

Package ‘teems’

June 17, 2026

Title Trade and Environment Equilibrium Modeling System

Version 0.1.1

Description Equilibrium models are frequently employed to examine the potential impacts of economic, energy, and trade policies as well as form the foundation of most integrated assessment models. The 'teems' package handles all aspects of running equilibrium models while facilitating transparent and reproducible workflows.

License AGPL (≥ 3)

URL <https://teemsphere.github.io/>,
<https://github.com/teemsphere/teems/>

BugReports <https://github.com/teemsphere/teems/issues>

Depends R ($\geq 4.3.0$)

Imports cli ($\geq 3.4.0$), data.table ($\geq 1.15.0$), purrr ($\geq 1.0.0$), R6 ($\geq 2.4.0$), rlang ($\geq 1.0.0$), stats, tibble ($\geq 3.0.0$), tools, utils

Suggests testthat ($\geq 3.0.0$), withr ($\geq 3.0.0$)

LinkingTo cpp11

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

SystemRequirements Docker (<https://www.docker.com>)

NeedsCompilation yes

Author Matthew Cantele [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-3729-1764>),
Tom Kompas [ctb, ths] (ORCID: <https://orcid.org/0000-0002-0665-0966>),
Ha Van Pham [ctb] (ORCID: <https://orcid.org/0000-0002-5927-6568>),
Martin Ingram [ctb] (ORCID: <https://orcid.org/0000-0001-6269-8342>)

Maintainer Matthew Cantele <matthew.cantele@protonmail.com>

Repository CRAN

Date/Publication 2026-06-17 14:10:03 UTC

Contents

ems_compose	2
ems_custom_shock	3
ems_data	4
ems_deploy	6
ems_example	8
ems_model	9
ems_option_get	11
ems_option_reset	12
ems_option_set	13
ems_scenario_shock	14
ems_solve	15
ems_swap	17
ems_uniform_shock	18
GTAP_convert	19
solve_in_situ	21
Index	24

ems_compose	<i>Compose model results into structured data objects</i>
-------------	---

Description

ems_compose() retrieves and processes results from a solved model run. Data validation and consistency checks are performed during composition.

Usage

```
ems_compose(cmf_path, which = "all")
```

Arguments

cmf_path	Character length 1, path to the CMF file generated by ems_deploy() .
which	Character vector of variable length (default "all"). When "all", all model variables are returned plus all coefficients if any coefficient output files are detected. Otherwise, a character vector of variable and/or coefficient names to retrieve — an error is raised for any name not found. Note that some coefficient names differ from their associated headers (e.g., VTWR/VTMFSD).

Value

A tibble with columns "name", "label", "type", and "dat" (a list-column of data.tables) containing model results.

See Also

[ems_solve\(\)](#) for solving the CGE model.

Examples

```
## Not run:  
# The following examples require that a model run has taken  
# place. See https://teemsphere.github.io/ to get started.  
  
# Return all variables and coefficients  
outputs <- ems_compose(cmf_path)  
  
# Return specific variables and/or coefficients by name  
outputs <- ems_compose(cmf_path, c("qfd", "EVFP"))  
  
## End(Not run)
```

ems_custom_shock	<i>Prepare a custom shock</i>
------------------	-------------------------------

Description

Prepares heterogeneous percentage-change shocks specified on a tuple-by-tuple basis. Only the tuples present in input are shocked.

Usage

```
ems_custom_shock(var, input)
```

Arguments

var	Character of length 1, the variable to be shocked.
input	Path to a CSV file, or a data frame or data frame extension (e.g., tibble, data.table). Must contain one column named "Value" with the percentage-change shock for each tuple, plus one column per set index for the variable. Note that set names must conform to the model-specific hybrid format (e.g., ACTSa, REGr). Column order is inconsequential.

Value

A list object with shock configuration to be passed to the shock argument of [ems_deploy\(\)](#).

See Also

[ems_deploy\(\)](#) for loading the output of this function, [ems_swap\(\)](#) for changing the standard model closure.

Examples

```
# Set elements
sectors <- c("crops", "food", "livestock", "mnfcs", "svces")
regions <- c("asia", "eit", "lam", "maf", "oecd")
time_steps <- 0:7

# Create data frame
aoall_data <- expand.grid(ACTSa = sectors,
                        REGr = regions,
                        ALLTIMEt = time_steps)

aoall_data$Value <- runif(nrow(aoall_data))
head(aoall_data)

# Assign values to a model variable (aoall)
ems_custom_shock("aoall", aoall_data)

# In the case of temporally dynamic models, chronological
# year(s) specified by the column "Year" may be used instead
# of the the time set.
pop_data <- data.frame(REGr = "asia",
                      Year = 2023:2030)

pop_data$Value <- runif(nrow(pop_data))
pop_data

ems_custom_shock("pop", pop_data)
```

ems_data

Load and prepare data inputs

Description

Loads input files, aggregates basedata and parameters according to set mappings, and returns prepared data for `ems_deploy()`. Accepts GTAP HAR files and objects generated by `GTAP_convert()` as inputs.

