\*\*\*

\* MESSAGE core formulation

\* ========================

\*

\* The |MESSAGEix| systems-optimization model minimizes total costs while satisfying given demand levels for commodities/services and considering a broad range of technical/engineering constraints and societal restrictions (e.g. bounds on greenhouse gas emissions, pollutants, system reliability).

\* Demand levels are static (i.e. non-elastic), but the demand response can be integrated by linking |MESSAGEix| to the single sector general-economy MACRO model included in this framework.

\*

\* For the complete list of sets, mappings and parameters, refer to the auto-documentation page :ref:`sets\_maps\_def` and :ref:`parameter\_def`.

\* The mathematical notation that is used to represent sets and mappings in the equations below can also be found in the tables in :ref:`sets\_maps\_def`.

\*

\* .. contents::

\* :local:

\* :backlinks: none

\*

\*\*\*

\*----------------------------------------------------------------------------------------------------------------------\*

\* Variable definitions \*

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_section\_variable\_def:

\*

\* Variable definitions

\* --------------------

\*

\* .. \_section\_decision\_variable\_def:

\*

\* Decision variables

\* ^^^^^^^^^^^^^^^^^^

\*

\* .. \_ACT:

\* .. \_CAP:

\* .. \_CAP\_NEW:

\* .. \_LAND:

\* .. \_STOCK\_CHG:

\*

\* =============================================================== ====================================================================================

\* Variable Explanatory text

\* =============================================================== ====================================================================================

\* :math:`\text{OBJ} \in \mathbb{R}` Objective value of the optimization program

\* :math:`\text{EXT}\_{n,c,g,y} \in \mathbb{R}\_+` Extraction of non-renewable/exhaustible resources from reserves

\* :math:`\text{STOCK}\_{n,c,l,y} \in \mathbb{R}\_+` Quantity in stock (storage) at start of period :math:`y`

\* :math:`\text{STOCK\_CHG}\_{n,c,l,y,h} \in \mathbb{R}` Input or output quantity into intertemporal commodity stock (storage)

\* :math:`\text{COST\_NODAL}\_{n,y} \in \mathbb{R}` System costs at the node level over time

\* :math:`\text{REN}\_{n,t,c,g,y,h} \in \mathbb{R}\_+` Activity of renewable technologies per grade

\* :math:`\text{CAP\_NEW}\_{n,t,y} \in \mathbb{R}\_+` Newly installed capacity (yearly average over period duration)

\* :math:`\text{CAP}\_{n,t,y^V,y} \in \mathbb{R}\_+` Maintained capacity in year :math:`y` of vintage :math:`y^V`

\* :math:`\text{CAP\_FIRM}\_{n,t,c,l,y,q} \in \mathbb{R}\_+` Capacity counting towards firm (dispatchable)

\* :math:`\text{ACT}\_{n,t,y^V,y,m,h} \in \mathbb{R}` Activity of a technology (by vintage, mode, subannual time)

\* :math:`\text{ACT\_RATING}\_{n,t,y^V,y,c,l,h,q} \in \mathbb{R}\_+` Auxiliary variable for activity attributed to a particular rating bin [#ACT\_RATING]\_

\* :math:`\text{CAP\_NEW\_UP}\_{n,t,y} \in \mathbb{R}\_+` Relaxation of upper dynamic constraint on new capacity

\* :math:`\text{CAP\_NEW\_LO}\_{n,t,y} \in \mathbb{R}\_+` Relaxation of lower dynamic constraint on new capacity

\* :math:`\text{ACT\_UP}\_{n,t,y,h} \in \mathbb{R}\_+` Relaxation of upper dynamic constraint on activity [#ACT\_BD]\_

\* :math:`\text{ACT\_LO}\_{n,t,y,h} \in \mathbb{R}\_+` Relaxation of lower dynamic constraint on activity [#ACT\_BD]\_

\* :math:`\text{LAND}\_{n,s,y} \in [0,1]` Relative share of land-use scenario (for land-use model emulator)

\* :math:`\text{EMISS}\_{n,e,\widehat{t},y} \in \mathbb{R}` Auxiliary variable for aggregate emissions by technology type

\* :math:`\text{REL}\_{r,n,y} \in \mathbb{R}` Auxiliary variable for left-hand side of relations (linear constraints)

\* :math:`\text{COMMODITY\_USE}\_{n,c,l,y} \in \mathbb{R}` Auxiliary variable for amount of commodity used at specific level

\* :math:`\text{COMMODITY\_BALANCE}\_{n,c,l,y,h} \in \mathbb{R}` Auxiliary variable for right-hand side of :ref:`commodity\_balance`

\* :math:`\text{STORAGE}\_{n,t,m,l,c,y,h} \in \mathbb{R}` State of charge or content of storage at each sub-annual time slice

\* :math:`\text{STORAGE\_CHARGE}\_{n,t,m,l,c,y,h} \in \mathbb{R}` Charging of storage in each sub-annual time slice (negative for discharging)

\* =============================================================== ====================================================================================

\*

\* The index :math:`y^V` is the year of construction (vintage) wherever it is necessary to

\* clearly distinguish between year of construction and the year of operation.

\*

\* All decision variables are by year, not by (multi-year) period, except :math:`\text{STOCK}\_{n,c,l,y}`.

\* In particular, the new capacity variable :math:`\text{CAP\_NEW}\_{n,t,y}` has to be multiplied by the number of years

\* in a period :math:`|y| = \text{duration\_period}\_{y}` to determine the available capacity :math:`\text{CAP}\_{n,t,y^V,y}`

\* in subsequent periods (assuming the newly build capacity is not immediately decommissioned):

\*

\* :math:`\text{CAP}\_{n,t,y^V,y} = \text{CAP\_NEW}\_{n,t,y} \cdot \text{duration\_period}\_{y}`

\*

\* :math:`\text{CAP\_NEW}\_{n,t,y}` is therefore the amount of newly installed capacity \*in one year\* and

\* :math:`\text{CAP}\_{n,t,y^V,y}` the amount, which is installed at the \*end of a (usually multi-year) period\*.

\* This formulation gives more flexibility when it comes to using periods of different duration

\* (more intuitive comparison across different periods).

\*

\* The current model framework allows both input or output normalized formulation.

\* This will affect the parametrization, see Section :ref:`efficiency\_output` for more details.

\*

\* .. [#ACT\_RATING] The auxiliary variable :math:`\text{ACT\_RATING}\_{n,t,y^V,y,c,l,h,q}` is defined in terms of input or

\* output of the technology.

\*

\* .. [#ACT\_BD] The dynamic activity constraints are implemented as summed over all modes;

\* therefore, the variables for the relaxation are not indexed over the set ``mode``.

\*

\*\*\*

Variables

OBJ 'Objective value of the optimisation problem'

COMMODITY\_BALANCE(node,commodity,level,year\_all,time) 'Balance of commodity flow'

;

Positive Variables

\* resource production/extraction variable

EXT(node,commodity,grade,year\_all) extraction of fossil resources

\* commodity in inter-temporal stock

STOCK(node,commodity,level,year\_all) total quantity in intertemporal stock (storage)

\* use of renewable resources

REN(node, tec, commodity, grade, year\_all, time) activity of renewables specified per renewables grade

\* investment and capacity variables

CAP\_NEW(node,tec,year\_all) new capacity by year

CAP(node,tec,vintage,year\_all) total installed capacity by year

CAP\_FIRM(node,tec,commodity,level,year\_all) capacity counting towards system reliability constraints

\* auxiliary variable for distributing total activity of a technology to a number of "rating bins"

ACT\_RATING(node,tec,vintage,year\_all,commodity,level,time,rating)

\* variables for soft relaxation of dynamic activity constraints

CAP\_NEW\_UP(node,tec,year\_all) relaxation variable for dynamic constraints on new capacity (upwards)

CAP\_NEW\_LO(node,tec,year\_all) relaxation variable for dynamic constraints on new capacity (downwards)

ACT\_UP(node,tec,year\_all,time) relaxation variable for dynamic constraints on activity (upwards)

ACT\_LO(node,tec,year\_all,time) relaxation variable for dynamic constraints on activity (downwards)

\* land-use model emulator

LAND(node,land\_scenario,year\_all) relative share of land-use scenario

\* content of storage

STORAGE(node,tec,mode,level,commodity,year\_all,time) state of charge (SoC) of storage at each sub-annual time slice (positive)

;

Variables

\* intertemporal stock variables (input or output quantity into the stock)

STOCK\_CHG(node,commodity,level,year\_all,time) annual input into and output from stocks of commodities

\* technology activity variables (can be negative for some technologies, upper and lower bounds stated explicitly)

ACT(node,tec,vintage,year\_all,mode,time) activity of technology by mode-year-timeperiod

\* auxiliary variables for finrm-capacity formulation

COMMODITY\_USE(node,commodity,level,year\_all) total amount of a commodity & level that was used or consumed

\* nodal system costs over time

COST\_NODAL(node, year\_all) system costs at the node level over time

\* auxiliary variable for aggregate emissions by technology type and land-use model emulator

EMISS(node,emission,type\_tec,year\_all) aggregate emissions by technology type and land-use model emulator

\* auxiliary variable for left-hand side of relations (linear constraints)

REL(relation,node,year\_all) auxiliary variable for left-hand side of user-defined relations

\* change in the content of storage device

STORAGE\_CHARGE(node,tec,mode,level,commodity,year\_all,time) charging of storage in each time slice (negative for discharge)

;

\*\*\*

\* .. \_section\_auxiliary\_variable\_def:

\*

\* Auxiliary variables

\* ^^^^^^^^^^^^^^^^^^^

\*

\* .. list-table::

\* :header-rows: 1

\*

\* \* - Variable

\* - Explanatory text

\* \* - :math:`\text{DEMAND}\_{n,c,l,y,h} \in \mathbb{R}`

\* - Demand level (in equilibrium with MACRO integration)

\* \* - :math:`\text{PRICE\_COMMODITY}\_{n,c,l,y,h} \in \mathbb{R}`

\* - Commodity price (undiscounted marginals of :ref:`commodity\_balance\_gt` and :ref:`commodity\_balance\_lt`)

\* \* - :math:`\text{PRICE\_EMISSION}\_{n,\widehat{e},\widehat{t},y} \in \mathbb{R}`

\* - Emission price (undiscounted marginals of :ref:`emission\_equivalence`)

\* \* - :math:`\text{COST\_NODAL\_NET}\_{n,y} \in \mathbb{R}`

\* - System costs at the node level net of energy trade revenues/cost

\* \* - :math:`\text{GDP}\_{n,y} \in \mathbb{R}`

\* - Gross domestic product (GDP) in market exchange rates for MACRO reporting

\*

\* .. warning::

\* Please be aware that transitioning from one period length to another

\* for consecutive periods may result in incorrect value for :math:`\text{PRICE\_EMISSION}`.

\* See :issue:`723` for further information and :pull:`726` for a potential resolution.

\*\*\*

Variables

\* auxiliary variables for demand, prices, costs and GDP (for reporting when MESSAGE is run with MACRO)

DEMAND(node,commodity,level,year\_all,time) demand

PRICE\_COMMODITY(node,commodity,level,year\_all,time) commodity price (derived from marginals of COMMODITY\_BALANCE constraint)

PRICE\_EMISSION(node,type\_emission,type\_tec,year\_all) emission price (derived from marginals of EMISSION\_EQUIVALENCE constraint)

COST\_NODAL\_NET(node,year\_all) system costs at the node level over time including effects of energy trade

GDP(node,year\_all) gross domestic product (GDP) in market exchange rates for MACRO reporting

;

\*----------------------------------------------------------------------------------------------------------------------\*

\* auxiliary bounds on activity variables (debugging mode, avoid inter-vintage arbitrage, investment technology) \*

\*----------------------------------------------------------------------------------------------------------------------\*

\* include upper and lower bounds (to avoid unbounded models)

%AUX\_BOUNDS% ACT.lo(node,tec,year\_all,year\_all2,mode,time)$( map\_tec\_lifetime(node,tec,year\_all,year\_all2)

%AUX\_BOUNDS% AND map\_tec\_act(node,tec,year\_all2,mode,time) ) = -%AUX\_BOUND\_VALUE% ;

%AUX\_BOUNDS% ACT.up(node,tec,year\_all,year\_all2,mode,time)$( map\_tec\_lifetime(node,tec,year\_all,year\_all2)

%AUX\_BOUNDS% AND map\_tec\_act(node,tec,year\_all2,mode,time) ) = %AUX\_BOUND\_VALUE% ;

\* to avoid "inter-vintage arbitrage" (across different vintages of technologies), all activities that

\* have positive lower bounds are assumed to be non-negative

ACT.lo(node,tec,year\_all,year\_all2,mode,time)$( map\_tec\_lifetime(node,tec,year\_all,year\_all2)

AND map\_tec\_act(node,tec,year\_all2,mode,time) AND bound\_activity\_lo(node,tec,year\_all2,mode,time) >= 0 ) = 0 ;

\* previous implementation using upper bounds

\* ACT.lo(node,tec,year\_all,year\_all2,mode,time)$( map\_tec\_lifetime(node,tec,year\_all,year\_all2)

\* AND map\_tec\_act(node,tec,year\_all2,mode,time)

\* AND ( NOT bound\_activity\_up(node,tec,year\_all2,mode,time)

\* OR bound\_activity\_up(node,tec,year\_all2,mode,time) >= 0 ) ) = 0 ;

\* assume that all "investment" technologies must have non-negative activity levels

ACT.lo(node,inv\_tec,year\_all,year\_all2,mode,time)$( map\_tec\_lifetime(node,inv\_tec,year\_all,year\_all2)

AND map\_tec\_act(node,inv\_tec,year\_all2,mode,time) ) = 0 ;

\*----------------------------------------------------------------------------------------------------------------------\*

\* fixing variables to pre-specified values \*

\*----------------------------------------------------------------------------------------------------------------------\*

EXT.fx(node,commodity,grade,year\_all)$( is\_fixed\_extraction(node,commodity,grade,year\_all) ) =

fixed\_extraction(node,commodity,grade,year\_all);

STOCK.fx(node,commodity,level,year\_all)$( is\_fixed\_stock(node,commodity,level,year\_all) ) =

fixed\_stock(node,commodity,level,year\_all) ;

CAP\_NEW.fx(node,tec,year\_all)$( is\_fixed\_new\_capacity(node,tec,year\_all) ) =

fixed\_new\_capacity(node,tec,year\_all) ;

CAP.fx(node,tec,vintage,year\_all)$( is\_fixed\_capacity(node,tec,vintage,year\_all) ) =

fixed\_capacity(node,tec,vintage,year\_all) ;

ACT.fx(node,tec,vintage,year\_all,mode,time)$( is\_fixed\_activity(node,tec,vintage,year\_all,mode,time) ) =

fixed\_activity(node,tec,vintage,year\_all,mode,time) ;

LAND.fx(node,land\_scenario,year\_all)$( is\_fixed\_land(node,land\_scenario,year\_all) ) =

fixed\_land(node,land\_scenario,year\_all) ;

\*----------------------------------------------------------------------------------------------------------------------\*

\* auxiliary variables for debugging mode (identifying infeasibilities) \*

\*----------------------------------------------------------------------------------------------------------------------\*

