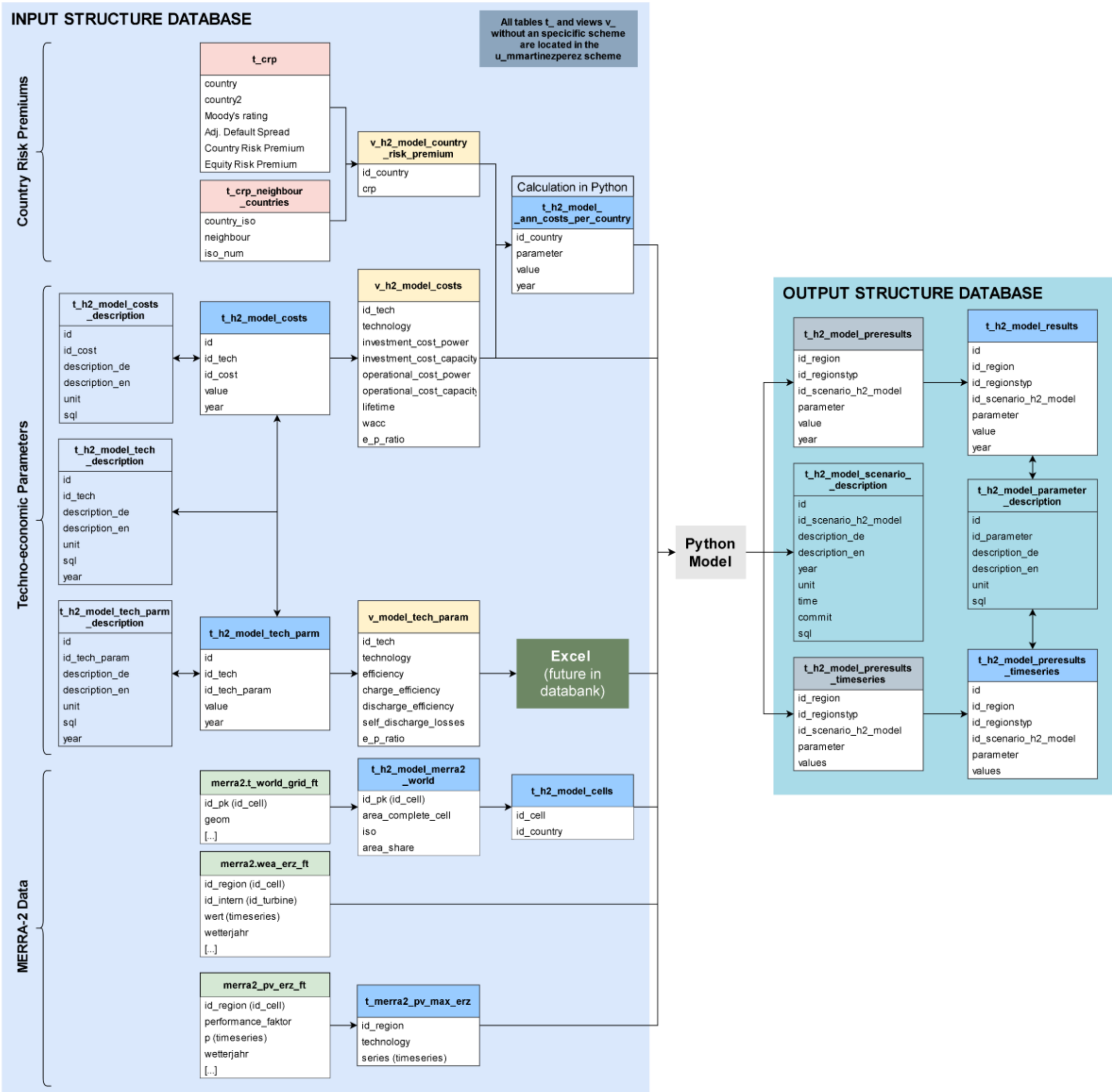


Figure 4-8 ERD overview of input and output structure in the database

4.4.3 Overview of the excel input structure for PyPSA

PyPSA is the software tasked to generate the hydrogen production system with the corresponding components, formulate the optimization problem, and finally, solve the problem. PyPSA's system, including all the components, is called a network. The network can integrate predefined components, which the user can choose. These predefined components in PyPSA are [61]:

- Buses
- Carriers
- Generators
- Global constraints
- Line types

- Lines
- Links
- Loads
- Shunt impedances
- Storage units
- Stores
- Subnetworks
- Transformer types
- Transformers

Each predefined PyPSA component has a specific function described in the PyPSA documentation. In addition, each component has specific attributes, which can be required input, optional input, or output [61].

PyPSA requires the definition of the predefined components and their attributes in a specific way so that PyPSA can create the right optimization model with its specific constraints and solve it. To do that, PyPSA allows three different paths to import the model data. These methods are further explained in the PyPSA documentation, and they are [61]:

- **Direct in python code:** after creating a network, the components are added to the network by coding in python language.
- **CSV files:** PyPSA has a defined function in its python package to import the network components from csv files. Each PyPSA predefined component requires a different csv file with a specific structure of the component attributes.
- **DataFrames:** PyPSA has a defined function in its python package to import the network components from DataFrame. Each PyPSA predefined component requires a different DataFrame with a specific structure of the component attributes.

The input attributes must be characterized in the code, csv files, or DataFrames to define the system network in PyPSA. Among these three methods, the direct definition in the python code is excluded as the flexibility to change the hydrogen production system would be time intensive and restricted to regular users of the PyPSA python package.

The alternative use of csv files is more attractive, as the definition of the system can be done externally without deep understanding of the python code and then later be imported to PyPSA. The main drawback regarding the csv files method regards the automatability criteria defined in 4.2.1 specific to the implementation of the electrolysis-based hydrogen production model. In this case, 51677 different optimization models must be defined, and each requires specific data for the cell level, see 4.3.2. For this reason, is the speed of the data import of high importance, and the comparison between the import of the components data in the PyPSA software through csv files and through DataFrames showed that the DataFrames resulted in a better performance.

This outcome is intuitive as in the case of the csv files, for each cell optimization, a csv file must be created from a DataFrame and saved in a specific directory before being able to import it to PyPSA. Using DataFrames saves the creation of the csv file, given that the DataFrame can also be imported directly to PyPSA. Therefore, the direct import of DataFrames to PyPSA is chosen over the csv files method, as it is more efficient for integrating the cell level data in each optimization.

Despite using the DataFrames, a specific structure is still needed inside these DataFrames. The name of the components must be the indexes, and the attribute names must be the column names of the DataFrames. This structure can easily be given with an excel file with as many sheets as predefined PyPSA components to be imported. Each of these sheets will be read as a DataFrame, which will later be imported to PyPSA. Figure 4-9 shows the simplified excel structure for some of the PyPSA predefined components.

Excel - H_2 Model Scenario X

sheet:	buses	carriers	generators	generators--p_max_pu	[...]
attributes:	name v_nom carrier [...]	name capital_cost co2_emissions efficiency marginal cost [...]	name bus capital_cost efficiency marginal_cost p_nom p_nom_extendable [...]	snapshots generator_1 generator_1 [...]	[...]

Figure 4-9 Excel structure for predefined components from PyPSA

Figure 4-10 is a scheme for transforming the excel sheets into general DataFrames. Together with the cell level and country level data, these DataFrames are imported into PyPSA, where the optimization takes place.

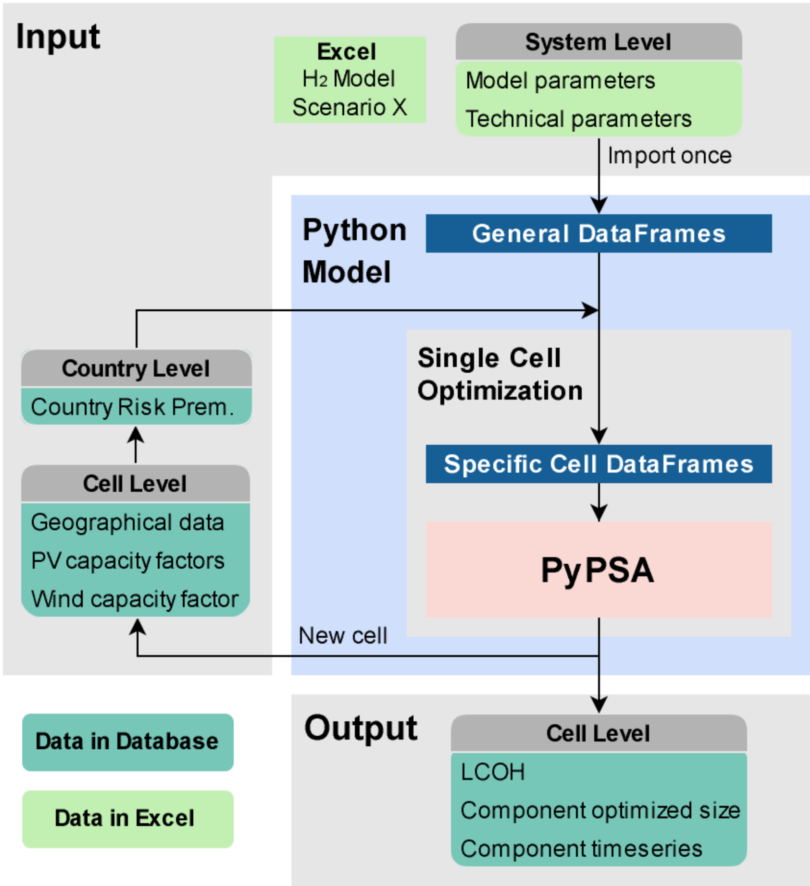


Figure 4-10 Overview of the import of excel input into the python model

The excel file is an easy path to create the needed structure in the DataFrames that must be imported into PyPSA. Nevertheless, all the data is already stored in the database, and a unique PostgreSQL query would be a cleaner and more unified solution, given that two input sources, excel and database, would be reduced to only the database. This would avoid possible errors due to possible name incompatibilities used in the database and the excel file.

This alternative path of creating the DataFrames is being considered for the future development of the model, where queries templates in PostgreSQL could be created, and the already used queries for a specific scenario could also be stored in the database table *t_h2_model_scenario_description*.

4.4.4 Data processing

The already defined input data in Input data4.3, must be processed before being introduced in PyPSA for the optimizations. This data processing can be done in the database, before entering the python model, or once the data is imported to the python model.

The following table shows where the data is processed for the three different classification levels of the data.

Table 4-7 Overview of the data levels and the data included

Input Data Level	Data	Processing in database (PostgreSQL)	Processing in Python
Country Level	Country Risk Premiums	X	
Cell level	Geometrical and geographical MERRA-2 cell data	X	
	PV capacity factors timeseries (derived from MERRA-2 solar irradiation data)	X	
	Wind capacity factors (derived from MERRA-2 wind speed data)	X	
System Level	Model parameters		
	Technical parameters		
	Economic parameters	X	X

The data processing for each of the six datasets indicated in the table above is explained below.

Country Risk Premium (Country Level)

As explained in the theoretical framework, see 2.1.3, the country risk premiums represent a specific interest rate for each country. This dataset comes from the Stern School of Business at New York University [16] and has been stored in the FfE database for further use.

This dataset integrates data for 187 countries and three additional cities from the United Arab Emirates: Abu Dhabi, Sharjah, and Ras Al Khaimah.

ISO 3166-1 enables the association of the different country names with a specific country code, often used in the FfE database. This norm integrates 249 countries, while the country risk premium dataset includes only 187 therefore, the country risk premiums dataset should be filled [82]. The FfE considers Kosovo as an additional country for some studies, which makes a total of 250 countries with its respective country codes, missing 63 countries in the country risk premiums dataset.

A simple approach is conducted to fill the country risk premiums for the missing countries. This approach consists of manually assigning the country risk premium from a neighbor country with a similar country risk based on different political and economic criteria from the Credendo analysis [20].

- Neighboring countries
- Similar country risk based on political and economic criteria [20]

This approach does not ensure the right country risk premium for the missing countries, and further analysis to assign more suitable country risk premiums might be convenient. Nevertheless, this simple method allows filling the gaps easily manually.

Once the original country risk premium dataset and the assigned neighboring countries are stored in the database, a view with the 250 countries is created to facilitate its access and visualization.

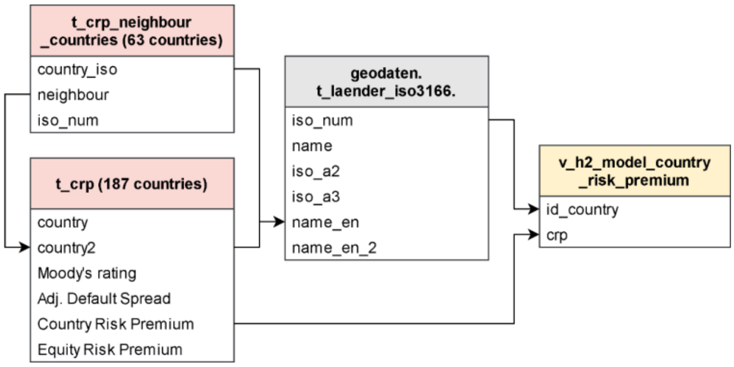


Figure 4-11 ERD for the country risk premium

The country risk premiums are needed for calculating the annualized costs per component and country, later explained.

Geometrical and geographical MERRA-2 cell data (Cell Level)

The MERRA-2 data includes the identification number of the cell and its geographical coordinates. The MERRA-2 cells do not follow political or natural borders, but they represent a rectangular grid division of the world, with cells of approximately 50x50km. Given that no political borders are followed, each MERRA-2 may belong to none (middle of the ocean), one, two, or even more countries [59].

The country assignment for each cell is needed so that the country risk premiums can be considered in calculating the country specific costs used to optimize the electrolysis-based hydrogen production system. The assignment process of a country per cell follows a simple criterion, consisting of assigning the country with the highest area share to the cell.

In the example shown below, the MERRA-2 cell with the identification number 107858 includes areas from three countries, France, Luxembourg, and Germany. The area share for each country is 44.6%, 41.2%, and 14.2%, respectively; therefore, as the biggest area share is in France, the French country code from the ISO 3166-1 will be assigned to this cell, and the French technology and component costs will be used in the optimization.

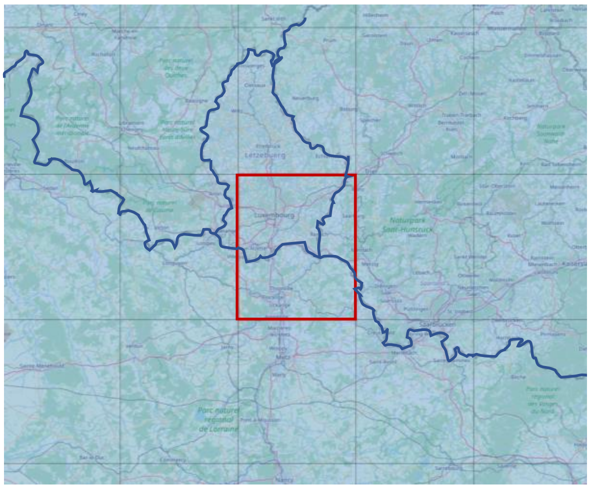


Figure 4-12 MERRA-2 cell 107858 with area shares in France, Luxembourg, and Germany

The assignment of a country per cell is accomplished with a series of PostgreSQL queries. To address this, the geometry of each MERRA-2 cell is intersected with the geometries of each country inside the cell. Once the intersection or intersections in each cell are extracted, the area share for each intersection is calculated. Two arrays are saved, the first one with the countries inside the cell and the second one with the area shares for each country. Also, the complete area of the cell is saved to facilitate future hydrogen potential analysis. The table where the processed data is saved is *t_model_merra2_world*.

The table *t_h2_model_cells* is created from this data with only two columns. The first column corresponds to the cell identification number, and the second corresponds to the country identifier with the highest area share for that specific cell. The entity relationship diagram is shown in Figure 4-13.

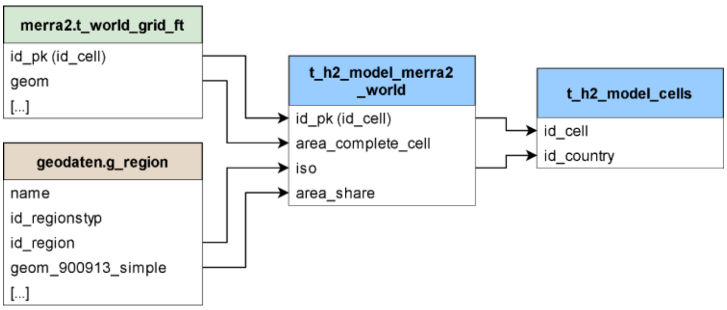


Figure 4-13 ERD for MERRA-2 cell country assignment

PV capacity factors timeseries

The PV capacity factors timeseries with an hourly resolution are derived from the MERRA-2 meteorological data and are already stored in the FfE database. As mentioned in 4.3.2, the calculated capacity factors are available in each location for five different inclinations, sixteen different cardinal orientations, and four different weather years.

The imported capacity factors timeseries to the python model correspond to the weather year 2012 and the maximum full load hours. In other words, for the 80 capacity factors timeseries for a single location in the weather year 2012 (five inclinations multiplied by sixteen orientations), only the one with the maximum full load hours, meaning the maximum sum of hourly capacity factors, is imported to the python model. These capacity factors timeseries are saved in table *t_merra2_pv_max_erz*.

Wind capacity factors timeseries (Cell Level)

The wind capacity factors timeseries with an hourly resolution are derived from the MERRA-2 meteorological data and are already stored in the FfE database. These capacity factors are calculated for five or ten different wind turbines defined in 4.3.2, depending on if the cell is completely onshore, offshore, or shares land and sea.

The developed electrolysis-based hydrogen production model considers only onshore wind turbines, see Table 4-4. Although it is possible to model the five onshore wind turbines, this would increase the computational time of the optimizations. Therefore, a pre-analysis is conducted to select which wind turbines will be the most convenient for the implementation of the model.

This pre-analysis consists of optimizing the electrolysis-based hydrogen production system size for 100 random cells, modelling the five different wind turbines. The results will show which wind turbines are installed in each cell, keeping the system costs at a minimum while fulfilling the hydrogen demand and the other system constraints.

As a hypothesis, the electrolysis-based hydrogen production system performance will be dominated by the electrolyzer, as a constant demand of hydrogen must be supplied hourly. The electrolyzer must run as steady as possible to fill the demand, requiring a regular source of electricity, PV, or wind. Regarding wind energy, the wind speed increases at higher altitudes [83], so the average wind speeds are more frequent and steadier with increasing height. The correction for wind speed based on height is given by Hellman formula [74, 83]. Higher wind turbines have then higher full load hours and are more capable of matching the constant electrolyzer electricity needs, despite being more expensive. Therefore, in the pre-analysis, most cells are expected to introduce the highest wind turbine. The highest turbine is classified as “very weak”, given that their power density is lower but steadier, so they cannot endure high wind speeds or gusts, as mentioned in 4.3.2. The pre-analysis results can be seen in 5.1.

Additionally, the norm IEC 61400-1:2019 is used to select the wind turbine in each cell. This norm centers on the design requirements of wind energy generation systems. This norm is addressed to understand the existing types of wind classes used to classify the different wind turbines [84].

The wind classes define how suitable a wind turbine is for a certain type of wind conditions, regarding the average wind speed (V_{ave}), the extreme 50-year gust (V_{ref}), and the turbulence (I_{ref}). The table with the parameters corresponding for each wind class can be found in

For certain locations, the wind is so strong that the very weak turbine is not appropriate for construction. The modelled very weak wind turbine is Enercon E-115, initially only suitable for Wind class (IEC) IIA [69]. However, the new E-115 E3 wind turbines with the same power (3MW) are suitable for the wind class (IEC) IA [85].

The MERRA-2 locations must be analyzed and classified to see which are over a wind class IA (tropical conditions), and therefore, unsuitable for the very weak wind turbine Enercon E-115 E3. The average wind speed (V_{ave}) and the reference wind speed (V_{ref}) for the 140m rotor height are then calculated according to the norm IEC 61400-1:2019. The MERRA-2 cells, which are above the limit conditions for wind class I, will consider only the construction of very strong wind turbines.

Different multiplication factors are needed to calculate the reference wind speed (V_{ref}). First, given that the weather data is only available for one year (2012), a correction factor is needed to consider them as 50-year wind speeds. This correction factor is 1,25.

Also, the available wind speed data is in hourly resolution, while the reference wind speed is calculated with the 10-min average. An additional correction factor, extracted from the World Meteorological Organization (WMO), with the value of 1,08, is used to transform the hourly average wind speed into a 10-min average wind speed [86].

The reference wind speed calculation multiplies the two correction factors by the maximum hourly average wind speed for one year:

$$V_{ref} = 1{,}25 \cdot 1{,}08 \cdot V_{2012,hourly,max}$$

Then with the reference wind speed and the average wind, the wind class for the specific location can be determined.

Model parameters (System Level)

The model parameters, defined in 4.3.3, are directly imported from the excel file and do not need any processing.

Technical parameters (System Level)

The components' technical parameters are specified in 4.3.3 and do not need any processing before being given to PyPSA.

In the database, the technical parameter values are found in the table *t_h2_model_tech_param*. To understand the parameter identifiers, two additional description tables must be considered. Figure 4-14 shows the ERD for the technical parameters in the database.

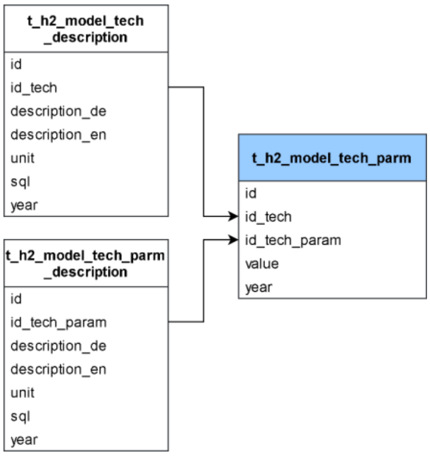


Figure 4-14 ERD for technical parameters of the components

Most technical parameters will be directly written in the excel file; see Figure 4-8 and Figure 4-10. However, taking the technical parameters directly from the database to create the DataFrames is being considered for the future development of the model.

Economic parameters (System Level)

In 4.3.3, the economic parameters for the different components are identified. In the database, the economic parameter values are found in the table *t_h2_model_costs*. To understand the parameter identifiers, two additional description tables must be considered. Figure 4-15 shows the ERD for the economic parameters in the database.

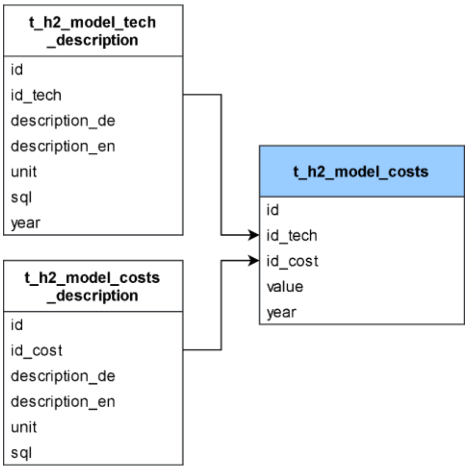


Figure 4-15 ERD for economic parameters of the components

The economic parameters and the country risk premiums are needed to calculate the annualized costs per technology and country. This calculation is made in python, independently from the electrolysis-based hydrogen production model. Its results are saved in the table, t_h2_model_ann_costs_per_country, later accessed by the python model for the optimizations of the cells.

The annualized costs are calculated using the annuity factor explained in 2.1.2, but it also depends on the component, as each component might have different parameter costs.

- **PV, wind turbines, and electrolyzer:**

These components only consider specific investment and operational costs regarding power in €/MW. Therefore, the calculation of their annualized cost is direct.

$$Ann.cost_i = CAPEX_{pow,i} \cdot (\frac{1}{AnF_i} + OPEX_{pow,i}(\%)) \quad in \quad \frac{€}{MW \cdot year}$$

- **Battery storage:**

Besides having power specific investment and operational costs in €/MW, the battery storage system has also capacity investment costs in €/MWh. The investment power and capacity costs can be related through the e/p ratio.

$$Ann.cost_{batt} = CAPEX_{total,batt} \cdot (\frac{1}{AnF_i} + OPEX_{pow,i}(\%)) \quad in \quad \frac{€}{MW \cdot year}$$

With:

$$CAPEX_{total,batt} = CAPEX_{pow,batt} + e/p_{ratio,batt} \cdot CAPEX_{cap,batt} \quad in \quad \frac{€}{MW}$$

- **Hydrogen storage:**

It has investment and operational costs for power and capacity. Therefore, two different annualized costs are calculated, for power characterization and energy capacity characterization.

$$Ann.cost_{pow,i} = CAPEX_{pow,i} \cdot (\frac{1}{AnF_i} + OPEX_{pow,i}(\%)) \quad in \quad \frac{€}{MW \cdot year}$$

$$Ann.cost_{cap,i} = CAPEX_{cap,i} \cdot (\frac{1}{AnF_i} + OPEX_{cap,i}(\%)) \quad in \quad \frac{€}{MWh \cdot year}$$

The country risk premium is included in the calculation of the annuity factor, as shown below.

$$AnF = \frac{(1-(1+r)^n)}{r} = \frac{(1+r)^n - 1}{r * (1+r)^n}$$

With:

$$r = WACC = WACC_{base} + Country\ Risk\ Premium$$

Figure 4-16 shows the entity relationships diagram for the calculation of the annualized cost per country and technology.

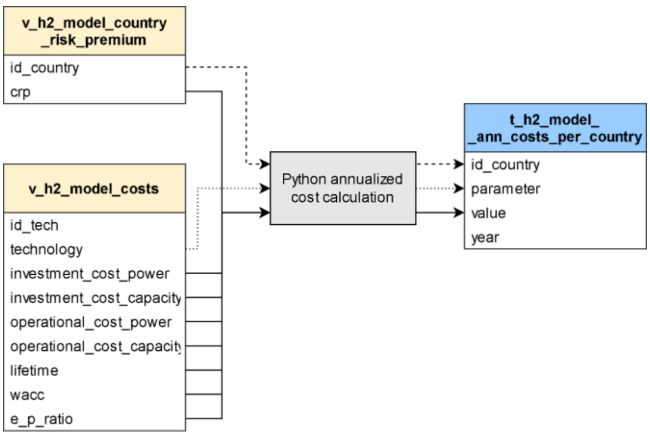


Figure 4-16 ERD for the calculation of the annualized cost per country and technology

4.5 Output data

The PyPSA optimizations of the electrolysis-based hydrogen production model result in a large amount of data that must be sorted and selected according to what is interesting for further analysis. The kept data is then stored in the database and can be classified into three groups. Firstly, the LCOH, then the optimized component sizes, and the timeseries for the behavior of the components.

Levelized Cost of Hydrogen (LCOH)

While the component sizes and the timeseries can be directly extracted from the optimization results generated by PyPSA, the LCOH was calculated in python and then stored in the database as a result so that it can be easily accessed. The LCOH calculation follows the equations explained in the Theoretical Framework points 2.1.1 and 2.1.2 and includes the following components:

- Electricity generators: PV, Wind very weak and Wind very strong
- PEM Electrolyzer
- Battery Storage System
- Hydrogen Storage System

Including the hydrogen storage system in the LCOH has raised the question of whether a hydrogen storage system should be truly included, given that the LCOH usually covers only the hydrogen production system. In the case of the levelized cost of energy (LCOE), it covers only the power plant for electricity production. However, hydrogen is not electricity, and arguments against and for its inclusion are identified.

For the inclusion, the main argument is that there is no equivalent to the electricity network in the hydrogen economy, where the hydrogen can be fed after production. Therefore, a hydrogen storage system is required to make the system more realistic. In addition, the hydrogen storage system gives the system certain flexibility needed for the cost optimization of hydrogen production. Another argument is that this system is found on the supply side, which means that hydrogen storage systems will realistically be installed on the production side, while also on the demand side, additional hydrogen storage systems will be installed, as in any other chemical industry.

On the other hand, the main argument against the inclusion of the hydrogen storage system focuses on the main objective of the LCOH, which is to make possible the comparison between different hydrogen production technologies with a single parameter. Therefore, the calculation of the LCOH should be uniform among the different studies. Suppose other studies consider the hydrogen storage system in their calculation, for example, cost production analysis of hydrogen from steam reforming of natural gas. In that case, the LCOH in this academic work should include it too. Nevertheless, there is no consistency in the analyzed hydrogen production optimization models, where [34, 37] include the hydrogen storage systems in the cost calculation, [42] plans to add, and the other five do not consider it in the cost optimization model at all [35, 36, 38–40].

For the LCOH calculation, the storage system is included as this system is meant to act as an intermediate storage needed to get realistic optimization results. Without hydrogen storage, the modelled system would not be flexible enough, and it would depend entirely on the battery storage system, strongly increasing the costs of the system.

One question that arises now is if the assumptions for the hydrogen storage system are correct, regarding mainly two aspects. Firstly, is a gas hydrogen storage system realistic? Or should another kind of hydrogen storage, such as a compressor and a liquid hydrogen tank, be considered? Secondly, should the size of the hydrogen storage tank be limited to a certain value? These questions are out of the scope of this academic work but are interesting for the outlook of the electrolysis-based hydrogen production model.

Optimized size of the modelled components

The optimizations results give the optimum size of the modelled components in MW or MWh, depending on if their characteristic parameter is power or energy capacity.

Timeseries of the modelled components

The optimization results include many timeseries describing the modelled components' performance and load profiles for all the timesteps.

The output structure can already be seen in Figure 4-7 and Figure 4-8, but the specific selected parameters from the optimization results and stored in the database can be found in the Appendix III: Output data stored in database

4.6 Python Code

The electrolysis-based hydrogen production model is developed in python. However, high interaction with a PostgreSQL database is needed to access the input and store the output data.

4.6.1 Module structure

The python model is built in different modules that facilitate the structuring and understanding of the model. These modules, shown in the scheme from Figure 4-17, are main.py, input.py, precalc.py, opt.py, extra_pypsa.py and output.py. The code can be found in Appendix VI: Electrolysis-based hydrogen production python model code.

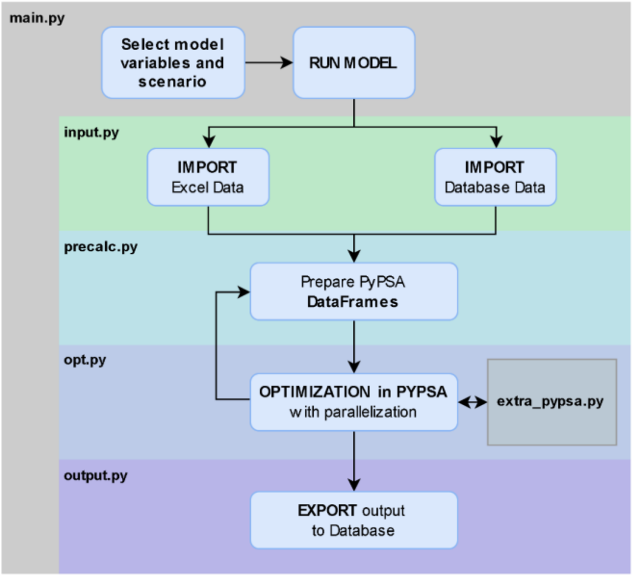


Figure 4-17 Overview of python code modules

Main.py module

The main.py module centralizes the execution of the electrolysis-based hydrogen production model. It is built so that the first lines of the main.py module are the only ones the user has to modify selecting the specific variables and scenario, before running the entire model.

