



Input-Output analysis and modelling with MARIO

Hands-on 10 – Implementing shocks & visualizing results

Please, be aware that all the supporting materials required for this hands-on session is available on Zenodo at the following link: <https://doi.org/10.5281/zenodo.8308514>

Please use the following citation for:

- **MARIO Software**

Mohammad Amin Tahavori, Lorenzo Rinaldi, & Nicolò Golinucci. (2022). SESAM-Polimi/MARIO: MARIO v0.1.0 (v0.1.0). Zenodo. <https://doi.org/10.5281/zenodo.5879382>



Learning outcomes

By the end of this exercise, you will learn how to:

- 1) Implement final demand and technology output mix changes in a SUT
- 2) Use the MARIO 'get_data' method to calculate matrices in non-baseline scenarios
- 3) Visualize results by scenario

Important requirement

Please make sure you have Microsoft Excel (or an equivalent alternative) installed on your PC.



Step 0: the Zenodo repository

All the supporting files for this and other Hands-ons and Lectures are available in the Zenodo repository associated to this course.

You find the repository at the following link: <https://doi.org/10.5281/zenodo.8308514>

Implement shocks into a SUT

In this Hands-on, we are going to implement shocks into a supply-use table (SUT). We are going to perform and compare the results of two shocks:

1. The first shock implies:
 - a. the change of the electricity production mix in China according to IEA projections in 2050
 - b. the increase in Li-ion batteries European final demand, according to IEA projections of battery-electric vehicles penetration in 2050.
2. The second shock implies to simulate the decrease of European population according to population projections in 2050

We'll move to the first shock once we parse the Excel table we exported at the end of Hands-on9.

```

"""
N.B: You can decide where to save your database
For the sake of keeping the material organized properly in the Zenodo repository,
we decided to save all the databases in the 'Databases' folder.
To do so, a small workaround is needed to get to a non-nested path and
contemporarily make it work indipendently of the user

N.B. #2: This will work if you downloaded the whole .zip folder contained
in the Zenodo repository and you did not changed the location of the files within it

N.B. #3: Set your current working directory to the script path first

"""
import os
path = "\\".join(os.getcwd().split("\\")[:-1])+"\\Databases\\Hands-on9.xlsx"

world = mario.parse_from_excel(path=path, table='SUT', mode='flows') # use function to parse SUT table in absolute values saved as .xlsx

```

Shock 1

As shown in Lecture10, to get the MARIO Excel template for implementing shocks, it is enough to call the ['get_shock_excel'](#) method. Just provide a proper path where to save the file.

```

# %% get MARIO Excel template to implement a shock
world.get_shock_excel(path='Hands-on10 - Shock batteries.xlsx')

```

By opening the "Filled" counterpart file available in the Zenodo repository, you can see that, regarding shock 1.a, it is enough to **update** the electricity **market share** in Chinese electricity production sectors according to a certain projection. This modification must be performed

in the **supply coefficients matrix (s)** which is addressable in the **z sheet** of the template by specifying *Activity* as row level and *Commodity* as column level.

	A	B	C	D	E	F	G	H
1	row region	row level	row sector	column region	column level	column sector	type	value
2	China	Activity	Production of electricity by coal	China	Commodity	Electricity	Update	0.0938
3	China	Activity	Production of electricity by gas	China	Commodity	Electricity	Update	0.0205
4	China	Activity	Production of electricity by nuclear	China	Commodity	Electricity	Update	0.0927
5	China	Activity	Production of electricity by petroleum and other oil derivatives	China	Commodity	Electricity	Update	0.0043
6	China	Activity	Production of electricity by solar photovoltaic	China	Commodity	Electricity	Update	0.3378
7	China	Activity	Production of electricity by wind	China	Commodity	Electricity	Update	0.2697
8	China	Activity	Production of electricity by biomass and waste	China	Commodity	Electricity	Update	0.0543
9	China	Activity	Production of electricity by Geothermal	China	Commodity	Electricity	Update	0.0091
10	China	Activity	Production of electricity by hydro	China	Commodity	Electricity	Update	0.1087
11	China	Activity	Production of electricity by solar thermal	China	Commodity	Electricity	Update	0.0045
12	China	Activity	Production of electricity by tide, wave, ocean	China	Commodity	Electricity	Update	0.0045
13								
14								
15								

In our case, we refer to the Announced Pledges Scenario projection reported in the 2022 IEA World Energy Outlook (<https://www.iea.org/reports/world-energy-outlook-2022>)

Concerning shock 1.b, we assumed to simulate the production of Li-ion batteries for electric vehicles (EVs) adoption as an increase in final demand of the Li-ion batteries commodity in EU27.