\* report mapping for debugging

Set

AUX\_ACT\_BOUND\_UP(node,tec,year\_all,year\_all2,mode,time) indicator whether auxiliary upper bound on activity is binding

AUX\_ACT\_BOUND\_LO(node,tec,year\_all,year\_all2,mode,time) indicator whether auxiliary upper bound on activity is binding

;

\* slack variables for debugging

Positive variables

SLACK\_COMMODITY\_EQUIVALENCE\_UP(node,commodity,level,year\_all,time) slack variable for commodity balance (upwards)

SLACK\_COMMODITY\_EQUIVALENCE\_LO(node,commodity,level,year\_all,time) slack variable for commodity balance (downwards)

SLACK\_CAP\_NEW\_BOUND\_UP (node,tec,year\_all) slack variable for bound on new capacity (upwards)

SLACK\_CAP\_NEW\_BOUND\_LO (node,tec,year\_all) slack variable for bound on new capacity (downwards)

SLACK\_CAP\_TOTAL\_BOUND\_UP (node,tec,year\_all) slack variable for upper bound on total installed capacity

SLACK\_CAP\_TOTAL\_BOUND\_LO (node,tec,year\_all) slack variable for lower bound on total installed capacity

SLACK\_CAP\_NEW\_DYNAMIC\_UP(node,tec,year\_all) slack variable for dynamic new capacity constraint (upwards)

SLACK\_CAP\_NEW\_DYNAMIC\_LO(node,tec,year\_all) slack variable for dynamic new capacity constraint (downwards)

SLACK\_ACT\_BOUND\_UP(node,tec,year\_all,mode,time) slack variable for upper bound on activity

SLACK\_ACT\_BOUND\_LO(node,tec,year\_all,mode,time) slack variable for lower bound on activity

SLACK\_ACT\_DYNAMIC\_UP(node,tec,year\_all,time) slack variable for dynamic activity constraint relaxation (upwards)

SLACK\_ACT\_DYNAMIC\_LO(node,tec,year\_all,time) slack variable for dynamic activity constraint relaxation (downwards)

SLACK\_LAND\_SCEN\_UP(node,land\_scenario,year\_all) slack variable for dynamic land scenario constraint relaxation (upwards)

SLACK\_LAND\_SCEN\_LO(node,land\_scenario,year\_all) slack variable for dynamic land scenario constraint relaxation (downwards)

SLACK\_LAND\_TYPE\_UP(node,year\_all,land\_type) slack variable for dynamic land type constraint relaxation (upwards)

SLACK\_LAND\_TYPE\_LO(node,year\_all,land\_type) slack variable for dynamic land type constraint relaxation (downwards)

SLACK\_RELATION\_BOUND\_UP(relation,node,year\_all) slack variable for upper bound of generic relation

SLACK\_RELATION\_BOUND\_LO(relation,node,year\_all) slack variable for lower bound of generic relation

;

\*----------------------------------------------------------------------------------------------------------------------\*

\* equation definitions \*

\*----------------------------------------------------------------------------------------------------------------------\*

Equations

OBJECTIVE objective value of the optimisation problem

COST\_ACCOUNTING\_NODAL cost accounting at node level over time

EXTRACTION\_EQUIVALENCE auxiliary equation to simplify the resource extraction formulation

EXTRACTION\_BOUND\_UP upper bound on extraction (by grade)

RESOURCE\_CONSTRAINT constraint on resources remaining in each period (maximum extraction per period)

RESOURCE\_HORIZON constraint on extraction over entire model horizon (resource volume in place)

COMMODITY\_BALANCE\_AUX(node,commodity,level,year\_all,time) 'Auxiliary equation for calculation of commodity balance'

COMMODITY\_BALANCE\_GT commodity supply greater than or equal demand

COMMODITY\_BALANCE\_LT commodity supply lower than or equal demand

STOCKS\_BALANCE commodity inter-temporal balance of stocks

CAPACITY\_CONSTRAINT capacity constraint for technology (by sub-annual time slice)

CAPACITY\_MAINTENANCE\_HIST constraint for capacity maintenance historical installation (built before start of model horizon)

CAPACITY\_MAINTENANCE\_NEW constraint for capacity maintenance of new capacity built in the current period (vintage == year)

CAPACITY\_MAINTENANCE constraint for capacity maintenance over the technical lifetime

END\_OF\_LIFETIME\_CAPACITY constraint to keep the capacity zero after the lifetime of a technology

OPERATION\_CONSTRAINT constraint on maximum yearly operation (scheduled down-time for maintenance)

MIN\_UTILIZATION\_CONSTRAINT constraint for minimum yearly operation (aggregated over the course of a year)

RENEWABLES\_POTENTIAL\_CONSTRAINT constraint on renewable resource potential

RENEWABLES\_CAPACITY\_REQUIREMENT lower bound on required overcapacity when using lower grade potentials

RENEWABLES\_EQUIVALENCE equation to define the renewables extraction

ADDON\_ACTIVITY\_UP addon-technology activity upper constraint

ADDON\_ACTIVITY\_LO addon technology activity lower constraint

COMMODITY\_USE\_LEVEL aggregate use of commodity by level as defined by total input into technologies

ACTIVITY\_BY\_RATING constraint on auxiliary rating-specific activity variable by rating bin

ACTIVITY\_RATING\_TOTAL equivalence of auxiliary rating-specific activity variables to actual activity

FIRM\_CAPACITY\_PROVISION contribution of dispatchable technologies to auxiliary firm-capacity variable

SYSTEM\_RELIABILITY\_CONSTRAINT constraint on total system reliability (firm capacity)

SYSTEM\_FLEXIBILITY\_CONSTRAINT constraint on total system flexibility

NEW\_CAPACITY\_BOUND\_UP upper bound on technology capacity investment

NEW\_CAPACITY\_BOUND\_LO lower bound on technology capacity investment

TOTAL\_CAPACITY\_BOUND\_UP upper bound on total installed capacity

TOTAL\_CAPACITY\_BOUND\_LO lower bound on total installed capacity

NEW\_CAPACITY\_CONSTRAINT\_UP dynamic constraint for capacity investment (learning and spillovers upper bound)

NEW\_CAPACITY\_SOFT\_CONSTRAINT\_UP bound on soft relaxation of dynamic new capacity constraints (upwards)

NEW\_CAPACITY\_CONSTRAINT\_LO dynamic constraint on capacity investment (lower bound)

NEW\_CAPACITY\_SOFT\_CONSTRAINT\_LO bound on soft relaxation of dynamic new capacity constraints (downwards)

ACTIVITY\_BOUND\_UP upper bound on activity summed over all vintages

ACTIVITY\_BOUND\_LO lower bound on activity summed over all vintages

ACTIVITY\_BOUND\_ALL\_MODES\_UP upper bound on activity summed over all vintages and modes

ACTIVITY\_BOUND\_ALL\_MODES\_LO lower bound on activity summed over all vintages and modes

SHARE\_CONSTRAINT\_COMMODITY\_UP upper bounds on share constraints for commodities

SHARE\_CONSTRAINT\_COMMODITY\_LO lower bounds on share constraints for commodities

SHARE\_CONSTRAINT\_MODE\_UP upper bounds on share constraints for modes of a given technology

SHARE\_CONSTRAINT\_MODE\_LO lower bounds on share constraints for modes of a given technology

ACTIVITY\_CONSTRAINT\_UP dynamic constraint on the market penetration of a technology activity (upper bound)

ACTIVITY\_SOFT\_CONSTRAINT\_UP bound on relaxation of the dynamic constraint on market penetration (upper bound)

ACTIVITY\_CONSTRAINT\_LO dynamic constraint on the market penetration of a technology activity (lower bound)

ACTIVITY\_SOFT\_CONSTRAINT\_LO bound on relaxation of the dynamic constraint on market penetration (lower bound)

EMISSION\_EQUIVALENCE auxiliary equation to simplify the notation of emissions

EMISSION\_CONSTRAINT nodal-regional-global constraints on emissions (by category)

LAND\_CONSTRAINT constraint on total land use (linear combination of land scenarios adds up to 1)

DYNAMIC\_LAND\_SCEN\_CONSTRAINT\_UP dynamic constraint on land scenario change (upper bound)

DYNAMIC\_LAND\_SCEN\_CONSTRAINT\_LO dynamic constraint on land scenario change (lower bound)

DYNAMIC\_LAND\_TYPE\_CONSTRAINT\_UP dynamic constraint on land-use change (upper bound)

DYNAMIC\_LAND\_TYPE\_CONSTRAINT\_LO dynamic constraint on land-use change (lower bound)

RELATION\_EQUIVALENCE auxiliary equation to simplify the implementation of relations

RELATION\_CONSTRAINT\_UP upper bound of relations (linear constraints)

RELATION\_CONSTRAINT\_LO lower bound of relations (linear constraints)

STORAGE\_CHANGE change in the state of charge of storage

STORAGE\_BALANCE balance of the state of charge of storage

STORAGE\_BALANCE\_INIT balance of the state of charge of storage at sub-annual time slices with initial storage content

STORAGE\_INPUT connecting an input commodity to maintain the activity of storage container (not stored commodity)

;

\*----------------------------------------------------------------------------------------------------------------------\*

\* equation statements \*

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_section\_objective:

\*

\* Objective function

\* ------------------

\*

\* The objective function of the |MESSAGEix| core model

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* .. \_equation\_objective:

\*

\* Equation OBJECTIVE

\* """"""""""""""""""

\*

\* The objective function (of the core model) minimizes total discounted systems costs including costs for emissions,

\* relaxations of dynamic constraints

\*

\* .. math::

\* \text{OBJ} = \sum\_{n,y \in Y^{M}} \text{df\_period}\_{y} \cdot \text{COST\_NODAL}\_{n,y}

\*

\*\*\*

OBJECTIVE..

OBJ =E= SUM( (node,year), df\_period(year) \* COST\_NODAL(node,year) ) ;

\*\*\*

\* Regional system cost accounting function

\* ----------------------------------------

\*

\* Accounting of regional system costs over time

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* .. \_equation\_cost\_accounting\_nodal:

\*

\* Equation COST\_ACCOUNTING\_NODAL

\* """"""""""""""""""""""""""""""

\*

\* Accounting of regional systems costs over time as well as costs for emissions (taxes),

\* land use (from the model land-use model emulator), relaxations of dynamic constraints,

\* and linear relations.

\*

\* .. math::

\* \text{COST\_NODAL}\_{n,y} & = \sum\_{c,g} \ \text{resource\_cost}\_{n,c,g,y} \cdot \text{EXT}\_{n,c,g,y} \\

\* & + \sum\_{t} \

\* \bigg( \text{inv\_cost}\_{n,t,y} \cdot \text{construction\_time\_factor}\_{n,t,y} \\

\* & \quad \quad \quad \cdot \text{end\_of\_horizon\_factor}\_{n,t,y} \cdot \text{CAP\_NEW}\_{n,t,y} \\[4 pt]

\* & \quad \quad + \sum\_{y^V \leq y} \ \text{fix\_cost}\_{n,t,y^V,y} \cdot \text{CAP}\_{n,t,y^V,y} \\

\* & \quad \quad + \sum\_{\substack{y^V \leq y \\ m,h}} \ \text{var\_cost}\_{n,t,y^V,y,m,h} \cdot \text{ACT}\_{n,t,y^V,y,m,h} \\

\* & \quad \quad + \Big( \text{abs\_cost\_new\_capacity\_soft\_up}\_{n,t,y} \\

\* & \quad \quad \quad

\* + \text{level\_cost\_new\_capacity\_soft\_up}\_{n,t,y} \cdot\ \text{inv\_cost}\_{n,t,y}

\* \Big) \cdot \text{CAP\_NEW\_UP}\_{n,t,y} \\[4pt]

\* & \quad \quad + \Big( \text{abs\_cost\_new\_capacity\_soft\_lo}\_{n,t,y} \\

\* & \quad \quad \quad

\* + \text{level\_cost\_new\_capacity\_soft\_lo}\_{n,t,y} \cdot\ \text{inv\_cost}\_{n,t,y}

\* \Big) \cdot \text{CAP\_NEW\_LO}\_{n,t,y} \\[4pt]

\* & \quad \quad + \sum\_{m,h} \ \Big( \text{abs\_cost\_activity\_soft\_up}\_{n,t,y,m,h} \\

\* & \quad \quad \quad

\* + \text{level\_cost\_activity\_soft\_up}\_{n,t,y,m,h} \cdot\ \text{levelized\_cost}\_{n,t,y,m,h}

\* \Big) \cdot \text{ACT\_UP}\_{n,t,y,h} \\

\* & \quad \quad + \sum\_{m,h} \ \Big( \text{abs\_cost\_activity\_soft\_lo}\_{n,t,y,m,h} \\

\* & \quad \quad \quad

\* + \text{level\_cost\_activity\_soft\_lo}\_{n,t,y,m,h} \cdot\ \text{levelized\_cost}\_{n,t,y,m,h}

\* \Big) \cdot \text{ACT\_LO}\_{n,t,y,h} \bigg) \\

\* & + \sum\_{\substack{\widehat{e},\widehat{t} \\ e \in E(\widehat{e})}}

\* \text{emission\_scaling}\_{\widehat{e},e} \cdot \ \text{emission\_tax}\_{n,\widehat{e},\widehat{t},y}

\* \cdot \text{EMISS}\_{n,e,\widehat{t},y} \\

\* & + \sum\_{s} \text{land\_cost}\_{n,s,y} \cdot \text{LAND}\_{n,s,y} \\

\* & + \sum\_{r} \text{relation\_cost}\_{r,n,y} \cdot \text{REL}\_{r,n,y}

\*\*\*

COST\_ACCOUNTING\_NODAL(node, year)..

