GENeSYS-MOD

The Global Energy System Model

Model Version 3.0

User Manual - Quick Start Guide

Version 1.0

December 2020

Contents

1 Model Overview

The Global Energy System Model (GENeSYS-MOD) is an open-source energy system model, originally based on the Open-Source Energy Modeling System (OSeMOSYS) framework. While still part of the OSeMOSYS family of models, various aspects have been redesigned, expanded, or added.

GENeSYS-MOD is a linear program, minimizing total system costs. Energy demands are exogenously predefined and the model needs to provide the necessary capacities to meet them. The modeling framework is very flexible in its use cases. Based on research question and input data, calculations can be done from a household-basis to a global aggregation of regions.

Various applications of GENeSYS-MOD have been published in the last years, such as:

- Löffler, K. / Hainsch, K. / Burandt, T. / Oei, P. / Kemfert, C. / von Hirschhausen, C. (2017): Designing a Model for the Global Energy System - GENeSYS-MOD: An Application of the Open-Source Energy Modeling System (OSeMOSYS). Energies 2017, 10(10), 1468
- Burandt, T. / Löffler, K. / Hainsch, K. (2018): GENeSYS-MOD v2.0 - Enhancing the Global Energy System Model: Model Improvements, Framework Changes, and European Data Set. DIW Data Documentation 94
- Hainsch, K. / Burandt, T. / Kemfert, C. / Löffler, K. / Oei, P. / von Hirschhausen, C. (2018): Emission Pathways Towards a Low-Carbon Energy System for Europe - A Model-Based Analysis of Decarbonization Scenarios. DIW Discussion Paper 1745
- Lawrenz, L. / Xiong, B. / Lorenz, L. / Krumm, A. / Hosenfeld, H. / Burandt, T. / Löffler, K. / Oei, P. / von Hirschhausen, C. (2018): Exploring Energy Pathways for the Low-Carbon Transformation in India - A Model-Based Analysis. Energies 11(11), 3001
- Bartholdsen, H.-K. / Eidens, A. / Löffler, K. / Seehaus, F. / Wejda, F. / Burandt, T. / Oei, P.-Y. / Kemfert, C. / von Hirschhausen, C. (2019): Pathways for Germany's Low-Carbon Energy Transformation Towards 2050. Energies 12(15), 2988
- Burandt, T. / Xiong, B. / Löffler, K. / Oei, P.-Y. (2019): Decarbonizing China's energy system – Modeling the transformation of the electricity, transportation, heat, and industrial sectors. Applied Energy 255
- Löffler, K. / Burandt, T. / Hainsch, K. / Oei, P.-Y. (2019): Modeling the low-carbon transition of the European energy system - A quantitative assessment of the stranded assets problem. Energy Strategy Reviews 26, 100422
- Oei, P. / Burandt, T. / Hainsch, K. / Löffler, K. / Kemfert, C. (2020): Lessons from modeling 100% renewable scenarios using GENeSYS-MOD. Economics of Energy & Environmental Policy 9 (1)

1.1 General Model Description

In essence, GENeSYS-MOD can be illustrated as a flow-based cost-optimization model. The different nodes are represented as *Technologies*, which are connected by *Fuels*. Examples for *Technologies* are production entities like wind or solar power, conversion technologies like heat pumps, storages, or vehicles. *Fuels* serve as connections between these technologies and can be interpreted as the arcs of the network. In general, *Fuels* represent energy carriers like electricity or fossil fuels, but also more abstract units like demands of a specific energy carrier or areas of land are classified as *Fuels*. Also, *Technologies* might require multiple different *Fuels* or can have more than one output fuel. As an example, a combined heat and power plant could use coal as an input fuel and produce electricity and heating energy as an output fuel. Efficiencies of the technologies are being accounted for in this exact process, which would allow to model energy losses due to conversion. Energy demands are classified into three main categories: electricity, heating, and transportation. They are exogenously defined for every region and each year. The model then seeks to meet these demands through a combination of *Technologies* and trade between the different regions. The following figure gives a general overview of the different *Technologies* and the connections between them:

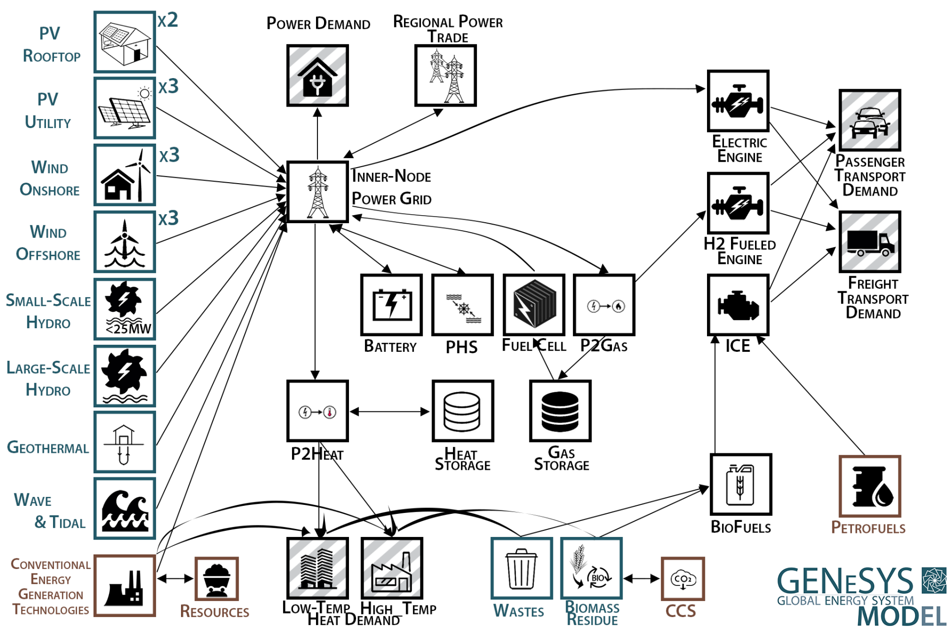


Figure 1: Abstracted overview over the model structure and available technology options.

