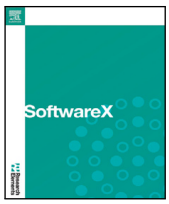
SoftwareX 27 (2024) 101844

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx



Original software publication

TulipaProfileFitting.jl: A Julia package for fitting renewable energy time series profiles

Diego A. Tejada-Arango [a,b], Abel S. Siqueira [d], Özge Özdemir [e], Germán Morales-España [a,c,*]

[a] TNO Energy & Materials Transition, Radarweg 60, Amsterdam, 1043 NT, The Netherlands
[b] Instituto de Investigación Tecnológica, Escuela Técnica Superior de Ingeniería, Universidad Pontificia Comillas, C. Alberto Aguilera, 23, Madrid, 28015, Spain
[c] Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 5, Delft, 2628 CD, The Netherlands
[d] The Netherlands eScience Center, Science Park 402, Amsterdam, 1098 XH, The Netherlands
[e] PBL, Bezuidenhoutseweg 30, Den Haag, 2594 AV, The Netherlands

ARTICLE INFO

Keywords:
Renewable energy profiles
Capacity factors
Energy modeling
Availability profiles
Full load hours
Renewable source potential

ABSTRACT

This paper introduces the TulipaProfileFitting.jl package, a tool developed in Julia to generate renewable energy profiles that fit a given capacity factor of full load hours. It addresses the limitations of naive methods in adjusting existing profiles to match improved technology efficiency, particularly in scenarios lacking detailed weather data or technology specifications. By formulating the problem mathematically, the package provides a computationally efficient solution for creating realistic renewable energy profiles based on existing data. It ensures that the adjusted profiles realistically reflect the improvements in technology efficiency, making it an essential tool for energy modelers in analyzing future energy systems.

Code metadata

Current code version	v0.3.3
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-24-00269
Permanent link to Reproducible Capsule	
Legal Code License	Apache License 2.0
Code versioning system used	git
Software code languages, tools, and services used	Julia
Compilation requirements, operating environments & dependencies	Julia
If available Link to developer documentation/manual	https://tulipaenergy.github.io/TulipaProfileFitting.jl/stable/
Support email for questions	german.morales@tno.nl

1. Motivation and significance

Energy system models, like SpineOpt [1] and PyPSA [2], are widely used to analyze different scenarios and pathways for transitioning towards CO_2-neutral energy systems. A significant part of this transition depends on weather-dependent renewable energy sources, such as wind and solar. Therefore, renewable energy profiles, which represent the available production of these resources, are crucial inputs for these models. Practitioners use raw weather data, such as wind speed and solar radiation, to determine these profiles [3]. Then, they determine the availability profile according to the technology characteristics, such as turbine type and height for wind and tilt and tracking system for solar [4]. It is important to highlight that operational capacity factors in the real world will be influenced by market dynamics, outages, and limitations in power flow physics, such as capacity and voltage constraints, see [5]. This work aims to determine a renewable resource's maximum availability profile without curtailment, i.e., the capacity factor of full load hours or renewable source potential. Therefore, the capacity factors of full load hours are the mean values (in p.u.) of these availability profiles in a year, which indicate an efficiency measure of the technology in producing energy. Renewable energy resources with higher capacity factors of full load hours are more efficient in producing more energy from the weather resources. With

* Corresponding author.
E-mail address: german.morales@tno.nl (Germán Morales-España).

https://doi.org/10.1016/j.softx.2024.101844
Received 17 May 2024; Received in revised form 11 July 2024; Accepted 30 July 2024
Available online 7 August 2024

improvements in renewable technology, capacity factors of full load hours are expected to increase by nearly 30% [4].

