Market Simulation Tool (MAST) in the Julia Programming Language

Luigi Teola

24 July 2025

Introduction

This is a basic guide on how to use MAST coded in the Julia Programming Language. The code in Julia was mostly based on the previous work by Shariq Riaz, Archie C. Chapman, and Gregor Verbič (2017) with added functionality by Luigi Teola, Mohsen Aldaadi, and Muhammad Adnan. Julia is a high-performance programming language designed for technical computing. This guide provides step-by-step instructions to install and run MAST.

1 Julia Installation

1.1 For macOS

1. **Download Julia**
   Visit `https://julialang.org`, go to the "Downloads" section, and get the latest macOS version (`.dmg` for ARM/Intel).

2. **Install Julia**
   Open the downloaded `.dmg` file and drag the Julia icon to Applications. Eject the `.dmg` after installation.

3. **Run Julia**
   Launch Julia from Applications or run `julia` in Terminal (if added to PATH).

1.2 For Windows

1. **Download Julia**
   Visit `https://julialang.org`, navigate to "Downloads," and get the latest Windows `.exe` installer.

2. **Install Julia**
   Run the installer, follow the prompts, and select "Add Julia to PATH" (recommended).

3. **Run Julia**
   Open Julia from the Start menu or run `julia` in Command Prompt/PowerShell (if added to PATH).

4. **Test the Installation**
   In Julia REPL, type `1 + 1` and press Enter. If it returns `2`, the installation is successful. Explore `https://julialang.org/learning/` for tutorials.

2 Solver

The default solver is Gurobi which can be used for free using an academic license. It can work with other solvers like HiGHS and CPLEX, but for now we will use Gurobi.

1. **Visit the Gurobi Website**
   Go to the official Gurobi website: `https://www.gurobi.com`. You will need an account to access downloads. Sign up or log in as needed.

2. **Navigate to Downloads**

   - Once logged in, click on **Downloads** in the top menu or go directly to `https://www.gurobi.com/downloads/`.
   - Locate the **Gurobi Optimizer** section.

3. **Select Your Platform**

   - For macOS, choose the current version for **macOS Universal2**.
   - For Windows, choose the current version for **x64 Windows**.
   - Ensure the version matches your system's architecture (e.g., 64-bit).

4. **Download the Installer**
   Click the download link for your platform. Save the file to a known location (e.g., your Downloads folder).

5. **Obtain a License**

   - After downloading, you'll need a license to activate Gurobi.
   - Academic users: Visit
     `https://www.gurobi.com/academia/academic-program-and-licenses/`
     to request a free academic license. It is highly recommended to select **Named-User Academic** for single machine use valid for 1 year. Follow the instructions to generate a license file (e.g., `gurobi.lic`).
   - Save the license file to a secure location; you'll need it during installation or setup.

3 Initializing MAST

1. **Download** the file from Github by clicking the green ⟨⟩**Code** drop-down button in the upper-right pane, go to the **Local** tab, and click **Download ZIP**. Extract it to a known location in your machine

2. **Configuration file**
   Open the **configuration file** `config.json` using any IDE (e.g. VSCode) or any text editor and manipulate the parameters of your model. Make sure to keep this file in the same directory. You can alter the default values of this file. The parameters, their description, and default values are seen here:

Parameter	**Description**	**Default Value**
data_directory	folder where the model files are located	models
model_file	XLSX file containing generator, bus, branch, and storage data	Test.xlsx
demand_directory	folder where demand traces are found	ESOO_2013_Load_Traces
solar_directory	folder where solar traces are found	Solar_Traces
wind_directory	folder where wind traces are found	0910_Wind_Traces
trace year, month, day	selected traces to be used	year = 2020, month = 7, day = 1 (start day of most trace files)
horizon_days	total planning days to simulate	3
rolling_horizon_days	number of days used for rolling horizon, should be ≤ horizon_days	2
loss_factor	transmission losses	0.1 (10%)
reserve_margin	active power reserves of the system	0.1 (10%)
solver_name	optimization solver	Gurobi
mipgap	measure of how close the current solution is to the optimal solution	0.01 (1%)

3. **Model file**
   Open the **model_file** and here you can view, change, and add generator, bus, branch, and utility storage data to your model. Do not change the headings and make sure that all relevant data are written above the row that contains `END OF DATA`.

4 Using MAST

- **Running via command line**
  Once you have finished initializing MAST together with the configuration and model files, you can run it by typing

```
julia MAST.jl
```

in the Command Prompt (cmd), PowerShell, or Windows Terminal (Note: This is different from being inside a `Julia REPL` environment if you are in VSCode). Make sure that the model file is closed before running MAST to avoid any errors. During your first run, the code will download and install packages needed to run MAST.

- **Bus demand**
  The code produces sorted CSV files of demand that are sorted for use in the optimization problem. The sorted prosumer demand is found in `psmDemand.csv` while the sorted consumer demand is found in `csmDemand.csv`. Both files are inside the main directory and are overwritten after every run.

- **Viewing results**
  All results are stored in the folder `results` inside the main directory. The optimal decision variables (solution to the optimization problem) are stored in the default output file `decision_variables.csv`. Every new run will generate and save a new file inside the `results` folder.

- **Plotting results**
  Currently, the program outputs a plot of the model's total system generation per technology saved as `system_generation.png`. Every new run will also generate and save a new file inside the `results` folder.

- **More information about MAST**
  Read the pdf file MAST_Guide(March 2018) from Github for more information on the optimization problem, variables, and constraints.

Notes

This conversion of MAST to Julia is still in beta. If you have any questions or encounter any problems, please email me at **luigi.teola@sydney.edu.au**.