House of Energy Markets & Finance

# Open Source Stochastic European Energy Market Model (osE2M2s) 1.0

February 2025

Contributors: C. Weber, M. Bucksteeg, G. Blumberg, M. Breder, R. Broll, M. J. Radek, S. Spiecker, D. Swider

Chair for Management Science and Energy Economics, University of Duisburg Essen

Chair for Business Management and Energy Economics, FernUniversität in Hagen

UNIVERSITÄT DUISBURG ESSEN

Open-Minded

# Open Source Stochastic European Energy Market Model (osE2M2s) 1.0 Documentation

CHRISTOPH WEBER

Chair for Management Science and Energy Economics

House of Energy Markets and Finance

University of Duisburg-Essen, Germany

Universitätsstr. 12, 45117 Essen

Christoph.Weber@uni-due.de

www.hemf.net

MICHAEL BUCKSTEEG

Chair for Business Management and Energy Economics

Faculty of Business and Economics

FernUniversität in Hagen, Germany

Universitätsstr. 41, 58097 Hagen

Michael.Bucksteeg@fernuni-hagen.de

www.fernuni-hagen.de/energiewirtschaft

# 1 Introduction

The Open Source Stochastic European Energy Market Model (osE2M2s) is a linear optimization model designed for the long-term development of European electricity, hydrogen, and heat markets. It has been applied in several studies, most recently in Blumberg et al. (2022). Additional applications are documented in works by Swider and Weber (2007), Spiecker et al. (2013), Spiecker and Weber (2014), Bucksteeg et al. (2019), and Radek et al. (2024). Originally developed during the GreenNet project (2006), the model is implemented using the General Algebraic Modeling System (GAMS).

Unlike dispatch-only models, osE2M2s can determine optimal capacity expansion for renewable and conventional generation technologies, storage technologies, and other flexibility options. It optimizes multiple simulation years dynamically, starting with the generation and flexibility stack of the base year. Investments in intermediate years are added to the existing capacity, with the results of earlier simulations influencing subsequent years through a myopic foresight approach.

The model minimizes total system costs, including investment, fixed, and operational costs. Existing capacities cover at least their fixed and operational costs, while additional capacities must also account for annualized investment costs. The model focuses on planning the expansion of electrolyzers and biomass, nuclear, and hydrogen-fueled power plants, renewable energy plants, battery storage, and heat pumps. It also captures the effects and interactions of these technologies with others.

Key constraints include meeting electricity and heat demand in all time segments and market areas. Further details on model constraints can be found in Spiecker et al. (2013). The model applies a typical day approach with aggregated time segments to reduce computational complexity. Eight typical days (weekdays and weekend days from four representative months) represent a full year. Each day is divided into seven time segments based on electricity demand patterns, resulting in 56[1] time segments in total.

The model accounts for uncertainties in renewable energy input through stochastic recombining trees. A typical day is divided into four equal parts, with transitions between different renewable infeed nodes (high, medium, low) based on probabilities derived from historical weather data. This creates 96[2] possible nodes in total, which are assigned to time segments via assignment sets. Other model features include startup and shutdown costs, power plant availability, reserve provision, time-coupled storage optimization, and cross-border energy trading via net transfer capacities (NTCs).

Political constraints, such as coal or nuclear phase-outs, can be incorporated. $CO_2$ emissions are regulated either by a fixed price or an emissions cap. The fixed price approach does not guarantee decarbonization targets, whereas the emissions cap approach ensures compliance with the $CO_2$ price determined endogenously by the $CO_2$ constraint margin. To ensure supply security and account for the variability of renewable energy, a capacity constraint is enforced. This guarantees sufficient levels of dispatchable generation capacity (e.g., gas or hydrogen turbines) and considers

---

[1] Two typical days * four months * seven segments per day = 56

[2] Two typical days * four months * four parts per day * three possible nodes = 96

renewable energy based on installed capacity and minimum capacity factors or inflows. Dispatchable capacity must meet the internally calculated maximum demand.

This report documents version 1.0 of osE2M2s, the first open-source publication. The open-source version has been extensively revised and rewritten to reflect the current state of the model.

The document is organized as follows:

- Chapter 2: General model organisation, folder structure, and associated model files.
- Chapter 3: Model description, including an overview of the sets, parameters, variables, and model equations defining the energy system.
- Chapter 4: Typical day and stochastic approaches.
- Annexes: Detailed documentation of input and output data structures.

# 2 General model organisation

## 2.1 Folder structure overview

The basic model structure is outlined below. All key model files are described in the following subchapters.

| | |
|---|---|
| **Run Folder** | Base folder |
| └── Input | Subfolder |
| └──── Inc_database | Data input as .inc files |
| └──── inc_structure | Input files for data reading as .inc files |
| └── Output | Sub folder |
| └──── GDX | stored model results as .gdx file |
| └── Model | osE2M2s.gms |

## 2.2 Files defining sets and parameters

Files defining GAMS definition of sets and parameters are in folder *"Input\inc_strucutre"*.

*Table 1: Files containing GAMS definition of sets and parameters.*

| File | Description |
|---|---|
| avail_param.inc | Definition of parameters containing hourly time series |
| cap_param.inc | Capacity parameters |
| co2_param.inc | CO2 specific parameters |
| Costs_param.inc | Cost specific parameters |
| demand param.inc | Demand specific parameters |
| eff_param.inc | Efficiency parameters |
| node_time_sets.inc | Sets for typical days and stochastic representation |
| Prob_freq_param.inc | Parameters for typical days and stochastic representation |
| Region_plant_sets.inc | Sets for plants and geoscope |
| tech.inc | Technology parameters |

## 2.3 Data input files for sets and parameters

There are roughly 152 files *"Set *.inc"* or "Par *.inc" as listed in Appendix A: Set data input files. These files specify the set and parameter elements in the GAMS model. The table in Appendix A: Set data input files has the columns shown in Table 2.

Note, that input parameters starting with "b" are defined over all considered simulation years (set byear). These parameters are passed to the respective input parameter of the optimization problem, e.g. bco2_bound (byear) → co2_bound or b_fuel_price (byear, primary_energy, zone) → fuel_price (primary_energy, zone).

*Table 2: Content of table in Appendix A: Set data input files.*

| Heading | Content | Example |
| --- | --- | --- |
| File | Name of the JMM input file. | Set Countries.inc |
| GAMS | SET(s) or PARAMETER(s) in the GAMS model defined by this file. | Set country |
| Description | A short description of the data. | contains the model countries for electricity supply |
| Type | Type of data. | Geography |
| Unit | - | - |

## 2.4 Output generation

For each output parameter, variable or marginal value of an equation an “out_*”-parameter at the end of the GAMS model code writes the values of the output parameter, variable or marginal value in question to the output gdx-file. The content of the output files is described in Appendix C: Output generation files.

