



Matching PV generation and energy demand: A review of available models and tools in FME SOLAR

The Norwegian PV Research Centre (FME SOLAR) is a national research centre funded by the Research Council of Norway and the centre's partners for the period 2025–2032. The centre was established to support the development of the broad Norwegian solar industry.

More information is available on the centre's website: fmesolar.no

Authors:	Igor Sartori	SINTEF
	Mari Juel	SINTEF
	Berhande D. Dimd	SINTEF
	Ingeborg T. Røe	SINTEF
	Stine Myhre	IFE
	Heine Nygard Riise	IFE
	Elin Sødahl	IFE
	Heidi S. Nygård	NMBU
	Leonardo Rydin Gorjão	NMBU
	Julien Cabrol	NVE
	Mari Øgaard	IFE/UiA
	Joao Leal	UiA

Date: 26/02/26

Executive summary

This report reviews the modelling resources available within FME SOLAR for analysing the interaction between photovoltaic (PV) generation, building energy demand, energy storage, energy-system behaviour, and grid operation under Norwegian conditions. The objective is to map the complementarity of existing tools and to identify where model interfaces already work effectively and where methodological gaps remain.

Solar-irradiation data is a foundational input to many of these tools. At high latitudes, accuracy varies across measurement-based, model-based, and satellite-based datasets. Satellite products generally perform best up to roughly 65°N but deteriorate under cloudy, snowy, or complex-terrain conditions. Local ground measurements remain valuable for site-specific assessments and benchmarking.

On the demand side, buildings are central because their load profiles are poorly aligned with PV generation in Norway, especially due to winter heating needs. Building-FLEXor provides a detailed and operationally realistic framework for modelling energy flexibility in buildings through thermal inertia, hot-water storage, stationary batteries, and EV charging. When combined with typical demand profiles from Building-PROFet, it generates flexible load profiles suitable for downstream energy-system and grid studies. Prior soft-linking studies show that building-level flexibility has limited impact on national electricity price formation, meaning buildings generally behave as price-takers. This simplifies future model coupling.

The review compares a range of PV modelling tools. PVsyst, PVMCase, and SAM provide high-detail engineering models for design, shading, losses, and techno-economics. Renewables.ninja and pvlib python offer transparent, programmatic modelling suitable for long-term and large-scale scenario generation. Together they cover both detailed system design and scalable profile generation.

Several energy-system models are used in the consortium, including EnergyModelsX (operational and investment optimisation), IFE-TIMES-Norway (long-term national planning), PyPSA-Eur (continental-scale optimisation), and JulES (market simulation with detailed hydropower modelling). Grid-level analysis is covered by pandapower for LV/MV networks and PowerFactory for advanced AC/DC grid studies.

Overall, the consortium already possesses a complete modelling chain capable of addressing PV–demand matching at multiple scales. Remaining challenges relate primarily to consistent interface design, harmonised assumptions, and coordinated workflows—particularly when linking Building-FLEXor with energy-system and grid models. With improved data pipelines and shared scenario definitions, FME SOLAR is well positioned to produce robust, reproducible analyses for Norwegian solar integration.

Table of Contents

Executive summary	3
1 Introduction	5
2 Solar irradiation source data	6
3 Energy demand models	8
3.1 Building-FLEXor – Energy flexible operation of buildings	8
4 PV generation models	21
4.1 PVsyst Photovoltaic Software	21
4.2 Renewables.ninja	23
4.3 System Advisor Model (SAM)	25
4.4 PVCase	27
4.5 Pvlib python	29
5 Energy storage models	30
5.1 VirtualFCS	30
6 Energy system models	33
6.1 EnergyModelsX	33
6.2 PyPSA-Eur economic dispatch for renewable energies	37
6.3 TIMES	41
6.4 JuLES	46
7 Grid models	48
7.1 Pandapower: Python Tool for Power System Analysis	48
7.2 DlgSILENT PowerFactory	54
8 Models linking and Gap analysis	55
8.1 Linking energy demand with energy system models	55
9 Conclusions	58
References	59

Introduction

This report surveys the models and tools available within FME SOLAR to analyse the coupling between photovoltaic (PV) generation and energy demand in buildings across scales – from component-level physics to national energy-system optimisation – and to identify where interfaces already exist and where gaps remain. Particular emphasis is placed on energy demand: its temporal patterns, its flexibility, and its role as the principal lever for matching PV output in buildings and neighbourhoods.

The report opens with a review of solar-irradiation data sources and their suitability in Norway, before moving to the demand side, documenting methods and datasets for constructing baseline load profiles and for activating demand-side flexibility in buildings. Then follows a description of several storage, system, and grid models that resolve interactions at different levels of aggregation and operational detail.

Energy demand is the connective tissue across these workflows. Temporal coincidence between PV output and loads is generally weak in Norway due to the high electricity demand for heating purposes in winter. Improving the match depends on exploiting building-level flexibility (thermal inertia, tank pre-heating, electric vehicles (EV) smart charging) and on coordinating demand with storage and market signals. In particular, EV charging in commercial building premises has a natural match with solar, since cars are parked mostly during daytime. Prior experience with soft-linking between flexible demand and energy system models shows that buildings largely remain price-takers, which simplifies coupling strategies while still capturing most of the system benefit.

Solar irradiation source data

By Stine Myhre and Heine Nygard Riise, IFE

Solar irradiation source data represents the amount of solar energy received on a surface per unit area (typically expressed as kWh/m²). It is an important input for solar resource assessment, PV system design, and performance analysis as it supports estimating the energy generation and performance of PV installations at various locations. Accurate solar irradiation data ensures reliable energy yield predictions and economic evaluations for solar projects.

There exist three main approaches for estimating solar irradiation.

Ground-based measurements are based on direct observation of the weather data, such as solar irradiation, and these measurements are acquired and generated locally through sensors. Since there are multiple factors that might impact solar irradiation, such as clouds, variation in solar zenith, aerosols, and reflection from the ground or objects, the most reliable and accurate solar irradiance data is generated locally. These types of measurements are often not available for every location, requiring other datasets to be utilized.

Model-based approaches rely on physical and statistical models rather than direct observation. Usually, reanalysis of historic weather observations is used to estimate the needed weather data. Reanalysis is used to generate time series data by running Numerical (or AI-based) Weather Prediction models on historical weather observations. Since the Numerical Weather Prediction model usually uses a grid cell volume larger than a cloud, the cloud information needs to be estimated through parametrization, which introduces some uncertainty in the estimation of surface irradiation. Examples of databases are ERA5, MERRA-2 (NASA), JRA-55 (JMA).

Satellite-based estimations use remote satellites to estimate weather data and solar irradiation. The satellite images can be used to extract information regarding cloud cover, surface reflectance, and other relevant parameters. Compared to the reanalysis approach, cloud cover in an area can be directly estimated when using satellites. However, the viewing angles for geostationary satellites are poor for latitudes over 60°N, resulting in less accurate data for higher latitudes. Additionally, snow on the ground affects the output data from the satellite since it can be mistaken for clouds. Examples of databases are CAMS, SARAH-2, SolarGIS.

For higher latitude areas such as Norway, it can be challenging to choose the right datasets. A benchmark study has been conducted to estimate solar irradiation data at higher latitudes [1]. The study has compared and benchmarked the data from six widely used solar irradiation

estimation databases against ground measurements in Norway. The compared databases are CAMS, PVGIS SARA2, and SolarGIS for satellite-based and PVGIS ERA5, Meteonorm, and NASA POWER for reanalysis. The main takeaway from the study is that satellite-based databases are most accurate for high latitude locations up to $\sim 65^\circ\text{N}$, while above 65°N , satellite estimates are nonexistent. However, the accuracy depends on site conditions, where the errors are larger for higher latitude locations, more cloudy and snowy conditions result in less accuracy, and horizon shading adds bias to the estimates.

Additionally, MET Norway has made an open dataset of hourly global solar irradiation measurements for Norway. Here, 106 stations have been taken into account when gathering data and performing quality control of the datasets. The process and methodology can be found in the report [2]. Additionally, the dataset with API can be found here: <https://thredds.met.no/thredds/catalog/sunpoint/ML-Optimized-Maps/catalog.html>

Energy demand models

Building-FLEXor – Energy flexible operation of buildings

By Igor Sartori, SINTEF Community

This chapter gives an overview of the model structure, its input and output, and a brief account of its applications, focusing on the anticipated main application within the FME Solar, which is the match between PV generation and flexible energy demand by EV charging in connection with commercial buildings. A more extensive and detailed description of the Building-FLEXor model is given in [3] while examples of its application in combination with energy system modelling are found in [4], [5] and [6].

Model structure

Energy flexibility refers to the ability of a building to activate its on-site flexibility sources (building's thermal mass, heat storage tanks, batteries, EV charging) while safeguarding user needs and comfort. Building-FLEXor is an optimization tool for flexible energy use, storage and generation in buildings, that generates optimal load profiles in response to grid signals, such as dynamic energy prices, thus -facilitating energy planning and optimal system dimensioning and control. It comprises interconnected component models (building envelope, hot water tank, heat sources, EV charging, PV systems and electric batteries), functional models (energy costs, fixed energy demand), and a model for typical demand profiles. All the components are designed to be self-standing, meaning the optimization process will include only the desired components. It is implemented in Python and uses the Pyomo software package for formulating, solving and analyzing optimization models. The tool is available in a backend solution, run on a SINTEF server, while their frontend is accessible in two ways: via a Web App, and via API (Application Programming Interface) for more advanced users and for use by other software, see next section §0.

Figure 1 shows a schematic of the interconnections between the modules that compose Building-FLEXor. The energy flow (represented by the thick blue lines) goes from left to right, while the calculation flow goes the opposite direction, from right to left. The calculations start with defining the Typical demand profiles that must be satisfied, obtained from PROFet [7], and continue through a series of flexibility options until obtaining the Flexible load profiles on the energy carriers that supply the building.

The first flexibility option in the schematic is called "Comfort flexibility" because it is achieved by allowing the indoor temperature to fluctuate within a predefined comfort band. In this way the thermal comfort remains satisfied while the energy use deviates from the typical demand (that is the meaning of the red cross over the icon representing the Space Heating demand). This is activated by means of charging and discharging thermal energy in the building's

envelope and exploiting its slow thermal inertia. Thus, by definition, this flexibility source is available in every building.

The second flexibility option is called "Storage flexibility" because it is achieved by controlling the charge and discharge of some physical storage component, whether thermal or electric storage. One thermal storage component, the DHW tank¹, is always present, while two electric storage components are possible: Electric Vehicles (EV) and stationary Batteries. The DHW tank can be preheated, so that its energy use can follow a different profile than that of DHW demand, which is always met without alteration. The same can be said for the electric battery, while the EV charging can be modulated but only within some constraints that safeguard the transport needs of the users.

A further flexibility option can be provided by fuel-switching, achieved by controlling the operation of the SH and DHW preparation components available in the system. Within the options currently available in the Building-FLEXor Web app, this type of flexibility can only be activated when using the *Other* type of heat source, which is the only one that uses fuel (i.e.: neither electricity or district heating). For example, by switching to fuel consumption instead of electricity, Building-FLEXor can reduce the peak loads and the subsequent fees related to the peak power demand.

¹ A heating system often has a buffer, represented by the tank inside the Heat source box, which allows the base unit to work at its nominal capacity also at part load. However, its effect is short term and therefore already included in the hourly profile (from PROFet).

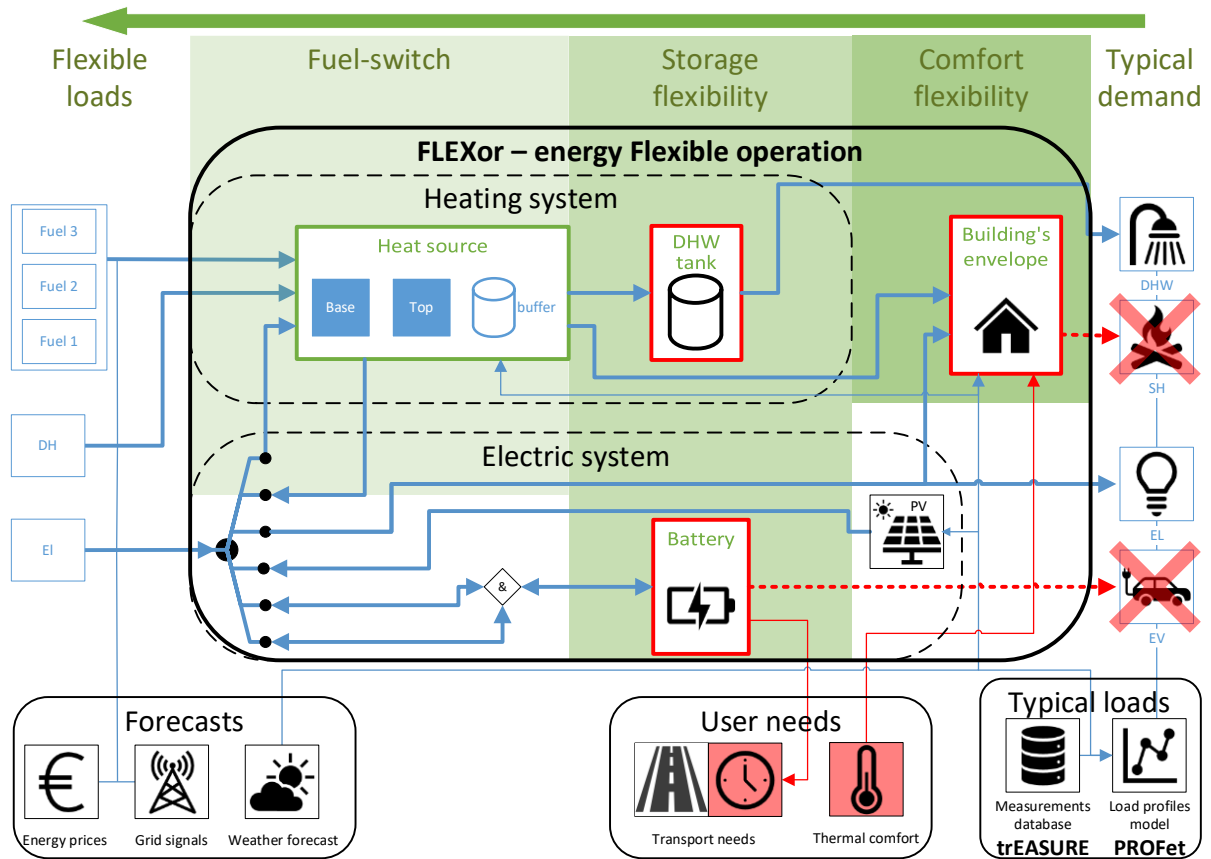


Figure 1. Schematic of the Building-FLEXor model showing the interconnections between the modules. Thick blue lines represent energy flows, thin ones represent information flows. Red boxes highlight the flexibility sources while red lines represent the possibility to deviate from typical profiles as long as user needs are satisfied.

It should be noted that all forms of flexibility come with associated energy losses.² For the battery this is because of inherent charge-discharge inefficiencies (the cycle losses). For the envelope and the DHW tank this is because only upward thermal flexibility is considered. This means that the temperature in the building and/or the tank is only allowed to become higher, never lower, than it is in the baseline. It is possible to preheat a building or tank, so that at a later time the energy use will be lower than in the baseline, but not the internal temperature in the module. This guarantees that the flexibility activation never comes to the detriment of user comfort. On the other hand, this causes additional heat losses and so, ultimately, a higher energy use.

Objective functions

The following objective function are possible:

- **Baseline:** is a subcase of the Energy import minimization function where the flexibility options are disabled. It is used for going from energy demand per service (as given by

² With the exception of EV charging, since it is only a modulation of a unidirectional flow, the EV is only charged, never discharged.

PROFet) to energy use per carrier, including all system losses. It also allows to calculate the energy costs per carrier under typical operation of a building.