Regarding how to generate these availability profiles, if an energy modeler has the raw weather data and knows the technical details of the new technology, then one can create the new availability profiles using the methods developed by Koivisto et al. [3] or by Staffell and Pfenninger [4]. However, if the modeler does not have the weather data, is unaware of the specific characteristics of the new technology, or needs a fast but good approximation of the improved capacity factor of full load hours? A naive approach multiplies an existing renewable profile by a factor that scales up the data until the target capacity factor of full load hours is achieved. However, this approach is flawed since it could result in unrealistic values. For example, if your profile has values equal to 1 p.u. and you consider an improvement of 5%, you will end up with a profile value of 1.05 p.u. Even if you end up with values below 1 p.u. using the naive approach, the higher values still receive a greater increase than the lower ones, which is unrealistic since technology efficiency improvements are typically allocated in the middle values of the profiles.

To overcome this problem, Özdemir et al. [6] used an optimization model to obtain new profiles without the drawbacks of the naive approach. However, this paper presents a way to obtain the same results without formulating an optimization problem, thus much faster. Therefore, we introduce the *TulipaProfileFitting.jl* package, developed in Julia the programming language [7], designed to create new renewable energy profiles from existing ones, fitting them to a target capacity factor of full load hours. We define the problem as a mathematical equation and reformulate it to find a coefficient fitting the existing renewable profile with a target capacity factor of full load hours. This approach simplifies the problem, making it more computationally efficient and relevant for real-world applications. The package includes a robust algorithm that identifies the most appropriate values for a new profile, ensuring that the average energy output aligns with the target capacity factor of full load hours chosen by the user.

2. Software description

2.1. Mathematical background

TulipaProfileFitting.jl is a package based on the mathematical equation (1), which considers a list of m numbers representing potential renewable energy production values ranging from 0 to 1.

$$\frac{1}{m}\sum_{i=1}^{m} p_i^x = \mu \tag{1}$$

The primary challenge is finding a power x that satisfies the condition where the mean of these values, each raised to the power of x, equals a target mean $\mu \in (0,1)$.

Using (1), we can analyze the following questions to solve the problem:

Q1. Is it possible to find such a value of x?
Q2. If such x exists, how do we find it?
Q3. if not possible to find such a value of x, what to do?

Notice that both p_i and x can be zero, resulting in an undefined value of p_i^x. Some programming languages, including Julia, define 0^0 as 1. However, this definition does not make sense for the fitting profile application, as the p_i values are fixed while x is variable. As a result, we define the function σ as $\sigma : [0,1]\times[0,\infty)$ to avoid this issue.

$$\sigma(p,x) = \begin{cases} p^x, & \text{if } p > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

This allows $x \mapsto \sigma(p,x)$ to be continuous for any value $p \in [0,1]$.

Therefore, the problem can be redefined as finding x such that

$$\frac{1}{m}\sum_{i=1}^{m} \sigma(p_i, x) = \mu.$$

However, to avoid this notation, we can assume, without loss of generality, that there are r non-zero p_i, i.e., $p_1 \geq \cdots \geq p_r > 0$, and $p_{r+1} = \cdots = p_m = 0$.

This simplifies the problem of finding x such that

$$\frac{1}{m}\sum_{i=1}^{r} p_i^x = \mu.$$

Let us define $S(x) = \frac{1}{m}\sum_{i=1}^{r} p_i^x$ to help us with the notation. To visualize the problem, we are going to use four sets of possible profile values:

Set 1. A set containing one 0 and one 1
Set 2. A set containing a lot of zeros and ones (25% each)
Set 3. A set that only contains numbers from 0 to $\alpha < 0.5$.
Set 4. A set that only contains numbers from $(1-\alpha)$ to 1.

First, let us visualize the effect of p_i^x on these sets. Fig. 1 displays the original values in *lightblue*, sorted from the maximum to the minimum. The mean value of the original data is indicated in *blue*. In addition, the value for $x = 0.5$ is indicated in *pink*, and the corresponding new mean is depicted in *red*.

Furthermore, Fig. 2 shows the function S for each set.

Answer to Q1: Since $p^0 = 1$ for positive p, then $S(0) = \frac{r}{m}$. S is non-increasing since

$$S'(x) = \frac{1}{m}\sum_{i=1}^{r} p_i^x \ln p_i \leq 0.$$

And assuming that there are n values such that $p_i = 1$, then:

$$\lim_{x\to\infty} S(x) = \frac{n}{m}.$$

This means that there is a solution to the problem if $\frac{n}{m} < \mu \leq \frac{r}{m}$.