The variables to be defined in the main.py module can be divided into Boolean and non-Boolean variables.

Boolean variables

- *calculate_ann_costs_in_db*: if True, calculate new annualized costs per technology and country and selected year, importing the economic technology parameters and country risk premiums from the database and saving the annualized costs table back in the database. Default False.
- *optimize_each_cell*: if True, an electrolysis-based hydrogen production system will be optimized for each cell. Default True. The following variables can only be used if *optimize_each_cell* is set to True.
    - *import_data_db*: if True, the data used for the optimizations of the cells are directly imported from the databank. Default True

- *save_imported_data_feather*: it can only be True if *import_data_db* is set to True and *load_data_feather* is set to False. If True, the imported data from the database are saved as feather files.
- *load_data_feather*: only can be True if *import_data_db* is set to False. If True, the data used for the optimizations are imported from the saved feather files.
- *store_cells_as_files*: if True, every optimized cell PyPSA network will be saved in a folder. These files can take high amounts of storage space. Default False
- *work_output_data*: if True, the results of the optimizations will be processed, and the data can either be saved as files or stored in the database. Default True.
- *save_results_as_files*: only can be True if *work_output_data* is True. If True, the results (LCOH and component optimized sizes, without timeseries) will be saved as csv, parquet, and feather files. Default False.
- *insert_data_in_db*: only can be True if *work_output_data* is True. If True, the results (LCOH, component optimized sizes, and timeseries) will be stored in the database.
- *overwrite_scenario_results_in_db*: if True and *insert_data_in_db* is set to True, the existing results in the database only for the chosen scenario will be replaced with the new results.

* *print_time_summary*: if True, a summary of the time needed for the different code modules will be displayed on the python console. The summary includes "Import or load data time", "Preparations time", "Optimization time", "Work with output time", and "Total time". Default True.

Non-Boolean variables:

* *num_parallel_processes*: integer. It indicates the number of optimizations to be done simultaneously. The number of cores used must be less than half of the available cores or virtual computers available in the used computer, given that Gurobi uses for each optimization two virtual computers to solve the optimization problem in a short time. In case another solver different than Gurobi or less than two virtual computers for each parallel optimization is preferred or required, the variables *"solver_name"* and *"solver_options={"threads": }"* in the opt.py module must be changed.

  If *num_parallel_processes* is set to 1, all the optimizations for the 51677 cells will take more than 17 days. Therefore, a minimum of *num_parallel_processes = 8* is recommended so that the whole model optimizations take around two days. Therefore, a powerful server is needed.

- h2_scenario_id: integer. It indicates which scenario is going to be calculated. The h2_scenario_id is related to the name of the excel name to which the scenario is called in the main.py module. Also, this number will be stored as the id_scenario_h2_model for the results in the database.
- scenario_year: integer. It indicates the year for the calculated scenario. 2020 is the only year considered in the scope of this academic work.

Input.py module

The input.py module contains all the functions related with:

- The data import from the database
- The save and load of input data in feather files
- The data import from the excel file
- The import of DataFrames into PyPSA
- Other functions related with the input data

Precalc.py module

The precalc.py module includes all the functions related with:

- The processing of the data python in dictionaries
- The calculation of the annualized costs per technology and country
- The update of the DataFrames for the PyPSA input, PV and wind capacity factors, and annualized costs of the components.

An important function here is the *import_id_cells_data_from_db*, which defines which cells will be optimized. As input, an established connection with the database is needed. This function imports the identification numbers of the cells with their respective belonging country as a DataFrame, through a PostgreSQL query. This function is predefined to call all the 51677 cells considered in 4.1.2. Therefore, if another sample of cells belonging to a specific world region is desired, the PostgreSQL query should be changed here. Currently, an alternative PostgreSQL query is shown as a comment that imports the 100 random cells.

Opt.py module

The opt.py module, named after optimization, contains the functions for:

- The optimization of all considered cells using PyPSA.

- Enabling the parallelization of the optimizations

Extra_pypsa.py module

The functions included in this module enable two main changes on the PyPSA networks. The first function, called *multiple_input_output*, allows the links in the network to accept more than one input and output. This function is important for model's future development to be able to give water or heat (SOEC electrolyzer) as additional inputs to the electrolyzer Link.

The second extra change involves the transformation of Storage Units without a predefined energy-power ratio into two Links, a Bus, and one Store component so that the size of its power and capacity can be optimized separately. This functionality is needed because the predefined component PyPSA Storage Unit allows only the optimization of the power size. In contrast, the Store component only optimizes its energy capacity size. Replacing the Storage Unit will make the system more complex, but the tank power and capacity will be optimized separately. Figure 4-18 shows the replacement.

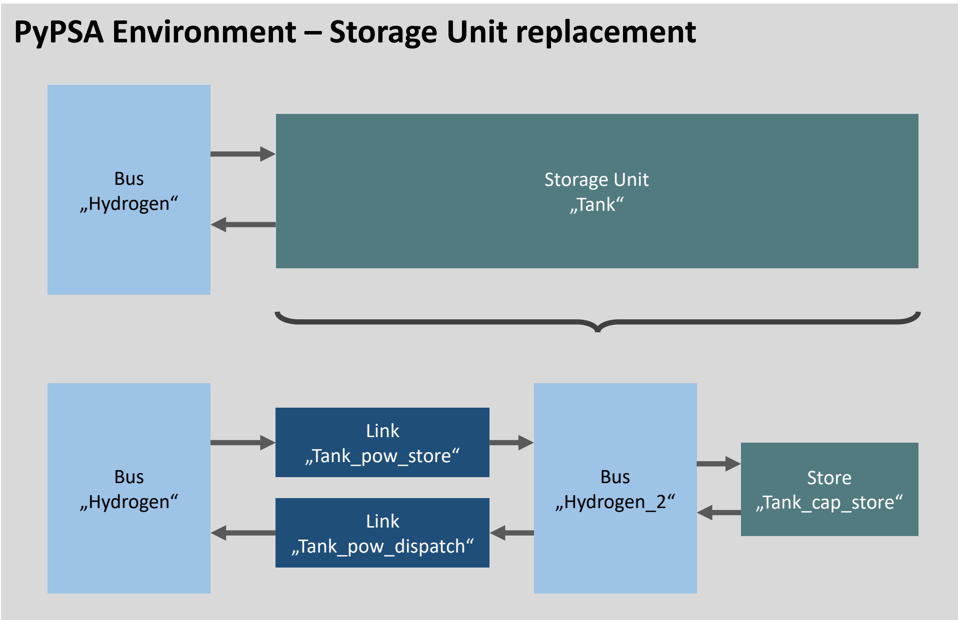


Figure 4-18 Replacement of Storage Unit with equivalent components in PyPSA

Link "Tank_pow_store": enables the storage of hydrogen. Its characteristic parameter is its power, which represents the amount of hydrogen, that can be stored per unit of time in MW.

Link "Tank_pow_dispatch": enables the dispatch of hydrogen. Its characteristic parameter is its power, which represents the amount of hydrogen that can be dispatched per unit of time in MW.

Store "Tank_cap_store": enables the storage of hydrogen. Its characteristic parameter is its capacity, which represents the amount of hydrogen, that can be stored in MWh.

Bus "Hydrogen_2": enables the connection between the new Links and the new Store.

Output.py module

The output.py module contains all the functions related with:

- The processing of the results
- The export of all the output results into the database
- Other functions to save and load output data from specific files (csv, parquet, feather)

4.6.2 Time performance and parallelization

The performance time of the model represents a challenge, given the high number of cells to be optimized, in total, 51677. To have an idea, if each linear optimization requires five seconds, the whole model will need at least three days, which may still be reasonable. Nevertheless, if each linear optimization takes 30 seconds, the performance time of the model will be around 18 days, which is not reasonable, especially if different calculation scenarios are considered.

The first tests of linear optimizations with a free software solver needed several minutes to find the solutions, which meant that different approaches were required to reduce the optimization time per cell. Two main strategies have been adopted to address this issue, the improvement of PyPSA performance and the parallelization of the optimizations.

Improvement of PyPSA performance

For the improvement of the PyPSA performance, three different approaches are considered. First, the import of input data to PyPSA, then, the use of a powerful commercial solver to calculate the optimizations, and finally, not using pyomo to formulate the optimization problem.

- Import of input data to PyPSA: as mentioned in 4.2.3 the input data for PyPSA can be introduced directly in the python code or be imported through csv files or DataFrames. The most time-efficient approach here is to consider DataFrames, as they save the step of saving and loading csv files for each optimization. The time saved using this approach is not significant, but it makes the code more efficient and understandable.

- Using Gurobi commercial solver: while there are existing free software optimizer solvers, such as GLPK or CPLEX, the use of commercial solvers, such as Gurobi, outrun the performance of the free software. The implementation of Gurobi as a solver for optimizing the cells significantly increased performance. It went from several minutes per optimization to around 40 seconds. A student license was used to implement the solver in the python code.
- Not using Pyomo to formulate the optimization problem: PyPSA uses pyomo as the default path for implementing the optimization problem, with all its variables and constraints. However, in 2020, PyPSA introduced a new internal optimization framework that does not use pyomo, which is faster and less memory intensive. This framework has been optimized, fixing small issues, and in February 2022, the last update of the non-pyomo approach was released. The downside of the non-pyomo path is that the additional constraints and functionalities require a specific structure. Changing to non-pyomo was the last approach to improve the runtime performance, reducing the optimization time to 30 seconds.

Parallelization of the optimizations

The parallelization of the optimizations consists of running different optimizations simultaneously using multiple cores or virtual computers. The parallelization should be conducted in powerful computers or servers with multiple cores or virtual computers, ideally at least 16. At the FfE, the employed server has 46 virtual computers.

The parallelization was conducted using the multiprocessing python package. In particular, the Pool class is used (instead of the Process class), given that the number of tasks is high [87].

For the Pool class, different parallelization methods are available in the multiprocessing python package. These methods can be implemented through the functions apply, map, apply_async, map_async, starmap, starmap_async, imap, and imap_unordenred [87]. The chosen function is imap, which saves memory by not converting the iterable into a list. Also, the chunk is given as input to accelerate the process. The alternatives async, starmap and unordered consider additional features not interesting for the model.

The number of parallel optimizations is given externally in the __main__.py module through the variable num_parallel_processes. It is defined that the num_parallel_processes cannot surpass half of the total cores of virtual computers available in the computer. This requirement is needed due to Gurobi, given that it requires more than one core for itself to perform properly in a short time.

By default, Gurobi uses all available CPU cores in the computer to provide the best possible performance. In the parallelization process, the number of CPU cores used for each simultaneous optimization road must be limited to avoid overloading the CPU resources. This limit is defined in the opt.py module with *"solver_options={"threads": 2}".*

Table 4-8 shows the results of a Gurobi performance test. This test was done with different combinations of CPU cores used per parallel process. Ten cells were optimized for the test for five different cases, repeating each case twice.

Table 4-8 Performance comparison for parallelization with Gurobi

Case	Parallel optimizations	CPU core / parallel optimization	Number of CPU cores	1. total time in s	1. time per it in s	2. total time in s	2. time per it in s
1	5	2	10	95.98	31.67	108.27	32.99
2	10	1	10	180.89	95.26	178.99	96.24
3	5	1	5	271.85	79.51	274.34	79.35
4	2	2	4	179,46	28,73	177,47	28,62
5	2	3	6	226,72	27,21	229,04	27,94

From Table 4-8, it can be derived that the performance of Gurobi decreases rapidly when only one CPU core is used to solve the optimization (CPE core per parallel optimization). Additionally, the performance difference between using 2 or 3 CPU cores to solve the optimizations is insignificant. Therefore, the configuration of two CPU cores per parallel process is adopted.

This chosen configuration explains the limit specified in the python code, where the number of parallel optimizations cannot exceed half of the total number of CPU cores. Therefore, each single parallel optimization process is programmed to use two CPU cores to enable a proper solver performance time.

An example concerning the FfE server, which has 46 available virtual computers (equivalent to CPU cores), is presented. If the variable num_parallel_processes is initialized to 40, then the python model will restrict the number of simultaneous optimizations to 23, which is the number of available virtual computers divided by two. In that case, 23 optimizations will be performed simultaneously, each using two virtual computers, and therefore the whole server capacity will be used.

The FfE server with 46 virtual computers was used to calculate the results. However, to avoid overloading the server, only a maximum of 60% of the server was used for the calculation, corresponding to 27 CPU cores. In the end, only 24 CPU cores (12 parallel optimization processes) were used, taking less than two days of calculation time.

The parallelization of the optimization allows a remarkable decrease of the calculation time down to a reasonable duration.

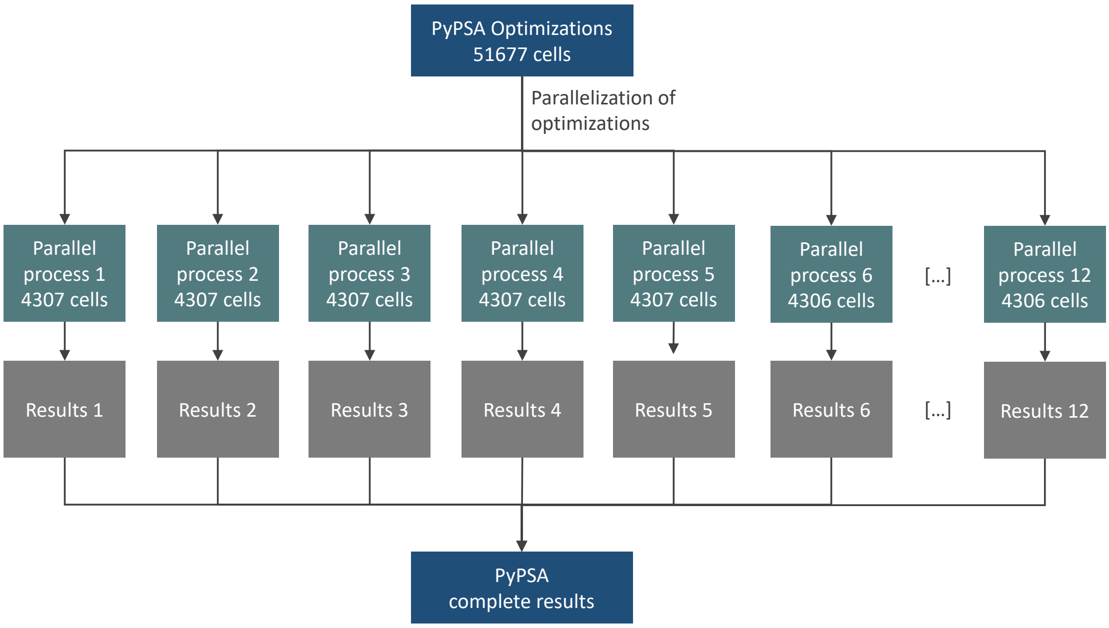


Figure 4-19 Parallelization of PyPSA optimizations in twelve processes

4.7 Selection of calculation scenarios

The calculation scenarios represent different configurations of the electrolysis-based hydrogen production system, each with specific components and assumptions. The effects of the different components and assumptions on the levelized cost of hydrogen can be studied by comparing these scenario results. The analysis of the effects should help answer the second and third research questions found in 1.2. These research questions regard the effects on the LCOH of:

- the PV-wind hybrid system,
- the battery storage and
- the country risk premiums (CRP)

The calculation scenarios are presented in the following table:

Table 4-9 Calculation scenarios for the electrolysis-based hydrogen production model

id	Scenario	CRP	Hybrid (PV-wind)	Battery storage	H_2 storage
1	_base_complete	x	x	x	x
2	_constant_wacc		x	x	x
3	_only_pv	x	Only PV	x	x
4	_only_wind	x	Only wind	x	x
5	_affordable_battery	x	x	50% cost	x

The base scenario, named "_base_complete" incorporates all components in the optimization model. Also, the country risk premiums are included in the int base scenario, considered a plausible assumption, for two reasons. First, the base scenario should be the most realistic one, and including the country risk premiums also comprises the economic attractiveness of the countries from an investor perspective. Secondly, as its name says, the base scenario is used for comparison with the other scenarios, retrieving a component or an assumption in the next calculation scenario so that the effects are easier to analyze.

The "_the_constant_wacc" scenario does not consider the country risk premiums. Therefore, the interest rate considered to annualize the costs of the components is the base WACC presented in Table 4-6 for the economic parameters of the different components, and will be the same for all the countries in the world. This scenario approaches the case when all countries are equally attractive to investors, independently of the country's political and economic situation. Then, with this scenario, it is expected to see how the country risk premiums affect the hydrogen production cost in regions or countries with high solar and wind renewable energy potential.

The "_only_pv" and "_only_wind" scenarios consider only PV and only wind technologies, respectively, to produce electricity. With these scenarios, it is expected to see the advantages of hybrid PV-wind production systems for certain areas.

The last scenario, called "_affordable_battery" was not initially planned. Instead, the idea was to consider a scenario with no battery storage. However, once the pre-analysis was done, see 5.1, it was observed that no battery was introduced in the optimization results for the 100 random cells. That means that the battery investment and operational costs are probably still too expensive for it to be economically interesting for an off-grid electrolysis-based hydrogen production system. Therefore, an alternative scenario was conceived, where the battery annualized costs are reduced to half, to see what effects a battery would have on the LCOH and in which regions and conditions.

The results for these scenarios are presented and analyzed in 5.2.

5 Results

This chapter examines firstly the results regarding the pre-analysis for 100 random cells, followed by the results analysis for the different calculation scenarios answering the research questions proposed in this academic work.

5.1 Pre-analysis results

As mentioned in 4.4.4, the wind capacity factors are available for five onshore wind turbines defined in 4.3.2. Although including the five onshore wind turbines in the model is possible, it would increase the computational time, which is not desired. Now the question is to identify if despite computational time increase, the optimization results considering the five onshore wind turbines would be any different than considering only a limited number of the available onshore turbines.

A pre-analysis is then conducted in which 100 random cells worldwide are optimized considering the five available onshore wind turbines, very strong, strong, middle, weak, and very weak.

The optimization results for the 100 random cells can be found in the Appendix IV: Pre-analysis results – 100 random cells. The results include the calculated LCOH in €/kg and the sizes of the modelled components in MW for installed power and MWh for the hydrogen storage and battery capacities; see the output data units in Appendix III: Output data stored in database. Additionally, the tank energy power ratio is calculated. From the pre-analysis results, different interesting aspects can be identified.

Firstly, the linear optimization never introduced the battery storage system, which means that curtailment is preferred to the storage of electricity, as the battery's investment and operational costs are too high to provide an economic advantage in the system. After analyzing these results, an alternative calculation scenario was introduced called "_affordable_battery" considering 50% of the annualized battery costs. This scenario pretends to identify the regions and conditions in which battery storage is more likely to be introduced. This calculation scenario is explained in 4.7.

Given that no battery is introduced and probably solar and wind is not constantly available for electricity production, an alternative component is needed to provide some flexibility, to meet the hourly demand. This component is the hydrogen storage tank, introduced in every optimized cell with energy capacities that differ from 1.33 MWh (40 hours of covered demand at its full capacity) up to 18.66MWh (560 hours or 23 days of covered demand). The energy power ratio of the hydrogen storage embodies high variations, where the highest energy power ratios correspond with higher energy capacities, and therefore, longer storage.

Regarding PV, the installed power of PV and the LCOH are correlated so that the lowest LCOH corresponds to the lowest installed PV power, and as one increases, the other does too.

The primary motivation for this pre-analysis concerns the wind turbines. Repeating the hypothesis described in 4.4.4, the electrolyzer would define the whole system's behavior, given that it is intended to run as constantly as possible to be able to fill the hourly hydrogen demand. To achieve this, a steady supply of electricity is needed, which is expected to be supplied by the very weak wind turbine, despite being more expensive, because it holds the highest full load hours.

As it is observed, the hypothesis is verified, where from the five modelled wind turbines, only three of them are introduced in the results, and the very weak wind turbine has the highest introduction rate by far. Table 5-1 summarizes the introduction rate for generation technologies in the pre-analysis results.

Table 5-1 Rate of introduction for electricity generation components in pre-analysis results

	PV	Very Weak Wind	Weak Wind	Middle Wind	Strong Wind	Very Strong Wind
Rate of introduction in %	99%	73%	0%	0%	1%	5%

At first sight, the weak wind turbine and the middle wind turbine can be excluded from the model without triggering any change in the optimization results. In addition, the strong wind turbine is also expected to be excluded, keeping the very weak and the very strong wind turbines in the model. Further analyses are carried out, considering two other scenarios.

- Scenario A (base): PV and five onshore wind turbines
- Scenario B: PV and only very weak wind turbine
- Scenario C: PV and only very strong wind turbine

These scenarios should enlighten the effects on the LCOH of restricting the modelled wind turbines. The analyses are implemented for 10 of the 100 random cells considered in the pre-analysis, which include the six cells with strong wind turbines and very strong turbines and other four random cells.

Table 5-2 shows the results for the 10 selected cells, with the absolute difference of LCOH between scenarios A and B and between scenarios A and C. If only the very weak turbine or only the very strong wind turbine are modelled, scenarios B and C respectively, then the LCOH always increases. The LCOH increase is mainly significant for high LCOH, however for the cells 96855 and 187101 with the lowest LCOH, the increase is respectively 0,08 €/kg (2%) and 0,10 €/kg (2%), supporting then the inclusion of both very weak and very strong wind turbines in the hydrogen production model.

Mention that in cell 162700, where neither the very weak nor the very strong turbines were introduced, the difference in LCOH is 0,01 €/kg, representing a 0,15% LCOH increase in 1% of the 100 cells sample. It is assumed that the exclusion of the strong wind turbine will not lead to significant LCOH changes.

Table 5-2 LCOH comparison for different wind turbine scenarios in the pre-analysis

Cell	Optimal Wind Turbine configuration	LCOH (€/kg_H2) Absolute Difference Comparison				
		A - 5 modelled wind turbines	B - Only Very Weak Turbine	B - A	C - Only Very Strong Turbine	C - A
16907	very_weak	5,79	5,79	0,00	5,98	**0,19**
119263	very_weak	5,82	5,82	0,00	5,93	**0,10**
184029	very_weak	10,79	10,79	0,00	12,36	**1,57**
187101	very_weak	4,71	4,71	0,00	4,81	**0,10**
54353	very_strong	10,68	10,86	**0,18**	10,68	0,00
96855	very_strong	4,84	4,92	**0,08**	4,84	0,00
114910	very_strong	5,79	5,80	**0,01**	5,79	0,00
152233	very_strong	5,86	5,88	**0,02**	5,86	0,00
194146	very_strong + very_weak	7,59	7,60	**0,01**	7,62	**0,04**
162700	strong	6,57	6,58	**0,01**	6,58	**0,01**

From the pre-analysis, it is deduced that the battery will not play a critical role in the worldwide optimization results. At the same time, the hydrogen storage system will assume the flexibility function for meeting the demand. In addition, for the specific cell sample, the best LCOH belongs to the cells with the lowest PV installation power; nevertheless, the effect of hybrid PV and wind technologies is yet to be analyzed. Finally, only the very weak and the very strong wind turbines will be modelled for the optimization of the 51677 cells.

5.2 Scenario results

The first research question concerning the most suitable open-source software for developing the electrolysis-based hydrogen production model was already answered in 4.2. The chosen software, PyPSA, is employed in the model to formulate and calculate the hydrogen systems optimization problems for each of the 51677 MERRA-2 cells.

The optimizations are performed for the five scenarios explained in 5.2 and the output results detailed in 4.5 are stored in the database. This output is analyzed to answer the two remaining research questions. The results presented in this chapter are subjected to the specific assumptions of the hydrogen production model for this academic work.

The two remaining research questions cover the effects of the PV-wind hybrid system, the battery storage system, and the country risk premiums on the levelized cost of hydrogen (LCOH). Additionally, some interesting results about the hydrogen storage system will be presented. Therefore, for the sake of a clear presentation of the results, five different result sections are introduced:

1. General LCOH
2. Country risk premiums
3. Hybrid PV-Wind electricity generation
4. Battery storage system
5. Hydrogen storage system

The results are mainly presented through the visualization of the parameters and correlations in worldwide maps that enable a straightforward interpretation of the results. Furthermore, bar diagrams and density scatter diagrams are also employed.

1. General LCOH

Although the assessment of the LCOH values is not meant to answer any research question, it supports the understanding of further data analysis. The LOCH results come from "_base_complete" scenario, which considers all the components and the country risk premiums.

Figure 5-1 shows the distribution of the LCOH worldwide in €/kg for green hydrogen production considering the techno-economic parameters for 2020. Additionally, as explained in 4.5, the calculation of the LCOH includes the hydrogen storage system installed power and capacity costs.

In Figure 5-1, the cheapest LCOH cells are located mainly in Australia, southern Chile, the United States, the North Sea region in Europa, Greenland, and a small area in Brazil. These locations (except Greenland, which is only known for its hydropower potential for hydrogen production [88]) are found in the 20 countries with the lowest LCOH in 2050, according to the optimistic scenario from the IRENA Green hydrogen cost and potential report [89]. Another region worth mentioning and included in the IRENA report is Saudi Arabia. Also, the study "The Future of Hydrogen" from IEA shows in Figure 14 similar economic LCOH regions as the optimization results [56].

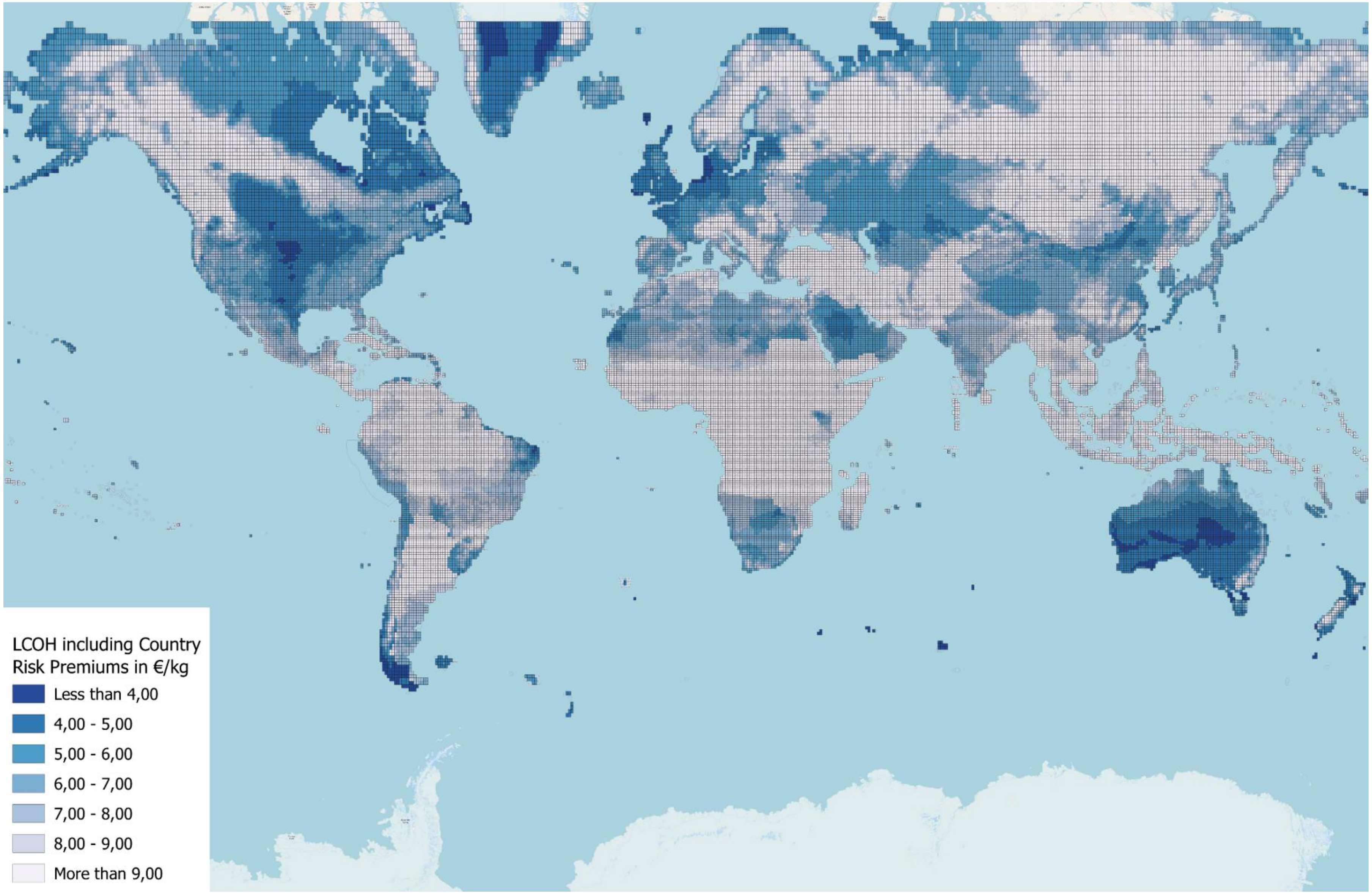


Figure 5-1 LCOH for the _base_complete scenario including country risk premiums

Nevertheless, in the IRENA report, other countries such as China, Argentina, or North African countries, are also predicted to lead the LCOH benchmark but do not show a high potential in the results for the "_base_complete" scenario. This effect is due to the country risk premiums and will be explained in the second section.

After the short qualitative comparison of the global LCOH results, further analysis for some specific locations is carried out. The breakdown of the LCOH for three calculation scenarios and five specific locations is presented below; see Figure 5-2.

The first two scenarios are the "_base_complete" and "_constant_wacc" respectively, with and without the country risk premiums (CRP). The third case corresponds to the minimum LCOH between the "_only_pv" and "_only_wind" scenarios for each specific cell, also including the CRP effect. The five studied locations are specified in Table 5-3 with the CRP and the LCOH for the three scenarios.