Usage

```
ems_data(dat_input, par_input, set_input, time_steps = NULL, ...)
```

Arguments

dat_input	Character vector of length 1, file name in working directory or path to a data input containing basedata coefficient data, or the corresponding "dat" list object generated by <code>GTAP_convert()</code> .
par_input	Character vector of length 1, file name in working directory or path to a data input containing parameter coefficient data, or the corresponding "par" list object generated by <code>GTAP_convert()</code> .

set_input	Character vector of length 1, file name in working directory or path to a data input containing set elements and attributes, or the corresponding "set" list object generated by <code>GTAP_convert()</code> .
time_steps	Integer vector of variable length (default is NULL). For use with temporally dynamic models. Input format is either chronological years of steps including initial reference year corresponding to the database employed or steps from t0 with initial value 0.
...	A named pairlist assigning set mappings to model sets that are explicitly read (not constructed via any set operation defined in the model file). Set mappings can be selected from a range of internally available options or provided by the user. <ul style="list-style-type: none"> • Internal mapping: character vector of length 1 corresponding to an internally available mapping. • User-provided mapping: character vector of length 1, file name in working directory or path to a CSV file; a data frame or data frame extension (e.g., tibble, data.table). <p>See teems-mappings for internally available mappings and formatting for user-provided mappings.</p>

Details

Set names passed via ... must match set names as they appear in the database. Only sets requiring aggregation need to be specified. Unspecified sets whose elements are fully covered by a provided mapping are derived automatically. Any elements not covered by a provided mapping are passed through without aggregation.

Value

A list of data.tables ready for input into the .data argument of `ems_deploy()`

See Also

`ems_deploy()` for loading the output of this function. `GTAP_convert()` for converting GTAP HAR files and data formats.

Examples

```
## Not run:
# The following examples require input data. See
# https://teemsphere.github.io/ to get started.

# Data for a static model using internal mappings
v7_data <- ems_data("v7_data/gsdmdat.har",
                  "v7_data/gsdmpar.har",
                  "v7_data/gsdmset.har",
                  REG = "AR5",
                  ACTS = "food",
                  ENDW = "labor_diff")
```

```

# Data for an intertemporal model with explicit time steps
int_data <- ems_data("v6_data/gsddat.har",
                   "v6_data/gsdpar.har",
                   "v6_data/gsdset.har",
                   REG = "WB23",
                   PROD_COMM = "services",
                   ENDW_COMM = "labor_agg",
                   time_steps = c(0, 1, 2, 4, 6, 8, 10, 15))

# Data for an intertemporal model with chronological time
# steps and a user-provided mapping for the ENDW set
int_data <- ems_data("v7_data/gsdfdat.har",
                   "v7_data/gsdfpar.har",
                   "v7_data/gsdfset.har",
                   REG = "R32",
                   ACTS = "medium",
                   ENDW = "user/endowment/mapping.csv",
                   time_steps = c(2023, 2025, 2027, 2030, 2035))

## End(Not run)

```

ems_deploy

Validate and write all model inputs

Description

Validates all data and model inputs, processes shocks and closure swaps, and writes the input files required by `ems_solve()`.

Usage

```

ems_deploy(
  .data,
  model,
  shock = NULL,
  swap_in = NULL,
  swap_out = NULL,
  shock_file = NULL
)

```

Arguments

<code>.data</code>	A list of data.tables generated by <code>ems_data()</code> .
<code>model</code>	A tibble generated by <code>ems_model()</code> .
<code>shock</code>	A shock object or list of shock objects produced by <code>ems_uniform_shock()</code> , <code>ems_custom_shock()</code> , or <code>ems_scenario_shock()</code> . When NULL (default), no shock is applied and input data is returned without change.

swap_in	A character vector of variable length, list generated by <code>ems_swap()</code> , or combination of the above provided as a list (default NULL). Variables to be made exogenous.
swap_out	A character vector of variable length, list generated by <code>ems_swap()</code> , or combination of the above provided as a list (default NULL). Variables to be made endogenous.
shock_file	A character vector length 1 (default NULL). Path to a file with .shf extension representing a fully prepared shock file. No checks or modifications are carried out on this file.

Value

Path to the CMF file, to be passed to `ems_solve()`.