Answer to Q2: Assuming S decreasing and μ in range, we can solve the problem by looking for an interval $[a,b]$ such that $S(x)-\mu$ changes sign. This can be done by creating an increasing sequence of points $v_1, v_2, \ldots$, such that $v_1 = 0$, and $v_{i+1} > v_i$ with $\lim_{i\to\infty} v_i = \infty$. For instance, the sequence $0, 1, 2, 4, 8, \ldots$. Since S is decreasing, and μ is in range, then either $S(v_i) = \mu$ for some v_i, or $S(x)-\mu$ will change sign in some interval $[v_i, v_{i+1}]$. Given the interval, we perform a bisection search using Root.jl [8].

Answer to Q3: If $\mu > \frac{r}{m}$, then $x = 0$ will yield $S(0) = \frac{r}{m}$, which is the closest to μ. Alternatively, if $\mu \leq \frac{n}{m}$, then $S(x) \to \frac{n}{m}$ as $x \to \infty$. In this case, we select a reasonably large value for x.

Fig. 3 shows the results of each set with a target mean of $\mu = 0.65$. The plots on the left-hand side highlight the target value in *red*. Meanwhile, the plots on the right-hand side depict the interval $\frac{n}{m} < \mu \leq \frac{r}{m}$. We can observe that the function S of each set of values intersects with the target value.

2.2. Software architecture

The package was developed in the Julia programming language [7]. For instructions on installing and running Julia code and packages, please visit https://julialang.org/. The package uses the following main dependencies: Root.jl [8] and Statistics.jl. In addition, it is registered in the official registry of general Julia packages and can be installed using the following commands in the Julia REPL.

```
using Pkg
Pkg.add("TulipaProfileFitting")
```

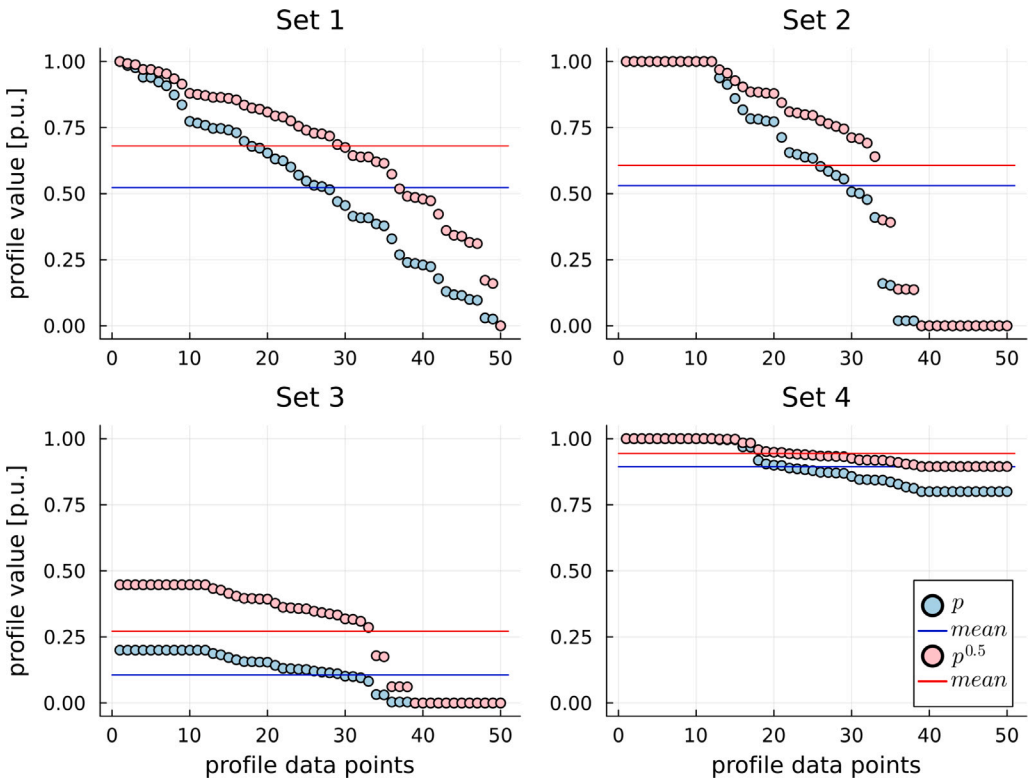


Fig. 1. Example of sets of values.