Table 5-3 Five specific locations for the LCOH breakdown

Id. cell	Location	CRP in %	LCOH in €/kg _base_complete	LCOH in €/kg _constant_wacc	LCOH in €/kg _only_pv or _only wind
64753	Chile – Atacama Desert	0,70%	5,29	5,03	5,29
108588	Germany – North Sea	0,00%	3,95	3,95	4,02
106020	Algeria – Hassi R'Mel	6,43%	7,18	4,52	8,91
123594	Mozambique – South Xai-Xai	8,90%	8,37	4,53	9,86
172670	Australia – South coast Albany	0,00%	3,96	3,96	4,03

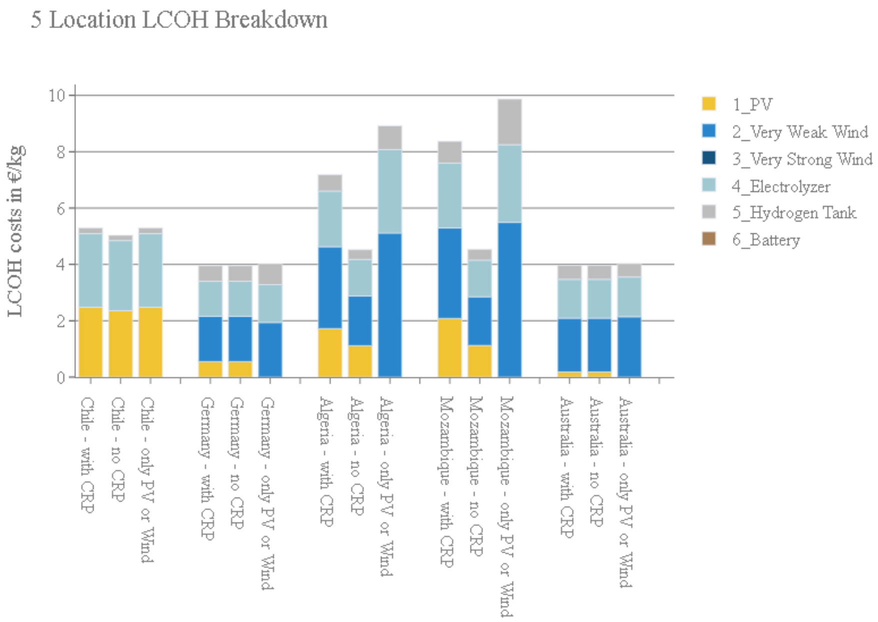


Figure 5-2 Bar diagram of LCOH breakdown for five specific locations

For the five specific locations, the production of electricity with PV and wind turbines represents at least half of the total LCOH, while the hydrogen storage system does not entail a high share of the production costs.

Concerning electricity production, in the locations where wind potential exists (all but the Atacama Desert in Chile), the LCOH is lower for the case with constant WACC (no country risk premiums considered). Also, comparing the first and third scenarios, it is observed that the hybrid PV-wind electricity production case usually offers a cost advantage. This cost advantage is significant for Algeria and Mozambique, given that they hold a higher ratio of PV-wind full load hours (0,46 and 0,44, respectively). On the other hand, the cost advantage is almost insignificant for Germany and Australia, which hold higher wind potential and then a low ratio of sun-wind full load hours (0,18 and 0,30 respectively); see Table 5-4.

Table 5-4 PV and Wind Full Load Hours (FLH) for the five specific locations

Id. cell	Location	PV FLH	Wind very weak FLH	Wind very strong FLH	Ratio PV FLH / wind very weak FLH
64753	Chile – Atacama Desert	2109	1559	672	1,35
108588	Germany – North Sea	1011	5584	3345	0,18
106020	Algeria – Hassi R'Mel	1746	3718	1769	0,46
123594	Mozambique – South Xai-Xai	1649	3684	1731	0,44
172670	Australia – South coast Albany	1544	5079	2771	0,30

Regarding the country risk premium effect, the higher the country risk premium, the higher the LCOH increase. Algeria, with a country risk premium of 6,43%, contemplates an increase of 59%, while Mozambique, with a country risk premium of 8,90%, experiences an increase of 85%. The effect of the country risk premiums will be further analyzed in the next section of the results.

Besides analyzing the different scenarios for the five selected locations, a quantitative comparison of the LCOH with other LCOH studies is conducted. This quantitative assessment is only carried out for the first three locations, Chile, Germany, and Algeria, employing the "_constant_wacc" scenario (3,5% WACC), as the other studies do not consider the country risk premiums. Other LCOH analysis studies do not consider the hydrogen storage system in the LCOH calculation; therefore, the hydrogen storage costs are extracted from the LCOH results and shown in Table 5-5.

Each study considers its own techno-economic assumptions, reason why the single value comparison is difficult to interpret.

Table 5-5 LCOH comparison with external studies for the five specific locations

Id. cell	Location	LCOH in €/kg _constant_wacc	LCOH in €/kg _constant_wacc (without tank)	Other Study LCOH in €/MW	Other Study WACC in %	Ref.
64753	Chile – Atacama Desert	5,03	4,85	3.31 (2018)	5,12	[53]
106020	Algeria – Hassi R'Mel	4,52	4,18	1,60-3,40 (2020)	5	[38]
108588	Germany – North Sea	3,95	3,40	4,40-4,60 (2025)	5	[90]

In the Atacama Desert, the techno-economic analysis considers only solar energy, just like the optimization results for the electrolysis-based hydrogen production model [53]. However, according the Chilean Hydrogen Association analysis [53], the LCOH is 3,31 €/kg, that is 32% cheaper than the model's optimized LCOH with a value of 4,85 €/kg. This difference can be explained by the assumptions considered in the paper, where the electrolyzer power investment costs are significantly lower, with a range of 770-1100 €/kW with a WACC of 5,12% and 20 years of lifetime [53], while the electrolysis-based hydrogen production model considers 1420 €/kW with a WACC of 3,5% and 20 years of lifetime.

Similarly, Algeria's optimized LCOH is lower than the electrolysis-based hydrogen production model results. They assume a more economical electrolyzer and wind turbines, while the PV technology costs are more expensive. Additionally, the linear optimization model published in the article considers that the produced hydrogen can be directly fed to a pipeline with an annual hydrogen demand and not hourly, which gives the model a higher hydrogen production flexibility [38].

Finally, the comparison represents a challenge for the North Sea location, as no onshore hydrogen production studies have been found. Nevertheless, the dena study for the Hy3 project, which considers offshore wind turbines for hydrogen production, might help assess the order of magnitude of the optimized LCOH. As shown in Table 5-5, dena LCOH is more expensive, which is already a good indication, as offshore wind turbines hold higher investment and operational costs than the onshore wind. Additionally, desalination systems, compressor, and pipelines from the offshore wind platform until the coast are considered, which correlate with a higher LCOH.

The dena Hy3 study focuses on a single region of the world, which corresponds to a single MERRA-2 cell from the 51677 optimized in the electrolysis-based hydrogen production model. Therefore, the dena study defines more concrete assumptions for the specific region, making the results more realistic while still clear. It is possible to consider more concrete assumptions in the electrolysis-based hydrogen production model;

however, it is not convenient as the geographical scope is worldwide, and the interpretability of the results would decline.

2. Country risk premiums (CRP)

In the previous section some effects of the country risk premiums on the LCOH were already observed. A further worldwide assessment of the country risk premiums effect on the LCOH is accomplished by comparing the first scenario, "_base_complete" with the second scenario, "_constant_wacc". Firstly, a visualization of the country risk premiums is presented in Figure 5-3. The base countries with zero country risk premium are United States, Canada, Norway, Sweden, Germany, Netherlands, Luxembourg, Lichtenstein, Switzerland, Singapore, Australia and New Zealand [16].

Two different world maps are visualized for the assessment. First, analog to the LCOH world map for the "_base_complete" scenario, an LCOH world map for the "_constant_wacc" scenario is created, see Figure 5-4. Furthermore, a world map with the absolute difference in LCOH between these two scenarios is also visualized, see Figure 5-5.

Figure 5-4 shows a noticeable change in the LCOH map, due to the absence of country risk premiums, with a higher number of locations with cheaper hydrogen production costs. Other countries also mentioned in the IRENA "Green hydrogen cost and potential" to have low LCOH in 2050, can be now be distinguished in the map, such as Argentina and North African countries, but also China and South Africa [89]. Additionally, Kenia and Mozambique are countries that stand out for their unpredicted low LCOH.

The country risk premiums capture the effect of the country's economic and political situations on the hydrogen production cost. For example, Argentina and North African countries, with high renewable energy potential, entail a higher economical a political risk, making them more unattractive for international investments in electrolysis-based hydrogen production systems. This means that the country risk premiums can shape the economic LCOH distribution worldwide by excluding the countries with high country risk premiums.

Figure 5-5 shows the absolute difference between the "_base_complete" scenario, which includes the country risk premium, and "_constant_wacc", which employs the same WACC, 3,5% for all the countries. The higher the country risk premium, the higher the absolute difference. In Central and South America, Africa and Central Asia, the absolute difference in LCOH frequently reaches more than 5€/kg. In countries with a country risk premium higher than 10%, such as Argentina, the LCOH is at least twice the cost without the country risk premium consideration.

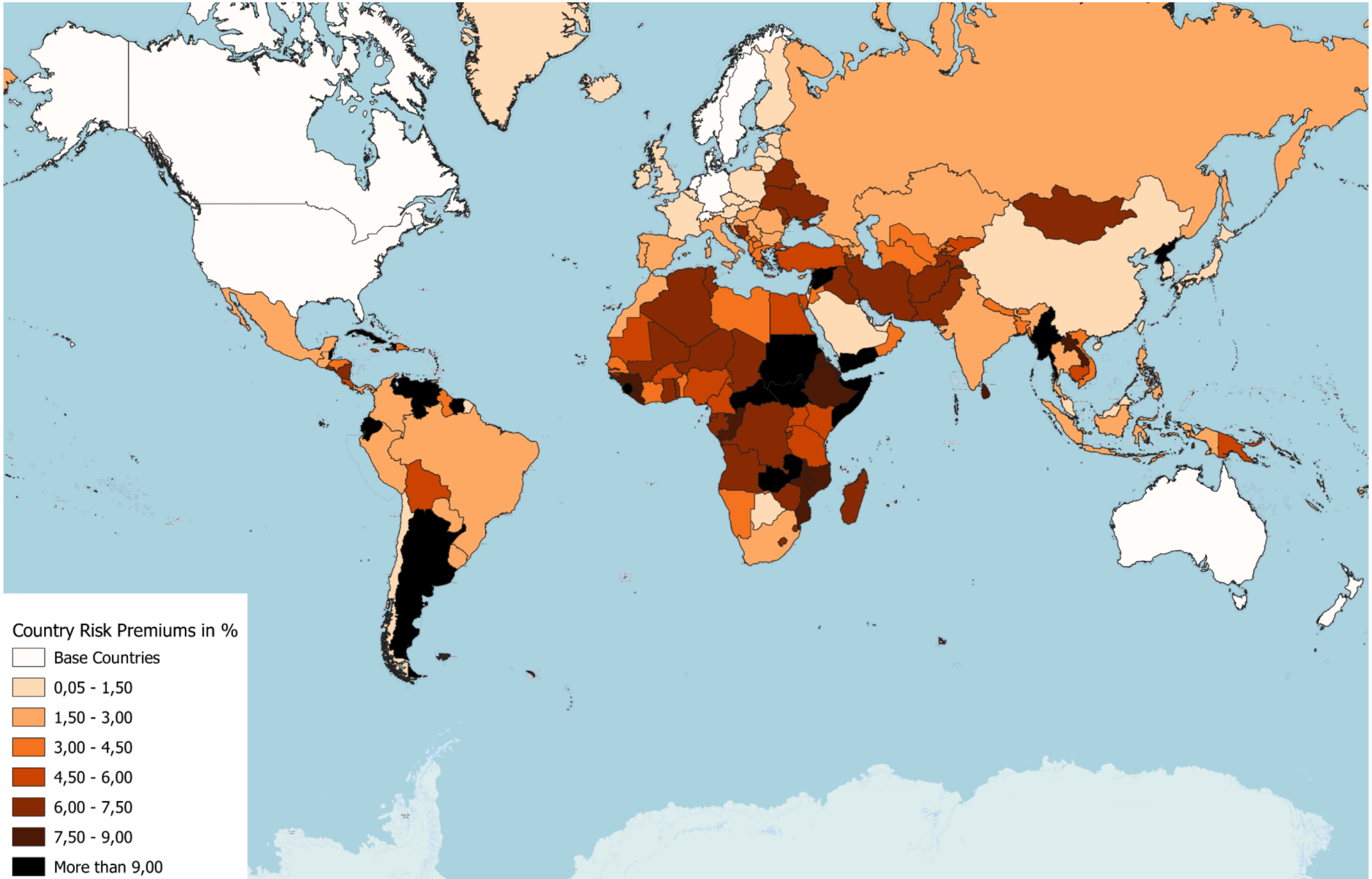


Figure 5-3 Country Risk Premiums map

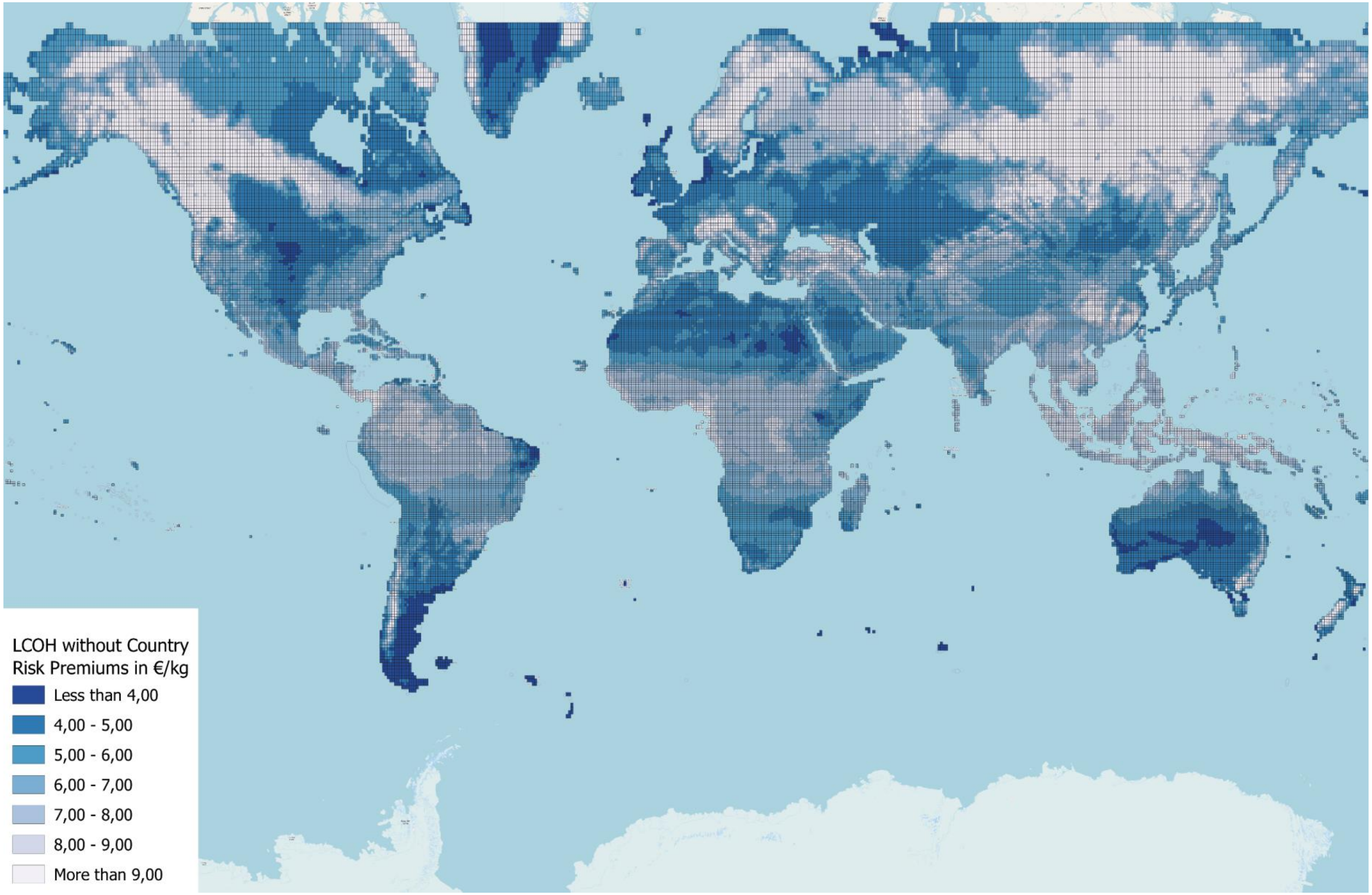


Figure 5-4 LCOH for the _constant_wacc scenario not including country risk premiums

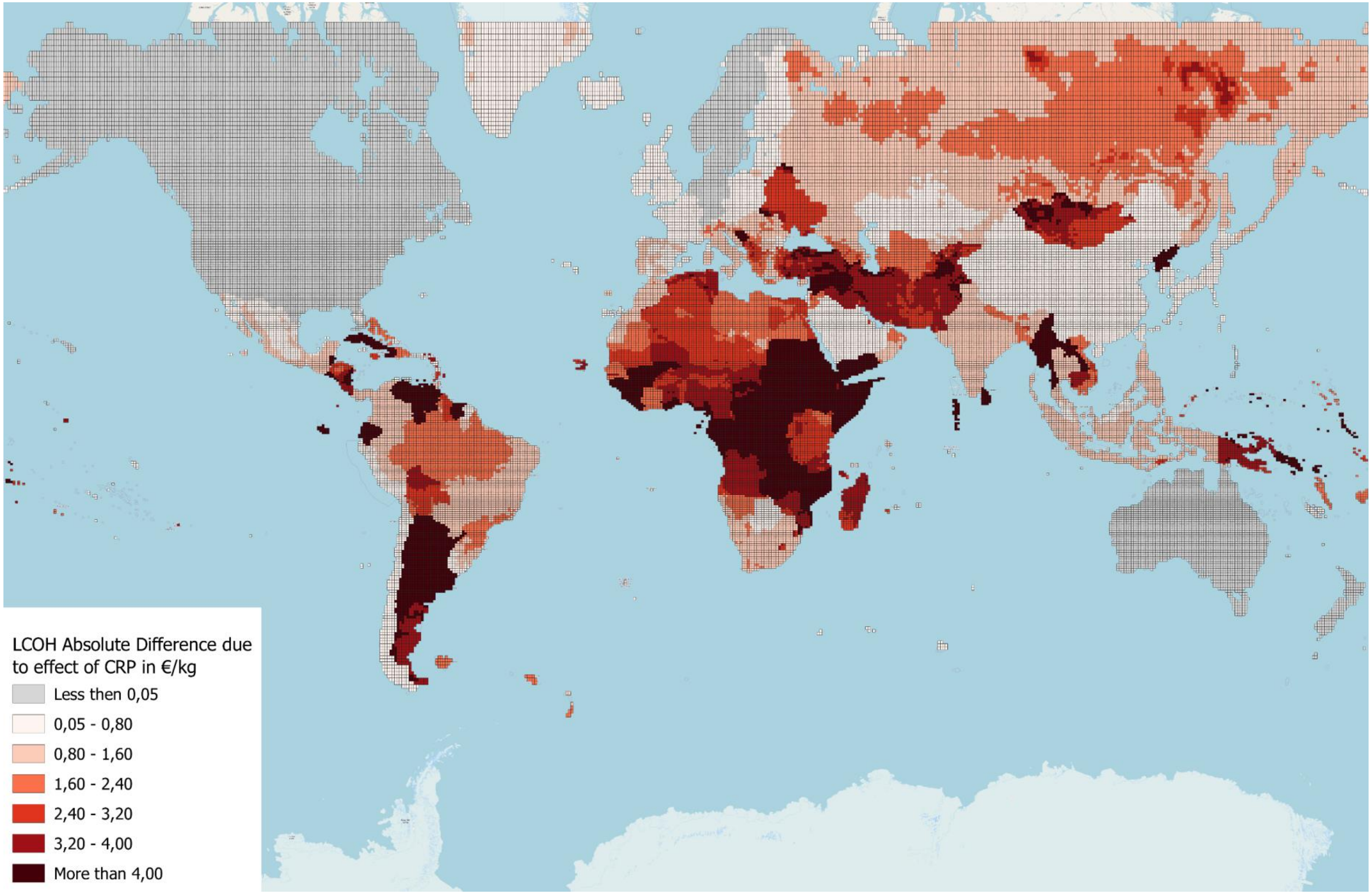


Figure 5-5 LCOH absolute difference between _base_complete and _constant_wacc scenarios

The previous visualizations showed the global effect of the country risk premiums. Now single country analysis and comparison between countries with similar country risk premiums are conducted. In particular, the effect of the country risk premiums for Argentina alone is analyzed, and then Algeria and Turkey, which hold a country risk premium in the same range.

Argentina's country risk premium is the second highest in South America and amounts to 11,87%, below Venezuela with 20,34% and above Ecuador and Suriname with 9,89% each. This high country risk premium makes Argentina unattractive from an investment perspective, as the LCOH increases strongly. As depicted in Figure 5-6 (b), this absolute difference of LCOH between the two scenarios "_base_complete" and "_constant_wacc", for Argentina varies between 3,39€/kg and 11,12 €/kg.

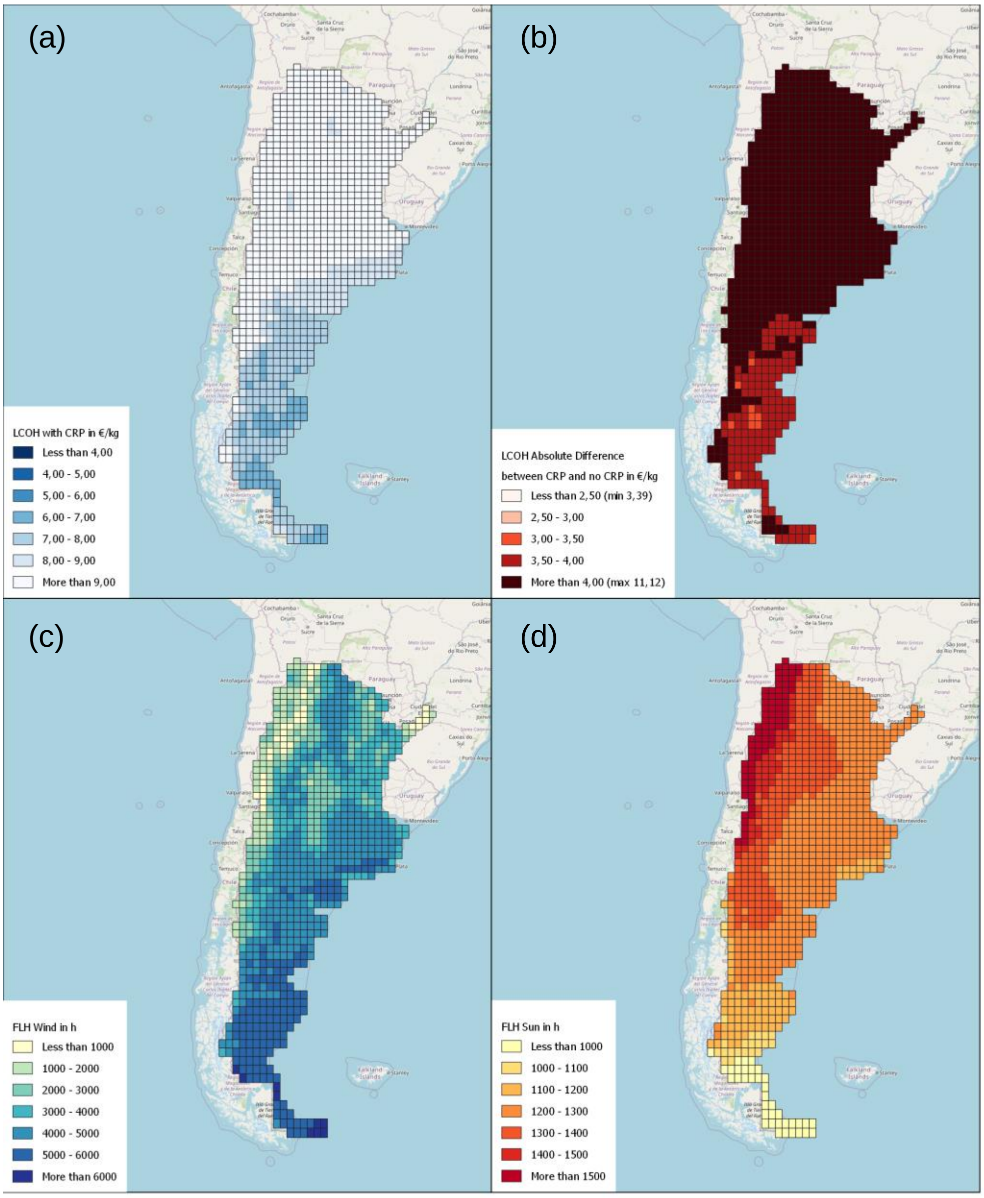


Figure 5-6 Argentina's country risk premium effect

This high variation in LCOH can be explained by the existing renewable energy potential in terms of full load hours (FLH). The subfigures (c) and (d) from Figure 5-6 represent Argentina's wind FLH and solar FLH, respectively. While the solar FLH do not seem to have an obvious correlation with the LCOH absolute difference, the wind FLH do. This correlation is preceded on the fact that the higher the wind FLH, the smaller the installed powers and capacities for the different components.

Further explaining this correlation, in high wind FLH conditions, the available wind is steadier; therefore, no high installed wind power is needed to supply the electrolyzer. Similarly, since the electricity available is then steadier, the size of the electrolyzer and hydrogen storage system can be reduced, as there is no need to produce more hydrogen for storage to meet the demand on later timesteps.

In essence, low wind FLH are associated with the decoupling of the production and supply of hydrogen, therefore needing bigger sizes for the wind turbines, electrolyzer and hydrogen storage. On the other hand, high wind FLH enable the coupling of the production and supply of hydrogen, which means smaller component sizes and, therefore, a lower LCOH, having high electrolyzer FLH. As a key result, high wind FLH can compensate the country risk premium negative effect, especially in countries with high country risk premium, as is the case for Argentina.

This compensation can also be easily recognized by comparing Algeria and Turkey, which have 6,43% and 5,44% country risk premiums, respectively. This comparison is accomplished through Figure 5-7 and Figure 5-8, containing the information with the same scale for Argelia and Turkey, respectively.

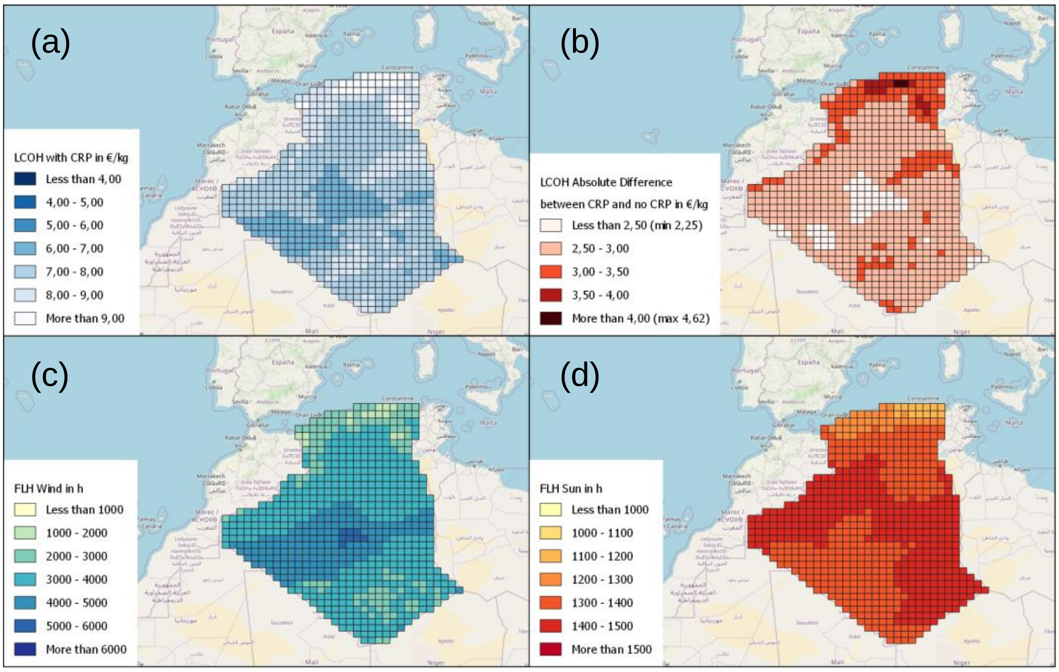


Figure 5-7 Algeria's country risk premium effect

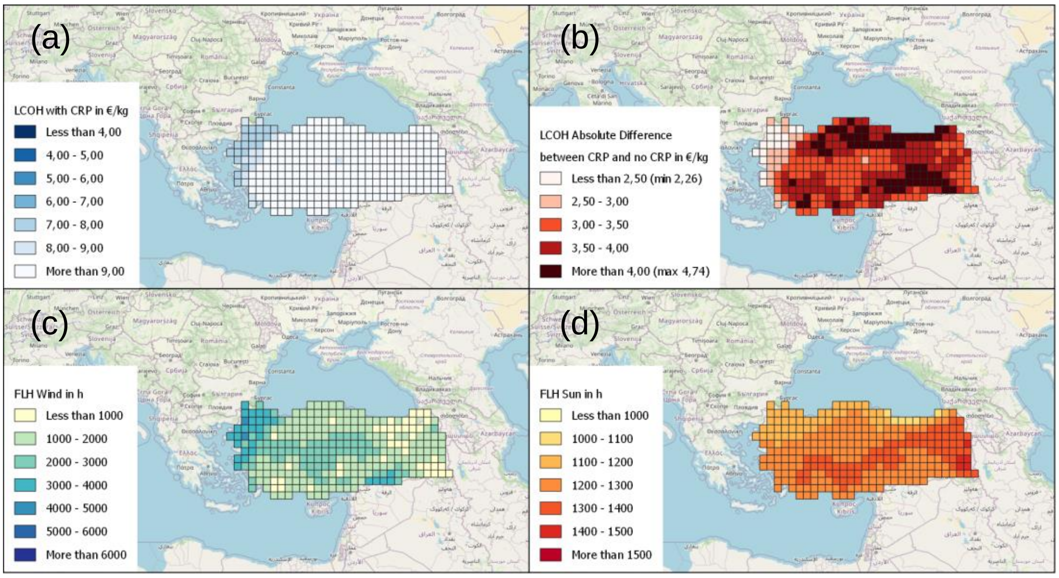


Figure 5-8 Turkey's country risk premium effect

Despite Algeria having a higher country risk premium, its LCOH is lower, as observed comparing Figure 5-7 (a) and Figure 5-8 (a). The explanation again is the compensation of the country risk premiums negative effect, through to the high wind FLH. Algeria has a higher wind potential than Turkey, and therefore, the optimized electrolysis-based hydrogen production systems are smaller in Algeria and less sensitive to high country risk premiums.

Similarly, is the LCOH absolute difference due to the country risk premium effect higher in Turkey, see Figure 5-8 (b) compared to Figure 5-7 (b). Not a significant correlation can be identified from the visualization of the solar FLH of both countries.

Even comparing the LCOH visualizations for Argentina and Algeria, with the same scale, it can be appreciated that for both countries the highest LCOH interval is 6,00 €/kg to 7,00€/kg, even though Argentina's country risk premium is much higher, 11,87% against 6,43% for Algeria. Consequently, the high wind potential from Argentina can compensate its unattractive investment situation characterized by its country risk premium.

