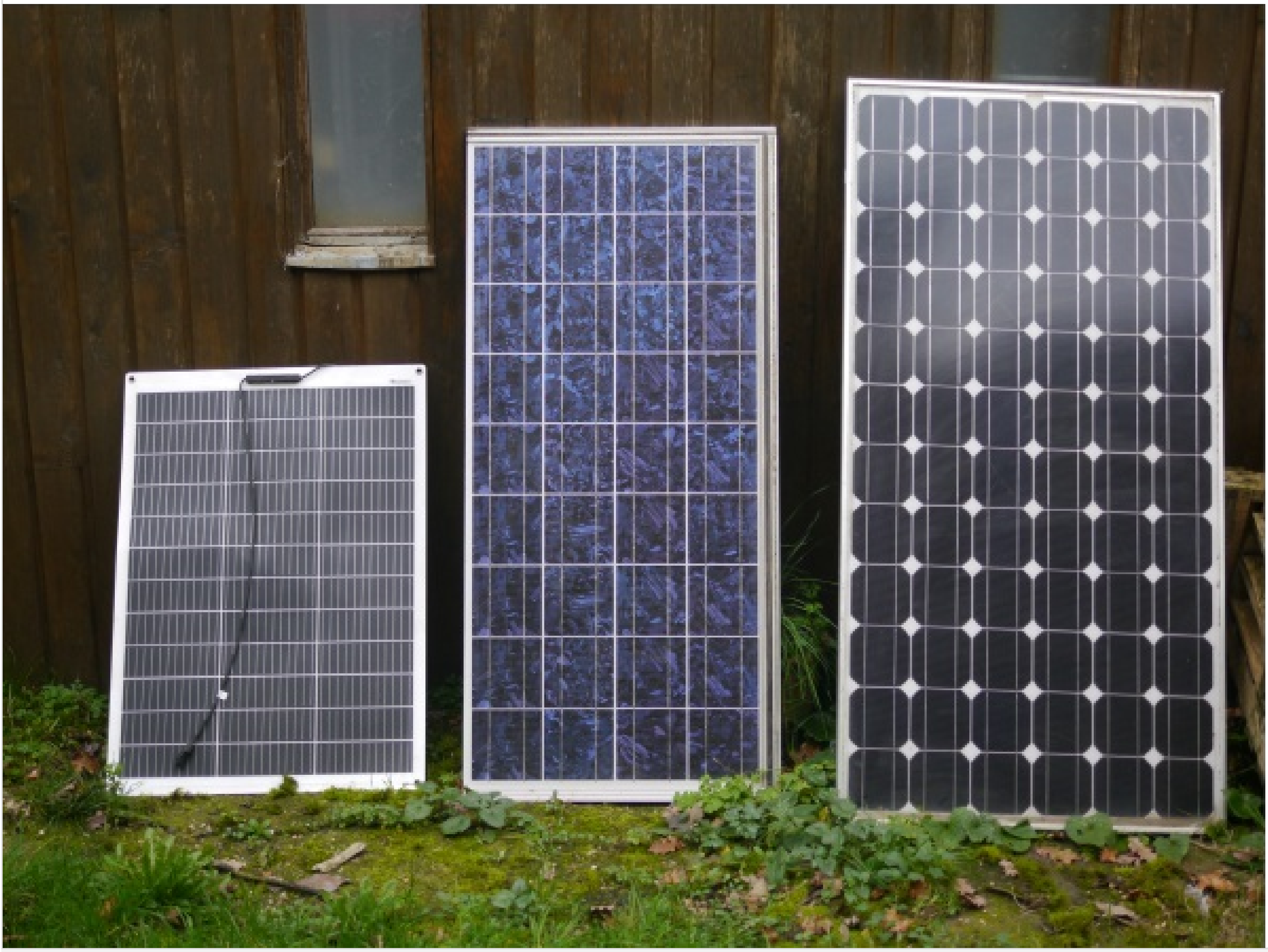
Offgrid photovoltaic installation sizing

Aurelpere



https://wiki.lowtechlab.org/wiki/Dimensionner_une_installation_photovolta%C3%AFque_autonome/en

Dernière modification le 03/08/2024

Difficulté **Moyen**

Durée **1 heure(s)**

Coût **0 EUR (€)**

Description

Tools for sizing an offgrid photovoltaic installation

Sommaire

Sommaire

Introduction

Most photovoltaic domestic installations are today
plugged to the grid (erdf/enedis in france), whether
on commercial packs called "surplus injection", or
"total injection".
With old electric meter, it is still possible
to "run backwards" the electric meter when
the pannels produce (a bit like if your smart electric meter like linky in France was subtracting
from your consumption what your pannels produce whatever
the time the production happens)
Being plugged to the electric grid is very useful because it avoids
the need to store the electricity produced.
However, wether it be for societal evolution reasons (let allow us
the right to dream to other urbanistic models where ecological autonomous sites
without needs for infrastructure are encouraged), or for
natural constraints reasons, or by choice, we can wish
to be network independant, 100% autonomous in electric energy.
I have initially made a bit of code for correctly sizing
a mobilhome i wanted to make offgrid,
and connect a fridge and a freezer (which necessitate a
constant power, ie without blackout)
The INES already has these tools here:
http://ines.solaire.free.fr/pvisole_1.php
and here:
https://autocalsol.ines-solaire.org/etude/localisation/
However, the hypothesis are 1kWc produces 1kWh in winter
for the first link et we only have mean average for the second link.
In addition, we do not have possibility to "data-test" the number of days
of blackout based on the sizing chosen.
This tutorial allows therefore to size by "data-testing", ie
with day-to-day winter productions hypothesis,
less "meaned", and allows to test the sizing chosen
less costly in storage capacity (still expensive in 2024).
Considered the number of patents declared in the
electricity storage these past 15 years, and considered
geopolitical evolutions in the brics, it is likely
that we end up with very low elecricity storage costs
in the forthcoming years and the optimizing algorithm
to fine tune sizing at the lower cost storage will probably
be less economically relevant in the years
to come.
But a storage a very low cost would dramatically challenge
the petro dollar system, so we have probably time to see it coming ;)
Any way, used batteries produce quite a few
waste, and that would still be interesting to have this piece of logic
to size more precisely to be in a low tech
perspective and avoiding oversizing when it is possible
to have seldom blackout episodes.
Interactive web demo here:
https://vpn.matangi.dev/sun

Dimensionner_une_installation_photovolta_que_autonome_jrc_a_u_t_o_n_o_m_i_e.ods

Étape 1 - Needs evaluation

Firstly we need to estimate your electric consumption
To do that, we need to take into account the correct temporal scales:
mean consumption on the one hand, but also daily variations (consumtion variations
according to day time) and seasonal variations (consumption in winter and in summer for example).
This exercise is not only interesting for sizing adequately the production,
but also to think how to reduce your consumption. For example, it is absurd to heat up
with electric heater if your main source of energy is photovoltaic.
Another example is the water heater that consumes consequent quantities of energy
which are relatively linear to the number of hot showers you take
and that can question the relevant source of energy to heat up your water or the number of showers
adequate for a non fossil production (we can usefully wonder
how to switch to a thermic solar source of energy and /or heat up with wood or biogaz)
NB: for a shower of 10min at 40°C, you approximately need 3kWh, so approximately the biomethane generated daily by 100 hen's poo (order of magnitude)
To measure your consumption, you can use your electric meter.
Thus, you can measure your daily consumption but alos the horary consumption
of every electrical device (the simpler is to remove the electric fuse of all devices
you dont want to measure)

Étape 2 - Storage sizing

Offgrid, the path the electrons take after the photovoltaic pannels have made them move is the path that go
to less resistance. Thus, if the electrons must "choose" between a running electric device
and the storage batteries, they will go to power the electric device.
This consideration can be important if you have an electric consumption important by day (when the pannels produce)
but in most cases, the daily production variations and the horary difference between the production and
the consumption imply we size the storage based on the maximul daily need.
Example: daily need of 4kWh
We take the maximum discharge percentage accepted by the batteries based on its storage technology.
For example 80% for lithium ion batteries.
Exemple: (daily needs)/(max discharge percentage)=(4)/(0,8)=5kWh
NB: the battery capacities are often expressed in Ah. To obtain the Ah from the capacity
expressed in kWh, simply divide the Wh by the storage voltage. For example in 24V:
500Wh/24V=208Ah
To size the storage correctly based on the maximum daily need, we will then
seek to know how many days without sun the storage must hold your maximum daily needs
To find the maximum number off days without sun, you can either measure yourself the sunlight (see
my other tutorial here Mesure de l'ensoleillement-luminosité avec un ordinateur monocarte (raspberry-orangepi),
either use the downloadable data from european jrc modelisation pvgis which are the reference data in the industry (warning, it is a modelisation,
Edit 27/02: after verification, the date comes from horary satellites photos, so empirical data as an input in particular for the clouds, but the hypothesis of the model regarding the quantity of sunlight going through the clouds are not very well explicited, see https://joint-research-centre.ec.europa.eu/photovoltaic-geographical-information-system-pvgis/getting-started-pvgis/pvgis-data-sources-calculation-methods_en?prefLang=fr)
To download jrc data, type this command in your linux terminal:

```
wget --max-redirect=10 -O output.csv "https://re.jrc.ec.europa.eu/api/v5_2/seriescalc?lat=44.203142&lon=0.616363&loss=14&angle=45&aspect=0&startyear=2005&endyear=2020&pvcalculation=1&peakpower=1&pvtechchoice=crystSi&browser=0&outputformat=csv"
```