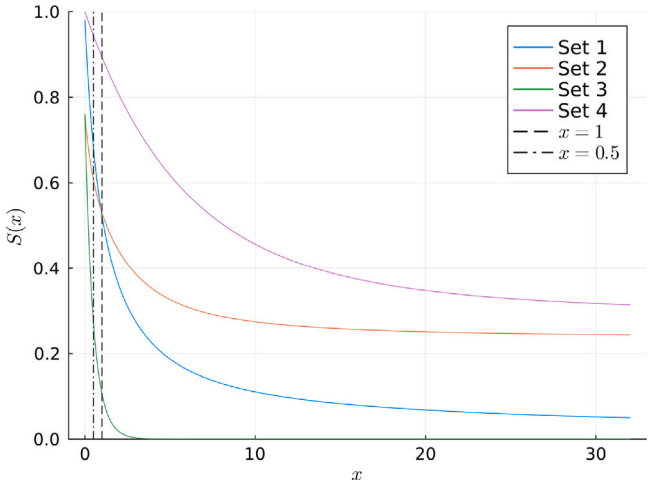


Fig. 2. Function S for each set of values.

2.3. Software functionalities

The package has the following functions or methods:

- $TulipaProfileFitting.find\_search\_interval$: Returns an interval such that $f(a) \cdot f(b) \leq 0$. It could be 0 for either endpoint, but it is not positive, ensuring that there is a root in $[a, b]$.
- $TulipaProfileFitting.find\_solution$: Finds x such that $S(x) = \mu$ if possible, where $S(x) = \frac{1}{|P|} \sum_{p \in P : p > 0} p^x$. If not possible, return either 0 or 1000, depending on what is most appropriate.
- $TulipaProfileFitting.validate\_profile$: Validates if the values of profile P are within 0 and 1. If not, it throws an error message and stops the calculation.

Table 1
Sample of available tools to generate base renewable profiles.

Name	Reference	Website
Renewables Ninja	[4]	https://www.renewables.ninja/
CorRes	[3]	https://corres.windenergy.dtu.dk/
Atlite	[9]	https://atlite.readthedocs.io/en/latest/

3. Illustrative examples

TulipaProfileFitting.jl is a package designed to fit power availability production profiles of renewable sources, such as wind and solar, from an existing base profile. These base profiles can be generated using different options and tools. Table 1 lists three of them. In this example, we generated a time series for a wind power plant's power production using the Renewables Ninja tool. The code below shows the link to the generated base wind availability profile and illustrates how to use the package to fit it to a new target capacity factor of full load hours.

```
# Load packages
using TulipaProfileFitting
using CSV
using Plots
using DataFrames
using HTTP

plotly()

# File location
file_url = "https://raw.githubusercontent.com/
    TulipaEnergy/TulipaProfileFitting.jl/main/docs/
    src/files/wind_power_profile.csv"

# Read file
df = DataFrame(CSV.File(HTTP.get(file_url).body,
    header=4))

# Get the profile from the dataframe
profile_values = df.electricity
```