See Also

`ems_data()` for loading and preparing data inputs. `ems_model()` for parsing and modifying model and closure files. `ems_uniform_shock()`, `ems_custom_shock()`, and `ems_scenario_shock()` for preparing shocks. `ems_swap()` for preparing closure swaps. `ems_solve()` for utilizing the output of this function.

Examples

```
## Not run:
# The following examples require input data. See
# https://teemsphere.github.io/ to get started.

# Uniform shock applied to full variable with value 1
shock <- ems_uniform_shock("qfd", value = 1)

# Full variable swap, qfd for tfd
cmf_path <- ems_deploy(
  .data = dat,
  model = model,
  shock = shock,
  swap_in = "qfd",
  swap_out = "tfd"
)

# Partial variable swap (element "row" of set REG with index r)
yp_row <- ems_swap("yp", REGr = "row")
dppriv_row <- ems_swap("dppriv", REGr = "row")

# Swap part of yp for part of dppriv; qfd for tfd
cmf_path <- ems_deploy(
  .data = dat,
  model = model,
  shock = shock,
  swap_in = list(yp_row, "qfd"),
  swap_out = list(dppriv_row, "tfd")
)
```

```
## End(Not run)
```

```
ems_example
```

```
Write teems example models and scripts to file
```

Description

The teems package includes example models and scripts. This function facilitates access to both, allowing new users to immediately peruse compatible models or generate scripts containing a range of package features.

Usage

```
ems_example(
  model,
  path,
  type = c("model_files", "scripts"),
  dat_input = NULL,
  par_input = NULL,
  set_input = NULL
)
```

Arguments

model	Character vector of length 1, name of model. Options include: <ul style="list-style-type: none"> • "GTAP-RE": GTAP Rational Expectations • "GTAP-INT": GTAP Intertemporal • "GTAPv7": Standard GTAP model • "GTAPv6": Classic GTAP model
path	Character vector of length 1. Directory where files will be written.
type	Character vector of length 1, (default is "model_files"). Determines the example output. Options include: <ul style="list-style-type: none"> • "model_files": Model file and closure file • "scripts": Example scripts for the chosen model. <code>dat_input</code>, <code>par_input</code>, and <code>set_input</code> are required.
dat_input	Character vector of length 1 (default is NULL, only used when <code>type = "scripts"</code>), file name in working directory or path to a data input containing basedata coefficient data, or the corresponding "dat" list object generated by <code>GTAP_convert()</code> .
par_input	Character vector of length 1 (default is NULL, only used when <code>type = "scripts"</code>), file name in working directory or path to a data input containing parameter coefficient data, or the corresponding "par" list object generated by <code>GTAP_convert()</code> .
set_input	Character vector of length 1 (default is NULL, only used when <code>type = "scripts"</code>), file name in working directory or path to a data input containing set elements and attributes, or the corresponding "set" list object generated by <code>GTAP_convert()</code> .

Value

For type = "model_files": a named character vector with elements model_file and closure_file.
 For type = "scripts": a character vector of paths to the written script files.

See Also

[GTAP_convert\(\)](#) for converting GTAP HAR files and data formats.

Examples

```
# Retrieve GTAP-RE model files
ems_example("GTAP-RE", tempdir())

## Not run:
# The following example requires input data. See
# https://teemsphere.github.io/ to get started.

# Generate GTAP-RE example scripts
ems_example(model = "GTAP-RE",
            path = "path/to/example/scripts",
            type = "scripts",
            dat_input = "v7_data/gsdmdat.har",
            par_input = "v7_data/gsdmpar.har",
            set_input = "v7_data/gsdmset.har")

# Generate GTAPv7 example scripts from a v6.2 format database
converted <- GTAP_convert(dat_har = "v6_data/gsdv6dat.har",
                        par_har = "v6_data/gsdv6par.har",
                        set_har = "v6_data/gsdv6set.har",
                        target = "GTAPv7")

ems_example(model = "GTAPv7",
            path = "path/to/example/scripts",
            type = "scripts",
            dat_input = converted$dat,
            par_input = converted$par,
            set_input = converted$set)

## End(Not run)
```

 ems_model

Parse and modify model and closure files

Description

Parses the model and closure files, conducts pre-deployment checks, and carries out specified modifications.

Usage

```
ems_model(model_file, closure_file, var_omit = NULL, ...)
```

Arguments

model_file	Character vector length 1, path to a TABLO file with .tab extension. See teems-models for vetted models and compatible Tablo file formatting.
closure_file	Character vector length 1, path to a closure file with .cls extension. See teems-models for closure file formatting.
var_omit	Character vector of variable length (default is NULL), variable names to substitute with \emptyset in the model and remove from the closure.
...	A named pairlist assigning values to model coefficients. Name must match a coefficient declared in the model file. Value may be a length-1 numeric, a data frame or data frame extension (e.g., tibble, data.table), or a path to a CSV file.

