



OpenMod  Africa

openTEPES model installation



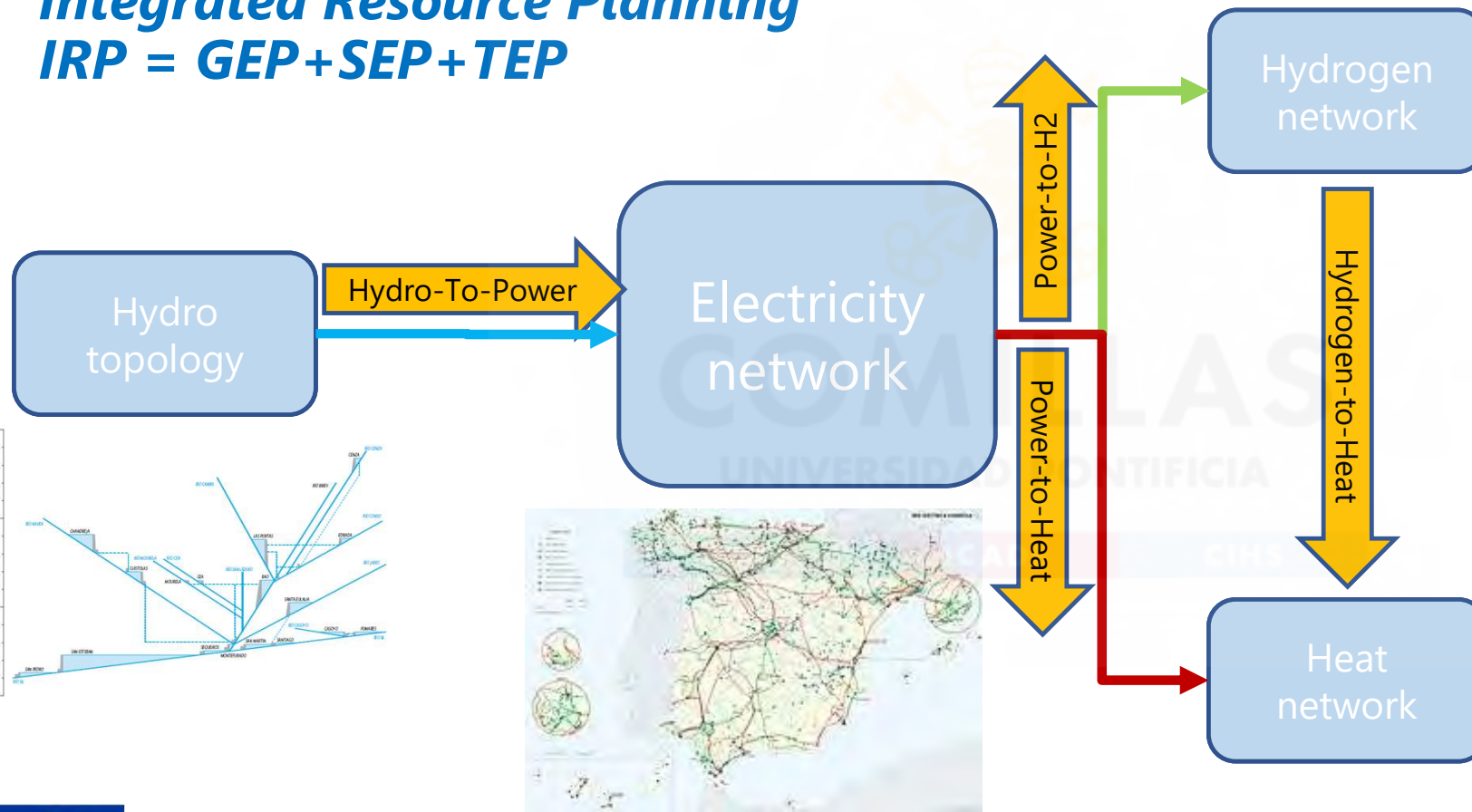
Andrés Ramos, Erik Álvarez
Comillas Pontifical University
Institute for Research in Technology
Madrid, Spain
January 2024



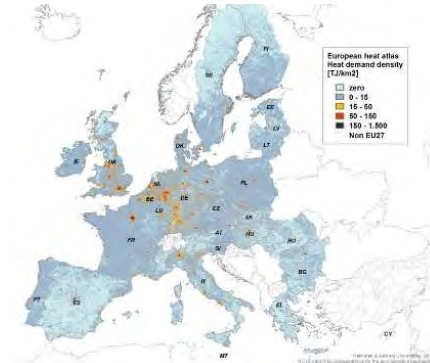
Electricity/hydrogen/heat/water networks