- *Energy import minimisation*: Depending on the system components available, this objective can give the same results as the Baseline operation. However, the presence of heat pumps and/or PV (with or without battery) would enable the system to deviate from the Baseline operation.
- *Cost minimisation*: minimises the total cost for the entire calculation period. The costs are given as input and comprise Fixed tariffs, Energy tariff (*energiledd*) and Peak power tariff (*effektledd*), where relevant, for all carriers: electricity, district heating and fuels.
- *Flat profile*: pursues a flattening of the load profile, as much as possible while limiting the associated energy losses (max 5% compared to the baseline). Thus, at the cost of some energy loss, it smoothens "high peaks" and avoids "deep valleys" and other sudden changes in the energy demand. These features might be desirable at an aggregated scale, for smooth operation of a grid and/or energy supply system. However, it should be noted that the flattening refers to the own energy use of each single building, without regard to what happens in other buildings or in the rest of the grid in general. In this sense it can be said that the flat profile is a building-centric optimization. The hypothesis behind the flat profile optimization is that this may be a good heuristic because peak shaving and valley filling are typically desired outcomes of energy flexible demand.

Building-FLEXor modules are modelled in state space form and are Linear time invariant (LTI) models. The following modules are briefly described here, being of particular interest for this report.

PV system

Two models for PV generation have been implemented, giving users the option to supply pre-calculated time-series (or profiles) of electricity generation, or to let the model calculate the time-series. The latter option uses the python library Global Solar Energy Estimator (GSEE) [8], while the former option allows linking Building-FLEXor with any other model that calculates generation from a PV system. Further details on the PV system models can be found in [9].

Electric vehicle charging

The EV models are similar to that of a battery in that EVs can be charged and discharged. However, the focus of the model presented here is on the charging demand, not stored energy. This is because the available data does not provide information about the total energy stored in the battery of the EV, nor of its State-of-Charge (SOC): it only shows how much energy needs to be charged. Stationary losses are not included. Further details on the EV model can be found in [10], and the underlying datasets are taken from measurement campaigns over prolonged

periods that collected data from tens of thousands of charging sessions from several locations in Norway [11] [12].

Figure 2 shows the typical charging profiles and average connected capacity per EV unit. In the case of Residential buildings (House and Apartments) the EV unit is a single user (or car), while for Commercial buildings the EV unit is a single charging point. The charging profile is the average of the measurements, per EV unit, and is divided into inflexible, for the charging events that did not have at least one idle hour during the time the car was connected, and flexible, for all the others. The connected capacity indicates how much charging capacity is available, per EV unit, in average for every hour. The difference between the average connected capacity and the average charging profile expresses how much flexibility the EV unit has to offer. Therefore, in Residential buildings the flexibility potential is highest at night while in Commercial buildings it is highest in daytime.

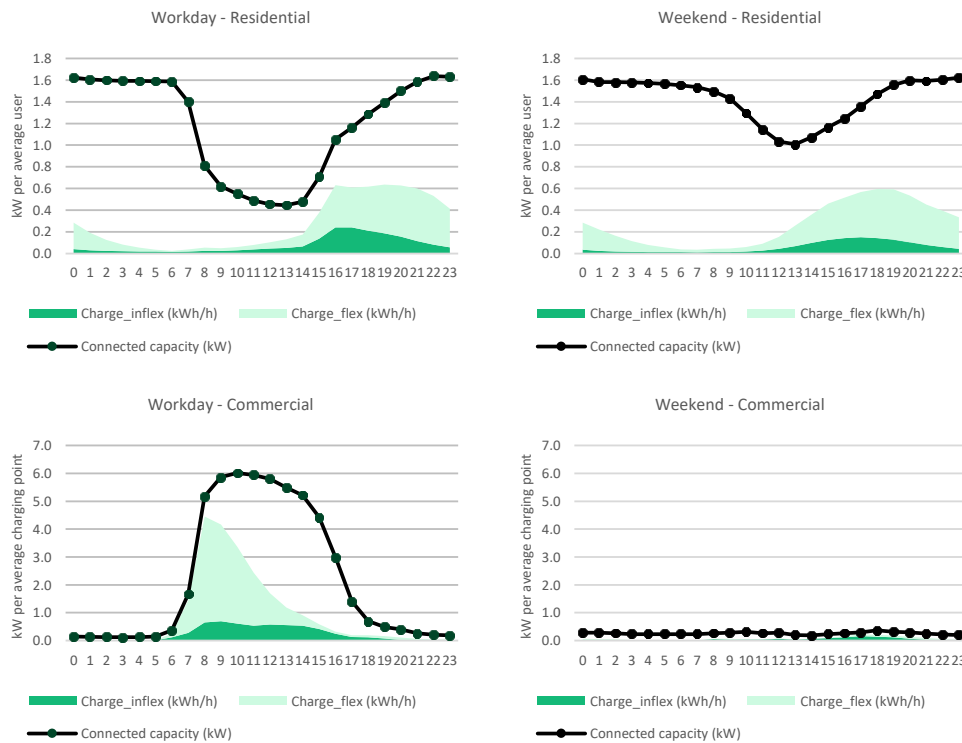


Figure 2. Electric Vehicle (EV) typical charging profiles and connected capacity in Residential (top) and Commercial (bottom) buildings; for average workdays (left) and weekend days (right).

Input, output and interface

The starting point is given by typical (non-flexible) energy demand load profiles taken from Building-PROFet (another tool, also available via API), based on a statistical analysis of hourly measurements from several buildings classified in different categories. The flexibility sources are modelled as internal variables (the model's states) such as indoor temperature, tank temperature, battery state of charge, and are subject to boundary conditions and constraints

that represent user comfort and user needs, such as a comfort band for indoor temperature, a lower bound for the hot water tank's temperature, and the charging of electric vehicles within the connection time and capacity.

The Building-FLEXor calculation engine is available online. The calculation engine, or backend, resides on a SINTEF's cloud server and can be accessed through a frontend via two entry points (or interfaces). Figure 3 shows a schematic representation of the access points and internal communications when using the calculation engines. The Building-FLEXor and Building-PROFet Web applications can be accessed through the following links:

- Building-PROFet Web app:
<https://profet.jollymeadow-8539f348.norwayeast.azurecontainerapps.io/>
- Building-FLEXor Web app:
<https://flexor.jollymeadow-8539f348.norwayeast.azurecontainerapps.io/>
- API for both:
<https://flexibilitysuite.byggforsk.no/index.html>

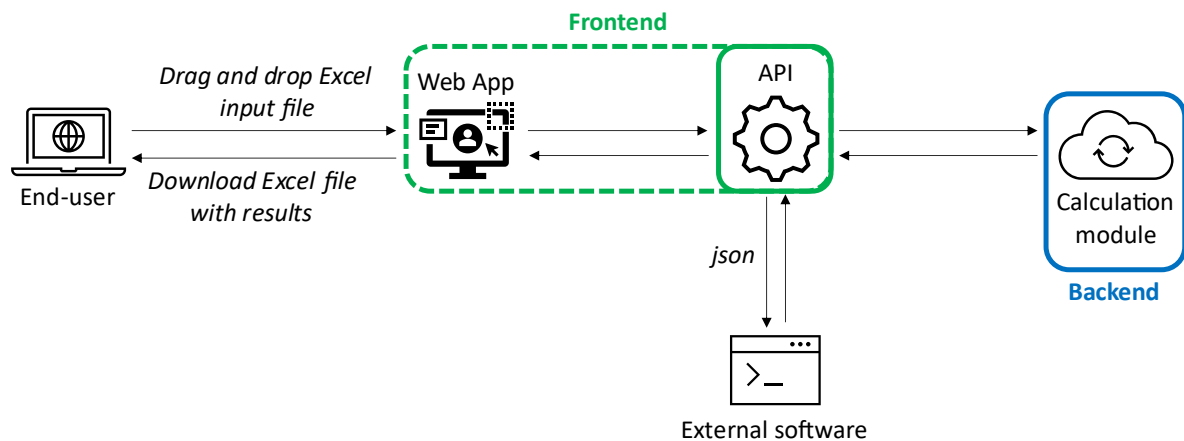


Figure 3. Access points and internal communications between end users and/or external software with the Building-FLEXor calculation engine.

The first entry point is a Web App that allows to perform energy flexibility optimizations for one building archetype and with the *Baseline*, *Energy import minimization* or *Cost minimization* objectives. Due to restrictions on the cloud server hosting the Web App interface, currently the optimisation only runs for one month at a time. The end-user can download a template excel file, in which one may select the building archetype, activate or deactivate space heating and other sources of flexibility, and modify some of the input parameters to the model. Upon making changes to these characteristics, the Excel file automatically adapts any other parameters as needed. This entry point has been created to allow a wider audience to use the tool, given the extended presence and large user-base of Excel. Upon loading the Excel file to the Web application (drag-and-drop), it is sent to an API which interacts with the calculation engine.

The second entry point is directly through the API and is intended for expert users or use by other software, using json files for input and output. In this case it is possible to run the optimisation for an entire year, as well as defining buildings with a higher degree of freedom, beyond the predefined archetypes.

When using the Excel file, some inputs are required (in dark yellow cells), some inputs may be changed (in blue cells), and defaults values (in green cells) are given for the inputs that may be changed, see Figure 4. The inputs that are required from the end-user are as follows:

- Building type (House, Apartment, Commercial) and efficiency (Regular, Efficient, Very efficient).
- Flexible space heating (Yes/No).
- Month to be calculated
- Objective of the optimization (Baseline, Energy import minimization or Cost minimization)
- Availability of PV (Yes/No).
- Availability of Battery (Yes/No). Note: if PV is not available, the battery will not be included in the model regardless of user input.
- Availability of EV (Yes/No).
- Flexible charging of EV (Yes/No).
- Type of base heating component (Electric heater, ground-source heat pump, air-source heat pump, air-to-air heat pump, district heating, other).
- Availability of Firewood heater (Yes/No). Note: only available for House and Apartment.
- Flexible charging of domestic hot water tank (Yes/No).

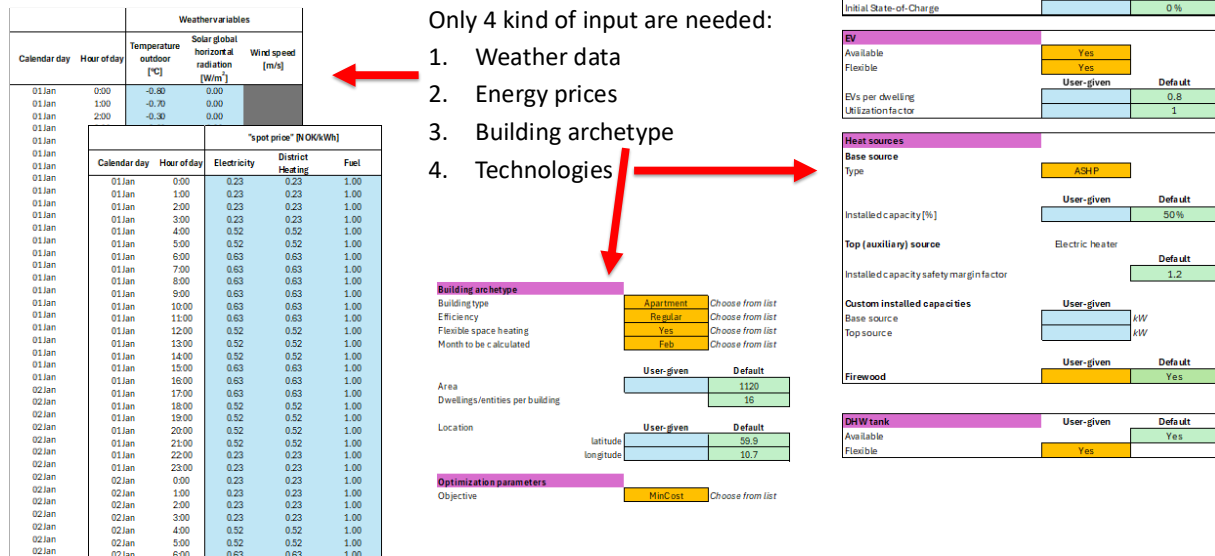


Figure 4. Screenshots of the Web App input excel file.

The main characteristics of the building archetypes available in the Excel file, and of interest here are shown in

Table 1 and Table 2. Some input parameters may be changed by the user, such as building area, energy costs, and presence and characteristics of some of the energy components. If the user does not give any values for these parameters, the default values are used. These default values can also be found in. The user should be aware that some input parameters are interdependent; therefore, changing some parameters may cause other parameters to be automatically updated. All the default values that required pre-calculations of the building operation were obtained using the NS3031:2014 standard climate data. Expert users will have access to an extensive list of parameters that can be adjusted: their description and interrelations are beyond the scope of this report.

Table 1. Geometric characteristics of the building archetypes (Source: [3]).

	House	Apartment	Commercial
Floor area (BRA) per building [m ²]	150	1120	3600
Floor area (BRA) per dwelling [m ²]	120	70	0
Dwellings per building [#]	1.25	16	0
Nr. floors [#]	2.14	4.16	1.63
Ceiling height [m]	2.4	2.4	2.4
BRA/BTA* ratio [#]	0.85	1.0	1.0
BTA [m ²]	176.5	1120	3600
Footprint (or roof) area [m ²]	82.5	269.2	2201.3
Ratio footprint/BRA	0.55	0.24	0.61
Area Windows [% BRA]	0.15	0.15	0.15
g-value	0.5	0.5	0.5

*BRA = useful floor area (*bruksareal*); BTA = gross floor area (*bruttoareal*)

Table 2 – Parameters for the PV systems, Batteries, and Electric Vehicles (Source: [3]).

	House	Apartment	Commercial
PV system			
Roof area for PV [%]	50%	50%	50%
PV system overall efficiency [%]	20%	20%	20%
PV capacity per m2 installation [Wp/m ²]	200	200	200
PV capacity per BRA [Wp/m ²]	55.00	24.03	61.15
PV capacity per building [kWp]	8.25	26.92	220.13
Tilt angle [°]	30	10	10
Orientation distribution			
South	50%	0%	0%
East	25%	50%	50%
West	25%	50%	50%

Battery			
Batt. capacity per PV installed [kWh/kWp]	0.50	0.50	0.50
Usable battery capacity [%]	80%	80%	80%
Batt. capacity per building [kWh]	4.12	13.46	110.07
Usable batt. capacity per building [kWh]	3.30	10.77	88.05
Initial State-Of-Charge (SOC) [%]	0%	0%	0%
Dis/Charge rate per hour, relative to cap. [%]	80%	40%	40%
Dis/Charge rate per hour [kWh/h]	3.30	5.38	44.03
Efficiency charging [%]	95%	95%	95%
Efficiency discharging [%]	95%	95%	95%
Electric Vehicles			
Annual energy use per EV unit [kWh/y]	2 342	2 342	5 483
Annual energy use per m2 [kWh/m ² y]	15.6	26.8	7.6
Annual energy use per building [kWh/y]	2 342	29 978	27 414
Nr. of EVs per household	0.80	0.80	-
Nr. of EVs per building	1.0	12.8	-
Nr. of charging points per building	-	-	5.0
Utilization factor	1.00	1.00	1.00
Nr. of EVs per 1000 m ²	6.67	11.42	-
Nr. of Charging points per 1000 m ²	-	-	1.39

Upon successful running, the output of the Web App consists of:

- Timeseries of energy carriers (electricity, district heating) *baseline* vs. *flexible* load;
- KPIs related to cost, energy use and peak load, see Figure 5;
- .json files for running the API with the baseline and flexible cases.

	Base	Flex	Flex/Bas e [%]
Cost [kr]			
Total	28 372	25 118	-11 %
Electricity	24 290	21 036	-13 %
Heat	0	0	
Fuel (incl. firewood)	4 082	4 082	0 %
Energy [kWh]			
Electricity use	19 961	20 202	1 %
Electricity export	0	0	
Heat use	0	0	
Fuel use	4 082	4 082	0 %
Peak load [kW]			
Peak load	60	39	-35 %
Electricity	Timestamp	2018-02-23 07:00	2018-02-01 00:00
	Peak export	0	0
	Timestamp	No export	No export
DH	Peak load	0	0
	Timestamp	Not used	Not used
Fuel (incl. firewood)	Peak load	23	23
	Timestamp	2018-02-18 08:00	2018-02-18 08:00

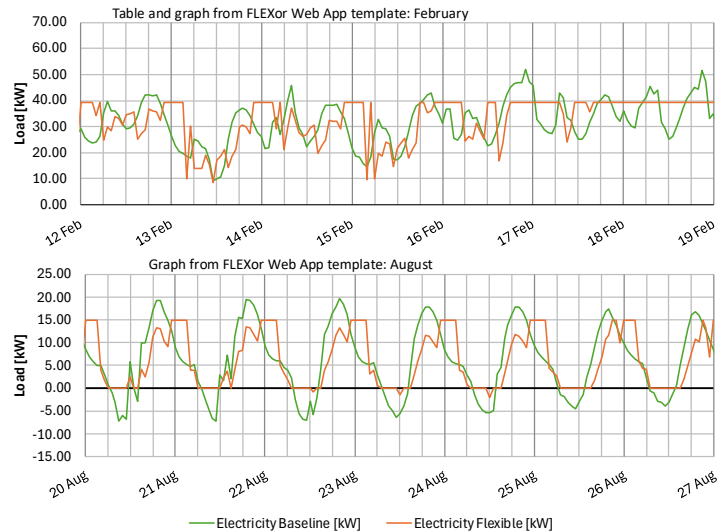


Figure 5. Screenshots of the Web App output excel file.