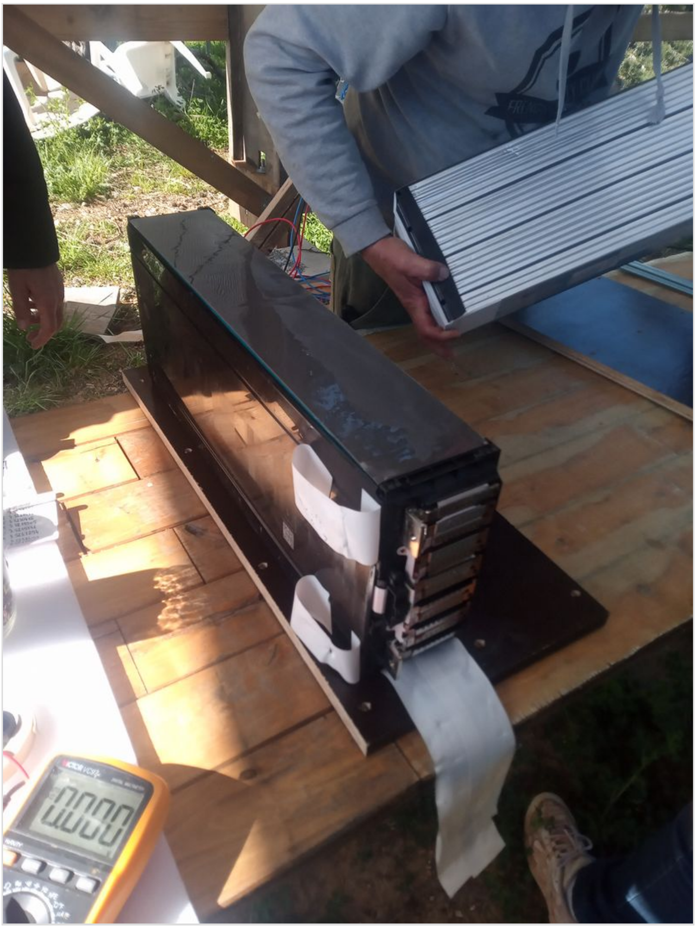
The parameters lat= and lon= allow to adjust latitude and longitude
startyear= and endyear= are the time limits of the considered period (here 15 years between 2005 and 2020)
peakpower= corresponds to the photovoltaic power in kWc (here 1kWc)
We then use a piece of python/pandas code to obtain the maximum number of consecutive days for 1kWc to produce respectively 1kWh, 2kWh and 3kWh

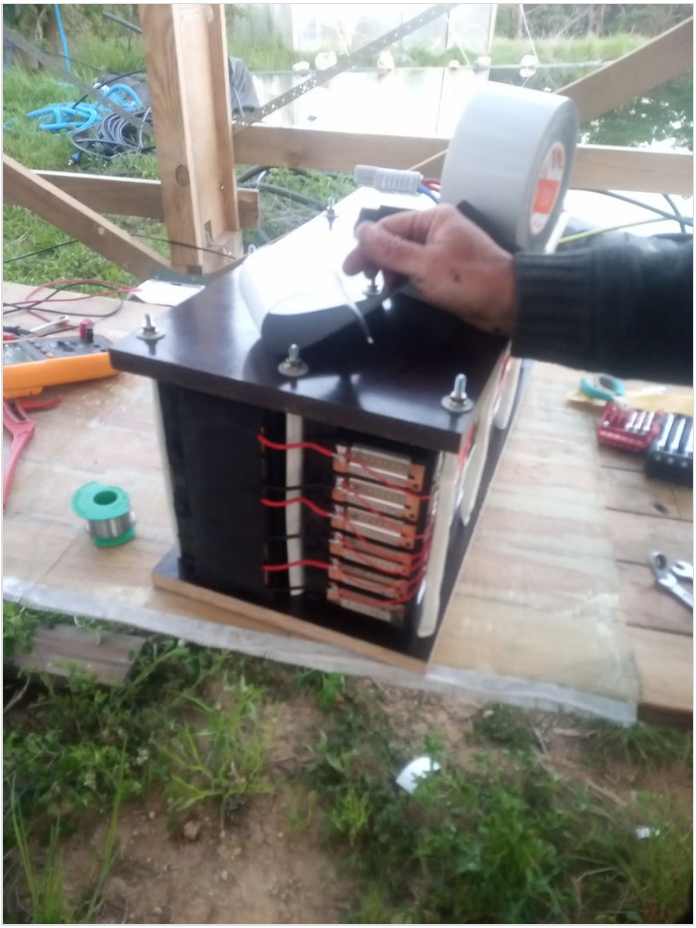
```
import pandas as pd
df = pd.read_csv('output.csv', skiprows=10, skipfooter=11, sep=',', engine='python')
df['time']=df['time'].astype(str)
df['time']=pd.to_datetime(df['time'],format="%Y%m%d:%H%M")
df=df.set_index('time')
daily_data = df.resample('D').sum()

max_streak = 0
current_streak = 0
current_sum=0
target=1000
streaks=[]
for value in daily_data['P']:
    if current_sum <= target:
        current_sum+=value
        current_streak += 1
        max_streak = max(max_streak, current_streak)
    else:
        streaks.append(current_streak)
        current_sum=0
        current_streak=0
print("maximum nb of consecutive days for 1kWc to produce 1kWh: "+str(max_streak)+" j")
print("nombre d'occurences: "+ str(streaks.count(max_streak)))
print("mean nb of consecutive days for 1kWc to produce 1kWh: "+str(sum(streaks)/len(streaks))+" j")

max_streak = 0
current_streak = 0
current_sum=0
target=2000
streaks=[]
for value in daily_data['P']:
    if current_sum <= target:
        current_sum+=value
        current_streak += 1
        max_streak = max(max_streak, current_streak)
    else:
        streaks.append(current_streak)
        current_sum=0
        current_streak=0
print("maximum nb of consecutive days for 1kWc to produce 2kWh: "+str(max_streak)+" j")
print("number of occurences: "+ str(streaks.count(max_streak)))
print("mean nb of consecutive days for 1kWc to produce 2kWh: "+str(sum(streaks)/len(streaks))+" j")

max_streak = 0
current_streak = 0
current_sum=0
target=3000
streaks=[]
for value in daily_data['P']:
    if current_sum <= target:
        current_sum+=value
        current_streak += 1
        max_streak = max(max_streak, current_streak)
    else:
        streaks.append(current_streak)
        current_sum=0
        current_streak=0
print("maximum nb of consecutive days for 1kWc to produce 3kWh: "+str(max_streak)+" j")
print("number of occurences: "+ str(streaks.count(max_streak)))
print("mean nb of consecutive days for 1kWc to produce 3kWh: "+str(sum(streaks)/len(streaks))+" j")
```

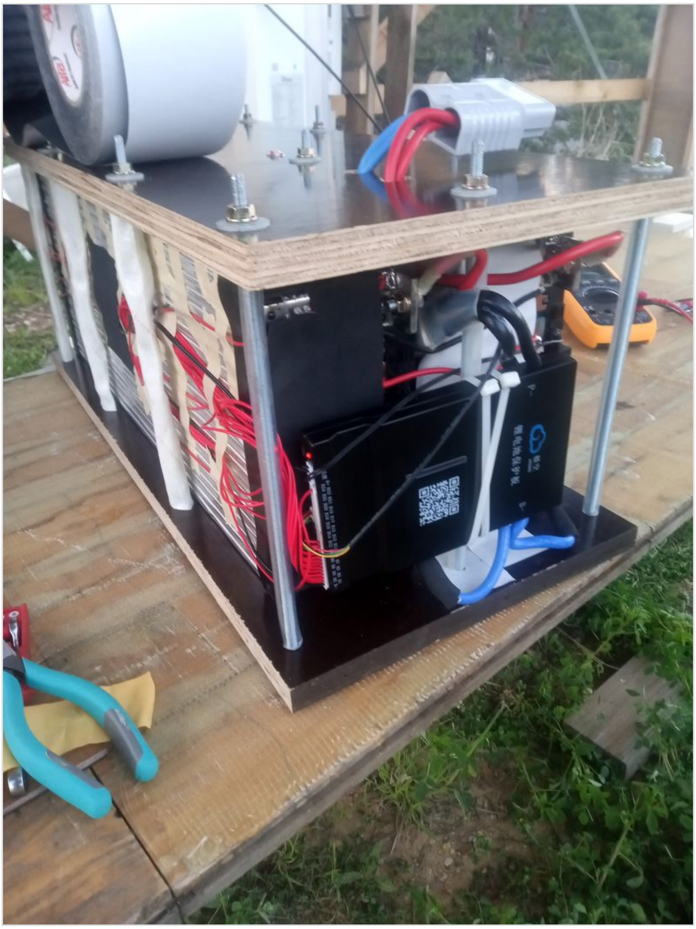
In our case, we get:
maximum number of consecutive days for 1kWc to produce 1kWh: 4d
number of occurences : 1
mean number of consecutive days for 1kWc to produce 1kWh: 1.1124367317425885d
maximum number of consecutive days for 1kWc to produce 2kWh: 5d
number of occurences : 3
mean number of consecutive days for 1kWc to produce 2kWh: 1.2801404603979711d
maximum number of consecutive days for 1kWc to produce 3kWh: 7d
number of occurences : 1

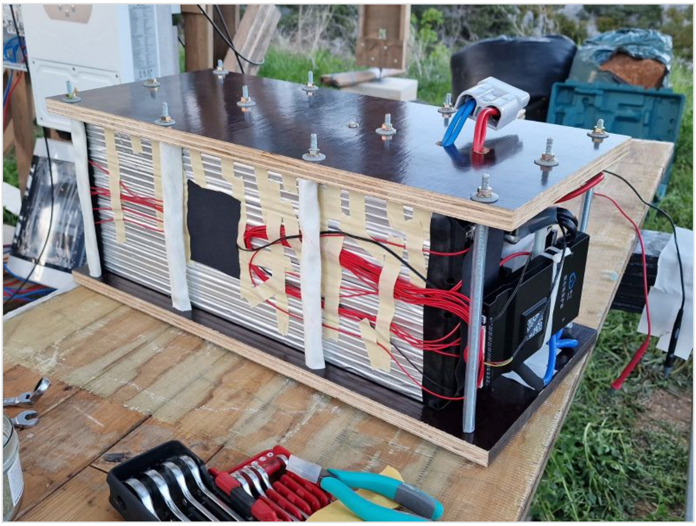
mean number of consecutive days for 1kWc to produce 3kWh: 1.4647827920708563d