1.1.1 Changes and additions compared to initial OSeMOSYS Version

GENeSYS-MOD v3.0 offers a fully revised data set for all global parameters, such as fuel prices, general cost assumptions, and emissions data. Furthermore, the list of available technologies has been revised and extended, now including more options in the transportation sector, as well as a representation of carbon capture, transport, and storage (CCTS) plants. Additionally, the model has been upgraded with new equations and revised formulations that offer more and new functionalities:

- The trade system (especially with respect to power trade) has further been improved. It introduces transmission capacities and the option for the model to endogenously expand them. The approach for the endogenous grid expansion is the same as described by Hosenfeld et al. (2017).
- New constraints limit the phase-in and phase-out of new technologies, as well as renewable electricity growth. The new equations make sure that new future technologies are not being used in one year and

then completely disregarded in the next, as well as old technologies being constructed and then ending as a stranded asset.

- Emission targets can be set globally as well as for individual regions.
- The efficiency of technologies depends now on the year of construction, rather than on the current date, avoiding an overestimation of potentials.
- The *ReserveMargin* has been redefined to better fit the flexibility requirements of a largely decarbonized system. The new formulation requires the model to produce a certain share of its production of a selected *Fuel* (e.g., power) with selected *Technologies* that offer the necessary flexibility when it comes to load balancing. As an example, this might include technologies with fast ramp-up times, such as electric storages or gas-fired plants.
- An option of running the model with reduced foresight has been added, as presented by Löffler et al. (2019).
- Additional feature to obtain the cost-optimal renewable energy sources share has been added. This routine consists of optimizing the power, transportation and industrial heating sectors under 20 different and binding shares of renewables. The binding shares start at 0% and grow in 5% intervals until reaching full decarbonization (compare Sarmiento et al. (2019)).
- Curtailment and different handling of dispatchable and non-dispatchable power generation technologies added.
- Revamped heating sector with four different temperature ranges and corresponding technologies, as presented by Burandt et al. (2019). The new structure of the heating sector features four different temperature ranges with a more distinct differentiation in industrial (0-100 ℃, 100-1000 ℃, and > 1000 ℃) and residential heating (0-100 ℃). For this new representation, a large variety of new technologies has been implemented to allow for alternative options to decarbonize industrial processes of more than 1000 ℃, as electrification poses only limited options for these cases. This new structure allows for a better illustration of sectoral carbon dioxide (CO_2) emissions, and thus allows for a more detailed analysis of the importance of the industry for a decarbonization of an energy system.
- Additional equations for ramping, ramping costs, and minimal runtine requirements added (compare Burandt et al. (2019))
- Added optional (reduced) hourly time resolution as used by the Dynamic Investment and Dispatch Model for the Future European Electricity Market (dynELMOD), presented by Gerbaulet and Lorenz (2017).
- Implemented performance optimization reduce the necessary memory resources and calculation time.

1.2 Components and Files Needed

To run properly, several files are needed, which are, after the program is started, called in a specified order. The following files are the primary files necessary for a successful run of GENeSYS-MOD:

- `genesysmod.gms` is the main program file including the model and solve statements. From here, the program can be started and some global settings and parameters like the choice of solver, can be changed. For an overview over the different parameters, please refer to Section 2.4.
- `genesysmod_dec.gms` is the first additional file that is called by the program. In here, all sets, parameters and variables (free, positive) are declared.
- `genesysmod_dataload.gms` is loaded hereafter. As the name suggests this file is used to feed in all the data needed for the program. The data can either be in the form of a `.gdx` file (which is the native form of GAMS data) or in form of an Excel sheet. `genesysmod_dataload.gms` will create a `.gdx` file itself. For higher usability and readability of the input-sheets, GENeSYS-MOD uses Excel-files by default. To allow the program to find the inputdata, the roots of respective files have to be included. By default, the Excel-files can be found in a folder named *Inputdata*, the then-created `.gdx` file are saved in a folder named *GdxFiles*. Temporary files and results are saved in the folders *Tempfiles* and *Results*. If you plan to work with different scenarios or variations of the same model, it is advisable to split up the Inputdata Excel into two: one that includes all data unchanged by the scenarios and one that is scenario specific.
- `genesysmod_bounds.gms` is used to set all settings that are specifically important for the model's application. This includes the manual setting of emission limits and the switching on and off of certain technologies for certain years. Smoothing factors and other restrictions can also be adjusted here.
- `genesysmod_settings.gms` is used to adjust global settings like the depreciation rate of capital. It is loaded after the bounds file, because in some conflicts multiple defined parameters, the global settings shall overwrite the specific settings.
- `genesysmod_equ.gms` is loaded the last. It contains every equation that is needed to have a properly running energy system model.
- `genesysmod_results.gms` is opened after the model finished the calculation process and is used to fit the results into a readable form, which can either be a `.gdx` file (easy to read in the GAMS-application) or a csv-file, which is better usable for further automatic data evaluation, transformation, and representation.

The following additional files can be modified by the user:

- `genesysmod_interpolation.gms` is used to interpolate missing data in certain year steps. E.g., when specific data for the year 2017 is missing, but the years 2015 and 2020 have values assigned, the value for 2017 will be linearly interpolated from those two.
- `genesysmod_subsets.gms` is called to assigns elements from the previously defined sets to specific subsets. Subsets always represent a certain partial quantity of a set and are defined to represent a

certain subgroup separately (for example, the group of transformation technologies in the group of technologies, the group of renewable fuels in the set of fuels, etc.).