Applications and results

Building-FLEXor has been used in combination with other models, especially energy system models, and the results from such applications are presented in Chap.7. Here comes the description of some results that are intrinsic to Building-FLEXor, i.e. results at the level of single buildings, and focused on the PV and Battery system.

In the [Flexbuild](#) project Building-FLEXor was used in combination with an investment cost optimisation, from the end user perspective, on the energy system of a single building. This resulted in a global cost optimisation, investment and operation, over a 30-years lifecycle. The results here presented are taken from [4]. Table 3 shows the main properties of the solar PV systems for all building types. In the thousands simulation needed to model the entire building stock, those properties result with different values, depending on the climatic area, the efficiency level, the chosen heating technology and so on. Table 3 reports both the minimum and the maximum values observed in the results of all simulations.

All building types in 2050 install as much solar PV as allowed, i.e., 50% of their roof area (see Table 2), except Houses for which the result varies from zero to the maximum, totalling 83% of the potential, meaning that in average every House has installed ca. 7 kWp in 2050.

More interesting are the results for the self-consumption and the peak export / peak import ratio. Self-consumption means the amount of electricity generated by the solar PV system that is "immediately" consumed in the same building, i.e., within the same hour since the simulation have hourly resolution. The calculation is done with hourly resolution while the share refers to the cumulative value over the year. The peak export / peak import ratio compares the highest value of electricity exported from the building to the grid (in summer) over the maximum value of electricity imported from the grid (in winter). A value higher than one may be problematic

because it indicates that for that building archetype the grid connection should be dimensioned based on the summer export rather than the winter import.

Regarding self-consumption we see that for all building types except Houses the values are rather high. Partly this is due to the geometry at hand: often these are buildings with large volumes and relatively little roof area (multistorey buildings), so that the installed solar PV capacity is relatively small compared to the high building's energy demand. Thus, it is easy to self-consume most of it. However, the maximum value, hitting 100% in most cases, is usually achieved in the Flexible simulations. In Houses there is large variability. The building's geometry is rather the opposite, offering a high surface to volume proportion, so that a relatively large solar PV capacity is available compared to the small building's energy demand. In addition, the actual installed capacity ranges from min to max in the various archetypes. Consequently, also the self-consumption varies from 0 to 100%, but here too the highest values of self-consumption are achieved in the Flexible cases.

Table 3. Results for the main properties of solar PV systems in the various building types. Minimum and Maximum results for a 2050 scenario (Source: [4]).

		max PV allowed [kWp]	PV installed [kWp]		Self-consumption [%]		Peak export / Peak import [-]	
Building type			Min	Max	Min	Max	Min	Max
House		9	0	9	0 %	100 %	0.0	1.5
Apartment		32	32	32	96 %	100 %	0.0	0.4
Commercial	Office	63	63	63	97 %	100 %	0.0	0.3
	Shop	61	61	61	96 %	100 %	0.0	0.5
	Hotel	21	21	21	82 %	100 %	0.2	1.4
	Culture & Sport	35	35	35	81 %	100 %	0.1	1.4
	Kindergarten	56	56	56	75 %	93 %	0.5	1.4
	School	56	53	56	65 %	95 %	0.4	1.9
	University	116	116	116	100 %	100 %	0.0	0.1
	Hospital	167	167	167	100 %	100 %	0.0	0.0
	Nursing home	61	61	61	97 %	100 %	0.0	0.6

House is one of the building archetypes where the peak export / peak import ratio can become bigger than one. The other typologies are marked with bold values in Table 3; most significantly Schools reach a value of 1.9 because of their low energy demand during summer. For all such

buildings the grid connection should then be dimensioned based on the summer export rather than the winter import. Whether this poses a challenge to the local grid depends on how much this effect is dampened down at the local aggregation level, e.g., at the level of a feeder, due to the coincidence factor between different buildings. For an area with mixed typologies of buildings one may expect no problem would arise since most other buildings show low enough ratios that it means, very likely, they would be able to absorb the extra generation from the nearby school, so that the reverse power flow would not propagate in the grid.

Different is the case for a residential neighbourhood dominated by small houses, such as single- and bi-family houses and row-houses. In such an area, where virtually all buildings are houses connected to the same feeder and/or low voltage transformer, the entire neighbourhood is well approximated by the House archetype and may thus incur in either reverse power flow or curtailment. A dedicated analysis modelling the distribution system would be needed to investigate further on these issues. Nevertheless, the results presented here tells us that while the cost-optimal level of solar PV installation in a house is, in average, less than 50% of its roof area, the desirable level for a residential neighbourhood may be even lower, lest risking incurring in problems for the local grid.

PV generation models

PVsyst Photovoltaic Software

By Mari Juel and Berhane D. Dimd, SINTEF Industry

Model structure

PVsyst is a simulation environment for PV system design and performance analysis [13]. Its model is based on a modular architecture and allows simulation of various PV system configurations, including grid-connected, stand-alone, and pumping systems. It integrates physical, electrical, and environmental processes, from solar irradiance to AC energy output, using sub-models. Each module is parameterized using data from manufacturer-validated component databases. It also incorporates meteorological models, accounting for irradiation, temperature, shading, and system losses. Degradation models are available to evaluate long-term performance and reliability. The modeling process in PVsyst is summarized in Figure 6.

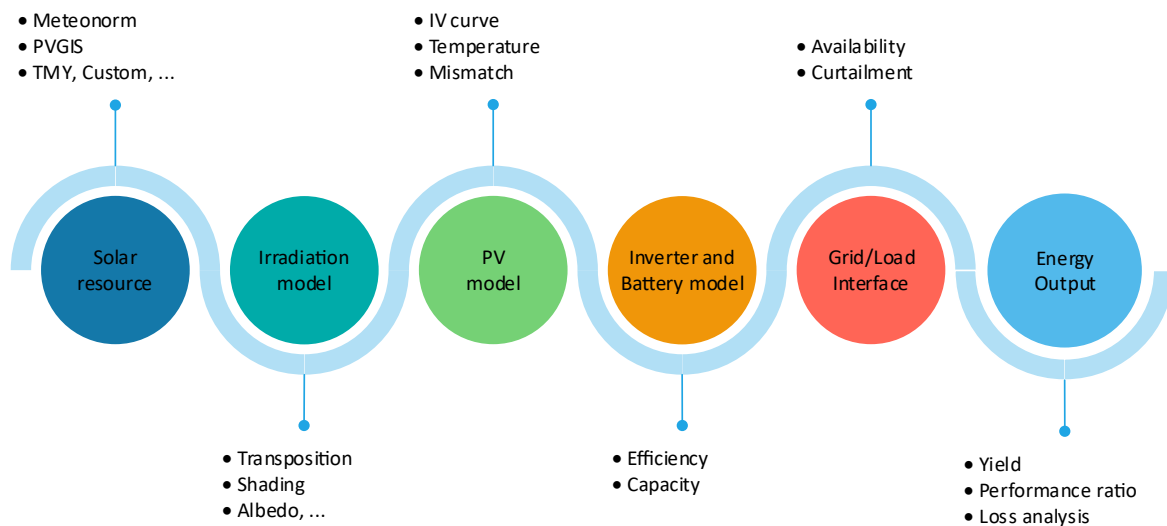


Figure 6. PVsyst modeling flow.

Input, output and interface

PVsyst functions through a workflow that transforms user-defined inputs into detailed performance and financial outputs. Users specify site conditions (meteorological data from sources such as Meteonorm, NASA, or PVGIS), system configuration (tilt and azimuth, PV array, inverter), and component data from built-in or custom databases. Economic parameters such as investment cost, O&M, and project lifetime can also be included. The simulation then generates hourly performance profiles, annual specific yields, performance ratios and a detailed loss diagram. Financial indicators such as net present cost, LCOE and payback period are also calculated. An overview of this structure is shown in Figure 7.



Figure 7. Input–output summary for PVsyst.

PVsyst has a user-friendly graphical interface for system design, shading analysis, and result visualization. Output can be exported as csv for use in other grid or economic modeling tools.

Applications and results

PVsyst is applied in the design, optimization, and evaluation of PV systems. It allows users to assess system performance under different design configurations, climatic conditions, and operational scenarios. It supports feasibility studies with reliable estimates of energy yield, system losses, and economic performance. In the context of FME WP3, PVsyst is particularly important for generating hourly production profiles. These outputs can be exported and integrated into grid simulation tools for low-voltage network studies, to assess PV integration impacts such as voltage variation and hosting capacity.

Renewables.ninja

By Heidi S. Nygård, NMBU

Model structure

Renewables.ninja is an online simulation platform for estimating hourly power output from solar PV and wind systems worldwide [14] [8] [15]. The tool uses validated physical models combined with historical weather datasets, including NASA MERRA [16] and CM-SAF SARAH [17] [18], to generate long-term, bias-corrected capacity factor profiles. For PV systems, it employs the GSEE Python package to model irradiance, temperature effects, and system configuration. For wind, it uses the vwf R package to simulate turbine output. Both models have been calibrated against real-world generation data to ensure accuracy and reliability.

Input, output and interface

Users can access Renewables.ninja through an interactive web-based map interface (Figure 8) or via an API for automated workflows.

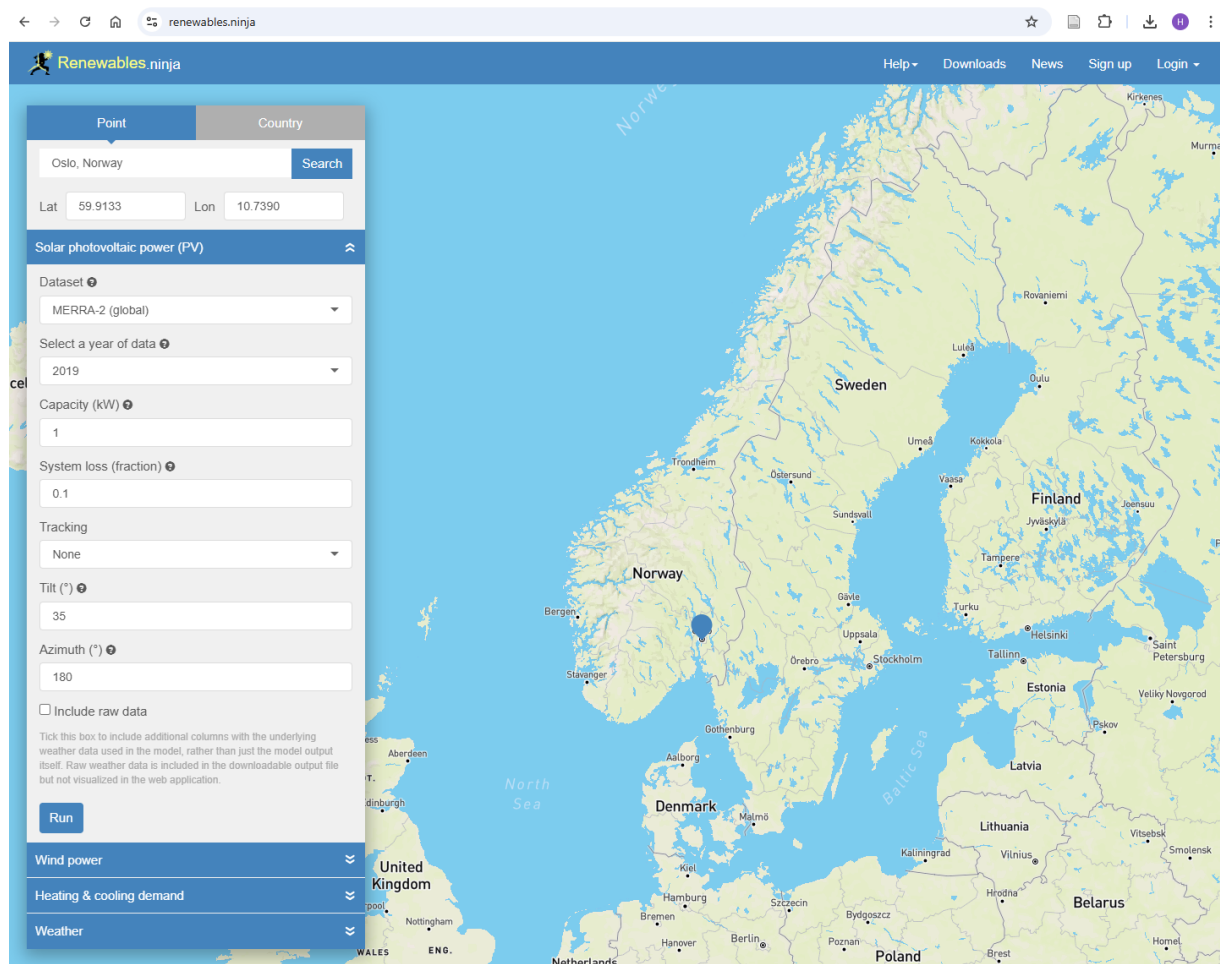


Figure 8. Web-based map interface for Renewables.ninja [14].

The platform allows users to specify location parameters such as latitude and longitude or select country-level datasets. Additional inputs include technology type (PV or wind), system

parameters such as tilt, azimuth, and installed capacity, and the desired time period, which can range from individual historical years to multi-decade datasets. The outputs consist of hourly capacity factors or generation profiles, downloadable CSV datasets for integration with other modeling tools, and API responses in JSON format for batch processing. Registered users can submit up to 50 API requests per hour, and Python scripts are available to facilitate automation and integration with other workflows.

Applications and results

Renewables.ninja can be used in energy system modeling to generate realistic PV and wind profiles for grid studies and flexibility analysis. It also supports techno-economic assessment of renewable integration and enables climate and policy scenario evaluations at regional and continental scales. Within the context of FME SOLAR WP3, Renewables.ninja is particularly relevant for producing long-term PV generation profiles for Norway and Europe, which can be linked to demand-side flexibility models and grid impact studies. The tool has already been used in academic research at NMBU, including several master's theses. One example is shown in Figure 9 [19], which illustrates a production profile for a 10 kWp PV system located in Oslo based on data from Renewables.ninja [14].

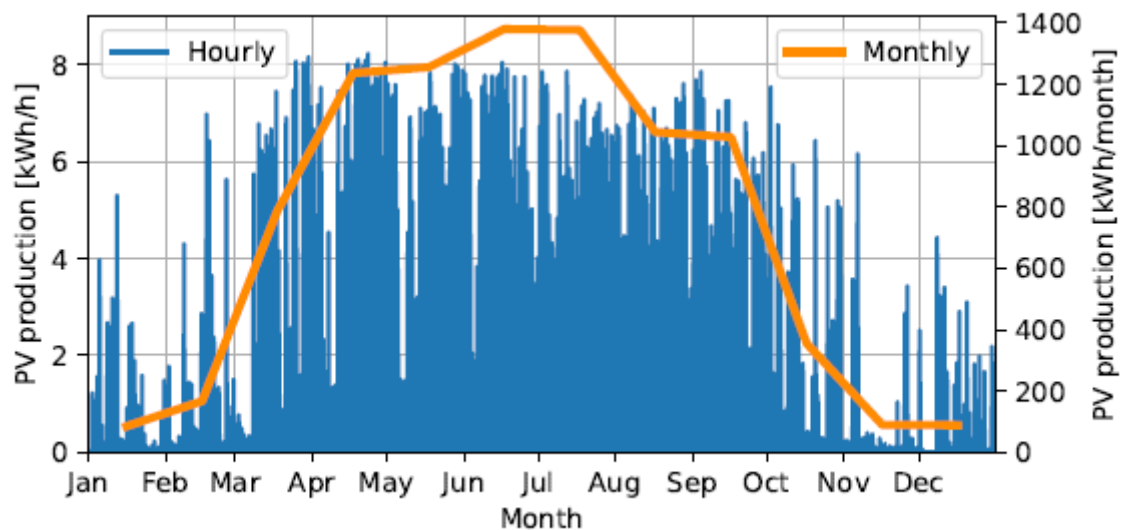


Figure 9. Production profile for a 10 kWp PV system located in Oslo [19]. The profile is based on data from Renewables.ninja .