Details

If a length-1 numeric is supplied via ..., this value is applied uniformly across all tuples for that coefficient. For heterogeneous values, data frame and CSV inputs must contain a Value column and one column per set index using the model-specific naming convention (set name concatenated with its index letter, e.g., REGr, COMMc). Inputs are subject to structure checks against the set aggregations specified in [ems_data\(\)](#) and all tuples must be present.

Value

A tibble comprising parsed model statements to be passed to the model argument of [ems_deploy\(\)](#).

See Also

[ems_example\(\)](#) for writing teems example models and scripts to file. [ems_deploy\(\)](#) for loading the output of this function as well as conducting closure swaps.

Examples

```
# Simple static model retrieval and load
GTAPv7 <- ems_example("GTAPv7", tempdir())
ems_model(GTAPv7[["model_file"]], GTAPv7[["closure_file"]])

# Retrieve intertemporal model
GTAP_RE <- ems_example("GTAP-RE", tempdir())

# Construct data frame
sectors <- c("crops", "food", "livestock", "mnfcs", "svces")
regions <- c("usa", "chn", "row")
time_steps <- 0:5

SUBPAR <- expand.grid(COMMc = sectors,
                     REGr = regions,
                     ALLTIMEt = time_steps)
```

```

SUBPAR$Value <- runif(nrow(SUBPAR))

# Model load with:
# 1) variable omission
# 2) uniform numeric value applied to KAPPA coefficient
# 3) heterogeneous values allocated to SUBPAR via data frame

ems_model(model_file = GTAP_RE[["model_file"]],
          closure_file = GTAP_RE[["closure_file"]],
          var_omit = c("atall", "avaall", "tfe", "tfm", "tgd", "tgm", "tid", "tim"),
          KAPPA = 0.03,
          SUBPAR = SUBPAR)

```

ems_option_get	<i>Get default package options</i>
----------------	------------------------------------

Description

ems_option_get() returns default package options. If name is NULL (the default), all option values are returned as a list.

Usage

```
ems_option_get(name = NULL)
```

Arguments

name	Name of the option for which to retrieve a value. One of: <ul style="list-style-type: none"> • NULL Returns all option values. • "verbose" Logical. If FALSE, function-specific diagnostics are silenced. • "tempdir" Character. Directory used for temporary file storage during a model run. • "ndigits" Integer. Exact number of digits to the right of the decimal point written to file for numeric type double. • "accuracy_threshold" Numeric. Threshold (converted to a percentage) against which 4-digit precision is compared. • "check_shock_status" Logical. If FALSE, no check on shock element endogenous/exogenous status is conducted. • "timestep_header" Character. Coefficient containing a numeric vector of timestep intervals. • "n_timestep_header" Character. Coefficient containing a numeric vector length one with sum of timestep intervals. • "full_exclude" Character vector. Headers to fully exclude from all aspects of the model run. • "docker_tag" Character. Docker tag specifying which Docker image to use.
------	--

Value

If name is NULL, a named list of all current option values. Otherwise, the value of the requested option.

See Also

[ems_option_set\(\)](#) for setting package options. [ems_option_reset\(\)](#) for resetting package options.

Examples

```
# Retrieve all options values
ems_option_get()

# Retrieve option value for `ndigits`
ems_option_get("ndigits")
```

ems_option_reset	<i>Reset to default package options</i>
------------------	---

Description

ems_option_reset() resets all package options to default values.

Usage

```
ems_option_reset()
```

Value

invisible(NULL), called for its side effects.

See Also

[ems_option_set\(\)](#) for setting package options. [ems_option_get\(\)](#) for retrieving package options.

Examples

```
# Set multiple options
ems_option_set(verbose = FALSE, ndigits = 8)

# Retrieve modified option value for `verbose`
ems_option_get("verbose")

# Reset options to default values
ems_option_reset()

# Retrieve default option value for `verbose`
ems_option_get("verbose")
```

ems_option_set *Set advanced package options*

Description

ems_option_set() allows the user to customize advanced options.