## 2.5 Model configuration and features

On top of the GAMS model code, the Input and Output path and Solver options can be stated.

So far, there is one switch for controlling the hydrogen market clearing based on yearly or time-specific demand series.

$SetGlobal *h2_yearly YES*

A switch value of YES results in a yearly hydrogen market-clearing condition.

# 3 Model description

## 3.1 Model overview

The model is grounded in the well-established principle that a competitive market, when functioning effectively, achieves the same outcomes as system optimization conducted by an all-knowing central planner. It assumes that the market operates efficiently to balance supply and demand, akin to the "invisible hand" concept. In the short term, markets are expected to maximize social welfare, ensuring cost-effective power plants fulfill electricity and heat demand. Since short-term demand is assumed to be inelastic, a cost-minimization approach can be employed. These costs include capital repayments, fixed annual expenses, and variable costs, broken down into fuel costs, emissions allowance expenses, other variable charges, and start-up costs. Ultimately, energy prices and payments for system services are derived from the shadow prices of various constraints.

The model is implemented in the General Algebraic Modeling System (GAMS) and uses CPLEX as its solver. It is formulated as a linear stochastic program, incorporating multiple time steps (typical days and hours), geographic regions, and all relevant stakeholders. A single optimization covers an entire year to account for seasonal variations in production, such as temperature-driven heat and electricity demand and managing large hydro reservoirs. The year is divided into four seasons—winter, spring, summer, and autumn—each represented by a typical working day and a typical non-working day to reduce computational complexity. These days are further divided into seven time segments to reflect demand fluctuations and renewable energy production patterns.

The first time segment covers six hours, the second five hours, and the third corresponds to the noon peak hour. After the peak hour, the remaining 12 hours are split into four equal segments. Shorter time segments are used during periods of higher demand or solar generation variability, while longer segments are applied at times of greater stability, such as nighttime or early morning. In total, the model encompasses eight typical days with seven time segments each.

A stochastic framework captures nearly the entire range of renewable energy output despite using aggregated time segments. This method enables the modelling of thermal power plant operations, including part-load operations and start-up decisions. Currently, the model includes around 100 power plant categories, differentiated by primary energy source, vintage, and technology. Efficiencies vary based on these factors, and additional technical constraints are implemented for specific technologies, such as steam turbines, gas turbines, combined-cycle plants, and various types of CHP plants. Availability also depends on technology type and time of year.

The model provides a detailed simulation of CHP plant operations, considering regional electricity load profiles and sub-regional heat load profiles. Each region may consist of several sub-regions, with power plants optimizing their operations based on these profiles in a given region. Heat can only be delivered within a power plant's sub-region, while electricity demand can be met across the broader electricity region.

The model endogenously determines optimal power plant operations, transmission line utilization within net transfer capacity (NTC) limits, and investments in new generation capacity. Investment decisions for new power plants and CHP units are also endogenous, constrained by factors such as nuclear phase-out policies or limited lignite resources. Within these limits, investments are cost-driven, replacing older plants when they become economically viable. New CHP plants are built if heat can be sold and their costs are lower than the opportunity costs of electricity and heat generation.

This investment process aligns with the Peak Load Pricing approach, where annual full-load hours are a key determinant of technology choice. Decision-making assumes myopic expectations, and optimization problems for successive years are solved in a dynamic, recursive sequence. Prices are calculated from the shadow prices of demand constraints in the optimization model, with costs being the primary influencing factor.

When interpreting shadow prices, it is essential to note that for technologies operating below capacity, the marginal cost of meeting additional demand equals the variable cost. However, when capacity limits are reached, the marginal cost includes a shadow price for capacity. Over a year, these shadow prices ensure that operational facilities recover their variable costs and fixed operating costs.

## 3.2 Objective function and restrictions

An objective function and several constraints define the model. To improve readability, we state the LP formulation by default wherever possible. Changes to the constraint in the MIP configurations are defined in the explanatory text. Where the MIP configuration uses completely different constraints, these are explicitly stated and explained.

### 3.2.1 Objective function

The model determines the marginal generation costs as a function of available generation and transmission capacities, primary energy prices, plant characteristics, and actual electricity demand. Also the impact of hydro-storage and start-up costs as well as endogenous investment decisions are taken into account. The principle of the model is cost minimization in the power network. The deterministic objective function to be minimized can be written as:

$$Total\ Cost = var\_cost\_opr + var\_cost\_startup + var\_fix\_cost\_irr + v\_fix\_cost\_sunk + var\_fix\_cost\_rev + var\_cost\_trans + var\_cost\_trans\_h2 + v\_cur\_cost + v\_h2\_cost + var\_cost\_co2$$

- **Variable Operating Costs** (eq_var_cost_opr)

This equation sums up the objective function's variable operating cost (var_cost_opr). The equation accounts for the different technological characteristics of power plants without CHP extraction-condensing and with CHP extraction-condensing, as well as heat pumps.

- **Startup Costs** (eq_var_cost_startup)

This equation summarises the objective function's variable startup cost (var_cost_startup).

- **Transport Costs** (eq_var_cost_trans)

This equation summarises the objective function's variable power transmission costs (var_cost_trans).

- **Transport Costs for hydrogen** (eq_var_cost_trans_h2)

This equation summarises the objective function's variable H2 transmission costs (var_cost_trans_h2).

- **CO2-Costs** (eq_var_cost_co2)

This equation summarises the objective function's variable costs of $CO_2$ emissions determined by the emissions from burning fossil fuels and $CO_2$ price (var_cost_co2).

- **Irreversible fixed costs incurred in the first year** (eq_fix_cost_irr)

This equation summarises the objective function's irreversible fixed costs incurred in the first year (var_fix_cost_irr).

- **Irreversible fixed costs attributed to the remained lifetime except the first year** (eq_fix_cost_sunk)

This equation summarises the objective function's sunk costs (v_fix_cost_sunk).

- **Reversible fixed costs dependent on the installed capacity** (eq_fix_cost_rev)

This equation summarises the objective function's reversible fixed costs, such as fixed operation costs (var_fix_cost_rev).

- **Curtailment-Costs** (eq_cur_cost)

This equation summarises the objective function's costs of curtailing renewable energy sources (v_cur_cost).

- **Hydrogen import costs from third countries** (eq_cost_import_h2)

This equation summarises the objective function's H2 import cost from third countries (outside of Europe) (v_h2_cost).

### 3.2.2 Restrictions

- **eq_supply**

This equation handles the power supply based on monthly availability.

- **eq_supply_heat**

This equation handles the heat supply based on monthly availability.

- **eq_supply_backpressure**

This equation handles the Power supply constraints for backpressure CHP units (IGBACKPR).

- **eq_demand_el**

This equation handles the demand balance for electricity.

- **eq_demand_heat**

This equation handles the demand balance for heat.

- **eq_demand_h2**

This equation handles the demand balance for hydrogen.