System Advisor Model (SAM)

By Heidi S. Nygård, NMBU

Model structure

The System Advisor Model (SAM) is a free, open-source desktop application developed by the U.S. Department of Energy's National Renewable Energy Laboratory (NREL) for techno-economic modeling of renewable energy systems [20]. SAM combines detailed physical models with financial analysis tools to evaluate the performance and economic feasibility of renewable energy projects. It supports multiple technologies, including photovoltaic (PV) systems, battery storage, concentrating solar power (CSP), wind power, and other renewable options. For PV systems, SAM uses hourly or sub-hourly time series data to simulate system behavior under realistic operating conditions, accounting for irradiance, temperature, shading, and system losses.

Input, output and interface

SAM's workflow begins by selecting a performance and financial model for the project. The software automatically populates input variables with default values, which users can adjust to reflect site-specific conditions. Inputs include location and resource data, system design parameters, installation and operating costs, and financial assumptions such as tariffs and incentives. Weather data can be imported from sources like NSRDB, TMY, and Meteonorm, or downloaded directly from integrated databases.

For PV systems, users specify module and inverter characteristics, array configuration, and tracking options. SAM offers detailed component-level modeling using databases such as the California Energy Commission (CEC) module library and Sandia inverter library. Users can also configure battery storage and define system losses (soiling, shading, DC/AC losses).

Outputs include detailed hourly performance profiles, annual energy yield, system losses, and economic indicators such as levelized cost of energy (LCOE), net present value (NPV), and payback period. SAM provides a graphical user interface for system design and analysis, as shown in Figure 10, and supports scripting through PySAM for automated workflows and integration with other modeling tools. Post-simulation checks help identify issues such as inverter undersizing or excessive clipping.

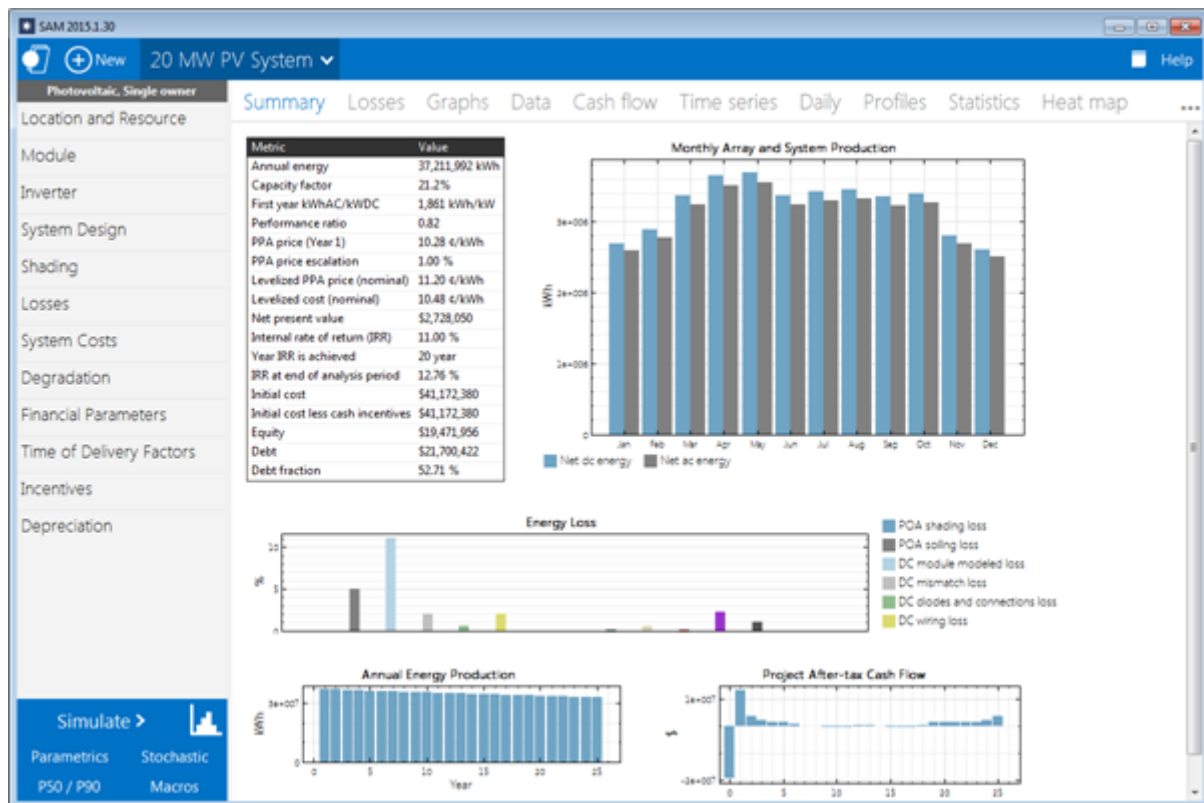


Figure 10. SAM interface showing PV system design and performance results [20].

Applications and results

SAM is widely used for feasibility studies, system optimization, and scenario analysis. It enables users to assess the impact of design choices, climatic conditions, and operational strategies on both technical performance and economic outcomes. In the context of FME SOLAR WP3, SAM is used in FYS375 at NMBU, a laboratory course in energy technology, for simulating PV systems supplying power to the grid.

PVCase

By Elin Sødahl, IFE

Model Structure

PVCase is a software suite for project development and PV system design [21]. The suite includes four products, Prospect for site selection and assessment, Ground Mount and Roof Mount for design of PV power plants, and Yield for energy yield assessment. PVCase can be used as an integrated workflow, as illustrated in Figure 11. Prospect and Yield are cloud-based services, whereas Ground Mount and Roof Mount rely on a local installation of AutoCAD. In addition to being an integrated workflow, PVCase also allows for exporting PV system layout to PVSyst. The design tools, Ground Mount and Roof Mount, are semi-automated, and include site grading, cabling and options to add DC and AC coupled battery energy storage systems to the PV plant design.



Figure 11. PVCase modelling workflow.

Input, output and interface

The specific inputs and outputs depend on which PVCase product is being used. The suite is designed such that the whole workflow from site selection to energy yield can be performed without requiring input files, but it also allows for the import of custom files, such as terrain data and meteorological information.

Prospect evaluates the suitability of land for PV development and does not require input files. It analyses sites based on built-in data for environmental factors (e.g. rivers, forests, wetlands) and infrastructure (e.g. roads and buildings). For sites in the USA it can also provide grid and property owner information. The software outputs a map for the buildable area at a given location.

Ground Mount and Roof Mount require input site information, either imported from Prospect or imported by the user. The local terrain can be imported directly from Google Global Terrain where available. Based on user-defined input, such as module type, tilt angle, tracking systems, and stringing specifications, the software builds a 3D model of the PV system. Component libraries for PV modules and inverters are included, but users can also define custom components. The outputs are a complete PV system layout, a detailed bill of materials, and a

simplified yield estimate. Layouts can also be imported from PVSol for further model development

Yield calculates the estimated energy yield of a PV power plant with an hourly resolution. It requires a 3D plant layout, which can be imported from Ground Mount or Roof Mount. The tool performs ray-tracing simulations to compute incident irradiation, accounting for near-shading effects and bifacial gain. Metrological data can be imported from PVGIS [22] or SolarAnywhere [23], or from custom files. Yield provides performance profiles with an hourly resolution as well as a detailed loss diagram.

Application and results

PVCase is used for PV plant development where feasibility studies can be performed using the built-in functionality for optimizing shading and total capacity. The conditions on site are implemented in the workflow through terrain and weather data delivered by third parties. PVCase is used in FME Solar in WP1, where the performance and functionality of PVSyst and PVcase in complex terrain will be compared.

Pvlib python

By Mari Øgaard, IFE/UiA

Pvlib python [24] [25] is a community developed python library that provides a set of functions and classes for simulating PV generation.

Model structure

The library contains multiple different models for the different steps in PV modeling, e.g. models for irradiance transposition, irradiance of bifacial or tracking systems, module temperature, PV module/cell generation, and inverter efficiency. Functions for collecting weather/irradiance data from different open sources are also implemented in the library. Pvlib has standardized model chains which enable quick implementation of PV generation simulation, but with the availability of multiple different functions for the different modeling steps it is also possible to easily build your own PV generation model. When building your own model, it is possible to choose between different models in the different modeling steps, and to choose how complex the model should be, for example how many loss mechanisms that should be included. The library is thus providing a flexible toolbox for simulating PV generation.

Input, output and interface

The toolbox is flexible, and the input and output of the simulation depends on which modeling step that is used. For PV generation simulation, the input is weather and irradiance data from any source and information about the location (latitude, longitude) and system configuration (module tilt, azimuth). What type of weather and irradiance data that is needed will depend on which specific models that are used. For simple models, global horizontal irradiance and ambient temperature are typical weather input. Output from all the parts of the model chain is available. How to use the different models and the model chain are shown in the pvlib python documentation at [24].

Application and results

Because of its flexible structure, pvlib python can easily be used in combination with other models. It can be used directly into other python based modeling framework, or it can be used to generate input to other models, such as PV output, plane of array irradiance, etc. Pvlib python is widely used in both research and commercial applications.

Energy storage models

VirtualFCS

By Mari Juel, SINTEF

Model structure

VirtualFCS is a Modelica library that can be used to perform dynamic simulations of hybrid energy systems. VirtualFCS was originally developed for hybrid fuel cell battery systems used in transport applications (cars, trucks, buses and ships) but is now extended to be used for also stationary applications. PV productions profiles from e.g. PVsyst can be given as input to the model. The strength of this library is its detailed models for storage technologies such as Li-ion batteries, hydrogen technologies and their balance of plant- (BoP) and sub-components. The library also includes sub-library on Control strategies for controlling hybrid energy systems. Virtual-FCS is based on Modelica language which is a equation-based, object-oriented language for modelling complex, multi-domain physical systems. This approach allows to build component models where mechanical, thermal and electrical properties can be split into different sub-models. The component models are connected to system models by physical principles. This approach facilitates the modeling of complex systems by focusing on the inherent physics and allowing for the easy reuse of components through its object-oriented structure. VirtualFCS consists currently of 11 sub-libraries including 44 components and sub-system models as illustrated in Figure xx. An XInTheLoop library makes it possible to perform X-in-the-loop simulations.

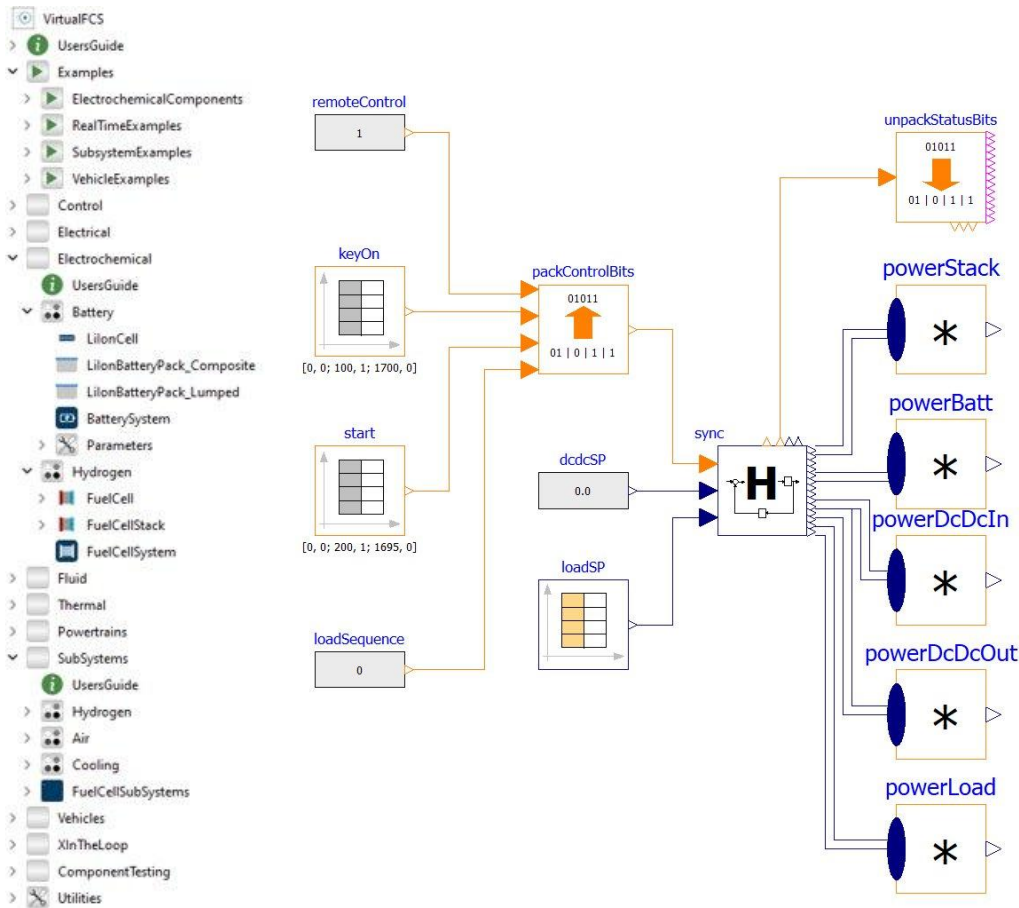


Figure 12. VirtualFCS sub-libraries (left) and example of a system diagram (right).

Input, output and interface

The input to VirtualFCS model is system architecture and battery and/or hydrogen system specifications including the specification of their sub-systems. (e.g. co specifications and size of storage technologies. Hybrid energy systems including PV systems require time series of PV production data either from real production facilities or simulated by the use of PV generator models.

The output of the VirtualFCS is process parameters such as power, temperatures, pressures and flow rates from each of the components, their efficiency and system efficiency. Based on degradation models for the different components it is possible to estimate the degradation and lifetime of components based on different operation conditions.

Application and results

VirtualFCS provide the effect of architecture changes on performance and dynamic response, as well as how low level control such as BMS of the battery and the fuel cell subsystem control and the high level control like power distribution/EMS. The models can be used to simulate different configurations of storage solutions and how they behave based on different PV and load profiles. The models also allow for testing different energy management strategies and how different use and configurations impacts e.g. the degradation and lifetime of the different storage units.

Energy system models

EnergyModelsX

By Ingeborg T. Røe, SINTEF Energy

Model structure

Energy Models X (EMX) is an open-source energy system modelling toolbox based on the Julia programming language and using the JuMP modelling framework [26]. The toolbox enables a fast and modular framework for multiple dispatch models, mirroring real energy systems through generic representations of different technologies.

These representations are defined in the base package, `EnergyModelsBase`, and include sink, source, generation and storage technologies, and links between the different technologies. The generic representations can be assigned features specific for the energy system in question such that specifications of that energy system are captured. The technologies are referred to as *nodes*. Three different fundamental nodes exist; source, sink and network nodes. Interactions between nodes are defined by creating *links* between them. Source nodes can only give and sink nodes only receive input from the surroundings (i.e. is only linked one way). Network nodes can have two-way interactions which enable the aforementioned generation and storage technologies. While detailed features can be assigned to nodes, links can only have limited features, such as an associated emission. Hence, most of the interaction between a node and its surrounds are defined in the node. Furthermore, links can be defined between any nodes, but modelling ease and flexibility are best achieved by defining an *Availability node*. The Availability node does not have any features assigned to itself, but acts as a collector of all links as illustrated in Figure 13. That way, new nodes can be defined in the framework without changing all the links between the existing nodes.