It is also important to mention that the evolution of the future country risk premiums can hardly be predicted. Therefore, in case the electrolysis-based hydrogen production model was to be used for future calculation scenarios, 2030, 2040 or 2050, instead of 2020, a constant WACC for all countries would be a valid assumption for the comparison of the LCOH unless reliable future country risk premiums are available.

3. Hybrid PV-Wind electricity generation

The scenario "only_pv" and "only_wind" are calculated to analyze the effect of a hybrid PV-wind configuration on the LCOH. Different visualizations of the results are presented. Figure 5-9 represents the distribution for the share of PV installed power in each optimized cell, according to the next equation.

$$Share\ of\ PV\ installed\ power = \frac{PV\ installed\ power}{(PV\ installed\ power + Wind\ installed\ power)}\ in\ \%$$

The visualization shows in yellow the locations with only PV systems and in blue the cells where exclusively wind turbine is installed. The transitional hybrid systems are represented in greens. From the visualization, most optimized systems are PV based.

Additionally, Table 5-6 gives the number and share of cells with a specific share of PV or wind installed power. It is then corroborated that most optimized locations are PV-based, 77,51% of all the cells (more than 50% share of PV installed power). Also, while the locations with only wind turbines as electricity generation systems represent 2,88% of the total cells, the locations with exclusively PV systems come up to 20,78%. Altogether it can be deduced that the PV systems prevail over the wind energy systems, and more than three-quarters of the cells have a hybrid system.

Table 5-6 Number and share of cells with a specific share of solar or wind energy installed power

Share of installed power	Number of cells	Share of cells
Only Wind (<5% share of PV)	1611	3,12%
Only PV (>95% share of PV)	11551	22,35%
Hybrid Systems	38515	74,53%
>50% share of PV	40054	77,51%
>60% share of PV	30620	59,25%

Additionally, Figure 5-10 represents in a world map the absolute difference of LCOH between the scenario "_base_complete" where the hybrid PV-wind configuration is modelled, and the minimum LCOH between the "_only_pv" and "_only_wind" scenarios. The darkest purple locations on the map represent the highest LCOH absolute reductions, which designate the areas where the hybrid PV-wind configuration embodies the most advantageous potential compared to only PV or wind turbine systems.

These two world maps suggest on the one hand, that hybrid systems are usually introduced and, on the other hand, that hybrid systems reduce the LCOH. However, these statements must be complemented to assess if the frequently introduced hybrid PV-wind systems truly entail a competitive advantage on the LCOH over non-hybrid configurations (only PV or only wind) and, if so, how significant the benefit is.

The bottom line is whether hybrid configurations are economically more competitive than non-hybrid configurations.

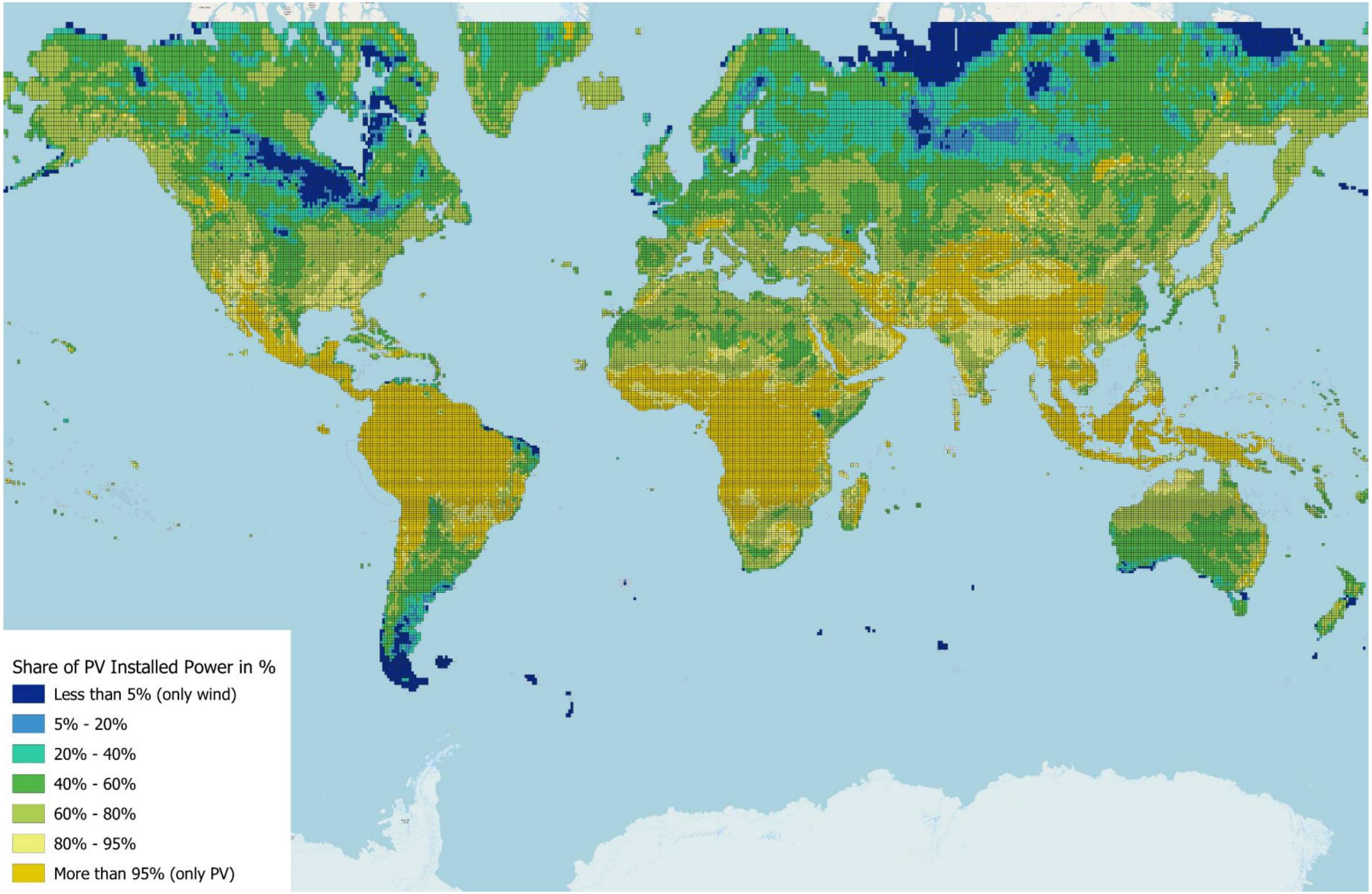


Figure 5-9 Optimized share of PV installed power worldwide

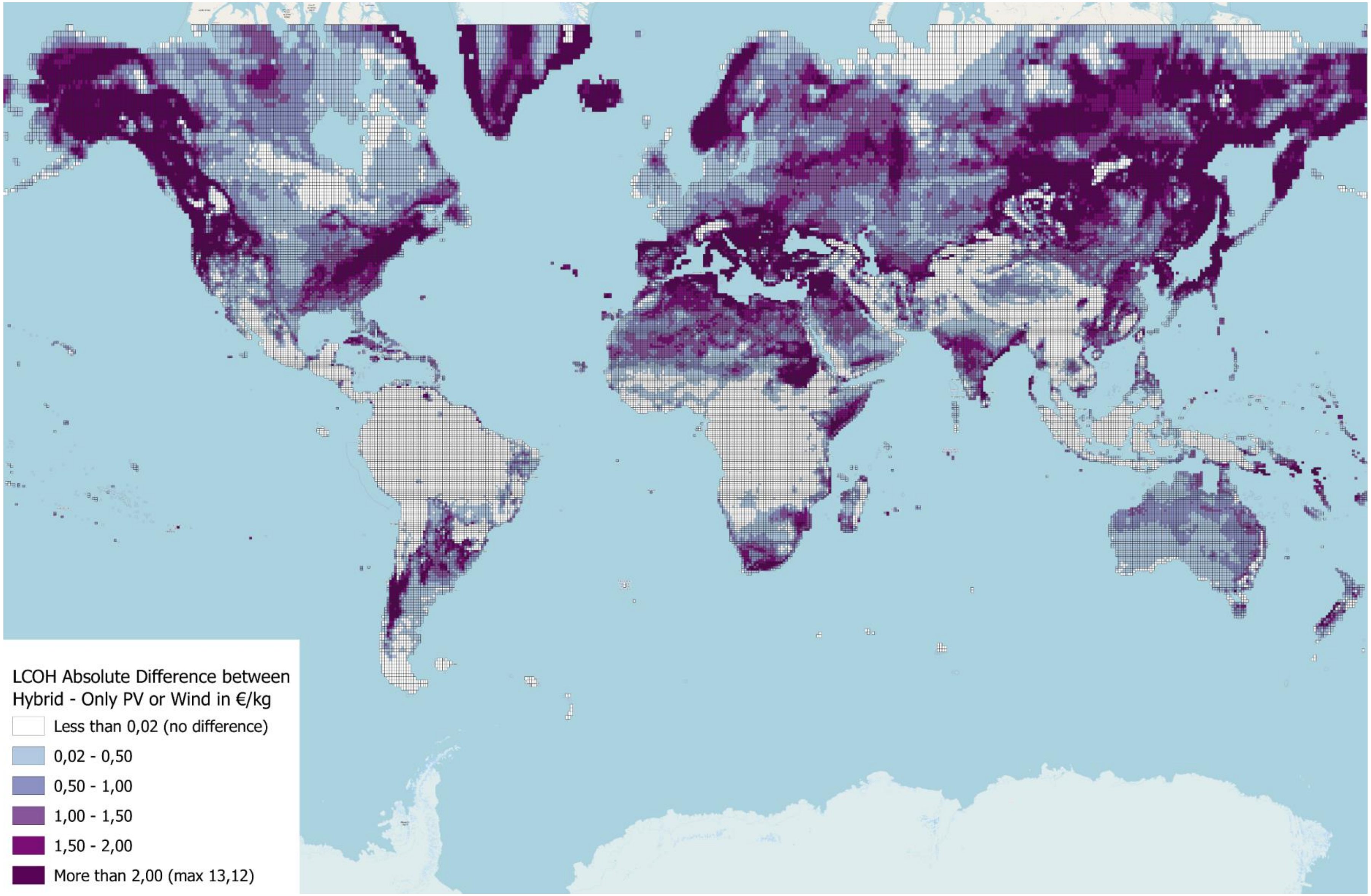


Figure 5-10 LCOH absolute difference between hybrid PV-wind and only PV or only wind

The complementary assessment is done primarily through two different scatter diagrams.

Figure 5-11 represents a density scatter diagram for the share of PV installed power against the LCOH from the hybrid scenario. The right bar represents the number of cells with the specific share of installed PV (y-axis) and LCOH (x-axis). From Figure 5-11 three clusters can be identified, the only PV cluster (>95% share of PV) with 11551 cells, the hybrid PV-wind cluster (40% to 70% share of PV) with 25474 cells, and finally, the only wind cluster (<5% share of PV) with 1611 cells. The only PV cluster is not economically competitive with the other two clusters, as it holds the highest LCOH.

Between the hybrid cluster and the only wind cluster, the last one has lower LCOH, with a minimum of 2,31€/kg and 154 cells under 4€/kg. In the case of the hybrid cluster, the minimum LCOH is 3,29€/kg, corresponding to a PV share of 41,48%. In this cluster, the number of cells under an LCOH of 4€/kg is 441. Also, for the hybrid cluster, 2205 cells have an LCOH below 4,5€/kg and up to 4496 cells below 5€/kg.

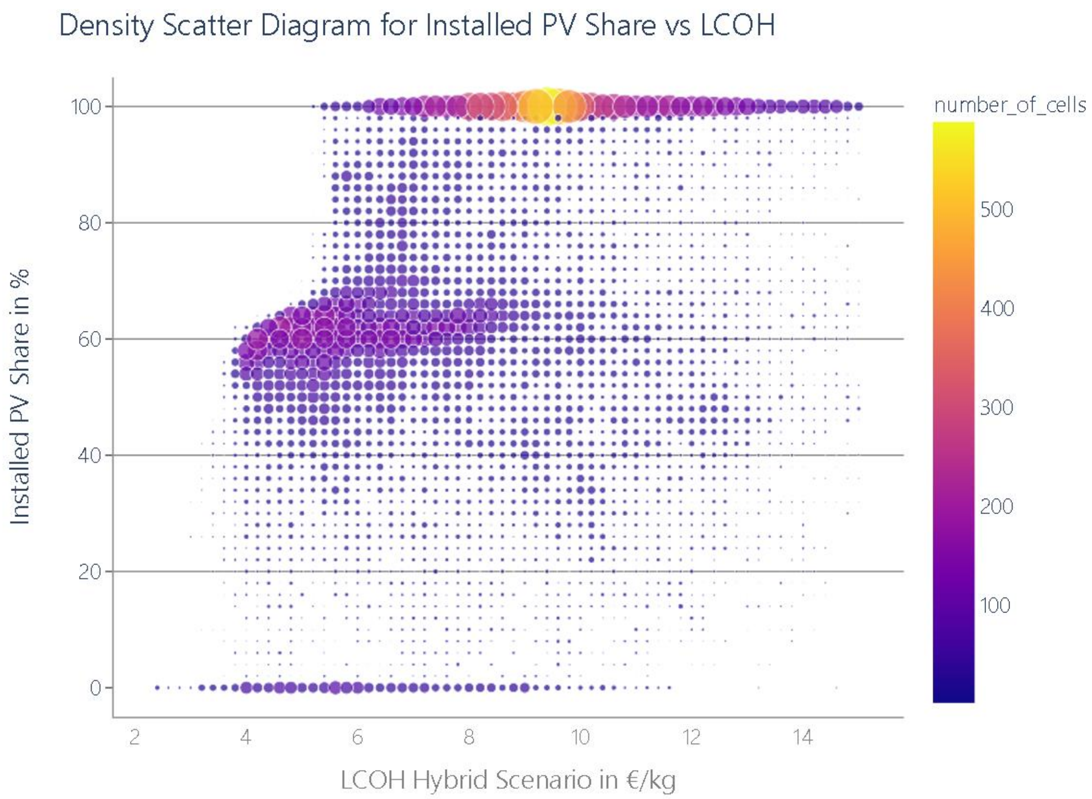


Figure 5-11 Density scatter diagram for the installed PV share against the LCOH Hybrid

A summary of the number of cells in each for each LCOH range is presented in Table 5-7. The transition range between the hybrid and wind clusters is also included.

Table 5-7 Summary of LCOH for the hybrid, wind, transition, and PV clusters

Cluster	Wind Cluster (0% - 5% PV share)	Transition (5%-40% PV share)	Hybrid cluster (40%-70% PV share)	PV cluster (95%-100% PV share)
LCOH\Total	1611	5273	25474	11551
< 3€/kg	19	2	0	0
3€/kg – 3,5€/kg	49	24	13	0
3,5€/kg – 4€/kg	86	131	428	0
4€/kg – 4,5€/kg	128	283	1764	0
4,5€/kg – 5€/kg	163	290	2291	0
Total < 5€/kg	445	730	4496	0

Even though hybrid cells have a slightly higher LCOH, the higher number of locations and a low LCOH might represent an advantage against the only wind cluster.

The question of how significantly the hybrid configuration affects the LCOH is answered through Figure 5-12, which represents a scatter diagram for the LCOH absolute difference between hybrid and non-hybrid configurations against the LCOH for the hybrid scenario.

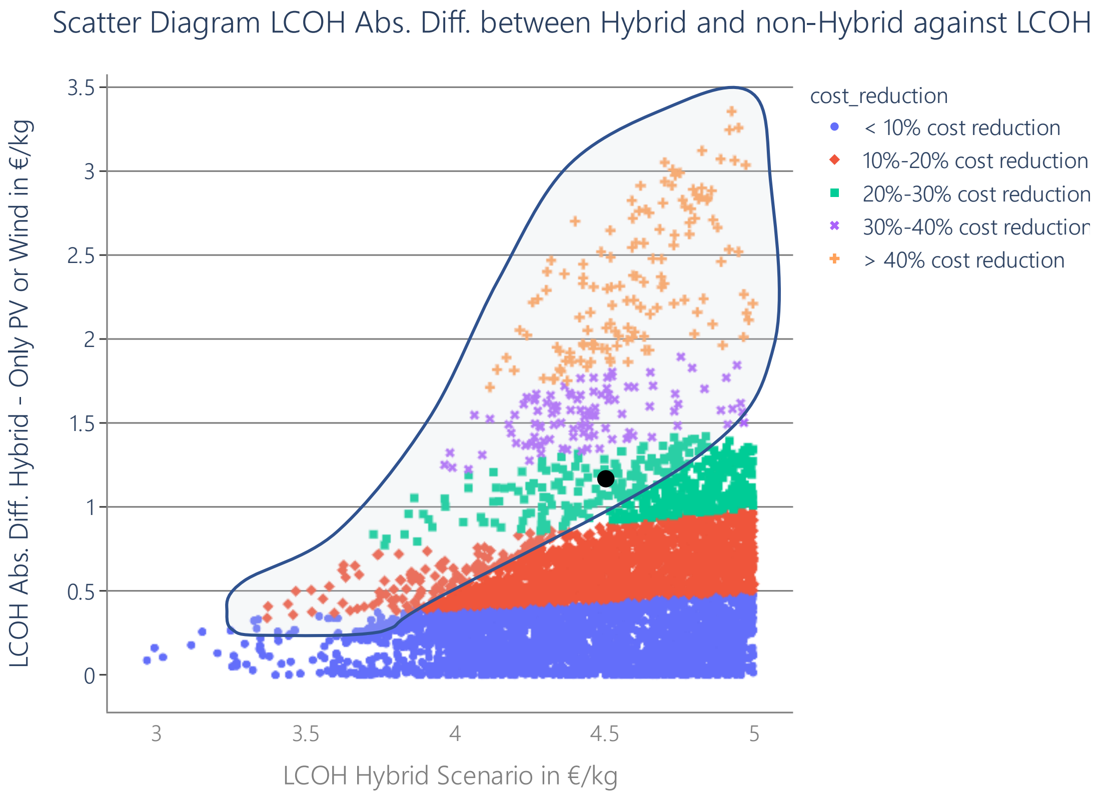


Figure 5-12 Scatter diagram for the absolute difference of LCOH between hybrid and non-hybrid configurations against the LCOH Hybrid

In Figure 5-12, the relative cost reduction for each cell is also illustrated through the different colors. The cells inside the shaded shape experience a competitiveness increase for developing an electrolysis-based hydrogen production system, thanks to the economic advantage provided by the hybrid PV-wind configurations, that is, a strong reduction of the LCOH. These cells are in four regions of the world, Greenland, the United States, Saudi Arabia, and Australia. This improvement can be easily exemplified by looking at the black point in Figure 5-12. This point represents cell 130191, located in the Saudi Arabian desert, with an LCOH for hybrid configuration of 4,48 €/kg and an LCOH absolute difference between hybrid and non-hybrid configurations of 1,20€/kg. For this cell, the hybrid configuration entails a cost reduction of 21,16%, taking the cell into a more economically competitive position, from 5.68€/kg to 4,48€/kg.

In brief, with the current model assumptions, the results show that hybrid PV-wind cells are commonly found in the optimization results, and the hybrid configuration can significantly reduce the LCOH, making cells economically competitive. Also, wind-based cells are usually more competitive than solar-based cells.

4. Battery storage system

As mentioned in 5.1, the pre-analysis results showed that no battery was introduced for the 100 random optimized cells, and therefore, instead of a scenario with no modelled battery, the "_affordable_battery" scenario with a 50% reduction of the battery annualized costs is introduced.

The results for the "_base_complete" scenario show that only 15 cells (<0,03% of all the optimized cells) include a battery storage system. This absence of battery storage systems in the optimization results indicates that the costs for the battery storage technologies are too high to bring an economical advantage to the system.

The cost of the battery is so high that curtailment is preferred over storing the produced electricity for future timesteps. Table 5-8 contains the average curtailed electricity percentage per cell and technology, considering only the cells with installed power for the specific electricity generator. The PV technology has the highest curtailment percentage, as on average more than a quarter of all the electricity produced from PV is curtailed in every system cell. The curtailment is due to solar energy peaks, not so frequent on wind energies, which is the reason behind the lower curtailment in the very weak and very strong wind turbines.

Table 5-8 Average percentage of curtailed electricity per technology

Electricity generator	Average Curtailed electricity pro cell
PV	27,51%
Very strong wind turbine	12,31%
Very weak wind turbine	11,91%

In the "_constant_wacc" scenario, no battery was introduced due to the interest rate (or WACC) and lifetime correlations with the annuity factor. Further explaining this, it is important to remind that low annuity factors mean high annual costs; see 2.1.2. Table 5-9 shows how by high interest rate (such as "_base_complete" for countries with high country risk premium), the effect of the lifetime is not so significant, as the annuity factor will always be low. In that case, the battery system has a similar annuity factor to the other technologies, and therefore, it is more economically viable to be introduced in the optimizations. However, in the case of low interest rate (such as in the "_constant_wacc" scenario or for the countries with low country risk premium in the "_base_complete" scenario), the component's lifetime gains significance, having the lower lifetime technologies also lower annuity factor, and therefore, given their high annual costs, the component will not be introduced in the optimized system. That is the case for the battery storage system in the "_constant_wacc" scenario.

Table 5-9 Annuity factors for representative interest rates (r) and lifetimes (n) values

	Interest rate (r)	Lifetime (n)	Annuity factor (AnF)
Case 1	3,5%	10	8,31 %
Case 2	3,5%	20	14,21 %
Case 3	13,5%	10	5,32 %
Case 4	13,5%	20	6,82 %

As no battery is introduced in the optimization results, the hydrogen storage system is used as the main flexibility technology to achieve the decoupling of hydrogen production and supply of the demand. This decoupling is needed in most locations as electricity is not always available for every timestep.

The research question regarding the effects of the battery storage system on the LCOH is then already answered; the battery has no effect on the LCOH optimization, given its high investments and operational costs. Nevertheless, a small battery will probably be installed in a real project to address small electricity deviations within each hour. This is reasonable as the model considers an hourly time resolution, and what happens in the minute range is out of the model's scope.

The optimization results for the "_affordable_battery" scenario are shown in Figure 5-13 and Figure 5-14. Firstly, Figure 5-13 shows the installed capacity for the affordable battery storage systems in hours from the demand side perspective, that is, how many hours can the demand for hydrogen be met with a full capacity battery. The maximum installed capacity is 18h and 27min. Comparing Figure 5-13 with the map for the share of PV installed power, Figure 5-9, the battery storage system is only introduced in locations with a high share of PV installed power, which are usually not economically competitive. This result is logical as the high PV share locations depend highly on the solar full load hours, and the battery storage system allows the electricity storage for hours without sun.

Additionally, Figure 5-14, shows the LCOH absolute difference between the "_base_complete" and "_affordable_battery" scenarios, where the LCOH reduction is a maximum of 1,77€/kg. From both visualizations, the largest battery capacities and the most noticeable LCOH cost reductions are found in countries with the highest country risk premiums, such as Ecuador, Venezuela, or South Sudan. This effect is explained through the above-mentioned correlation between the battery storage system lifetime and the country risk premiums with the annuity factor. The short component's lifetime effect for high country risk premiums is not so significant in the annuity factor as for low country risk premiums. Therefore, the annuity factor for the battery system (ten years of lifetime) and other components (at least 20 years of lifetime) are in a similar range; see Table 5-9. This means that for high country risk premiums, the shorter lifetime of the battery does not represent a big disadvantage in its annualized costs as in low country risk premiums. Therefore, halving the battery cost has a more significant effect on the regions with high country risk premiums, where the annuity factors between the battery system and other technologies are similar.

The "_affordable_battery" scenario conclusion is that modelling an affordable battery with 50% cost reduction does not entail any change in the worldwide distribution of the economically competitive LCOH, and no new economically competitive locations are encountered.

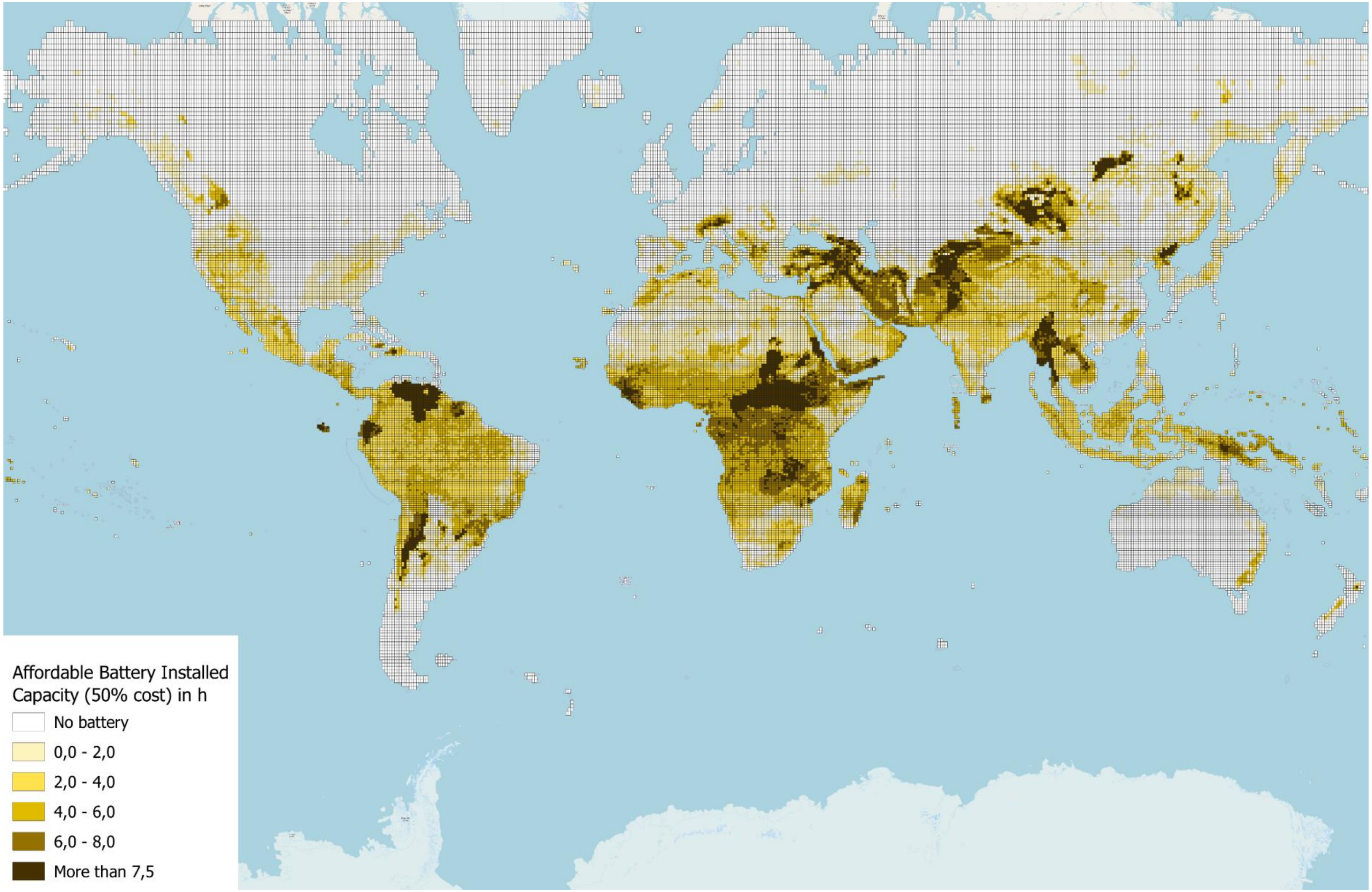


Figure 5-13 Affordable battery storage (50% cost reduction) installed capacity for "_affordable_battery" scenario

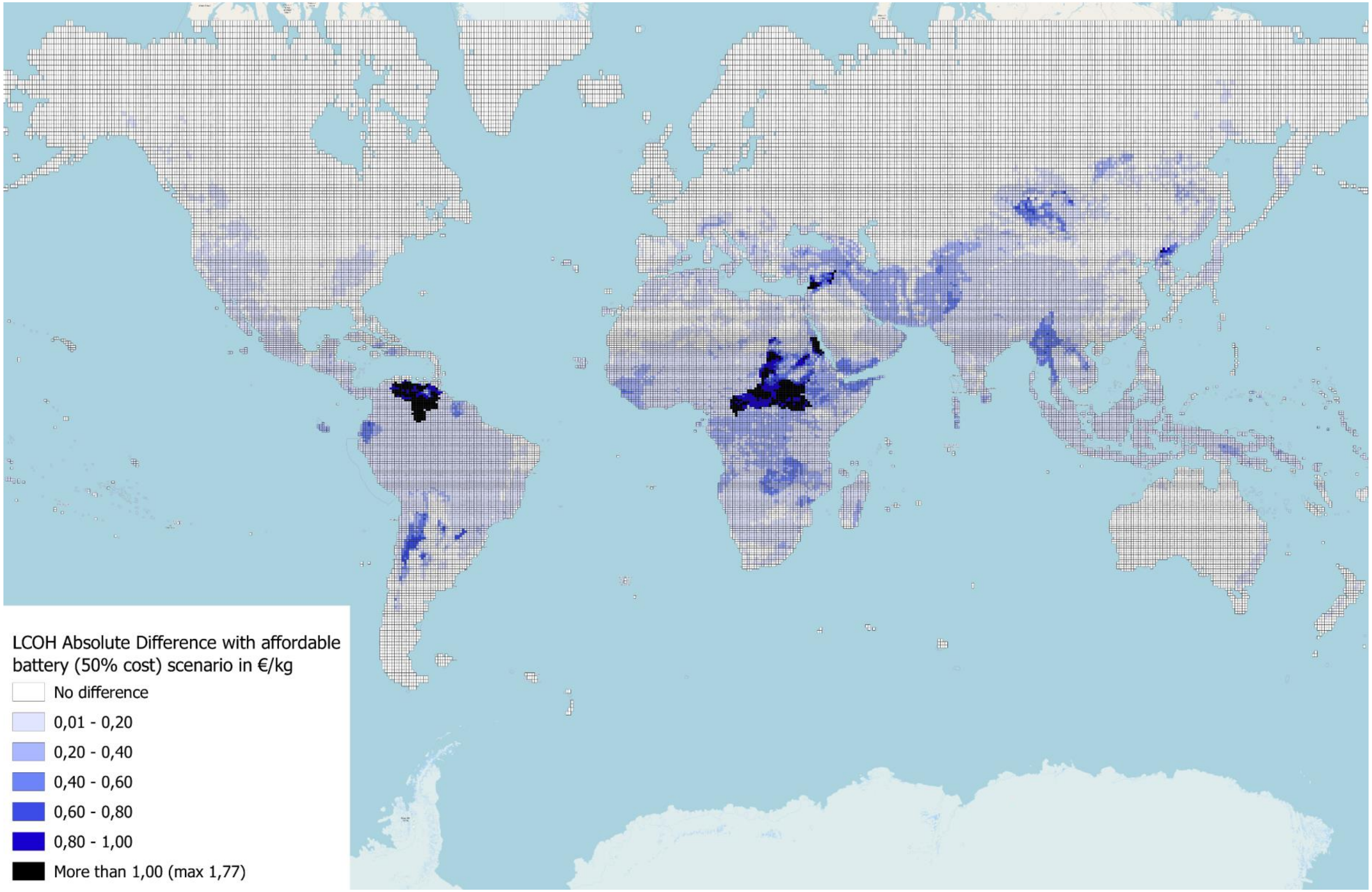


Figure 5-14 LCOH absolute difference between "_base_complete" and "_affordable_battery" scenarios

5. Hydrogen storage system

For the "_base_complete" scenario, a hydrogen storage system is introduced in every optimized cell. Hydrogen storage enables the decoupling of hydrogen production and hydrogen supply. This decoupling is needed since not enough electricity is available in every timestep, and there no battery storage system is introduced in the optimizations. Consequently, the hourly hydrogen demand can be met thanks to the hydrogen storage system.