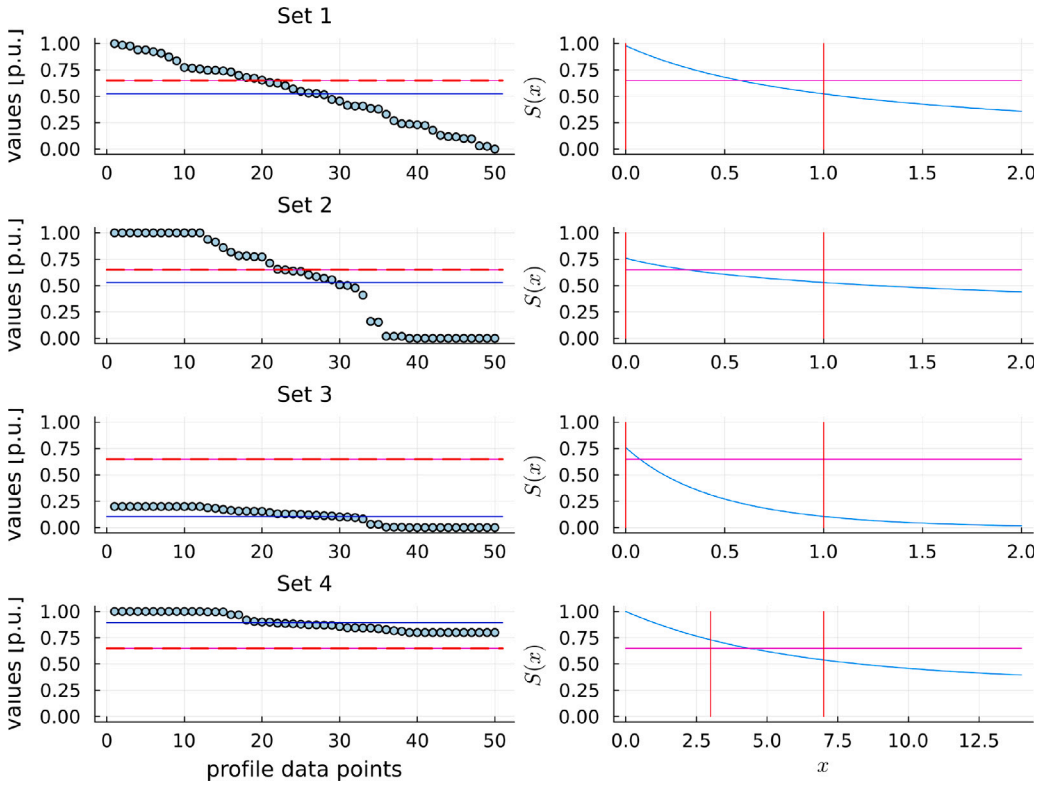


Fig. 3. Results for four sets of values with a target mean of $\mu = 0.65$.

```
# Current capacity factor of full load hours (e.g., mean value)
current_cp = round(sum(profile_values)/8760;digits=2)

# New capacity factor of full load hours as definition
target_cp = 0.6

# Obtain the coefficient that fits the values to the target.
coefficient = find_solution(profile_values, target_cp)

# Determine the fitted profile
fitted_profile = profile_values.^coefficient

# Plot chronological values
plot(profile_values, label="profile")
plot!(fitted_profile, label="fitted")

# Plot sorted values
plot(sort(profile_values,rev=true), label="profile")
plot!(sort(fitted_profile,rev=true), label="fitted")
```

The example results are shown in Figs. 4 and 5. It can be noticed that the fitted curve mainly affects the middle values within the entire range, while the values closer to 0 or 1 remain relatively unchanged. This outcome is expected since the package's main objective is to adjust the middle values, increasing or decreasing the capacity factor of full load hours. In addition, this and one extra example are available as a notebook in TulipaProfileFitting.jl/notebooks to help new users execute the package and explore the results.

Finally, Since the package uses well-known numerical methods, solution times are less than a second and can be parallelized for multiple profiles.

4. Impact

The package is designed to provide a quick and reliable method to generate new renewable energy profiles based on existing ones. It is particularly useful for complex scenarios that involve a combination of high- and low-output renewable sources. The package is a practical tool for modelers who work with energy system models and wish to analyze multiple scenarios with several availability profiles with different capacity factors of full load hours for renewable energy sources. These availability profiles are used as the upper limit for renewable energy production in energy system models. This bound allows the models to optimize production and handle any necessary curtailment of renewable energy sources. In these situations, the operational capacity factor should be lower than the capacity factor of full load hours determined using this package.