- **eq_demand_heatpump**

This equation handles the power demand constraint for heat pumps (IGHEATPUMP).

- **eq_demand_pth2**

This equation handles the power demand constraint for power-to-hydrogen units.

- **eq_demand_ptm**

This equation handles the power demand constraint for power-to-methane units.

- **eq_max_demand_el**

This equation determines the maximum electricity demand of each zone.

- **eq_supply_extraction_2**

This equation handles the power supply constraints for extraction-condensing CHP units (IGEXTRACTION).

- **eq_supply_energy_total**

This equation handles the total energy a CHP unit supplies, converted to the equivalent electricity supply.

- **eq_supply_wind_onshore**

This equation handles the onshore wind power supply based on the natural wind availability and capacity availability.

- **eq_supply_wind_offshore**

This equation handles the offshore wind power supply based on the natural wind availability and capacity availability.

- **eq_supply_PV**

This equation handles the solar power supply based on the natural radiation availability and capacity availability.

- **eq_cap**

This equation handles the capacity restriction in the simulated year, which is the sum of the installed capacity in the reference year plus invested capacity during the simulated periods.

- **eq_cap_max**

This equation handles the maximum exogenous capacity per fuel and zone.

- **eq_cap_max_wind**

This equation handles the exogenous maximum capacity restriction per wind technology.

- **eq_cap_startup**

This equation handles the calculation of startup capacity.

- **eq_transpo_CF**

This equation handles the restriction for electricity transmission capacities.

- **eq_transpo_CF_h2**

This equation handles the restriction for hydrogen transmission capacities.

- **eq_MaxChargePower**

This equation handles the capacity restriction for pumping energy.

- **eq_MaxChargePower_Sim**

This equation handles the maximum simultaneous capacity restriction for the charging of electric vehicles.

- **eq_MaxChargePower_ptg**

This equation handles the capacity restriction for the electricity consumption of electrolysers.

- **eq_MaxDischargePower**

This equation handles the bound for electricity production from daily storages (IGELECSTORAGE, especially E-mobility).

- **eq_MaxVolumeBATT**

This equation handles the upper bound for loading (pumping) energy into storage through (available) storage volume (BATT_STO).

- **eq_pump_standing_pos**

This equation handles the pumped power, which can be used for standing or spinning positive reserve. Thus, this equation ensures separation.

- **eq_supply_river**

This equation handles the capacity restriction for hydro run-of-river plants.

- **eq_resvr_annual**

This equation handles the formation of the reservoir level for annual storage.

- **eq_resvr_daily**

This equation handles the formation of reservoir level for IGELECSTORAGE (daily storages like pumping storages or E-mobility).

- **eq_resvrmax_annual**

This equation handles the maximum reservoir level of annual storage.

- **eq_resvrmin_annual**

This equation handles the maximum reservoir level of annual storage.

- **eq_MaxVolume**

This equation handles the upper bound for loading (pumping) energy into storage through (available) storage volume (IGELECSTORAGE).

- **eq_prod_plant_ub**

This equation handles the upper bound for the production of power plants except for chp IGEXTRACTION plants.

- **eq_prod_plant_ub2**

This equation handles the upper bound for the production of extraction-condensing CHP plants (IGEXTRACTION).

- **eq_prod_plant_ub3**

This equation handles the upper bound for the production of heat boilers.

- **eq_prod_plant_ub4**

This equation handles the lower bound for the production of heat boilers.

- **eq_prod_plant_ub5**

This equation handles the bound for electricity production of PTG.

- **eq_prod_plant_ub_VRE**

This equation handles the constraint for wind and pv power production.

- **eq_prod_plant_lb**

This equation handles the lower bound for the production of power plants except for extraction-condensing CHP plants (IGEXTRACTION).

- **eq_BanVehicle2Grid**

If this equation is used, v_production (discharging to the grid) from E-mob is denied.

- **eq_reserve_cap_spinningPos**

This equation handles the incremental spinning reserve capacity (primary and secondary).

- **eq_reserve_cap_spinningNeg**

This equation handles the decremental spinning reserve capacity (primary and secondary).

- **eq_reserve_cap_standingPos**

This equation handles the incremental standing reserve capacity (tertiary).

- **eq_reserve_cap_standingNeg**

This equation handles the decremental standing reserve capacity (tertiary).

- **eq_reserve_cap**

This equation ensures an adequate level of installed capacity to maintain the security of supply.

- **eq_pump_onlyPump**

This equation ensures that only technologies which could charge/pump to do so.

- **eq_pump_standing_pos_onlyPump**

This equation ensures that this reserve type is only for technologies that could charge/pump.

- **eq_pump_standing_neg_onlyPump**

This equation ensures that this reserve type is only for technologies that could charge/pump.

- **eq_prod_nucl**

This equation handles the upper bound nuclear production.

- **eq_cap_nucl**

This equation forces nuclear and lignite to produce constant electricity within a month.

- **eq_cap_coal**

This equation forces hard coal to constant electricity production within a month.

- **eq_co2_bound**

This equation handles the $CO_2$ bound for the modelled geographical scope (e.g. Europe).

- **eq_co2_bound2**

This equation can handle a country specific $CO_2$ bound differently (e.g. Germany).

## 3.3 Renewables stochastic

The representation of stochastic processes in long-term system modelling is crucial as renewable energy integration increases. With the inclusion of variable and uncontrollable energy sources, system operators must manage the risk of rapid fluctuations in renewable generation. Stochastic inputs necessitate sufficient system flexibility, a factor not fully captured by deterministic planning tools. As a result, deterministic models may underestimate the required flexibility, leading to insufficient capacity to handle intermittent generation and compromising system security. On the other hand, a stochastic model identifies the most cost-effective dispatch and ensures that the system can accommodate significant variations in renewable energy output at the lowest possible cost.

In this model version, a recombining tree approach addresses the intermittency of wind and solar energy generation, effectively extending beyond the limitations of typical time segments. Rather than relying on a single operational mode for each of the 56 time segments, the model considers alternative modes based on the stochastic states of renewable energy sources (RES) generation. The typical time segments are aggregated into four stages per typical day, each lasting six hours. Each stage corresponds to specific time segments: the first stage covers the first time segment, the second stage includes the second and third time segments, and so on.

Three possible branches (nodes) are defined for each stage, representing high, medium, and low RES output. The model uses a recombining structure to prevent exponential growth in the number of nodes (see Fig. 1). This approach involves approximations of state variables. Still, studies by Küchler and Vigerske (2007) demonstrate that this method provides a consistent approximation to the full stochastic model. Similarly, Spiecker et al. (2013) confirmed the consistency between realized and approximated RES outputs.