Figure 5-15 illustrates the installed storage capacity in hours from a demand side perspective. It is observed that most hydrogen storage systems have a capacity under 480 hours or 20 days. At the bottom of Figure 5-15, only the competitive cells with a LCOH under 5€/kg are shown with hydrogen storage capacities under 240 hours or 10 days. Most hydrogen storage systems hold a minimum capacity of 50 hours. Long-term storage of hydrogen is not frequent in the results but only in some specific locations, which should be further analyzed to assess its plausibility.

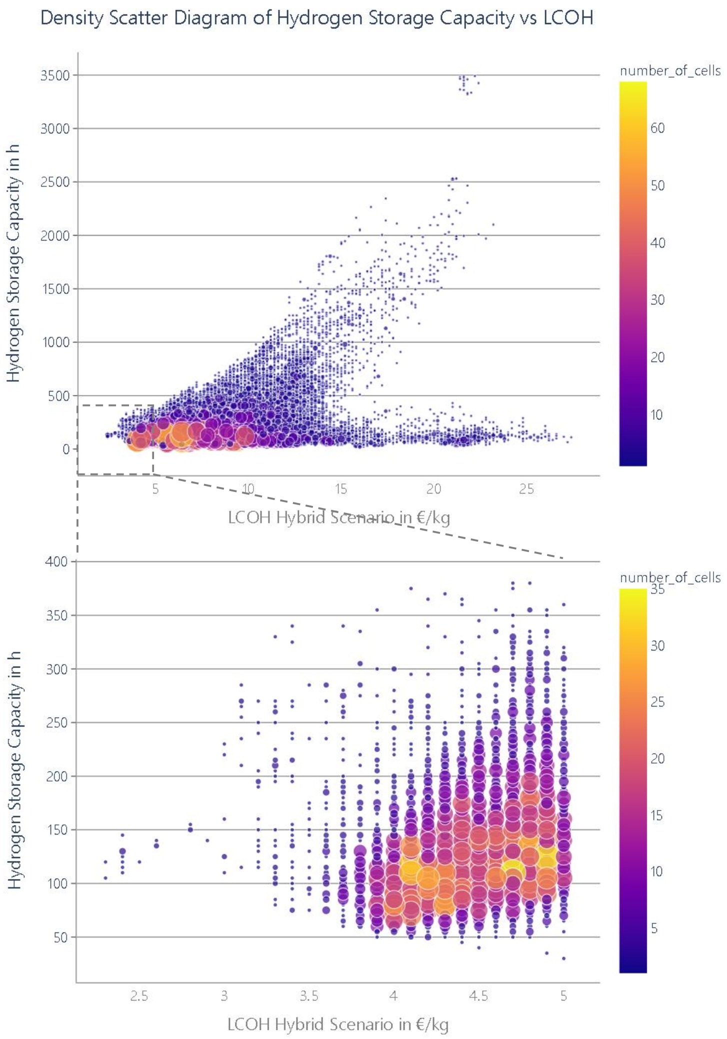


Figure 5-15 Scatter diagram for the installed hydrogen storage capacity against the LCOH Hybrid

6 Conclusions and outlook

This thesis focuses on the development of an electrolysis-based hydrogen production model using an open-source environment to minimize and analyze the levelized cost of hydrogen (LCOH) worldwide. The outcomes of this thesis are insights concerning the methodology for the hydrogen production model implementation and the analysis of the optimization results.

The following conclusion section reviews the key points regarding the research questions. Afterward, the possible future work for the development of the model is discussed in the outlook section.

6.1 Conclusions

The findings of this thesis aim to answer three research questions. The first research question addresses the choice of the appropriate open-source software for the development of the model and is answered through a qualitative assessment. The other two research questions embrace the effects of the country risk premiums, battery storage system, and hybrid PV-wind configuration on the levelized cost of hydrogen (LCOH). These two questions are addressed by implementing a quantitative assessment through the development of the hydrogen production model with linear optimization.

The conclusions are then divided into qualitative and quantitative parts.

Conclusions for the qualitative assessment

The selection of the appropriate open-source software requires a qualitative assessment of the existing open-source environments. Certain assessment criteria are identified for this task, concerning the suitability, automatability and flexibility, together with general key aspects. Respectively these criteria cover if the devised system can be modelled with the software, if the optimizations can be automated for multiple optimizations altering the input data and if the model is easily adjustable in case new assumptions are to be adopted. The general aspects concern mainly the update frequency of the software and the existence of a community behind it.

Several open-source software was initially discarded, and a deeper analysis was implemented for urbs and PyPSA. Despite neither having clear exclusion points, PyPSA was finally chosen for developing the electrolysis-based hydrogen production model.

Conclusions for the quantitative assessment

The implementation of the electrolysis-based hydrogen production model requires a structured approach. The first step in this methodology involves the definition of the hydrogen production system, including the components, the relationships among them, the assumptions, and the constraints. Once the system is devised, the needed input data are identified and classified in different levels, as specific data for a single optimization (cell level), for a group of optimizations (country level), or for all the optimizations (system level).

Additionally, a consistent structure for storing and processing the input and output data is created in the FfE database to easily access the data during the optimizations process and the analysis of the results. Then the program code is prepared with a meaningful module structure that embodies the program's flow. Also, strategies for improving the runtime performance of the program are implemented.

Once the programming code is finished, the methodology's final step is properly defining the calculation scenarios so that the results are enough to answer the research questions. The base scenario includes PV, very weak and very strong wind turbines, electrolyzer, battery storage, and hydrogen storage systems. The base scenario also considers the country risk premiums.

The optimizations resolved for the different calculation scenarios lead to the data needed for the quantitative assessment. These data are then processed, visualized mainly on world maps, and employed to analyze the effects on the LCOH of the different components and assumptions. In the following points, the results are shortly reviewed.

- **The distribution of the lowest LCOH locations is consistent with other studies.** Under the consideration of the country risk premiums, the locations with the cheapest LCOH are the United States, Canada, Australia, south of Chile, the North Sea in Europe, and the north of Brazil. Greenland also stands outs for its low LCOH, but no other studies that corroborate this result were found. Although with higher LCOH, Saudi Arabia also holds a good potential for electrolysis-based hydrogen production.
- **The country risk premiums have a big impact on the LCOH distribution worldwide.** Without considering the country risk premiums, other locations with inexpensive electrolysis-based hydrogen potential are identified, such as south of Argentina, Kenia, and North African countries. The hydrogen production potential for these countries is high, but they embody low investment interest due to their political and economic situation reflected in the country risk premiums.

- **The negative effect of the country risk premiums on the LCOH can be compensated with high full load hours of wind.** South Argentina region exemplifies this effect.
- **The PV-wind hybrid configuration for electricity generation is more common than only PV or wind**. 74,53% of all optimized cells have installed power of both electricity generation technologies.
- **Most locations are PV-based.** 77,51% of all optimized cells hold at least 50% of installed PV in their total installed power for electricity generation.
- **Three clusters of cells concerning the share of installed PV and the LCOH are identified, only wind, hybrid, and only solar; see Figure 5-11.** The only wind cluster (<5% PV share) is the smallest including 3,12% of the cells, but also with the lowest LCOH. Then the hybrid cluster (40%-70% PV share) is the largest cluster, with 49,29% of all the cells. This cluster has a relatively low LCOH which can compete with the wind cluster in some locations. Finally, the PV cluster (>95% PV share), with 22,35% of all the cells, holds the highest LCOH and is not competitive in most locations. The transition area between the wind and hybrid clusters also contains economically competitive locations.
- **Certain locations reach an economically competitive position thanks to the hybrid configuration.** These cells are located in the United States, Australia, Saudi Arabia, and Greenland.
- **The battery storage does not bring an economical advantage to the system due to its high costs.** Curtailment is preferred over installing a battery. Considering a battery system with a 50% cost reduction can lower the LCOH for PV-based cells. However, this LCOH decrease is not enough to make these locations economically competitive.
- **Hydrogen storage represents a flexibility component to meet the demand in every timestep.** The hydrogen storage system enables the decoupling of hydrogen production and hydrogen supply of the demand.

In conclusion, the results of the quantitative assessment show, on the one hand, that the hybrid PV-wind configuration for electricity generation and the country risk premiums hold respectively positive and negative significant effects on the LCOH. On the other hand, the battery storage system does not affect the LCOH, given its high costs.

6.2 Outlook

The further development of the electrolysis-based hydrogen production model can take different approaches.

- Expand the hydrogen production system with:
  - Further components such as offshore wind turbines, compressor, different kind of electrolyzers (AEL, SOEC additional to PEM), desalination system and heat production system (in case of SOEC).
  - Adjustment or flexibilization of the constant hourly hydrogen demand (Demand Side Management).
  - Input MERRA-2 weather data for different years (now, only 2012), so that the results are consistent independently of the weather conditions of a concrete weather year.
- Include additional calculation scenarios considering other years (now, only 2020 techno-economic parameters), using techno-economic parameter estimations for 2030, 2040 and 2050 to see the development of the LCOH worldwide.
- Include land use potential analysis and calculate cost-potential curves per country.
  - Exclude, among others, nature reserves and conservation areas, water scarcity regions, high-density populated areas, and steep slopes regions.
  - Obtain percentage of the still available land to be used for hydrogen production.
- Include synthetic fuel production in the model.
- Analyze if selling the produced electricity in each timestep is more profitable than using it to produce hydrogen. Additionally, analyze if marketing the curtailed electricity could significantly reduce the LCOH in certain locations.

7 References

[1] Hydrogen Council, “Hydrogen decarbonization pathways: Introduction to the two reports: “Hydrogen Decarbonization Pathways: A Life-Cycle Assessment" and " Hydrogen Decarbonization Pathways: Potential Supply Scenarios”,” Hydrogen Council, Brüssel, Jan. 2021. Accessed: Mar. 4, 2022. [Online]. Available: https://hydrogencouncil.com/wp-content/uploads/2021/01/Hydrogen-Council-Report_Decarbonization-Pathways_Executive-Summary.pdf

[2] Nationaler Wasserstoffrat, “Wasserstoff Aktionsplan Deutschland 2021–2025,” Berlin, Jul. 2021. Accessed: Mar. 4, 2022. [Online]. Available: https://www.wasserstoffrat.de/fileadmin/wasserstoffrat/media/Dokumente/2021-07-02_NWR-Wasserstoff-Aktionsplan.pdf

[3] S. Kigle, T. Kern, A. Murmann, A. Bruckmeier, C. Wendlinger, and K. Ganz. “ISAaR: Integriertes Simulationsmodell zur Anlageneinsatz- und Ausbauplanung mit Regionalisierung.” https://www.ffe.de/tools/isaar/ (accessed Mar. 4, 2022).

[4] Institute, Corporate Finance, “Levelized Cost of Energy (LCOE),” Jan. 2020. Accessed: Apr. 24, 2022. [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/finance/levelized-cost-of-energy-lcoe/

[5] C. Kost, S. Shammugam, V. Fluri, D. Peper, A. D. Memar, and T. Schlegl, “Study: Levelized Cost of Electricity Renewable Energy Technologies,” Fraunhofer Institute for Solar Energy Systems ISE, Freiburg, Jun. 2021. Accessed: Mar. 4, 2022. [Online]. Available: https://www.ise.fraunhofer.de/content/dam/ise/en/documents/publications/studies/EN2021_Fraunhofer-ISE_LCOE_Renewable_Energy_Technologies.pdf

[6] Department for Business, Energy and Industrial Strategy, “Hydrogen production costs 2021,” London, Aug. 2021. Accessed: Apr. 24, 2022. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1011506/Hydrogen_Production_Costs_2021.pdf

[7] J. Wilde, “Generalization of the Annuity Factor,” *AFR*, vol. 7, no. 2, p. 83, 2018, doi: 10.5430/afr.v7n2p83.

[8] R. S. Pindyck and D. L. Rubinfeld, *Microeconomics* (The Pearson series in economics). Harlow: Pearson, 2018.

[9] J. B. Berk and P. M. DeMarzo, *Corporate finance*. Harlow, England, London, New York, Boston, San Francisco, Toronto, Sydney, Dubai, Singapore, Hong Kong, Tokyo, Seoul, Taipei, New Delhi, Cape Town, Sao Paulo, Mexico City, Madrid,

Amsterdam, Munich, Paris, Milan: Pearson, 2020. [Online]. Available: https://ebookcentral.proquest.com/lib/munchentech/detail.action?docID=5834076

[10] W. Kenton, "Equivalent Annual Cost – EAC Definition," Jan. 2004. Accessed: Aug. 10, 2022. [Online]. Available: https://www.investopedia.com/terms/e/eac.asp

[11] Institute, Corporate Finance, "Weighted Average Cost of Capital (WACC)," Feb. 2020. Accessed: Aug. 6, 2022. [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/finance/what-is-wacc-formula/

[12] E. Picardo, "Country Risk Premium (CRP)," Dec. 2009. Accessed: Aug. 23, 2022. [Online]. Available: https://www.investopedia.com/terms/c/country-risk-premium.asp

[13] J. Chen, "Equity Risk Premium," Nov. 2003. Accessed: Aug. 6, 2022. [Online]. Available: https://www.investopedia.com/terms/e/equityriskpremium.asp

[14] A. Damodara. "Country Default Spreads and Risk Premium." https://pages.stern.nyu.edu/~adamodar/New_Home_Page/datafile/ctryprem.html (accessed Aug. 10, 2022).

[15] Aswath Damodaran. "Measuring Company Exposure to Country Risk: Theory and Practice." http://pages.stern.nyu.edu/~adamodar/New_Home_Page/valquestions/CountryRisk.htm (accessed Apr. 24, 2022).

[16] Aswath Damodaran. "Country Default Spreads and Risk Premiums: Last updated: January 5, 2022." https://pages.stern.nyu.edu/~adamodar/New_Home_Page/datafile/ctryprem.html (accessed Apr. 24, 2022).

[17] A. Damodaran, "Equity Risk Premiums (ERP): Determinants, Estimation and Implications," *SSRN Journal*, 2008, doi: 10.2139/ssrn.1274967.

[18] A. Damodaran, "Country Risk: Determinants, Measures and Implications - The 2021 Edition," *SSRN Journal*, 2021, doi: 10.2139/ssrn.3879109.

[19] K. Emery, "Rating Symbols and Definitions," Moody's Investors Service, New York, 2022. Accessed: Aug. 23, 2022. [Online]. Available: https://www.moodys.com/researchdocumentcontentpage.aspx?docid=PBC_79004

[20] Credendo. "Country risk - World | Credendo." https://credendo.com/en/country-risk/world# (accessed Aug. 18, 2022).

[21] Institute, Corporate Finance, "Credit Default Swap," Mar. 2020. Accessed: Sep. 8, 2022. [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/finance/credit-default-swap-cds/

[22] W. Kenton, "What is a Bank Rate?," Nov. 2003. Accessed: Aug. 23, 2022. [Online]. Available: https://www.investopedia.com/terms/b/bankrate.asp

[23] IRENA, “Green Hydrogen Cost Reduction: Scaling up Electrolysers to Meet the 1.5ºC Climate Goal,” International Renewable Energy Agency, Abu Dhabi, 2020. Accessed: Dec. 8, 2022. [Online]. Available: https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2020/Dec/IRENA_Green_hydrogen_cost_2020.pdf

[24] sbh4 GmbH, “Electrolysers: AEC, AEM, PEM and SOE for hydorgen (and syngas) production,” sbh4 GmbH, Gelting, 2021. Accessed: Aug. 13, 2022. [Online]. Available: http://www.sbh4.de/assets/electrolyser-aec-aem-pem-soe.pdf

[25] J. Zou *et al.,* “Electrochemical Compression Technologies for High-Pressure Hydrogen: Current Status, Challenges and Perspective,” *Electrochem. Energ. Rev.*, vol. 3, no. 4, pp. 690–729, 2020, doi: 10.1007/s41918-020-00077-0.

[26] Hydrogen and Fuel Cell Technologies Office (HFTO), “Hydrogen Storage,” Office of Energy Efficiency & Renewavle Energy, Washington DC, Aug. 2022. Accessed: Aug. 23, 2022. [Online]. Available: https://www.energy.gov/eere/fuelcells/hydrogen-storage

[27] IRENA, “Green hydrogen: A guide to policy making,” International Renewable Energy Agency, Abu Dhabi, 2020. Accessed: Aug. 23, 2022. [Online]. Available: https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2020/Nov/IRENA_Green_hydrogen_policy_2020.pdf

[28] The Royal Society, “Ammonia: zero-carbon fertiliser, fuel and energy store,” The Royal Society, London, 2020. Accessed: Aug. 23, 2022. [Online]. Available: https://royalsociety.org/-/media/policy/projects/green-ammonia/green-ammonia-policy-briefing.pdf

[29] S. C. Bhattacharyya and G. R. Timilsina, “A review of energy system models,” *International Journal of Energy Sector Management*, vol. 4, no. 4, pp. 494–518, 2010, doi: 10.1108/17506221011092742.

[30] Eniscuola. “The energy system - Eniscuola.” https://www.eniscuola.net/en/argomento/the-energy-system/the-man-and-energy/the-energy-system1/ (accessed Sep. 12, 2022).

[31] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*. Boston, MA: Robert J.Vanderbei, 2008.

[32] R. Sioshansi and A. J. Conejo, *Optimization in engineering: Models and algorithms* (Springer optimization and its applications). Cham: Springer, 2017.

[33] M. J. Panik, *Linear Programming and Resource Allocation Modeling*. Newark, UNITED STATES: John Wiley & Sons, Incorporated, 2018. [Online]. Available: http://ebookcentral.proquest.com/lib/munchentech/detail.action?docID=5558401

[34] M. Pfennig, M. von Bonin, and N. Gerhardt. “DeV-KopSys - PtX-Atlas: Weltweite Potenziale für die Erzeugung von grünem Wasserstoff und klimaneutralen synthetischen Kraft- und Bresnnstoffen.” https://devkopsys.de/ptx-atlas/ (accessed Mar. 31, 2022).

[35] G. Brändle, M. Schönfisch, and S. Schulte, “Estimating long-term global supply costs for low-carbon hydrogen,” *Applied Energy*, vol. 302, p. 117481, 2020, doi: 10.1016/j.apenergy.2021.117481.

[36] A. Rosenstiel, N. Monnerie, J. Dersch, M. Roeb, R. Pitz-Paal, and C. Sattler, “Electrochemical Hydrogen Production Powered by PV/CSP Hybrid Power Plants: A Modelling Approach for Cost Optimal System Design,” *Energies*, vol. 14, no. 12, p. 3437, 2021, doi: 10.3390/en14123437.

[37] H. Tebibel, “Methodology for multi-objective optimization of wind turbine/battery/electrolyzer system for decentralized clean hydrogen production using an adapted power management strategy for low wind speed conditions,” *Energy Conversion and Management*, vol. 238, p. 114125, 2021, doi: 10.1016/j.enconman.2021.114125.

[38] S. Timmerberg and M. Kaltschmitt, “Hydrogen from renewables: Supply from North Africa to Central Europe as blend in existing pipelines – Potentials and costs,” *Applied Energy*, pp. 795–809, 2019, doi: 10.1016/j.apenergy.2019.01.030.

[39] Agora Energiewende and AFRY Management Consulting, “No-regret hydrogen: Charting early steps for H_2 infrastructure in Europe.,” 2021. Accessed: Jan. 5, 2022. [Online]. Available: https://static.agora-energiewende.de/fileadmin/Projekte/2021/2021_02_EU_H2Grid/A-EW_203_No-regret-hydrogen_WEB.pdf

[40] AFRY, “AFRY's Global Hydrogen Trade Model: The future market for low carbon hydrogen,” 2022. Accessed: May 27, 2022. [Online]. Available: https://afry.com/sites/default/files/2022-05/afrys_global_hydrogen_trade_model_v100_0.pdf

[41] Bundesministerium für Bildung und Förschung. “HYPAT - Globaler H2-Potenzialatlas.” https://www.hypat.de/hypat/index.php (accessed Aug. 4, 2022).

[42] J. Thomann *et al.,* “Hintergrundpapier zu nachhaltigem grünen Wasserstoff und Syntheseprodukten: HYPAT Working Paper,” Fraunhofer ISI (Hrsg), Karlsruhe, 2022. Accessed: Aug. 4, 2022. [Online]. Available: https://www.hypat.de/hypat-wAssets/docs/new/publikationen/HyPAT_Working-Paper_01-2022_Hintergrundpapier_nachhaltiger_gruener_Wasserstoff_Syntheseprodukten.pdf

[43] Forschungszentrum Jülich GmbH. “H2Atlas-Africa.” https://www.h2atlas.de/en/ (accessed Jul. 4, 2022).

[44] Geoscience Australia - Australian Government. “AusH2 - Australia's Hydrogen Opportunities Tool.” https://portal.ga.gov.au/persona/hydrogen (accessed Aug. 4, 2022).

[45] National Renewable Energy Laboratory - NREL. “HyDRA: Hydrogen Demand and Resource Analysis Tool.” https://maps.nrel.gov/?da=hydra (accessed Aug. 4, 2022).

[46] IRENA, “World Energy Transitions Outlook 2022: 1.5°C Pathway,” International Renewable Energy Agency, Abu Dhabi, 2022. Accessed: Apr. 6, 2022. [Online]. Available: https://irena.org/-/media/Files/IRENA/Agency/Publication/2022/Mar/IRENA_World_Energy_Transitions_Outlook_2022.pdf

[47] M. Sponseller, “Lazard's Levelized Cost of Hydrogen Analysis - Version 2.0.pptx,” Hamilton, Bermuda. Accessed: Mar. 31, 2022. [Online]. Available: https://www.lazard.com/media/451922/lazards-levelized-cost-of-hydrogen-analysis-version-20-vf.pdf

[48] J. Yates *et al.,* “Techno-economic Analysis of Hydrogen Electrolysis from Off-Grid Stand-Alone Photovoltaics Incorporating Uncertainty Analysis,” *Cell Reports Physical Science*, vol. 1, no. 10, p. 100209, 2020, doi: 10.1016/j.xcrp.2020.100209.

[49] A. Christensen, “Assessment of Hydrogen Production Costs from Electrolysis: United States and Europe,” 2020. [Online]. Available: https://theicct.org/sites/default/files/publications/final_icct2020_assessment_of%20_hydrogen_production_costs%20v2.pdf

[50] J. Lehmann, A. Wabbes, E. Miguelañez Gonzalez, and S. Scheerlinck, “Levelized Cost of Hydrogen Calculation from Off-Grid Photovoltaic Plants Using Different Methods,” *Solar RRL*, p. 2100482, 2021, doi: 10.1002/solr.202100482.

[51] A. Khouya, “Levelized costs of energy and hydrogen of wind farms and concentrated photovoltaic thermal systems. A case study in Morocco,” *International Journal of Hydrogen Energy*, vol. 45, no. 56, pp. 31632–31650, 2020, doi: 10.1016/j.ijhydene.2020.08.240.

[52] A. Khouya, “Hydrogen production costs of a polymer electrolyte membrane electrolysis powered by a renewable hybrid system,” *International Journal of Hydrogen Energy*, vol. 46, no. 27, pp. 14005–14023, 2021, doi: 10.1016/j.ijhydene.2021.01.213.

[53] F. I. Gallardo, A. Monforti Ferrario, M. Lamagna, E. Bocci, D. Astiaso Garcia, and T. E. Baeza-Jeria, “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," *International Journal of Hydrogen Energy*, vol. 46, no. 26, pp. 13709–13728, 2021, doi: 10.1016/j.ijhydene.2020.07.050.

[54] H. Niaz, M. M. Lakouraj, and J. Liu, "Techno-economic feasibility evaluation of a standalone solar-powered alkaline water electrolyzer considering the influence of battery energy storage system: A Korean case study," *Korean J. Chem. Eng.*, vol. 38, no. 8, pp. 1617–1630, 2021, doi: 10.1007/s11814-021-0819-z.

[55] J. L. Janssen, M. Weeda, R. J. Detz, and B. van der Zwaan, "Country-specific cost projections for renewable hydrogen production through off-grid electricity systems," *Applied Energy*, vol. 309, p. 118398, 2022. doi: 10.1016/j.apenergy.2021.118398. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261921016342

[56] International Energy Agency - IEA, "The Future of Hydrogen: Seizing today's opportunities," Paris, Jun. 2019. Accessed: Aug. 4, 2022. [Online]. Available: https://iea.blob.core.windows.net/assets/9e3a3493-b9a6-4b7d-b499-7ca48e357561/The_Future_of_Hydrogen.pdf

[57] M. Ni, M. K. Leung, and D. Y. Leung, "Energy and exergy analysis of hydrogen production by a proton exchange membrane (PEM) electrolyzer plant," *Energy Conversion and Management*, vol. 49, no. 10, pp. 2748–2756, 2008, doi: 10.1016/j.enconman.2008.03.018.

[58] J. Tiktak, "Heat Management of PEM Electrolysis: A study on the potential of excess heat from medium- to large-scale PEM electrolysis and the performance analysis of a dedicated cooling system," Maritime and Materials Engineering, TU Delft Mechanical, 2019. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%3Ac046820a-72bc-4f05-b72d-e60a3ecb8c89

[59] M. Bosilovich, R. Lucchesi, and S. Max, "MERRA-2: File Specification. GMAO Office Note No. 9 (Version 1.1)," NASA - Global Modeling and Assimilation Office - Earth Sciences Division, Greenbelt, Maryland, 2016. Accessed: Aug. 15, 2022. [Online]. Available: http://gmao.gsfc.nasa.gov/pubs/office_notes

[60] S. Milanzi, C. Spiller, B. Grosse, L. Hermann, and J. Müller-Kirchenbauer, "Technischer Stand und Flexibilität des Power-to-Gas-Verfahrens," Technische Universtität Berlin, Berlin, 2018, doi: 10.5281/ZENODO.2620253.

[61] T. Brown *et al., PyPSA: Python for Power System Analysis* (2022). Accessed: May 27, 2022. [Online]. Available: https://pypsa.readthedocs.io/en/latest/index.html#

[62] J. Dorfner, *urbs: A linear optimisation model for distributed energy systems* (2022). Accessed: May 27, 2022. [Online]. Available: https://urbs.readthedocs.io/en/latest/#

[63] C. Gaete-Morales, A. Roth, M. Kittel, W.-P. Schill, and A. Zerrahn, *DIETERpy* (2022). Accessed: Aug. 15, 2022. [Online]. Available: https://pypi.org/project/dieterpy/

[64] R. D. Zimmerman, C. E. Murillo-Sánchez, and J. R. Thomas, *MATPOWER: Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education.* Accessed: Aug. 15, 2022. [Online]. Available: https://pypi.org/project/PYPOWER/

[65] M. Kaltschmitt, N. J. Themelis, and L. Y. Bronicki, *Renewable energy systems*. New York, NY: Springer, 2013.

[66] Siemens AG, "Siemens Wind Turbine SWT-3.6-120: Thoroughly tested, utterly reliable," Siemens AG Energy Sector, Erlangen, 2011. [Online]. Available: https://pdf.archiexpo.com/pdf/siemens-gamesa/swt-36-120/88089-134487.html

[67] Vestas, "Vestas V164 8.0 MW," Vestas Wind Systems, Aarhus, 2011. [Online]. Available: https://pdf.archiexpo.com/pdf/vestas/vestas-v164-80-mw/88087-134417.html

[68] Siemens AG, "Siemens Wind Turbine SWT-3.6-107," Siemens AG Energy Sector, Erlangen, 2009. [Online]. Available: https://pdf.archiexpo.de/pdf/siemens-gamesa/swt-36-107/88089-148795.html

[69] Enercon, "Enercon Wind Turbines: Product Portfolio - Technical Data Sheets (Last Updated 09/2018)," Enercon, Aurich, 2018. [Online]. Available: https://www.enercon.de/fileadmin/Redakteur/Medien-Portal/broschueren/pdf/EC_Datenblaetter_WEA_en.pdf

[70] Nordex, "Nordex N100/2500: Facts and figures," Nordex SE, Rostock, 2010. [Online]. Available: https://www.renugen.co.uk/content/large_wind_turbine_brochures/large_wind_turbine_brochures/nordex_n100.pdf

[71] Vestas, "2.0 MW Wind Turbines," Vestas Wind Systems, Aarhus, 2012. [Online]. Available: https://pdf.archiexpo.es/pdf-en/vestas/2-mw-v80-20-mw-v90-18-20-mw-v100-18-mw-20-mw/88087-134431.html

[72] The Hydrogen and Fuel Cell Information System. "Hydrogen Data." http://www.h2data.de/ (accessed Aug. 29, 2022).

[73] T. Schmid and Schmidt-Achert Tapio, "Photovoltaic cost: Assumptions for eXtremOS," FfE, Munich, 2021.

[74] Schmidt-Achert Tapio, C. Pellinger, and T. Schmid, "Wind generation cost: Cost assumptions for eXtremOS," FfE, Munich, 2021.

[75] P.-M. Heuser, “Weltweite Infrastruktur zur Wasserstoffbereitstellung auf Basis erneuerbarer Energien,” Forschungszentrum Jülich GmbH, Jülich, 2021. [Online]. Available: https://juser.fz-juelich.de/record/893099/files/Energie_Umwelt_532.pdf

[76] P. Sterchele *et al.,* “Anhang zur Studie Wege zu einem klimaneutralen Energiesystem,” Fraunhofer-Institut für Solare Energiesysteme ISE, Freiburg, 2020. [Online]. Available: https://www.ise.fraunhofer.de/content/dam/ise/de/documents/publications/studies/Anhang-Studie-Wege-zu-einem-klimaneutralen-Energiesystem.pdf

[77] Schmidt-Achert Tapio and C. Pellinger, “Future sationary battery cost: Assumptions for eXtremOS,” FfE, Munich, 2021.

[78] S. Fattler *et al.,* “Dynamis - Datenanhang: Dynamische und intersektorale Maßnahmenbewertung zur kosteneffizienten Dekarbonisierung des Energiesystems,” FfE, Munich, 2019.