The impact of the proposed methodology is evident in two recent studies that utilized this package and a previous version of it [10]. Jimenez et al. [11] examined whether the energy-only market enables resource adequacy in a decarbonized power system, while Morales-España et al. [12] studied the impact of large-scale hydrogen electrification and the retrofitting of natural gas infrastructure on the European power system. Both studies have created new renewable scenarios for analysis using our proposal, based on fitting existing availability profiles to new target capacity factors of full load hours.

5. Conclusions

This package provides a mathematical representation and an intuitive solution to generate new scenarios of availability profiles of renewables from a base profile, accounting for changes on the capacity factor of full load hours quickly and reliably. Its rigorous mathematical foundation, combined with its practical application in energy system models, positions it as a valuable tool for researchers and practitioners in the energy modeling field. Using Julia as a programming language provides the package with the advantage of being in an open-source environment, allowing for further development and future research opportunities.

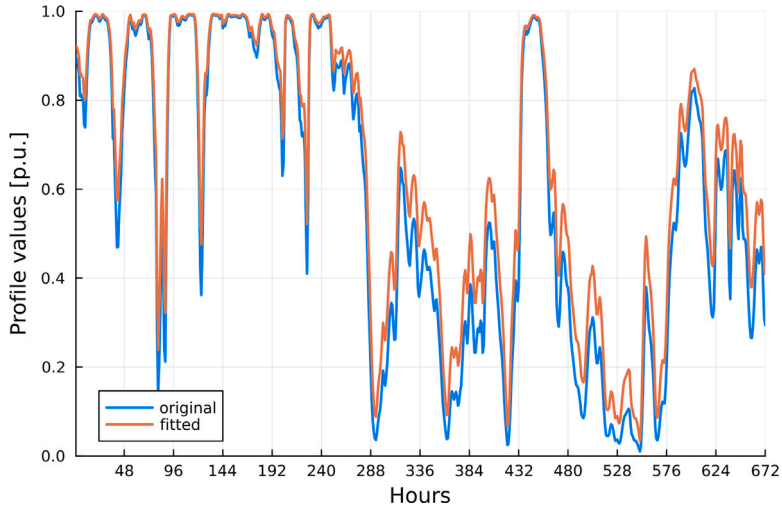


Fig. 4. Example of the hourly profile for one week.

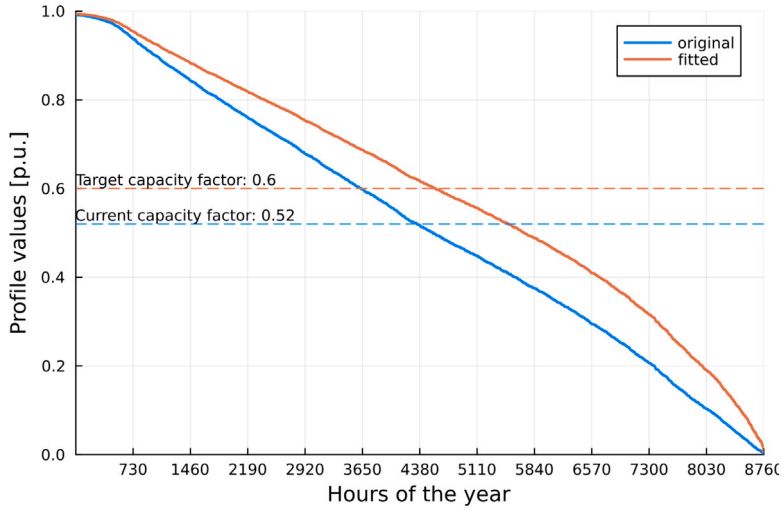


Fig. 5. Sorted values for the profile in the year.

CRediT authorship contribution statement

Diego A. Tejada-Arango: Writing – original draft, Visualization, Validation, Software, Formal analysis, Data curation. **Abel S. Siqueira:** Writing – review & editing, Validation, Software, Methodology, Conceptualization. **Özge Özdemir:** Writing – review & editing, Methodology, Conceptualization. **Germán Morales-España:** Writing – review & editing, Validation, Methodology, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Diego A. Tejada-Arango reports financial support was provided by Dutch Research Council. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data is available in the code online repository.