- `genesysmod_scenariodata_%Region%.gms` includes relevant adjustments for different scenarios in case required. These might include general changes relevant for all scenarios or specific ones for single variants.
- `genesysmod_results_excel.gms` is used to generate additional outputs in the form of Excel workbooks. These are smaller and more limited in the scope as compared as the large model outputs generated by `genesysmod_results.gms`.

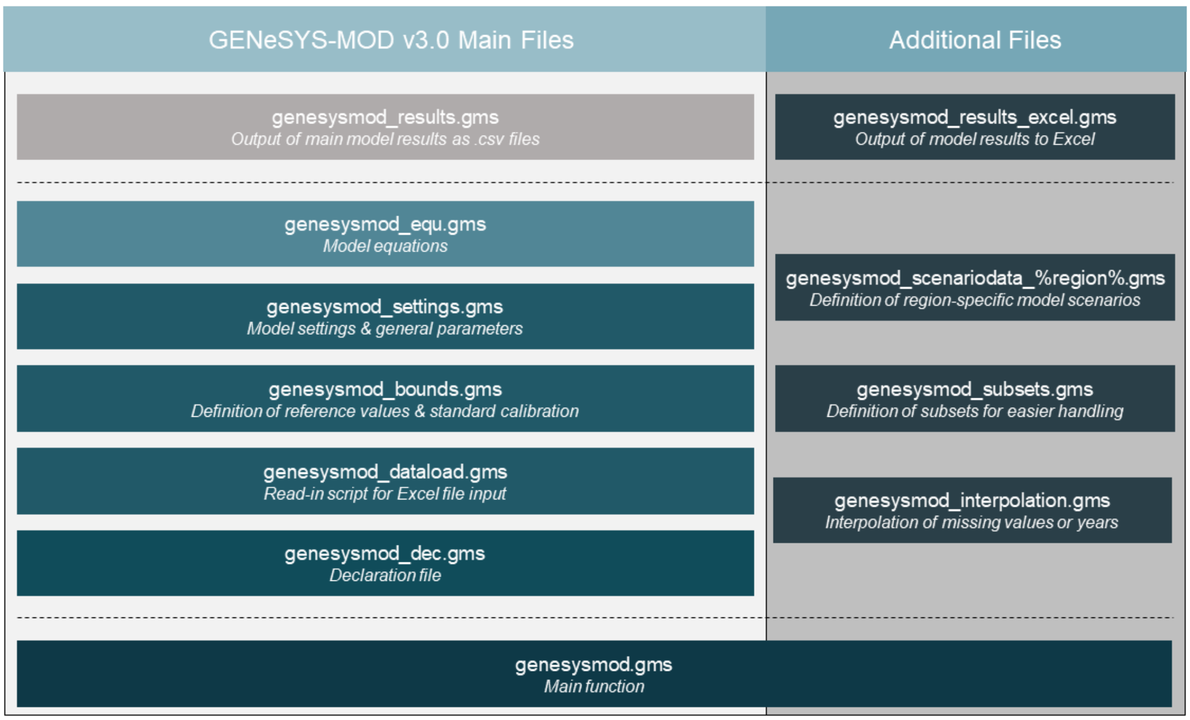


Figure 2: Overview over the primary and secondary model files used to run GENeSYS-MOD.

1.3 Description of Fundamental Concepts

Though the fundamental concepts of optimization problems, as well as the set up of a problem that is transferred into machine-readable code should be known by the user, a quick recap is provided in this chapter. Each problem that is set up to run properly needs to consist of several parts. In the case of GENeSYS-MOD, they are separated into different files (see above, Section 1.2).

```
* ############# Model Definition #############

* # Sets #
Set YEAR_FULL /2015*2050/;
alias (y_full, yy_full, YEAR_FULL);
set YEAR(y_full);
```

```
alias (y,yy,YEAR);

set FUEL;
alias (f,FUEL);
set EMISSION;
alias (e,EMISSION);

* Sets are defined using the command 'set'. It can either be filled using code (see first set) or using the input-excel (see e.g. set 'FUEL')

* # Parameters #
parameter StartYear;
parameter YearSplit(TIMESLICE,y_full);
parameter DiscountRate(REGION_FULL);

* Parameters are defined within the code and connected to the sets in brackets.

* # Model Variables #
positive variable NewCapacity(y_full,TECHNOLOGY,REGION_FULL);
positive variable AccumulatedNewCapacity(y_full,TECHNOLOGY,REGION_FULL);
positive variable TotalCapacityAnnual(y_full,TECHNOLOGY,REGION_FULL);

* Variables can either be set separately (e.g. to limit certain aspects manually). Mainly variables are used by the model and are used for data evalution.

* ############# Equations, Model and Solve Statements #############

* # Objective Function #
free variable z;
equation cost;
cost.. z =e= sum((y,r), TotalDiscountedCost(y,r)) + sum((y,r), DiscountedAnnualTotalTradeCosts(y,r)) + sum((y,f,r,rr), DiscountedNewTradeCapacityCosts(y,f,r,rr));

* The objective function shall describe the model's objective. It includes the objective value (in this case 'z') that shall either be maximized or minimized.

* # Constraints #
equation CAa1_TotalNewCapacity(YEAR_FULL,TECHNOLOGY,REGION_FULL);
CAa1_TotalNewCapacity(y,t,r).. AccumulatedNewCapacity(y,t,r) =e= sum(yy$((YearVal(y)-YearVal(yy) < OperationalLife(r,t)) AND (YearVal(y)-YearVal(yy) >= 0)), NewCapacity(yy,t,r));

* Every other constraint or side condition that can be described as an equation is listed in the same block of the target function. Constraints that are made by a simple value definition can already be made in the parameter definition part.
* Equations shall consist of variables (free, positive) and some kind of relation.

* ####### Model and Solve statements #############
```

```
model genesys /all/;

$ifthen  %UseMipSolver% == yes
solve genesys minimizing z using Mip;
$else
solve genesys minimizing z using lp;
$endif

* The last part of the model includes the model and solve statements: in this case, every aforementioned set, parameter, variable and equation shall be included, the solver and the objective value is set.
```

2 Quick Start and Commands

This chapter shall help you to first install the General Algebraic Modelling System (GAMS) on your machine and setting up GENeSYS-MOD in a way to start the first model run. Also, this chapter will describe how to use the model and switch between different modes of model runs.