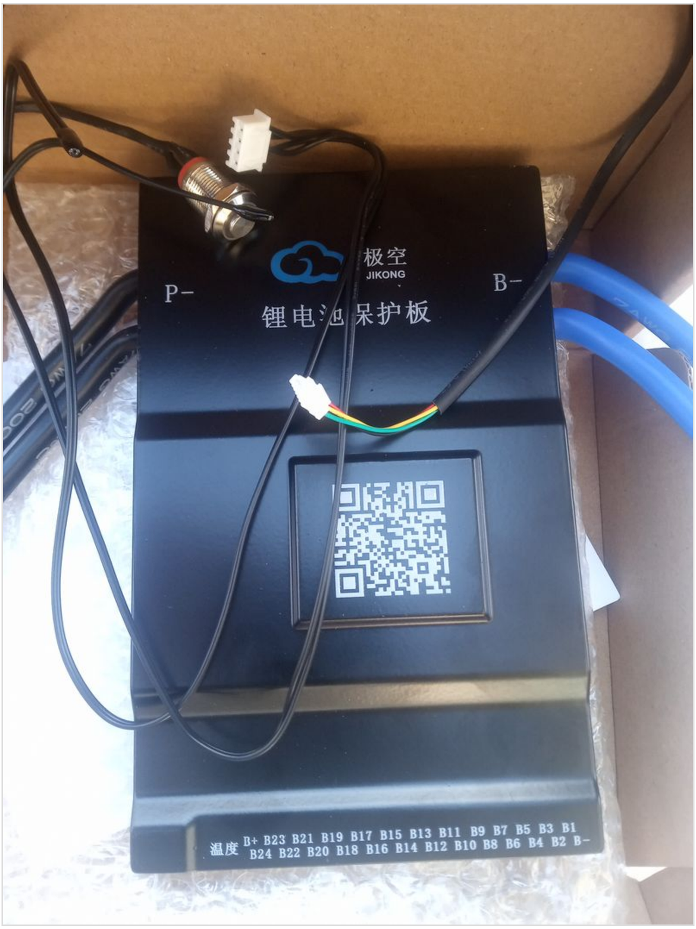
What are these numbers telling us?
That the maximum number of days with a low sunlight vary from 4 to 7d based on the level of sunlight hoped
But alos that the number of occurences are relatively low: 1 to 3 occurences only of these episodes with very few sunlight and production in 15 years.
(if you are not entirely satisfied with the modelised data considered your experience of photovoltai production based on your experience of the weather, and my hypothesis is
that solar pannels produce more that the jrc modelised data give us, i encourage you to
use empirical data using my other tutorial here : Mesure de l'ensoleillement-luminosité avec un ordinateur monocarte (raspberry-orangepi))
And finally that the mean number of days for 1kWc to produce 1kWh to 3kWh varies from 1 to 1,5d
We then have a choice:
-either oversize the storage to hold 4d needs to face these seldom very low sunlight episodes : 20kWh of batteries for 4kWh needs
-either oversize the storage to hold 1 to 2d to and have a fossil fuel generator on the side: 5 to 10kWh of batteries for 4kWh needs
to avoid a costly oversizing for vey seldom occasions
-either, if criticity of needs is not too high, size the storage to hold 1 to 2d and
cope wihtout electricity when there ar long episodes without sun: 5 to 10kWh of batteries for 4kWh needs
-(or also play on the production side with oversizing production -a produced kWh is much cheaper than a stored kWh in 2024- in the next stage to fully charge the batteries the first day after sunlightless episodes)
We are in may 2024, et we just leave the informal industrial consensus of 1€/Wh for high performance batteries.
If you want to size your storage "at scale" after you have reduced your electric needs, the best
is still to reuse car batteries.
See photos from a lowtech workshop at astrolab for lithium ion car battery reuse.
In may 2024, we can get 14kWh for 3k€ with thise kind of reuse.
It is still difficul to get away with BMS (Battery management system) for security norms,
but we can find some on aliexpress (see photos for references).

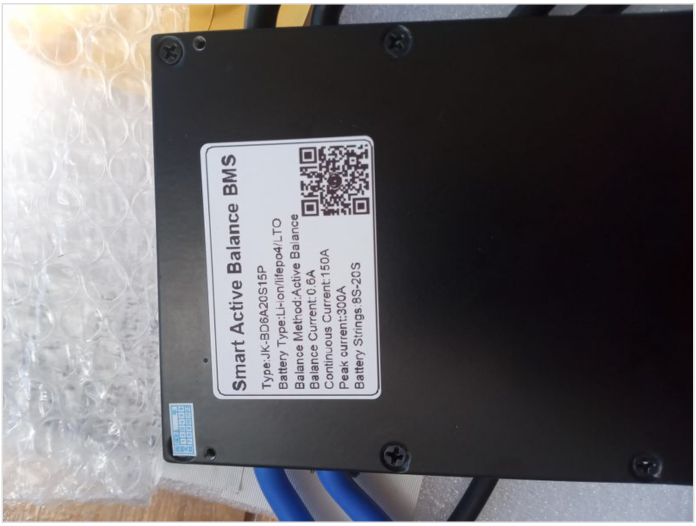









极空
JIKONG
P−
B−
温度 B+ B23 B21 B19 B17 B15 B13 B11 B9 B7 B5 B3 B1
B24 B22 B20 B18 B16 B14 B12 B10 B8 B6 B4 B2 B−



Étape 3 - Production sizing

We suppose we size the batteries at 10kWh for 4kWh needs -so 2d of autonomy-
We can size production several ways: 1. So as batteries are entirely charged on an mean interval in december/january (sunless months of the year) 2.So as batteries are entirely charged in one sunny mean day in december/january
We notice sizing based on winter months allow to produce enough to provide with electri needs all year long, but imply an overproduction in summer that would be useful to reuse (for example by hyrdolyzing water to produce hydrogen and then transform it in methane, cf scenario negawatt 2011) in particular if this type of installation was to be generalized.
We will reuse the previous piece of code and adapt it to find the same stats as before but only in december and january.

```
import pandas as pd
df = pd.read_csv('output.csv', skiprows=10, skipfooter=11, sep=',', engine='python')
df['time']=df['time'].astype(str)
df['time']=pd.to_datetime(df['time'],format="%Y%m%d:%H%M")
df=df.set_index('time')
daily_data = df.resample('D').sum()
dec_jan_data = daily_data[(daily_data.index.month == 12) <span>|</span> (daily_data.index.month == 1)]
print("Mean daily production in january and december")
print(str(dec_jan_data['P'].mean()/1000)+" kWh")
max_streak = 0
current_streak = 0
current_sum=0
target=1000
streaks=[]
for value in dec_jan_data['P']:
    if current_sum <= target:
        current_sum+=value
        current_streak += 1
        max_streak = max(max_streak, current_streak)
    else:
        streaks.append(current_streak)
        current_sum=0
        current_streak=0
print("maximum nb of consecutive days for 1kWc to produce 1kWh: "+str(max_streak)+" j")
print("number of occurences: "+ str(streaks.count(max_streak)))
print("mean nb of consecutive days for 1kWc to produce 1kWh: "+str(sum(streaks)/len(streaks))+" j")

max_streak = 0
current_streak = 0
current_sum=0
target=2000
streaks=[]
for value in dec_jan_data['P']:
    if current_sum <= target:
        current_sum+=value
        current_streak += 1
        max_streak = max(max_streak, current_streak)
    else:
        streaks.append(current_streak)
        current_sum=0
        current_streak=0
print("maximum nb of consecutive days for 1kWc to produce 2kWh: "+str(max_streak)+" j")
print("number of occurences: "+ str(streaks.count(max_streak)))
print("mean nb of consecutive days for 1kWc to produce 2kWh: "+str(sum(streaks)/len(streaks))+" j")

max_streak = 0
current_streak = 0
current_sum=0
target=3000
streaks=[]
for value in dec_jan_data['P']:
    if current_sum <= target:
        current_sum+=value
        current_streak += 1
        max_streak = max(max_streak, current_streak)
    else:
        streaks.append(current_streak)
        current_sum=0
        current_streak=0
print("maximum nb of consecutive days for 1kWc to produce 3kWh: "+str(max_streak)+" j")
print("number of occurences: "+ str(streaks.count(max_streak)))
print("mean nb of consecutive days for 1kWc to produce 3kWh: "+str(sum(streaks)/len(streaks))+" j")
```

In our case, we get:
Mean daily production in january and december:1.9924362701612903 kWh
Maximum number of consecutive days for 1kWc to produce 1kWh: 4d
number of occurences: 1
mean number of consecutive days for 1kWc to produce 1kWh: 1.3539192399049882d
Maximum number of consecutive days for 1kWc to produce 2kWh: 5d
number of occurences: 3

mean number of consecutive days for 1kWc to produce 2kWh: 1.8022598870056497d
Maximum number of consecutive days for 1kWc to produce 3kWh: 7d
number of occurences: 1
mean number of consecutive days for 1kWc to produce 3kWh: 2.263157894736842d
1. When sizing on the mean 1kWh interval : 1kWc produces on average 1kWh in 1,35d, so 0,74kWh scaled to a 1 day interval, so a sized production of 10kWh/0,74=13,5kWc When sizing on the mean 2kWh interval : 1kWc produces on average 2kWh in 1,8d, so 1,11kWh scaled to a 1 day interval, so a sized production of 10kWh/1,11=9kWc When sizing on the mean 3kWh interval : 1kWc produces on average 3kWh in 2,26d, so 1,33kWh scaled to a 1 day interval, so a sized production of 10kWh/1,33=7,5kWc
2. When sizing on a mean day, 1kWc produces on average 1,99kWh per day, so a production need of 10kWh/1,99=5kWc
We notice that the reference ines data for production of 1kWc in december corresponds to 1kWc===1kWh approximately (https://autocalsol.ines-solaire.org/etude/localisation/ et http://ines.solaire.free.fr/pvisole_1.php pour site isolés)
We can here talk about more or less conservative hypothesis, ie hypothesis more or less "careful", in order for the battries to be always charged in winter. We notice that whatever the hypothesis to satisfy in a continuous way the needs, includin winter, we have overproduction in summer. It is then relavant to use overproduction reuse system in absence of grid in which we can inject the surplus (for example by hyrdolyzing water to produce hydrogen and then transform hydrogen in methane, cf scenario negawatt 2011).