Usage

```
ems_option_set(
  verbose = NULL,
  tempdir = NULL,
  ndigits = NULL,
  accuracy_threshold = NULL,
  check_shock_status = NULL,
  timestep_header = NULL,
  n_timestep_header = NULL,
  full_exclude = NULL,
  docker_tag = NULL
)
```

Arguments

verbose	Logical of length 1 (default is TRUE). If FALSE, function-specific diagnostics are silenced.
tempdir	A character vector length 1. Default is what is returned by tempdir().
ndigits	Integer (default is 6). Exact number of digits to the right of the decimal point to be written to file for numeric type double. This value is passed to the format() nsmall argument and round() digits argument.
accuracy_threshold	Numeric length 1 (default 0.8), converted to a percentage. 4-digit precision is compared against this threshold; a warning is generated if it is not met.
check_shock_status	Logical of length 1 (default is TRUE). If FALSE, no check on shock element endogenous/exogenous status is conducted.
timestep_header	A character vector length 1 (default is "YEAR"). Coefficient containing a numeric vector of timestep intervals. For novel intertemporal models — modify with caution.
n_timestep_header	A character vector length 1 (default is "NTSP"). Coefficient containing a numeric vector length one with sum of timestep intervals. For novel intertemporal models — modify with caution.
full_exclude	A character vector of variable length (default is c("DREL", "DVER", "XXCR", "XXCD", "XXCP", "SLUG", "EFLG")). Headers to fully exclude from all aspects of the model run. Failure to designate these headers properly will result in errors. Modify with caution.

docker_tag Character length 1 (default "latest"). Docker tag specifying which Docker image to use.

Value

invisible(NULL), called for its side effects.

See Also

[ems_option_get\(\)](#) for retrieving package options. [ems_option_reset\(\)](#) for resetting package options.

Examples

```
# Set multiple options
ems_option_set(verbose = FALSE, ndigits = 8)

# Retrieve value of `verbose`
ems_option_get("verbose")

# Reset options to default values
ems_option_reset()
```

ems_scenario_shock *Prepare a scenario shock*

Description

Prepares heterogeneous absolute-value trajectory shocks. Inputs are aggregated according to the set mappings in [ems_data\(\)](#) and converted to percentage changes internally, making scenario shocks portable across aggregations. All tuples must be present in `input` at full pre-aggregation resolution. Partial variable scenario shocks are not permitted.

See the [scenario shocks](#) chapter of the user manual for an illustrative example.

Usage

```
ems_scenario_shock(var, input)
```

Arguments

`var` Character of length 1, the variable to be shocked.

`input` Path to a CSV file, or a data frame or data frame extension (e.g., `tibble`, `data.table`). Must contain a `Value` column denominated in the actual values of the variable to be shocked (e.g., million USD) and a `Year` column of chronological years, plus one column per set index for the variable. Note that set names must conform to the model-specific hybrid format (e.g., ACTSa, REGr). Column order is inconsequential.

Value

A list object with shock configuration to be passed to the shock argument of `ems_deploy()`.

See Also

`ems_deploy()` for loading the output of this function. `ems_swap()` for changing the standard model closure.

ems_solve

Configure and solve model

Description

Calls the `teems-solver` Docker image to solve the constrained optimization problem. Runs singularity, accuracy, and error checks on the results and structures outputs with `ems_compose()`.

Usage

```
ems_solve(
  cmf_path,
  solution_method = c("Johansen", "mod_midpoint"),
  matrix_method = c("LU", "DBBD", "SBBBD", "NDBBD"),
  n_subintervals = 1L,
  steps = c(2L, 4L, 8L),
  n_tasks = 1L,
  laA = 300L,
  laD = 200L,
  laDi = 500L,
  suppress_outputs = FALSE,
  terminal_run = FALSE,
  append_args = NULL
)
```

Arguments

`cmf_path` Character length 1, path to the CMF file generated by `ems_deploy()`.

`solution_method`

The solution method to be used, default is the one-step "Johansen". Choices:

- "Johansen": The Johansen method is fast however it should only be used as a rough approximation due to handling of nonlinear equations. See: [Johansen](#)
- "mod_midpoint": The modified midpoint method performs multiple passes. See: [modified midpoint](#)

`matrix_method` Character of length 1, matrix solution method (default is "LU"). Choices:

- "LU": Standard LU decomposition, the most robust and potentially slowest for a large model. For use with both static and dynamic models.

- "DBBD": Doubly bordered block diagonal, parallel solution method for static models. Potentially faster than "LU" although less robust.
- "SBBD": Singly bordered block diagonal, parallel solution method for intertemporal models. Potentially faster than "LU" although less robust.
- "NDBBD": Nested doubly bordered block diagonal, parallel solution method for large intertemporal models with many timesteps.