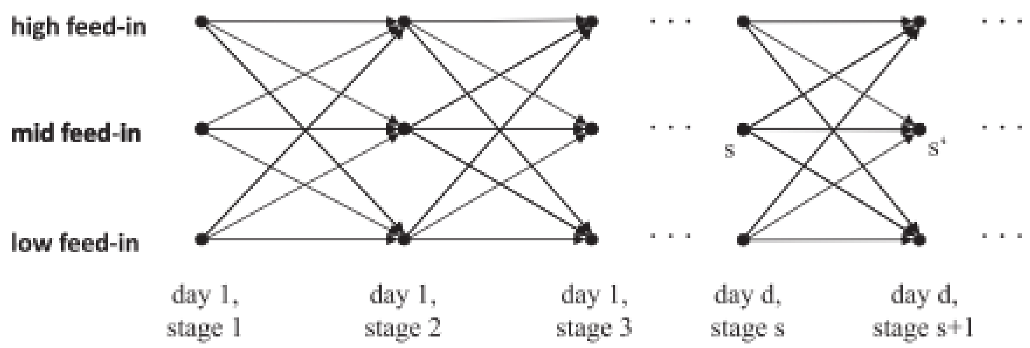


**Fig. 1.** Application of recombining decision tree.

The probabilities of the nodes and the corresponding transition probabilities are derived using cluster analysis of historical wind and solar generation data. First, the historical time series for different regions are mapped to the defined stages. Then, cluster analysis (following MacQueen, 1967) is used to determine three stochastic states or nodes for each stage. The time segments linked to each node (cluster) are counted and compared to the total time segments for that stage, yielding the probability $\psi_{s,n}$ for each node at a given stage.

Transition probabilities $\tau_{s \rightarrow s+1, n \rightarrow n'}$ describe the likelihood of moving from one node to another between stages. For typical days, these probabilities also account for transitions between weekdays and weekends. These probabilities are calculated by counting historical occurrences and comparing them to the total number of relevant time segments.

Finally, the availability of wind and solar energy in each node is determined as the average availability during the historical time segments associated with that node. This approach ensures a realistic representation of stochastic RES generation within the model.

## Appendix A: Set data input files

| File name | Gams set | Description |
|---|---|---|
| Set b_Plant_HeatRegio New.inc | b_inv_plant_regio(byear, power_plant, heat_regio) | Contains all the power plant types which could be invested in a certain region |
| Set b_Plant_zone Exist.inc | b_exist_plant_zone(byear,power_plant, zone) | First active plant class in a zone |
| Set bregio.inc | bregio | Contains all the electricity regions and heating regions |
| Set Countries.inc | country | Contains the model countries for electricity supply |
| Set Day_Segment.inc | day_segment | Determines time segments one day is divided into in the model |
| Set Day_Type.inc | day_type | Describes the characteristic days considered in the model |
| Set h2 import zones.inc | h2_import_zones(bregio) | Contains zones that can import h2 from third countries |
| Set heatregio.inc | heat_regio(bregio) | Contains heating regions, there allocation to the elctricity regions is defined in the parameter heatregio_in_regio |
| Set heatregio_co2_subset1.inc | heat_regio_co2_subset1(heat_regio) | Contains heating regions, there allocation to the elctricity regions is defined in the parameter heatregio_in_regio |
| Set heatregio_co2_subset2.inc | heat_regio_co2_subset2(heat_regio) | Contains only heating regions for one country |
| Set heatRegions - Country.inc | heatregio_in_country(heat_regio, country) | Heat regions in a country |

| | | |
|---|---|---|
| Set heatRegions_in_zone.inc | heatregio_in_zone(heat_regio, zone) | Heat regions in a zone |
| Set Hour.inc | hour | Represents the smallest time unit simulated in the model |
| Set Hour_Segment.inc | hour_segment(hour, day_segment) | Relates set 'hour' tset 'day_segment' |
| Set Hour_Time.inc | hour_time(hour, time) | Associates a specific hour with a time |
| Set Line Exist CF.inc | b_exist_line_CF(byear,zone, zzone) | Existing transmission line between the adjacent regions |
| Set Line Exist H2.inc | b_exist_line_CF_h2(byear,zone, zzone) | Existing h2 transmission line between the adjacent regions |
| Set Line Exist.inc | exist_line(trans_line,zone, zzone) | Existing transmission line between the adjacent regions |
| Set Month.inc | month | Describes the aggregation of months in the model |
| Set Month_Node.inc | month_node(month, node) | Represents the node to which a specific month is being assigned |
| - | month_succ | Succeeding order of months |
| Set Month_Time.inc | month_time(month, time) | Associates a specific month with a time |
| Set Node.inc | node | Defines the type of node |
| Set plant exist.inc | power_plant | Contains all the power plant types to be modelled |
| Set plant new.inc | power_plant | Contains all the power plant types to be modelled |
| Set plant_fuel exist.inc | fuel(power_plant,primary_energy) | Assigns thermal power plant types to used fuels |
| Set plant_fuel new.inc | fuel(power_plant,primary_energy) | Assigns thermal power plant types to used fuels |
| Set plant_type.inc | plant_type | Type of power plant groups |
| Set Power_Plant_Type.inc | power_plant_type(power_plant, plant_type) | Combination of power plant types and the region which possesses such ones<br>Associates a specific power plant with its type |

| | | |
|---|---|---|
| Set primaryenergy.inc | primary_energy | Set of primary energies (fuels) |
| Set Segment_Node.inc | segment_node(day_segment, node) | Assigns a specific day segment to a node |
| Set simYears.inc | simyear(byear) | Set of simulated years which are executed consecutively |
| Set Time.inc | time | Dates |
| Set Type_Node.inc | daytype_node(day_type, node) | Represents the node to which the day type is being assigned. |
| Set Type_Time.inc | daytype_time(day_type, time) | Associates a specific day type with a time |
| Set Water_Scen.inc | water_scen | Determines how many water scenarios are considered in the model |
| Set Wind_Scen.inc | wind_scen | Determines how many wind scenarios are considered in the model |
| Set Zone - Country.inc | zone_in_country(zone, country) | Zones in a country |
| Set zone_ptdf.inc | zone(bregio) | Contains the model regions for electricity supply |