[79] T. Smolinka *et al.,* “Studie IndWEDe: Industrialisierung der Wasser-elektrolyse in -Deutschland: -Chancen und -Herausforderungen für nachhaltigen Wasserstoff für Verkehr, Strom und -Wärme,” NOW, Berlin, 2018. Accessed: Aug. 26, 2022. [Online]. Available: https://www.now-gmbh.de/wp-content/uploads/2020/09/indwede-studie_v04.1.pdf

[80] J. McLaren, “Batteries 101 Series: How to Talk About Batteries and Power-To-Energy Ratios,” National Renewable Energy Laboratory - NREL, Aug. 2022. Accessed: Aug. 26, 2022. [Online]. Available: https://www.nrel.gov/state-local-tribal/blog/posts/batteries-101-series-how-to-talk-about-batteries-and-power-to-energy-ratios.html

[81] T. Schmid, *SQL Script for the Calculation of Wind Turbines Costs*. Munich, 2021.

[82] *Codes for the representation of names of countries and their subdivisions: Part 1: Country code (ISO 3166-1:2020)*, 3166-1:2020, International Organization for Standarization, Sep. 2020.

[83] E. Hau, *Windkraftanlagen: Grundlagen Technik Einsatz Wirtschaftlichkeit,* 6th ed. Berlin: Springer Vieweg, 2016.

[84] *Wind energy generation systems: Part 1: Design requirements*, 61400-1:2019, International Electrotechnical Commission, Dec. 2019.

[85] Enercon, “Enercon Wind Turbines: Product Portfolio - Technical Data Sheets (Last Updated 09/2021),” Enercon, Aurich, 2021. [Online]. Available: https://www.enercon.de/fileadmin/Redakteur/Medien-Portal/broschueren/EC_WEA_EN.pdf

[86] B. A. Harper, J. D. Kepert, and J. D. Ginger, “Guidelines for converting between various wind averaging periods in tropical cyclone conditions,” World Meteorological Organization (WMO), Geneva, Aug. 2010. [Online]. Available: https://library.wmo.int/doc_num.php?explnum_id=290

[87] Python Software Foundation. “Multiprocessing Python Package: Process-based parallelism - Python 3.10.7 documentation.” https://docs.python.org/3/library/multiprocessing.html (accessed Sep. 19, 2022).

[88] N. Husomanovic, “Greenland wants to sell hydrogen to the rest of the world,” *Green Hydrogen News*, 11 May., 2022. https://energynews.biz/greenland-wants-to-sell-hydrogen-to-the-rest-of-the-world/ (accessed: Sep. 7, 2022).

[89] IRENA, “Global hydrogen trade to meet the 1.5°C climate goal: Part III – Green hydrogen cost and potential,” International Renewable Energy Agency, Abu Dhabi, 2022. Accessed: Sep. 1, 2022. [Online]. Available: https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2022/May/IRENA_Global_Hydrogen_Trade_Costs_2022.pdf

[90] Deutsche Energie-Agentur (Publisher) and dena, “Hy3 – Large-Scale Hydrogen Production from Offshore Wind to Decarbonise the Dutch and German Industry,” 2022. Accessed: Sep. 2, 2022. [Online]. Available: https://www.dena.de/fileadmin/dena/Publikationen/PDFs/2022/Hy3_Large-scale_Hydrogen_Production_from_Offshore_Wind_to_Decarbonise_the_Dutch_and_German_Industry.pdf

8 Appendix

This chapter contains the different appendix supporting the academic work.

8.1 Appendix I: Country Risk Premiums (January 5, 2022)

The following table shows the country risk premiums calculated by the Stern School of Business in the New York University. Updated in the January 5 from 2022 [14].

Country	Moody's rating	Adj. Default Spread	Country Risk Premium	Equity Risk Premium
Abu Dhabi	Aa2	0.42%	0.49%	4.73%
Albania	B1	3.83%	4.45%	8.69%
Algeria	NR	5.53%	6.43%	10.67%
Andorra (Principality of)	Baa2	1.62%	1.88%	6.12%
Angola	B3	5.53%	6.43%	10.67%
Anguilla	NR	5.88%	6.83%	11.07%
Antigua & Barbuda	NR	5.88%	6.83%	11.07%
Argentina	Ca	10.21%	11.87%	16.11%
Armenia	Ba3	3.06%	3.56%	7.80%
Aruba	Baa2	1.62%	1.88%	6.12%
Australia	Aaa	0.00%	0.00%	4.24%
Austria	Aa1	0.34%	0.39%	4.63%
Azerbaijan	Ba2	2.56%	2.97%	7.21%
Bahamas	Ba3	3.06%	3.56%	7.80%
Bahrain	B2	4.68%	5.44%	9.68%
Bangladesh	Ba3	3.06%	3.56%	7.80%
Barbados	Caa1	6.38%	7.41%	11.65%
Belarus	B3	5.53%	6.43%	10.67%
Belgium	Aa3	0.51%	0.60%	4.84%
Belize	Caa3	8.51%	9.89%	14.13%
Benin	B1	3.83%	4.45%	8.69%
Bermuda	A2	0.72%	0.84%	5.08%
Bolivia	B2	4.68%	5.44%	9.68%
Bosnia and Herzegovina	B3	5.53%	6.43%	10.67%
Botswana	A3	1.02%	1.19%	5.43%
Brazil	Ba2	2.56%	2.97%	7.21%
British Virgin Islands	NR	5.88%	6.83%	11.07%
Brunei	NR	0.72%	0.84%	5.08%
Bulgaria	Baa1	1.36%	1.58%	5.82%
Burkina Faso	B2	4.68%	5.44%	9.68%
Cambodia	B2	4.68%	5.44%	9.68%
Cameroon	B2	4.68%	5.44%	9.68%
Canada	Aaa	0.00%	0.00%	4.24%
Cape Verde	B3	5.53%	6.43%	10.67%
Cayman Islands	Aa3	0.51%	0.60%	4.84%
Channel Islands	NR	0.72%	0.83%	5.07%
Chile	A1	0.60%	0.70%	4.94%
China	A1	0.60%	0.70%	4.94%
Colombia	Baa2	1.62%	1.88%	6.12%
Congo (Democratic Republic	Caa1	6.38%	7.41%	11.65%
Congo (Republic of)	Caa2	7.66%	8.90%	13.14%
Cook Islands	B1	3.83%	4.45%	8.69%
Costa Rica	B2	4.68%	5.44%	9.68%
Croatia	Ba1	2.13%	2.47%	6.71%

Country	Moody's rating	Adj. Default Spread	Country Risk Premium	Equity Risk Premium
Cuba	Ca	10.21%	11.87%	16.11%
Curacao	Baa2	1.62%	1.88%	6.12%
Cyprus	Ba1	2.13%	2.47%	6.71%
Czech Republic	Aa3	0.51%	0.60%	4.84%
Denmark	Aaa	0.00%	0.00%	4.24%
Dominican Republic	Ba3	3.06%	3.56%	7.80%
Ecuador	Caa3	8.51%	9.89%	14.13%
Egypt	B2	4.68%	5.44%	9.68%
El Salvador	Caa1	6.38%	7.41%	11.65%
Estonia	A1	0.60%	0.70%	4.94%
Ethiopia	Caa2	7.66%	8.90%	13.14%
Falkland Islands	NR	5.88%	6.83%	11.07%
Fiji	B1	3.83%	4.45%	8.69%
Finland	Aa1	0.34%	0.39%	4.63%
France	Aa2	0.42%	0.49%	4.73%
French Guiana	NR	3.26%	3.79%	8.03%
Gabon	Caa1	6.38%	7.41%	11.65%
Gambia	NR	4.68%	5.44%	9.68%
Georgia	Ba2	2.56%	2.97%	7.21%
Germany	Aaa	0.00%	0.00%	4.24%
Ghana	B3	5.53%	6.43%	10.67%
Gibraltar	NR	0.72%	0.83%	5.07%
Greece	Ba3	3.06%	3.56%	7.80%
Greenland	NR	0.72%	0.83%	5.07%
Guatemala	Ba1	2.13%	2.47%	6.71%
Guernsey	Aa3	0.51%	0.60%	4.84%
Guinea	NR	7.66%	8.90%	13.14%
Guinea-Bissau	NR	5.53%	6.43%	10.67%
Guyana	NR	3.83%	4.45%	8.69%
Haiti	NR	8.51%	9.89%	14.13%
Honduras	B1	3.83%	4.45%	8.69%
Hong Kong	Aa3	0.51%	0.60%	4.84%
Hungary	Baa2	1.62%	1.88%	6.12%
Iceland	A2	0.72%	0.84%	5.08%
India	Baa3	1.87%	2.18%	6.42%
Indonesia	Baa2	1.62%	1.88%	6.12%
Iran	NR	5.53%	6.43%	10.67%
Iraq	Caa1	6.38%	7.41%	11.65%
Ireland	A2	0.72%	0.84%	5.08%
Isle of Man	Aa3	0.51%	0.60%	4.84%
Israel	A1	0.60%	0.70%	4.94%
Italy	Baa3	1.87%	2.18%	6.42%
Ivory Coast	Ba3	3.06%	3.56%	7.80%
Jamaica	B2	4.68%	5.44%	9.68%
Japan	A1	0.60%	0.70%	4.94%
Jersey	Aaa	0.00%	0.00%	4.24%
Jordan	B1	3.83%	4.45%	8.69%
Kazakhstan	Baa2	1.62%	1.88%	6.12%
Kenya	B2	4.68%	5.44%	9.68%
Korea, D.P.R.	NR	10.21%	11.87%	16.11%
Kuwait	A1	0.60%	0.70%	4.94%
Kyrgyzstan	B2	4.68%	5.44%	9.68%
Laos	Caa2	7.66%	8.90%	13.14%
Latvia	A3	1.02%	1.19%	5.43%
Lebanon	C	17.50%	20.34%	24.58%
Liberia	NR	7.66%	8.90%	13.14%
Libya	NR	3.83%	4.45%	8.69%
Liechtenstein	Aaa	0.00%	0.00%	4.24%

Country	Moody's rating	Adj. Default Spread	Country Risk Premium	Equity Risk Premium
Lithuania	A2	0.72%	0.84%	5.08%
Luxembourg	Aaa	0.00%	0.00%	4.24%
Macao	Aa3	0.51%	0.60%	4.84%
Macedonia	Ba3	3.06%	3.56%	7.80%
Madagascar	NR	5.53%	6.43%	10.67%
Malawi	NR	7.66%	8.90%	13.14%
Malaysia	A3	1.02%	1.19%	5.43%
Maldives	Caa1	6.38%	7.41%	11.65%
Mali	Caa1	6.38%	7.41%	11.65%
Malta	A2	0.72%	0.84%	5.08%
Martinique	NR	3.26%	3.79%	8.03%
Mauritius	Baa2	1.62%	1.88%	6.12%
Mexico	Baa1	1.36%	1.58%	5.82%
Moldova	B3	5.53%	6.43%	10.67%
Mongolia	B3	5.53%	6.43%	10.67%
Montenegro	B1	3.83%	4.45%	8.69%
Montserrat	Baa3	1.87%	2.18%	6.42%
Morocco	Ba1	2.13%	2.47%	6.71%
Mozambique	Caa2	7.66%	8.90%	13.14%
Myanmar	NR	10.21%	11.87%	16.11%
Namibia	Ba3	3.06%	3.56%	7.80%
Netherlands	Aaa	0.00%	0.00%	4.24%
Netherlands Antilles	NR	5.88%	6.83%	11.07%
New Zealand	Aaa	0.00%	0.00%	4.24%
Nicaragua	B3	5.53%	6.43%	10.67%
Niger	B3	5.53%	6.43%	10.67%
Nigeria	B2	4.68%	5.44%	9.68%
Norway	Aaa	0.00%	0.00%	4.24%
Oman	Ba3	3.06%	3.56%	7.80%
Pakistan	B3	5.53%	6.43%	10.67%
Palestinian Authority	NR	1.38%	1.60%	5.84%
Panama	Baa2	1.62%	1.88%	6.12%
Papua New Guinea	B2	4.68%	5.44%	9.68%
Paraguay	Ba1	2.13%	2.47%	6.71%
Peru	Baa1	1.36%	1.58%	5.82%
Philippines	Baa2	1.62%	1.88%	6.12%
Poland	A2	0.72%	0.84%	5.08%
Portugal	Baa2	1.62%	1.88%	6.12%
Qatar	Aa3	0.51%	0.60%	4.84%
Ras Al Khaimah (Emirate of)	A3	1.02%	1.19%	5.43%
Reunion	NR	4.51%	5.25%	9.49%
Romania	Baa3	1.87%	2.18%	6.42%
Russia	Baa3	1.87%	2.18%	6.42%
Rwanda	B2	4.68%	5.44%	9.68%
Saint Lucia	NR	5.88%	6.83%	11.07%
Saudi Arabia	A1	0.60%	0.70%	4.94%
Senegal	Ba3	3.06%	3.56%	7.80%
Serbia	Ba2	2.56%	2.97%	7.21%
Sharjah	Baa3	1.87%	2.18%	6.42%
Sierra Leone	NR	8.51%	9.89%	14.13%
Singapore	Aaa	0.00%	0.00%	4.24%
Slovakia	A2	0.72%	0.84%	5.08%
Slovenia	A3	1.02%	1.19%	5.43%
Solomon Islands	Caa1	6.38%	7.41%	11.65%
Somalia	NR	10.21%	11.87%	16.11%
South Africa	Ba2	2.56%	2.97%	7.21%
South Korea	Aa2	0.42%	0.49%	4.73%

Country	Moody's rating	Adj. Default Spread	Country Risk Premium	Equity Risk Premium
Spain	Baa1	1.36%	1.58%	5.82%
Sri Lanka	Caa2	7.66%	8.90%	13.14%
St. Maarten	Ba2	2.56%	2.97%	7.21%
St. Vincent & the Grenadines	B3	5.53%	6.43%	10.67%
Sudan	NR	17.50%	20.34%	24.58%
Suriname	Caa3	8.51%	9.89%	14.13%
Swaziland	B3	5.53%	6.43%	10.67%
Sweden	Aaa	0.00%	0.00%	4.24%
Switzerland	Aaa	0.00%	0.00%	4.24%
Syria	NR	17.50%	20.34%	24.58%
Taiwan	Aa3	0.51%	0.60%	4.84%
Tajikistan	B3	5.53%	6.43%	10.67%
Tanzania	B2	4.68%	5.44%	9.68%
Thailand	Baa1	1.36%	1.58%	5.82%
Togo	B3	5.53%	6.43%	10.67%
Trinidad and Tobago	Ba2	2.56%	2.97%	7.21%
Tunisia	Caa1	6.38%	7.41%	11.65%
Turkey	B2	4.68%	5.44%	9.68%
Turks and Caicos Islands	Baa1	1.36%	1.58%	5.82%
Uganda	B2	4.68%	5.44%	9.68%
Ukraine	B3	5.53%	6.43%	10.67%
United Arab Emirates	Aa2	0.42%	0.49%	4.73%
United Kingdom	Aa3	0.51%	0.60%	4.84%
United States	Aaa	0.00%	0.00%	4.24%
Uruguay	Baa2	1.62%	1.88%	6.12%
Uzbekistan	B1	3.83%	4.45%	8.69%
Venezuela	C	17.50%	20.34%	24.58%
Vietnam	Ba3	3.06%	3.56%	7.80%
Yemen	NR	10.21%	11.87%	16.11%
Zambia	Ca	10.21%	11.87%	16.11%
Zimbabwe	NR	6.38%	7.41%	11.65%

8.2 Appendix II: IEC 61400-1:2019 Wind classes criteria

The following table from the IEC 61400-1:2019 shows the criteria for the classification of the different wind classes.

WEA-Klasse		I	II	III	S
V_{ave}	(m/s)	10	8,5	7,5	Vom Konstrukteur festzulegende Werte
V_{ref}	(m/s)	50	42,5	37,5	
	Tropisch (m/s) $V_{ref,T}$	57	57	57	
A+	I_{ref} (−)	0,18			
A	I_{ref} (−)	0,16			
B	I_{ref} (−)	0,14			
C	I_{ref} (−)	0,12			

Die Parameterwerte gelten in Nabenhöhe. Dabei ist

V_{ave} das Jahresmittel der Windgeschwindigkeit;

V_{ref} der 10-min-Mittelwert der Referenzwindgeschwindigkeit;

$V_{ref,T}$ der für tropische Wirbelstürme geltende 10-min-Mittelwert der Referenzwindgeschwindigkeit;

A+ bezeichnet die Kategorie für sehr hohe Turbulenzkennwerte;

A bezeichnet die Kategorie für höhere Turbulenzkennwerte;

B bezeichnet die Kategorie für mittlere Turbulenzkennwerte;

C bezeichnet die Kategorie für niedrigere Turbulenzkennwerte; und

I_{ref} ein Bezugswert der Turbulenzintensität (siehe 6.3.2.3 [+]).

8.3 Appendix III: Output data stored in database

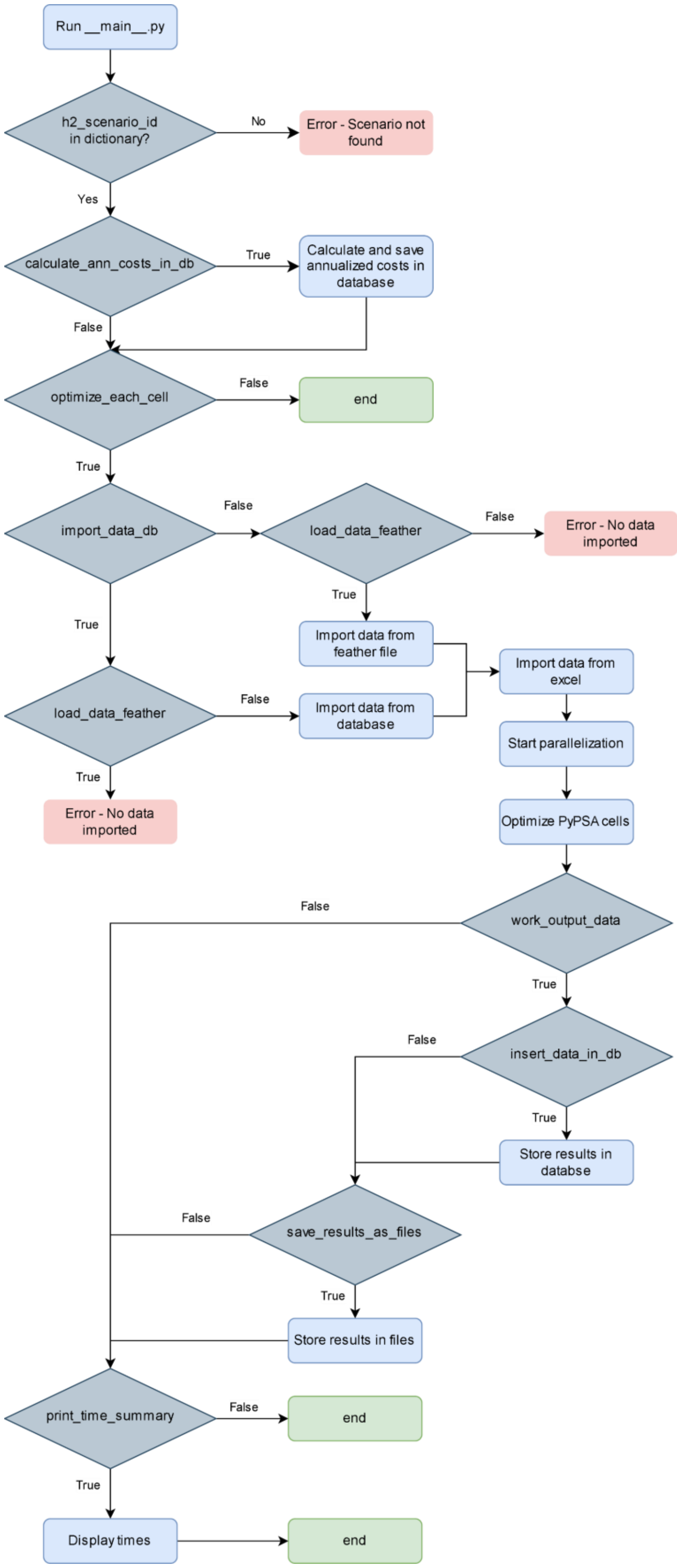
The following table shows the data output extracted from the optimization results and stored in the database.

Group	Id	Parameter	Unit
LCOH	1	Levelized Cost of Hydrogen	€/kg
Size	2	PV - Nominal power	MW
	3	Wind very weak - Nominal power	MW
	4	Wind very strong - Nominal power	MW
	5	Electrolyzer - Nominal power	MW
	6	Tank power dispatch of hydrogen - Nominal power	MW
	7	Tank power store of hydrogen - Nominal power	MW
	8	Tank capacity store unit hydrogen - Nominal capacity	MWh
	9	Battery - Nominal power	MW
Timeseries (Hourly array)	10	Generator PV - Produced power timeseries	MW
	11	Generator Wind very weak - Produced power	MW
	12	Generator Wind very strong - Produced power	MW
	13	Link Electrolyzer - Lower shadow price	€/MW
	14	Link Tank power dispatch of hydrogen - Lower shadow price	€/MW
	15	Link Tank power store of hydrogen - Lower shadow price	€/MW
	16	Link Electrolyzer - Upper shadow price	€/MW
	17	Link Tank power dispatch of hydrogen - Upper shadow price	€/MW
	18	Link Tank power store of hydrogen - Upper shadow price	€/MW
	19	Link Electrolyzer - Input power	MW
	20	Link Tank power dispatch of hydrogen - Input power	MW
	21	Link Tank power store of hydrogen - Input power	MW
	22	Link Electrolyzer - Output power	MW
	23	Link Tank power dispatch of hydrogen - Output power	MW
	24	Link Tank power store of hydrogen - Output power	MW
	25	Storage Unit soc battery - State of charge	MWh
	26	Store Tank capacity store hydrogen - State of charge	MWh
	27	Store Tank capacity store hydrogen - Dispatch or charge power	MW
	28	Storage Unit dispatch Battery power	MW
	29	Storage Unit store Battery power	MW
	30	Generator power curtailed PV	MW
	31	Generator power curtailed Wind very weak	MW
	32	Generator power curtailed Wind very strong	MW

8.4 Appendix IV: Pre-analysis results – 100 random cells

id_cell	id_country	lcoh	pv	wind_very_weak	wind_weak	wind_middle	wind_strong	wind_very_strong	electrolyzer	tank_pow_dispatch_of_hydrogen	tank_pow_store_of_hydrogen	tank_cap_store_unit_hydrogen	battery	tank e/p ratio
185316	36	3,78	0,097	0,091	0	0	0	0	0,084	0,0342	0,0153	2,857	0	187,1
103547	826	4,13	0,047	0,093	0	0	0	0	0,092	0,0342	0,0198	6,307	0	318,4
173039	36	4,16	0,090	0,104	0	0	0	0	0,090	0,0342	0,0187	4,109	0	219,8
45382	840	4,17	0,137	0,097	0	0	0	0	0,082	0,0342	0,0144	3,099	0	215,9
46463	840	4,18	0,143	0,095	0	0	0	0	0,084	0,0342	0,0154	2,875	0	186,4
177006	36	4,18	0,021	0,117	0	0	0	0	0,098	0,0342	0,0233	5,480	0	235,1
183506	36	4,52	0,141	0,102	0	0	0	0	0,092	0,0342	0,0201	4,070	0	202,8
41068	840	4,63	0,159	0,107	0	0	0	0	0,087	0,0342	0,0170	3,789	0	222,4
187101	36	4,71	0,129	0,100	0	0	0	0	0,098	0,0342	0,0237	6,001	0	252,9
96855	654	4,84	0,016	0,000	0	0	0	0,186	0,097	0,0342	0,0230	6,477	0	281,6
42163	124	4,88	0,108	0,117	0	0	0	0	0,093	0,0342	0,0207	7,016	0	338,9
64596	304	4,90	0,141	0,098	0	0	0	0	0,093	0,0342	0,0204	5,264	0	257,9
48686	124	4,91	0,111	0,120	0	0	0	0	0,101	0,0342	0,0254	5,373	0	211,8
127660	682	4,98	0,127	0,117	0	0	0	0	0,098	0,0342	0,0233	3,547	0	152,0
46164	124	5,09	0,125	0,123	0	0	0	0	0,110	0,0342	0,0303	4,654	0	153,7
40387	124	5,09	0,149	0,116	0	0	0	0	0,099	0,0342	0,0244	5,592	0	229,5
16542	840	5,16	0,201	0,104	0	0	0	0	0,104	0,0342	0,0268	4,259	0	159,1
56580	124	5,24	0,204	0,098	0	0	0	0	0,100	0,0342	0,0249	6,254	0	251,0
131321	398	5,38	0,127	0,107	0	0	0	0	0,092	0,0342	0,0199	5,349	0	268,1
47254	124	5,53	0,091	0,131	0	0	0	0	0,104	0,0342	0,0272	10,402	0	382,5
176079	156	5,55	0,135	0,120	0	0	0	0	0,099	0,0342	0,0240	7,299	0	304,1
47615	124	5,61	0,084	0,135	0	0	0	0	0,106	0,0342	0,0284	10,688	0	376,1
36064	124	5,68	0,039	0,161	0	0	0	0	0,115	0,0342	0,0334	8,916	0	267,2
116734	72	5,73	0,219	0,089	0	0	0	0	0,108	0,0342	0,0292	5,417	0	185,8
16907	840	5,79	0,263	0,101	0	0	0	0	0,120	0,0342	0,0360	4,373	0	121,4
114910	710	5,79	0,076	0,000	0	0	0	0,196	0,104	0,0342	0,0271	9,360	0	345,4
119263	72	5,82	0,317	0,044	0	0	0	0	0,152	0,0342	0,0551	1,337	0	24,3
152233	156	5,86	0,299	0,000	0	0	0	0,067	0,168	0,0342	0,0642	4,386	0	68,3
35685	124	6,11	0,086	0,170	0	0	0	0	0,117	0,0342	0,0347	8,088	0	233,2
165957	156	6,26	0,216	0,095	0	0	0	0	0,120	0,0342	0,0361	9,605	0	266,0
50152	124	6,26	0,180	0,146	0	0	0	0	0,118	0,0342	0,0351	7,183	0	204,4
60798	604	6,31	0,375	0,015	0	0	0	0	0,174	0,0342	0,0677	1,985	0	29,3
39984	840	6,34	0,274	0,115	0	0	0	0	0,139	0,0342	0,0476	3,716	0	78,1
48710	124	6,38	0,153	0,160	0	0	0	0	0,118	0,0342	0,0354	7,735	0	218,7
154103	643	6,47	0,116	0,130	0	0	0	0	0,106	0,0342	0,0279	8,028	0	287,6
58008	840	6,56	0,394	0,064	0	0	0	0	0,152	0,0342	0,0551	4,607	0	83,6
162700	156	6,57	0,366	0,000	0	0	0,047	0	0,193	0,0342	0,0784	3,993	0	50,9
20368	250	6,60	0,240	0,144	0	0	0	0	0,116	0,0342	0,0342	4,609	0	134,9
104564	12	6,64	0,138	0,094	0	0	0	0	0,086	0,0342	0,0166	3,163	0	190,7
160891	156	6,72	0,399	0,008	0	0	0	0	0,210	0,0342	0,0884	3,793	0	42,9
120340	710	6,78	0,308	0,054	0	0	0	0	0,149	0,0342	0,0533	1,976	0	37,1
61466	152	6,78	0,191	0,168	0	0	0	0	0,116	0,0342	0,0338	4,532	0	133,9
35647	840	6,94	0,240	0,158	0	0	0	0	0,131	0,0342	0,0425	5,896	0	138,6
47511	484	6,96	0,457	0,000	0	0	0	0	0,184	0,0342	0,0736	3,176	0	43,1
155548	643	7,03	0,108	0,152	0	0	0	0	0,120	0,0342	0,0361	7,423	0	205,4
123334	818	7,23	0,187	0,110	0	0	0	0	0,097	0,0342	0,0231	2,268	0	98,1
179332	156	7,33	0,325	0,120	0	0	0	0	0,144	0,0342	0,0503	4,753	0	94,5
141465	643	7,39	0,000	0,173	0	0	0	0	0,120	0,0342	0,0361	13,636	0	377,6
140363	643	7,48	0,112	0,188	0	0	0	0	0,111	0,0342	0,0309	6,180	0	200,0
108216	250	7,59	0,238	0,164	0	0	0	0	0,131	0,0342	0,0428	8,242	0	192,6
194146	643	7,59	0,285	0,052	0	0	0	0,066	0,119	0,0342	0,0355	11,868	0	334,1
148258	356	7,69	0,416	0,000	0	0	0	0	0,212	0,0342	0,0899	5,484	0	61,0
202614	554	7,72	0,256	0,139	0	0	0	0	0,129	0,0342	0,0416	15,492	0	372,3
202481	643	7,77	0,124	0,125	0	0	0	0	0,104	0,0342	0,0272	18,669	0	685,6
200304	643	7,85	0,159	0,172	0	0	0	0	0,115	0,0342	0,0331	8,088	0	244,4
137833	643	7,87	0,144	0,173	0	0	0	0	0,105	0,0342	0,0278	10,257	0	368,8
146874	643	8,23	0,136	0,211	0	0	0	0	0,118	0,0342	0,0349	5,723	0	163,9
173122	608	8,39	0,343	0,103	0	0	0	0	0,122	0,0342	0,0374	12,307	0	329,3
115476	752	8,54	0,239	0,217	0	0	0	0	0,144	0,0342	0,0505	9,511	0	188,5
77414	76	8,56	0,550	0,000	0	0	0	0	0,190	0,0342	0,0767	3,249	0	42,3
166228	360	8,61	0,622	0,000	0	0	0	0	0,193	0,0342	0,0786	3,466	0	44,1
161583	764	8,77	0,646	0,000	0	0	0	0	0,195	0,0342	0,0799	4,458	0	55,8
96601	478	8,78	0,251	0,128	0	0	0	0	0,113	0,0342	0,0320	2,859	0	89,4
33118	840	8,86	0,297	0,222	0	0	0	0	0,150	0,0342	0,0536	7,117	0	132,7
172068	156	8,99	0,616	0,000	0	0	0	0	0,239	0,0342	0,1054	7,069	0	67,1
58095	124	8,99	0,115	0,239	0	0	0	0	0,154	0,0342	0,0563	16,981	0	301,8
101999	384	9,12	0,534	0,029	0	0	0	0	0,178	0,0342	0,0699	2,468	0	35,3
160833	360	9,22	0,671	0,000	0	0	0	0	0,210	0,0342	0,0882	3,105	0	35,2
77423	76	9,25	0,615	0,000	0	0	0	0	0,199	0,0342	0,0820	2,934	0	35,8
77787	76	9,27	0,589	0,000	0	0	0	0	0,202	0,0342	0,0840	4,371	0	52,1
77782	76	9,28	0,609	0,000	0	0	0	0	0,197	0,0342	0,0812	3,788	0	46,7
162277	458	9,29	0,725	0,000	0	0	0	0	0,203	0,0342	0,0844	4,812	0	57,0
70918	76	9,34	0,612	0,000	0	0	0	0	0,200	0,0342	0,0827	3,725	0	45,0
165942	156	9,47	0,437	0,167	0	0	0	0	0,161	0,0342	0,0602	6,428	0	106,8
161183	360	9,71	0,700	0,000	0	0	0	0	0,200	0,0342	0,0825	6,465	0	78,3
182234	643	9,90	0,335	0,190	0	0	0	0	0,149	0,0342	0,0530	5,364	0	101,2
144638	586	9,93	0,446	0,009	0	0	0	0	0,198	0,0342	0,0817	2,855	0	35,0
74855	76	9,94	0,536	0,000	0	0	0	0	0,221	0,0342	0,0949	10,136	0	106,8
111034	562	9,99	0,385	0,066	0	0	0	0	0,146	0,0342	0,0514	4,122	0	80,2
180067	643	10,04	0,295	0,201	0	0	0	0	0,149	0,0342	0,0528	7,177	0	135,9
150118	643	10,09	0,081	0,307	0	0	0	0	0,127	0,0342	0,0403	6,874	0	170,4
19091	840	10,10	0,260	0,228	0	0	0	0	0,189	0,0342	0,0763	14,866	0	194,8
60487	170	10,14	0,773	0,000	0	0	0	0	0,202	0,0342	0,0836	4,988	0	59,7
131658	364	10,49	0,314	0,065	0	0	0	0	0,162	0,0342	0,0606	9,322	0	153,8
54353	188	10,68	0,381	0,000	0	0	0	0,191	0,131	0,0342	0,0427	11,076	0	259,4
184029	643	10,79	0,230	0,236	0	0	0	0	0,121	0,0342	0,0370	15,132	0	408,4
145736	586	11,24	0,448	0,000	0	0	0	0	0,208	0,0342	0,0872	10,255	0	117,6
20895	840	11,24	0,299	0,291	0	0	0	0	0,169	0,0342	0,0644	15,664	0	243,2
174677	643	11,32	0,286	0,227	0	0	0	0	0,140	0,0342	0,0476	14,810	0	310,9
55436	188	11,59	0,702	0,000	0	0	0	0	0,198	0,0342	0,0813	2,249	0	27,7
168528	643	11,61	0,145	0,297	0	0	0	0	0,125	0,0342	0,0389	16,652	0	427,8
173587	643	11,94	0,245	0,256	0	0	0	0	0,134	0,0342	0,0444	18,345	0	413,6
66182	32	12,25	0,206	0,135	0	0	0	0	0,109	0,0342	0,0300	2,660	0	88,5
95509	624	12,35	0,561	0,043	0	0	0	0	0,181	0,0342	0,0716	6,104	0	85,3
191862	598	12,47	0,715	0,000	0	0	0	0	0,213	0,0342	0,0902	4,882	0	54,1
117861	180	12,80	0,640	0,000	0	0	0	0	0,204	0,0342	0,0848	3,924	0	46,3
97674	324	12,81	0,581	0,000	0	0	0	0	0,194	0,0342	0,0791	3,238	0	40,9
60149	192	13,03	0,203	0,118	0	0	0	0	0,090	0,0342	0,0186	10,931	0	587,8
65534	862	24,38	0,679	0,000	0	0	0	0	0,209	0,0342	0,0878	3,984	0	45,4
62652	862	24,62	0,656	0,000	0	0	0	0	0,196	0,0342	0,0803	7,411	0	92,3

8.5 Appendix V: __main__.py module decision tree



8.6 Appendix VI: Electrolysis-based hydrogen production python model code

The python code for the different modules is presented below.