n_subintervals	Integer length 1 (default is 1L), number of subintervals for the applied shock. More subintervals may alleviate accuracy issues stemming from large shock magnitudes.
steps	Integer length 3 (default is c(2L, 4L, 8L)). A vector of steps for the modified midpoint method, must be all odd or all even and length 3. A larger number of steps may improve accuracy for some model runs.
n_tasks	Integer length 1 (default is 1L), number of tasks to run in parallel. Must be 1L if "matrix_method" == "LU".
laA	Integer length 1 (default is 300L). Memory parameter (%) for "LU" and "SBBD".
laD	Integer length 1 (default is 200L). Memory parameter (%) for "DBBD" and "NDBBD".
laDi	Integer length 1 (default is 500L). Memory parameter (%) for "NDBBD" only.
suppress_outputs	Logical length 1 (default is FALSE). When TRUE solver outputs are not automatically converted into structured data with <code>ems_compose()</code> .
terminal_run	Logical length 1 (default is FALSE). When TRUE, the function is exited without running the solver. This allows the user to close any R IDE or other programs prior to running from the terminal. When TRUE solver outputs are not automatically converted into structured data with <code>ems_compose()</code> .
append_args	Character vector (default NULL). Additional arguments appended to the Docker run command (e.g., c("-smlthreads 2", "-maxthreads 2")).

Details

Increase `laA`, `laD`, or `laDi` gradually if the solver returns "Error return from MA48B/BD because LA is ...". The relevant parameter depends on the `matrix_method` in use (see above).

Value

A tibble of model output variables and coefficients. `invisible(NULL)` if `suppress_outputs = TRUE`. Instructions for terminal execution if `terminal_run = TRUE`.

See Also

`ems_deploy()` for generating `"cmf_path"`. `solve_in_situ()` for calling the solver on existing input files. `ems_compose()` for structuring data when `"suppress_outputs"` or `"terminal_run"` is TRUE.

Examples

```
## Not run:
# The following examples require the teems solver to be built.
# See https://teemsphere.github.io/ to get started.

# Solving a static model with Johansen:
ems_solve(cmf_path)

# Solving a dynamic model with the SBB method:
ems_solve(cmf_path,
          solution_method = "mod_midpoint",
          matrix_method = "SBB",
          n_tasks = 6)

## End(Not run)
```

 ems_swap

Prepare closure swaps

Description

Prepares a closure swap for input into the `swap_in` or `swap_out` arguments of `ems_deploy()`.

Usage

```
ems_swap(var, ...)
```

Arguments

<code>var</code>	Character of length 1, model variable to swap.
<code>...</code>	Optional named arguments restricting the swap to a subset of variable elements. Each name must use the model-specific set-index format (set name concatenated with its index letter, e.g., REGr, COMMc). Unspecified sets default to all elements. SET corresponds to the generic set name while <code>i</code> is specific to the variable index. Three forms are accepted: <ul style="list-style-type: none"> • Single element: SETi = "element" • Multiple elements: SETi = c("element1", "element2") • Subset: SETi = "SUBSET"

Details

`ems_swap()` return values have no purpose used in isolation. The closure provided to the `closure_file` argument of `ems_model()` will be used if no swaps are specified. Note that full variable swaps can be directly inputted as a character string in the `"swap_in"` and `"swap_out"` arguments of `ems_deploy()`.

Swaps "in" (joining the list of exogenous variables) are processed prior to swaps out (leaving the list of exogenous variables), and in the same order as they are loaded into `ems_deploy()`.

Value

A list of class "full" or "partial" and "single" or "multi" containing the swap specification. Intended for use as swap_in or swap_out in `ems_deploy()`.

See Also

`ems_model()` for parsing and modifying model and closure files. `ems_deploy()` for loading the output of this function.

Examples

```
# Full variable swaps
ems_swap("tfd") # out
ems_swap("qfd") # in

# Partial variable swaps with uniform "PROD_COMM" set
# application (note distinction between "REGr" and "REGs") in
# the classic GTAP model (version 6.2)
ems_swap("tfd", TRAD_COMMi = "food", REGr = "chn") # out
ems_swap("qfd", TRAD_COMMi = "food", REGs = "chn") # in

# Partial variable, multiple element swaps with uniform "ACTS"
# set application in the standard GTAP model (version 7.0).
ems_swap("tfd", COMMc = c("food", "crops"), REGr = "usa") # out
ems_swap("qfd", COMMc = c("food", "crops"), REGr = "usa") # in

# Valid subsets may also be selected.
ems_swap("txs", COMMc = "NMRG", ALLTIMEt = "FWDTIME") # out
ems_swap("qxs", COMMc = "NMRG", ALLTIMEt = "FWDTIME") # in
```

ems_uniform_shock	<i>Prepare a uniform shock</i>
-------------------	--------------------------------

Description

Prepares a uniform shock to be implemented as a single percentage-change value uniformly across all or a subset of variable tuples.