## Appendix B: Parameter data input files

| GAMS Parameter | Filename as .inc | Description |
|---|---|---|
| annuity(power_plant) | - | Annuity of the invested plants |
| availability (power_plant, month) | Par availability exist.inc, Par availability new.inc | Availability of power plants |
| b_co2_capture_fct (byear, power_plant) | Par CO2_Capture_exist.inc, Par CO2_Capture_new.inc | Capture rate CO2 |
| b_demand_clusterEmob (byear, power_plant, heat_regio) | Par DemandClusterEmob.inc | Overall demand of e-mobility cluster (power_plant) in heatregion and simyear |
| b_demand_emob_fix (byear, time, bregio, product) | Par demand_emob_fix.inc | Annual time resolution for electricity demand of fixed e-mob demand (dumb charging) |

| | | |
|---|---|---|
| b_fill_level_max (byear, power_plant, bregio) | Par fill_level_max exist.inc | The maximal possible fill level of hydro storages |
| b_fuel_price (byear, primary_energy, zone) | Par b_fuel_price.inc | Predicted fuel prices in the basis year |
| b_h2_costs_import (byear) | Par b_h2_costs_import.inc | Import price for hydrogen |
| b_heat_dem_max (byear, bregio) | Par heat_dem_max.inc | Max hourly demand for heat in each region |
| b_iArrive (time, power_plant, byear) | Par iArriveEmob.inc | Proportional amount of arriving cars of a e-mobility cluster (power_plant) in a timestep |
| b_idemand (time, power_plant, byear) | Par iDemandEmob.inc | Proportional demand of a e-mobility cluster (demand_clusteremob) in a timestep |
| b_iLeave (time, power_plant, byear) | Par iLeaveEmob.inc | Proportional amount of leaving cars of a e-mobility cluster (power_plant) in a timestep |
| b_iLoadPoss (time, power_plant, byear) | Par iLoadPossEmob.inc | Proportional amount cars of a e-mobility cluster (power_plant) which are able to load in a timestep |
| b_iLoadSimultaneity (time, power_plant, byear) | Par iLoadSimultaneity.inc | Proportional amount cars of a e-mobility cluster (power_plant) which are able to load in a timestep simultaneously |
| b_inflow_annual_storage (byear, node, time, power_plant, bregio) | Par inflow_annual_storage.inc | Total inflow |
| b_inflow_run_river (byear, node, time, power_plant, bregio) | Par b_inflow_run_river.inc | Maximum water inflow of the river plants |
| b_max_cap_nuclear (byear, country) | Par max_cap_nuclear.inc | Maximum capacity nuclear |
| b_max_cap_wind_off (byear, country) | Par max_cap_wind_off.inc | Maximum capacity wind offshore |
| b_prob_node (byear, node) | Par prob_node.inc | Probability from the state at one day segment |
| b_prob_node_trans (byear, node, node1) | Par Prob_trans.inc | Transition probability from the state at one day segment to the state at the succeeding day segment |
| b_PV (byear, bregio, node, time) | Par PV.inc | Natural sun irradiance intensity, given as a proportion in the installed pv capacities |
| b_wind_offshore (byear, bregio, node, time) | Par wind_offshore.inc | Natural wind intensity offshore, given as a proportion in the installed wind power capacities |
| b_wind_onshore (byear, bregio, node, time) | Par wind_onshore.inc | Natural wind intensity onshore, given as a proportion in the installed wind power capacities |
| bcap_ref (power_plant, heat_regio, byear) | Par cap_ref exist.inc | Total installed capacity in modelling regions in reference year |
| bcap_ref_heat (power_plant, heat_regio, byear) | Par cap_ref_heat exist.inc | Total installed capacity in modelling regions in reference year |

| | | |
|---|---|---|
| bcap_res_wat (power_plant, heat_regio, byear) | Par Cap_res wat.inc | Total installed capacity in modelling regions in reference year |
| bco2_bound (byear) | Par bco2_bound.inc | CO2-restriction for all single years |
| bco2_bound2 (byear) | Par bco2_bound2.inc | CO2-restriction for all single years |
| bmax_cap (byear, primary_energy, zone) | Par bmax_cap.inc | Restriction for max capacity depending on fuel type, year and zon |
| bmax_cap_wind (byear, plant_type, zone) | Par bmax_cap_wind.inc | Restriction for max capacity depending on plant type, year and zone |
| btrans_cap (zone, zzone, byear, month) | Par trans_cap_m.inc | Transmission capacity |
| btrans_cap_h2 (zone, zzone, byear, month) | Par trans_cap_m H2.inc | Transmission capacity for hydrogen |
| cap_n(power_plant, heat_regio) | - | Container for new power plant capacity during the simulation process |
| cap_n_sunk(power_plant, heat_regio) | - | Container for new power plant capacity during the simulation process, for the purpose of caculating sunk costs |
| cap_n_sunk_heat(power_plant, heat_regio) | - | Container for new power plant capacity during the simulation process, for the purpose of caculating sunk costs |
| cap_ref(power_plant, heat_regio) | - | Container for adapted power plant capacities |
| cap_ref_heat(power_plant, heat_regio) | - | Container for adapted power plant capacities |
| co2_bound | - | CO2 cap (e.g. Europewide) |
| co2_bound2 | - | CO2 cap per country |
| co2_capture_fct (power_plant) | - | Capture rate CO2 |
| co2_cost | - | CO2-costs for the year in the optimisation |
| co2emis(power_plant, heat_regio) | - | CO2 emission factor at normal performance |
| co2emis_min(power_plant, heat_regio) | - | CO2 emission factor at minimal performance |
| co2factor (primary_energy) | Par co2factor.inc | Fuel specific CO2 emission factor |
| cost_fix (power_plant) | - | Fix cost of existing plants dependent on the regions in simulation year |
| cost_fix0 (power_plant) | Par cost_fix exist.inc, Par cost_fix new.inc | Fix cost of existing plants dependent on the regions in base year |
| cost_inv (power_plant) | - | Investment cost of new plants in simulation year |
| cost_inv0 (power_plant) | Par cost_inv new.inc, Par cost_inv exist.inc | Investment cost of new plants in base year |

| | | |
|---|---|---|
| cost_misc (power_plant) | - | Miscellaneous costs in simulation year |
| cost_misc0 (power_plant) | Par cost_misc exist.inc, Par cost_misc new.inc | Miscellaneous costs in base year |
| cost_opr(time, power_plant,bregio) | - | Specific operating costs of existing plants when started up, will be calculated later |
| cost_opr_min(time, power_plant,bregio) | - | Specific operating costs of existing plants at minimum load, will be calculated later |
| cost_startup(time, power_plant, bregio) | - | Specific startup cost |
| cost_startup_abr (power_plant) | Par cost_startup_fuel exist.inc, Par cost_startup_fuel new.inc | Variable abrasion costs for startup process in simulation year |
| cost_startup_abr0 (power_plant) | Par cost_startup_abr exist.inc, Par cost_startup_abr new.inc | Variable abrasion costs for startup process in base year |
| cost_startup_fuel (power_plant) | Par cost_startup_fuel exist.inc Par cost_startup_fuel new.inc | Variable fuel costs coefficient for startup process |
| cost_trans (zone, zone) | - | Fix cost of existing plants dependent on the regions in simulation year |
| cost_trans_h2 (zone, zone) | - | Fix cost of existing plants dependent on the regions in base year |
| cost_trans0 (zone, zone) | Par trans_cost.inc | Fix cost of existing plants dependent on the regions in base year |
| cost_trans0_h2 (zone, zone) | Par trans_cost H2.inc | Fix cost of existing plants dependent on the regions in simulation year |
| cost_transpoCapInv (zone, zzone) | Par cost_transpoCapInv.inc | Degression rate |
| demand (time, bregio, product) | - | Demand for electricity and heat in each region filled from demand_Y |
| demand_clusterEmob (power_plant, heat_regio) | - | Overall demand of e-mobility cluster (power_plant) in heatregion and simyear |
| demand_emob_fix (time, bregio, product) | - | Electricity demand of fixed e-mob demand (dumb charging) filled from b_demand_emob_fix |
| demand_reserve(zone, product) | - | To be endogenous calculated reserve requirement |
| demand_Y (byear, time, bregio, product) | Par demand.inc | Annual demand for electricity and heat in each region |
| eff_plant (power_plant, heat_regio) | Par eff_plant exist.inc, Par eff_plant new.inc | Marginal efficiency of power plants started up |