	A	B	C	D	E	F	G	H
1	row region	row level	row sector	column region	demand category	type	value	
2	EU27	Commodity	Li-ion batteries	EU27	Final demand	Update	1.8E+12	
3								
4								
5								
6								

Again, we referred to the 2022 IEA Global EV Outlook report, which forecast 150 million BEVs circulating in EU27 in 2050 (<https://www.iea.org/reports/global-ev-outlook-2022>). We assumed an average battery size of 80 kWh and an average price of batteries of 150 EUR/kWh, to get to a total final consumption of batteries equal to:

$$\text{Battery demand} = 150 \frac{\text{EUR}}{\text{kWh}} * 80 \frac{\text{kWh}}{\text{BEV}} * 150 \text{ Million BEVs} = 1.8 \text{ e12 EUR}$$

In the end, we implemented the shock via the MARIO '[shock_calc](#)' method, similarly to how we did in Lecture10 in the IOT case.

```
## Shock batteries: implementation
world.shock_calc(
    io='Hands-on10 - Shock batteries - Filled.xlsx',
    z=True,
    Y=True,
    scenario='Batteries and EE mix'
)
```

Shock 2.

In this case, we will decrease the European final consumption of all commodities to simulate the expected 1.2% EU27 population reduction (Source: [Eurostat](#)).

In this case, we need to introduce a useful feature to avoid the template filling procedure to be intensively time consuming. This feature is the **clusters** argument, which can be defined as a Python dictionary, containing a key for each *set* ('Activity', 'Commodity', 'Region'...) for which the user desires to set some clusters.

The value of each key must be an other Python dictionary, containing a list for each desired cluster to be defined for that key (or *set*).

To clarify, let's assume we want to define two clusters:

- The first includes all the commodities (called 'All')
- The second includes all the regions (called 'All' as well)

The proper way to do it is the following:

```
clusters = {
    'Commodity': {
        'All': world.get_index('Commodity'), # use get_index to get a list of all the commodities
    },
    'Region': {
        'All': world.get_index('Region'), # use get_index to get a list of all the regions
    },
}

world.get_shock_excel(path='Hands-on10 - Shock population.xlsx', **clusters)
```

Once the clusters dictionary is defined, just call the 'get_shock_excel' method to save another template to be filled, **adding '**clusters' as an additional argument.**

By opening the “Filled” counterpart file available in the Zenodo repository, you can see that **the clusters ‘All’ are now available for selection in the matrices sheets**, dramatically easing the filling process.

	A	B	C	D	E	F	G	H
1	row region	row level	row sector	column region	demand category	type	value	
2	All	commodity	All	EU27	Final demand	Percentage	-0.012	
3								
4								
5								
6								
7								
8								

The whole EU27 final demand (**including all regions and all commodities**) is now reduced by 1.2% (-0.012, implemented as **percentage change type**),

Clusters must be provided as argument also of the ‘shock_calc’ method:

```

%% Shock population: implementation
world.shock_calc(
    io='Hands-on10 - Shock population - Filled.xlsx',
    z=True,
    Y=True,
    scenario='Reduce population',
    **clusters,
)

```

The get_data method

To explore results, a paramount important method to use is the ‘**get_data**’ method. It allows to calculate **any matrix** in **any scenario** desired by the user.

Also, it is possible to **calculate differences between matrices of any scenario with respect to a reference scenario**. Such differences can be calculated as **simple subtractions** or as **relative** changes.

An example here is provided to calculate the relative variation of specific CO2 footprint between the “Batteries and EE mix” scenario (shock 1) and the baseline scenario ('base_scenario'). Some familiarity with **Pandas Dataframes and their 'loc' method** is recommended.

```
##% Result 1. Changes of batteries footprints
# Let's discover the get_data method

delta_f = world.get_data(
    matrices=['f'], # provide a list of matrices you desire to get. Specific footprints in this case
    scenarios=['Batteries and EE mix'], # specify a list of scenarios you want the matrices from
    base_scenario='baseline', # specify if you want to calculate the variation of the desired matrices with respect to the ones of another scenario
    type="relative", # specify if you desire a simple matrices difference ('absolute') of a 'relative' difference
)

##% Result 1. Now isolate the value of specific CO2 footprints of Li-ion batteries in EU27

delta_f['Batteries and EE mix'][0].loc[
    'CO2',
    ('EU27', 'Commodity', 'Li-ion batteries'),
]
```

Visualizing results

We are going to adopt the [plot_matrix](#) method, already shown in Lecture10, to plot the total variation in CO2 emissions in both shocks with respect to the baseline.