Étape 4 - Regulation sizing

Basics recall: plugging in serie (+ on - and + on -): we add voltage and we keep same amperage plugging in paralell (+ on + and - on -): we add amperage and we keep same voltage
same for batteries: to keep in paralell to keep same voltage
The first problematic for lowtech offgrid photovoltaic is sizing
You can use the libreoffice calc sheet attached in this tutorial to do diy sizing
Previous stages have permitted to compute batty storage size based on jrc data and the average production of 1 kWc in winter

You can read the characteristics of your solar pannels you were give or found on second hand website:
-peak power: they add up to obtain the necessary power found when sizing production
-voltage: 12V, 24V or 48V. See basics recall for additions. You can also make a mixed serie/paralell system to adjust voltage
-amperage: variable according to the models but generally lower than 10A. See basics recall for addition. You can also make mixed serie/paralelle system to adjust
To recharge batteries, in principle, if you connect your pannels directly on a battery, you only need your voltage to be the same as the battery for your pannel charge the battery when producing energy
There is an important device to recall to charge correctly your batteries:
The charging regulator (nb: generally inverter said "hybrid", ie with an output for battery and one output for grid, include a charging regulator, mostly mppt, that conditions the battery types that can be plugged)
3 types exist: TOR (Tout ou rien) MPPT (maximum power point tracking) PWM (Pulse Width Modulation)
They are made of DC/DC adapter (direct current to direct current) and a circuit breaker. The mppt also includes an impedence adapter (resistance to adapt amperage injected into the batteries). The mppt generally accept higher nominal power, ie higher voltage and intensity.
The charging regulator allows mainly to break the circuit when the battery is fully charged by monitoring the voltage and intensity of charge and breaking the circuit if amperage and voltage values get higher than the reference (to do so the regulator stop chargin and measure the voltage at the terminals)
The mppt has an electronic "algorithm" that seeks the maximum power with its resistance adapter
If you connect several pannels and several batteries, it is recommended to have a regulator to stop charging correctly when the battery is fully charged.
The charging voltage of reference are 12V,24V and 48V.
However, the prices of the models get higher when the nominal power they accept gets higher (so it will mainly rely on amperage)
To limit current amperage of the photovoltaic production, it is useful to use pannels with a higher power which are generally at higher voltage (recall P=U*I
recall E=P*t keeps equal in a closed system).
note: if the storage system or the device connected to the pannel doesnt absorb all the produced power, and if the regulator doesnt cut the circuit, the rest of energy will be dissipated in heat.
Amperage will also depend on the storage capacity of your batteries, sized to cover your needs during the time periode defined at sizing.
The charge current is calculated by dividing by 4 or 5 the nominal capacity of the battery expressed in Ah, which should then charge in 4 to 5h. However, a battery can recharge with a lower charge current, for example nominal battery capacity divided by 20, so more slowly in 20h.
To size correctly the regulator, we can:
-either mount all the pannels in serie(which will lead to a higher voltage and it might be difficult to find an adequate regulator)
-either mount all the pannels in paralell (which will raise the amperage)
-either mount in serie/paralelle to adjust voltage and amperage to the characteristics of your regulator
There's finaly a last point to be cautious with: the trigger of the battery charge by the regulator (which triggers when the voltage of the battery goes below a treshold)
Indeed, if the power taken from the battery is too low, it is possible that the time necessary to discharge it with your daily consumption to triggers the charge will be longer than the daily sunlight exposure. The battery would then not charge during the day

In this case, your battery will only charge every other day (depending on the charge trigger treshold)
This is a parameter to take into account if your consumption is very low
The regulator has 3 phases:
bulk: the regulator lets the current pass
floating: the regulator alterns open and closed at a frequency to maintain the battery charged
In addition, it's important to take precautions because the charge of the battery can be risky
absorption (for mppt): the charging voltage rises a bit to create a potential difference to continue to charge the batterie almost full
In theory, the charge current diminishes when the battery is neraly full
Chargin on batteries in paralell or in serie on second hand batteries which have not the same amperage an voltage is theorethically risky.
Indeed you can read a bit everywhere that the resistance in the wires to make a junction
creating potential differences between the batteries producing discharges from one battery to another
creating risks of explosion, degassing for lead batteries
We must recall batteries are units of low voltage put together in serie and paralell to botain a generator with a target voltage and amperage and doing the same with full batteries is not really risky
We often talk about "battery management system" (bms) "embedded" for lithium ion batteries
In reality, the regulator is already a "bms". In theory, the embeded bms makes sure voltage and amperage of each unit of the battery is the same and balances them when needed
We can of course wonder if all this is not a way to make electric storage more expensive with vms components artificially expensive and if this is not a way to avoid reuse of batteries
It is for example surprising no bms exist to balance automatically lead acid batteries, which would make usable all car batteries disposed to sotre photovoltaic energy without risks
In any case, if you use lead bateries, use a regulator to avoid to keep on chargin your fully charged batteries (risks of hydrogen prodution) -or if you dont use batteries, size very carefully-, avoid profound discharges, and maintain your batteries at a constant temperature as much as possible

---

Étape 5 - Sizing optimisation

We will now use a piece of python code to iterate (a bit on the principle of tries and errors) by raising progressively the power of the modules, the capacity of the battery, and the number of acceptable days of blackout and testing day-to-day storage result - electric consumption + production with jrc data previously downloaded
explanations: the code is commented to explain what is being done and the names of variables are relatively explicit
For the iteration loop: we start at power0=(dailyconsumption)/2 (hypothesis winter 1kWc produces 2kWh on average) we start at battery0=(dailyconsumption)/0,8 with lithium and (dailyconsumption)/0,5 with lead
We iterate 14 times the power with 1/3*power0 kWc steps We iterate 40 times the capacity of the batteries by step of 1/2*battery0 kWh
We iterate 20 times the acceptable days of blackout with 1d steps
We start at full battery and we test each day if the battery storage minus the consumption plus the photovoltaic production leads to a blackout or allows to meet the daily consumption needs. If theres is a blackout, we stop the loop after the acceptable number of consecutive days given as a entry parameter
<u>Software requirements</u>
To use python under another os than linux, cope with being on a proprietary system
Under linux, python is generally installed and to use the python code shared in this tutorial, you just have to copy and paste the code in a text file processing.py and then enter

```
python processing.py
```

However, you have to install pandas library which is massively used in finance industries and in science, in particular for its time series and data vectorization capacities

```
To do so, here are the commands to enter in a linux debian before launching the script processing.py to be ok
```

```
sudo apt install python3 python3-venv python3-pip python-is-python3

cd ~ && python -m venv venv

cd ~ && python -m venv venv

pip install pandas
```

Think to activate virtual environment in which the pandas was installaed each time you use the script (after a reboot for example) by typing:

```
cd ~ && source venv/bin/activate
```

We are in 2024 and if you are targeted by surveillance and shackle anti eco terrorist measures like me, and as a good scientist, you would inspect your instruments before use, you can check the source code of pandas which is free software here: https://github.com/pandas-dev/pandas or can make the hypothesis that we can trust a software so massively used in the science and finance industries
Python relies on C libraries for a few basic tasks, and hack, including scientific hack, is nevre impossible, but we will put aside these consideration pro-lowtech that dont enter in the scope of this tutorial

