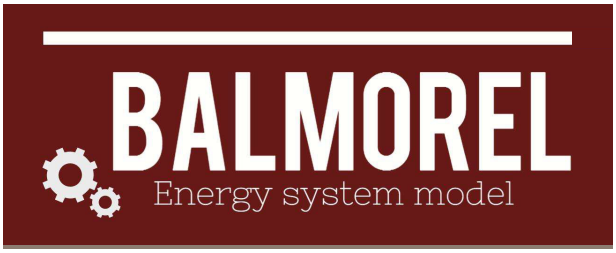

Documentation for the Balmorel-Antares Soft-Linking Framework

Version 1.0

Mathias Berg Rosendal

PhD Student at the Technical University of Denmark
Department of Technology, Management and Economics

Funded by the French transmission system operator,
Réseau de Transport d'Électricité (RTE)

February 27, 2025

Contents

List of Code Listings

Acronyms

1 Introduction

The purpose of this document is to provide a practical guide on how to use the Balmorel-Antares Soft-Coupling Framework (BAF). For a scientific application, see Rosendal et al. [4]. BAF is still at a very early stage of development, meaning this relatively scarce documentation will be extended over time.

2 Installation

GAMS 37, Antares 8.6.1 and Python 3.9.11 are required to run the BAF. GAMS 41 has also been tested succesfully. It is also important to install the Python-API, which is distributed for Python 3.9 in the GAMS installations. Follow sections 2.1.1-2.1.3 thoroughly to ensure an adequate installation. The GitHub repository only includes a very limited amount of model data for both Antares and Balmorel. Remember therefore to download the model data after installing necessary software as described in sec. 2.2.

2.1 Software

2.1.1 GAMS 37

Balmorel is written in GAMS, which can be downloaded here[1]. Use version 37 or newer. GAMS, unfortunately, requires a license in order to run the solver, meaning Balmorel is not an open-solver model.

2.1.2 Antares 8.6.1

The version of Antares used in the BAF is Antares 8.6.1, which can be downloaded from here[2].

2.1.3 Python 3.9.11

To ensure compatibility, download Python 3.9.11[3], and create a virtual environment. Open up a command prompt, cd to the top level of the BAF (see fig. 1), and follow the commands illustrated in listing 1 (assuming Windows). This will create an environment in the top level folder called ".BAF-Env", activate it

[1] https://www.gams.com/download/
[2] https://antares-simulator.org/pages/antares-simulator/6/
[3] https://www.python.org/downloads/release/python-3911/

and install the necessary packages: pandas, matplotlib, geopandas, openpyxl and scipy.

```
cd folder/to/the/top/level/of/BAF

py -m venv .BAF-Env

.\BAF-Env\Scripts\activate

pip install matplotlib

pip install pandas

pip install openpyxl

pip install geopandas

pip install scipy

pip install plotly

pip install nbformat

# If R is installed:
pip install rpy2
```

Listing 1: Windows example of how to install and activate the necessary Python environment in a folder called folder/to/the/top/level/of/BAF. Note that the rpy2 package is not necessary but useful for interacting with R through Python and the packages developed for Antares.

Further documentation on Python virtual environments (e.g. how to operate in Unix or Mac) can be found in the documentation[4]. It is recommended to use standard Python and pip as described above. However, if you have an Anaconda[5] installation of Python, the environment creation command is "conda create -n .BAF-Env". While still being in the *base* conda environment, packages can be installed to the new environment using "conda install -n .BAF-Env package_name". *After* that, the environment should be setup and can be activated using "conda activate .BAF-Env" (this was tested with the conda version 23.3.1, check yours using "conda –version").

Finally, it is necessary to install the GAMS Python API. Follow the commands in listing 2.

[4] https://docs.python.org/3/tutorial/venv.html
[5] https://www.anaconda.com/

```
set PYTHONPATH=C:\path\to\gams\apifiles\Python\api_39
set PYTHONPATH=C:\path\to\gams\apifiles\Python\gams;%PYTHONPATH%

.\BAF-Env\Scripts\activate

cd C:\path\to\gams\apifiles\Python\api_39
C:\path\to\my\python\python.exe setup.py install
```

Listing 2: How to install the Python package in your new .BAF-Env environment, for interacting with GAMS output files.

If you encounter any errors, consult these instructions[6].

2.1.4 R 4.2.3

Useful R packages exist for visualising Antares results, but are not necessary to run BAF. Download and install R-4.2.3, e.g. using the Windows file R-4.2.3-win.exe[7] from Download R for Windows/base/Previous releases/R4.2.3. Open up R, and follow the instructions in listing 3.

```
install.packages("antaresViz")

library(antaresViz)

runAppAntaresViz()
```

Listing 3: How to install the necessary R package and make sure that it works. If a GUI is opened in a browser with no errors, the installation was succesful.

2.2 Model Data

The files for the specific models of a European bidding zone, electricity and hydrogen system in Balmorel and Antares cannot be shared by the authors, but were based on a combination of Kountouris et al. [3] and Janin et al. [2]. The framework itself can be downloaded from a private repository [8]. Copy paste the folders Balmorel and Antares to the top level of the BAF, and the framework should be able to run successfully.