First, we decide to aggregate the Database, to reduce the number of sectors. You can check the aggregation performed by opening the 'Hands-on10 - Aggregation - Filled.xlsx' Excel file.

```
# Before plotting, we want to aggregate the activities and commodities
world.get_aggregation_excel("Hands-on10 - Aggregation.xlsx")

world.aggregate("Hands-on10 - Aggregation - Filled.xlsx", ignore_nan=True)
```

Secondly, we plot **matrix E** to check the CO2 emissions variation. We add the 'animation_frame' argument to add a slider showing a separate plot for each scenario.

Specify the *mode* argument equal to 'relative' in case you expect both positive and negative contributions.

```
world.plot_matrix(  
    matrix='E',  
    x = 'Region_to',  
    color = 'Activity_to', # not sectors"  
    base_scenario='baseline',  
    animation_frame = 'scenario', # add a slider on to compare scenarios  
    filter_Satellite_account = ['CO2'],  
    path=r'Hands-on10 - CO2 emissions.html',  
    mode='relative'  
)
```

An HTML plot file has been generated. By opening it and moving the slider on the bottom part to the “Batteries and EE mix” scenario (Figure 1), we can see that **the increase of Chinese ‘Electricity from fossil’ emissions are abundantly offset by a decrease of emissions related to the adoption of ‘Electricity from RES’**.

However, the production of batteries is **not localized only in China** and this lead to the increase in CO2 emissions in all the other regions, especially in the mining and electricity from fossil activities.

To **understand better** this result, you need to wonder whether any of the changes you applied via shock 1 involves the activation of mining activities. If you open the ‘Hands-on9 – Add activities – Filled.xlsx’ file, please check which are the commodities input of the ‘Manufacturing of Li-ion batteries’ activity. **You will see that the ‘ores’ commodities are often coming from many regions outside EU27.**