Multi-energy carriers. Sector coupling

Integrated Resource Planning
IRP = GEP+SEP+TEP



<https://www.energyplan.eu/heat-roadmap-europe-2050/>





Open Generation, Storage, and Transmission Operation and Expansion Planning Model with F and ESS (**openTEPES**)

Read the Docs

<https://opentepes.readthedocs.io/en/latest/index.html>

“Simplicity and Transparency in Energy Systems Planning”

The **openTEPES** model has been developed at the [Instituto de Investigación Tecnológica \(IIT\)](#) of the [Universidad Pontificia Comillas](#).

The **openTEPES** model presents a decision support system for defining the integrated generation, storage, and transmission expansion plan (GEP+SEP+TEP) of a **large-scale electric system** at a tactical level (i.e., time horizons of 10-20 years), defined as a set of **generation, storage, and (electricity, hydrogen, and heat) networks dynamic investment decisions for multiple future years**.

It is integrated into the [open energy system modelling platform](#), helping model Europe’s energy system.

It has been used by the **Ministry for the Ecological Transition and the Demographic Challenge (MITECO)** to analyze the electricity sector in the latest Spanish National Energy and Climate Plan (NECP) Update 2023-2030 in September 2024.

Reference: A. Ramos, E. Quispe, S. Lumbreras “[OpenTEPES: Open-source Transmission and Generation Expansion Planning](#)” SoftwareX 18: June 2022 [10.1016/j.softx.2022.101070](https://doi.org/10.1016/j.softx.2022.101070)

openTEPES: [summary presentation \(English\)](#), [présentation \(French\)](#), and [installation guide](#)

downloads 133k

DOI: <https://doi.org/10.24433/CO.8709849.v1>



[GitHub - IIT-EnergySystemModels/opentepes: Open Generation, Storage, and Transmission Operation and Expansion Planning Model with RES and ESS \(openTEPES\)](https://github.com/IIT-EnergySystemModels/opentepes)

Index

- [Introduction](#)
- [Electric System Input Data](#)
 - [Acronyms](#)
 - [Dictionaries. Sets](#)
 - [Input files](#)
 - [Options](#)
 - [Parameters](#)
 - [Period](#)
 - [Scenario](#)
 - [Stage](#)
 - [Adequacy reserve margin](#)
 - [Maximum CO2 emission](#)
 - [Minimum RES energy](#)

openTEPES

version 4.18.1

Search Go

Navigation

- [Introduction](#)
- [Electric System Input Data](#)
- [Hydropower System Input Data](#)
- [Hydrogen System Input Data](#)
- [Heat System Input Data](#)
- [Output Results](#)
- [Mathematical Formulation](#)
- [Research projects](#)
- [Publications](#)
- [Download & Installation](#)
- [Contact Us](#)

Installing Python and solvers in Windows/Linux

<https://opentepes.readthedocs.io/en/latest/Download.html>

Python is an interpreted, high-level, general-purpose programming language.

Please follow the next steps for its installation with **Administrator privileges**

1. Install **miniconda**: Python 3.12.3 (<https://www.anaconda.com/download/success>)

Conda is an open-source package management system and environment management system that runs on Windows, MacOS, and Linux.

Miniconda is a free minimal installer for conda.

2a. In **Windows**, open **Anaconda Powershell Prompt (Miniconda3)** with Administrator privileges

2b. In **Ubuntu**, open a conda console typing in a terminal `source /anaconda3/bin/activate`

3. Install some **solvers**

```
gurobi      conda install -c gurobi      gurobi
```

The Gurobi license (<https://www.gurobi.com/features/academic-named-user-license/>) must be copied to the Python folder (`C:\ProgramData\miniconda3`)

You can also ask for an evaluation license (<https://www.gurobi.com/downloads/request-an-evaluation-license/>)

```
highs      pip install highspy
```

```
glpk       conda install glpk
```

```
scip       conda install -c conda-forge pycscipopt
```

```
CBC        conda install -c conda-forge coincbc (only for Linux)
```

Update all the packages and check the consistency among them

```
conda update conda
```

```
conda update --all
```