COST\_NODAL(node, year) =E=

\* resource extraction costs

SUM((commodity,grade)$( map\_resource(node,commodity,grade,year) ),

resource\_cost(node,commodity,grade,year) \* EXT(node,commodity,grade,year) )

\* technology capacity investment, maintainance, operational cost

+ SUM((tec)$( map\_tec(node,tec,year) ),

( inv\_cost(node,tec,year) \* construction\_time\_factor(node,tec,year)

\* end\_of\_horizon\_factor(node,tec,year) \* CAP\_NEW(node,tec,year)

+ SUM(vintage$( map\_tec\_lifetime(node,tec,vintage,year) ),

fix\_cost(node,tec,vintage,year) \* CAP(node,tec,vintage,year) ) )$( inv\_tec(tec) )

+ SUM((vintage,mode,time)$( map\_tec\_lifetime(node,tec,vintage,year) AND map\_tec\_act(node,tec,year,mode,time) ),

var\_cost(node,tec,vintage,year,mode,time) \* ACT(node,tec,vintage,year,mode,time) )

)

\* additional cost terms (penalty) for relaxation of 'soft' dynamic new capacity constraints

+ SUM((inv\_tec)$( map\_tec(node,inv\_tec,year) ),

SUM((mode,time)$map\_tec\_act(node,inv\_tec,year,mode,time),

( ( abs\_cost\_new\_capacity\_soft\_up(node,inv\_tec,year)

+ level\_cost\_new\_capacity\_soft\_up(node,inv\_tec,year) \* inv\_cost(node,inv\_tec,year)

) \* CAP\_NEW\_UP(node,inv\_tec,year) )$( soft\_new\_capacity\_up(node,inv\_tec,year) )

+ ( ( abs\_cost\_new\_capacity\_soft\_lo(node,inv\_tec,year)

+ level\_cost\_new\_capacity\_soft\_lo(node,inv\_tec,year) \* inv\_cost(node,inv\_tec,year)

) \* CAP\_NEW\_LO(node,inv\_tec,year) )$( soft\_new\_capacity\_lo(node,inv\_tec,year) )

)

)

\* additional cost terms (penalty) for relaxation of 'soft' dynamic activity constraints

+ SUM((tec)$( map\_tec(node,tec,year) ),

SUM(time$( map\_tec\_time(node,tec,year,time) ),

( ( abs\_cost\_activity\_soft\_up(node,tec,year,time)

+ level\_cost\_activity\_soft\_up(node,tec,year,time) \* levelized\_cost(node,tec,year,time)

) \* ACT\_UP(node,tec,year,time) )$( soft\_activity\_up(node,tec,year,time) )

+ ( ( abs\_cost\_activity\_soft\_lo(node,tec,year,time)

+ level\_cost\_activity\_soft\_lo(node,tec,year,time) \* levelized\_cost(node,tec,year,time)

) \* ACT\_LO(node,tec,year,time) )$( soft\_activity\_lo(node,tec,year,time) )

)

)

\* emission taxes (by parent node, type of technology, type of year and type of emission)

+ SUM((type\_emission,emission,type\_tec,type\_year)$( emission\_scaling(type\_emission,emission)

AND cat\_year(type\_year,year) ),

emission\_scaling(type\_emission,emission)

\* tax\_emission(node,type\_emission,type\_tec,type\_year)

\* EMISS(node,emission,type\_tec,year) )

\* cost terms from land-use model emulator (only includes valid node-land\_scenario-year combinations)

+ SUM(land\_scenario$( land\_cost(node,land\_scenario,year) ),

land\_cost(node,land\_scenario,year) \* LAND(node,land\_scenario,year) )

\* cost terms associated with linear relations

+ SUM(relation$( relation\_cost(relation,node,year) ),

relation\_cost(relation,node,year) \* REL(relation,node,year) )

\* implementation of slack variables for constraints to aid in debugging

+ SUM((commodity,level,time)$( map\_commodity(node,commodity,level,year,time) ), ( 0

%SLACK\_COMMODITY\_EQUIVALENCE% + SLACK\_COMMODITY\_EQUIVALENCE\_UP(node,commodity,level,year,time)

%SLACK\_COMMODITY\_EQUIVALENCE% + SLACK\_COMMODITY\_EQUIVALENCE\_LO(node,commodity,level,year,time)

) \* 1e6 )

+ SUM((tec)$( map\_tec(node,tec,year) ), ( 0

%SLACK\_CAP\_NEW\_BOUND\_UP% + 10 \* SLACK\_CAP\_NEW\_BOUND\_UP(node,tec,year)

%SLACK\_CAP\_NEW\_BOUND\_LO% + 10 \* SLACK\_CAP\_NEW\_BOUND\_LO(node,tec,year)

%SLACK\_CAP\_NEW\_DYNAMIC\_UP% + 10 \* SLACK\_CAP\_NEW\_DYNAMIC\_UP(node,tec,year)

%SLACK\_CAP\_NEW\_DYNAMIC\_LO% + 10 \* SLACK\_CAP\_NEW\_DYNAMIC\_LO(node,tec,year)

%SLACK\_CAP\_TOTAL\_BOUND\_UP% + 10 \* SLACK\_CAP\_TOTAL\_BOUND\_UP(node,tec,year)

%SLACK\_CAP\_TOTAL\_BOUND\_LO% + 10 \* SLACK\_CAP\_TOTAL\_BOUND\_LO(node,tec,year)

) \* ABS( 1000 + inv\_cost(node,tec,year) ) )

+ SUM((tec,time)$( map\_tec\_time(node,tec,year,time) ), ( 0

%SLACK\_ACT\_BOUND\_UP% + 10 \* SUM(mode$( map\_tec\_act(node,tec,year,mode,time) ), SLACK\_ACT\_BOUND\_UP(node,tec,year,mode,time) )

%SLACK\_ACT\_BOUND\_LO% + 10 \* SUM(mode$( map\_tec\_act(node,tec,year,mode,time) ), SLACK\_ACT\_BOUND\_LO(node,tec,year,mode,time) )

%SLACK\_ACT\_DYNAMIC\_UP% + 10 \* SLACK\_ACT\_DYNAMIC\_UP(node,tec,year,time)

%SLACK\_ACT\_DYNAMIC\_LO% + 10 \* SLACK\_ACT\_DYNAMIC\_LO(node,tec,year,time)

) \* ( 1e8

+ ABS( SUM(mode$map\_tec\_act(node,tec,year,mode,time), var\_cost(node,tec,year,year,mode,time) ) )

+ fix\_cost(node,tec,year,year) ) )

+ SUM(land\_scenario, 0

%SLACK\_LAND\_SCEN\_UP% + 1e6 \* SLACK\_LAND\_SCEN\_UP(node,land\_scenario,year)

%SLACK\_LAND\_SCEN\_LO% + 1e6 \* SLACK\_LAND\_SCEN\_LO(node,land\_scenario,year)

)

+ SUM(land\_type, 0

%SLACK\_LAND\_TYPE\_UP% + 1e6 \* SLACK\_LAND\_TYPE\_UP(node,year,land\_type)

%SLACK\_LAND\_TYPE\_LO% + 1e6 \* SLACK\_LAND\_TYPE\_LO(node,year,land\_type)

)

+ SUM((relation), 0

%SLACK\_RELATION\_BOUND\_UP% + 1e6 \* SLACK\_RELATION\_BOUND\_UP(relation,node,year)$( is\_relation\_upper(relation,node,year) )

%SLACK\_RELATION\_BOUND\_LO% + 1e6 \* SLACK\_RELATION\_BOUND\_LO(relation,node,year)$( is\_relation\_lower(relation,node,year) )

)

;

\*\*\*

\* Here, :math:`n^L \in N(n)` are all nodes :math:`n^L` that are sub-nodes of node :math:`n`.

\* The subset of technologies :math:`t \in T(\widehat{t})` are all tecs that belong to category :math:`\widehat{t}`,

\* and similar notation is used for emissions :math:`e \in E`.

\*\*\*

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_section\_resource\_commodity:

\*

\* Resource and commodity section

\* ------------------------------

\*

\* Constraints on resource extraction

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* .. \_extraction\_equivalence:

\*

\* Equation EXTRACTION\_EQUIVALENCE

\* """""""""""""""""""""""""""""""

\*

\* This constraint translates the quantity of resources extracted (summed over all grades) to the input used by

\* all technologies (drawing from that node). It is introduced to simplify subsequent notation in input/output relations

\* and nodal balance constraints.

\*

\* .. math::

\* \sum\_{g} \text{EXT}\_{n,c,g,y} =

\* \sum\_{\substack{n^L,t,m,h,h^{\text{OD}} \\ y^V \leq y \\ \ l \in L^{\text{RES}} \subseteq L }}

\* \text{input}\_{n^L,t,y^V,y,m,n,c,l,h,h^{\text{OD}}} \cdot \text{ACT}\_{n^L,t,m,y,h}

\*

\* The set :math:`L^{\text{RES}} \subseteq L` denotes all levels for which the detailed representation of resources applies.

\*\*\*

EXTRACTION\_EQUIVALENCE(node,commodity,year)..

SUM(grade$( map\_resource(node,commodity,grade,year) ), EXT(node,commodity,grade,year) )

=G= SUM((location,tec,vintage,mode,level\_resource,time\_act,time\_od)$( map\_tec\_act(node,tec,year,mode,time\_act)

AND map\_tec\_lifetime(node,tec,vintage,year) ),

input(location,tec,vintage,year,mode,node,commodity,level\_resource,time\_act,time\_od)

\* ACT(location,tec,vintage,year,mode,time\_act) ) ;

\*\*\*

\* .. \_equation\_extraction\_bound\_up:

\*

\* Equation EXTRACTION\_BOUND\_UP

\* """"""""""""""""""""""""""""

\*

\* This constraint specifies an upper bound on resource extraction by grade.

\*

\* .. math::

\* \text{EXT}\_{n,c,g,y} \leq \text{bound\_extraction\_up}\_{n,c,g,y}

\*

\*\*\*

EXTRACTION\_BOUND\_UP(node,commodity,grade,year)$( map\_resource(node,commodity,grade,year)

AND is\_bound\_extraction\_up(node,commodity,grade,year) )..

EXT(node,commodity,grade,year) =L= bound\_extraction\_up(node,commodity,grade,year) ;

\*\*\*

\* .. \_equation\_resource\_constraint:

\*

\* Equation RESOURCE\_CONSTRAINT

\* """"""""""""""""""""""""""""

\*

\* This constraint restricts that resource extraction in a year guarantees the "remaining resources" constraint,

\* i.e., only a given fraction of remaining resources can be extracted per year.

\*

\* .. math::

\* \text{EXT}\_{n,c,g,y} \leq

\* \text{resource\_remaining}\_{n,c,g,y} \cdot

\* \Big( & \text{resource\_volume}\_{n,c,g} \\

\* & - \sum\_{y' < y} \text{duration\_period}\_{y'} \cdot \text{EXT}\_{n,c,g,y'} \Big)

\*

\*\*\*

RESOURCE\_CONSTRAINT(node,commodity,grade,year)$( map\_resource(node,commodity,grade,year)

AND resource\_remaining(node,commodity,grade,year) )..

\* extraction per year

EXT(node,commodity,grade,year) =L=

\* remaining resources multiplied by remaining-resources-factor

resource\_remaining(node,commodity,grade,year)

\* ( resource\_volume(node,commodity,grade)

- SUM(year2$( year\_order(year2) < year\_order(year) ),

duration\_period(year2) \* EXT(node,commodity,grade,year2) ) ) ;

\*\*\*

\* .. \_equation\_resource\_horizon:

\*

\* Equation RESOURCE\_HORIZON

\* """""""""""""""""""""""""

\* This constraint ensures that total resource extraction over the model horizon does not exceed the available resources.

\*

\* .. math::

\* \sum\_{y} \text{duration\_period}\_{y} \cdot \text{EXT}\_{n,c,g,y} \leq \text{resource\_volume}\_{n,c,g}

\*

\*\*\*

RESOURCE\_HORIZON(node,commodity,grade)$( SUM(year$map\_resource(node,commodity,grade,year), 1 ) )..

SUM(year, duration\_period(year) \* EXT(node,commodity,grade,year) ) =L= resource\_volume(node,commodity,grade) ;

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* Constraints on commodities and stocks

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* .. \_equation\_commodity\_balance\_aux:

\*

\* Equation COMMODITY\_BALANCE\_AUX

\* """"""""""""""""""""""""""""""

\*

\* This equation sets the value of variable :math:`\COMMODITYBALANCE`,

\* which is then constrained by :ref:`commodity\_balance\_gt` and :ref:`commodity\_balance\_lt`.

\*

\* For each :math:`(n, c, l, y, h)`, this variable includes:

\*

\* - Net |input| minus |output| of commodities based on technology activity (|ACT|).

\* - Net |land\_input| minus |land\_output| of commodities based on |LAND|.

\* - Inter-period transfers via |STOCK\_CHG|.

\* - If the :class:`.MESSAGE` option :py:`cap\_comm=True` is given,

\* flows of commodities (e.g. ‘materials’)

\* associated with construction and retirement of technology capacity (|CAP|).

\* For |y0|, this representation requires |historical\_new\_capacity| parameter values,

\* and depends on conditions for |remaining\_capacity|, |duration\_period|, and |technical\_lifetime|

\* to determine when a technology is retired and account for the relevant flows.

\*

\* .. math::

\* \sum\_{\substack{n^L,t,m,h^A \\ y^V \leq y}} output\_{n^L,t,y^V,y,m,n,c,l,h^A,h}

\* \cdot duration\\_time\\_rel\_{h,h^A} \cdot ACT\_{n^L,t,y^V,y,m,h^A} & \\

\* - \sum\_{\substack{n^L,t,m,h^A \\ y^V \leq y}} input\_{n^L,t,y^V,y,m,n,c,l,h^A,h}

\* \cdot duration\\_time\\_rel\_{h,h^A} \cdot ACT\_{n^L,t,m,y,h^A} & \\

\* + \sum\_{\substack{n^L,t}} output\\_cap\\_new\_{n^L,t,y,n,c,l,h}

\* \cdot CAP\\_NEW\_{n^L,t,y} & \\

\* - \sum\_{\substack{n^L,t}} input\\_cap\\_new\_{n^L,t,y,n,c,l,h}

\* \cdot CAP\\_NEW\_{n^L,t,y} & \\

\* + \sum\_{\substack{n^L,t \\ y^V \leq y}} output\\_cap\\_ret\_{n^L,t,y^V,n,c,l,h}

\* \cdot ( CAP\_{n^L,t,y^V,y-1} - CAP\_{n^L,t,y^V,y} ) \cdot \frac{1}{duration\\_period\_y} & \\

\* - \sum\_{\substack{n^L,t \\ y^V \leq y}} input\\_cap\\_ret\_{n^L,t,y^V,n,c,l,h}

\* \cdot ( CAP\_{n^L,t,y^V,y-1} - CAP\_{n^L,t,y^V,y} ) \cdot \frac{1}{duration\\_period\_y} & \\

\* + \sum\_{\substack{n^L,t \\ y^V \leq y}} output\\_cap\_{n^L,t,y^V,y,n,c,l,h}

\* \cdot CAP\_{n^L,t,y^V,y} & \\

\* - \sum\_{\substack{n^L,t \\ y^V \leq y}} input\\_cap\_{n^L,t,y^V,y,n,c,l,h}

\* \cdot CAP\_{n^L,t,y^V,y} & \\

\* + \ STOCK\\_CHG\_{n,c,l,y,h} + \ \sum\_s \Big( land\\_output\_{n,s,y,c,l,h} - land\\_input\_{n,s,y,c,l,h} \Big) \cdot & LAND\_{n,s,y} \\[4pt]

\* - \ demand\\_fixed\_{n,c,l,y,h}

\* = \COMMODITYBALANCE\_{n,c,l,y,h} \quad \forall \ l \notin (L^{RES}, & L^{REN}, L^{STOR} \subseteq L)

\*

\* The commodity balance constraint at the resource level is included in the `Equation RESOURCE\_CONSTRAINT`\_,

\* while at the renewable level, it is included in the `Equation RENEWABLES\_EQUIVALENCE`\_,

\* and at the storage level, it is included in the `Equation STORAGE\_BALANCE`\_.

\*\*\*

COMMODITY\_BALANCE\_AUX(node,commodity,level,year,time)$(

map\_commodity(node,commodity,level,year,time)

AND NOT (level\_renewable(level) OR level\_resource(level))

) ..