Usage

```
ems_uniform_shock(var, value, ...)
```

Arguments

var	Character of length 1, the variable to be shocked.
value	Numeric length 1, percentage-change value of the shock.

... Optional named arguments restricting the shock to a subset of the variable's elements. Each name must use the model-specific set-index format (set name concatenated with its index letter, e.g., REGr, COMMc). Value may consist of one or more elements or a subset. When omitted, the shock is applied to all variable tuples.

Value

A list object with shock configuration to be passed to the shock argument of [ems_deploy\(\)](#).

See Also

[ems_deploy\(\)](#) for loading the output of this function. [ems_swap\(\)](#) for changing the standard model closure.

Examples

```
# Full uniform: 2% shock imposed on all afeall tuples
ems_uniform_shock(var = "afeall", value = 2)

# Partial uniform by element: applied only to the "chn"
# element in set REGr (REG). Note that set designations must
# consist of the concatenation of the standard set (e.g., REG)
# and variable-specific index (e.g., r).
ems_uniform_shock(var = "afeall",
                  REGr = "chn",
                  value = 2)

# Partially uniform by subset and element: applied to "chn"
# and "usa" across the MARG and FWDTIME subsets.
ems_uniform_shock(var = "qxs",
                  COMMc = "MARG",
                  REGs = c("chn", "usa"),
                  ALLTIMEt = "FWDTIME",
                  value = -1)
```

GTAP_convert

Convert GTAP HAR files and data formats

Description

GTAP_convert() converts GTAP HAR database files into lists of arrays as well as converts between the classic v6.2 and standard v7.0 data formats. The output can then be passed directly to [ems_data\(\)](#) via its dat_input, par_input, and set_input arguments, or written to disk for later use.

Usage

```
GTAP_convert(dat_har, par_har, set_har, target = NULL)
```

Arguments

dat_har	Character vector of length 1, file name in working directory or path to a GTAP HAR file with basedata coefficient data.
par_har	Character vector of length 1, file name in working directory or path to a GTAP HAR file with parameter coefficient data.
set_har	Character vector of length 1, file name in working directory or path to a GTAP HAR file with set elements and attributes.
target	Character vector of length 1 (default is NULL). The target format to convert to. Currently supports "GTAPv6" or "GTAPv7". If NULL, no data format conversion will take place.

Value

A named list with elements `dat`, `par`, and `set`, each containing a list of arrays corresponding to database headers. Suitable for direct use with `ems_data()`.

See Also

`ems_data()` for loading and preparing converted data for a model run.

Examples

```
## Not run:
# The following examples require input data. See
# https://teemsphere.github.io/ to get started.

# Convert HAR files to lists of arrays
ls_arrays <- GTAP_convert(
  dat_har = "path/to/gsdmdat.har",
  par_har = "path/to/gsdmpar.har",
  set_har = "path/to/gsdmset.har"
)

# Convert v7.0 files to v6.2 format
converted2v6 <- GTAP_convert(
  dat_har = "v7_data/gsdmdat.har",
  par_har = "v7_data/gsdmpar.har",
  set_har = "v7_data/gsdmset.har",
  target = "GTAPv6"
)

# Convert v6.2 files to v7.0 format
converted2v7 <- GTAP_convert(
  dat_har = "v6_data/gsdmdat.har",
  par_har = "v6_data/gsdmpar.har",
  set_har = "v6_data/gsdmset.har",
  target = "GTAPv7"
)

## End(Not run)
```

solve_in_situ	<i>Configure and solve model in-situ</i>
---------------	--

Description

Calls the **teems-solver** Docker image directly with user-supplied input files, bypassing the `ems_data()` / `ems_model()` / `ems_deploy()` pipeline. All input files must be provided in their final form.

Usage

```

solve_in_situ(
  ...,
  model_dir,
  model_file,
  closure_file,
  shock_file,
  solution_method = c("Johansen", "mod_midpoint"),
  matrix_method = c("LU", "DBBD", "SBBD", "NDBBD"),
  n_subintervals = 1L,
  steps = c(2L, 4L, 8L),
  n_tasks = 1L,
  laA = 300L,
  laD = 200L,
  laDi = 500L,
  suppress_outputs = FALSE,
  terminal_run = FALSE,
  append_args = NULL
)

```

Arguments

...	Named arguments corresponding to input files necessary for an in-situ model run. Names must correspond to "File" statements within the model Tablo file. Values correspond to file paths where these files are found. No checks or modifications are conducted on input files used in this manner. All model declared input files as well as "model_file", "closure_file", and "shock_file" are required for in-situ model runs.
model_dir	Character of length 1, base directory where input files will be copied (if not all already present) and model outputs will be written.
model_file	Character vector length 1, path to a TABLO file with .tab extension. See teems-models for vetted models and compatible Tablo file formatting.
closure_file	Character vector length 1, path to a closure file with .cls extension. See teems-models for closure file formatting.
shock_file	A character vector length 1, path to a file with .shf extension representing a fully prepared shock file. No checks or modifications are carried out on this file.