[6] https://www.gams.com/37/docs/API_PY_TUTORIAL.html
[7] https://cran.rstudio.com/
[8] https://github.com/Mathias157/balmorel-antares

3 Architecture

This section will describe the general structure of the BAF. Fig. 1 illustrates the top level, and sections 3.1-3.5 are dedicated to describing the contents that are necessary to operate the BAF.

BAF assumes basic knowledge of Python and GAMS programming and an understanding of Balmorel and Antares.

The documentation folder contains the LaTeXsource code behind this PDF, including Affinity Designer 2[9] files used to create nearly all figures in this document.

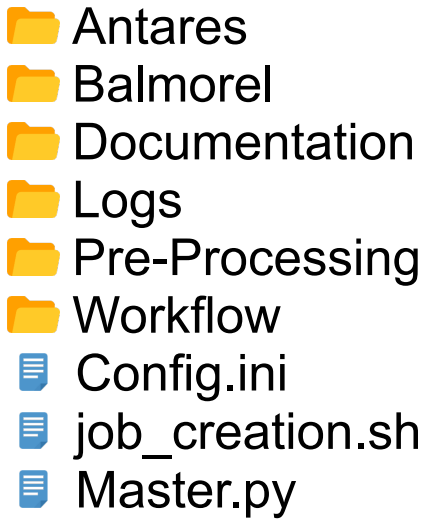

Figure 1: The top level of the Balmorel-Antares Soft-Coupling Framework.

3.1 Antares

Antares is a network flow model developed by the French transmission system operator, Réseau de Transport d'Électricité (RTE). The model is described in Doquet et al. [1], with further documentation here[10]. The specific model from Janin et al. [2] was used.

Note that a Windows installation may require you to run the .exe file as administrator.

[9] https://affinity.serif.com/en-us/designer/
[10] https://antares-doc.readthedocs.io/en/latest/

3.2 Balmorel

Balmorel is a partial equilibrium, least-cost optimisation energy system model described in Frauke et al. [5]. The version used in this framework is based on Kountouris et al. [3].

3.3 Logs and Job Creation

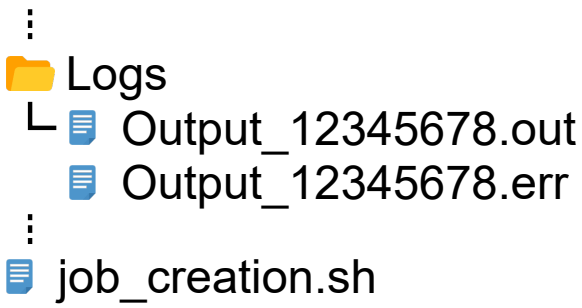

Figure 2: The Logs folder and job_creation.sh script (see top level in fig. 2).

The Logs folder and job_creation.sh script is only relevant when running the BAF framework on a High-Performance Computer (HPC) (the Logs folder is not included in the repository but will be created by thep Initialisation.py script if it does not exist). Logs will include an .out file where every printed output of python scripts and Balmorel and Antares executions is logged. An .err logs errors in all executions. Log files are helpful for checking in on the progress of a time-consuming framework, such as the BAF.

Listing 4 is an example of a script for specifying commands of your run on a HPC. It is based on the HPC setup at the Technical University of Denmark (DTU)'s[11] clusters as of February 27, 2025. Note that it will be necessary to install as described in sec. 2, activate the Python environment and assign path variables so the HPC knows where to fetch binaries. The Python environment is assumed to be called "BAF-Env" in the example in listing 4.

This job will be named "Name_of_the_job" (-J command) and will be submitted to a queue named "Queue_Name" (-q). Contact your institute or organisation for access to this queue. These settings illustrate a job that

[11] https://www.hpc.dtu.dk/?page_id=2534

utilise 10 cores (-n) with maximum 3 GB pr. slot used by the same host (-R). The job should be killed if more than 20 hours passes (-W). -o and -e commands specify where to put output logs, and their names. %J is a command for the job number, which in the DTU system is an 8-digit number, such as 12345678, as in the example illustrated in fig. 2.

```
###!/bin/sh
### General options
### -- specify queue --
#BSUB -q Queue_name
### -- set the job Name --
#BSUB -J Name_of_the_job
### -- ask for number of cores (default: 1) --
#BSUB -n 10
### -- specify that the cores must be on the same host --
#BSUB -R "span[hosts=1]"
### -- specify that we need 3GB of memory per core/slot --
#BSUB -R "rusage[mem=3GB]"
### -- specify that we want the job to get killed if it exceeds 3 GB per core/slot --
#BSUB -M 3GB
### -- set walltime limit: hh:mm --
#BSUB -W 20:00
### -- set the email address --
##BSUB -u your_email@domain.com
### -- send notification at start --
###BSUB -B
### -- send notification at completion --
###BSUB -N
### -- Specify the output and error file. %J is the job-id --
### -- -o and -e mean append, -oo and -eo mean overwrite --
#BSUB -o ./Logs/Output_%J.out
#BSUB -e ./Logs/Output_%J.err

### Load modules and find binaries
module load python3/3.9.11
source BAF-Env/bin/activate
export PATH=/zhome/c0/2/105719/Desktop/Antares-8.6.1/bin:$PATH
export LD_LIBRARY_PATH=/appl/gams/37.1.0:$LD_LIBRARY_PATH
export PYTHONPATH=/appl/gams/37.1.0/apifiles/Python/gams:/appl/gams/37.1.0/apifiles/Python/api_39:$PYTHONPATH

### Run framework
python3 Master.py
```

Listing 4: An example of a script for creating a job on a high-performance computer.