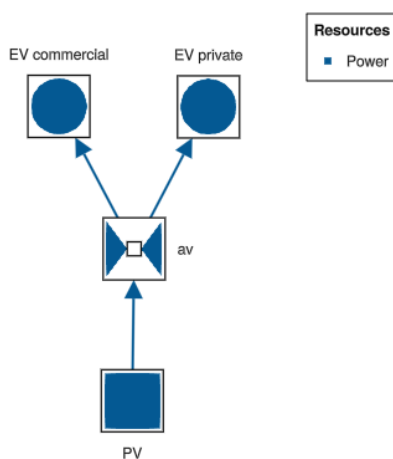


Figure 13 Illustration of a system with electric vehicles (EV) charged in private and commercial buildings from PV power.

Together, the nodes and links make up the EMX model, on which either operational or investment modelling can be run. In operational mode, the operation of the different

components, or nodes, in the model is optimised, but changes to the nodes themselves are not allowed. Investment modelling, on the other hand, allows e.g. capacity expansion given that information on the capital expenditure (CAPEX) for the relevant expansions is available. EMX will then minimize the net present value of all costs (accumulated operational expenses (OPEX) and CAPEX) within the planning period.

The modularity of the EMX framework is achieved by Julia's multiple dispatch functionality when extending the EnergyModelsBase package. Several such extensions have been implemented to ease geographical representation (EnergyModelsGeography) and investments (EnergyModelsInvestments). Additionally, a series of technology specific extensions have been developed for implementation of e.g. PV power, hydrogen electrolysis and storage, and heat integration. A detailed description of the model and its use can be found in Ref. [27].

Input, output and interface

Once all the components in the energy system have been build using the different EMX packages, case-specific input is defined for the analyses. EMX can take input from yaml, excel or csv-files. The input depends on the type of components and the type of analyses performed. For operational analyses, input to sources typically contain:

- Installed capacity over the operational period (in the case of renewable energy source such as PV, this will vary virtually over time to mimic the variable energy output)
- Variable OPEX
- Fixed OPEX
- Output energy type
- Associated emissions

Network nodes (such as electric batteries) would additionally need input on:

- Input energy type

Sink nodes have different input requirements, typically comprising:

- Consumption profile over the operational period
- Penalty for surpluses or deficits
- Input energy type

We notice that subclasses can have additional input requirements. Heat pumps (built under EnergyModelsHeat) will for example require input on the temperature of the supply and return heat stream and the Carnot efficiency. Furthermore, certain links have properties attached to them, such as heat loss in pipeline networks, which also needs to be quantified in the input.

When investment analyses are run, each component needs additional CAPEX information. Furthermore, limits on the total capacity that can be invested in, and the steps in which investments can occur are inputted.

Finally, emission caps for the overall model in each strategic period can be inputted in order to mimic reduction in emissions over time. Input on the rate of return is also set for the overall model when investment analyses are performed.

EMX comes with a graphical user interface where the set-up of the model is displayed along with visualization of (some) results. Additionally, full output with respect to energy flows, capacity investments, energy curtailment, emissions and costs of operation over the planning horizon are stored and available.

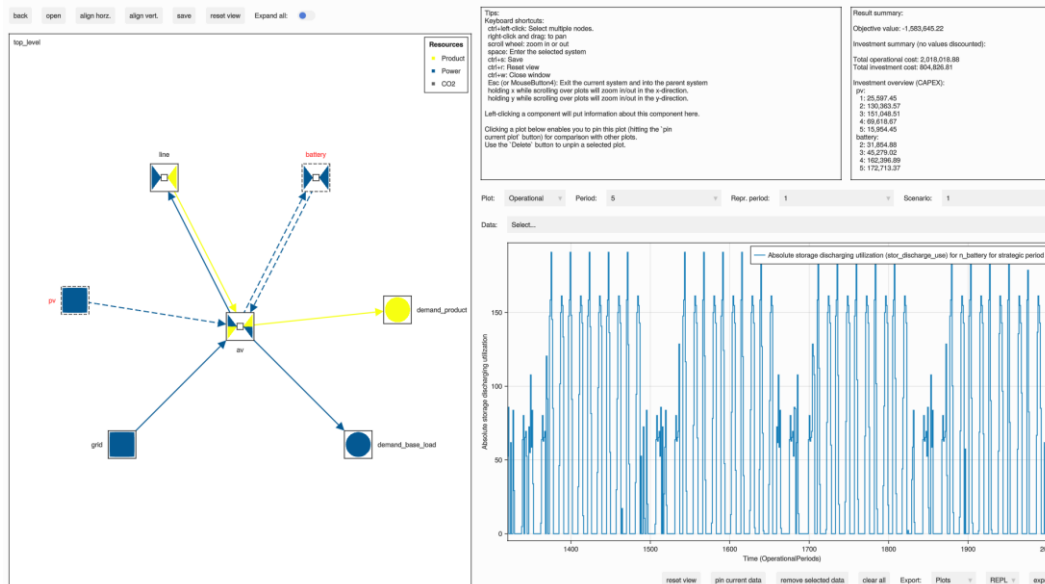


Figure 14 Illustration of the GUI with display of a battery discharge curve in an isolated system containing PV and battery, providing renewable power to the consumers.

Applications and results

EMX is used to obtain insights into the properties and behaviour of a range of energy systems, whether for isolated or geographically interconnected systems. The framework has the capability of mimicking the range of technologies relevant for the evolution of energy systems, as well as the operation of the energy system under specified conditions. It is best fitted to inform policy makers, industry and other stakeholders about how the energy system should and could evolve, especially regarding technology and capacity expansions, and less well-suited as a real-time operational tool even though certain insights into how the system or subprocesses operate can be obtained.

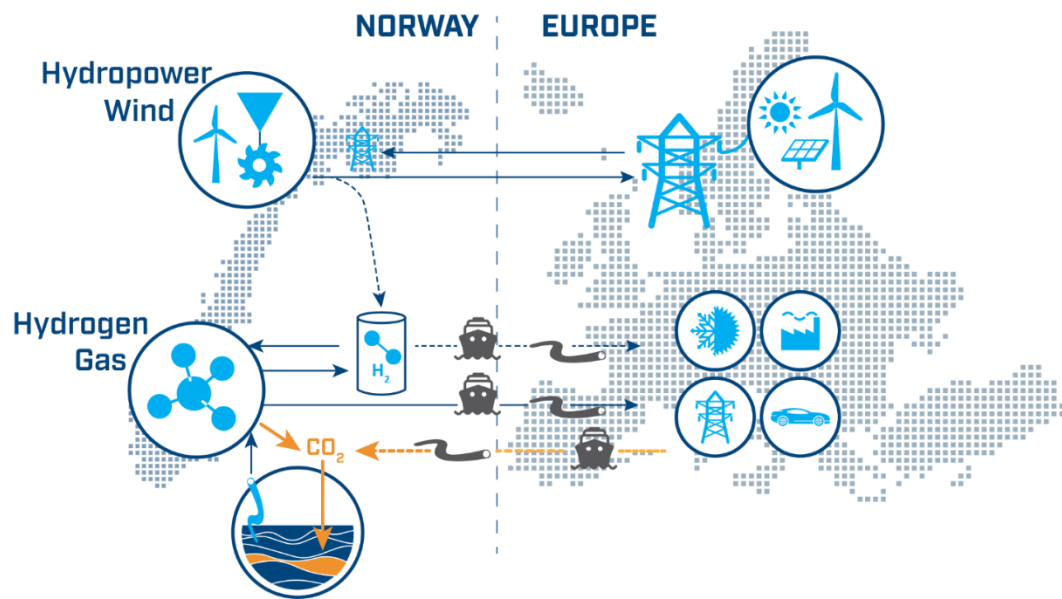


Figure 15 Illustration of an application for EMX, where the technology requirements and investment strategies for a trans-European energy system is investigated.

PyPSA-Eur economic dispatch for renewable energies

By Leonardo Rydin Gorjão, NMBU

Model structure

PyPSA-Eur is an open-source, large-scale model of the European power system that enables comprehensive techno-economic analyses of electricity networks under the transition to renewable energy [28] [29]. Built upon the Python for Power System Analysis (PyPSA) framework, it formulates system optimisation problems as linear programs that minimise total system cost while enforcing operational, technical, and policy constraints. By providing a transparent and flexible modelling environment, PyPSA-Eur performs cost-optimisation under CO₂-constrained budgets and capacity limitations, in a spatially extended energy systems, with variable control of the mesh of the energy system. It allows the user to explore the impacts of infrastructure investments, cross-border power exchange, and climate policy scenarios across an interconnected European grids. The model represents the European transmission system at the ENTSO-E level, including AC lines at and above 220 kV, all high-voltage DC links, and several thousand substations, based entirely on open data, while incorporating heuristic sector coupling from industry, farming, and heating demands, as well as energy storage technologies.

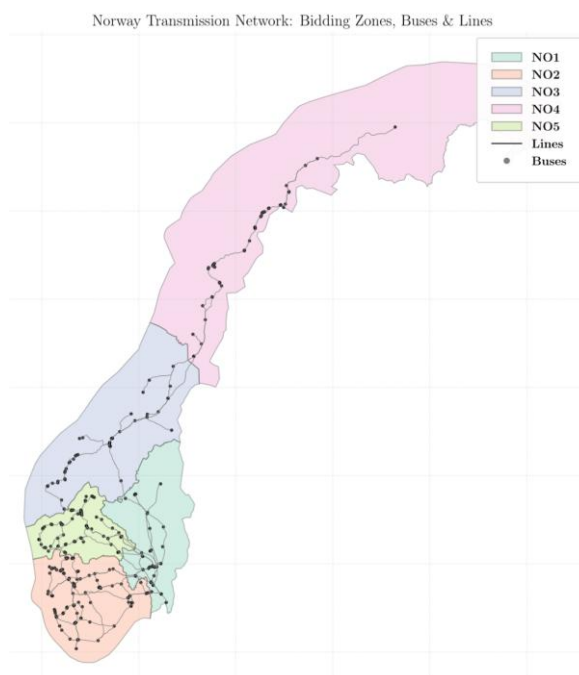


Figure 16 Model of the Norwegian high-voltage transmission lines from OpenStreetMap embedded in PyPSA-Eur with major substation connections [30].

PyPSA-Eur focuses on the high-voltage transmission grids, aggregated consumption predictions per node, and coarse-grain simplifications of grid parameter models, such as line capacities. There is a strong interplay with pandapower (c.f. sec. 7.1), as both share parameter information on the grid models. PyPSA-Eur models long-term, future cost-optimisation models in large European power systems, with an hourly resolution, where system dynamics are

modelled heuristically. Integration with realistic medium- and low-voltage grids is possible and augments the detail for voltage effects and improved demand models. PyPSA-Eur also integrates with the larger PyPSA-Earth project [33], which maps the entire world with open-source data and transfers the same model structure and workflow beyond Europe.

Input, output and interface

PyPSA-Eur integrates publicly available datasets on transmission infrastructure, demand, generation potentials, and weather-dependent renewable resources. These data include detailed network topology and parameters, conventional power plants, and spatially and temporally resolved time series for electrical demand and renewable generator availability, assembled into a consistent model of the European transmission system.

The data processing and model-building are organised in an automated workflow, implemented with Snakemake, which constructs scenario-specific networks from the raw datasets. Scenario definitions, technology and cost assumptions, clustering options, and optimisation settings are controlled through human-readable YAML configuration files, allowing users to switch between scenarios and configurations with minimal manual intervention.

At the model level, the inputs are the capacities, marginal costs, and time series for demand and availability, which are kept strictly separate from outputs, e.g. dispatch, power flows, prices, ensuring that input data are not overwritten by optimisation results and that workflows remain reproducible. The outputs from PyPSA-Eur, including such as optimal generation and storage capacities, dispatch time series, line loadings, curtailment levels, nodal prices, and total system costs, are stored in formats (CSV, NetCDF) that can be analysed directly in Python or with external tools. A prototypical excel input mechanism exists but its application is limited.

PyPSA is a purely programmatic systems and does not have a GUI. Users typically interact with PyPSA-Eur either via command-line Snakemake workflows, which run the full pipeline from data download to optimisation and plotting, or by working directly with the underlying PyPSA Network objects in Python for customised analyses. The structural compatibility with pandapower facilitates exchange of network parameters and the use of PyPSA-Eur results as boundary conditions for more detailed AC-grid studies (high-voltage vs low-voltage), while the integration with PyPSA-Earth provides a consistent interface for extending the same input-output logic to power systems outside Europe.

Applications and results

A central application of PyPSA-Eur lies in evaluating the role of variable renewable energy sources, particularly photovoltaic (PV) systems, in achieving cost-efficient decarbonisation pathways. The model incorporates spatially and temporally resolved solar resource data to simulate PV generation profiles and assess their contribution to system adequacy, flexibility needs, and network congestion. By coupling PV deployment with complementary technologies

such as storage, demand response, and transmission expansion, PyPSA-Eur facilitates the study of strategies for integrating high shares of solar energy at continental scale. Consequently, it provides a tool for identifying optimal investment portfolios and operational strategies that align with Europe’s long-term energy and climate objectives.

Applications of PyPSA-Eur allow, for example, the estimation of the impact of different technologies in the Nordics. An initial analysis of Statnett’s targets for 2040 points to a capacity expansion dominated by wind power (especially offshore DC) and existing hydro, with representative outcomes such as offshore DC at 9.8 GW, onshore wind at 3.7 GW, and hydro at 28.3 GW for the Nordics under the high-demand case, while solar remains effectively absent from the build-out. Within these boundary conditions – no sector coupling and an isolated Nordic market – the optimisation does not assign material value to PV for meeting load or reducing system cost; solar neither drives transmission expansion nor shifts the marginal generation mix. This pattern sits uneasily alongside forward-looking planning narratives that anticipate meaningful Nordic solar growth after 2030. The modelling signal is clear: without sector coupling (e.g. flexible electrified heat, hydrogen production, EV smart charging) or broader interregional exchange that raises the coincident value of daytime generation, PV does not emerge endogenously in the Nordics at policy-relevant scale [31].

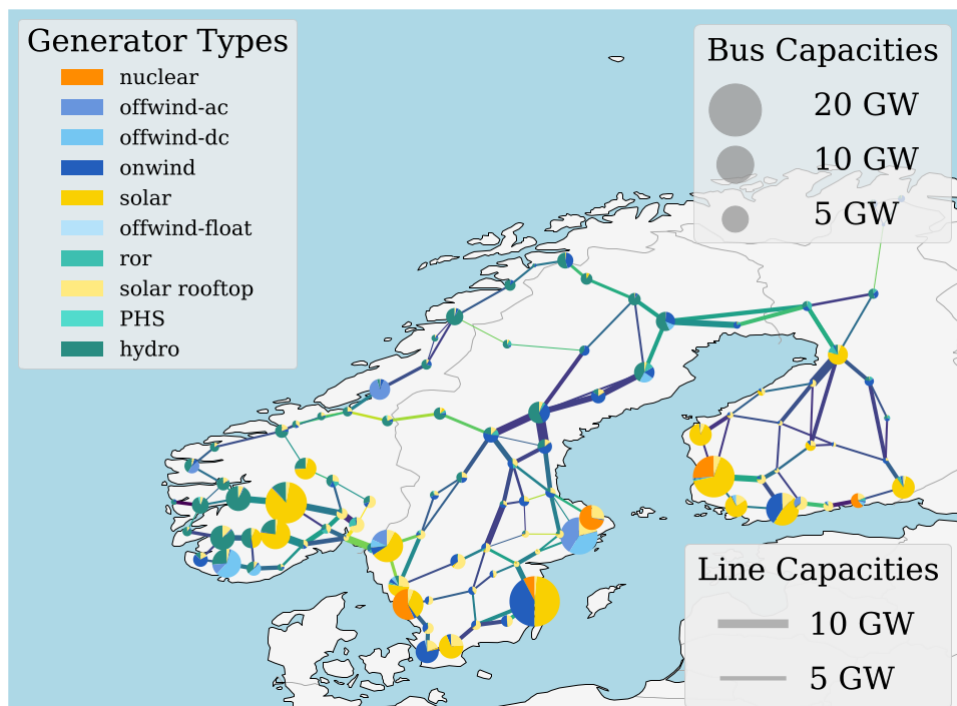


Figure 17 Economic dispatch optimisation for the Nordics in 2050 considering the doubling of the electrical load and sector coupling in relation to 2022. The requirements of 30 GW of solar power in Norway met with a very large distribution of local power generation by solar generation and solar rooftop generation [32].