| | | |
|---|---|---|
| eff_plant_min (power_plant, heat_regio) | Par eff_plant_min exist.inc, Par eff_plant_min new.inc | Efficiency of power plants kept online at their minimal performance |
| ex_foreign_trade (byear, time, bregio, product) | Par ex_foreign_trade.inc | Annual foreign trade for electricity in each region |
| fct_PQ_BP (power_plant) | Par fct_pq_bp exist.inc, Par fct_pq_bp new.inc | PQ factor of backpressures = power performance / heat performance |
| fct_PQ_Extr (power_plant) | Par fct_pq_extr exist.inc, Par fct_pq_extr new.inc | PQ factor of extractions = -power performance / heat performance |
| fill_level_max (power_plant, bregio) | - | The maximal possible fill level of hydro storages |
| fl_ratio_max (bregio, month) | Par fl_ratio_max.inc | Maximum fill level ratio depending on months |
| fl_ratio_min (bregio, month) | Par fl_ratio_min.inc | Minimum fill level ratio depending on months |
| freq_time (time) | Par freq_time.inc | Frequency of a time unit |
| fuel_price(primary_energy,zone) | - | Container fo fuel prices during the simulation process |
| gr_cost_inv | - | Change in investment costs over the years |
| h2_costs_import | - | Container for h2 import costs |
| heat_dem_max (heat_regio) | - | Max hourly demand for heat in each region |
| heat_demand_ratio (heat_regio) | Par heat_demand_ratio.inc | Proportion ratio of individual heat regions in the total heat demand in a modelling region |
| hour_resolution (time) | Par hour_resolution.inc | Hour resolution of a time unit |
| iArrive (time, power_plant) | - | Proportional amount of arriving cars of a e-mobility cluster (power_plant) in a timestep |
| idemand (time, power_plant) | - | Proportional amount of demand of a e-mobility cluster |
| iLeave (time, power_plant) | - | Proportional amount of leaving cars of a e-mobility cluster (power_plant) in a timestep |
| iLoadPoss (time, power_plant) | - | Proportional amount cars of a e-mobility cluster (power_plant) which are able to load in a timestep |
| iLoadSimultaneity (time, power_plant) | - | Proportional amount cars of a e-mobility cluster (power_plant) which are able to load in a timestep simultaneously |
| inflow_annual_storage (node, time, power_plant, bregio) | - | Total inflow |
| inflow_run_river (node, time, power_plant, bregio) | - | Maximum water inflow of the river plants |

| | | |
|---|---|---|
| ir (power_plant) | Par ir.inc | Discount rate |
| lifetime (power_plant) | Par lifetime new.inc, Par lifetime exist.inc | Lifetime of power plants |
| max_cap(primary_energy,zone) | - | Maximum capacity |
| max_cap_imports | - | Upper or lower bound for h2 imports |
| max_cap_nuclear (country) | - | Maximum capacity nuclear |
| max_cap_wind(plant_type, zone) | - | Maximum wind capacity depending on type (onshore or offshore) |
| max_cap_wind_off (country) | - | Maximum capacity wind offshore |
| min_load_fct (power_plant, heat_regio) | Par min_load_fct exist.inc, Par min_load_fct new.inc | Minimum performance factor |
| numyear (byear) | | Numerical value of year |
| plant_degr_fct (byear, inv_plant) | Par plant_degr_fct.inc | Degression rate |
| plant_degr_fct2 (byear, power_plant) | Par plant_degr_fct.inc | Degression rate |
| prob_node (node) | - | Probability from the state at one day segment |
| prob_node_trans (node, node1) | - | Transition probability from the state at one day segment to the state at the succeeding day segment |
| pump_cap (power_plant, heat_regio) | - | Restriction for max capacity |
| pump_cap_fct (power_plant, bregio, time) | Par pump_cap_fct exist.inc, Par pump_cap_fct new.inc | Capacity factor of pumping storages |
| pump_eff (power_plant, heat_regio) | Par pump_eff exist.inc, Par pump_eff new.inc | Efficiency factor of hydro storages |
| trans_cap_m (zone, zzone, month) | - | Transmission capacity |
| trans_cap_m_h2 (zone, zzone, month) | - | Transmission capacity for hydrogen |
| wind_offshore (bregio, node, time) | - | Natural wind intensity offshore, given as a proportion in the installed wind power capacities |
| wind_onshore (bregio, node, time) | - | Natural wind intensity onshore, given as a proportion in the installed wind power capacities |

## Appendix C: Output generation files

| GAMS Parameter | Description |
| --- | --- |
| out_cap(simyear, power_plant, bregio) | Capacity |
| out_cap_infeasible(simyear, node, time, product, bregio) | Capacity infeasible |
| out_cap_new(simyear, power_plant, zone) | New invested capacity |
| out_cap_onl(simyear, time, node, power_plant, bregio) | Capacity online |
| out_cap_startup(simyear, time, node, time1, node1, power_plant, bregio) | Capacity startup |
| out_co2emissions(zone,simyear) | Total $CO_2$ emission from all power plants |
| out_co2_price(simyear) | $CO_2$ prices |
| out_co2_price2(simyear) | $CO_2$ prices specific country from co2_bound2 |
| out_demand(simyear, time, zone, product) | Demand per time segment |
| out_demand_y(simyear, zone, product) | Demand yearly basis |
| out_el_price(simyear, node, time, zone) | Electricity prices |
| out_fill_level_h_exp(simyear, time, power_plant, bregio) | Filling level |
| out_heat_price(simyear, node, time, heat_regio) | Heat prices |
| out_h2_price(simyear, zone) | Hydrogen prices |
| out_h2_price(simyear, node, time, zone) | Hydrogen prices |
| out_production(simyear, time, node, power_plant, bregio, product) | Production |
| out_production_y(simyear, power_plant, bregio, product) | Yearly production per power plant and product |
| out_tcost(simyear) | Total costs |