2.1 Installing Gams & First Start of GENeSYS-MOD

The model is coded in GAMS. Therefore, the first step is to install GAMS on your machine. GAMS is a algebraic modeling language designed to solve mathematical optimization and equilibrium problems. The language makes it possible to write as close as possible to the respective mathematical equivalents and saves the user in many cases the learning of a special syntax. Compared to other classic programming languages, learning GAMS is easily possible, if you have the mathematical formulation on hand. In general, GAMS consists of a compiler and a multitude of different solution algorithms, which can be used depending on the modelling. Furthermore, it also brings two different integrated development environments (IDEs) that allow for advanced code writing, enhanced error handling, and partially code completion (i.e., *GAMS IDE* and *GAMS Studio*).

The most recent version of GAMS can be retrieved from their website (www.gams.com). It is available for MS Windows (32, 64 Bit), MacOS, and Linux (both 64 Bit). GAMS has a free licensed version, but it is limited to some solvers and, more importantly, to only a small number of variables and elements. AS GENeSYS-OD is a large-scale energy system models, the free version of the provided Solvers are not applicable and you will likely need to obtain a license for a commercial solver.

Important Note:

GENeSYS-MOD uses the command-line tool `gdxxrw.exe`, which is provided by GAMS, to read-in and convert Excel-files to .gdx-files. Unfortunately, this command-line tool is only working a DOS/Windows environment and cannot be used in Unix systems (Linux/Mac) without the usage of a emulation software (e.g., *Wine* for Linux). Therefore, you either need to run GENeSYS-MOD under a Windows environment, or you need to transform Excel-files to .gdx-files with other tools (e.g., with the *exceltogdx*-package for python).

Installation and License File:

After downloading GAMS you may install the program to whatever directory you wish. Please check, the option of which GAMS distribution you will install: To run GENeSYS-MOD properly, select *GAMS IDE*. The GAMS license is a plain text .txt-file. After installation you will be asked to include the license, either by having it in the clipboard (the installer will check for that automatically) or by selecting your GAMS license -.txt-file. If you don't have one during installation you can add your license later (see next section).

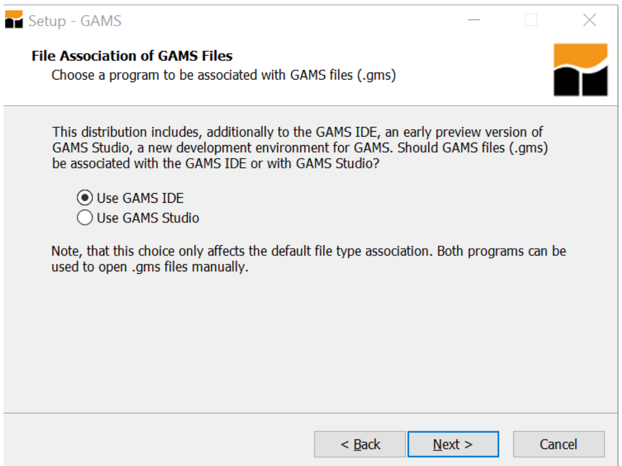


Figure 3: When asked during installation, select IDE of your choice (*GAMS IDE* or *GAMS Studio*).

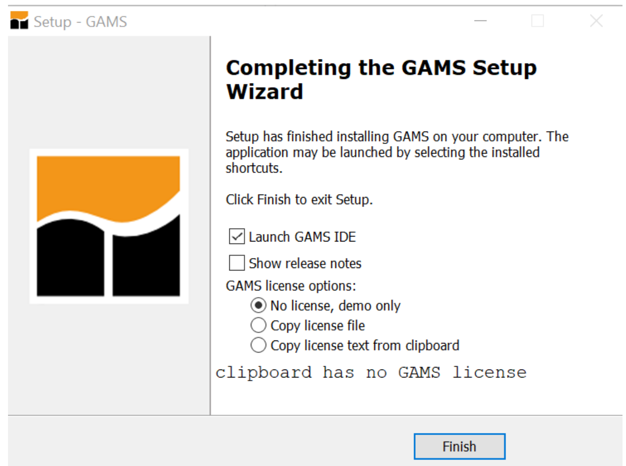


Figure 4: Possible of ways to add the GAMS license file.

If you choose the no-license-option, you can implement a license file later. Therefore, you need to open *GAMS IDE*, open the FILE tab in the top-left corner (Hotkey: ALT+F) and navigate to OPTIONS. In the OPTIONS-window, select the Licenses-tab, check the box "Use alternate License file" and add the directory of where you have saved the license file. Similarly, the license file can later be added using *GAMS Studio* instead of *GAMS IDE*.