In a conceptual model for the Nordics in 2050, with roughly double the electrical load of 2022, a dedicated solar expansion would require about 94.5 GW of PV (66.06 GW utility-scale and 28.42 GW rooftop), and delivers a total annual system cost of €34.6 bn. Grid reinforcement needs remain modest, about 4% additional line capacity, reflecting the geographic spread of PV and the buffering effect of hydro and other modelled stores (hydrogen and methane).

Compared across other explored scenarios, expanding solar keeps system cost close to the minimum: an optimisation case with sector coupling – shifting the solar mix toward rooftop and tracking – achieves the lowest cost at €33.9 bn with 69.5 GW total solar, whereas an offshore-wind-power-centric optimisation raises cost to €37.3 bn [32].

PyPSA-Eur's integration with PyPSA-Earth extends these applications beyond Europe. Promising results on the application of solar expansion in Tanzania point to the possibility of coupling the existing hydropower-driven system with localised, potentially decentralised PV generation in the inland part of the country, thereby actively contributing to ongoing electrical cooking policies and broader electrification goals [33] [34].

TIMES

By Stine F. Myhre, IFE

Model structure

The TIMES modelling tool is a technology rich, bottom-up model generator which utilizes linear programming to optimize based on least-cost for possible future energy scenarios. The TIMES modelling framework is a widely used optimization tool for energy systems and is developed by IEA-ETSAP [35]. The optimization framework is designed to support long-term energy planning, policy analysis, and scenario development by simulating the evolution of energy systems over time.

The TIMES framework is built to simulate the entire energy system from resource extraction to final energy consumption to meet energy service demand for all included sectors while satisfying the constraints such as emission targets, resource availability, and technology deployment limits. And by this aim to give results on, for example, technology deployments in the future energy system to meet the demand, emissions, trade-offs between different energy and climate policies, and sector coupling and interaction between different technologies.

TIMES is a highly modular and adaptable tool that allows the user to define and build their own system and support multi-spatial, multi-sectoral, and multi-temporal modelling. The tool includes a detailed representation of energy technologies such as power plants and power production, heating systems, vehicles, and multiple energy carriers. Additionally, the energy flow in the model starts from resource extraction, which includes domestic production and import of energy, conversion and transformation, and transmission and distribution to the final energy consumption, such as residential, commercial, industry, agriculture, transport, and heat. The model can incorporate technology learning, behavioral changes, resource constraints and policies and regulations. It also allows for scenario analysis, which can be used to explore different future energy transitions based on, for example, different emission goals, policies, population and energy development and regulation. TIMES supports a wide range of modelling capabilities such as linear and mixed-integer optimization, demand forecasting and end-use modelling, technology competition and substitution, emission accounting and reduction targets, resource depletion and trade modelling, storage and flexibility options, intertemporal investment decisions, and multi-objective and stochastic modelling.

Input, output and interface

The energy system models that utilizes the TIMES framework can have multiple different input parameters. The input parameters can be more or less divided into different categories:

1. **Economic parameters:** This includes cost data such as investment costs, fixed and variable operational and maintenance costs, fuel prices, prices for subsidies or incentives, electricity prices, CO₂ costs, and grid tariff costs.
2. **Technical parameters:** The TIMES framework allows for technology-rich modelling and will need technology-specific parameters that describe these technologies. The

technologies could include technologies such as generation units, storage units, transmission and distribution system infrastructure, energy conversion technologies, demand side technologies, transport, and carbon capture technologies. Some different relevant input parameters are installed capacity, efficiencies, capacity factors, ramp rates, lifetime, availability factors, emission factors, fuel requirements, and lower and upper limit for capacity expansion for different technologies.

3. **Demand-side related parameters:** The TIMES framework allows for detailed demand-side modelling. Demand-side related parameters includes for example, consumption data per sector, projected demand growth rates, price elasticity of demand, time of use pricing, flexibility in demand and production, variation in demand and generation profiles, penetration of different demand-side technologies such as in transport, self-consumption rates connected to energy resources, and division of demand use based on different processes such as heat, electricity, fuels, ect.
4. **Availability parameters:** This gives information about the availability of different technologies and industrial processes and can be related to the utilization of fuels, resources, and limitations and spatial variations in, for example transmission, renewable energy, fuels, and different energy carriers.
5. **Environmental, regulatory and policy parameters:** Input related parameters that can be used to set environmental and policy targets. This could include quantification of rules, incentives, restrictions, environmental constraints, different policies and regulations. Relevant input parameters are CO₂ and renewable penetration targets, air pollution values, carbon intensity of different technologies, carbon pricing, land use requirements, resilience of infrastructure, subsidies, net-zero targets, and electrification adaptation in demand-side.
6. **Market parameters:** These parameters define the rules and pricing mechanisms and structure for the market in the system. Could include input parameters to regulate electricity markets, different energy pricings and tariffs, transmission and generation capacity pricing for use of infrastructure and availability of generation units, reserve markets, and ancillary services markets.
7. **Scenario related parameters:** The TIMES framework supports scenario analysis. Relevant parameters used to analyze different scenarios and energy transition pathways can be modelled in own scenario files in the framework.

The TIMES framework has a supporting interface, called VEDA, where the user gets a great overview over input parameters, item details, scenario runs, and the output and results. The interface allows user-defined specifications and easy result gathering.

Output of the models includes, for example, technology deployments, emissions, cost of technologies and total system costs, production, operation of different technologies and investment and need for new infrastructure

Applications and results

The models utilizing the TIMES framework can provide a large set of results that can give key insights into how the energy system models operate under different technological, economic, policy, and environmental conditions. The results from the models can be used to understand how the energy system might develop under different conditions and provide input to policy makers, energy planning, and other stakeholders in the energy system.

IFE-TIMES-Norway

IFE-TIMES-Norway is a long-term optimisation model of the Norwegian energy system developed in the described TIMES modelling framework [36]. IFE-TIMES-Norway is a technology-rich model of the Norwegian energy system, divided into five regions corresponding to the current electricity market spot price areas (NO1-NO5) and offshore regions, as seen in Figure 18. The model provides operational and investment decisions from the starting year 2018, towards 2050. The model optimises investment and operational decisions with a model period for every fifth year from 2020 to 2050. To capture operational variations in energy generation and end-use, each model period is divided into 96 sub-annual time slices, where four seasons are represented by a day of 24 hours. The electricity spot prices in the bidding areas in Norway are endogenous, as those are the dual values of the electricity balance equation, while the electricity prices in the neighbouring countries with transmission capacities to Norway are exogenous.

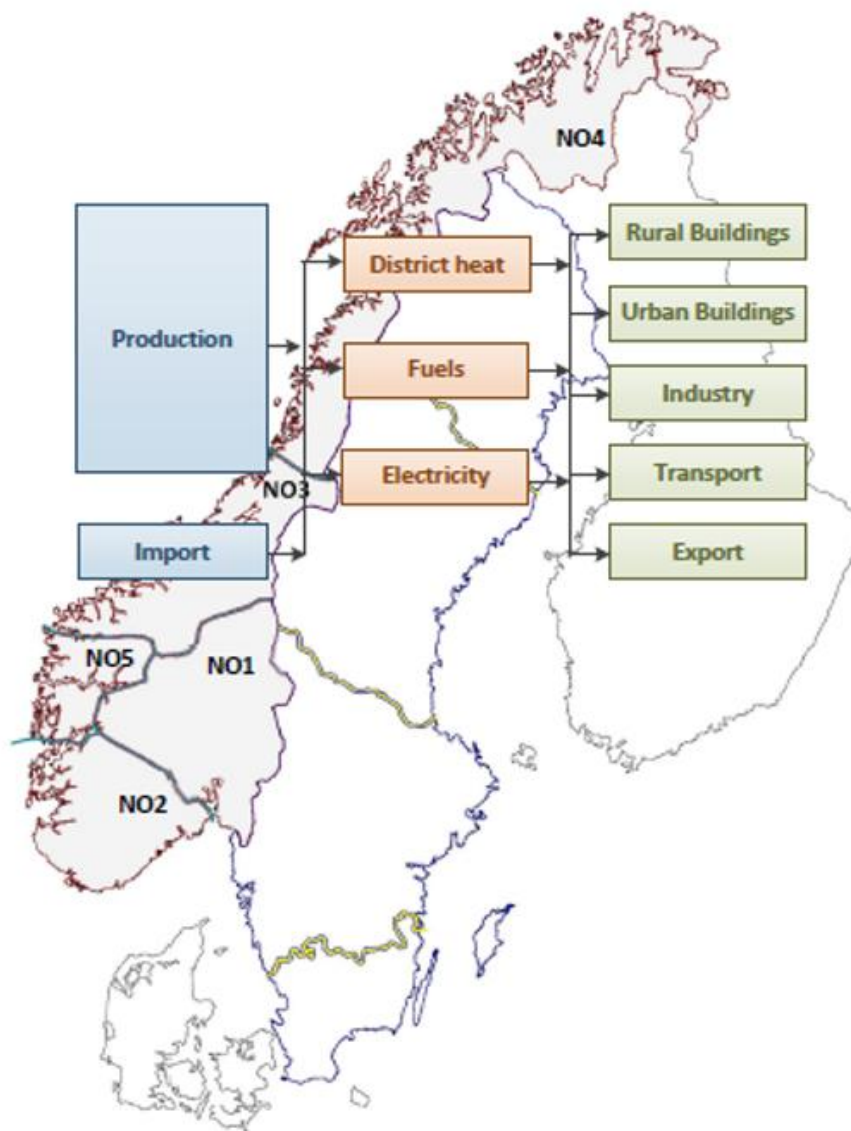


Figure 18 Geographical coverage and topology of the IFE-TIMES-Norway model, including the five spot price regions (NO1-NO5) in Norway.

IFE-TIMES-Norway includes a large variation in energy technologies and various options for input, such as demand for different energy carriers, technology-specific parameters, electricity prices, grid-related parameters, and emission factors. A schematic overview of some input data, the core functionality of the model, and the output is shown in Figure 19. For more specifications related to input, output, results, interface and application, see sections 0 since they also apply for IFE-TIMES-Norway.

The model captures the interplay between sectors, technologies, energy carriers, and emissions. Additionally, the model provides a detailed description of energy end use, and energy demand is divided into numerous end-use categories across industry, buildings, and transport. The electricity generation consists of offshore wind, onshore wind, building-applied solar power, solar parks, and hydropower, divided between regulated and run-of-river hydropower. In addition, the model allows for investment in hydrogen, thermal storage, district

heating, and energy storage technologies. The model will give results connected to investment decisions in different technologies, operation decisions, energy flows, emissions, domestic electricity prices, marginal costs, information about energy carriers in end-use, development of different end-use sectors and rate of electrification, how different sectors interact, and scenario-specific indications.

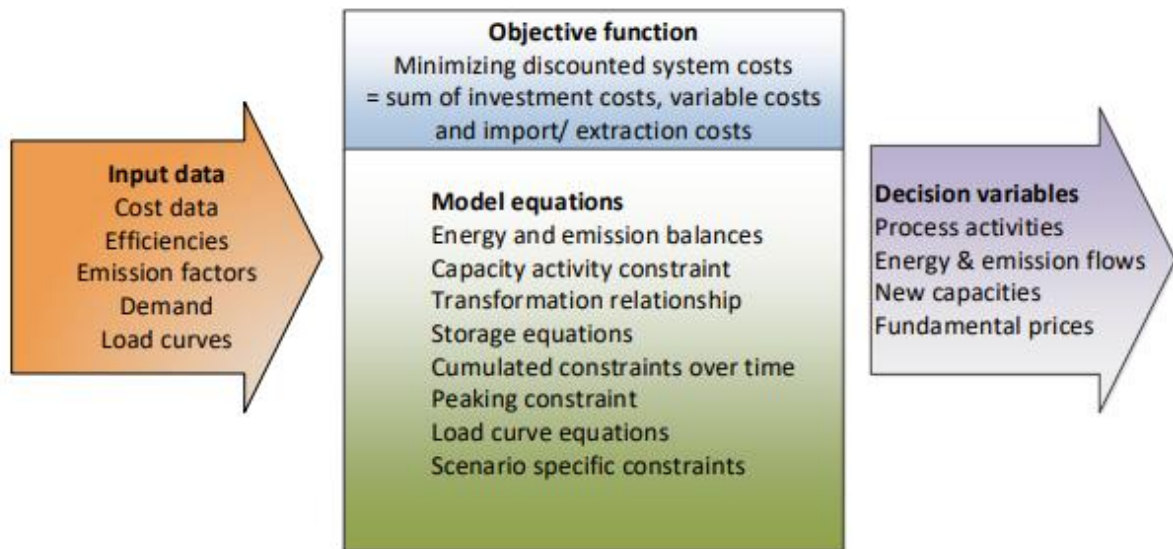


Figure 19 Schematic of model inputs, objective, and outputs.

The results from the IFE-TIMES-Norway model provides with key insights in how the Norwegian energy system develops under different input parameters and energy transition pathways. The modelling results from IFE-TIMES-Norway can be utilized to understand how the Norwegian energy system might develop under different conditions and provide input to policy makers, energy planning, and other stakeholders in the energy system. The model will give results connected to investment decisions in different technologies, operation decisions, energy flows, emissions, domestic electricity prices, marginal costs, information about energy carriers in end-use, development of different end-use sectors and rate of electrification, how different sectors interact, and scenario-specific indications.

JulES

By Julien Cabrol, NVE

JulES is an open-source energy market simulation model for operational planning [37]. It is developed by NVE and can be run through the FRAM modelling framework [38].

Model structure

JulES is structured around three main components in the forward simulation steps: price prognosis models, storage valuation models, and a market clearing model. The price prognosis models generate long-, medium-, and short-term price forecasts for multiple scenarios, forming the basis for operational decisions. These prices feed into the storage valuation models, which calculate water and battery values for individual subsystems using stochastic or deterministic approaches tailored to each technology's characteristics and time horizon. Finally, the market clearing model optimizes dispatch for a couple of days, incorporating detailed system representation and the storage values computed earlier. This modular design, built on TuLiPa's [39] generic component framework, enables JulES to flexibly represent different technologies - such as hydropower and batteries - while solving complex problems efficiently through decomposition and parallelization.

Input, output and interface

Technical and cost data for generation and storage technologies, electricity-demand time series, grid topology and transmission capacities, and renewable-generation profiles imported from external models (e.g. solar PV profiles).

Detailed results including market prices, generation dispatch, hydropower-reservoir operation and storage values, and trade.

Implemented in Julia and can be run through the FRAM modelling framework in Python. FRAM allows you to build and run different power market models and compare them against each other. It gets data from a common model-neutral database and efficiently performs all necessary data transformations and operations needed to build a model, solve it, and get the results.

Applications and results

JulES is designed to simulate the Northern European power market with high temporal resolution, detailed hydropower (or any other technology), uncertainty in weather, and using only open-source software. It is particularly suited for studying market behaviour and operational planning in hydro-dominated systems. For solar-energy applications, JulES can assess how variable PV generation affects electricity prices, hydropower operation, and the utilisation of flexibility resources such as batteries. The model supports scenario testing

renewable-energy integration and price impacts, providing insights for research, policy development, and strategic planning.

Grid models

Pandapower: Python Tool for Power System Analysis

By Joao Leal, UiA

Model structure

Pandapower is an open-source Python library designed for power system modelling, analysis, and optimization. Built on top of solid Python libraries, pandas and NumPy, it provides a user-friendly and efficient framework for simulating electric power systems ranging from simple radial grids to complex meshed transmission networks [40]. Its primary goal is to make power system studies accessible, reproducible, and easy to integrate with other Python-based scientific and machine learning workflows. Its flexibility, transparency, and strong community support make it a very useful tool for engineers, researchers, and students working in the fields of:

- Distribution and Transmission Network Studies
- Integration of Renewable Energy Sources
- Grid Planning and Operation Optimization
- Smart Grid and Microgrid Simulations
- Education and Research in Power Engineering

Key Features

- Network Modeling: Comprehensive set of components (buses, lines, transformers, loads, generators, storage, switches).
- Power Flow Analysis: AC and DC load flow with Newton-Raphson, backward/forward sweep, and Gauss-Seidel methods.
- Optimal Power Flow (OPF): Optimization of generation dispatch, voltage profiles, and grid operation.
- Short-Circuit Analysis: Fault current calculations according to IEC-standards.
- State Estimation: Estimation of grid states based on sensor data.
- Time Series Simulation: Dynamic studies with varying demand, generation, and weather inputs.
- Data Integration: Native compatibility with pandas for seamless data handling.
- Visualization: Built-in plotting functions with matplotlib and plotly support.