Acknowledgments

This research received funding from the Dutch Research Council (NWO) under grant number ESI.2019.008. It reflects only the author's views, and NWO is not liable for any use that may be made of the information contained therein. The authors thank Professor Benjamin F. Hobbs from the Department of Environmental Health and Engineering at Johns Hopkins University for his insightful suggestion to use optimization techniques to fit renewable energy profiles. This initial proposal laid our research's foundation, and we are grateful for his valuable contribution.

References

[1] Ihlemann M, Kouveliotis-Lysikatos I, Huang J, Dillon J, O'Dwyer C, Rasku T, et al. SpineOpt: A flexible open-source energy system modelling framework. Energy Strategy Rev 2022;43:100902. http://dx.doi.org/10.1016/j.esr.2022.100902, URL https://www.sciencedirect.com/science/article/pii/S2211467X22000955.

[2] Hörsch J, Hofmann F, Schlachtberger D, Brown T. PyPSA-Eur: An open optimisation model of the European transmission system. Energy Strategy Rev 2018;22:207–15. http://dx.doi.org/10.1016/j.esr.2018.08.012, URL https://www.sciencedirect.com/science/article/pii/S2211467X18300804.

[3] Koivisto M, Jónsdóttir GM, Sørensen P, Plakas K, Cutululis N. Combination of meteorological reanalysis data and stochastic simulation for modelling wind generation variability. Renew Energy 2020;159:991–9. http://dx.doi.org/10.

1016/j.renene.2020.06.033, URL https://www.sciencedirect.com/science/article/pii/S0960148120309277.

[4] Staffell I, Pfenninger S. Using bias-corrected reanalysis to simulate current and future wind power output. Energy 2016;114:1224–39. http://dx.doi.org/10.1016/j.energy.2016.08.068, URL https://www.sciencedirect.com/science/article/pii/S0360544216311811.

[5] Kazmi H, Tao Z. How good are TSO load and renewable generation forecasts: Learning curves, challenges, and the road ahead. Appl Energy 2022;323:119565. http://dx.doi.org/10.1016/j.apenergy.2022.119565, URL https://www.sciencedirect.com/science/article/pii/S0306261922008753.

[6] Özdemir Ö, Hobbs BF, van Hout M, Koutstaal PR. Capacity vs energy subsidies for promoting renewable investment: Benefits and costs for the EU power market. Energy Policy 2020;137:111166. http://dx.doi.org/10.1016/j.enpol.2019.111166, URL https://www.sciencedirect.com/science/article/pii/S0301421519307529.

[7] Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A fresh approach to numerical computing. SIAM Rev 2017;59(1):65–98. http://dx.doi.org/10.1137/141000671, URL https://epubs.siam.org/doi/10.1137/141000671.

[8] Verzani J. Roots.jl: Root finding functions for Julia. 2020, https://github.com/JuliaMath/Roots.jl.

[9] Hofmann F, Hampp J, Neumann F, Brown T, Hörsch J. Atlite: A lightweight python package for calculating renewable power potentials and time series. J Open Source Softw 2021;6(62):3294. http://dx.doi.org/10.21105/joss.03294.

[10] Morales-Espana G, Tejada-Arango DA. Nonlinear programming model to scale renewable energy sources profiles. 2023, URL https://github.com/TNO/scaling-res-profiles.

[11] Jimenez IS, Ribó-Pérez D, Cvetkovic M, Kochems J, Schimeczek C, de Vries L. Can an energy only market enable resource adequacy in a decarbonized power system? A co-simulation with two agent-based-models. Appl Energy 2024;360:122695. http://dx.doi.org/10.1016/j.apenergy.2024.122695, URL https://www.sciencedirect.com/science/article/pii/S0306261924000783.

[12] Morales-España G, Hernández-Serna R, Tejada-Arango DA, Weeda M. Impact of large-scale hydrogen electrification and retrofitting of natural gas infrastructure on the European power system. Int J Electr Power Energy Syst 2024;155:109686. http://dx.doi.org/10.1016/j.ijepes.2023.109686, URL https://www.sciencedirect.com/science/article/pii/S0142061523007433.