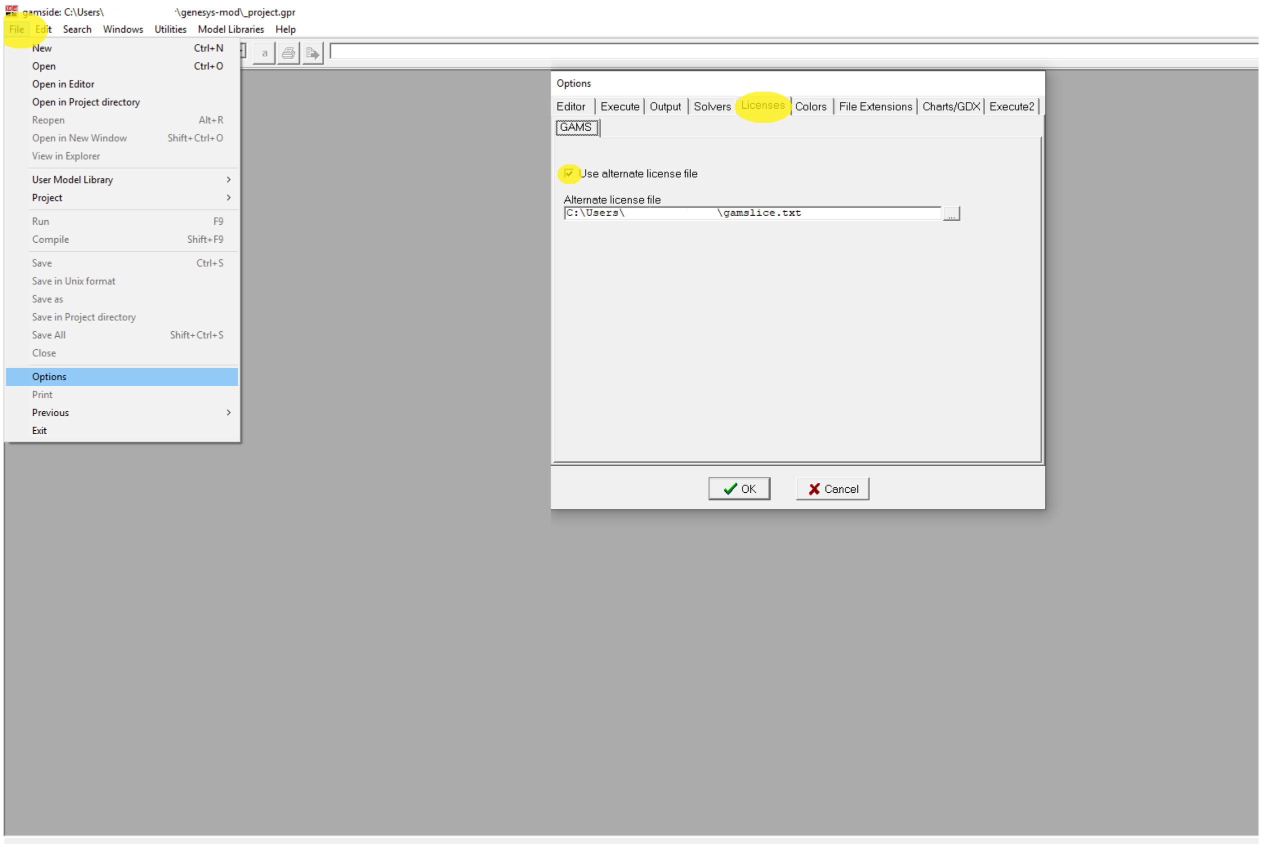


Figure 5: Manually choose your GAMS-license-file.

You need up to two different solvers to run GENeSYS-MOD: one LP solver and one DNLP solver. The LP solver is used for the general execution of GENeSYS-MOD, while the DNLP solver is required if the time-series reduction algorithm is used. To change the default solver for a specific class of optimization problem, see the "Solver" tab in the previous illustration and simply click the respective field in the table.

Recommendated Solvers:

The recommended solvers for GENeSYS-MOD are **GUROBI** as the LP solver and **PATHNLP** as the DNLP solver. Solver options are also given for **CPLEX** as an alternate LP solver. **IPOPT** can be used as a free DNLP solver, however, discrepancies in results could be noticed when switching the DNLP solver.

Downloading GENeSYS-MOD:

GENeSYS-MOD can be downloaded from its GitLab page. After download, please extract the .zip-file to your preferred destination. All required files and folders should be included in the downloaded file.

Setting up GENeSYS-MOD in GAMS:

For using GENeSYS-MOD in *GAMS IDE*, you need a GAMS project that is located in the GENeSYS-folder. One sample project file is provided in the .zip archive, however, if there are any issues, a new project file needs to be created. To create a new project file, you need to click on the FILE-tab (Hotkey: ALT+F) and navigate to Project > New Project. A window will open, in which you should select the directory of your

GENeSYS-MOD folder. Now, you can create a new project file and give it a name you like. The suffix .gpr will be added automatically when you click Open. It is important that your .gpr-file is in the same directory as the other GAMS-files, because only then GAMS will be able to locate all necessary files. For using GENeSYS-MOD in *GAMS Studio*, simply open the file genesysmod.gms (and other files you want to change) with GAMS Studio. In this case, no dedicated project file is needed.