Installing Gurobi



- Install gurobi from an Anaconda prompt (run as Administrator or not)
`conda install -c gurobi gurobi`
- Register for a free Gurobi account as an academic and log in
<https://portal.gurobi.com/iam/register/>
- Request for a free academic license: Named-User Academic
<https://portal.gurobi.com/iam/licenses/request/?type=academic>
- You will get something like this
`grbgetkey ae36ac20-16e6-acd2-f242-4da6e765fa0a`
- Create a cmd prompt and go to the Python folder
`C:\ProgramData\miniconda3` or `C:\ProgramData\anaconda3`
- Use `grbgetkey ae36ac20-16e6-acd2-f242-4da6e765fa0a`
- You will get a `gurobi.lic` text file with the license linked to your username and PC name
- Copy the license into the Python folder
`C:\ProgramData\miniconda3`
`C:\ProgramData\anaconda3`



Installing CPLEX



- Go to **IBM ILOG CPLEX Optimization Studio** (<https://www.ibm.com/products/ilog-cplex-optimization-studio>) and click **Try it free**.
- You will be asked to **create an account as an academic** or use an already existing one
- You will be directed to the download page
- **Download** the version corresponding to your operating system and install it
- Add the folder where **cplex.exe** is located (C:\Program Files\IBM\ILOG\CPLEX_Studio2211\cplex\bin\x64_win64) to the PATH Windows environment variable



Installing openTEPES as a Python package

<https://opentepes.readthedocs.io/en/latest/Download.html#>

1. Launch an [Anaconda command prompt](#)
2. [Install openTEPES via pip](#) as a Python package

```
pip install openTEPES
```

The openTEPES installation automatically installs these [additional packages](#)

pandas	conda install pandas
psutil	conda install psutil
matplotlib	conda install matplotlib
altair	conda install altair
plotly	conda install plotly
colour	conda install colour
networkx	conda install networkx
pyomo	conda install -c conda-forge pyomo



Run openTEPES from an Anaconda prompt

If installed with pip

(located in C:\ProgramData\miniconda3\Scripts)

(located in C:\ProgramData\anaconda3\Scripts)

openTEPES_Main

Then select

- Directory
- Case
- Solver
- Results
- Log information

Installing an IDE

Python language can be coded in PyCharm, Google Colaboratory, Notepad++, Jupyter Notebooks, or Visual Studio Code.

Use of **Pycharm**

1. Install Pycharm Community (<https://www.jetbrains.com/pycharm/>)
2. Click on Configuration -> Settings...
3. On the left bar, click on Project Interpreter
4. Up on the right menu, click the cogwheel and select Add...
5. Choose a New environment
 - Location: select the folder C:\Users\YourUser\PycharmProject\EnvironmentName
 - Base interpreter: select python.exe inside the folder Miniconda3

Use of **Visual Studio Code**

1. Install Visual Studio Code (<https://code.visualstudio.com/>)

Use of **Google Colaboratory**

1. Go to Google Colaboratory and enter with your Gmail account
3. Install pyomo and glpk using:

```
!pip install openTEPES  
!pip install pyomo  
!apt install glpk-utils libglpk-dev  
!pip install glpk
```

The installation of pyomo and glpk must be done every time you enter in Google Colaboratory
gurobi is available under this platform
2. If you want to save the code, input data, and output results, you must move them to your space in GoogleDrive in /content/drive/MyDrive
3. Create a new notebook, add code, and then execute

```
!python /content/drive/MyDrive/openSDUC_main.py  
!python /content/drive/MyDrive/openTEPES_Main.py
```

You will need to modify the filenames in some modules to point to the folder in your GoogleDrive.

openTEPES installation for using from an IDE

<https://opentepes.readthedocs.io/en/latest/Download.html#>

1. Clone the openTEPES repository.



1. Launch the **Anaconda command prompt**

2. Set up the PATH to

```
cd "C:\Users\<<username>\...\openTEPES"
```

(Note that the path is where the repository was cloned)

3. Type: **pip install .**

2. Download from the website (ReadTheDocs), and install as if this has been cloned.



Run openTEPES from an IDE

If cloned, run from the folder where it has been cloned.

```
python openTEPES_main.py
```

Then select

- Directory
- Case
- Solver
- Results
- Log information

An alternative way to run the model is by creating a new script script.py and writing the following:

```
from openTEPES.openTEPES import openTEPES_run  
openTEPES_run(<dir>, <case>, <solver>, <results>, <log>)
```



OpenMod Africa

Thanks for your attention!



Andrés Ramos, Erik Álvarez
Universidad Pontificia Comillas
Instituto de Investigación Tecnológica

Andres.Ramos@comillas.edu, ealvarezq@comillas.edu
<https://opentepes.readthedocs.io/en/latest/index.html>