__main__.py module

```python
# h2_model __main__.py module

##########################################################################################
################################

from functools import partial
import logging
import os
import pandas as pd
import sys
import time

import input
import opt
import output
import precalc

logger = logging.getLogger(__name__)
##########################################################################################
################################

if __name__ == '__main__':
    calculate_ann_costs_in_db = False  # only if major changes in costs
    optimize_each_cell = True
    import_data_db = True
    save_imported_data_feather = False
    load_data_feather = False
    store_cells_as_files = False
    work_output_data = True
    save_results_as_files = False
    insert_data_in_db = True
    overwrite_scenario_results_in_db = True
    print_time_summary = True

    # ==== CHOSE NUMBER OF PARALLEL OPTIMIZATION PROCESSES (maximum
num_parallel_processes = mp.cpu_count()/2)
    num_parallel_processes = 8

    # ==== CHOSE CALCULATION SCENARIO (to import the defined base components of PyPSA
from the corresponding Excel)
    h2_scenario_id = 1
    scenario_year = 2020

    h2_scenarios = {
        1: "_base_complete",
        2: "_constant_wacc",
        3: "_only_pv",
```

```
        4: "_only_wind",
        5: "_affordable_battery",
        6: "_free_h2_storage",
        7: "_min_only_pv_only_wind",
        100: "_preanalysis_complete",
        101: "_preanalysis_complete_only_very_weak",
        102: "_preanalysis_complete_only_very_strong",
        103: "_preanalysis_complete_very_weak_and_very_strong",
        104: "_affordable_battery"
    }

    if h2_scenario_id in h2_scenarios.keys():
        scenario_name = h2_scenarios[h2_scenario_id]
        print("The calcualtion scenario is " + scenario_name + " for the year " +
str(scenario_year))
    else:
        sys.exit("The chosen h2_scenario_id doesn't exist")

    # ==== CALCULATE AND SAVE ANNUALIZED COSTS PER COUNTRY IN DB ===
    if calculate_ann_costs_in_db:
        (cur, conn) = input.establish_connection_with_db()
        tech_costs_data_df = input.import_tech_costs_data_from_db(cur)
        crp_data_df = input.import_crp_data_from_db(cur)
        cur.close()
        ann_costs_per_country =
precalc.calculate_tech_costs_per_country(tech_costs_data_df, crp_data_df)
        input.save_import_as_feather_file(ann_costs_per_country,
"ann_costs_per_country")
        output.insert_ann_costs_in_db(ann_costs_per_country, scenario_year)

    # ==== COMPLETE OPTIMIZATION OF CELLS WITH PARALLELIZATION
    if optimize_each_cell:
        time_10 = time.time()
        # ==== IMPORT TECHNO-ECONOMICAL DATA FROM DATABANK AND SAVE IT IN FEATHER FORMAT
FILE
        if import_data_db is True and load_data_feather is False:
            (cur, conn) = input.establish_connection_with_db()
            tech_costs_data_df = input.import_tech_costs_data_from_db(cur)
            crp_data_df = input.import_crp_data_from_db(cur)
            id_cells_data_df = input.import_id_cells_data_from_db(cur)
            cur.close()
            if save_imported_data_feather:
                input.save_import_as_feather_file(tech_costs_data_df,
"tech_costs_data_df")
                input.save_import_as_feather_file(crp_data_df, "crp_data_df")
                input.save_import_as_feather_file(id_cells_data_df, "id_cells_data_df")
        # ==== LOAD DATA FROM FEATHER FORMAT FILES ===
        elif import_data_db is False and load_data_feather is True:
            tech_costs_data_df = input.load_input_feather_file("tech_costs_data_df")
            crp_data_df = input.load_input_feather_file("crp_data_df")
            id_cells_data_df = input.load_input_feather_file("id_cells_data_df")
        else:
            sys.exit("Impossible to optimize because you are not importing or loading
any data. "
                     "\nSet to True ONLY one of those parameters at the __main__
module.")
        time_11 = time.time()
```

```
        output.truncate_preresults_timeseries_in_db()
        if store_cells_as_files or save_results_as_files:
            input.create_cells_results_structure(store_cells_as_files, id_cells_data_df,
h2_scenario_id)
        dict_scen_components =
input.transform_excel_scenario_sheets_to_dfs_in_dict(scenario_name)
        constant_wacc = True if scenario_name == "_constant_wacc" else False
        affordable_battery = True if scenario_name == "_affordable_battery" else False
        free_h2_storage = True if scenario_name == "_free_h2_storage" else False
        time_12 = time.time()

        # ==== OPTIMIZATION OF CELLS WITH PARALLELIZATION
        (results) = opt.parallelize_df(num_parallel_processes, id_cells_data_df,
                               partial(opt.optimize_chunk_of_cells,

dict_scen_components=dict_scen_components,
                                       h2_scenario_id=h2_scenario_id,
                                       constant_wacc=constant_wacc,
                                       affordable_battery=affordable_battery,
                                       free_h2_storage=free_h2_storage,
                                       store_cells_as_files=store_cells_as_files
                                       ))

        time_13 = time.time()

        # ==== WORK WITH OUTPUT -- SAVE DATA IN DB OR FILES
        if work_output_data:
            results_df = pd.DataFrame(results) if isinstance(results["id_region"], list)
else pd.DataFrame([results])
            results_df = output.insert_id_scenario_time(results_df, h2_scenario_id)
            if save_results_as_files:
                results_df.to_csv(os.getcwd() + "\\results_id_scenario_" +
str(h2_scenario_id) + "\\summary.csv",
                                  index=False)
                output.save_results_as_feather(results_df, h2_scenario_id)
                output.save_results_as_parquet(results_df, h2_scenario_id)

            if insert_data_in_db:
                output.insert_scenario_in_db(h2_scenario_id, scenario_name,
scenario_year)
                output.insert_results_in_db(overwrite_scenario_results_in_db,
results_df, h2_scenario_id, scenario_year)
                output.insert_results_timeseries_in_db(overwrite_scenario_results_in_db,
h2_scenario_id, scenario_year)

        time_14 = time.time()

        if print_time_summary:
            print("\nImport or load data time: ", time_11 - time_10)
            print("Preparations time: ", time_12 - time_11)
            print("Optimization time: ", time_13 - time_12)
            print("Work with output time: ", time_14 - time_13)
            print("Total time: ", time_14 - time_10)
```

input.py module

```python
# h2_model - read input data and structure

# READ INPUT DATA FUNCTIONS (from DB or others)
#       transform_excel_scenario_sheets_to_dfs_in_dict
#       establish_connection_with_db
#       import_id_cells_data_from_db
#       import_tech_costs_data_from_db
#       import_crp_data_from_db
#       import_single_cell_cf_series_frem
#       import_single_cell_ann_costs
#       import_exclusive_very_weak_wind_cells
#       import_single_cell_scenario_ann_costs
#       save_import_as_feather_file
#       load_input_feather_file

# CREATE RESULTS STRUCTURE
#       cells_results_structure

# DATAFRAME INPUTS IN PYPSA
#       import_pypsa_dataframes

########################################################################################
################################

import os
import psycopg2

import openpyxl
import pandas as pd
import shutil


########################################################################################
################################


def transform_excel_scenario_sheets_to_dfs_in_dict(excel_name):
    excel_path = os.getcwd() + "\\" + excel_name + ".xlsx"
    excel = openpyxl.load_workbook(excel_path)
    dict_model_components = {}
    for sheet_name in excel.sheetnames:
        component_df = pd.read_excel(excel_path, sheet_name, dtype=object, index_col=0)
        dict_model_components[sheet_name] = component_df

    return dict_model_components


def establish_connection_with_db():
    # Establish a connection to the DB
    while True:
        dbuser = "mmartinezperez"
        dbpass = "AbMamp@1797_doble"
        try:
            conn = psycopg2.connect(dbname='rem', host='10.71.0.11', port='5432',
user=dbuser, password=dbpass)
```

```
            break
        except Exception as e:
            print('Failed to connect to DB: ' + str(e))

    cur = conn.cursor()

    return cur, conn


def import_id_cells_data_from_db(cur):
    cur.execute("""SELECT * FROM u_mmartinezperez.t_h2_model_cells;""")

    """SELECT * FROM u_mmartinezperez.t_h2_model_cells;"""

    """SELECT a.id_pk as id_cell, a.iso[array_position(a.area_share,
array_max(a.area_share))] as id_country
    FROM u_mmartinezperez.t_h2_model_merra2_world as a
    JOIN u_mmartinezperez.t_h2_model_100_random_cells as b using (id_pk);"""

    data = cur.fetchall()
    cols = []
    for elt in cur.description:
        cols.append(elt[0])
    id_cells_data = pd.DataFrame(data=data, columns=cols)

    return id_cells_data


def import_tech_costs_data_from_db(cur):
    cur.execute("""select CASE WHEN technology ilike '%PVA - Open%' then 'pv'
                            WHEN technology ilike '%WEA_very_weak%' then
'wind_very_weak'
                            WHEN technology ilike '%WEA_weak%' then 'wind_weak'
                            WHEN technology ilike '%WEA_middle%' then 'wind_middle'
                            WHEN technology ilike '%WEA_strong%' then 'wind_strong'
                            WHEN technology ilike '%WEA_very_strong%' then
'wind_very_strong'
                            WHEN technology ilike '%tank%' then 'tank'
                            WHEN technology ilike '%battery%' then 'battery'
                            WHEN technology ilike '%electrolyzer%' then 'electrolyzer'
end as technology,
                       investment_cost_power,
                       case when investment_cost_capacity is null then 0
                           else investment_cost_capacity end as
investment_cost_capacity,
                       operational_cost_power,
                       case when operational_cost_capacity is null then 0
                           else operational_cost_capacity end as
operational_cost_capacity,
                       lifetime,
                       wacc,
                       case when e_p_ratio is null then 0
                           else e_p_ratio end as e_p_ratio
                from u_mmartinezperez.v_h2_model_costs
                where technology not ilike '%PVA - Roof%';""")
    data = cur.fetchall()
    cols = []
```

```
    for elt in cur.description:
        cols.append(elt[0])
    tech_costs_data = pd.DataFrame(data=data, columns=cols)

    return tech_costs_data


def import_crp_data_from_db(cur):
    # Country Risk Premium Data
    cur.execute("SELECT * FROM u_mmartinezperez.v_h2_model_country_risk_premium")
    data = cur.fetchall()
    cols = []
    for elt in cur.description:
        cols.append(elt[0])
    crp_data = pd.DataFrame(data=data, columns=cols)

    return crp_data


def import_single_cell_cf_series_frem(cur, id_cell):
    # Single Cell Capacity Factors Timeseries (PV and Wind) Data
    cur.execute("""SELECT technology, series
                    FROM  u_mmartinezperez.t_merra2_pv_max_erz
                    WHERE id_region=(%s)
                    UNION ALL
                    select case when id_intern=101 then 'wind_very_weak'
                                when id_intern=102 then 'wind_weak'
                                when id_intern=103 then 'wind_middle'
                                when id_intern=105 then 'wind_strong'
                                when id_intern=104 then 'wind_very_strong' end as
technology,
                           wert as series
                    FROM merra2.wea_erz_ft
                    WHERE wetterjahr=2012 and id_region=(%s) and id_intern in
(101,102,103,104,105);""",
                (id_cell, id_cell))
    data = cur.fetchall()
    cols = []
    for elt in cur.description:
        cols.append(elt[0])
    cell_data = pd.DataFrame(data=data, columns=cols)
    cell_data = cell_data.set_index("technology")

    return cell_data


def import_single_cell_ann_costs(cur, id_country):
    # Single Cell for the specific country annualized costs
    cur.execute("""SELECT *
                    FROM  u_mmartinezperez.t_h2_model_ann_costs_per_country
                    WHERE id_country=(%s);""", (id_country,))
    data = cur.fetchall()
    cols = []
    for elt in cur.description:
        cols.append(elt[0])
    cell_costs_data = pd.DataFrame(data=data, columns=cols)
```

```python
    cell_costs_data = cell_costs_data.pivot(index="id_country", columns="parameter",
values="value")

    return cell_costs_data


def import_exclusive_very_weak_wind_cells(cur):
    # id_cells to exclude
    cur.execute("""select id_region as id_cell
                from u_mmartinezperez.v_max_auslegung_140m_cells as a
                where v_max_complex_140m > 50;""")
    data = cur.fetchall()
    cols = []
    for elt in cur.description:
        cols.append(elt[0])
    cells_to_exclude = pd.DataFrame(data=data, columns=cols)

    return cells_to_exclude


def import_single_cell_scenario_ann_costs(cur, constant_wacc, affordable_battery,
free_h2_storage, id_cell, id_country):
    cells_to_exclude = import_exclusive_very_weak_wind_cells(cur)
    if constant_wacc:
        single_cell_ann_costs = import_single_cell_ann_costs(cur, str(276))
        if id_cell in cells_to_exclude["id_cell"].values:
            new_wind_cap_cost = single_cell_ann_costs.iloc[0]["wind_very_weak"] * 10000
            single_cell_ann_costs.loc[id_country, "wind_very_weak"] = new_wind_cap_cost
    if affordable_battery:
        single_cell_ann_costs = import_single_cell_ann_costs(cur, str(id_country))
        new_battery_cost = single_cell_ann_costs.iloc[0]["battery"] * 0.5
        single_cell_ann_costs.loc[id_country, "battery"] = new_battery_cost
        if id_cell in cells_to_exclude["id_cell"].values:
            new_wind_cap_cost = single_cell_ann_costs.iloc[0]["wind_very_weak"] * 10000
            single_cell_ann_costs.loc[id_country, "wind_very_weak"] = new_wind_cap_cost
    if free_h2_storage:
        single_cell_ann_costs = import_single_cell_ann_costs(cur, str(id_country))
        single_cell_ann_costs.loc[id_country, "tank"] = 0
        if id_cell in cells_to_exclude["id_cell"].values:
            new_wind_cap_cost = single_cell_ann_costs.iloc[0]["wind_very_weak"] * 10000
            single_cell_ann_costs.loc[id_country, "wind_very_weak"] = new_wind_cap_cost
    else:
        single_cell_ann_costs = import_single_cell_ann_costs(cur, str(id_country))
        if id_cell in cells_to_exclude["id_cell"].values:
            new_wind_cap_cost = single_cell_ann_costs.iloc[0]["wind_very_weak"] * 10000
            single_cell_ann_costs.loc[id_country, "wind_very_weak"] = new_wind_cap_cost

    return single_cell_ann_costs


def save_import_as_feather_file(df, name_feather_file):
    # name_feather_file = df_variable_name
    if df.first_valid_index() != 0:
        df.reset_index(inplace=True)

    tmp_feather_path = os.getcwd() + "\\import_data_feather"
```

```
    tmp_feather_file_path = os.getcwd() + "\\import_data_feather\\" + name_feather_file
+ ".feather"
    if not os.path.exists(tmp_feather_path):
        os.mkdir(tmp_feather_path)
    df.to_feather(tmp_feather_file_path)


def load_input_feather_file(name_feather_file):
    # name_feather_file = without .feather extension
    tmp_feather_file_path = os.getcwd() + "\\import_data_feather\\" + name_feather_file
+ ".feather"
    df_to_load = pd.read_feather(tmp_feather_file_path)

    return df_to_load


def create_cells_results_structure(store_file_for_each_cell, id_cells_df,
h2_scenario_id):
    # import_data_from_frem_db must be called before, to take the id_cell as input
    current_path = os.getcwd()

    # Delete id_cells directory and create it again
    id_cells_path = current_path + "\\results_id_scenario_" + str(h2_scenario_id)
    if os.path.exists(id_cells_path):
        shutil.rmtree(id_cells_path)
    os.mkdir(id_cells_path)

    # Create as many directories as cells
    if store_file_for_each_cell:
        for index_cell, data_row_cell in id_cells_df.iterrows():
            single_cell_path = id_cells_path + "\\" + str(data_row_cell["id_cell"])
            os.mkdir(single_cell_path)


def import_pypsa_dataframes(network, dict_components):
    network.set_snapshots(dict_components["snapshots"].index)

    # Static Parameters
    if "buses" in dict_components:
        network.import_components_from_dataframe(dict_components["buses"], "Bus")
    if "carriers" in dict_components:
        network.import_components_from_dataframe(dict_components["carriers"], "Carrier")
    if "generators" in dict_components:
        network.import_components_from_dataframe(dict_components["generators"],
"Generator")
    if "global_constraints" in dict_components:
        network.import_components_from_dataframe(dict_components["global_constraints"],
"GlobalConstraint")
    if "links" in dict_components:
        network.import_components_from_dataframe(dict_components["links"], "Link")
    if "lines" in dict_components:
        network.import_components_from_dataframe(dict_components["lines"], "Line")
    if "loads" in dict_components:
        network.import_components_from_dataframe(dict_components["loads"], "Load")
    if "shunt_impedances" in dict_components:
        network.import_components_from_dataframe(dict_components["shunt_impedances"],
"ShuntImpedance")
```

```
    if "storage_units" in dict_components:
        network.import_components_from_dataframe(dict_components["storage_units"],
"StorageUnit")
    if "stores" in dict_components:
        network.import_components_from_dataframe(dict_components["stores"], "Store")
    if "transformers" in dict_components:
        network.import_components_from_dataframe(dict_components["transformers"],
"Transformer")

    # Dynamic Parameters
    if "generators-p_max_pu" in dict_components:
        network.import_series_from_dataframe(dict_components["generators-p_max_pu"],
"Generator", "p_max_pu")
    if "generators-p_min_pu" in dict_components:
        network.import_series_from_dataframe(dict_components["generators-p_min_pu"],
"Generator", "p_min_pu")
    if "links-p_max_pu" in dict_components:
        network.import_series_from_dataframe(dict_components["links-p_max_pu"], "Link",
"p_max_pu")
    if "links-p_min_pu" in dict_components:
        network.import_series_from_dataframe(dict_components["links-p_min_pu"], "Link",
"p_min_pu")
    if "storage_units-p_max_pu" in dict_components:
        network.import_series_from_dataframe(dict_components["storage_units-p_max_pu"],
"StorageUnit", "p_max_pu")
    if "storage_units-p_min_pu" in dict_components:
        network.import_series_from_dataframe(dict_components["storage_units-p_min_pu"],
"StorageUnit", "p_min_pu")
    if "storage_units-state_of_charge_set" in dict_components:
        network.import_series_from_dataframe(dict_components["storage_units-
state_of_charge_set"], "StorageUnit",
                                             "state_of_charge_set")
    if "storage_units-soc" in dict_components:
        dict_components["storage_units-state_of_charge_set"] =
dict_components["storage_units-soc"]
        network.import_series_from_dataframe(dict_components["storage_units-
state_of_charge_set"], "StorageUnit",
                                             "state_of_charge_set")
    if "stores-e_max_pu" in dict_components:
        network.import_series_from_dataframe(dict_components["stores-e_max_pu"],
"Store", "e_max_pu")
    if "stores-e_min_pu" in dict_components:
        network.import_series_from_dataframe(dict_components["stores-e_min_pu"],
"Store", "e_min_pu")
```

precalc.py module

```python
# h2_model - precalculations

# INTERMEDIATE FUNCTIONS
#       append_value_to_dict
#       append_new_row_to_dict
#       merge_dictionaries
#       transform_dict_to_df

# PRECALCULATION FUNCTIONS
#       calc_single_country_ann_costs
#       calculate_tech_costs_per_country

# SPECIFIC DF UPDATE FOR PYPSA INPUT
#       update_gen_cf_timeseries_for_specific_cell
#       update_comp_cap_costs_for_specific_cell

########################################################################################
################################

import pandas as pd

########################################################################################
################################


def append_value_to_dict(dict_obj, key, value):
    # Append a specific value in a specific key of a dictionary
    # Check if key exist in dict or not
    if key in dict_obj:
        # Key exist in dict.
        # Check if type of value of key is list or not
        if not isinstance(dict_obj[key], list):
            # If type is not list then make it list
            dict_obj[key] = [dict_obj[key]]
        # Append the value in list
        dict_obj[key].append(value)
    else:
        # As key is not in dict,
        # so, add key-value pair
        dict_obj[key] = value


def append_new_row_to_dict(dict_obj, dict_new_row):
    for key in dict_new_row:
        if key in dict_obj:
            # Key exist in dict.
            # Check if type of value of key is list or not
            if not isinstance(dict_obj[key], list):
                # If type is not list then make it list
                dict_obj[key] = [dict_obj[key]]
            # Append the value in list
            dict_obj[key].append(dict_new_row[key])
        else:
            # As key is not in dict,
            # so, add key-value pair
```

```python
            dict_obj[key] = dict_new_row[key]


def merge_dictionaries(dict_obj, dict_new):
    for key, values in dict_new.items():
        if key in dict_obj:
            # Key exist in dict.
            # Check if type of value of key is list or not
            if not isinstance(dict_obj[key], list):
                # If type is not list then make it list
                dict_obj[key] = [dict_obj[key]]
            # Append the value or list
            if not isinstance(values, list):
                dict_obj[key] = dict_obj[key] + [values]
            else:
                dict_obj[key] = dict_obj[key] + values
        else:
            # As key is not in dict,
            # so, add key-value pair
            dict_obj[key] = values


def transform_dict_to_df(dictionary, transform_to_float):
    df = pd.DataFrame(dictionary)
    df = df.iloc[1:, :]
    df.reset_index(drop=True, inplace=True)
    if transform_to_float:
        df = df.astype(float)
    return df


def calc_single_country_ann_costs(inv_cost_pow, inv_cost_cap, op_cost_pow, op_cost_cap,
                                  lifetime, wacc, e_p_ratio, crp):
    # Calculate for a country the annualized costs of each technology considering the
specific Country Risk Premium
    if inv_cost_cap != 0 and e_p_ratio != 0:                # batteries (maybe also
future tanks)
        inv_cost = inv_cost_cap * e_p_ratio + inv_cost_pow
        annualized_cost = inv_cost * (
                ((wacc + crp) * (1 + wacc + crp) ** lifetime) / ((1 + wacc + crp) **
lifetime - 1) + op_cost_pow)
        e_p_cost_ratio = "NaN"
    elif inv_cost_cap != 0 and e_p_ratio == 0:              # tanks (maybe also future
batteries)
        annualized_cost = inv_cost_pow * (
                ((wacc + crp) * (1 + wacc + crp) ** lifetime) / ((1 + wacc + crp) **
lifetime - 1) + op_cost_pow)
        annualized_cost_cap = inv_cost_cap * (
                ((wacc + crp) * (1 + wacc + crp) ** lifetime) / ((1 + wacc + crp) **
lifetime - 1) + op_cost_cap)
        e_p_cost_ratio = annualized_cost_cap/annualized_cost
    else:                                                   # generators and links
        annualized_cost = inv_cost_pow * (
                ((wacc + crp) * (1 + wacc + crp) ** lifetime) / ((1 + wacc + crp) **
lifetime - 1) + op_cost_pow)
        e_p_cost_ratio = "NaN"
```

```
    return annualized_cost, e_p_cost_ratio


def calculate_tech_costs_per_country(tech_costs_data, crp_data):
    # Calculate DataFrame with annualized costs per country and component, with country
specific WACC (WACC + CRP)
    costs_components_dict = {}

    # Loop to fill the dictionary with the calculated annualized costs per country
(rows) and component (keys)
    # Batteries and Tanks have a capacity and a power cost, so they have to be
considered differently
    for index_crp, data_row_crp in crp_data.iterrows():
        append_value_to_dict(costs_components_dict, "id_country",
int(data_row_crp["id_country"]))
        for index_costs, data_row_costs in tech_costs_data.iterrows():
            (value, e_p_cost_ratio) =
calc_single_country_ann_costs(data_row_costs["investment_cost_power"],

data_row_costs["investment_cost_capacity"],

data_row_costs["operational_cost_power"],

data_row_costs["operational_cost_capacity"],

data_row_costs["lifetime"],

data_row_costs["wacc"],

data_row_costs["e_p_ratio"],
                                                                    data_row_crp["crp"])
            append_value_to_dict(costs_components_dict, data_row_costs["technology"],
value)
            if e_p_cost_ratio != "NaN":
                append_value_to_dict(costs_components_dict, data_row_costs["technology"]
+ "_e_p_cost_ratio",
                                     e_p_cost_ratio)

    ann_costs_per_country = pd.DataFrame(costs_components_dict)

    return ann_costs_per_country


def update_gen_cf_timeseries_for_specific_cell(dict_scenario_components,
cell_weather_series_data):
    for gen in dict_scenario_components["generators"].index:
        new_timeseries = cell_weather_series_data.loc[gen, "series"]
        dict_scenario_components["generators-p_max_pu"][gen] = new_timeseries

    return dict_scenario_components


def update_comp_cap_costs_for_specific_cell(dict_scenario_components,
single_cell_ann_costs):
    if "generators" in dict_scenario_components:
        for index, data_row_gen in dict_scenario_components["generators"].iterrows():
            if index in single_cell_ann_costs.columns:
```

```
                new_capital_cost = single_cell_ann_costs.iloc[0][index]
                dict_scenario_components["generators"].loc[index, "capital_cost"] =
new_capital_cost

    if "links" in dict_scenario_components:
        for index, data_row_link in dict_scenario_components["links"].iterrows():
            if index in single_cell_ann_costs.columns:
                new_capital_cost = single_cell_ann_costs.iloc[0][index]
                dict_scenario_components["links"].loc[index, "capital_cost"] =
new_capital_cost

    if "storage_units" in dict_scenario_components:
        for index, data_row_storage in
dict_scenario_components["storage_units"].iterrows():
            if index in single_cell_ann_costs.columns:
                new_capital_cost = single_cell_ann_costs.iloc[0][index]
                dict_scenario_components["storage_units"].loc[index, "capital_cost"] =
new_capital_cost

    if "stores" in dict_scenario_components:
        for index, data_row_store in dict_scenario_components["stores"].iterrows():
            if index in single_cell_ann_costs.columns:
                new_capital_cost = single_cell_ann_costs.iloc[0][index]
                dict_scenario_components["stores"].loc[index, "capital_cost"] =
new_capital_cost

    return dict_scenario_components
```