COMMODITY\_BALANCE(node,commodity,level,year,time)

=E=

SUM(

(location,tec,vintage,mode,time2)$(

map\_tec\_act(location,tec,year,mode,time2)

AND map\_tec\_lifetime(location,tec,vintage,year)

),

# Import into node and output from technologies located at 'location' sending to 'node' and 'time2' sending to 'time

# Export from node and input into technologies located at 'location' taking from 'node' and 'time2' taking from 'time'

(

output(location,tec,vintage,year,mode,node,commodity,level,time2,time)

- input(location,tec,vintage,year,mode,node,commodity,level,time2,time)

) \* duration\_time\_rel(time,time2)

\* ACT(location,tec,vintage,year,mode,time2)

)

# Quantity taken out from ( >0 ) or put into ( <0 ) inter-period stock (storage)

+ STOCK\_CHG(node,commodity,level,year,time)$map\_stocks(node,commodity,level,year)

# Yield from land-use model emulator

+ SUM(

land\_scenario,

(

land\_output(node,land\_scenario,year,commodity,level,time)

- land\_input(node,land\_scenario,year,commodity,level,time)

) \* LAND(node,land\_scenario,year)

)

# Final demand (exogenous parameter to be satisfied by the commodity system)

- demand\_fixed(node,commodity,level,year,time)

$IFTHEN %MESSAGE\_CAP\_COMM% == "1"

# Commodity flows associated with CAP and CAP\_NEW

#

# This section contains 5 SUM()s that are included only if MESSAGE\_CAP\_COMM is set to "1".

# (1) CAP and {in,out}put\_cap: flows due to operation of existing capacity

+ SUM(

(location,tec,vintage)$(inv\_tec(tec) AND map\_tec\_lifetime(location,tec,vintage,year)),

(

output\_cap(location,tec,vintage,year,node,commodity,level,time)

- input\_cap(location,tec,vintage,year,node,commodity,level,time)

) \* CAP(location,tec,vintage,year)

)

# (2) CAP\_NEW and {in,out}put\_cap\_new: flows due to construction of new capacity (during vintage period)

+ SUM(

(location,tec)$(inv\_tec(tec) AND map\_tec(location,tec,year)),

(

output\_cap\_new(location,tec,year,node,commodity,level,time)

- input\_cap\_new(location,tec,year,node,commodity,level,time)

) \* CAP\_NEW(location,tec,year)

)

# (3) CAP and {in,out}put\_cap\_ret: flows due to retirement of CAP (any model period after the first)

+ SUM(

(location,tec,vintage,year2)$map\_cap\_ret(location,tec,vintage,year2,year),

(

output\_cap\_ret(location,tec,vintage,node,commodity,level,time)

- input\_cap\_ret(location,tec,vintage,node,commodity,level,time)

) \* (

# Differential of capacity in year2 vs. year

CAP(location,tec,vintage,year2) - CAP(location,tec,vintage,year)

) / duration\_period(year)

)

# (4) historical\_new\_capacity and {in,out}put\_cap\_ret (1 of 2)

#

# Flows due to CAP(…,vintage,…) that reaches EOL in the final pre-horizon period, 'year\_all2'.

# These are counted in the first model period, 'year'.

+ SUM(

(location,tec,vintage,year\_all2)$map\_cap\_ret\_hist\_1(location,tec,vintage,year\_all2,year),

(

output\_cap\_ret(location,tec,vintage,node,commodity,level,time)

- input\_cap\_ret(location,tec,vintage,node,commodity,level,time)

) \* historical\_new\_capacity(node,tec,vintage)

\* remaining\_capacity\_extended(node,tec,vintage,year\_all2)

)

# (5) historical\_new\_capacity and {in,out}put\_cap\_ret (2 of 2)

#

# Flows due to CAP(…,vintage,…) that reaches EOL in the first model period, 'year'.

+ SUM(

(location,tec,vintage,year\_all2)$map\_cap\_ret\_hist\_2(location,tec,vintage,year\_all2,year),

(

output\_cap\_ret(location,tec,vintage,node,commodity,level,time)

- input\_cap\_ret(location,tec,vintage,node,commodity,level,time)

) \* historical\_new\_capacity(node,tec,vintage)

\* (1 - remaining\_capacity\_extended(node,tec,vintage,year))

)

$ENDIF

;

\*\*\*

\* .. \_commodity\_balance\_gt:

\*

\* Equation COMMODITY\_BALANCE\_GT

\* """""""""""""""""""""""""""""

\* This constraint ensures that supply is greater or equal than demand for every commodity-level combination.

\*

\* .. math::

\* \COMMODITYBALANCE\_{n,c,l,y,h} \geq 0

\*

\*\*\*

COMMODITY\_BALANCE\_GT(node,commodity,level,year,time)$(

map\_commodity(node,commodity,level,year,time)

AND NOT (level\_resource(level) OR level\_renewable(level) OR level\_storage(level))

)..

COMMODITY\_BALANCE(node,commodity,level,year,time)

\* relaxation of constraints for debugging

%SLACK\_COMMODITY\_EQUIVALENCE% + SLACK\_COMMODITY\_EQUIVALENCE\_UP(node,commodity,level,year,time)

=G=

0

;

\*\*\*

\* .. \_commodity\_balance\_lt:

\*

\* Equation COMMODITY\_BALANCE\_LT

\* """""""""""""""""""""""""""""

\* This constraint ensures that the supply is smaller than or equal to the demand for all commodity-level combinations

\* given in the :math:`\text{balance\_equality}\_{c,l}`. In combination with the constraint above, it ensures that supply

\* is (exactly) equal to demand.

\*

\* .. math::

\* \COMMODITYBALANCE\_{n,c,l,y,h} \leq 0

\*

\*\*\*

COMMODITY\_BALANCE\_LT(node,commodity,level,year,time)$(

map\_commodity(node,commodity,level,year,time)

AND NOT (level\_resource(level) OR level\_renewable(level) OR level\_storage(level))

AND balance\_equality(commodity,level)

)..

COMMODITY\_BALANCE(node,commodity,level,year,time)

\* relaxation of constraints for debugging

%SLACK\_COMMODITY\_EQUIVALENCE% - SLACK\_COMMODITY\_EQUIVALENCE\_LO(node,commodity,level,year,time)

=L=

0

;

\*\*\*

\* .. equation\_stock\_balance:

\*

\* Equation STOCKS\_BALANCE

\* """""""""""""""""""""""

\* This constraint ensures the inter-temporal balance of commodity stocks.

\* The parameter :math:`\text{commodity\_stocks}\_{n,c,l}` can be used to model exogenous additions to the stock

\*

\* .. math::

\* \text{STOCK}\_{n,c,l,y} + \text{commodity\_stock}\_{n,c,l,y} =

\* \text{duration\_period}\_{y} \cdot & \sum\_{h} \text{STOCK\_CHG}\_{n,c,l,y,h} \\

\* & + \text{STOCK}\_{n,c,l,y+1}

\*

\*\*\*

STOCKS\_BALANCE(node,commodity,level,year)$( map\_stocks(node,commodity,level,year) )..

STOCK(node,commodity,level,year)$( NOT first\_period(year) )

+ commodity\_stock(node,commodity,level,year) =E=

duration\_period(year) \* SUM(time$( map\_commodity(node,commodity,level,year,time) ),

STOCK\_CHG(node,commodity,level,year,time) )

+ SUM(year2$( seq\_period(year,year2) ), STOCK(node,commodity,level,year2) ) ;

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_section\_technology:

\*

\* Technology section

\* ------------------

\*

\* Technical and engineering constraints

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\* The first set of constraints concern technologies that have explicit investment decisions

\* and where installed/maintained capacity is relevant for operational decisions.

\* The set where :math:`T^{\text{INV}} \subseteq T` is the set of all these technologies.

\*

\* .. \_equation\_capacity\_constraint:

\*

\* Equation CAPACITY\_CONSTRAINT

\* """"""""""""""""""""""""""""

\* This constraint ensures that the actual activity of a technology at a node cannot exceed available (maintained)

\* capacity summed over all vintages, including the technology capacity factor :math:`\text{capacity\_factor}\_{n,t,y,t}`.

\*

\* .. math::

\* \sum\_{m} \text{ACT}\_{n,t,y^V,y,m,h}

\* \leq \text{duration\_time}\_{h} \cdot \text{capacity\_factor}\_{n,t,y^V,y,h} \cdot \text{CAP}\_{n,t,y^V,y}

\* \quad \forall \ t \ \in \ T^{\text{INV}}

\*

\*\*\*

CAPACITY\_CONSTRAINT(node,inv\_tec,vintage,year,time)$( map\_tec\_time(node,inv\_tec,year,time)

AND map\_tec\_lifetime(node,inv\_tec,vintage,year) )..

SUM(mode$( map\_tec\_act(node,inv\_tec,year,mode,time) ), ACT(node,inv\_tec,vintage,year,mode,time) )

=L= duration\_time(time) \* capacity\_factor(node,inv\_tec,vintage,year,time) \* CAP(node,inv\_tec,vintage,year) ;

\*\*\*

\* .. \_equation\_capacity\_maintenance\_hist:

\*

\* Equation CAPACITY\_MAINTENANCE\_HIST

\* """"""""""""""""""""""""""""""""""

\* The following three constraints implement technology capacity maintenance over time to allow early retirement.

\* The optimization problem determines the optimal timing of retirement, when fixed operation-and-maintenance costs

\* exceed the benefit in the objective function.

\*

\* The first constraint ensures that historical capacity (built prior to the model horizon) is available

\* as installed capacity in the first model period.

\*

\* .. math::

\* CAP\_{n,t,y^V,'first\\_period'} & \leq

\* remaining\\_capacity\_{n,t,y^V,'first\\_period'} \cdot

\* duration\\_period\_{y^V} \cdot

\* historical\\_new\\_capacity\_{n,t,y^V} \\

\* & \text{if } y^V < first\\_period \text{ and } |y| - |y^V| < technical\\_lifetime\_{n,t,y^V}

\* \quad \forall \ t \in T^{INV}

\*

\*\*\*

CAPACITY\_MAINTENANCE\_HIST(node,inv\_tec,vintage,first\_period)$( map\_tec\_lifetime(node,inv\_tec,vintage,first\_period)

AND historical(vintage))..

CAP(node,inv\_tec,vintage,first\_period)

=L= remaining\_capacity(node,inv\_tec,vintage,first\_period) \*

duration\_period(vintage) \* historical\_new\_capacity(node,inv\_tec,vintage) ;

\*\*\*

\* .. \_equation\_capacity\_maintenance\_new:

\*

\* Equation CAPACITY\_MAINTENANCE\_NEW

\* """""""""""""""""""""""""""""""""

\* The second constraint ensures that capacity is fully maintained throughout the model period

\* in which it was constructed (no early retirement in the period of construction).

\*

\* .. math::

\* \text{CAP}\_{n,t,y^V,y^V} =

\* \text{remaining\_capacity}\_{n,t,y^V,y^V} \cdot

\* \text{duration\_period}\_{y^V} \cdot

\* \text{CAP\_NEW}\_{n,t,y^V}

\* \quad \forall \ t \in T^{\text{INV}}

\*

\* The current formulation does not account for construction time in the constraints, but only adds a mark-up

\* to the investment costs in the objective function.

\*\*\*

CAPACITY\_MAINTENANCE\_NEW(node,inv\_tec,vintage,vintage)$( map\_tec\_lifetime(node,inv\_tec,vintage,vintage) )..

CAP(node,inv\_tec,vintage,vintage)

=E= remaining\_capacity(node,inv\_tec,vintage,vintage)

\* duration\_period(vintage) \* CAP\_NEW(node,inv\_tec,vintage) ;

\*\*\*

\* .. \_equation\_capacity\_maintenance:

\*

\* Equation CAPACITY\_MAINTENANCE

\* """""""""""""""""""""""""""""

\* The third constraint implements the dynamics of capacity maintenance throughout the model horizon.

\* Installed capacity can be maintained over time until decommissioning, which is irreversible.

\*

\* .. math::

\* CAP\_{n,t,y^V,y} & \leq

\* remaining\\_capacity\_{n,t,y^V,y} \cdot

\* CAP\_{n,t,y^V,y-1} \\

\* \quad & \text{if } y > y^V \text{ and } y^V > first\\_period \text{ and } |y| - |y^V| < technical\\_lifetime\_{n,t,y^V}

\* \quad \forall \ t \in T^{INV}

\*

\*\*\*

CAPACITY\_MAINTENANCE(node,inv\_tec,vintage,year)$( map\_tec\_lifetime(node,inv\_tec,vintage,year)

AND NOT first\_period(year) AND year\_order(vintage) < year\_order(year))..

CAP(node,inv\_tec,vintage,year)

=L= remaining\_capacity(node,inv\_tec,vintage,year) \*

( SUM(year2$( seq\_period(year2,year) ),

CAP(node,inv\_tec,vintage,year2) ) ) ;

\*\*\*

\* .. \_equation\_end\_of\_lifetime\_capacity:

\*

\* Equation END\_OF\_LIFETIME\_CAPACITY

\* """""""""""""""""""""""""""""""""

\* This constraint ensures that the capacity is not preserved after the technical lifetime of a technology.

\*

\* .. math::

\* CAP\_{n,t,y^V,y} = 0 \\

\* \quad & \text{if } y > y^V \text{ and } y^V > first\\_period \text{ and } |y| - |y^V| >= technical\\_lifetime\_{n,t,y^V}

\* \quad \forall \ t \in T^{INV}

\*

\*\*\*

END\_OF\_LIFETIME\_CAPACITY(node,inv\_tec,vintage,year)$(

cap\_comm

AND map\_tec(node,inv\_tec,vintage)

AND NOT map\_tec\_lifetime(node,inv\_tec,vintage,year)

AND NOT first\_period(year)

AND year\_order(vintage) < year\_order(year)

) ..

CAP(node,inv\_tec,vintage,year) =L= 0 ;

\*\*\*

\* .. \_equation\_operation\_constraint:

\*

\* Equation OPERATION\_CONSTRAINT

\* """""""""""""""""""""""""""""

\* This constraint provides an upper bound on the total operation of installed capacity over a year.

\* It can be used to represent reuqired scheduled unavailability of installed capacity.

\*

\* .. math::

\* \sum\_{m,h} \text{ACT}\_{n,t,y^V,y,m,h}

\* \leq \text{operation\_factor}\_{n,t,y^V,y} \cdot \text{capacity\_factor}\_{n,t,y^V,y,m,\text{'year'}} \cdot \text{CAP}\_{n,t,y^V,y}

\* \quad \forall \ t \in T^{\text{INV}}

\*

\* This constraint is only active if :math:`\text{operation\_factor}\_{n,t,y^V,y} < 1`.

\*\*\*

OPERATION\_CONSTRAINT(node,inv\_tec,vintage,year)$( map\_tec\_lifetime(node,inv\_tec,vintage,year)

AND operation\_factor(node,inv\_tec,vintage,year) < 1 )..

SUM((mode,time)$( map\_tec\_act(node,inv\_tec,year,mode,time) ), ACT(node,inv\_tec,vintage,year,mode,time) ) =L=

operation\_factor(node,inv\_tec,vintage,year) \* capacity\_factor(node,inv\_tec,vintage,year,'year')

\* CAP(node,inv\_tec,vintage,year) ;

\*\*\*

\* .. \_equation\_min\_utlitation\_constraint:

\*

\* Equation MIN\_UTILIZATION\_CONSTRAINT

\* """""""""""""""""""""""""""""""""""

\* This constraint provides a lower bound on the total utilization of installed capacity over a year.