| | |
|---|---|
| out_transpo(simyear, node, time, zone, zzone) | Transport of energy between zones |
| out_transpo_y(simyear, zone, zzone) | Calculated yearly transport of energy between zones |
| out_transpo_h2(simyear, node, time, zone, zzone) | Transport of h2 between zones |
| out_transpo_h2_y(simyear, zone, zzone) | Calculated yearly transport of h2 between zones |
| out_v_pump(simyear, node, time, power_plant, zone) | Pumping/charging power of each power_plant in zone |
| out_v_pump_y(simyear, power_plant, zone) | Yearly pumping/charging power of each power_plant in zone |
| out_v_pump_reserve(simyear, node, time, power_plant, zone) | Pumping reserves |
| out_v_cur(simyear, node, time, power_plant, bregio) | Output of variable v_curtailment |
| out_v_cur_y(simyear, power_plant, bregio) | Yearly output of variable v_curtailment |
| out_cur_cost(simyear) | Total curtailment costs |
| out_var_cost_opr(simyear,time,power_plant,heat_regio) | Operating costs |
| out_var_cost_startup(simyear,time,power_plant,heat_regio) | Startup costs |
| out_var_cost_trans(simyear,time,zone,zzone) | Transport costs |
| out_var_cost_trans_h2(simyear,time,zone,zzone) | Transport costs H2 |
| out_var_cost_co2(simyear,time,power_plant,heat_regio) | $CO_2$ costs |
| out_fix_cost_irr(simyear,power_plant,heat_regio) | Irreversible costs |
| out_fix_cost_sunk(simyear,power_plant,heat_regio) | Sunk costs |
| out_fix_cost_rev(simyear,power_plant,heat_regio) | Reversible costs |
| out_total_cost_reg(simyear, power_plant,heat_regio) | Regular total costs |
| out_demand_max(simyear, bregio, product) | Output parameter to safe max demands per demand group |

| | |
|---|---|
| cap_exist(simyear, power_plant, bregio) | Existing Capacity |
| val_spill(simyear, node, time,spill, power_plant, bregio) | Spillage |
| out_h2_import_costs(simyear) | Output parameter for total h2 import costs |
| out_v_import_h2_y(simyear, zone) | Output parameter for yearly h2 imports |
| out_v_import_h2(simyear, node, time, zone) | Output parameter for h2 imports |

## Appendix D: GAMS Equations

| GAMS Equation | Description |
|---|---|
| *-------------------------------- Objective function and cost terms ------------------------------------------- | |
| eq_total_cost | Objective function - describes the total cost over the modelling period - to be minimized |
| *eq_var_cost_opr* | Variable operation costs |
| *eq_var_cost_startup* | Variable startup costs |
| eq_var_cost_trans | Variable power transmission costs |
| eq_var_cost_trans_h2 | Variable H2 transmission costs |
| eq_var_cost_co2 | Variable costs of $CO_2$ emissions |
| eq_fix_cost_irr | Irreversible fixed costs incurred in the first year |
| eq_fix_cost_sunk | Irreversible fixed costs attributed to the remained lifetime except the first year |
| eq_fix_cost_rev | Reversible fixed costs dependent on the installed capacity |
| *----------------------------------- Restrictions -------------------------------------------- | |

| | |
|---|---|
| *eq_supply(node, time, month, power_plant, heat_regio)* | Power supply based on monthly availability |
| *eq_supply_heat(node, time, month, power_plant, heat_regio)* | Heat supply based on monthly availability |
| eq_supply_backpressure(node, time, power_plant, heat_regio) | Power supply constraints for IGBACKPR CHPs |
| eq_demand_heatpump(node, time, power_plant, heat_regio) | Power demand constraint for heat pumps (IGHEATPUMP) |
| eq_demand_pth2(node, time, power_plant, heat_regio) | Power-demand constraint for hydrogen |
| eq_demand_ptm(node, time, power_plant, heat_regio) | Power-demand constraint for methane |
| eq_supply_extraction_2(node, time, power_plant, heat_regio) | Power supply constraints for IGEXTRACTION CHPs |
| eq_supply_energy_total(node, time, power_plant, heat_regio) | Total energy supplied by a CHP converted to the equivalent electricity supply |
| eq_supply_wind_onshore(node, time, month, power_plant, heat_regio) | Onshore wind power supply based on the natural wind availability and capacity availability |
| eq_supply_wind_offshore(node, time, month, power_plant, heat_regio) | Offshore wind power supply based on the natural wind availability and capacity availability |
| eq_supply_PV(node, time, month, power_plant, heat_regio) | SUN power supply based on the natural radiation availability and capacity availability |
| eq_cap(power_plant, heat_regio) | Capacity restriction |
| eq_cap_max(primary_energy, zone) | Maximum exogenous capacity per fuel and zone |
| eq_cap_max_wind(plant_type, zone) | Capacity restriction for wind |
| *eq_demand_el(node, time, zone)* | Demand balance for power |
| eq_demand_heat(node, time, heat_regio) | Demand balance for heat |
| eq_demand_h2(zone) | Demand balance for hydrogen |
| *eq_demand_h2(node, time, zone)* | Demand balance for hydrogen |

| | |
|---|---|
| eq_transpo_CF(node, time, zone, zzone) | Restriction for transmission capacities |
| *eq_transpo_CF_h2(node, time, zone, zzone)* | Restriction for h2 transmission capacities |
| eq_MaxChargePower(node, time, month, power_plant, heat_regio) | Capacity restriction for pumping energy |
| eq_MaxChargePower_Sim(node, time,month) | Max simultaneous capacity restriction for charging of EV |
| *eq_MaxChargePower_ptg(node, time,month,power_plant, heat_regio)* | Capacity restriction for pumping of electrolysers |
| eq_pump_standing_pos(node,time,power_plant, heat_regio) | Pumped power could either be used for standing or spinning pos reserve --> this equations ensures separation |
| eq_supply_river(node, time, power_plant, heat_regio) | Capacity restriction for HYDR_ROR plants |
| eq_resvr_annual(water_scen, month, month1, power_plant, heat_regio) | Formation of reservoir level for annual storages |
| eq_resvr_daily(node, node1, time, time1, power_plant, heat_regio) | Formation of reservoir level for IGELECSTORAGE (daily storages like pumpstorage or emob) |
| eq_resvrmax_annual(water_scen, month, power_plant, heat_regio) | Maximum reservoir level of annual storages |
| eq_resvrmin_annual(water_scen, month, power_plant, heat_regio) | Minimum reservoir level of annual storages |
| eq_MaxVolume(node, time, power_plant, heat_regio) | Upper bound for loading (pumping) energy into storage through (available) storage volume (IGELECSTORAGE) |
| eq_prod_plant_ub(node, time, power_plant, heat_regio) | Upper bound for the production of power plants except chp IGEXTRACTION plants |
| eq_prod_plant_ub2(node, time, power_plant, heat_regio) | Upper bound for the production of chp IGEXTRACTION plants |