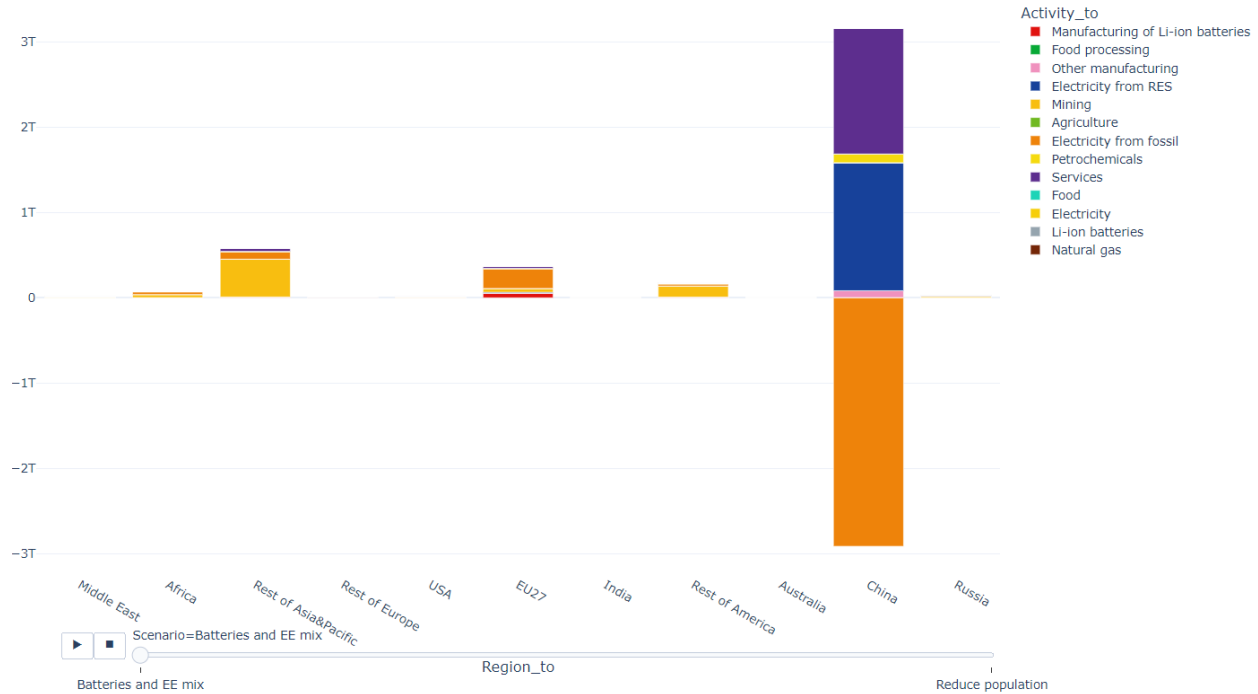


Figure 1. CO2 emissions variation in "Batteries and EE mix" with respect to baseline "scenario"

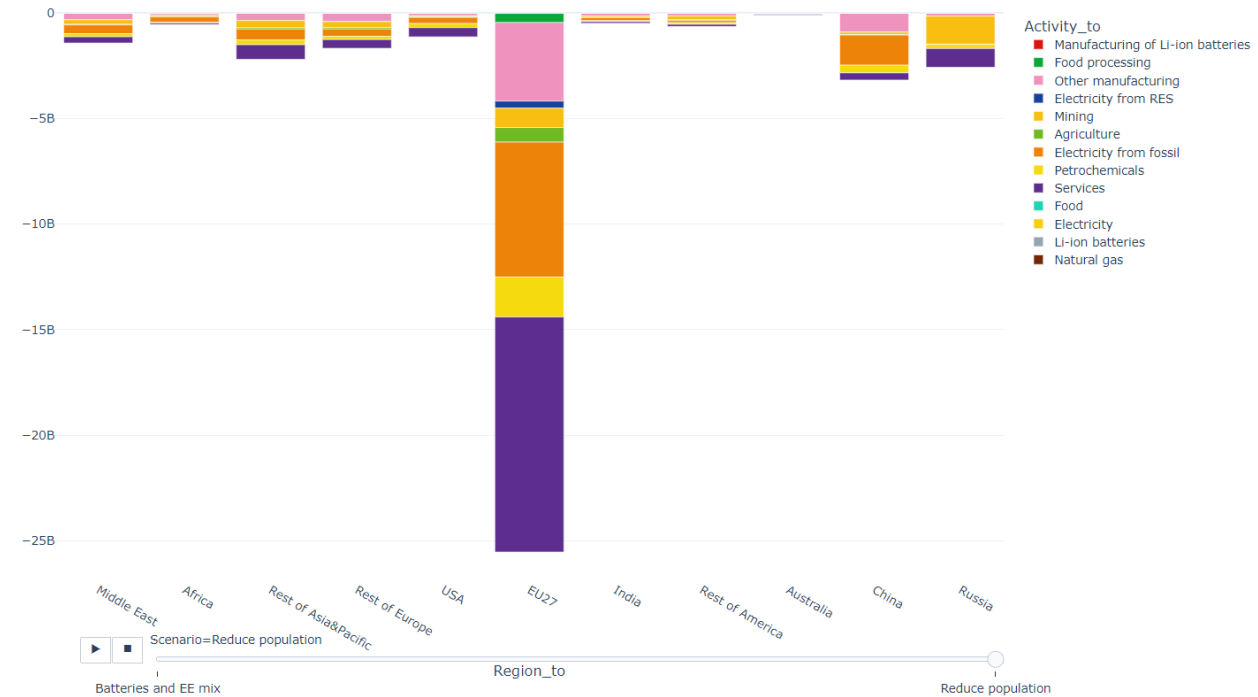


Figure 2. CO2 emissions variation in "Reduce population" with respect to baseline "scenario"



The other scenario ("Reduce population", move the slider to the right) lead to much reduced impact (look at the y-axis). By zooming in the interactive HTML chart, you can see that the CO₂ emissions of all the regions reduce, but **much less than in the previous shock**, meaning that 1.2% less people in Europe **would most probably be far from counterbalancing** the additional CO₂ emissions attributable to the expected European battery demand.



This document was updated on 01.11.2024