\*

\* .. math::

\* \sum\_{m,h} \text{ACT}\_{n,t,y^V,y,m,h} \geq \text{min\_utilization\_factor}\_{n,t,y^V,y} \cdot \text{CAP}\_{n,t,y^V,y}

\* \quad \forall \ t \in T^{\text{INV}}

\*

\* This constraint is only active if :math:`\text{min\_utilization\_factor}\_{n,t,y^V,y}` is defined.

\*\*\*

MIN\_UTILIZATION\_CONSTRAINT(node,inv\_tec,vintage,year)$( map\_tec\_lifetime(node,inv\_tec,vintage,year)

AND min\_utilization\_factor(node,inv\_tec,vintage,year) )..

SUM((mode,time)$( map\_tec\_act(node,inv\_tec,year,mode,time) ), ACT(node,inv\_tec,vintage,year,mode,time) ) =G=

min\_utilization\_factor(node,inv\_tec,vintage,year) \* CAP(node,inv\_tec,vintage,year) ;

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_section\_renewable\_integration:

\*

\* Constraints representing renewable integration

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* .. \_equation\_renewables\_equivalence:

\*

\* Equation RENEWABLES\_EQUIVALENCE

\* """""""""""""""""""""""""""""""

\* This constraint defines the auxiliary variables :math:`\text{REN}`

\* to be equal to the output of renewable technologies (summed over grades).

\*

\* .. math::

\* \sum\_{g} \text{REN}\_{n,t,c,g,y,h} \leq

\* \sum\_{\substack{n,t,m,l,h,h^{\text{OD}} \\ y^V \leq y \\ \ l \in L^{\text{REN}} \subseteq L }}

\* \text{input}\_{n^L,t,y^V,y,m,n,c,l,h,h^{\text{OD}}} \cdot \text{ACT}\_{n^L,t,m,y,h}

\*

\* The set :math:`L^{\text{REN}} \subseteq L` denotes all levels for which the detailed representation of renewables applies.

\*\*\*

RENEWABLES\_EQUIVALENCE(node,renewable\_tec,commodity,year,time)$(

map\_tec(node,renewable\_tec,year) AND map\_ren\_com(node,renewable\_tec,commodity,year) )..

SUM(grade$( map\_ren\_grade(node,commodity,grade,year) ), REN(node,renewable\_tec,commodity,grade,year,time) )

=E= SUM((location,vintage,mode,level\_renewable,time\_act)$(

map\_tec\_act(node,renewable\_tec,year,mode,time\_act)

AND map\_tec\_lifetime(node,renewable\_tec,vintage,year) ),

input(location,renewable\_tec,vintage,year,mode,node,commodity,level\_renewable,time\_act,time)

\* ACT(location,renewable\_tec,vintage,year,mode,time\_act) ) ;

\*\*\*

\* .. \_equation\_renewables\_potential\_constraint:

\*

\* Equation RENEWABLES\_POTENTIAL\_CONSTRAINT

\* """"""""""""""""""""""""""""""""""""""""

\* This constraint sets the potential potential by grade as the upper bound for the auxiliary variable :math:`REN`.

\*

\* .. math::

\* \sum\_{\substack{t,h \\ \ t \in T^{R} \subseteq t }} \text{REN}\_{n,t,c,g,y,h}

\* \leq \sum\_{\substack{l \\ l \in L^{R} \subseteq L }} \text{renewable\_potential}\_{n,c,g,l,y}

\*

\*\*\*

RENEWABLES\_POTENTIAL\_CONSTRAINT(node,commodity,grade,year)$( map\_ren\_grade(node,commodity,grade,year) )..

SUM((renewable\_tec,time)$( map\_ren\_com(node,renewable\_tec,commodity,year) ),

REN(node,renewable\_tec,commodity,grade,year,time) )

=L= SUM(level\_renewable, renewable\_potential(node,commodity,grade,level\_renewable,year) ) ;

\*\*\*

\* .. \_equation\_renewables\_capacity\_requirement:

\*

\* Equation RENEWABLES\_CAPACITY\_REQUIREMENT

\* """"""""""""""""""""""""""""""""""""""""

\* This constraint connects the capacity factor of a renewable grade to the

\* installed capacity of a technology. It sets the lower limit for the capacity

\* of a renewable technology to the summed activity over all grades (REN) devided

\* by the capactiy factor of this grade.

\* It represents the fact that different renewable grades require different installed

\* capacities to provide their full potential.

\*

\* .. math::

\* \sum\_{y^V, h} & \text{CAP}\_{n,t,y^V,y} \cdot \text{operation\_factor}\_{n,t,y^V,y} \cdot \text{capacity\_factor}\_{n,t,y^V,y,h} \\

\* & \quad \geq \sum\_{g,h,l} \frac{1}{\text{renewable\_capacity\_factor}\_{n,c,g,l,y}} \cdot \text{REN}\_{n,t,c,g,y,h}

\*

\* This constraint is only active if :math:`\text{renewable\_capacity\_factor}\_{n,c,g,l,y}` is defined.

\*\*\*

RENEWABLES\_CAPACITY\_REQUIREMENT(node,inv\_tec,commodity,year)$(

SUM( (vintage,mode,time,grade,level\_renewable),

map\_tec\_lifetime(node,inv\_tec,vintage,year) AND map\_tec\_act(node,inv\_tec,year,mode,time)

AND map\_ren\_com(node,inv\_tec,commodity,year)

AND renewable\_capacity\_factor(node,commodity,grade,level\_renewable,year) > 0 ) )..

SUM( (vintage,time)$map\_ren\_com(node,inv\_tec,commodity,year),

CAP(node,inv\_tec,vintage,year)

\* operation\_factor(node,inv\_tec,vintage,year)

\* capacity\_factor(node,inv\_tec,vintage,year,time) )

=G= SUM((grade,time,level\_renewable)$(renewable\_capacity\_factor(node,commodity,grade,level\_renewable,year) > 0),

REN(node,inv\_tec,commodity,grade,year,time)

/ renewable\_capacity\_factor(node,commodity,grade,level\_renewable,year)) ;

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_section\_addon\_technologies:

\*

\* Constraints for addon technologies

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* .. \_equation\_addon\_activity\_up:

\*

\* Equation ADDON\_ACTIVITY\_UP

\* """"""""""""""""""""""""""

\* This constraint provides an upper bound on the activity of an addon technology that can only be operated

\* jointly with a parent technology (e.g., abatement option, SO2 scrubber, power plant cooling technology).

\*

\* .. math::

\* \sum\_{\substack{t^a, y^V \leq y}} \text{ACT}\_{n,t^a,y^V,y,m,h}

\* \leq

\* \sum\_{\substack{t, y^V \leq y}}

\* & \text{addon\_up}\_{n,t,y,m,h,\widehat{t^a}} \cdot

\* \text{addon\_conversion}\_{n,t,y^V,y,m,h,\widehat{t^a}} \\

\* & \cdot \text{ACT}\_{n,t,y^V,y,m,h} \quad \forall \ t^a \in T^{A}

\*

\*\*\*

ADDON\_ACTIVITY\_UP(node,type\_addon,year,mode,time)..

\* activity of addon technology

sum( (addon,vintage)$(

cat\_addon(type\_addon,addon) AND

map\_tec\_act(node,addon,year,mode,time) AND

map\_tec\_lifetime(node,addon,vintage,year) ),

ACT(node,addon,vintage,year,mode,time) )

=L=

\* activity of corresponding parent-technology multiplied by upper bound of share

sum((tec,vintage)$(

map\_tec\_addon(tec,type\_addon) AND

map\_tec\_act(node,tec,year,mode,time) AND

map\_tec\_lifetime(node,tec,vintage,year)

),

addon\_up(node,tec,year,mode,time,type\_addon)

\* addon\_conversion(node,tec,vintage,year,mode,time,type\_addon)

\* ACT(node,tec,vintage,year,mode,time) )

;

\*\*\*

\* .. \_equation\_addon\_activity\_lo:

\*

\* Equation ADDON\_ACTIVITY\_LO

\* """"""""""""""""""""""""""

\* This constraint provides a lower bound on the activity of an addon technology that has to be operated

\* jointly with a parent technology (e.g., power plant cooling technology). The parameter `addon\_lo` allows to define

\* a minimum level of operation of addon technologies relative to the activity of the parent technology.

\* If `addon\_lo = 1`, this means that it is mandatory to operate the addon technology at the same level as the

\* parent technology (i.e., full mitigation).

\*

\* .. math::

\* \sum\_{\substack{t^a, y^V \leq y}} \text{ACT}\_{n,t^a,y^V,y,m,h}

\* \geq

\* \sum\_{\substack{t, y^V \leq y}}

\* & \text{addon\_lo}\_{n,t,y,m,h,\widehat{t^a}} \cdot

\* \text{addon\_conversion}\_{n,t,y^V,y,m,h,\widehat{t^a}} \\

\* & \cdot \text{ACT}\_{n,t,y^V,y,m,h} \quad \forall \ t^a \in T^{A}

\*

\*\*\*

ADDON\_ACTIVITY\_LO(node,type\_addon,year,mode,time)..

\* activity of addon technology

sum( (addon,vintage)$(

cat\_addon(type\_addon,addon) AND

map\_tec\_act(node,addon,year,mode,time) AND

map\_tec\_lifetime(node,addon,vintage,year) ),

ACT(node,addon,vintage,year,mode,time) )

=G=

\* activity of corresponding parent-technology times lower bound of share

sum((tec,vintage)$(

map\_tec\_addon(tec,type\_addon) AND

map\_tec\_act(node,tec,year,mode,time) AND

map\_tec\_lifetime(node,tec,vintage,year)

),

addon\_lo(node,tec,year,mode,time,type\_addon)

\* addon\_conversion(node,tec,vintage,year,mode,time,type\_addon)

\* ACT(node,tec,vintage,year,mode,time) ) ;

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_section\_system\_reliability:

\*

\* System reliability and flexibility requirements

\* -----------------------------------------------

\* This section followi allows to include system-wide reliability and flexility considerations.

\* The current formulation is based on Sullivan et al., 2013 :cite:`Sullivan-2013`.

\*

\* Aggregate use of a commodity

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\* The system reliability and flexibility constraints are implemented using an auxiliary variable representing

\* the total use (i.e., input of each commodity per level).

\*

\* .. \_equation\_commodity\_use\_level:

\*

\* Equation COMMODITY\_USE\_LEVEL

\* """"""""""""""""""""""""""""

\* This constraint defines the auxiliary variable :math:`\text{COMMODITY\_USE}\_{n,c,l,y}`, which is used to define

\* the rating bins and the peak-load that needs to be offset with firm (dispatchable) capacity.

\*

\* .. math::

\* \text{COMMODITY\_USE}\_{n,c,l,y}

\* = & \sum\_{n^L,t,y^V,m,h} \text{input}\_{n^L,t,y^V,y,m,n,c,l,h,h} \\

\* & \quad \cdot \text{duration\_time\_rel}\_{h,h} \cdot \text{ACT}\_{n^L,t,y^V,y,m,h}

\*

\* This constraint and the auxiliary variable is only active if :math:`\text{peak\_load\_factor}\_{n,c,l,y,h}` or

\* :math:`\text{flexibility\_factor}\_{n,t,y^V,y,m,c,l,h,r}` is defined.

\*\*\*

COMMODITY\_USE\_LEVEL(node,commodity,level,year,time)$(

peak\_load\_factor(node,commodity,level,year,time) OR

SUM( (tec,vintage,mode,rating), flexibility\_factor(node,tec,vintage,year,mode,commodity,level,time,rating) ) )..

COMMODITY\_USE(node,commodity,level,year)

=E=

SUM( (location,tec,vintage,mode,time2)$( map\_tec\_act(location,tec,year,mode,time2)

AND map\_tec\_lifetime(location,tec,vintage,year) ),

input(location,tec,vintage,year,mode,node,commodity,level,time2,time)

\* duration\_time\_rel(time,time2)

\* ACT(location,tec,vintage,year,mode,time2) ) ;

\*\*\*

\* .. \_rating\_bin:

\*

\* Auxilary variables for technology activity by "rating bins"

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\* The capacity and activity of certain (usually non-dispatchable) technologies

\* can be assumed to only partially contribute to the system reliability and flexibility requirements.

\*

\* .. \_equation\_activity\_rating\_bin:

\*

\* Equation ACTIVITY\_RATING\_BIN

\* """"""""""""""""""""""""""""

\* The auxiliary variable for rating-specific activity of each technology cannot exceed

\* the share of the rating bin in relation to the total commodity use.

\*

\* .. math::

\* \text{ACT\_RATING}\_{n,t,y^V,y,c,l,h,q}

\* \leq \text{rating\_bin}\_{n,t,y,c,l,h,q} \cdot \text{COMMODITY\_USE}\_{n,c,l,y}

\*

\*\*\*

ACTIVITY\_BY\_RATING(node,tec,year,commodity,level,time,rating)$(

rating\_bin(node,tec,year,commodity,level,time,rating) )..

sum(vintage$( sum(mode,map\_tec\_act(node,tec,year,mode,time) ) AND map\_tec\_lifetime(node,tec,vintage,year) ),

ACT\_RATING(node,tec,vintage,year,commodity,level,time,rating) )

=L= rating\_bin(node,tec,year,commodity,level,time,rating) \* COMMODITY\_USE(node,commodity,level,year)

;

\*\*\*

\* .. \_equation\_activity\_share\_total:

\*

\* Equation ACTIVITY\_SHARE\_TOTAL

\* """""""""""""""""""""""""""""

\* The sum of the auxiliary rating-specific activity variables need to equal the total input and/or output

\* of the technology.

\*

\* .. math::

\* \sum\_q \text{ACT\_RATING}\_{n,t,y^V,y,c,l,h,q}

\* = \sum\_{\substack{n^L,t,m,h^A \\ y^V \leq y}} &

\* ( \text{input}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} + \text{output}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} ) \\

\* & \quad \cdot \text{duration\_time\_rel}\_{h,h^A} \cdot \text{ACT}\_{n^L,t,y^V,y,m,h^A} \\

\*

\*\*\*

ACTIVITY\_RATING\_TOTAL(node,tec,vintage,year,commodity,level,time)$(

sum(rating$( rating\_bin(node,tec,year,commodity,level,time,rating) ), 1 )

AND sum(mode, map\_tec\_act(node,tec,year,mode,time))

AND map\_tec\_lifetime(node,tec,vintage,year) )..

sum(rating$( rating\_bin(node,tec,year,commodity,level,time,rating) ),

ACT\_RATING(node,tec,vintage,year,commodity,level,time,rating) )

=E=

SUM((location,mode,time2)$(

map\_tec\_act(location,tec,year,mode,time2)

AND map\_tec\_lifetime(location,tec,vintage,year) ),

( output(location,tec,vintage,year,mode,node,commodity,level,time2,time)

+ input(location,tec,vintage,year,mode,node,commodity,level,time2,time) )

\* duration\_time\_rel(time,time2)

\* ACT(location,tec,vintage,year,mode,time2) ) ;

\*\*\*

\* .. \_reliability\_constraint:

\*

\* Reliability of installed capacity

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\* The "firm capacity" that a technology can contribute to system reliability depends on its dispatch characteristics.