Piece of logic coded in python: (update 8.4.24: added comments and input angle and orientation and battery discharge)

```
import pandas as pd
import sys
import os
import math
import time
# Ask user to enter latitude and longitude
x_input = input("Enter latitude: ")
y_input = input("Enter longitude: ")

# Replace comma with points
x_input = float(x_input.replace(',', '.'))
y_input = float(y_input.replace(',', '.'))

#Enter orientation (south,west,east,nord)
aspect_input=input("Enter orientation (south,west,east,nord)")
dictaspect={'est':-90,'ouest':90,'nord':180,'sud':0}
if aspect_input in dictaspect:
    aspect=dictaspect[aspect_input]
else:
    aspect=0

#Enter angle of the modules
angle_input=input("Entre module angle in °")
try:
    angle=float(angle_input)
except Exception as err:
    print(f'{err} error, used angle will be 45°')
    angle=45

# Download the data
try:
    os.system(f'wget --max-redirect=10 -O output.csv "https://re.jrc.ec.europa.eu/api/v5_2/seriescalc?lat={x_input}&lon={y_input}&loss=14&angle={angle}&
aspect={aspect}&startyear=2005&endyear=2020&pvcalculation=1&peakpower=1&pvtechchoice=crystSi&browser=0&outputformat=csv"')
except:
    print("data could not be downloaded, exiting")
    sys.exit()

df = pd.read_csv('output.csv', skiprows=10, skipfooter=11, sep=',', engine='python')
df['time']=df['time'].astype(str)
df['time']=pd.to_datetime(df['time'],format="%Y%m%d:%H%M")
df=df.set_index('time')
daily_data = df.resample('D').sum()

max_streak = 0
current_streak = 0
current_sum=0
target=1000
streaks=[]
for value in daily_data['P']:
    if current_sum <= target:
        current_sum+=value
        current_streak += 1
        max_streak = max(max_streak, current_streak)
    else:
        streaks.append(current_streak)
        current_sum=0
        current_streak=0
print("\n")
print("maximum nb of consecutive days for 1kWc to produce 1kWh: "+str(max_streak)+" j")
print("number of occurences: "+ str(streaks.count(max_streak)))
print("mean nb of consecutive days for 1kWc to produce 1kWh: "+str(sum(streaks)/len(streaks))+" j")
```

```
max_streak = 0
current_streak = 0
current_sum=0
target=2000
streaks=[]
for value in daily_data['P']:
    if current_sum <= target:
        current_sum+=value
        current_streak += 1
        max_streak = max(max_streak, current_streak)
    else:
        streaks.append(current_streak)
        current_sum=0
        current_streak=0
print("\n")
print("maximum nb of consecutive days for 1kWc to produce 2kWh: "+str(max_streak)+" j")
print("number of occurences: "+ str(streaks.count(max_streak)))
print("mean nb of consecutive days for 1kWc to produce 2kWh: "+str(sum(streaks)/len(streaks))+" j")

max_streak = 0
current_streak = 0
current_sum=0
target=3000
streaks=[]
for value in daily_data['P']:
    if current_sum <= target:
        current_sum+=value
        current_streak += 1
        max_streak = max(max_streak, current_streak)
    else:
        streaks.append(current_streak)
        current_sum=0
        current_streak=0
print("\n")
print("maximum nb of consecutive days for 1kWc to produce 3kWh: "+str(max_streak)+" j")
print("number of occurences: "+ str(streaks.count(max_streak)))
print("mean nb of consecutive days for 1kWc to produce 3kWh: "+str(sum(streaks)/len(streaks))+" j")

daily_data['P']=daily_data['P']/1000
#Production of 1kWh based on jrc data:
resultday=daily_data['P'].resample('D').sum()
resultday.index=pd.to_datetime(resultday.index,format="%Y%m%d")
print("\nMean daily production (kWh):\n", resultday.mean())
print("Minimum daily production (kWh):\n", resultday.min())
print("Maximum daily production  (kWh):\n", resultday.max())

#Calculate weekly production sums
resultweek=daily_data['P'].resample('W').sum()

# Calculate monthly production sums
resultmonth=daily_data['P'].resample('M').sum()

# Calculate quarterly production sums
resulttrim=resultday.resample('Q').sum()
resulttrim=resulttrim.rename_axis('trimestre')
print("\nQuarterly production (kWh):\n",resulttrim.to_string())

# Calculate yearly production sums
resultyear=daily_data['P'].resample('Y').sum()
print("\nYearly mean production of 1kWc (kWh):\n",resultyear.mean())

# Calculate maximum number of consecutive days without production
max_streak = 0
current_streak = 0
for value in resultday:
    if value == 0:
        current_streak += 1
        max_streak = max(max_streak, current_streak)
    else:
        current_streak = 0  # Reset the streak if the value is not zero
print(f"\nMaximum number of consecutive days without production: {max_streak}")

# Quarterly mean for each quarter
```

```
moyenne_trimestrielle_par_trimestre = resulttrim.groupby(resulttrim.index.quarter).mean()

# Quarterly minimum for each quarter
min_trimestrielle_par_trimestre = resulttrim.groupby(resulttrim.index.quarter).min()

# Quarterly maximum for each quarter
max_trimestrielle_par_trimestre = resulttrim.groupby(resulttrim.index.quarter).max()

# Imprimer les résultats
print("\nQuarterly mean for each quarter (kWh):\n", moyenne_trimestrielle_par_trimestre)
print("\nQuarterly minimum for each quarter (kWh):\n", min_trimestrielle_par_trimestre)
print("\nQuarterly maximum for each quarter (kWh):\n", max_trimestrielle_par_trimestre)

# Ask user his/her daily electricity consumption
inputelecconsoday = input("Enter daily electricity consumption (kWh): ")

# Replace comma with points
inputelecconsoday = float(inputelecconsoday.replace(',', '.'))

# Ask user lead or lithium battery
typebatterie = input("Do you wnat to use lead batteries? Type o for yes (lithium by default or if nothing is typed)")

# Initial battery sizing for 24h of autonomy
if typebatterie=="oui" or typebatterie=="o" or typebatterie=="y" or typebatterie=="yes":
    batterie0=int(math.ceil(inputelecconsoday/0.5))
    typebatterie="plomb"
else:
    batterie0=int(math.ceil(inputelecconsoday/0.8))
    typebatterie="lithium"
print(f"""Initial battery treshold with entered hypothesis and user provided data (battery {typebatterie}))
hypothesis: (daily elec consumption/0.8 with lithium and daily elec consumption/0.5 with lead)
{batterie0} kWh""")

# Maximum percentage discharge
d_input=input('Enter maximum authorized percentage discharge (20% if you want your battery doesnt get empty to less than 20%. Hit enter if you want to use the default 50% values for lead and 0% otherwise -hypothesis solid state batteries-')
try:
    d=float(d_input)/100
except Exception as err:
    if typebatterie=='plomb':
        d=0.5
    else:
        d=0

# Ask user how many blackout days are acceptable
inputjnoelec = input("Combien de jours maximum consécutifs sans electricité (ou avec un groupe electrogene) pouvez vous supporter (0 par défaut si reponse vide)")

# Ask user how many blackout days are acceptable
inputjnoelec = input("Combien de jours maximum consécutifs sans electricité (ou avec un groupe electrogene) pouvez vous supporter (0 par défaut si reponse vide)")

try:
    jnoelec=int(inputjnoelec)
except Exception as err:
    jnoelec=0
print(f"""Maximum number of consecutive days without electricity (or with a generator) taken as hypothesis
{jnoelec} j""")

# Power sizing of the modules (kWc) initially to produce enough in winter (hypothesis 2kWh per kWc per day)
puissance0=int(math.ceil(inputelecconsoday/2))

print(f"""\nInitial power treshold with user provided hypothesis and data
hypothesis : daily elec consumption/0.8 with lithium and 0.5 with lead
{batterie0} kWh""")

print(f"""\nInitial battery capacity treshold with user provided hypothesis and data
Hypothesis: daily elec consumption/0.8 with lithium and 0.5 with lead
{batterie0} kWh""")

batterie0_input = input("\n\nIf you want to correct the initial value of the battery (kWh) for iterations, enter your value, otherwise hit enter")
try:
    _=float(batterie0_input)
    batterie0=
```

```
except Exception as err:
    print(f"\ntype error or empty user value, continuing with batterie0={batterie0}kWh")

    puissance0_input = input("\n\nIf you cant to correct initial peak power (kWc) for iterations, enter your value, otherwise hit enter")
try:
    _=float(puissance0_input)
    puissance0=_
except Exception as err:
    print(f"\ntype error or empty user value, continuing with batterie0= puissance0={puissance0}kWc")

puissance0_input = input("\n\nIf you cant to correct initial peak power (kWc) for iterations, enter your value, otherwise hit enter")
try:
    _=float(puissance0_input)
    puissance0=_
except Exception as err:
    print(f"\ntype error or empty user value, continuing with batterie0= puissance0={puissance0}kWc")

# Algorithmic iterations for storage and consumption

#function check surface volume
def iter(data,consoelecday,v0,p0,joursnoelec):
    #elec is energy in the battery : initial elec=v0 (battery volume at t0)
    elec=v0
    #current_streak is the number of days of consecutive blackout
    current_streak = 0
    #listjnoelec is the list in which we register the number of days of blackout
    listjnoelec=[]
    #loop on the input date (argument of the function)
    for i in range(len(data)):
        #recupday is electricity produced at day i (power_of_1kWc*p0)
        #p0 is the peak power entered as argument of the function
        recupday=data.iloc[i]*p0
        #consoday is the consumption of day i entered as argument of the function
        consoday=consoelecday
        #energy updated with consoday and recupday
        elec=elec+recupday-consoday
        #if updated energy is higher than battery volume
        if elec>v0:
            #print("battery full")
            #number of consecutive blackout days is reinitialized
            current_streak=0
            #energy is equal to the volume of the battery
            elec=v0 #hypothese gestion du trop plein ok
            continue
        #if updated energy is lower than the max percentage of discharge and the number of consecutive days of blackout is lower than the tolerated numbe
r
        elif elec<d*v0 and current_streak<joursnoelec:
            #if the number of consecutive days of blackout is equal to zero
            if current_streak==0:
                #add current day's date and a counter to the listjnoelec list
                listjnoelec.append([data.index[i].strftime('%d-%m-%Y'),1])
            #otherwise
            else:
                #increment the counter of the last entry of listjnoelec
                listjnoelec[-1][-1]+=1
            #reinitialization of the negative value of elec
            elec=0
            #incrementation of the consecutive days of blackout
            current_streak+=1
            continue
        #if energy is lower to the max percentage of discharge and the number of consecutive days of blackout is higher than the number of tolerated days
        elif elec<d*v0 and current_streak>=joursnoelec:
            #the function returns a null tuple
            return (0,0)
        #otherwise
        else:
            #reinitialization of the number of consecutive days of blackout
            current_streak=0
    #nb of episode of blackout  = lenght of list listjnoelec
    nb_episode_no_elec=len(listjnoelec)
    #if listjnoelect is not empty
    if len(listjnoelec)!=0:
        #list of durations of blackout episodes
        list_duree_episode_no_elec=[length[1] for length in listjnoelec]
```

```
        list_duree_episode_no_elec=[length[1] for length in listjnoelec]
        #mean of this list
        duree_moy_episode_no_elec=sum(list_duree_episode_no_elec)/len(list_duree_episode_no_elec)
    #otherwise
    else:
        #mean duration equal to zero
        duree_moy_episode_no_elec=0
    #print("powers and batteries permit to meet the electricity consumption needs on the dataset")
    #return variable necessary to process results
    return (v0,p0,joursnoelec,nb_episode_no_elec,duree_moy_episode_no_elec,listjnoelec)
#hypothesis full battery at t0
elec=batterie0
resultpuissancevolume=(batterie0,puissance0,jnoelec,0)

#iteration loop
listpuissance0=[puissance0*(1+i*0.33) for i in range(0,999)]
listbatterie0=[batterie0*(1+i*0.5) for i in range(0,999)]
listresult_0blackout=[]
listresult_blackout=[]

# increment with hypothesis jnoelec=0 (k in range 0,1)
for i in range(0,14): #iteration loop on modules surfaces
    for j in range (0,i+40):#iteration loop on battery volume
        for k in range(0,1):
            resultpuissancevolume=iter(resultday,inputelecconsoday,listbatterie0[j],listpuissance0[i],k)
            #if results are not null
            if resultpuissancevolume!=(0,0):
                #add results to a list of results
                listresult_0blackout.append(resultpuissancevolume)
                #stop k loop
                break
            else:#otherwise continue
                continue
        #stop j loop if results are not null
        if resultpuissancevolume!=(0,0):
            break
        else:
            continue

#same with increment hypothesis jnoelec between 0 and 13 (i in range 0,14)
for i in range(0,14):
    for j in range (0,i+40):
        for k in range(0,21):
            resultpuissancevolume=iter(resultday,inputelecconsoday,listbatterie0[j],listpuissance0[i],k)
            if resultpuissancevolume!=(0,0):
                listresult_blackout.append(resultpuissancevolume)
                break
            else:
                continue
        if resultpuissancevolume!=(0,0):
            break
        else:
            continue

# Display of iteration loop results
for k in listresult_0blackout:
    print(f"""with user input data and
    a battery of{k[0]}kWh
    and a peak power of {k[1]}kWc,
    and an hypothesis of  {k[2]}j without electricity as acceptable
    et {k[3]} blackout episodes (or with an power generator)
    between 2005 and 2020:
    List of episodes without electricity (nb of days):
    {k[5]}
    for a mean duration of electricityless episodes of  {k[4]}j
    we meet the user needs ({inputelecconsoday}kWh/jour) entered as hypothesis""")

for k in listresult_blackout:
    print(f"""with user input data and
    a battery of{k[0]}kWh
    and a peak power of {k[1]}kWc,
    and an hypothesis of  {k[2]}j without electricity as acceptable
    et {k[3]} blackout episodes (or with an power generator):
    between 2005 and 2020:
    List of episodes without electricity (nb of days):
```

```
    List of episodes without electricity (nb of days):
    {k[5]}
    for a mean duration of electricityless episodes of  {k[4]}j
    we meet the user needs ({inputelecconsoday}kWh/jour) entered as hypothesis""")

print('\n\n\n\n')
print('display of optimized results')
# Optimisation résultats:

# first result incrementing peak power where battery capacity is minimal
batterie_0blackout=[k[0] for k in listresult_0blackout]
minbatterie_0blackout=min(batterie_0blackout)
for k in listresult_0blackout:
    if k[0]==minbatterie_0blackout:
        print(f"""with user input data and
        a battery of{k[0]}kWh
        and a peak power of {k[1]}kWc,
        and an hypothesis of  {k[2]}j without electricity as acceptable
        et {k[3]} blackout episodes (or with an power generator):
        between 2005 and 2020:
        List of episodes without electricity (nb of days):
        {k[5]}
        for a mean duration of electricityless episodes of  {k[4]}j
        we meet the user needs ({inputelecconsoday}kWh/jour) entered as hypothesis""")
        break
# first result incrementing peak power with a number of days of blackout lower or equal
# to the accepted user input
for k in listresult_blackout:
    if k[2]<=jnoelec:
        print(f"""with user input data and
        a battery of{k[0]}kWh
        and a peak power of {k[1]}kWc,
        and an hypothesis of  {k[2]}j without electricity as acceptable
        et {k[3]} blackout episodes (or with an power generator):
        between 2005 and 2020:
        List of episodes without electricity (nb of days):
        {k[5]}
        for a mean duration of electricityless episodes of  {k[4]}j
        we meet the user needs ({inputelecconsoday}kWh/jour) entered as hypothesis""")
        break
```