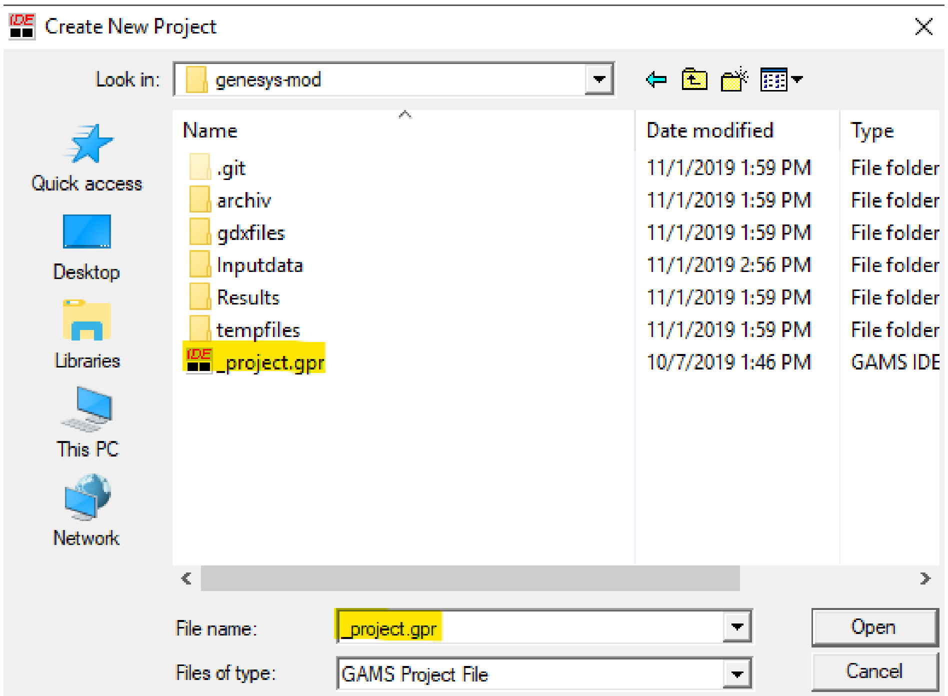


Figure 6: Create a .gpr-file in the GENeSYS-MOD directory. You can name it however you want.

Opening genesysmod.gms

For the beginning, only the main file needs to be opened: genesysmod.gms. As mentioned in the Section 1.2, this is the file from where the model is started, where all information concerning the model are set, and where the links to other files are coded. However, to run the program, you need all other files in the same directory genesysmod.gms (and as your project-file in case of using *GAMS IDE*)! Often, the model will return error messages, when the project file is not in the same directory as the files that are executed. A hint to check if everything is correct can be found in the top bars: The GAMS environment will open a window in which all opened GAMS files are displayed in tabs. In the headers, you will always find the respective file path (see Figure 7: No. 1 and 6).

2.2 The User Interface of GAMS IDE

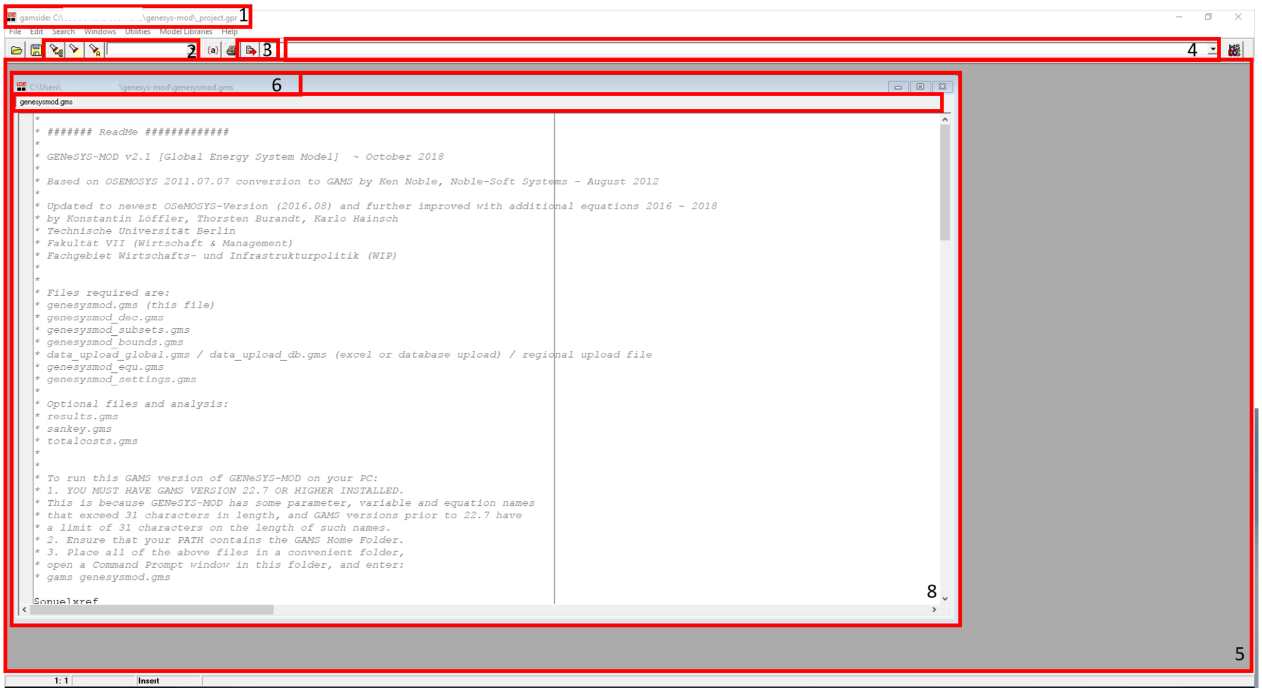


Figure 7: The main elements of the GUI of GAMS IDE

1. Main window header and file path of the project.
2. Different search options.
3. Run model (hotkey F9).
4. Command-line.
5. Project environment.
6. File editor window header and file path of the open file.
7. File tabs. You can have several files open in the same time. Each file will have its own tab.
8. File editor environment.

GAMS Studio

For a description of the main elements and features of *GAMS Studio*, please refer to https://www.gams.com/latest/docs/T_STUDIO.html.

2.3 Running GENeSYS-MOD

To run the model, you simply need to click on the icon left to the command prompt (No. 3 in Figure 7) or F9. Then, the model will start in a certain default setting which will be explained in the next section. If you want to change several things for the model run, you can add specific command-line options before running the model. These additional options will also be explained in the following section.