\* For dispatchable technologies, the total installed capacity counts toward the firm capacity constraint.

\* This is active if the parameter is defined over :math:`\text{reliability\_factor}\_{n,t,y,c,l,h,\text{'firm'}}`.

\* For non-dispatchable technologies, or those that do not have explicit investment decisions,

\* the contribution to system reliability is calculated

\* by using the auxiliary variable :math:`\text{ACT\_RATING}\_{n,t,y^V,y,c,l,h,q}` as a proxy,

\* with the :math:`\text{reliability\_factor}\_{n,t,y,c,l,h,q}` defined per rating bin :math:`q`.

\*

\* .. \_equation\_firm\_capacity\_provision:

\*

\* Equation FIRM\_CAPACITY\_PROVISION

\* """"""""""""""""""""""""""""""""

\* Technologies where the reliability factor is defined with the rating `firm`

\* have an auxiliary variable :math:`\text{CAP\_FIRM}\_{n,t,c,l,y}`, defined in terms of output.

\*

\* .. math::

\* \text{CAP\_FIRM}\_{n,t,c,l,y}

\* = \sum\_{y^V \leq y} & \text{output}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} \cdot \text{duration\_time}\_h \\

\* & \quad \cdot \text{capacity\_factor}\_{n,t,y^V,y,h} \cdot \text{CAP}\_{n,t,y^Y,y}

\* \quad \forall \ t \in T^{\text{INV}}

\*

\*\*\*

FIRM\_CAPACITY\_PROVISION(node,inv\_tec,year,commodity,level,time)$(

reliability\_factor(node,inv\_tec,year,commodity,level,time,'firm') )..

CAP\_FIRM(node,inv\_tec,commodity,level,year) =E=

SUM( (location,vintage,mode,time2)$(

map\_tec\_act(location,inv\_tec,year,mode,time2)

AND map\_tec\_lifetime(location,inv\_tec,vintage,year) ),

output(location,inv\_tec,vintage,year,mode,node,commodity,level,time2,time)

\* duration\_time(time)

\* capacity\_factor(node,inv\_tec,vintage,year,time)

\* CAP(node,inv\_tec,vintage,year) ) ;

\*\*\*

\* .. \_equation\_system\_reliability\_constraint:

\*

\* Equation SYSTEM\_RELIABILITY\_CONSTRAINT

\* """"""""""""""""""""""""""""""""""""""

\* This constraint ensures that there is sufficient firm (dispatchable) capacity in each period.

\* The formulation is based on Sullivan et al., 2013 :cite:`Sullivan-2013`.

\*

\* .. math::

\* \sum\_{t, q \substack{t \in T^{\text{INV}} \\ y^V \leq y} } &

\* \text{reliability\_factor}\_{n,t,y,c,l,h,\text{'firm'}}

\* \cdot \text{CAP\_FIRM}\_{n,t,c,l,y} \\

\* + \sum\_{t,q,y^V \leq y} &

\* \text{reliability\_factor}\_{n,t,y,c,l,h,q}

\* \cdot \text{ACT\_RATING}\_{n,t,y^V,y,c,l,h,q} \\

\* & \quad \geq \text{peak\_load\_factor}\_{n,c,l,y,h} \cdot \text{COMMODITY\_USE}\_{n,c,l,y}

\*

\* This constraint is only active if :math:`\text{peak\_load\_factor}\_{n,c,l,y,h}` is defined.

\*\*\*

SYSTEM\_RELIABILITY\_CONSTRAINT(node,commodity,level,year,time)$( peak\_load\_factor(node,commodity,level,year,time) )..

SUM(inv\_tec$( reliability\_factor(node,inv\_tec,year,commodity,level,time,'firm') ),

reliability\_factor(node,inv\_tec,year,commodity,level,time,'firm')

\* CAP\_FIRM(node,inv\_tec,commodity,level,year) )

+ SUM((tec, mode, vintage, rating\_unfirm)$(

reliability\_factor(node,tec,year,commodity,level,time,rating\_unfirm)

AND map\_tec\_act(node,tec,year,mode,time)

AND map\_tec\_lifetime(node,tec,vintage,year) ),

reliability\_factor(node,tec,year,commodity,level,time,rating\_unfirm)

\* ACT\_RATING(node,tec,vintage,year,commodity,level,time,rating\_unfirm) )

=G= peak\_load\_factor(node,commodity,level,year,time) \* COMMODITY\_USE(node,commodity,level,year) ;

\*\*\*

\* .. \_flexibility\_constraint:

\*

\* Equation SYSTEM\_FLEXIBILITY\_CONSTRAINT

\* """"""""""""""""""""""""""""""""""""""

\* This constraint ensures that, in each sub-annual time slice, there is a sufficient

\* contribution from flexible technologies to ensure smooth system operation.

\*

\* .. math::

\* \sum\_{\substack{n^L,t,m,h^A \\ y^V \leq y}} &

\* \text{flexibility\_factor}\_{n^L,t,y^V,y,m,c,l,h,\text{'unrated'}} \\

\* & \quad \cdot ( \text{output}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} + \text{input}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} ) \\

\* & \quad \cdot \text{duration\_time\_rel}\_{h,h^A}

\* \cdot \text{ACT}\_{n,t,y^V,y,m,h} \\

\* + \sum\_{\substack{n^L,t,m,h^A \\ y^V \leq y}} &

\* \text{flexibility\_factor}\_{n^L,t,y^V,y,m,c,l,h,1} \\

\* & \quad \cdot ( \text{output}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} + \text{input}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} ) \\

\* & \quad \cdot \text{duration\_time\_rel}\_{h,h^A}

\* \cdot \text{ACT\_RATING}\_{n,t,y^V,y,c,l,h,q}

\* \geq 0

\*

\*\*\*

SYSTEM\_FLEXIBILITY\_CONSTRAINT(node,commodity,level,year,time)$(

SUM( (tec, vintage, mode, rating),

flexibility\_factor(node,tec,vintage,year,mode,commodity,level,time,rating) ) )..

SUM( (tec, vintage, mode)$( flexibility\_factor(node,tec,vintage,year,mode,commodity,level,time,'unrated') ),

flexibility\_factor(node,tec,vintage,year,mode,commodity,level,time,'unrated')

\* SUM((location,time2)$(

map\_tec\_act(location,tec,year,mode,time2)

AND map\_tec\_lifetime(location,tec,vintage,year) ),

( output(location,tec,vintage,year,mode,node,commodity,level,time2,time)

+ input(location,tec,vintage,year,mode,node,commodity,level,time2,time) )

\* duration\_time\_rel(time,time2)

\* ACT(location,tec,vintage,year,mode,time2) ) )

+ SUM((tec, vintage, mode, rating\_unrated)$(

flexibility\_factor(node,tec,vintage,year,mode,commodity,level,time,rating\_unrated)

AND map\_tec\_act(node,tec,year,mode,time)

AND map\_tec\_lifetime(node,tec,vintage,year)),

flexibility\_factor(node,tec,vintage,year,mode,commodity,level,time,rating\_unrated)

\* ACT\_RATING(node,tec,vintage,year,commodity,level,time,rating\_unrated) )

=G= 0 ;

ACT.LO(node,tec,vintage,year,mode,time)$sum(

(commodity,level,rating), flexibility\_factor(node,tec,vintage,year,mode,commodity,level,time,rating) ) = 0 ;

\*\*\*

\* .. \_section\_bounds\_capacity\_activity:

\*

\* Bounds on capacity and activity

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* .. \_equation\_new\_capacity\_bound\_up:

\*

\* Equation NEW\_CAPACITY\_BOUND\_UP

\* """"""""""""""""""""""""""""""

\* This constraint provides upper bounds on new capacity installation.

\*

\* .. math::

\* \text{CAP\_NEW}\_{n,t,y} \leq \text{bound\_new\_capacity\_up}\_{n,t,y} \quad \forall \ t \ \in \ T^{\text{INV}}

\*

\*\*\*

NEW\_CAPACITY\_BOUND\_UP(node,inv\_tec,year)$( is\_bound\_new\_capacity\_up(node,inv\_tec,year) )..

CAP\_NEW(node,inv\_tec,year) =L= bound\_new\_capacity\_up(node,inv\_tec,year)

%SLACK\_CAP\_NEW\_BOUND\_UP% + SLACK\_CAP\_NEW\_BOUND\_UP(node,inv\_tec,year)

;

\*\*\*

\* .. \_equation\_new\_capacity\_bound\_lo:

\*

\* Equation NEW\_CAPACITY\_BOUND\_LO

\* """"""""""""""""""""""""""""""

\* This constraint provides lower bounds on new capacity installation.

\*

\* .. math::

\* \text{CAP\_NEW}\_{n,t,y} \geq \text{bound\_new\_capacity\_lo}\_{n,t,y} \quad \forall \ t \ \in \ T^{\text{INV}}

\*

\*\*\*

NEW\_CAPACITY\_BOUND\_LO(node,inv\_tec,year)$( is\_bound\_new\_capacity\_lo(node,inv\_tec,year) )..

CAP\_NEW(node,inv\_tec,year) =G= bound\_new\_capacity\_lo(node,inv\_tec,year)

%SLACK\_CAP\_NEW\_BOUND\_LO% - SLACK\_CAP\_NEW\_BOUND\_LO(node,inv\_tec,year)

;

\*\*\*

\* .. \_equation\_total\_capacity\_bound\_up:

\*

\* Equation TOTAL\_CAPACITY\_BOUND\_UP

\* """"""""""""""""""""""""""""""""

\* This constraint gives upper bounds on the total installed capacity of a technology in a specific year of operation

\* summed over all vintages.

\*

\* .. math::

\* \sum\_{y^V \leq y} \text{CAP}\_{n,t,y,y^V} \leq \text{bound\_total\_capacity\_up}\_{n,t,y} \quad \forall \ t \ \in \ T^{\text{INV}}

\*

\*\*\*

TOTAL\_CAPACITY\_BOUND\_UP(node,inv\_tec,year)$( is\_bound\_total\_capacity\_up(node,inv\_tec,year) )..

SUM(vintage$( map\_period(vintage,year) AND map\_tec\_lifetime(node,inv\_tec,vintage,year) ),

CAP(node,inv\_tec,vintage,year) )

=L= bound\_total\_capacity\_up(node,inv\_tec,year)

%SLACK\_CAP\_TOTAL\_BOUND\_UP% + SLACK\_CAP\_TOTAL\_BOUND\_UP(node,inv\_tec,year)

;

\*\*\*

\* .. \_equation\_total\_capacity\_bound\_lo:

\*

\* Equation TOTAL\_CAPACITY\_BOUND\_LO

\* """"""""""""""""""""""""""""""""

\* This constraint gives lower bounds on the total installed capacity of a technology.

\*

\* .. math::

\* \sum\_{y^V \leq y} \text{CAP}\_{n,t,y,y^V} \geq \text{bound\_total\_capacity\_lo}\_{n,t,y} \quad \forall \ t \ \in \ T^{\text{INV}}

\*

\*\*\*

TOTAL\_CAPACITY\_BOUND\_LO(node,inv\_tec,year)$( is\_bound\_total\_capacity\_lo(node,inv\_tec,year) )..

SUM(vintage$( map\_period(vintage,year) AND map\_tec\_lifetime(node,inv\_tec,vintage,year) ),

CAP(node,inv\_tec,vintage,year) )

=G= bound\_total\_capacity\_lo(node,inv\_tec,year)

%SLACK\_CAP\_TOTAL\_BOUND\_LO% - SLACK\_CAP\_TOTAL\_BOUND\_LO(node,inv\_tec,year)

;

\*\*\*

\* .. \_activity\_bound\_up:

\*

\* Equation ACTIVITY\_BOUND\_UP

\* """"""""""""""""""""""""""

\* This constraint provides upper bounds by mode of a technology activity, summed over all vintages.

\*

\* .. math::

\* \sum\_{y^V \leq y} \text{ACT}\_{n,t,y^V,y,m,h} \leq \text{bound\_activity\_up}\_{n,t,m,y,h}

\*

\*\*\*

ACTIVITY\_BOUND\_UP(node,tec,year,mode,time)$(

is\_bound\_activity\_up(node,tec,year,mode,time) AND map\_tec\_act(node,tec,year,mode,time)

)..

SUM(

vintage$( map\_tec\_lifetime(node,tec,vintage,year) ),

ACT(node,tec,vintage,year,mode,time)

)

=L=

bound\_activity\_up(node,tec,year,mode,time)

%SLACK\_ACT\_BOUND\_UP% + SLACK\_ACT\_BOUND\_UP(node,tec,year,mode,time)

;

\*\*\*

\* .. \_equation\_activity\_bound\_all\_modes\_up:

\*

\* Equation ACTIVITY\_BOUND\_ALL\_MODES\_UP

\* """"""""""""""""""""""""""""""""""""

\* This constraint provides upper bounds of a technology activity across all modes and vintages.

\*

\* .. math::

\* \sum\_{y^V \leq y, m} \text{ACT}\_{n,t,y^V,y,m,h} \leq \text{bound\_activity\_up}\_{n,t,y,'all',h}

\*

\*\*\*

ACTIVITY\_BOUND\_ALL\_MODES\_UP(node,tec,year,time)$( is\_bound\_activity\_up(node,tec,year,'all',time) )..

SUM(

(vintage,mode)$( map\_tec\_lifetime(node,tec,vintage,year) AND map\_tec\_mode(node,tec,year,mode) ),

ACT(node,tec,vintage,year,mode,time)

)

=L=

bound\_activity\_up(node,tec,year,'all',time)

%SLACK\_ACT\_BOUND\_UP% + SLACK\_ACT\_BOUND\_UP(node,tec,year,'all',time)

;

\*\*\*

\* .. \_activity\_bound\_lo:

\*

\* Equation ACTIVITY\_BOUND\_LO

\* """"""""""""""""""""""""""

\* This constraint provides lower bounds by mode of a technology activity, summed over

\* all vintages.

\*

\* .. math::

\* \sum\_{y^V \leq y} \text{ACT}\_{n,t,y^V,y,m,h} \geq \text{bound\_activity\_lo}\_{n,t,y,m,h}

\*

\* We assume that :math:`\text{bound\_activity\_lo}\_{n,t,y,m,h} = 0`

\* unless explicitly stated otherwise.

\*\*\*

ACTIVITY\_BOUND\_LO(node,tec,year,mode,time)$( map\_tec\_act(node,tec,year,mode,time) )..

SUM(

vintage$( map\_tec\_lifetime(node,tec,vintage,year) ),

ACT(node,tec,vintage,year,mode,time)

)

=G=

bound\_activity\_lo(node,tec,year,mode,time)

%SLACK\_ACT\_BOUND\_LO% - SLACK\_ACT\_BOUND\_LO(node,tec,year,mode,time)

;

\*\*\*

\* .. \_equation\_activity\_bound\_all\_modes\_lo:

\*

\* Equation ACTIVITY\_BOUND\_ALL\_MODES\_LO

\* """"""""""""""""""""""""""""""""""""

\* This constraint provides lower bounds of a technology activity across all modes and vintages.