Pandapower therefore enables advanced and comprehensive grid studies with relatively little code. Its strengths are particularly evident in applications involving distributed energy resources such as rooftop PV integration in LV distribution grids [41] and smart grid development. A series of benchmark tests in [42] demonstrated that pandapower fills a gap among open-source power system tools by providing a powerful, flexible, and user-friendly platform that is suitable for real-world distribution grid studies.

Input, output and interface

Pandapower relies on a **tabular data structure**, in which each component type (line, bus, transformer, load, etc.) is represented by a dedicated table containing its electrical parameters. A corresponding result table stores the outputs of power flow, OPF, short-circuit, or state-estimation calculations. Because the tables use the pandas framework, they can store heterogeneous data types (electrical parameters, status flags, metadata, and descriptive fields) while remaining fully expandable without affecting core functionality.

The typical steps to build a distribution grid model in pandapower include the following:

1. **Define the LV bus configuration** (bus type: "slack" for external grid connection, "muff" for customer distribution centers, "bus" or "node" for operational nodes or load connection points; connected element: what cable, load, transformer, or generator is attached downstream; voltage level in kV), see example of a net.bus table:

bus	name	vn_kv	Type	zone	in_service
0	MV_Slack	22.0	slack	0	True
:	:	:	:	:	:
2	CDC_2	0.415	muff	1	True
:	:	:	:	:	:
10	EndBus_1	0.415	node	1	True
:	:	:	:	:	:

For visualization, a geodata table may be added:

bus	x	y
0	0.0	0.0
2	85.0	-40.0
:	:	:

2. **Define the transformer** (require rated voltages, apparent power, short-circuit parameters, iron losses, no-load currents, and tap settings), see example in table:

trafo	hv_bus	lv_bus	sn_mva	vn_hv_kv	vn_lv_kv	vk_percent	vk_r_percent	pfe_kw	i0_percent
0	0	1	0.2	22.0	0.415	6.06	0.75	1.53	0.17
:	:	:	:	:	:	:	:	:	:

3. **Define the lines** (LV underground cables, e.g., TFXP 1 kV, are added with their physical lengths and electrical characteristics), see example in table:

line	from_bus	to_bus	length_km	r_ohm_per_km	x_ohm_per_km	c_nf_per_km	imax_ka
0	1	4	0.120	0.642	0.083	210	0.24
1	4	10	0.080	0.642	0.083	210	0.24
2	2	15	0.350	0.860	0.090	180	0.18
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

4. **Define loads and generation** (static loads active and reactive power and distributed generation), see example in tables:

Load	bus	p_mw	q_mvar
0	5	0.045	0.012
1	6	0.032	0.008
⋮	⋮	⋮	⋮

sgen	bus	p_mw	q_mvar
0	6	0.015	0.000
1	10	0.050	0.000
⋮	⋮	⋮	⋮

In practical studies, like the application presented next, these static values are replaced by hourly time-series imported from measurement data or PV performance simulations.

Applications and results

In a Master Thesis at the University of Agder (UiA) [43], a representative LV grid, based on Lnett system data, was implemented in pandapower, see Figure 20.

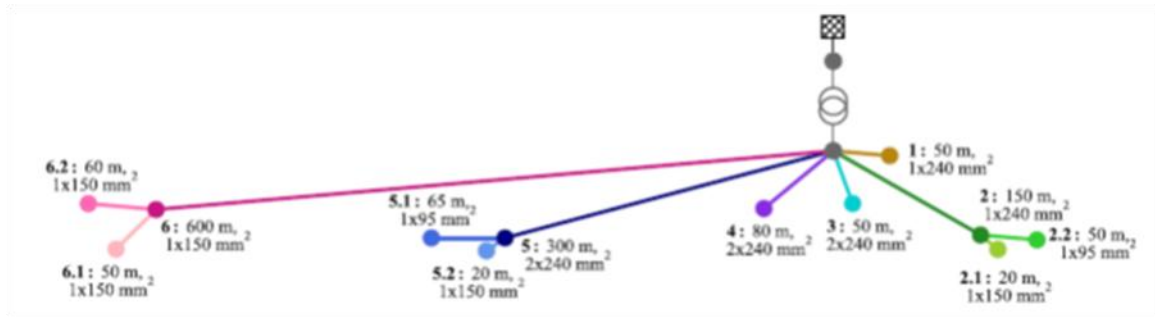


Figure 20. LV grid.

Hourly consumption data from different consumer categories and yearly PV generation profiles from PVSyst (for the Rogaland region) were then incorporated (Figure 21 and Figure 22). Each hourly value was inserted into the pandapower model via a time-series loop, resulting in outputs such as:

- net transformer power
- line loading time series
- end-node voltage variation

Example results include the hourly net transformer power (Figure 23), normalized line loading (Figure 24), and end-node voltage evolution (Figure 25).

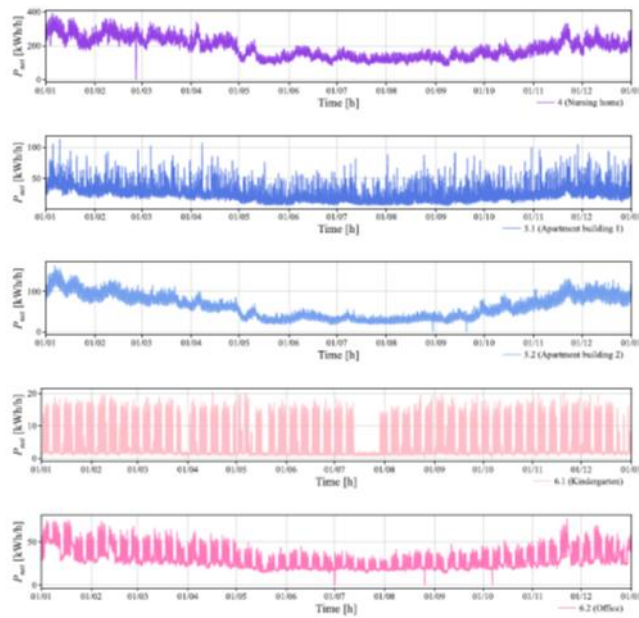


Figure 21. Consumption profiles.

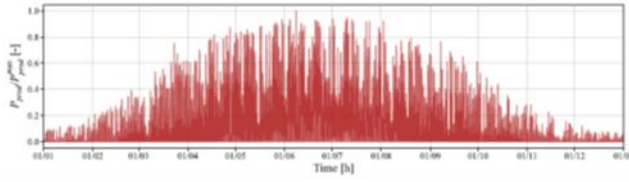


Figure 22. Normalised hourly PVPP production.

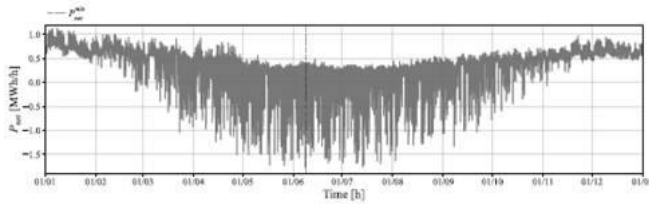


Figure 23. Net transformer power P_{net} (hourly consumption minus production).

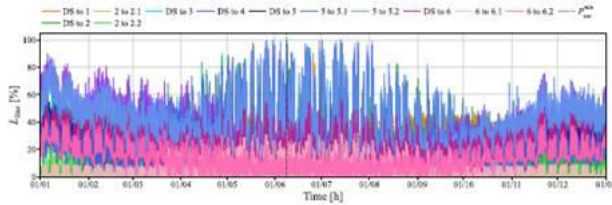


Figure 24. Line loading time series (normalised).

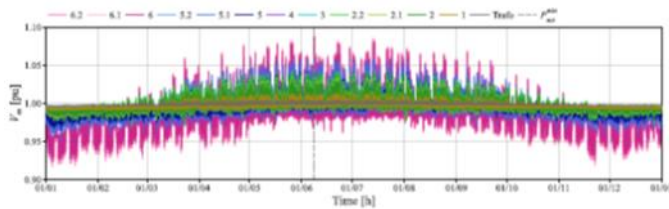


Figure 25. End-node voltage time series.

Possible coupling with other models

Because pandapower is implemented in Python, it can be easily coupled with other Python-based tools:

- **pvlb**: pvlb outputs standard Python structures (NumPy arrays, pandas Series), making it directly compatible. Pandapower requires only p_{mw} (and optionally q_{mvar}) for each PV generator, which pvlb can supply as AC power time series.
- **PVSyst**: Although PVSyst is not a Python-based tool and has no direct interface with pandapower, it can export AC power time series in CSV or Excel format. These can be imported into pandapower, as demonstrated in the previous application.

- **PyPSA-Eur:** Pandapower can also be used downstream of PyPSA-Eur in a multi-scale workflow. PyPSA-Eur may determine regional load growth, renewable deployment, or transmission flows, while pandapower can analyze the detailed LV/MV impacts, such as voltage quality, thermal limits, transformer loading, and cable constraints. However, some effort is required to align the two frameworks because PyPSA-Eur uses xarray-based network objects (.nc) whereas pandapower uses pandas element tables.

DlgSILENT PowerFactory

By Ingeborg T. Røe, SINTEF Energy

Model structure

PowerFactory [44] is a licensed power system software used for load flow analyses of the generation, transmission and distribution systems. All components of a power system, such as generators, transformers, transmission lines, loads, and protection devices, can be represented as discrete objects with defined attributes and behaviour. The software runs on Windows and can model advanced and comprehensive, dynamic systems, including balanced and unbalanced AC/DC grids and dedicated PV systems accounting for variations due to solar irradiation. Further, the models can include control strategies, such as active and reactive power regulation. The modular architecture ensures scalability, making it suitable for small-scale networks as well as for large, interconnected grids.

Input, output and interface

PowerFactory provides an advanced and functional graphical user interface, where system design and input, such as network topology, equipment parameters, operational constraints and time-series data for dynamic simulations, can be modulated. This input data can also be imported directly via scripting interfaces with Python or DPL. Outputs include numerical results, graphical plots, and detailed reports for analyses such as load flow, short-circuit, harmonic distortion, and stability studies. Results can be exported in multiple formats for integration with other engineering tools or for documentation purposes, or be displayed and visualized in the GUI.

Applications and results

PowerFactory is used for planning, operation and optimization of electrical power systems. Typical applications include load flow analysis, contingency studies, fault calculations, dynamic and transient stability assessments, protection coordination, and renewable integration studies. While models for the Nordic energy system have been build [45], and PowerFactory can be interfaced with Python, the software does not connect directly with other tools.

Models linking and Gap analysis

By Igor Sartori, SINTEF Community

Linking energy demand with energy system models

The Building-FLEXor model has been used in combination with energy system models as part of the activities in both the [Flexbuild](#) project and the [FME ZEN](#) centre – Zero Emission Neighbourhoods in Smart Cities. To scale up the energy demand to represent the entire building stock of Norway, eventually partitioned in climatic or market or other sub-areas, Building-FLEXor likewise Building-PROFet that provides the baseline non-flexible demand, have been combined with the building stock model RE-BUILDS as first described in [46] and subsequently applied also in [47] [48] [5]. However, it is in [4] and [6] that a complete linking has been performed between Building-FLEXor and the energy system models IFE-TIMES-Norway and EMPIRE, respectively. The linking with EMPIRE focused on analysing the impact of activating energy demand flexibility in the building stock on the power exchange between Norway and Europe. However, since EMPIRE is not a model that will be used in the FME Solar this linking experience is not described further.

Linking Building-FLEXor with IFE-TIMES-Norway

The Flexbuild project aimed to understand how end-use flexibility could impact the future energy system in Norway. Flexbuild used a set of models with different sectoral coverage, thus capturing different aspects of the future energy system:

- BUILDopt, which was basically Building-FLEXor (performing operational optimisation) with an additional layer for investment optimisation of a building's energy system. To scale up the results at aggregated level, this was used in combination with RE-BUILDS, which is a dynamic model simulating the long-term development of the Norwegian building stock;
- IFE-TIMES-Norway as a model of the energy system in Norway;
- EMPIRE, here directly linked only with TIMES, as a long-term stochastic optimization model of the European power market;
- EMPS as a power market model of the Nordics with detailed representation of the Nordic hydropower system.

The models are linked two-by-two in a soft linking, as shown schematically in Figure 26. Soft linking means that the models do not exchange information while processing, i.e., at each time step of the simulation. Rather, when one model has completed its simulation for the entire period of interest (in this project from 2020 to 2050) its results are used as input by the other model with which it is linked. Since (some of) the inputs of a model are (some of) the outputs of the other, the two models influence each other and there is the need to iterate between them until the key variables – those that form the mutual input/output soft link – converge to values that do not vary significantly between one iteration and the previous (within a margin of tolerance). EMPS was not linked but simply used to verify the impact of the results on the Norwegian hydropower system.

The key variables exchanged in the linking between BUILDopt (/alias Building-FLEXor) and IFE-TIMES-Norway are, as shown in Figure 26, energy prices and energy use. These are meant per each energy carrier and aggregated per each NO-area, within each storyline and simulation year. While both models operate with an hourly resolution, BUILDopt simulates an entire year, thus 8760 hours, and TIMES simulates only representative days, one per season. The representative days were extrapolated as the average of the working days from the full year dataset from BUILDopt.

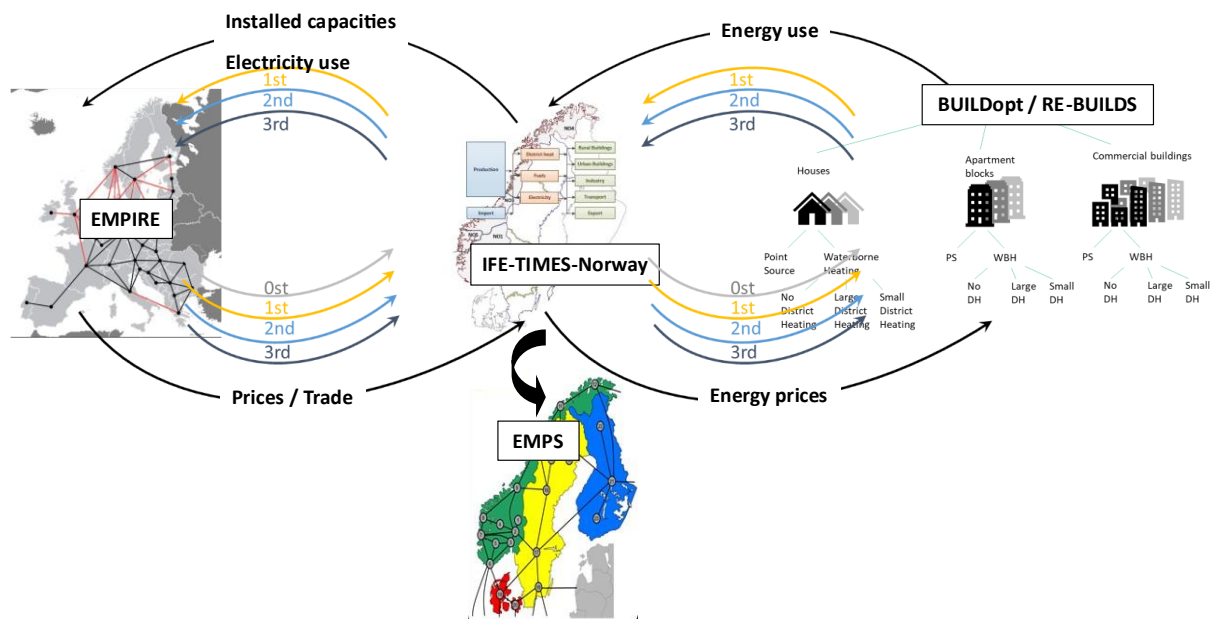


Figure 26. Schematics of the linking between the models, showing the bi-directional interdependencies with the respective key variables, and the iterations needed for their convergence (Source: [4]).