2.4 Command-line parameters and default settings

To run the model, simply execute the genesysmod.gms file. No further command-line parameters are required. Command-line parameters (see Table 1) can be provided via the utilized IDE or when executing GAMS directly from the command-line. You can add variables (marked with a '-') and GAMS-commands (marked with a '--') into the command-line. The variables will activate switches to alternate some parts of the code or to read in alternative input files. For example, `--gdx=outputgdx -region=germany -emissionScenarion=GD` will run GENeSYS-MOD for the region *Germany* and with the scenario *GD*, provided the correct input-files are available. Additionally, a full output .gdx-file named *'outputgdx.gdx'* with all symbols and equations will be created (compare https://www.gams.com/latest/docs/UG_GDX.html#UG_GDX_WRITE_AFTER_COMPILE_OR_EXECUTION).

```
*
* ############################################################



$onuelxref
scalar starttime;
starttime = jnow;

$if not set year                        $setglobal year 2015

$if not set switch_unixPath             $setglobal switch_unixPath 0
$if not set switch_investLimit          $setglobal switch_investLimit 1
$if not set switch_ccs                  $setglobal switch_ccs 0
$if not set switch_ramping              $setglobal switch_ramping 1
$if not set switch_short_term_storage   $setglobal switch_short_term_storage 1
$if not set switch_all_regions          $setglobal switch_all_regions 1
$if not set switch_infeasibility_tech   $setglobal switch_infeasibility_tech 1
$if not set switch_base_year_bounds     $setglobal switch_base_year_bounds 1
$if not set switch_only_load_gdx        $setglobal switch_only_load_gdx 0
$if not set switch_write_output_excel   $setglobal switch_write_output_excel 1
$if not set switch_aggregate_region     $setglobal switch_aggregate_region 0
$if not set switch_intertemporal        $setglobal switch_intertemporal 0
$if not set switch_weighted_emissions   $setglobal switch_weighted_emissions 1
$if not set switch_employment_calculation $setglobal switch_employment_calculation 0
$if not set switch_test_data_load       $setglobal switch_test_data_load 0
$if not set switch_only_write_results   $setglobal switch_only_write_results 1


$if not set solver                      $setglobal solver cplex
$if not set model_region                $setglobal model_region southafrica
$if not set data_base_region            $setglobal data_base_region SA-EC
$if not set data_file                   $setglobal data_file Data_SouthAfrica_v39_nm_03_02_2020
$if not set eployment_data_file         $setglobal employment_data_file Employment_v01_06_11_2019
$if not set hourly_data_file            $setglobal hourly_data_file Hourly_Data_SouthAfrica_v11_kl_06_12_2019
$if not set global_data_file            $setglobal global_data_file Global_Data_v02_kl_04_12_2019_SA
$if not set threads                     $setglobal threads 3
$if not set timeseries                  $setglobal timeseries elmod
$if not set elmod_nthhour               $setglobal elmod_nthhour 250
$if not set elmod_starthour             $setglobal elmod_starthour 8
$if not set elmod_dunkelflaute          $setglobal elmod_dunkelflaute 0

$if not set emissionPathway             $setglobal emissionPathway 2degree
$if not set emissionScenario            $setglobal emissionScenario globalLimit

$if not set socialdiscountrate          $setglobal socialdiscountrate 0.05
```

Figure 8: A Screenshot of the first block of code in the genesysmod.gms file, showing the command-line parameters.

Table 1: Command-line Parameters

Command	Values [Default]	Description
-year	user defined [2015]	Set the starting year
-switch_unixPath	0, 1 [0]	Turn on, if you work in a Unix-based file system.
-switch_ccs	0, 1 [0]	Allow carbon capture and storage (CCS) technologies
-switch_ramping	0, 1 [0]	Enable additional ramping constraints for power plants
-switch_short_term_storage	0, 1 [0]	Enable a different storage formulation, specifically designed for a better representation of short-term storages with consecutive hours. This setting is recommended when using a reduced hourly time-series instead of the time-slice approach
-switch_all_regions	0, 1 [1]	Include all regions in model run of one model region. When set to 0, only the first four listed regions will be included in the calculation.
-switch_infeasibility_tech	0, 1 [1]	Enable *Infeasiblity* Technologies which are being used in case of the model being infeasible otherwise. This allows for an enhanced debugging of potential data errors.
-switch_base_year_bounds	0, 1 [1]	Include base-year bounds for capacities and/or production.
-switch_only_load_gdx	0, 1 [0]	Required when operating on systems where excel-files can not be read. In this case, .gdx-files are needed to be provided manually.
-switch_write_output	xls, csv, gdx [gdx]	Creates the outputs in different output formats. GDX-Files are always written as intermediate output files. Setting the parameter to the corresponding values, allows for additional generation of .csv-files (*csv*) or Excel-Files (*xls*).
-switch_aggregate_region	0, 1 [1]	Aggregates all regions into one. This results in a largely reduced model size with the trade-of of neglecting regional differences.

Command	Values [Default]	Description
-switch_intertemporal	0, 1 [0]	Efficiencies of existing capacities improve over time if set to 0, stems from old OSeMOSYS formulation. Setting it to 1 increases computational time
-switch_weighted_emissions	0, 1 [1]	Use weighted emissions for the years which are not being calculated.
-switch_employment_calculation	0, 1 [1]	If data for the specific region is available, a certain module can be turned on, which allows to also calculate employment effects
-switch_test_data_load	0, 1 [0]	Only runs steps until dataload is finished which is helpful to analyze whether a change in input data is read-in correctly
-switch_only_write_results	0, 1 [0]	Only (re-)writes results as an Excel-file. An existing result .gdx-file is required for this operation.
-solver	cplex, gurobi [cplex]	Sets the solver, currently only option files for `cplex` and `gurobi` are provided. Other solver can nonetheless be used and entered.
-model_region	user defined	A list of all model regions can be found in the appendix. Also depends on which input files are available.
-data_base_region	user defined	Sets the region from where base data, including fixed, capital and variable costs for technologies might be loaded from. Changes with different model regions.
-data_file	user defined	Sets the main data file for the region being calculated.
-employment_data_file	user defined	Sets additional employment data, if `-switch_employment_calculation` is activated
-hourly_data_file	user defined	Input-file for full-hourly data used for generation time-series and time-slices data.
-global_data_file	user defined	Sets the input file for the global technology data. This file is being used as fallback option when technology data is missing in the `data_file`.