\*

\* .. math::

\* \sum\_{y^V \leq y, m} \text{ACT}\_{n,t,y^V,y,m,h} \geq \text{bound\_activity\_lo}\_{n,t,y,'all',h}

\*

\* We assume that :math:`\text{bound\_activity\_lo}\_{n,t,y,'all',h} = 0`

\* unless explicitly stated otherwise.

\*\*\*

ACTIVITY\_BOUND\_ALL\_MODES\_LO(node,tec,year,time)$( bound\_activity\_lo(node,tec,year,'all',time) )..

SUM(

(vintage,mode)$( map\_tec\_lifetime(node,tec,vintage,year) AND map\_tec\_mode(node,tec,year,mode) ),

ACT(node,tec,vintage,year,mode,time)

)

=G=

bound\_activity\_lo(node,tec,year,'all',time)

%SLACK\_ACT\_BOUND\_LO% - SLACK\_ACT\_BOUND\_LO(node,tec,year,'all',time)

;

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_share\_constraints:

\*

\* Constraints on shares of technologies and commodities

\* -----------------------------------------------------

\* This section allows to include upper and lower bounds on the shares of modes used by a technology

\* or the shares of commodities produced or consumed by groups of technologies.

\*

\* Share constraints on activity by mode

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\* .. \_equation\_shares\_mode\_up:

\*

\* Equation SHARES\_MODE\_UP

\* """""""""""""""""""""""

\* This constraint provides upper bounds of the share of the activity of one mode

\* of a technology. For example, it could limit the share of heat that can be produced

\* in a combined heat and electricity power plant.

\*

\* .. math::

\* \text{ACT}\_{n^L,t,y^V,y,m,h^A}

\* \leq \text{share\_mode\_up}\_{p,n,t,y,m,h} \cdot

\* \sum\_{m'} \text{ACT}\_{n^L,t,y^V,y,m',h^A}

\*

\*\*\*

SHARE\_CONSTRAINT\_MODE\_UP(shares,node,tec,mode,year,time)$(

map\_tec\_act(node,tec,year,mode,time) AND

share\_mode\_up(shares,node,tec,mode,year,time)

)..

\* activity of mode to be constrained

SUM(

vintage$( map\_tec\_lifetime(node,tec,vintage,year) ),

ACT(node,tec,vintage,year,mode,time)

)

=L=

share\_mode\_up(shares,node,tec,mode,year,time) \*

\* activity aggregated over all modes

SUM(

(vintage,mode2)$( map\_tec\_lifetime(node,tec,vintage,year) AND map\_tec\_mode(node,tec,year,mode2) ),

ACT(node,tec,vintage,year,mode2,time)

) ;

\*\*\*

\* .. \_equation\_shares\_mode\_lo:

\*

\* Equation SHARES\_MODE\_LO

\* """""""""""""""""""""""

\* This constraint provides lower bounds of the share of the activity of one mode of a technology.

\*

\* .. math::

\* \text{ACT}\_{n^L,t,y^V,y,m,h^A}

\* \geq \text{share\_mode\_lo}\_{p,n,t,y,m,h} \cdot

\* \sum\_{m'} \text{ACT}\_{n^L,t,y^V,y,m',h^A}

\*

\*\*\*

SHARE\_CONSTRAINT\_MODE\_LO(shares,node,tec,mode,year,time)$(

map\_tec\_act(node,tec,year,mode,time) AND

share\_mode\_lo(shares,node,tec,mode,year,time)

)..

\* activity of mode to be constrained

SUM(

vintage$( map\_tec\_lifetime(node,tec,vintage,year) ),

ACT(node,tec,vintage,year,mode,time)

)

=G=

share\_mode\_lo(shares,node,tec,mode,year,time) \*

\* activity aggregated over all modes

SUM(

(vintage,mode2)$( map\_tec\_lifetime(node,tec,vintage,year) AND map\_tec\_mode(node,tec,year,mode2) ),

ACT(node,tec,vintage,year,mode2,time)

) ;

\*\*\*

\* .. \_section\_share\_constraints\_commodities:

\*

\* Share constraints on commodities

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\* These constraints allow to set upper and lower bound on the quantity of commodities produced or consumed by a group

\* of technologies relative to the commodities produced or consumed by another group.

\*

\* The implementation is generic and flexible, so that any combination of commodities, levels, technologies and nodes

\* can be put in relation to any other combination.

\*

\* The notation :math:`P^{\text{share}}` represents the mapping set `map\_shares\_commodity\_share` denoting all technology types,

\* nodes, commodities and levels to be included in the numerator, and :math:`P^{\text{total}}` is

\* the equivalent mapping set `map\_shares\_commodity\_total` for the denominator.

\*

\* .. \_equation\_share\_constraint\_commodity\_up:

\*

\* Equation SHARE\_CONSTRAINT\_COMMODITY\_UP

\* """"""""""""""""""""""""""""""""""""""

\* .. math::

\* & \sum\_{\substack{n^L,t,m,h^A \\ y^V \leq y, (n,\widehat{t},m,c,l) \sim P^{\text{share}}}}

\* ( \text{output}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} + \text{input}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} ) \\

\* & \quad \cdot \text{duration\_time\_rel}\_{h,h^A} \cdot \text{ACT}\_{n^L,t,y^V,y,m,h^A} \\

\* & \geq

\* \text{share\_commodity\_up}\_{p,n,y,h} \cdot

\* \sum\_{\substack{n^L,t,m,h^A \\ y^V \leq y, (n,\widehat{t},m,c,l) \sim P^{\text{total}}}}

\* ( \text{output}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} + \text{input}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} ) \\

\* & \quad \cdot \text{duration\_time\_rel}\_{h,h^A} \cdot \text{ACT}\_{n^L,t,y^V,y,m,h^A}

\*

\* This constraint is only active if :math:`\text{share\_commodity\_up}\_{p,n,y,h}` is defined.

\*\*\*

SHARE\_CONSTRAINT\_COMMODITY\_UP(shares,node\_share,year,time)$( share\_commodity\_up(shares,node\_share,year,time) )..

\* activity by type\_tec\_share technologies with map\_shares\_generic\_share entries and a specific mode

SUM( (node,location,type\_tec\_share,tec,vintage,mode,commodity,level,time2)$(

( map\_shares\_commodity\_share(shares,node\_share,node,type\_tec\_share,mode,commodity,level) OR

map\_shares\_commodity\_share(shares,node\_share,node,type\_tec\_share,'all',commodity,level) ) AND

cat\_tec(type\_tec\_share,tec) AND

map\_tec\_act(location,tec,year,mode,time2) AND

map\_tec\_lifetime(location,tec,vintage,year)

),

(

output(location,tec,vintage,year,mode,node,commodity,level,time2,time) +

input(location,tec,vintage,year,mode,node,commodity,level,time2,time)

) \*

duration\_time\_rel(time,time2) \*

ACT(location,tec,vintage,year,mode,time2)

)

=L=

share\_commodity\_up(shares,node\_share,year,time) \* (

\* total input and output by `type\_tec\_total` technologies mapped to respective commodity, level and node

SUM( (node,location,type\_tec\_total,tec,vintage,mode,commodity,level,time2)$(

( map\_shares\_commodity\_total(shares,node\_share,node,type\_tec\_total,mode,commodity,level) OR

map\_shares\_commodity\_total(shares,node\_share,node,type\_tec\_total,'all',commodity,level) ) AND

cat\_tec(type\_tec\_total,tec) AND

map\_tec\_act(location,tec,year,mode,time2) AND

map\_tec\_lifetime(location,tec,vintage,year)

),

(

output(location,tec,vintage,year,mode,node,commodity,level,time2,time) +

input(location,tec,vintage,year,mode,node,commodity,level,time2,time)

) \*

duration\_time\_rel(time,time2) \*

ACT(location,tec,vintage,year,mode,time2)

) ) ;

\*\*\*

\* .. \_equation\_share\_constraint\_commodity\_lo:

\*

\* Equation SHARE\_CONSTRAINT\_COMMODITY\_LO

\* """"""""""""""""""""""""""""""""""""""

\* .. math::

\* & \sum\_{\substack{n^L,t,m,h^A \\ y^V \leq y, (n,\widehat{t},m,c,l) \sim P^{\text{share}}}}

\* ( \text{output}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} + \text{input}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} ) \\

\* & \quad \cdot \text{duration\_time\_rel}\_{h,h^A} \cdot \text{ACT}\_{n^L,t,y^V,y,m,h^A} \\

\* & \leq

\* \text{share\_commodity\_lo}\_{p,n,y,h} \cdot

\* \sum\_{\substack{n^L,t,m,h^A \\ y^V \leq y, (n,\widehat{t},m,c,l) \sim P^{\text{total}}}}

\* ( \text{output}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} + \text{input}\_{n^L,t,y^V,y,m,n,c,l,h^A,h} ) \\

\* & \quad \cdot \text{duration\_time\_rel}\_{h,h^A} \cdot \text{ACT}\_{n^L,t,y^V,y,m,h^A}

\*

\* This constraint is only active if :math:`\text{share\_commodity\_lo}\_{p,n,y,h}` is defined.

\*\*\*

SHARE\_CONSTRAINT\_COMMODITY\_LO(shares,node\_share,year,time)$( share\_commodity\_lo(shares,node\_share,year,time) )..

\* total input and output by `type\_tec\_share` technologies mapped to respective commodity, level and node

SUM( (node,location,type\_tec\_share,tec,vintage,mode,commodity,level,time2)$(

( map\_shares\_commodity\_share(shares,node\_share,node,type\_tec\_share,mode,commodity,level) OR

map\_shares\_commodity\_share(shares,node\_share,node,type\_tec\_share,'all',commodity,level) ) AND

cat\_tec(type\_tec\_share,tec) AND

map\_tec\_act(location,tec,year,mode,time2) AND

map\_tec\_lifetime(location,tec,vintage,year)

),

(

output(location,tec,vintage,year,mode,node,commodity,level,time2,time) +

input(location,tec,vintage,year,mode,node,commodity,level,time2,time)

) \*

duration\_time\_rel(time,time2) \*

ACT(location,tec,vintage,year,mode,time2)

)

=G=

share\_commodity\_lo(shares,node\_share,year,time) \* (

\* total input and output by `type\_tec\_total` technologies mapped to respective commodity, level and node

SUM( (node,location,type\_tec\_total,tec,vintage,mode,commodity,level,time2)$(

( map\_shares\_commodity\_total(shares,node\_share,node,type\_tec\_total,mode,commodity,level) OR

map\_shares\_commodity\_total(shares,node\_share,node,type\_tec\_total,'all',commodity,level) ) AND

cat\_tec(type\_tec\_total,tec) AND

map\_tec\_act(location,tec,year,mode,time2) AND

map\_tec\_lifetime(location,tec,vintage,year)

),

(

output(location,tec,vintage,year,mode,node,commodity,level,time2,time) +

input(location,tec,vintage,year,mode,node,commodity,level,time2,time)

) \*

duration\_time\_rel(time,time2) \*

ACT(location,tec,vintage,year,mode,time2)

) ) ;

\*\*\*

\* .. \_dynamic\_constraints:

\*

\* Dynamic constraints on new capacity and activity

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* The constraints in this section specify dynamic upper and lower bounds on new capacity and activity.

\* These can be used to model limits on market penetration and/or rates of expansion or phase-out of a technology.

\*

\* The formulation directly includes the option for 'soft' relaxations of dynamic constraints

\* (cf. Keppo and Strubegger, 2010 :cite:`Keppo-2010`).

\*

\* See also the :ref:`corresponding parameter definitions <section\_parameter\_dynamic\_constraints>`.

\*

\* .. \_equation\_new\_capacity\_constraint\_up:

\*

\* Equation NEW\_CAPACITY\_CONSTRAINT\_UP

\* """""""""""""""""""""""""""""""""""

\* The level of new capacity additions cannot be greater than an initial value (compounded over the period duration),

\* annual growth of the existing 'capital stock', and a "soft" relaxation of the upper bound.

\*

\* .. math::

\* gncu^1\_{n,t,y} =

\* & \left( 1 + \text{growth\_new\_capacity\_up}\_{n,t,y} \right)^{|y|} \\

\* gncu^2\_{n,t,y} =

\* & \frac{gncu^1\_{n,t,y} -1 }{|y| \cdot \log{\left(1 + \text{growth\_new\_capacity\_up}\_{n,t,y} \right)}} \\

\* k^{gncu}\_{n,t,y\_a,y\_b} =

\* & \frac{gncu^1\_{n,t,y\_a} \cdot gncu^2\_{n,t,y\_b}}{gncu^2\_{n,t,y\_a}} \\

\* \text{CAP\_NEW}\_{n,t,y} \leq

\* & \big( \text{CAP\_NEW}\_{n,t,y\_{-1}} + \text{historical\_new\_capacity}\_{n,t,y\_{-1}} \big. \\

\* & \big. + \text{initial\_new\_capacity\_up}\_{n,t,y} \big) \cdot k^{gncu}\_{n,t,y\_{-1},y} \\

\* & + \left(\text{CAP\_NEW\_UP}\_{n,t,y} \cdot \left((1 + \text{soft\_new\_capacity\_up}\_{n,t,y})^{|y|} - 1 \right)\right) \\

\* \forall & \ t \ \in \ T^{\text{INV}}

\*

\* Here, :math:`|y|` is the number of years in period :math:`y`, i.e., :math:`\text{duration\_period}\_{y}`.

\*\*\*

\* Compute auxiliary values

gncu\_1(n,t,y\_all)$inv\_tec(t) = POWER(1 + growth\_new\_capacity\_up(n,t,y\_all), duration\_period(y\_all));

gncu\_2(n,t,y\_all) = 1;

gncu\_2(n,t,y\_all)$growth\_new\_capacity\_up(n,t,y\_all) = (

(gncu\_1(n,t,y\_all) - 1) / (duration\_period(y\_all) \* LOG(1 + growth\_new\_capacity\_up(n,t,y\_all)))

);

\* Ratio of CAP\_NEW(n,t,y\_all) to CAP\_NEW(n,t,y\_prev)

k\_gncu(n,t,y\_prev,y\_all)$(

seq\_period(y\_prev,y\_all)

# Same condition as the equation, below

AND inv\_tec(t) AND map\_tec(n,t,y\_all) AND is\_dynamic\_new\_capacity\_up(n,t,y\_all)

) = gncu\_1(n,t,y\_prev) \* gncu\_2(n,t,y\_all) / gncu\_2(n,t,y\_prev);

NEW\_CAPACITY\_CONSTRAINT\_UP(n,t,y\_)$(

inv\_tec(t) AND map\_tec(n,t,y\_) AND is\_dynamic\_new\_capacity\_up(n,t,y\_)

)..