Flexible energy demand responds to the price signal, which in turn is affected by end-use flexibility when this is applied in large scale. This creates a need to iterate between these variables until convergence is achieved: this is a *full linking* between the models. The full linking revealed being a time-consuming and computationally heavy process. Nevertheless, it was found that three iterations were enough, in general, to reach a satisfying convergence of the key variables.

Trying to avoid the complexity of a full linking justified adopting a predefined tactic on how to use flexibility, also known as heuristics. The heuristics adopted was that of flattening the load profile, as described in the *Flat profile* objective in §2.1.1. The hypothesis behind the flat profile optimization is that this may be a good heuristic for the full linking, because peak shaving and valley filling are typically desired outcomes of energy flexible demand. The hypothesis was that the load profiles resulting from the full linking, after convergence, would have tended towards the flat profile.

Discussing the results of the Flexbuild project is beyond the scope and the interest of this report. However, it is worth commenting on the aspects regarding the modality of the models linking, i.e. full linking vs. heuristics. In particular, although peak shaving and valley filling was observed to some extent in all seasons in the full linking, the hypothesis of an overall resemblance with the flat profile was not verified. However, the first iteration of the full linking revealed being a good approximation of the convergent case, suggesting that is sufficient to reduce the linking process to only one iteration. In other words, the results from the linking showed that buildings remain fundamentally price-takers, and even if energy demand becomes flexible in the entire building stock, this has only a marginal impact on the energy price formation. This was a major takeaway from the Flexbuild project and should be considered in future linking between Building-FLEXor and any energy system model.

Conclusions

This review shows that expertise available in the FME SOLAR consortium already hosts a complementary toolchain spanning resource assessment, demand modelling, PV generation, storage, system planning, market simulation and grid analysis. The portfolio is sufficient to address most questions around matching PV with demand in Norway, provided we invest in robust interfaces and shared assumptions. The most consequential leverage point is the demand side. Building-FLEXor, when driven by consistent typical-demand datasets and exposed to realistic prices and operational constraints, produces flexible load profiles that downstream models can use as inputs. However, there are limitations in model coupling, such as the asymmetry in temporal resolution (e.g., hourly in Building-FLEXor vs. representative-day in TIMES), differences in the treatment of prices and flexibility, and the need for careful iteration when performing soft-linking. These factors can introduce require disciplined interface definitions.

Key takeaways: (1) High-latitude solar resource data require careful selection and site awareness; satellite products perform best up to $\sim 65^\circ\text{N}$ but degrade with horizon complexity and snow cover, reinforcing the value of local measurements and shading information. (2) PV generation modelling needs to remain bifurcated: engineering design tools (PVsyst, PVMCase, SAM) for detailed layouts and losses, and programmatic/statistical tools (pvlib, Renewables.ninja) suitable for long term scenario analysis. (3) Demand modelling and flexibility are central: Building-FLEXor provides an adequate operational representation of thermal, storage and EV charging flexibility while safeguarding user comfort. (4) For energy-system planning, EMX and TIMES answer complementary questions (multi-year investment vs. detailed technology pathways), while PyPSA-Eur provides continental boundary conditions; JULES adds operational market fidelity in hydro-dominated contexts. (5) LV/MV integration studies can be conducted today using pandapower and PowerFactory fed by PV and demand time series from the preceding tools.

Insights from prior model linking studies indicate that full iterative coupling between building-level flexibility and system-level prices converges quickly; buildings tend to remain price-takers, so a single iteration captures most effects. Flat-profile heuristics are not a reliable substitute for optimisation with explicit comfort and device constraints.

In summary, FME SOLAR has the ingredients to analyse and operationalise the match between PV generation and energy demand. With shared datasets, standardised interfaces and a demand-centred workflow built around Building-FLEXor, the consortium can deliver credible, reproducible analyses and targeted recommendations for Norwegian conditions.

References

- [1] H. Nygard Riise, M. Moe Nygård, B. Lupton Aarseth, A. Dobler, and E. Berge, 'Benchmark of estimated solar irradiance data at high latitude locations', *Sol. Energy*, vol. 282, p. 112975, Nov. 2024, doi: 10.1016/j.solener.2024.112975.
- [2] A. Dobler, E. Berge, H. O. Hygen, S. Grini, and H. N. Riise, 'Quality control of radiation data for solar resource mapping', Norwegian Meteorological Institute. [Online]. Available: https://www.met.no/publikasjoner/met-report/met-report-2024/_/attachment/inline/5f04e262-7e7b-4fb2-a909-6878d128e2cd:2d20f84d7261f993f687b9e3cb73ff56f3592a42/MET-report-8-2024.pdf
- [3] M. D. Benjamin *et al.*, 'FLEXor – The development of simulation and optimization models for energy-flexible operation in the built environment', SINTEF Academic Press, 2025. Accessed: Nov. 20, 2025. [Online]. Available: <https://sintef.brage.unit.no/sintef-xmlui/handle/11250/3213394>
- [4] I. Sartori *et al.*, 'Flexbuild final report. The value of end-use flexibility in the future Norwegian energy system', SINTEF akademisk forlag, 103, 2023. Accessed: Nov. 20, 2025. [Online]. Available: <https://hdl.handle.net/11250/3097899>
- [5] H. Kauko, B. M. Delgado, S. Backe, and I. Sartori, 'Reducing electricity demand and enhancing heat supply flexibility through energy efficiency and district heating', *Energy*, vol. 322, p. 135310, May 2025, doi: 10.1016/j.energy.2025.135310.
- [6] S. Backe, B. M. Delgado, M. Askeland, H. T. Walnum, Å. se L. Sørensen, and I. Sartori, 'Impact of activating energy demand flexibility in the building stock: A case study of Norway as a highly electrified country in the European power market', *Energy*, vol. 318, p. 134688, Mar. 2025, doi: 10.1016/j.energy.2025.134688.
- [7] J. Smertinas, H. T. Walnum, B. Ludvigsen, Å. L. Sørensen, I. Sartori, and H. Madsen, 'Building-Profet: A Scalable Data-Driven Modelling Approach for Estimating Heat and Electricity Load Profiles of Buildings', Jul. 03, 2025, *Social Science Research Network, Rochester, NY*: 5338245. doi: 10.2139/ssrn.5338245.
- [8] S. Pfenninger and I. Staffell, 'Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data', *Energy*, vol. 114, pp. 1251–1265, Nov. 2016, doi: 10.1016/j.energy.2016.08.060.
- [9] B. M. Delgado, H. T. Walnum, and I. Sartori, 'Heat Pump, solar PV and Battery Systems Modelling The development of simulation and optimization models for energy-flexible operation in the built environment', Memo 49, 2023.
- [10] B. M. Delgado, M. Bagle, Å. L. Sørensen, K. Byskov, and I. Sartori, 'Electric Vehicle Modelling', Memo 43, 2022.
- [11] Å. L. Sørensen, K. B. Lindberg, I. Sartori, and I. Andresen, 'Residential electric vehicle charging datasets from apartment buildings', *Data Brief*, vol. 36, p. 107105, Jun. 2021, doi: 10.1016/j.dib.2021.107105.
- [12] Å. L. Sørensen, B. M. Delgado, and I. Sartori, 'Average EV charging profiles for residential and commercial buildings'. Outcome of work within the Research Center on Zero Emission Neighbourhoods in Smart Cities (FME ZEN), WP4 Energy flexible neighbourhoods, 2025.
- [13] 'PVsyst | Photovoltaic system simulation software products', PVsyst | Photovoltaic software. Accessed: Nov. 20, 2025. [Online]. Available: <https://www.pvsyst.com/en/>
- [14] 'Renewables.ninja'. Accessed: Nov. 20, 2025. [Online]. Available: <https://www.renewables.ninja/>
- [15] I. Staffell and S. Pfenninger, 'Using bias-corrected reanalysis to simulate current and future wind power output', *Energy*, vol. 114, pp. 1224–1239, Nov. 2016, doi: 10.1016/j.energy.2016.08.068.
- [16] M. M. Rienecker *et al.*, 'MERRA: NASA's Modern-Era Retrospective Analysis for Research and Applications', *J. Clim.*, vol. 24, no. 14, pp. 3624–3648, Jul. 2011, doi: 10.1175/JCLI-D-11-00015.1.

- [17] R. Müller, U. Pfeifroth, C. Träger-Chatterjee, J. Trentmann, and R. Cremer, 'Digging the METEOSAT Treasure—3 Decades of Solar Surface Radiation', *Remote Sens.*, vol. 7, no. 6, pp. 8067–8101, Jun. 2015, doi: 10.3390/rs70608067.
- [18] R. Müller, U. Pfeifroth, C. Träger-Chatterjee, R. Cremer, J. Trentmann, and R. Hollmann, 'Surface Solar Radiation Data Set - Heliosat (SARAH) - Edition 1'. EUMETSAT Satellite Application Facility on Climate Monitoring (CM SAF), p. 3.6 TiB, Feb. 25, 2015. doi: 10.5676/EUM_SAF_CM/SARAH/V001.
- [19] L. M. Engan, 'Load Shifting Potential of Local Energy Communities with Seasonal Thermal Energy Storage', Master thesis, Norwegian University of Life Sciences, 2024. Accessed: Nov. 20, 2025. [Online]. Available: <https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/3148036>
- [20] 'Welcome - System Advisor Model - SAM'. Accessed: Nov. 20, 2025. [Online]. Available: <https://sam.nrel.gov/>
- [21] 'PVcase', PVcase. Accessed: Nov. 20, 2025. [Online]. Available: <https://pvcase.com/>
- [22] 'PVGIS data sources & calculation methods - The Joint Research Centre: EU Science Hub'. Accessed: Nov. 20, 2025. [Online]. Available: https://joint-research-centre.ec.europa.eu/photovoltaic-geographical-information-system-pvgis/getting-started-pvgis/pvgis-data-sources-calculation-methods_en
- [23] 'Historic Solar Irradiance Data (TGY & Sites)', SolarAnywhere. Accessed: Nov. 20, 2025. [Online]. Available: <https://www.solaranywhere.com/products/solaranywhere-data/>
- [24] pvlib, 'pvlib-python/docs/sphinx/source/index.rst at v0.13.1 · pvlib/pvlib-python', GitHub. Accessed: Nov. 20, 2025. [Online]. Available: <https://github.com/pvlib/pvlib-python/blob/v0.13.1/docs/sphinx/source/index.rst>
- [25] K. S. Anderson, C. W. Hansen, W. F. Holmgren, A. R. Jensen, M. A. Mikofski, and A. Driesse, 'pvlib python: 2023 project update', *J. Open Source Softw.*, vol. 8, no. 92, p. 5994, Dec. 2023, doi: 10.21105/joss.05994.
- [26] L. Hellemo, E. F. Bødal, S. E. Holm, D. Pinel, and J. Straus, 'EnergyModelsX: Flexible Energy Systems Modelling with Multiple Dispatch', *J. Open Source Softw.*, vol. 9, no. 97, p. 6619, May 2024, doi: 10.21105/joss.06619.
- [27] E. F. Bødal *et al.*, 'Hydrogen for harvesting the potential of offshore wind: A North Sea case study', *Appl. Energy*, vol. 357, p. 122484, Mar. 2024, doi: 10.1016/j.apenergy.2023.122484.
- [28] J. Hörsch, F. Hofmann, D. Schlachtberger, and T. Brown, 'PyPSA-Eur: An open optimisation model of the European transmission system', *Energy Strategy Rev.*, vol. 22, pp. 207–215, Nov. 2018, doi: 10.1016/j.esr.2018.08.012.
- [29] M. Victoria, E. Zeyen, and T. Brown, 'Speed of technological transformations required in Europe to achieve different climate goals', *Joule*, vol. 6, no. 5, pp. 1066–1086, May 2022, doi: 10.1016/j.joule.2022.04.016.
- [30] E. Huse, 'Exploring the Impact of Dynamic Line Rating on the Norwegian Transmission System using Historical Data', Master thesis, NMBU, 2025. [Online]. Available: <https://hdl.handle.net/11250/5274215>
- [31] M. Saleem, 'Exploring Offshore Wind Development in Norway Through Heuristic Modelling', Master thesis, NMBU, 2024. [Online]. Available: <https://hdl.handle.net/11250/3154057>
- [32] V. Malum, 'Large-Scale Optimization of the Nordic Power System in Connection with Statnett's Grid Development Plans using PyPSA', Master thesis, NMBU, 2024. [Online]. Available: <https://hdl.handle.net/11250/3188543>
- [33] M. Parzen *et al.*, 'PyPSA-Earth. A new global open energy system optimization model demonstrated in Africa', *Appl. Energy*, vol. 341, p. 121096, Jul. 2023, doi: 10.1016/j.apenergy.2023.121096.
- [34] H. Eide, 'The Impact of eCooking in the Tanzanian Power System: Modelling eCooking transitions using PyPSA- Earth', Master thesis, NMBU, 2025. [Online]. Available: <https://hdl.handle.net/11250/3209207>

- [35] 'IEA-ETSAP | Times'. Accessed: Nov. 21, 2025. [Online]. Available: <https://iea-etsap.org/index.php/etsap-tools/model-generators/times>
- [36] Kristina Haaskjold, Eva Rosenberg, P. M. S. Seljom, Miguel Chang, and Stine Fleischer Myhre, 'Documentation of IFE-TIMES-Norway v4', IFE - Institute for Energy Technology, IFE/E-2024/004, Sep. 2024.
- [37] NVE, 'GitHub - NVE/JulES: JulES is a fundamental energy market simulation model for operational planning, that uses TuLiPa as building blocks.', GitHub. Accessed: Nov. 21, 2025. [Online]. Available: <https://github.com/NVE/JulES>
- [38] 'FRAM'. Accessed: Nov. 21, 2025. [Online]. Available: <https://nve.github.io/fram/>
- [39] NVE, 'GitHub - NVE/TuLiPa: TuLiPa is a modular and extensible system for working with linear programming (LP) problems for use in energy market modelling.', GitHub. Accessed: Nov. 21, 2025. [Online]. Available: <https://github.com/NVE/TuLiPa>
- [40] 'About pandapower', pandapower. Accessed: Nov. 21, 2025. [Online]. Available: <https://www.pandapower.org/about/>
- [41] J.-P. Beck, J. Reinhard, K. Kamps, J. Kupka, and C. Derksen, 'Model experiments in operational energy system analysis: Power grid focused scenario comparisons', *Renew. Sustain. Energy Rev.*, vol. 164, p. 112550, Aug. 2022, doi: 10.1016/j.rser.2022.112550.
- [42] L. Thurner *et al.*, 'Pandapower—An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems', *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018, doi: 10.1109/TPWRS.2018.2829021.
- [43] N. S. Moe and M. Nærlund, 'Impacts on Norwegian LV Distribution Grid from EU's Solar Rooftop Initiative', Master thesis, University of Agder, 2025. Accessed: Nov. 21, 2025. [Online]. Available: <https://uia.brage.unit.no/uia-xmlui/handle/11250/3205702>
- [44] 'PowerFactory - DIgSILENT'. Accessed: Nov. 21, 2025. [Online]. Available: <https://www.digsilent.de/en/powerfactory.html>
- [45] T. K. Vrana, E. S. Aas, T. I. Reigstad, and O. Mo, 'Impact of present and future HVDC links on the nordic power grid', presented at the 13th IET International Conference on AC and DC Power Transmission (ACDC 2017), 2017. doi: 10.1049/cp.2017.0056.
- [46] I. Sartori *et al.*, 'Flexbuild annual report. 2 - Technical report with results analysis', SINTEF Academic Press, Oslo, 87, 2022. [Online]. Available: <https://hdl.handle.net/11250/2985359>
- [47] S. K. Lien, N. H. Sandberg, K. B. Lindberg, E. Rosenberg, P. Seljom, and I. Sartori, 'Comparing model projections with reality: Experiences from modelling building stock energy use in Norway', *Energy Build.*, vol. 268, p. 112186, Aug. 2022, doi: 10.1016/j.enbuild.2022.112186.
- [48] N. H. Sandberg *et al.*, 'Energisparepotensialet i bygg fram mot 2030 og 2050 – Hva koster det å halvere energibruken I bygningsmassen?', SINTEF akademisk forlag, 75, 2023. Accessed: Nov. 21, 2025. [Online]. Available: <https://sintef.brage.unit.no/sintef-xmlui/handle/11250/3083980>