solution_method	The solution method to be used, default is the one-step "Johansen". Choices: <ul style="list-style-type: none"> "Johansen": The Johansen method is fast however it should only be used as a rough approximation due to handling of nonlinear equations. See: Johansen "mod_midpoint": The modified midpoint method performs multiple passes. See: modified midpoint
matrix_method	Character of length 1, matrix solution method (default is "LU"). Choices: <ul style="list-style-type: none"> "LU": Standard LU decomposition, the most robust and potentially slowest for a large model. For use with both static and dynamic models. "DBBD": Doubly bordered block diagonal, parallel solution method for static models. Potentially faster than "LU" although less robust. "SBBD": Singly bordered block diagonal, parallel solution method for intertemporal models. Potentially faster than "LU" although less robust. "NDBBD": Nested doubly bordered block diagonal, parallel solution method for large intertemporal models with many timesteps.
n_subintervals	Integer length 1 (default is 1L), number of subintervals for the applied shock. More subintervals may alleviate accuracy issues stemming from large shock magnitudes.
steps	Integer length 3 (default is c(2L, 4L, 8L)). A vector of steps for the modified midpoint method, must be all odd or all even and length 3. A larger number of steps may improve accuracy for some model runs.
n_tasks	Integer length 1 (default is 1L), number of tasks to run in parallel. Must be 1L if "matrix_method" == "LU".
laA	Integer length 1 (default is 300L). Memory parameter (%) for "LU" and "SBBD".
laD	Integer length 1 (default is 200L). Memory parameter (%) for "DBBD" and "NDBBD".
laDi	Integer length 1 (default is 500L). Memory parameter (%) for "NDBBD" only.
suppress_outputs	Logical length 1 (default is FALSE). When TRUE solver outputs are not automatically converted into structured data with ems_compose() .
terminal_run	Logical length 1 (default is FALSE). When TRUE, the function is exited without running the solver. This allows the user to close any R IDE or other programs prior to running from the terminal. When TRUE solver outputs are not automatically converted into structured data with ems_compose() .
append_args	Character vector (default NULL). Additional arguments appended to the Docker run command (e.g., c("-smlthreads 2", "-maxthreads 2")).

Value

A tibble containing model output variables and coefficients. Alternatively, if "suppress_outputs" is TRUE, file path to a CMF file that may be used with [ems_compose\(\)](#).

See Also

[ems_solve\(\)](#) for the standard package-supported solver.

Examples

```
## Not run:  
# The following examples require the teems solver to be built.  
# See https://teemsphere.github.io/ to get started.  
  
solve_in_situ(  
  GTAPDATA = "path/to/dat_file.txt",  
  GTAPPARM = "path/to/par_file.txt",  
  GTAPSETS = "path/to/sets_file.txt",  
  model_dir = "existing/dir/for/model_run",  
  model_file = "path/to/model_file.tab",  
  closure_file = "path/to/closure_file.cls",  
  shock_file = "path/to/shock_file.shf"  
)  
  
## End(Not run)
```

Index

ems_compose, [2](#)
ems_compose(), [15](#), [16](#), [22](#)
ems_custom_shock, [3](#)
ems_custom_shock(), [6](#), [7](#)
ems_data, [4](#)
ems_data(), [6](#), [7](#), [10](#), [14](#), [19–21](#)
ems_deploy, [6](#)
ems_deploy(), [2–5](#), [10](#), [15–19](#), [21](#)
ems_example, [8](#)
ems_example(), [10](#)
ems_model, [9](#)
ems_model(), [6](#), [7](#), [17](#), [18](#), [21](#)
ems_option_get, [11](#)
ems_option_get(), [12](#), [14](#)
ems_option_reset, [12](#)
ems_option_reset(), [12](#), [14](#)
ems_option_set, [13](#)
ems_option_set(), [12](#)
ems_scenario_shock, [14](#)
ems_scenario_shock(), [6](#), [7](#)
ems_solve, [15](#)
ems_solve(), [2](#), [6](#), [7](#), [22](#)
ems_swap, [17](#)
ems_swap(), [3](#), [7](#), [15](#), [19](#)
ems_uniform_shock, [18](#)
ems_uniform_shock(), [6](#), [7](#)

GTAP_convert, [19](#)
GTAP_convert(), [4](#), [5](#), [8](#), [9](#)

solve_in_situ, [21](#)
solve_in_situ(), [16](#)