Command	Values [Default]	Description
-threads	user defined	Choose the number of threads being used by the solvers. No. of (virtual) kernels -1 is appropriate if you want to use your machine during model runs.
-timeseries	elmod	Which timeseries should be used
-elmod_nthhour	n*24+1<8760 [121]	Defines the level of aggregation when using `timeseries = elmod`
-elomd_starthour	1-24 [1]	Sets the starting hour of the time-series reduction algorithm when using `timeseries = elmod`
-elmod_dunkelflaute	0, 1 [0]	Defines whether a specific timeseries is included on top of the existing with bad properties for wind and solar
-emissionPathway	user defined	User defined scenario pathway
-emissionScenario	user defined	User defined scenario
-socialdiscountrate	user defined [0.05]	How strong future investments are discounted

2.5 Adding a new region to GENeSYS-MOD

I. Set up data files Set up and provide data files for general (data_file) and hourly (hourly_data_file) data. Use existing files as template. Suggestion: copy and rename the existing files and try to run them as your new region.

II. Add region-specific bounds file Add new file `genesysmod_scenariodata_[YOUR REGION].gms`. This file is used to add scenarios for your case-study. Also, because this file is loaded after setting general bounds and limitations, you can use is to overwrite predefined bounds. Restrain from changing the general bounds as much as possible.

III. Test your region and input files Provide the new data-files via command-line parameter or change the default values in the `genesysmod.gms` file.

Further hints and tips

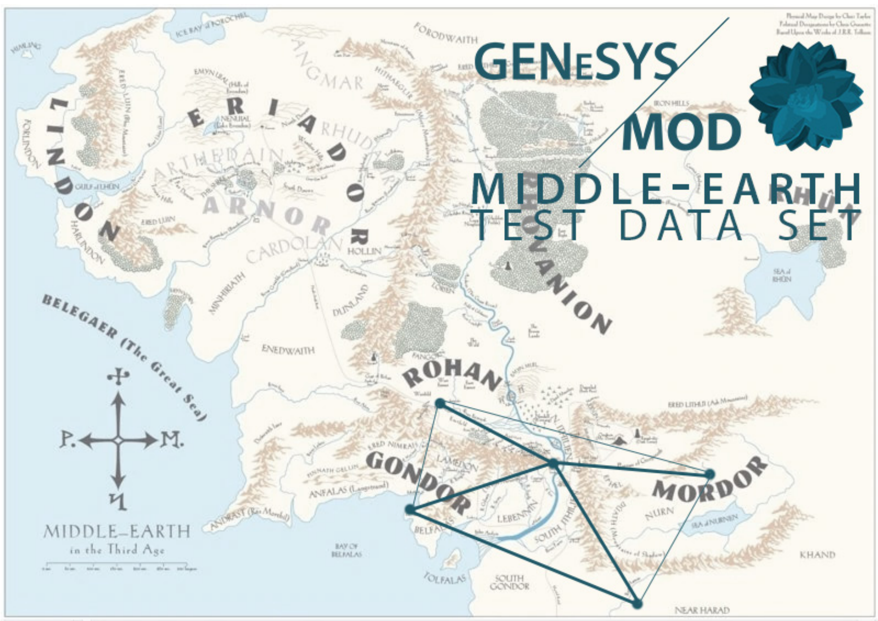
- Start slow and change as little as much possible between the model runs.
- Start with `timeseries=classic` and without ramping. The model size is greatly reduced and you can calibrate the base-year and general pathway transition of your model much easier.

3 Handling the Model Output

GENeSYS-MOD creates two output files of relevance: a .gdx file and an excel file. After a succesful execution of GENeSYS-MOD, they can be found in the "GdxFiles" folder and the "Results" folder. The gdx file contains all information regarding the model inputs and outputs, while the excel file shows a selection of all model results. The respective outputs can be adjusted in the `genesysmod_results` file.

4 Result verification: example data

To verify the integrity of model code and data, as well as to make sure that the model is running properly, please run the model once with the included "Middle-Earth" data set. The computation will use about 6GB of RAM and take about 20-40 minutes, depending on your machine. Please note that the comparison values have been obtained with the LP solver GUROBI, the DNLP solver IPOPT, and GAMS version 32.0.



When computing the Middle-Earth example data set provided with GENeSYS-MOD v3.0, you should obtain the following results (note that these might differ slightly if you use different solvers and/or GAMS versions):

		Symbol in GAMS
Objective value	4.095906985e+06	z
Production of lignite power plants in Mordor in 2015	184.275 PJ	ProductionByTechnologyAnnual.l ('2015','P_Lignite','Power','Mordor')
Total capacity of onshore wind (avg) in Gondor in 2050	36.571 GW	TotalCapacityAnnual.l ('2050','RES_Wind_Onshore_avg','Gondor')

5 Further information and contact data

For more information on the technical details of GENeSYS-MOD, please refer to our Technical User Manual, found here. For any enquiries regarding open questions and support, please contact genesys-mod@coaltransitions.org or visit our GitLab page at https://git.tu-berlin.de/genesysmod/genesys-mod-public.