opt.py

```
# h2_model - optimization module

# DYNAMIC INPUT DATA: for each cell
#         Solar capacity factor timeseries
#         Wind capacity factor timeseries
#         Annualized capital cost per country
#                   PV, Wind, Electrolyzer, Battery, Tank
#                   Wind: different turbines already in MERRA2

# OPTIMIZATION FUNCTIONS
#       optimize_chunk_of_cells
#       parallelize_df

########################################################################################
################################

import copy
import logging
import multiprocessing as mp
import os
import time

import numpy as np
import pypsa

from tqdm import tqdm
import input
import precalc
import profiling
import extra_pypsa
import output

logger = logging.getLogger(__name__)
logging.disable()
########################################################################################
################################


def optimize_chunk_of_cells(id_cell_data_df, dict_scen_components, h2_scenario_id,
constant_wacc, affordable_battery,
                            free_h2_storage, store_cells_as_files):
    current_path = os.getcwd()
    (cur, conn) = input.establish_connection_with_db()
    scenario_id = str(h2_scenario_id)

    dict_new_comp = copy.deepcopy(dict_scen_components)
    override_component_attrs = extra_pypsa.multiple_input_output()

    tqdm_disable = False if id_cell_data_df.first_valid_index() == 0 else True
    id_cell_data_df.reset_index(drop=True, inplace=True)
    results_chunk = {}

    for index_cell, data_row_cell in tqdm(id_cell_data_df.iterrows(),
total=id_cell_data_df.shape[0],
```

```
                                          desc="Aproximate optimization progress for
each core: ", position=0,
                                          disable=tqdm_disable, ncols=100):

        time_1 = time.time()
        id_cell = str(data_row_cell["id_cell"])
        single_cell_results_path_csv = current_path + "\\results_id_scenario_" +
scenario_id + "\\" + id_cell
        single_cell_results_path_netcdf = current_path + "\\results_id_scenario_" +
scenario_id + "\\" + id_cell + ".nc"

        # prepare single cell data
        single_cell_series_data = input.import_single_cell_cf_series_frem(cur, id_cell)
        dict_new_comp =
precalc.update_gen_cf_timeseries_for_specific_cell(dict_new_comp,
single_cell_series_data)

        single_cell_ann_costs = input.import_single_cell_scenario_ann_costs(cur,
constant_wacc, affordable_battery,

free_h2_storage,

data_row_cell["id_cell"],

data_row_cell["id_country"])
        dict_new_comp = precalc.update_comp_cap_costs_for_specific_cell(dict_new_comp,
single_cell_ann_costs)
        time_2 = time.time()

        network = pypsa.Network(name="Hydrogen_Island_" + id_cell + "_" +
str(data_row_cell["id_country"]),
                                override_component_attrs=override_component_attrs)
        input.import_pypsa_dataframes(network, dict_new_comp)
        extra_pypsa.replace_storage_units(network, single_cell_ann_costs)

        profile_each_optimization = False
        cp = profiling.start_profiling() if profile_each_optimization else 0

        time_3 = time.time()
        network.lopf(network.snapshots, pyomo=False, solver_name="gurobi",
solver_options={"OutputFlag": 0,

"LogToConsole": 0,

"threads": 2})
        time_4 = time.time()

        if profile_each_optimization:
            name_csv_profiling_file = current_path + "\\performance\\cell_" + id_cell
            profiling.end_profiling(cp, name_csv_profiling_file)

        if store_cells_as_files:
            network.export_to_csv_folder(single_cell_results_path_csv)
            # network.export_to_netcdf(single_cell_results_path_netcdf)
            # output.save_network_df_to_parquets(network, h2_scenario_id, id_cell)
        time_5 = time.time()
```

```
        # Get data out
        (results_row, results_timeseries_row) =
output.calculate_new_row_results(network, data_row_cell["id_cell"],

data_row_cell["id_country"])
        precalc.append_new_row_to_dict(results_chunk, results_row)
        output.insert_cell_preresults_timeseries_in_db(results_timeseries_row,
data_row_cell["id_cell"], h2_scenario_id)

        time_6 = time.time()

        print_cell_times = False
        if print_cell_times:
            print("\nZeit Vorbereitung Components DataFrame: ", time_2 - time_1)
            print("Zeit PyPSA Override + import + extra_fun", time_3 - time_2)
            print("Zeit PyPSA Optimierung: ", time_4 - time_3)
            print("Zeit PyPSA CSV Files Export: ", time_5 - time_4)
            print("Zeit für Output: ", time_6 - time_5)
            print("\nCell " + id_cell + " optimization completed")

    cur.close()
    return results_chunk


def parallelize_df(num_parallel_processes, df, func):
    if num_parallel_processes > mp.cpu_count()/2:
        num_parallel_processes = mp.cpu_count()//2
    data_split = np.array_split(df, num_parallel_processes)
    with mp.Pool(num_parallel_processes) as pool:
        results = {}
        for results_chunk in pool.imap(func, data_split):
            precalc.merge_dictionaries(results, results_chunk)
        # Shutdown the process pool
        pool.close()
        # Wait for all issued tasks to complete
        pool.join()

    return results
```

extra_pypsa.py module

```
# h2_modell - pypsa additional functions

# PYPSA EXTRA FUNCTIONS
#       multiple_input_output
#       replace_single_su (pypsa extra_functionality)
#       replace_storage_units (pypsa extra_functionality)

#####################################################################################
################################

import numpy as np
import pandas as pd
import pypsa

#####################################################################################
################################


def multiple_input_output():
    # To tell PyPSA that links will have a 2nd bus by overriding the component_attrs.
    # This is needed so that electrolyzer or other components can have more than just
one input or output
    override_component_attrs = pypsa.descriptors.Dict({k: v.copy()
                                                       for k, v in
pypsa.components.component_attrs.items()})

    override_component_attrs["Link"].loc["bus2"] = [
        "string",
        np.nan,
        np.nan,
        "2nd bus",
        "Input (optional)",
    ]
    override_component_attrs["Link"].loc["bus3"] = [
        "string",
        np.nan,
        np.nan,
        "3rd bus",
        "Input (optional)",
    ]
    override_component_attrs["Link"].loc["efficiency2"] = [
        "static or series",
        "per unit",
        1.0,
        "2nd bus efficiency",
        "Input (optional)",
    ]
    override_component_attrs["Link"].loc["efficiency3"] = [
        "static or series",
        "per unit",
        1.0,
        "3rd bus efficiency",
        "Input (optional)",
    ]
    override_component_attrs["Link"].loc["p2"] = [
```

```
        "series",
        "MW",
        0.0,
        "2nd bus output",
        "Output",
    ]
    override_component_attrs["Link"].loc["p3"] = [
        "series",
        "MW",
        0.0,
        "3rd bus output",
        "Output",
    ]

    return override_component_attrs


def replace_single_su(network, su_to_replace, e_p_cost_ratio):
    """Replace the storage unit su_to_replace with a bus for the energy
    carrier, two links for the conversion of the energy carrier to and from electricity,
    a store to keep track of the depletion of the energy carrier and its
    CO2 emissions, and a variable generator for the storage inflow.

    Because this function can only be entered by those components with free energy size
and power size ratio,
    no extra functionality is needed to add a constraint. e/p ratio will be free"""

    su = network.storage_units.loc[su_to_replace]

    bus_name = "{}_{}".format(su["bus"], su["carrier"])
    link_1_name = "{}_pow_store_of_{}".format(su_to_replace, su["carrier"])
    link_2_name = "{}_pow_dispatch_of_{}".format(su_to_replace, su["carrier"])
    store_name = "{}_cap_store_unit_{}".format(su_to_replace, su["carrier"])

    network.add("Bus", bus_name, carrier=su["carrier"])

    # dispatch link
    network.add(
        "Link",
        link_2_name,
        bus0=bus_name,
        bus1=su["bus"],
        capital_cost=(su["capital_cost"] * su["efficiency_dispatch"]/2),
        p_nom=su["p_nom"] / su["efficiency_dispatch"],
        p_nom_extendable=su["p_nom_extendable"],
        p_nom_max=su["p_nom_max"] / su["efficiency_dispatch"],
        p_nom_min=su["p_nom_min"] / su["efficiency_dispatch"],
        p_max_pu=su["p_max_pu"],
        marginal_cost=su["marginal_cost"] * su["efficiency_dispatch"],
        efficiency=su["efficiency_dispatch"],
    )

    # store link
    network.add(
        "Link",
        link_1_name,
        bus1=bus_name,
```

```
        bus0=su["bus"],
        capital_cost=(su["capital_cost"] * su["efficiency_store"] / 2),
        p_nom=su["p_nom"],
        p_nom_extendable=su["p_nom_extendable"],
        p_nom_max=su["p_nom_max"],
        p_nom_min=su["p_nom_min"],
        p_max_pu=-su["p_min_pu"],
        efficiency=su["efficiency_store"],
    )

    if (
        su_to_replace in network.storage_units_t.state_of_charge_set.columns
        and (
            ~pd.isnull(network.storage_units_t.state_of_charge_set[su_to_replace])
        ).any()
    ):
        e_max_pu = pd.Series(data=1.0, index=network.snapshots)
        e_min_pu = pd.Series(data=0.0, index=network.snapshots)
        non_null = ~pd.isnull(
            network.storage_units_t.state_of_charge_set[su_to_replace]
        )
        e_min_pu[non_null] = network.storage_units_t.state_of_charge_set[su_to_replace][
            non_null
        ]
    else:
        e_max_pu = 1.0
        e_min_pu = 0.0

    # Battery FfE has e_p_cost_ratio = 0. All capital cost in Link €/MW (as e_p_ratio =
max_hours is fixed)
    # Tank in FfE has e_p_cost_ratio != 0. Capital cost in Link €/MW and in Store €/MWh
    network.add(
        "Store",
        store_name,
        bus=bus_name,
        e_nom=su["p_nom"] * su["max_hours"],
        e_nom_min=su["p_nom_min"] / su["efficiency_dispatch"] * su["max_hours"],
        e_nom_max=su["p_nom_max"] / su["efficiency_dispatch"] * su["max_hours"],
        e_nom_extendable=su["p_nom_extendable"],
        e_max_pu=e_max_pu,
        e_min_pu=e_min_pu,
        standing_loss=su["standing_loss"],
        e_cyclic=su["cyclic_state_of_charge"],
        e_initial=su["state_of_charge_initial"],
        capital_cost=su["capital_cost"] * e_p_cost_ratio
    )

    network.remove("StorageUnit", su_to_replace)


def replace_storage_units(network, single_cell_ann_costs):
    e_p_cost_ratio_df = single_cell_ann_costs.filter(regex="e_p_cost_ratio")
    e_p_cost_ratio_df.columns = e_p_cost_ratio_df.columns.str.replace("_e_p_cost_ratio",
"")
    e_p_cost_ratio_df = e_p_cost_ratio_df.transpose()
    e_p_cost_ratio_df.rename(columns={e_p_cost_ratio_df.columns[0]: "e_p_cost_ratio"},
inplace=True)
```

```python
idx_su_network = network.storage_units.index
idx_su_costs = e_p_cost_ratio_df.index
intersection = idx_su_network.intersection(idx_su_costs)

if not intersection.empty:
    for index, data_row_e_p in e_p_cost_ratio_df.iterrows():
        if index in intersection:
            replace_single_su(network, index, data_row_e_p["e_p_cost_ratio"])
```

output.py module

```
# h2_model - output
# here we define the functions for the output of the pypsa optimization network.
# That is which output do we want for each cell

# WORK WITH OUTPUT FUNCTIONS
#       curtailment_diagramms
#       prepare_results_new_row
#       prepare_results_timeseries_new_row
#       calculate_new_row_results
#       insert_id_scenario_time

# INSERT/TRUNCATE IN DB FUNCTIONS
#       insert_cell_preresults_timeseries_in_db
#       truncate_preresults_timeseries_in_db
#       insert_scenario_in_db
#       insert_results_in_db
#       insert_results_timeseries_in_db
#       insert_ann_costs_in_db

# ALTERANTIVE SAVE AND LOAD FUNCTIONS
#       save_results_as_feather
#       load_results_from_feather
#       save_results_as_parquet
#       load_results_from_parquet
#       save_network_df_to_parquets
#       load_single_network_from_parquets
#       load_all_scenario_networks_from_parquets
#

##########################################################################################
################################

import git
import os
import pandas as pd
import psycopg2
import shutil
from io import StringIO
import sys
import datetime

import input
import precalc

##########################################################################################
################################


def curtailment_diagrams():
    """
    ### Curtailment Graph ###
    p_by_carrier = network.generators_t.p.groupby(network.generators.carrier,
axis=1).sum()
    #p_by_carrier.drop(
    #     (p_by_carrier.max()[p_by_carrier.max() < 1700.0]).index, axis=1, inplace=True
```

```
    #)
    # print(p_by_carrier)

    carrier = "wind"

    capacity = network.generators.groupby("carrier").sum().at[carrier, "p_nom_opt"]
    p_available =
network.generators_t.p_max_pu.multiply(network.generators["p_nom_opt"])
    p_available_by_carrier = p_available.groupby(network.generators.carrier,
axis=1).sum()
    p_curtailed_by_carrier = p_available_by_carrier - p_by_carrier
    p_df = pd.DataFrame(
        {
            carrier + " available": p_available_by_carrier[carrier],
            carrier + " dispatched": p_by_carrier[carrier],
            carrier + " curtailed": p_curtailed_by_carrier[carrier],
        }
    )

    p_df[carrier + " capacity"] = capacity
    #p_df["Wind Onshore curtailed"][p_df["Wind Onshore curtailed"] < 0.0] = 0.0
    fig, ax = plt.subplots(figsize=(15, 4))
    p_df[[carrier + " dispatched", carrier + " curtailed"]].plot(
        kind="area", ax=ax, linewidth=1
    )
    p_df[[carrier + " available", carrier + " capacity"]].plot(ax=ax, linewidth=1)

    ax.set_xlabel("")
    ax.set_ylabel("Power [MW]")
    ax.set_ylim([0, 0.15])
    ax.legend()
    fig.tight_layout()

    plt.savefig("curtailment_"+ carrier + id_cell + ".svg", dpi=150)

    carrier = "solar"

    capacity = network.generators.groupby("carrier").sum().at[carrier, "p_nom_opt"]
    p_available =
network.generators_t.p_max_pu.multiply(network.generators["p_nom_opt"])
    p_available_by_carrier = p_available.groupby(network.generators.carrier,
axis=1).sum()
    p_curtailed_by_carrier = p_available_by_carrier - p_by_carrier
    p_df = pd.DataFrame(
        {
            carrier + " available": p_available_by_carrier[carrier],
            carrier + " dispatched": p_by_carrier[carrier],
            carrier + " curtailed": p_curtailed_by_carrier[carrier],
        }
    )

    p_df[carrier + " capacity"] = capacity
    #p_df["Wind Onshore curtailed"][p_df["Wind Onshore curtailed"] < 0.0] = 0.0
    fig, ax = plt.subplots(figsize=(15, 4))
    p_df[[carrier + " dispatched", carrier + " curtailed"]].plot(
        kind="area", ax=ax, linewidth=1
    )
```

```
    p_df[[carrier + " available", carrier + " capacity"]].plot(ax=ax, linewidth=1)

    ax.set_xlabel("")
    ax.set_ylabel("Power [MW]")
    ax.set_ylim([0, 0.6])
    ax.legend()
    fig.tight_layout()

    plt.savefig("curtailment_"+ carrier + id_cell + ".svg", dpi=150)


    curt = pypsa.stats.calculate_curtailment(network)
    print("Curtailment is:", curt)
    """


def calculate_results_for_cell(network):
    yearly_cost = 0
    cell_opt_comp_size_dict = {}
    for gen, data_row_gen in network.generators.iterrows():
        yearly_cost += data_row_gen["p_nom_opt"] * data_row_gen["capital_cost"]
        precalc.append_value_to_dict(cell_opt_comp_size_dict, gen,
data_row_gen["p_nom_opt"])
    for link, data_row_link in network.links.iterrows():
        yearly_cost += data_row_link["p_nom_opt"] * data_row_link["capital_cost"]
        precalc.append_value_to_dict(cell_opt_comp_size_dict, link,
data_row_link["p_nom_opt"])
    for store, data_row_store in network.stores.iterrows():
        yearly_cost += data_row_store["e_nom_opt"] * data_row_store["capital_cost"]
        precalc.append_value_to_dict(cell_opt_comp_size_dict, store,
data_row_store["e_nom_opt"])
    for su, data_row_su in network.storage_units.iterrows():
        yearly_cost += data_row_su["p_nom_opt"] * data_row_su["capital_cost"]
        precalc.append_value_to_dict(cell_opt_comp_size_dict, su,
data_row_su["p_nom_opt"])

    # Define the low heating value of hydrogen MWh/kg
    h2_lhv = 0.03333
    h2_amount = network.loads.loc["load", "p_set"] * len(network.snapshots) / h2_lhv

    cell_lcoh = yearly_cost/h2_amount

    return cell_lcoh, cell_opt_comp_size_dict


def prepare_results_new_row(id_cell, id_country, cell_lcoh, cell_opt_comp_size_dict):
    results_new_row = {"id_region": id_cell, "id_country": id_country, "lcoh":
cell_lcoh}
    results_new_row.update(cell_opt_comp_size_dict)

    return results_new_row


def prepare_results_timeseries_new_row(network):
    results_timeseries_new_row = {}
    for (columnName, columnData) in network.generators_t.p.iteritems():
        key_name = "generator_p_" + columnName
```

```
        precalc.append_value_to_dict(results_timeseries_new_row, key_name,
str(columnData.to_numpy().tolist()))
    for (columnName, columnData) in network.links_t.mu_lower.iteritems():
        key_name = "link_mu_lower_" + columnName
        precalc.append_value_to_dict(results_timeseries_new_row, key_name,
str(columnData.to_numpy().tolist()))
    for (columnName, columnData) in network.links_t.mu_upper.iteritems():
        key_name = "link_mu_upper_" + columnName
        precalc.append_value_to_dict(results_timeseries_new_row, key_name,
str(columnData.to_numpy().tolist()))
    for (columnName, columnData) in network.links_t.p0.iteritems():
        key_name = "link_p0_" + columnName
        precalc.append_value_to_dict(results_timeseries_new_row, key_name,
str(columnData.to_numpy().tolist()))
    for (columnName, columnData) in network.links_t.p1.iteritems():
        key_name = "link_p1_" + columnName
        precalc.append_value_to_dict(results_timeseries_new_row, key_name,
str(columnData.to_numpy().tolist()))
    for (columnName, columnData) in network.storage_units_t.state_of_charge.iteritems():
        key_name = "storage_unit_soc_" + columnName
        precalc.append_value_to_dict(results_timeseries_new_row, key_name,
str(columnData.to_numpy().tolist()))
    for (columnName, columnData) in network.storage_units_t.p_dispatch.iteritems():
        key_name = "storage_unit_p_dispatch_" + columnName
        precalc.append_value_to_dict(results_timeseries_new_row, key_name,
str(columnData.to_numpy().tolist()))
    for (columnName, columnData) in network.storage_units_t.p_store.iteritems():
        key_name = "storage_unit_p_store_" + columnName
        precalc.append_value_to_dict(results_timeseries_new_row, key_name,
str(columnData.to_numpy().tolist()))
    for (columnName, columnData) in network.stores_t.e.iteritems():
        key_name = "store_e_" + columnName
        precalc.append_value_to_dict(results_timeseries_new_row, key_name,
str(columnData.to_numpy().tolist()))
    for (columnName, columnData) in network.stores_t.p.iteritems():
        key_name = "store_p_" + columnName
        precalc.append_value_to_dict(results_timeseries_new_row, key_name,
str(columnData.to_numpy().tolist()))
    # Curtailment
    max_pu = network.generators_t.p_max_pu
    p_avail = max_pu.multiply(network.generators.p_nom_opt.loc[max_pu.columns])
    p_used = network.generators_t.p[max_pu.columns]
    p_curtailed = p_avail.sub(p_used, axis=1)
    for (columnName, columnData) in p_curtailed.iteritems():
        key_name = "generator_p_curtailed_" + columnName
        precalc.append_value_to_dict(results_timeseries_new_row, key_name,
str(columnData.to_numpy().tolist()))

    return results_timeseries_new_row


def calculate_new_row_results(network, id_cell, id_country):
    (cell_lcoh, cell_opt_comp_size_dict) = calculate_results_for_cell(network)
    results_new_row = prepare_results_new_row(id_cell, id_country, cell_lcoh,
cell_opt_comp_size_dict)
    results_timeseries_new_row = prepare_results_timeseries_new_row(network)
```

```python
    return results_new_row, results_timeseries_new_row


def insert_id_scenario_time(results_df, h2_scenario_id):
    results_df.insert(1, "id_scenario_h2_model", h2_scenario_id)
    current_time = datetime.datetime.now()
    results_df["time"] = current_time

    return results_df


def insert_cell_preresults_timeseries_in_db(results_timeseries, id_cell,
h2_scenario_id):
    results_series = pd.Series(results_timeseries)
    results_timeseries_db = pd.DataFrame(results_series)
    results_timeseries_db = results_timeseries_db.rename(columns={0: "values"})
    results_timeseries_db["values"] = results_timeseries_db["values"].apply(lambda x:
x.replace("[", "{"))
    results_timeseries_db["values"] = results_timeseries_db["values"].apply(lambda x:
x.replace("]", "}"))
    results_timeseries_db = results_timeseries_db.reset_index()
    results_timeseries_db = results_timeseries_db.rename(columns={"index": "parameter"})
    results_timeseries_db.insert(0, "id_scenario_h2_model", h2_scenario_id)
    results_timeseries_db.insert(0, "id_regionstyp", 59)
    results_timeseries_db.insert(0, "id_region", id_cell)

    buffer = StringIO()
    results_timeseries_db.to_csv(buffer, index=False, header=False, sep="|")
    buffer.seek(0)
    (cur, conn) = input.establish_connection_with_db()
    try:
        cur.copy_expert("COPY u_mmartinezperez.t_h2_model_preresults_timeseries FROM
STDIN DELIMITER '|' CSV", buffer)
        # cur.copy_from(buffer, "rem.u_mmartinezperez.t_h2_model_preresults_timeseries",
sep="|")
    except (Exception, psycopg2.DatabaseError) as error:
        print("Error_a: %s" % error)
        print("Execute Queries failed.")
        conn.rollback()
        cur.close()
    conn.commit()
    cur.close()


def truncate_preresults_timeseries_in_db():
    (cur, conn) = input.establish_connection_with_db()
    try:
        cur.execute("TRUNCATE u_mmartinezperez.t_h2_model_preresults_timeseries")
    except (Exception, psycopg2.DatabaseError) as error:
        print("Error_b: %s" % error)
        print("Execute Queries failed.")
        conn.rollback()
        cur.close()
    conn.commit()
    cur.close()
```

```
def insert_scenario_in_db(h2_scenario_id, scenario_name, scenario_year):
    repo = git.Repo(search_parent_directories=True)
    sha = repo.head.object.hexsha
    sql_text = "see function insert_scenario_in_db in python module output.py "
    sql_check_scenario = "DELETE FROM u_mmartinezperez.t_h2_model_scenario_description
WHERE id_scenario_h2_model = %s;"
    sql_insert_scenario = """INSERT INTO
u_mmartinezperez.t_h2_model_scenario_description (id_scenario_h2_model,
                            description_de, description_en, year, commit, sql) VALUES
(%s, %s, %s, %s, %s, %s);"""
    (cur, conn) = input.establish_connection_with_db()
    try:
        cur.execute(sql_check_scenario, (h2_scenario_id,))
        cur.execute(sql_insert_scenario, (h2_scenario_id, scenario_name, scenario_name,
scenario_year, sha, sql_text))
    except (Exception, psycopg2.DatabaseError) as error:
        print("Error_c: %s" % error)
        print("Execute Queries failed.")
        conn.rollback()
        cur.close()
    conn.commit()
    cur.close()


def insert_results_in_db(overwrite_scenario_results, results_df, h2_scenario_id,
scenario_year):
    results_db = results_df.drop(["id_country", "time"], axis=1)

    results_db = results_db.melt(id_vars=["id_region", "id_scenario_h2_model"],
                                 var_name="parameter",
                                 value_name="value")
    results_db.insert(len(results_db.columns), "year", scenario_year)
    results_db.insert(1, "id_regionstyp", 59)

    buffer = StringIO()
    results_db.to_csv(buffer, index=False, header=False)
    buffer.seek(0)
    sql_check_scenario = "DELETE FROM u_mmartinezperez.t_h2_model_results WHERE
id_scenario_h2_model = %s;"
    sql_truncate_preresults = "TRUNCATE u_mmartinezperez.t_h2_model_preresults;"
    sql_insert_results = """INSERT INTO u_mmartinezperez.t_h2_model_results (id_region,
id_regionstyp,
                                        id_scenario_h2_model, id_parameter, value, year)
                                SELECT id_region,
                                       id_regionstyp,
                                       id_scenario_h2_model,
                                       id_parameter,
                                       value,
                                       year
                                FROM u_mmartinezperez.t_h2_model_preresults as a
                                JOIN u_mmartinezperez.t_h2_model_parameter_description
as b on (a.parameter=b.name);"""

    (cur, conn) = input.establish_connection_with_db()
    try:
        if overwrite_scenario_results:
            cur.execute(sql_check_scenario, (h2_scenario_id,))
```

```
        cur.execute(sql_truncate_preresults)
        cur.copy_expert("COPY u_mmartinezperez.t_h2_model_preresults FROM STDIN CSV",
buffer)
        # cur.copy_from(buffer, "rem.u_mmartinezperez.t_h2_model_preresults", sep=",")
        cur.execute(sql_insert_results)
    except (Exception, psycopg2.DatabaseError) as error:
        print("Error_d: %s" % error)
        print("Execute Queries failed.")
        conn.rollback()
        cur.close()
    conn.commit()
    cur.close()


def insert_results_timeseries_in_db(overwrite_scenario_results, h2_scenario_id,
scenario_year):
    sql_check_scenario = "DELETE FROM u_mmartinezperez.t_h2_model_results_timeseries
WHERE id_scenario_h2_model = %s;"
    sql_insert_results = """INSERT INTO u_mmartinezperez.t_h2_model_results_timeseries
(id_region, id_regionstyp,

id_scenario_h2_model, id_parameter, values)
                             SELECT id_region,
                                    id_regionstyp,
                                    id_scenario_h2_model,
                                    id_parameter,
                                    values
                             FROM u_mmartinezperez.t_h2_model_preresults_timeseries as a
                             JOIN u_mmartinezperez.t_h2_model_parameter_description as b
on (a.parameter=b.name);"""
    sql_update_year_results = """UPDATE u_mmartinezperez.t_h2_model_results_timeseries
                                SET year=%s WHERE id_scenario_h2_model=%s;"""
    (cur, conn) = input.establish_connection_with_db()
    try:
        if overwrite_scenario_results:
            cur.execute(sql_check_scenario, (h2_scenario_id,))
        cur.execute(sql_insert_results)
        cur.execute(sql_update_year_results, (scenario_year, h2_scenario_id))
    except (Exception, psycopg2.DatabaseError) as error:
        print("Error_e: %s" % error)
        print("Execute Queries failed.")
        conn.rollback()
        cur.close()
    conn.commit()
    cur.close()


def insert_ann_costs_in_db(ann_costs_per_country_df, scenario_year):
    if ann_costs_per_country_df.first_valid_index() != 0:
        ann_costs_per_country_df = ann_costs_per_country_df.reset_index()
    ann_costs_per_country_db = ann_costs_per_country_df.melt(id_vars=["id_country"],
                                                              var_name="parameter",
                                                              value_name="value")
    ann_costs_per_country_db.insert(len(ann_costs_per_country_db.columns), "year",
scenario_year)

    buffer = StringIO()
```

```
    ann_costs_per_country_db.to_csv(buffer, index=False, header=False)
    buffer.seek(0)
    (cur, conn) = input.establish_connection_with_db()

    try:
        cur.execute("TRUNCATE u_mmartinezperez.t_h2_model_ann_costs_per_country;")
        cur.copy_expert("COPY u_mmartinezperez.t_h2_model_ann_costs_per_country FROM
STDIN CSV", buffer)
        # cur.copy_from(buffer, table_ann_costs_per_country, sep=",")
    except (Exception, psycopg2.DatabaseError) as error:
        print("Error_f: %s" % error)
        print("Execute Queries failed.")
        conn.rollback()
        cur.close()
    conn.commit()
    cur.close()


def save_results_as_feather(results_df, h2_scenario_id):
    feather_results_path = os.getcwd() + "\\results_id_scenario_" + str(h2_scenario_id)
    feather_results_file_path = feather_results_path + "\\summary.feather"
    if not os.path.exists(feather_results_path):
        os.mkdir(feather_results_path)
    results_df.reset_index(drop=True, inplace=True)
    results_df.to_feather(feather_results_file_path)


def load_results_from_feather(h2_scenario_id):
    feather_results_path = os.getcwd() + "\\results_id_scenario_" + str(h2_scenario_id)
    dir_list = os.listdir(feather_results_path)
    if len(dir_list) == 0:
        sys.exit("There is no data in" + feather_results_path + "directory")
    results = pd.read_feather(feather_results_path + "\\summary.feather")

    return results


def save_results_as_parquet(results_df, h2_scenario_id):
    parquet_results_path = os.getcwd() + "\\results_id_scenario_" + str(h2_scenario_id)
    parquet_results_file_path = parquet_results_path + "\\summary.parquet"
    if not os.path.exists(parquet_results_path):
        os.mkdir(parquet_results_path)
    results_df.reset_index(drop=True, inplace=True)
    results_df.to_parquet(parquet_results_file_path)


def load_results_from_parquet(h2_scenario_id):
    parquet_results_path = os.getcwd() + "\\results_id_scenario_" + str(h2_scenario_id)
    dir_list = os.listdir(parquet_results_path)
    if len(dir_list) == 0:
        sys.exit("There is no data in" + parquet_results_path + "directory")
    results = pd.read_parquet(parquet_results_path + "\\summary.parquet")

    return results


def save_network_df_to_parquets(network, h2_scenario_id, id_cell):
```

```
    # Here only the most interesting df will be saved
    parquets_path = os.getcwd() + "\\parquet_results_scenario_" + str(h2_scenario_id)
    cell_parquets_path = parquets_path + "\\" + id_cell
    if os.path.exists(cell_parquets_path):
        shutil.rmtree(cell_parquets_path)
    os.makedirs(cell_parquets_path)

    network.buses.to_parquet(cell_parquets_path + "\\buses.parquet")
    network.buses_t.marginal_price.to_parquet(cell_parquets_path + "\\buses-
marginal_price.parquet")
    network.buses_t.p.to_parquet(cell_parquets_path + "\\buses-p.parquet")
    network.carriers.to_parquet(cell_parquets_path + "\\carriers.parquet")
    network.generators.to_parquet(cell_parquets_path + "\\generators.parquet")
    network.generators_t.p.to_parquet(cell_parquets_path + "\\generators-p.parquet")
    network.generators_t.p_max_pu.to_parquet(cell_parquets_path + "\\generators-
p_max_pu.parquet")
    network.links.to_parquet(cell_parquets_path + "\\links.parquet")
    network.links_t.mu_lower.to_parquet(cell_parquets_path + "\\links-mu_lower.parquet")
    network.links_t.mu_upper.to_parquet(cell_parquets_path + "\\links-mu_upper.parquet")
    network.links_t.p0.to_parquet(cell_parquets_path + "\\links-p0.parquet")
    network.links_t.p1.to_parquet(cell_parquets_path + "\\links-p1.parquet")
    network.loads.to_parquet(cell_parquets_path + "\\loads.parquet")
    network.storage_units.to_parquet(cell_parquets_path + "\\storage_units.parquet")
    network.storage_units_t.state_of_charge.to_parquet(cell_parquets_path +
                                                        "\\storage_units-
state_of_charge.parquet")
    network.stores.to_parquet(cell_parquets_path + "\\stores.parquet")
    network.stores_t.e.to_parquet(cell_parquets_path + "\\stores-e.parquet")
    network.stores_t.p.to_parquet(cell_parquets_path + "\\stores-p.parquet")


def load_single_network_from_parquets(parquet_path, id_cell):
    dict_network_cell = {}
    for file in os.listdir(parquet_path + "\\" + id_cell):
        key_name = os.path.splitext(file)[0]
        file_path = os.path.join(parquet_path + "\\" + id_cell, file)
        dict_network_cell[key_name] = pd.read_parquet(file_path)

    return dict_network_cell


def load_all_scenario_networks_from_parquets(h2_scenario_id):
    dict_scenario_networks_cells = {}
    parquet_scenario_folder = os.getcwd() + "\\parquet_results_scenario_" +
str(h2_scenario_id)
    for id_cell in os.listdir(parquet_scenario_folder):
        dict_scenario_networks_cells[id_cell] =
load_single_network_from_parquets(parquet_scenario_folder, id_cell)

    return dict_scenario_networks_cells
```