| | |
|---|---|
| eq_prod_plant_ub3(node, time, power_plant, heat_regio) | Upper bound for the production of heatboilers |
| eq_prod_plant_ub4(node, time, power_plant, heat_regio) | Bound for electricity production of heatboilers |
| eq_prod_plant_ub5(node, time, power_plant, heat_regio) | Bound for electricity production of PtG |
| eq_BanVehicle2Grid(node, time, power_plant, heat_regio) | If used than v_production (discharging to grid) from E-mob is denied |
| eq_MaxDischargePower(node, time, month, power_plant, heat_regio) | Bound for electricity production from daily storages (IGELECSTORAGE, especially E-mobility) |
| eq_prod_plant_lb(node, time, power_plant, heat_regio) | Lower bound for the production of power plants except chp IGEXTRACTION plants |
| eq_prod_plant_ub_VRE(node, time, power_plant, heat_regio) | Constraint for wind and pv power production |
| eq_cap_startup(node, node1, time, time1, power_plant, heat_regio) | Calculation of startup capacity |
| eq_reserve_cap_spinningPos(node, time, month, zone) | Incremental spinning reserve capacity (primary and secondary) |
| eq_reserve_cap_spinningNeg(node, time, month, zone) | Decremental spinning reserve capacity (primary and secondary) |
| eq_pump_onlyPump(node,time,power_plant, heat_regio) | Ensure only techs which could charge/pump do so |
| eq_pump_standing_pos_onlyPump(node,time,power_plant, heat_regio) | This reserve type is only for techs which could charge/pump |
| eq_pump_standing_neg_onlyPump(node,time,power_plant, heat_regio) | This reserve type is only for techs which could charge/pump |
| eq_reserve_cap(country) | Reserve restriction |
| eq_reserve_cap_heat(heat_regio) | Reserve restriction heat |
| eq_reserve_cap_standingPos(node, time, month, country) | Incremental standing reserve capacity (tertiary) |
| eq_reserve_cap_standingNeg(node, time, month, country) | Incremental standing reserve capacity (tertiary) |

| | |
|---|---|
| *eq_prod_nucl(node, time, country)* | Nuclear production restriction |
| eq_prod_wind_off(node, time, country) | Wind Offshore restriction |
| eq_co2_bound | $CO_2$ Bound |
| *eq_co2_bound2* | Country specific $CO_2$ bound (e.g. Germany) |
| eq_cap_nucl(node, node1, time, time1, month, power_plant, heat_regio) | Constant production restriction |
| *eq_cap_coal(node, time, time1, power_plant, heat_regio)* | Constant production restriction |
| eq_max_demand_el(zone,time,node) | Rquation determining the max electricity demand of each zone |
| *eq_MaxVolumeBATT(node, time, power_plant, heat_regio)* | Upper bound for loading (pumping) energy into storage through (available) storage volume (BATT_STO) |
| eq_cur_cost | Total cost of curtailment |
| *eq_cost_import_h2* | Total cost of h2 import |
| eq_min_gen(power_plant, heat_regio) | Minimum generation |
| eq_resvr_daily_h2(node, node1, time, time1, power_plant, heat_regio) | Fillling level restriction |
| *eq_MaxVolumeH2(node, time, power_plant, heat_regio)* | *Maximum reservoir level* |
| eq_MaxChargeH2(node, time, month, power_plant, heat_regio) | *Maximum charge* |
| eq_MaxDischargeH2(node, time, month, power_plant, heat_regio) | *Maximum discharge H2* |
| eq_pump_onlyPumpH2(node,time,power_plant, heat_regio) | *Only H2_STO can pump h2* |
| eq_import_constraint(zone) | Import constraint H2 |

| | |
|---|---|
| *eq_max_import* | *Max import H2* |

# Appendix E: GAMS Variables

**Free Variables**

| | |
|---|---|
| total_cost | Total system cost over the current simulated period (simyear) |
| var_cost_opr | Operating Costs over the current simulated period |
| var_cost_startup | Startup Costs over the current simulated period |
| var_cost_trans | Transport Costs (Electricity) over the current simulated period |
| var_cost_trans_h2 | Transport Costs (H2) over the current simulated period |
| var_cost_co2 | Costs for emissions |
| var_fix_cost_irr | Irreversible costs |
| var_fix_cost_rev | Reversible fixed costs |
| var_trans_cost | Transport costs |
| v_cur_cost | Total costs of curtailment |
| v_h2_cost | Total costs of h2 import |

**Positive Variables**

| | |
|---|---|
| v_fix_cost_sunk | Sunk costs |
| v_cap_new(power_plant,heat_regio) | Invested new capacities |

| | |
|---|---|
| v_cap(power_plant, heat_regio) | Installed capacities of existing plants |
| v_cap_heat(power_plant, heat_regio) | Installed capacities of existing plants of type heat |
| v_production(node, time, power_plant, heat_regio, product) | Production of all power plants in one region and to one node |
| v_production_energy(node, time, power_plant, heat_regio) | Energy supply of a power plant in one region and to one node including electricity and heat |
| v_cap_onl(node, time, power_plant, heat_regio) | Capacities kept online for electricity production |
| v_cap_startup(node, node1, time, time1, power_plant, heat_regio) | Capacities started up at the transition from a node the next one |
| v_transpo(node, time, zone, zzone) | Power transport from zone to zzone |
| v_transpo_h2(node, time, zone, zzone) | H2 transport from zone to zzone |
| v_pump(node, time, power_plant, heat_regio) | Used pumping capacity for the electricity storage |
| v_pump_standing_neg(node, time, power_plant, bregio) | Charging/pumping capacity reserved for (negative) standing reserve |
| v_pump_standing_pos(node, time, power_plant, bregio) | Charging/pumping capacity reserved for (positive) standing reserve |
| v_fill_level_mon(water_scen, month, power_plant, heat_regio) | Monthly fill level of the annual storages |
| v_fill_level_h(node, time, power_plant, heat_regio) | Hourly fill level of IGELECSTORAGE storages (daily storages like pumpstorage or e-mobility) |
| v_fill_level_heat(node, time, heat_regio) | Hourly fill level of heat storages |
| v_fuelusage(node,time,power_plant,heat_regio) | Fuelusage when producing energy |
| v_spill_ror(node, time, power_plant,heat_regio) | Penalty for spill for RES |
| v_spill_ror2(node, time, power_plant,heat_regio) | Penalty for spill for RES |
| v_demand_max(bregio,product) | Highest electricity demand in each region (fixed and flexible demand) |
| v_curtailment(node, time, power_plant, heat_regio) | Curtailment of fluctuating renewable energy sources |

| | |
|---|---|
| v_import_h2(zone) | Import of hydrogen |
| v_import_h2(node, time, zone) | Import of hydrogen |