Étape 6 - Optimized results

For our example we have these results (with 1d of allowed blackout): the "optimized results" are the the bottom

```
Enter latitude: 44.2
Enter longitude: 0.6
Enter orientation (south,west,east,nord)x (est,ouest,nord,sud)sud
Entre module angle in °45
--2024-04-08 20:04:29--  https://re.jrc.ec.europa.eu/api/v5_2/seriescalc?lat=44.2&lon=0.6&loss=14&angle=45.0&aspect=0&startyear=2005&endyear=2020&pvcalculation=1&peakpower=1&pvtechchoice=crystSi&browser=0&outputformat=csv
Résolution de re.jrc.ec.europa.eu (re.jrc.ec.europa.eu)… 64:ff9b::8bbf:dd12, 139.191.221.18
Connexion à re.jrc.ec.europa.eu (re.jrc.ec.europa.eu)|64:ff9b::8bbf:dd12|:443… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 6291134 (6,0M) [text/html]
Enregistre : «output.csv»

output.csv          100%[===================>]   6,00M  1,03MB/s    ds 7,2s

2024-04-08 20:04:43 (852 KB/s) - «output.csv» enregistré [6291134/6291134]

maximum nb of consecutive days for 1kWc to produce 1kWh: 4 j
number of occurences: 1
mean nb of consecutive days for 1kWc to produce 1kWh: 1.1124367317425885 j

maximum nb of consecutive days for 1kWc to produce 2kWh: 5 j
number of occurences: 3
mean nb of consecutive days for 1kWc to produce 2kWh: 1.2801404603979711 j

maximum nb of consecutive days for 1kWc to produce 3kWh: 7 j
number of occurences: 1
```

mean nb of consecutive days for 1kWc to produce 3kWh: 1.4647827920708563 j

Mean daily production (kWh):
3.4641937012320336
Minimum daily production (kWh):
0.0
Maximum daily production (kWh):
6.65871

Quarterly production (kWh):
trimestre

2005-03-31	270.56038
2005-06-30	375.15833
2005-09-30	399.84068
2005-12-31	227.76775
2006-03-31	247.03158
2006-06-30	406.70165
2006-09-30	389.10805
2006-12-31	233.93622
2007-03-31	229.46528
2007-06-30	348.05120
2007-09-30	390.13982
2007-12-31	250.21269
2008-03-31	250.87365
2008-06-30	347.15819
2008-09-30	385.60970
2008-12-31	202.44585
2009-03-31	286.11329
2009-06-30	350.51328
2009-09-30	397.86835
2009-12-31	237.76417
2010-03-31	270.32507
2010-06-30	366.21454
2010-09-30	394.93503
2010-12-31	236.08537
2011-03-31	271.73968
2011-06-30	400.68298
2011-09-30	374.99079
2011-12-31	255.64396
2012-03-31	294.48135
2012-06-30	344.82346
2012-09-30	394.74104
2012-12-31	217.83773
2013-03-31	259.24581
2013-06-30	323.20427
2013-09-30	402.14528
2013-12-31	228.40704
2014-03-31	268.96648
2014-06-30	385.06742
2014-09-30	388.64375
2014-12-31	252.08195
2015-03-31	242.45034
2015-06-30	402.47347
2015-09-30	400.23336
2015-12-31	273.02084
2016-03-31	248.06779
2016-06-30	354.35530
2016-09-30	411.73628
2016-12-31	224.28598
2017-03-31	267.90392
2017-06-30	410.47864
2017-09-30	363.44330
2017-12-31	251.00240
2018-03-31	231.96132
2018-06-30	362.16785
2018-09-30	426.21631
2018-12-31	219.18595
2019-03-31	313.23495
2019-06-30	368.76834
2019-09-30	407.61175
2019-12-31	201.18704
2020-03-31	283.16855
2020-06-30	389.45566
2020-09-30	395.65480

2020-12-31   240.10074
Freq: Q-DEC

Yearly mean production of 1kWc (kWh):
 1265.2967493749998

Maximum consecutive days at zero production: 2

 Quarterly mean for each quarter (kWh):
 trimestre
1   264.724340
2   370.954661
3   395.182393
4   234.435355
Name: P, dtype: float64

Quarterly minimum for each quarter (kWh):
 trimestre
1   229.46528
2   323.20427
3   363.44330
4   201.18704
Name: P, dtype: float64

Quarterly maximum for each quarter (kWh):
 trimestre
1   313.23495
2   410.47864
3   426.21631
4   273.02084
Name: P, dtype: float64
Enter daily electricity consumption (kWh): 4
Do you wnat to use lead batteries? Type o for yes (lithium by default or if nothing is typed)
Initial battery treshold with entered hypothesis and user provided data (battery lithium))
hypothesis: (daily elec consumption/0.8 with lithium and daily elec consumption/0.5 with lead)
5 kWh
Enter maximum authorized percentage discharge (20% if you want your battery doesnt get empty to less than 20%. Hit enter if you want to use the default
 50% values for lead and 0% otherwise -hypothesis solid state batteries-
Combien de jours maximum consécutifs sans electricité (ou avec un groupe electrogene) pouvez vous supporter (0 par défaut si reponse vide)1
Maximum number of consecutive days without electricity (or with a generator) taken as hypothesis
1 j

Initial power treshold with user provided hypothesis and data
hypothèse: conso elec journaliere/2 (2kWh produit par kWc en hiver)
2 kWc

Initial power treshold with user provided hypothesis and data
hypothesis : daily elec consumption/0.8 with lithium and 0.5 with lead
5 kWh

If you want to correct the initial value of the battery (kWh) for iterations, enter your value, otherwise hit enter

type error or empty user value, continuing with batterie0=5kWh

If you want to correct initial peak power (kWc) for iterations, enter your value, otherwise hit enter

type error or empty user value, continuing with batterie0= puissance0=2kWc
with user input data and
    a battery of75.0kWh
    and a peak power of 2.0kWc,
    and an hypothesis of 0j without electricity as acceptable
    et 0 blackout episodes (or with an power generator)
    between 2005 and 2020:
    List of episodes without electricity (nb of days):
    []
    for a mean duration of electricityless episodes of 0j
    we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
    a battery of22.5kWh
    and a peak power of 2.66kWc,
    and an hypothesis of 0j without electricity as acceptable
    et 0 blackout episodes (or with an power generator)
    between 2005 and 2020:
    List of episodes without electricity (nb of days):