3.4 Pre-Processing

The Pre-Processing folder include data and configuration files, that are either used or created by the Pre-Processing.py script. E.g., this includes mappings of nodes and resource constraints from Balmorel to Antares. It's not necessary to run the Pre-Processing.py script, as the necessary files will be saved in git if Pre-Processing steps are changed.

3.5 Configuration, Workflow and Master Script

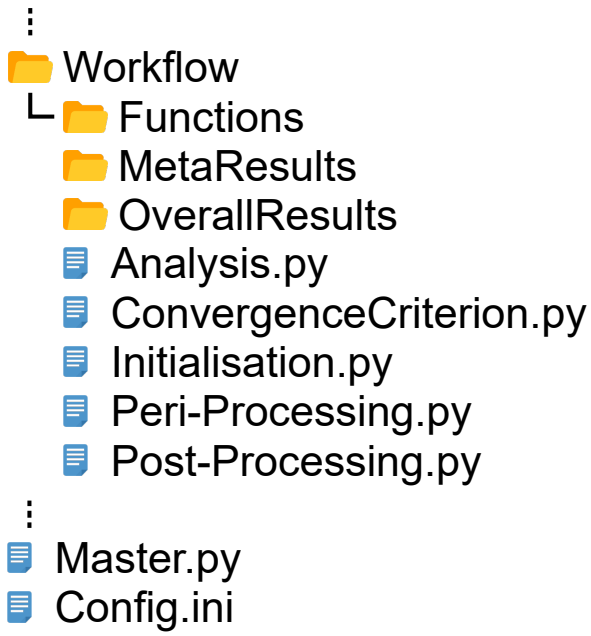

Figure 3: Workflow folder, Master script and Configuration file.

The workflow folder and Master.py script is where all data processing between model executions occur. The procedure of executing one scenario of the BAF will include editing the Config.ini file in order to set the specific scenario name (SC parameter) and running the Master.py script, consecutively.

The Master.py script will trigger scripts in the workflow folder, following the general logic of a bi-directional model coupling algorithm as illustrated in fig. 4.

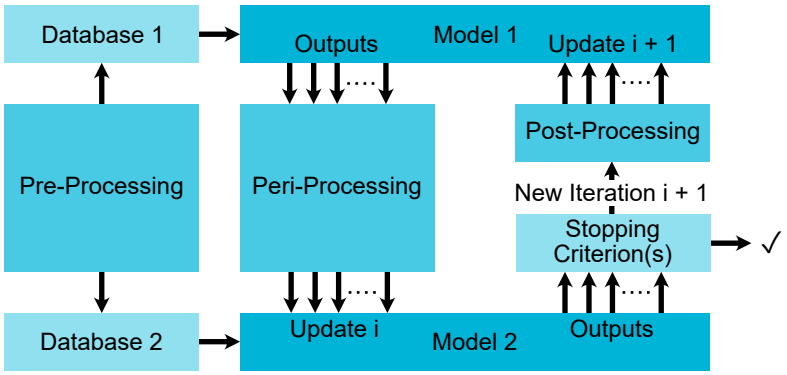

Figure 4: A generalised, bi-directional model coupling algorithm.

References

[1] M Doquet et al. “A New Tool for Adequacy Reporting of Electric Systems: ANTARES”. In: 2008.

[2] Jean-Marc Janin, Saga Guillbrandsson, and Dante Powell. “Coordinated Development of Renewable Energy Sources and Electrolysers in Europe”. In: *Renewable Integration Week*. 2023.

[3] Ioannis Kountouris et al. “A Unified European Hydrogen Infrastructure Planning to Support the Rapid Scale-up of Hydrogen Production”. In: *Nature Communications* 15.1 (June 2024), p. 5517. ISSN: 2041-1723. DOI: 10.1038/s41467-024-49867-w. (Visited on 08/19/2024).

[4] M. Rosendal et al. “The Benefits and Challenges of Soft-Linking Investment and Operational Energy System Models”. In: *Applied Energy* 385 (May 2025), p. 125512. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2025.125512. (Visited on 02/27/2025).

[5] Frauke Wiese et al. “Balmorel Open Source Energy System Model”. In: *Energy Strategy Reviews* 20 (2018), pp. 26–34. ISSN: 2211-467X. DOI: 10.1016/j.esr.2018.01.003.