List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of17.5kWh
and a peak power of 3.3200000000000003kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of15.0kWh
and a peak power of 3.98kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of12.5kWh
and a peak power of 4.640000000000001kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of12.5kWh
and a peak power of 5.300000000000001kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of10.0kWh
and a peak power of 5.96kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of10.0kWh
and a peak power of 6.62kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of10.0kWh
and a peak power of 7.28kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and

with user input data and
a battery of10.0kWh
and a peak power of 7.94kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of10.0kWh
and a peak power of 8.600000000000001kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of10.0kWh
and a peak power of 9.260000000000002kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of10.0kWh
and a peak power of 9.92kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of10.0kWh
and a peak power of 10.58kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator)
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of5.0kWh
and a peak power of 2.0kWc,
and an hypothesis of 10j without electricity as acceptable
et 185 blackout episodes (or with an power generator):
between 2005 and 2020:
List of episodes without electricity (nb of days):
[['14-01-2005', 1], ['19-01-2005', 3], ['23-01-2005', 2], ['01-02-2005', 1], ['11-02-2005', 3], ['17-11-2005', 1], ['26-11-2005', 3], ['02-12-2005', 1], ['04-12-200
5', 2], ['07-12-2005', 2], ['12-12-2005', 6], ['22-12-2005', 2], ['25-12-2005', 3], ['31-12-2005', 2], ['03-01-2006', 10], ['18-01-2006', 4], ['29-01-2006', 7], ['23-02
-2006', 1], ['09-03-2006', 2], ['21-11-2006', 2], ['25-11-2006', 2], ['28-11-2006', 1], ['30-11-2006', 5], ['08-12-2006', 3], ['20-12-2006', 7], ['08-01-2007', 1], ['20
-01-2007', 4], ['30-01-2007', 4], ['06-02-2007', 3], ['08-10-2007', 1], ['10-10-2007', 1], ['01-12-2007', 1], ['07-12-2007', 3], ['15-12-2007', 1], ['25-12-2007', 5], [
'01-01-2008', 3], ['05-01-2008', 5], ['11-01-2008', 1], ['18-01-2008', 2], ['22-01-2008', 2], ['30-01-2008', 1], ['01-02-2008', 1], ['07-03-2008', 1], ['11-03-2008', 2]
, ['17-05-2008', 2], ['28-10-2008', 1], ['30-10-2008', 2], ['02-11-2008', 2], ['06-11-2008', 2], ['15-11-2008', 9], ['30-11-2008', 2], ['03-12-2008', 2], ['12-12-2008',
9], ['25-12-2008', 2], ['04-01-2009', 9], ['14-01-2009', 1], ['23-01-2009', 2], ['19-04-2009', 2], ['16-08-2009', 1], ['11-11-2009', 1], ['02-12-2009', 1], ['04-12-20
09', 1], ['06-12-2009', 3], ['14-12-2009', 3], ['23-12-2009', 2], ['28-12-2009', 2], ['04-01-2010', 4], ['09-01-2010', 4], ['14-01-2010', 1], ['16-01-2010', 2], ['05-05
-2010', 2], ['01-11-2010', 1], ['04-11-2010', 3], ['08-11-2010', 5], ['25-11-2010', 1], ['28-11-2010', 1], ['02-12-2010', 2], ['17-12-2010', 2], ['22-12-2010', 3], ['07
-01-2011', 3], ['13-01-2011', 3], ['31-01-2011', 2], ['04-02-2011', 1], ['23-02-2011', 4], ['02-03-2011', 2], ['29-10-2011', 1], ['05-11-2011', 4], ['25-11-2011', 4], [
'04-12-2011', 5], ['14-12-2011', 3], ['21-12-2011', 3], ['13-01-2012', 3], ['23-01-2012', 1], ['08-02-2012', 1], ['15-02-2012', 1], ['20-10-2012', 2], ['06-11-2012', 1]
, ['20-11-2012', 4], ['25-11-2012', 5], ['14-12-2012', 1], ['22-12-2012', 1], ['05-01-2013', 6], ['19-01-2013', 1], ['22-01-2013', 1], ['26-01-2013', 2], ['08-11-2013',
1], ['10-11-2013', 1], ['15-11-2013', 1], ['17-11-2013', 3], ['23-11-2013', 3], ['03-12-2013', 4], ['25-01-2014', 3], ['01-02-2014', 1], ['15-02-2014', 1], ['24-11-20
14', 3], ['01-12-2014', 3], ['06-12-2014', 1], ['13-12-2014', 8], ['24-12-2014', 2], ['27-12-2014', 2], ['03-01-2015', 4], ['10-01-2015', 1], ['24-01-2015', 1], ['29-01
-2015', 2], ['03-03-2015', 1], ['12-11-2015', 5], ['19-11-2015', 1], ['26-11-2015', 1], ['04-12-2015', 2], ['03-01-2016', 8], ['21-01-2016', 3], ['01-02-2016', 2], ['16
-10-2016', 2], ['05-11-2016', 1], ['18-11-2016', 1], ['22-11-2016', 4], ['20-12-2016', 1], ['24-12-2016', 3], ['30-12-2016', 6], ['11-01-2017', 2], ['06-12-2017', 3], [
'10-12-2017', 2], ['13-12-2017', 6], ['22-12-2017', 4], ['28-12-2017', 3], ['03-01-2018', 5], ['14-01-2018', 3], ['20-01-2018', 4], ['30-01-2018', 2], ['05-02-2018', 2]

, ['08-02-2018', 1], ['20-02-2018', 1], ['28-03-2018', 1], ['28-10-2018', 2], ['02-11-2018', 1], ['09-11-2018', 2], ['04-12-2018', 1], ['09-12-2018', 1], ['13-12-2018',
4], ['21-12-2018', 5], ['29-12-2018', 4], ['08-01-2019', 1], ['13-01-2019', 2], ['22-01-2019', 1], ['30-01-2019', 2], ['23-10-2019', 2], ['02-11-2019', 1], ['04-11-20
19', 5], ['16-11-2019', 3], ['01-12-2019', 2], ['07-12-2019', 3], ['12-12-2019', 3], ['23-12-2019', 3], ['04-01-2020', 2], ['29-02-2020', 1], ['04-03-2020', 2], ['23-04
-2020', 1], ['12-05-2020', 1], ['06-10-2020', 1], ['03-12-2020', 2], ['10-12-2020', 2], ['13-12-2020', 3], ['18-12-2020', 2], ['21-12-2020', 5], ['28-12-2020', 2]]

for a mean duration of electricityless episodes of 2.5513513513513515j

we meet the user needs (4.0kWh/jour) entered as hypothesis

with user input data and

a battery of5.0kWh

and a peak power of 2.66kWc,

and an hypothesis of 8j without electricity as acceptable

et 103 blackout episodes (or with an power generator):

between 2005 and 2020:

List of episodes without electricity (nb of days):

[['19-01-2005', 3], ['23-01-2005', 1], ['01-02-2005', 1], ['12-02-2005', 2], ['26-11-2005', 3], ['05-12-2005', 1], ['08-12-2005', 1], ['12-12-2005', 6], ['23-12-200
5', 1], ['26-12-2005', 2], ['31-12-2005', 1], ['04-01-2006', 4], ['09-01-2006', 4], ['20-01-2006', 2], ['29-01-2006', 4], ['03-02-2006', 2], ['22-11-2006', 1], ['04-12-
2006', 1], ['10-12-2006', 1], ['22-12-2006', 5], ['22-01-2007', 2], ['30-01-2007', 4], ['08-02-2007', 1], ['01-12-2007', 1], ['08-12-2007', 2], ['26-12-2007', 4], ['03-
01-2008', 1], ['06-01-2008', 4], ['11-01-2008', 1], ['23-01-2008', 1], ['07-03-2008', 1], ['17-05-2008', 2], ['07-11-2008', 1], ['16-11-2008', 8], ['01-12-2008', 1], ['
04-12-2008', 1], ['15-12-2008', 6], ['26-12-2008', 1], ['05-01-2009', 8], ['24-01-2009', 1], ['19-04-2009', 2], ['16-08-2009', 1], ['11-11-2009', 1], ['08-12-2009', 1]
, ['14-12-2009', 3], ['23-12-2009', 2], ['05-01-2010', 3], ['09-01-2010', 2], ['12-01-2010', 1], ['06-05-2010', 1], ['05-11-2010', 2], ['12-11-2010', 1], ['22-12-2010',
3], ['08-01-2011', 1], ['31-01-2011', 2], ['06-11-2011', 3], ['25-11-2011', 4], ['07-12-2011', 2], ['21-12-2011', 3], ['14-01-2012', 2], ['20-10-2012', 2], ['21-11-20
12', 2], ['25-11-2012', 2], ['28-11-2012', 1], ['14-12-2012', 1], ['22-12-2012', 1], ['05-01-2013', 6], ['19-01-2013', 1], ['18-11-2013', 2], ['24-11-2013', 2], ['04-12
-2013', 2], ['25-01-2014', 3], ['25-11-2014', 2], ['02-12-2014', 2], ['14-12-2014', 7], ['05-01-2015', 2], ['13-11-2015', 2], ['04-01-2016', 5], ['10-01-2016', 1], ['23
-01-2016', 1], ['17-10-2016', 1], ['23-11-2016', 1], ['25-11-2016', 1], ['20-12-2016', 1], ['25-12-2016', 2], ['31-12-2016', 5], ['12-01-2017', 1], ['22-12-2017', 4], [
'29-12-2017', 1], ['04-01-2018', 3], ['16-01-2018', 1], ['21-01-2018', 3], ['31-01-2018', 1], ['04-12-2018', 1], ['15-12-2018', 2], ['22-12-2018', 4], ['30-12-2018', 3]
, ['31-01-2019', 1], ['17-11-2019', 2], ['02-12-2019', 1], ['08-12-2019', 1], ['13-12-2019', 2], ['29-12-2020', 1]]

for a mean duration of electricityless episodes of 2.203883495145631j

we meet the user needs (4.0kWh/jour) entered as hypothesis

with user input data and

a battery of5.0kWh

and a peak power of 3.3200000000000003kWc,

and an hypothesis of 6j without electricity as acceptable

et 66 blackout episodes (or with an power generator):

between 2005 and 2020:

List of episodes without electricity (nb of days):

[['20-01-2005', 2], ['23-01-2005', 1], ['01-02-2005', 1], ['12-02-2005', 2], ['27-11-2005', 1], ['12-12-2005', 6], ['27-12-2005', 1], ['06-01-2006', 2], ['09-01-200
6', 1], ['11-01-2006', 2], ['29-01-2006', 3], ['03-02-2006', 2], ['24-12-2006', 3], ['23-01-2007', 1], ['01-02-2007', 2], ['27-12-2007', 3], ['23-01-2008', 1], ['07-03-
2008', 1], ['17-05-2008', 2], ['16-11-2008', 4], ['21-11-2008', 1], ['23-11-2008', 1], ['19-12-2008', 2], ['26-12-2008', 1], ['05-01-2009', 1], ['07-01-2009', 6], ['19-
04-2009', 1], ['16-08-2009', 1], ['14-12-2009', 3], ['24-12-2009', 1], ['05-01-2010', 1], ['10-01-2010', 1], ['12-01-2010', 1], ['05-11-2010', 2], ['23-12-2010', 2], ['
01-02-2011', 1], ['07-11-2011', 1], ['26-11-2011', 3], ['21-12-2011', 3], ['14-01-2012', 1], ['20-10-2012', 2], ['21-11-2012', 1], ['26-11-2012', 1], ['14-12-2012', 1]
, ['05-01-2013', 6], ['24-11-2013', 2], ['05-12-2013', 1], ['26-01-2014', 2], ['26-11-2014', 1], ['02-12-2014', 2], ['17-12-2014', 3], ['13-11-2015', 1], ['05-01-2016',
1], ['07-01-2016', 2], ['17-10-2016', 1], ['23-11-2016', 1], ['01-01-2017', 4], ['23-12-2017', 3], ['06-01-2018', 1], ['22-01-2018', 2], ['31-01-2018', 1], ['16-12-20
18', 1], ['23-12-2018', 3], ['30-12-2018', 3], ['02-12-2019', 1], ['14-12-2019', 1]]

for a mean duration of electricityless episodes of 1.8636363636363635j

we meet the user needs (4.0kWh/jour) entered as hypothesis

with user input data and

a battery of5.0kWh

and a peak power of 3.98kWc,

and an hypothesis of 6j without electricity as acceptable

et 43 blackout episodes (or with an power generator):

between 2005 and 2020:

List of episodes without electricity (nb of days):

[['20-01-2005', 2], ['01-02-2005', 1], ['13-02-2005', 1], ['12-12-2005', 5], ['27-12-2005', 1], ['07-01-2006', 1], ['12-01-2006', 1], ['29-01-2006', 1], ['31-01-200
6', 1], ['04-02-2006', 1], ['25-12-2006', 2], ['02-02-2007', 1], ['28-12-2007', 2], ['23-01-2008', 1], ['07-03-2008', 1], ['17-05-2008', 2], ['16-11-2008', 4], ['21-11-
2008', 1], ['08-01-2009', 2], ['11-01-2009', 2], ['19-04-2009', 1], ['16-08-2009', 1], ['15-12-2009', 2], ['24-12-2009', 1], ['05-01-2010', 1], ['10-01-2010', 1], ['06-
11-2010', 1], ['23-12-2010', 2], ['01-02-2011', 1], ['26-11-2011', 3], ['22-12-2011', 2], ['21-10-2012', 1], ['05-01-2013', 6], ['25-11-2013', 1], ['26-01-2014', 1], ['
03-12-2014', 1], ['17-10-2016', 1], ['01-01-2017', 4], ['23-12-2017', 3], ['23-01-2018', 1], ['23-12-2018', 1], ['25-12-2018', 1], ['30-12-2018', 3]]

for a mean duration of electricityless episodes of 1.697674418604651j

we meet the user needs (4.0kWh/jour) entered as hypothesis

with user input data and

a battery of5.0kWh

and a peak power of 4.640000000000001kWc,

and an hypothesis of 4j without electricity as acceptable

et 22 blackout episodes (or with an power generator):

between 2005 and 2020:

List of episodes without electricity (nb of days):

[['12-12-2005', 1], ['14-12-2005', 3], ['27-12-2005', 1], ['26-12-2006', 1], ['07-03-2008', 1], ['17-05-2008', 2], ['17-11-2008', 3], ['09-01-2009', 1], ['11-01-200
9', 2], ['19-04-2009', 1], ['16-08-2009', 1], ['15-12-2009', 2], ['23-12-2010', 2], ['27-11-2011', 2], ['23-12-2011', 1], ['06-01-2013', 4], ['25-11-2013', 1], ['03-12-
2014', 1], ['02-01-2017', 3], ['24-12-2017', 2], ['23-12-2018', 1], ['01-01-2019', 1]]

for a mean duration of electricityless episodes of 1.6818181818181819j

we meet the user needs (4.0kWh/jour) entered as hypothesis

with user input data and

a battery of5.0kWh

and a peak power of 5.300000000000001kWc,

and an hypothesis of 4j without electricity as acceptable

et 16 blackout episodes (or with an power generator):
between 2005 and 2020:
List of episodes without electricity (nb of days):
[['14-12-2005', 3], ['27-12-2005', 1], ['17-05-2008', 2], ['17-11-2008', 3], ['09-01-2009', 1], ['12-01-2009', 1], ['19-04-2009', 1], ['16-08-2009', 1], ['15-12-2009', 1], ['24-12-2010', 1], ['28-11-2011', 1], ['06-01-2013', 4], ['25-11-2013', 1], ['03-12-2014', 1], ['03-01-2017', 2], ['01-01-2019', 1]]
for a mean duration of electricityless episodes of 1.5625j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of5.0kWh
and a peak power of 5.96kWc,
and an hypothesis of 2j without electricity as acceptable
et 10 blackout episodes (or with an power generator):
between 2005 and 2020:
List of episodes without electricity (nb of days):
[['15-12-2005', 2], ['17-05-2008', 2], ['18-11-2008', 2], ['19-04-2009', 1], ['16-08-2009', 1], ['15-12-2009', 1], ['24-12-2010', 1], ['06-01-2013', 2], ['09-01-2013', 1], ['03-12-2014', 1]]
for a mean duration of electricityless episodes of 1.4j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of5.0kWh
and a peak power of 6.62kWc,
and an hypothesis of 2j without electricity as acceptable
et 5 blackout episodes (or with an power generator):
between 2005 and 2020:
List of episodes without electricity (nb of days):
[['16-12-2005', 1], ['18-05-2008', 1], ['19-04-2009', 1], ['16-08-2009', 1], ['06-01-2013', 2]]
for a mean duration of electricityless episodes of 1.2j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of5.0kWh
and a peak power of 7.28kWc,
and an hypothesis of 1j without electricity as acceptable
et 4 blackout episodes (or with an power generator):
between 2005 and 2020:
List of episodes without electricity (nb of days):
[['18-05-2008', 1], ['19-04-2009', 1], ['16-08-2009', 1], ['07-01-2013', 1]]
for a mean duration of electricityless episodes of 1.0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of5.0kWh
and a peak power of 7.94kWc,
and an hypothesis of 1j without electricity as acceptable
et 4 blackout episodes (or with an power generator):
between 2005 and 2020:
List of episodes without electricity (nb of days):
[['18-05-2008', 1], ['19-04-2009', 1], ['16-08-2009', 1], ['07-01-2013', 1]]
for a mean duration of electricityless episodes of 1.0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of5.0kWh
and a peak power of 8.600000000000001kWc,
and an hypothesis of 1j without electricity as acceptable
et 3 blackout episodes (or with an power generator):
between 2005 and 2020:
List of episodes without electricity (nb of days):
[['18-05-2008', 1], ['19-04-2009', 1], ['16-08-2009', 1]]
for a mean duration of electricityless episodes of 1.0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of5.0kWh
and a peak power of 9.260000000000002kWc,
and an hypothesis of 1j without electricity as acceptable
et 3 blackout episodes (or with an power generator):
between 2005 and 2020:
List of episodes without electricity (nb of days):
[['18-05-2008', 1], ['19-04-2009', 1], ['16-08-2009', 1]]
for a mean duration of electricityless episodes of 1.0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of5.0kWh
and a peak power of 9.92kWc,
and an hypothesis of 1j without electricity as acceptable
et 3 blackout episodes (or with an power generator):
between 2005 and 2020:

List of episodes without electricity (nb of days):
[['18-05-2008', 1], ['19-04-2009', 1], ['16-08-2009', 1]]
for a mean duration of electricityless episodes of 1.0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of5.0kWh
and a peak power of 10.58kWc,
and an hypothesis of 1j without electricity as acceptable
et 3 blackout episodes (or with an power generator):
between 2005 and 2020:
List of episodes without electricity (nb of days):
[['18-05-2008', 1], ['19-04-2009', 1], ['16-08-2009', 1]]
for a mean duration of electricityless episodes of 1.0j
we meet the user needs (4.0kWh/jour) entered as hypothesis

Optimized results display
with user input data and
a battery of10.0kWh
and a peak power of 5.96kWc,
and an hypothesis of 0j without electricity as acceptable
et 0 blackout episodes (or with an power generator):
between 2005 and 2020:
List of episodes without electricity (nb of days):
[]
for a mean duration of electricityless episodes of 0j
we meet the user needs (4.0kWh/jour) entered as hypothesis
with user input data and
a battery of5.0kWh
and a peak power of 7.28kWc,
and an hypothesis of 1j without electricity as acceptable
et 4 blackout episodes (or with an power generator):
between 2005 and 2020:
List of episodes without electricity (nb of days):
[['18-05-2008', 1], ['19-04-2009', 1], ['16-08-2009', 1], ['07-01-2013', 1]]
for a mean duration of electricityless episodes of 1.0j
we meet the user needs (4.0kWh/jour) entered as hypothesis