CAP\_NEW(n,t,y\_)

=L=

# 'Hard' constraint value

SUM(

y\_prev$seq\_period(y\_prev, y\_),

(

# New capacity in previous model period

CAP\_NEW(n,t,y\_prev)$(model\_horizon(y\_prev) AND map\_tec(n,t,y\_prev))

# New capacity in previous historical period

+ historical\_new\_capacity(n,t,y\_prev)

# Otherwise, maximum initial value

# FIXME Do not use this if any of the above are non-zero

+ initial\_new\_capacity\_up(n,t,y\_)

)

\* k\_gncu(n,t,y\_prev,y\_)

)

# 'Soft' relaxation of constraint

+ (

CAP\_NEW\_UP(n,t,y\_) \* (POWER(1 + soft\_new\_capacity\_up(n,t,y\_), duration\_period(y\_)) - 1)

)$soft\_new\_capacity\_up(n,t,y\_)

# Additional relaxation for calibration and debugging

%SLACK\_CAP\_NEW\_DYNAMIC\_UP% + SLACK\_CAP\_NEW\_DYNAMIC\_UP(n,t,y\_)

;

\* GAMS implementation comment:

\* The sums in the constraint have to be over `year\_all2` (not `year2`) to also get the dynamic effect from historical

\* new capacity. If one were to sum over `year2`, periods prior to the first model year would be ignored.

\* Furthermore, as `CAP\_NEW` is derived from the value in a previous period, any change in the duration of two consecutive

\* model periods needs to be accounted for. This is done by using the ratio of two consecutive model periods as a

\* multiplication factor.

\*\*\*

\* .. \_equation\_new\_capacity\_soft\_constraint\_up:

\*

\* Equation NEW\_CAPACITY\_SOFT\_CONSTRAINT\_UP

\* """"""""""""""""""""""""""""""""""""""""

\* This constraint ensures that the relaxation of the dynamic constraint on new capacity (investment) does not exceed

\* the level of the investment in the previous period (cf. Keppo and Strubegger, 2010 :cite:`Keppo-2010`).

\*

\* .. math::

\* \text{CAP\_NEW\_UP}\_{n,t,y} \leq \sum\_{y-1} \text{CAP\_NEW}\_{n^L,t,y-1} & \text{if } y \neq \text{'first\_period'} \\

\* + \sum\_{y-1} \text{historical\_new\_capacity}\_{n^L,t,y-1} & \text{if } y = \text{'first\_period'} \\

\* \quad \forall \ t \ \in \ T^{\text{INV}}

\*

\*\*\*

NEW\_CAPACITY\_SOFT\_CONSTRAINT\_UP(node,inv\_tec,year)$( soft\_new\_capacity\_up(node,inv\_tec,year) )..

CAP\_NEW\_UP(node,inv\_tec,year) =L=

SUM(year2$( seq\_period(year2,year) ),

CAP\_NEW(node,inv\_tec,year2)) $ (NOT first\_period(year))

+ SUM(year\_all2$( seq\_period(year\_all2,year) ),

historical\_new\_capacity(node,inv\_tec,year\_all2)) $ first\_period(year)

;

\*\*\*

\* .. \_equation\_new\_capacity\_constraint\_lo:

\*

\* Equation NEW\_CAPACITY\_CONSTRAINT\_LO

\* """""""""""""""""""""""""""""""""""

\* This constraint gives dynamic lower bounds on new capacity.

\*

\* .. math::

\* \text{CAP\_NEW}\_{n,t,y}

\* \geq & \Bigg(- \text{initial\_new\_capacity\_lo}\_{n,t,y}

\* \cdot \frac{ \Big( 1 + \text{growth\_new\_capacity\_lo}\_{n,t,y} \Big)^{|y|} }

\* { \text{growth\_new\_capacity\_lo}\_{n,t,y} } \\

\* & + \Big( \text{CAP\_NEW}\_{n,t,y-1} + \text{historical\_new\_capacity}\_{n,t,y-1} \Big) \\

\* & \hspace{2 cm} \cdot \Big( 1 + \text{growth\_new\_capacity\_lo}\_{n,t,y} \Big)^{|y|} \\

\* & - \text{CAP\_NEW\_LO}\_{n,t,y} \cdot \Bigg( \Big( 1 + \text{soft\_new\_capacity\_lo}\_{n,t,y}\Big)^{|y|} - 1 \Bigg)\Bigg) \\

\* & \* \frac{|y-1|}{|y|} \\

\* & \quad \forall \ t \ \in \ T^{\text{INV}}

\*

\*\*\*

NEW\_CAPACITY\_CONSTRAINT\_LO(node,inv\_tec,year)$( map\_tec(node,inv\_tec,year)

AND is\_dynamic\_new\_capacity\_lo(node,inv\_tec,year) )..

\* actual new capacity

CAP\_NEW(node,inv\_tec,year) =G=

\* initial new capacity (compounded over the duration of the period)

(- initial\_new\_capacity\_lo(node,inv\_tec,year) \* (

( ( POWER( 1 + growth\_new\_capacity\_lo(node,inv\_tec,year) , duration\_period(year) ) - 1 )

/ growth\_new\_capacity\_lo(node,inv\_tec,year) )$( growth\_new\_capacity\_lo(node,inv\_tec,year) )

+ ( duration\_period(year) )$( NOT growth\_new\_capacity\_lo(node,inv\_tec,year) )

)

\* growth of 'capital stock' from previous period

+ SUM(year\_all2$( seq\_period(year\_all2,year) ),

CAP\_NEW(node,inv\_tec,year\_all2)$( map\_tec(node,inv\_tec,year\_all2) AND model\_horizon(year\_all2) )

+ historical\_new\_capacity(node,inv\_tec,year\_all2)

# placeholder for spillover across nodes, technologies, periods (other than immediate predecessor)

) \* POWER( 1 + growth\_new\_capacity\_lo(node,inv\_tec,year) , duration\_period(year) )

\* 'soft' relaxation of dynamic constraints

- ( CAP\_NEW\_LO(node,inv\_tec,year)

\* ( POWER( 1 + soft\_new\_capacity\_lo(node,inv\_tec,year) , duration\_period(year) ) - 1 )

)$( soft\_new\_capacity\_lo(node,inv\_tec,year) ))

\* SUM(year\_all2$( seq\_period(year\_all2,year) ),

( duration\_period(year\_all2) / duration\_period(year) ))

\* optional relaxation for calibration and debugging

%SLACK\_CAP\_NEW\_DYNAMIC\_LO% - SLACK\_CAP\_NEW\_DYNAMIC\_LO(node,inv\_tec,year)

;

\* GAMS implementation comment:

\* The sums in the constraint have to be over `year\_all2` (not `year2`) to also get the dynamic effect from historical

\* new capacity. If one would sum over `year2`, periods prior to the first model year would be ignored.

\* Furthermore, as `CAP\_NEW` is derived from the value in a previous period, any change in the duration of two consecutive

\* model periods needs to be accounted for. This is done by using the ratio of two consecutive model periods as a

\* multiplication factor.

\*\*\*

\* .. \_equation\_new\_capacity\_soft\_constraint\_lo:

\*

\* Equation NEW\_CAPACITY\_SOFT\_CONSTRAINT\_LO

\* """"""""""""""""""""""""""""""""""""""""

\* This constraint ensures that the relaxation of the dynamic constraint on new capacity does not exceed

\* level of the investment in the previous year.

\*

\* .. math::

\* \text{CAP\_NEW\_LO}\_{n,t,y} \leq \sum\_{y-1} \text{CAP\_NEW}\_{n^L,t,y-1} & \text{if } y \neq \text{'first\_period'} \\

\* + \sum\_{y-1} \text{historical\_new\_capacity}\_{n^L,t,y-1} & \text{if } y = \text{'first\_period'} \\

\* \quad \forall \ t \ \in \ T^{\text{INV}}

\*

\*\*\*

NEW\_CAPACITY\_SOFT\_CONSTRAINT\_LO(node,inv\_tec,year)$( soft\_new\_capacity\_lo(node,inv\_tec,year) )..

CAP\_NEW\_LO(node,inv\_tec,year) =L=

SUM(year2$( seq\_period(year2,year) ),

CAP\_NEW(node,inv\_tec,year2) ) $ (NOT first\_period(year))

+ SUM(year\_all2$( seq\_period(year\_all2,year) ),

historical\_new\_capacity(node,inv\_tec,year\_all2) ) $ first\_period(year)

;

\*\*\*

\* .. \_equation\_activity\_constraint\_up:

\*

\* Equation ACTIVITY\_CONSTRAINT\_UP

\* """""""""""""""""""""""""""""""

\* This constraint gives dynamic upper bounds on the market penetration of a technology activity.

\*

\* .. math::

\* \sum\_{y^V \leq y,m} \text{ACT}\_{n,t,y^V,y,m,h}

\* \leq & ~ \text{initial\_activity\_up}\_{n,t,y,h}

\* \cdot \frac{ \Big( 1 + \text{growth\_activity\_up}\_{n,t,y,h} \Big)^{|y|} - 1 }

\* { \text{growth\_activity\_up}\_{n,t,y,h} } \\

\* & + \bigg( \sum\_{y^V \leq y-1,m} \text{ACT}\_{n,t,y^V,y-1,m,h}

\* + \sum\_{m} \text{historical\_activity}\_{n,t,y-1,m,h} \bigg) \\

\* & \hspace{2 cm} \cdot \Big( 1 + \text{growth\_activity\_up}\_{n,t,y,h} \Big)^{|y|} \\

\* & + \text{ACT\_UP}\_{n,t,y,h} \cdot \Bigg( \Big( 1 + \text{soft\_activity\_up}\_{n,t,y,h} \Big)^{|y|} - 1 \Bigg)

\*

\*\*\*

ACTIVITY\_CONSTRAINT\_UP(node,tec,year,time)$( map\_tec\_time(node,tec,year,time)

AND is\_dynamic\_activity\_up(node,tec,year,time) )..

\* actual activity (summed over modes)

SUM((vintage,mode)$( map\_tec\_lifetime(node,tec,vintage,year) AND map\_tec\_mode(node,tec,year,mode) ),

ACT(node,tec,vintage,year,mode,time) ) =L=

\* initial activity (compounded over the duration of the period)

initial\_activity\_up(node,tec,year,time) \* (

( ( POWER( 1 + growth\_activity\_up(node,tec,year,time) , duration\_period(year) ) - 1 )

/ growth\_activity\_up(node,tec,year,time) )$( growth\_activity\_up(node,tec,year,time) )

+ ( duration\_period(year) )$( NOT growth\_activity\_up(node,tec,year,time) )

)

\* growth of 'capital stock' from previous period

+ SUM((year\_all2)$( seq\_period(year\_all2,year) ),

SUM((vintage,mode)$( map\_tec\_lifetime(node,tec,vintage,year\_all2) AND map\_tec\_mode(node,tec,year\_all2,mode)

AND model\_horizon(year\_all2) ),

ACT(node,tec,vintage,year\_all2,mode,time) )

+ SUM(mode, historical\_activity(node,tec,year\_all2,mode,time) )

# placeholder for spillover across nodes, technologies, periods (other than immediate predecessor)

)

\* POWER( 1 + growth\_activity\_up(node,tec,year,time) , duration\_period(year) )

\* 'soft' relaxation of dynamic constraints

+ ( ACT\_UP(node,tec,year,time)

\* ( POWER( 1 + soft\_activity\_up(node,tec,year,time) , duration\_period(year) ) - 1 )

)$( soft\_activity\_up(node,tec,year,time) )

\* optional relaxation for calibration and debugging

%SLACK\_ACT\_DYNAMIC\_UP% + SLACK\_ACT\_DYNAMIC\_UP(node,tec,year,time)

;

\*\*\*

\* .. \_equation\_activity\_soft\_constraint\_up:

\*

\* Equation ACTIVITY\_SOFT\_CONSTRAINT\_UP

\* """"""""""""""""""""""""""""""""""""

\* This constraint ensures that the relaxation of the dynamic activity constraint does not exceed the

\* level of the activity in the previous period.

\*

\* .. math::

\* \text{ACT\_UP}\_{n,t,y,h} \leq \sum\_{y^V \leq y,m,y-1} \text{ACT}\_{n^L,t,y^V,y-1,m,h} & \text{if } y \neq \text{'first\_period'} \\

\* + \sum\_{m,y-1} \text{historical\_activity}\_{n^L,t,y-1,m,h} & \text{if } y = \text{'first\_period'}

\*

\*

\*\*\*

ACTIVITY\_SOFT\_CONSTRAINT\_UP(node,tec,year,time)$( soft\_activity\_up(node,tec,year,time) )..

ACT\_UP(node,tec,year,time) =L=

SUM((vintage,mode,year2)$( map\_tec\_lifetime(node,tec,vintage,year2) AND map\_tec\_act(node,tec,year2,mode,time)

AND seq\_period(year2,year) ),

ACT(node,tec,vintage,year2,mode,time) ) $ (NOT first\_period(year))

+ SUM((mode,year\_all2)$( seq\_period(year\_all2,year) ),

historical\_activity(node,tec,year\_all2,mode,time) ) $ first\_period(year)

;

\*\*\*

\* Equation ACTIVITY\_CONSTRAINT\_LO

\* """""""""""""""""""""""""""""""

\* This constraint gives dynamic lower bounds on the market penetration of a technology activity.

\*

\* .. math::

\* \sum\_{y^V \leq y,m} \text{ACT}\_{n,t,y^V,y,m,h}

\* \geq & - \text{initial\_activity\_lo}\_{n,t,y,h}

\* \cdot \frac{ \Big( 1 + \text{growth\_activity\_lo}\_{n,t,y,h} \Big)^{|y|} - 1 }

\* { \text{growth\_activity\_lo}\_{n,t,y,h} } \\

\* & + \bigg( \sum\_{y^V \leq y-1,m} \text{ACT}\_{n,t,y^V,y-1,m,h}

\* + \sum\_{m} \text{historical\_activity}\_{n,t,y-1,m,h} \bigg) \\

\* & \hspace{2 cm} \cdot \Big( 1 + \text{growth\_activity\_lo}\_{n,t,y,h} \Big)^{|y|} \\

\* & - \text{ACT\_LO}\_{n,t,y,h} \cdot \Bigg( \Big( 1 + \text{soft\_activity\_lo}\_{n,t,y,h} \Big)^{|y|} - 1 \Bigg)

\*

\*\*\*

ACTIVITY\_CONSTRAINT\_LO(node,tec,year,time)$( map\_tec\_time(node,tec,year,time)

AND is\_dynamic\_activity\_lo(node,tec,year,time) )..

\* actual activity (summed over modes)

SUM((vintage,mode)$( map\_tec\_lifetime(node,tec,vintage,year) AND map\_tec\_mode(node,tec,year,mode) ),

ACT(node,tec,vintage,year,mode,time) ) =G=

\* initial activity (compounded over the duration of the period)

- initial\_activity\_lo(node,tec,year,time) \* (

( ( POWER( 1 + growth\_activity\_lo(node,tec,year,time) , duration\_period(year) ) - 1 )

/ growth\_activity\_lo(node,tec,year,time) )$( growth\_activity\_lo(node,tec,year,time) )

+ ( duration\_period(year) )$( NOT growth\_activity\_lo(node,tec,year,time) )

)

\* growth of 'capital stock' from previous period

+ SUM((year\_all2)$( seq\_period(year\_all2,year) ),

SUM((vintage,mode)$( map\_tec\_lifetime(node,tec,vintage,year\_all2) AND map\_tec\_mode(node,tec,year\_all2,mode)

AND model\_horizon(year\_all2)),

ACT(node,tec,vintage,year\_all2,mode,time) )

+ SUM(mode, historical\_activity(node,tec,year\_all2,mode,time) )

# placeholder for spillover across nodes, technologies, periods (other than immediate predecessor)

)

\* POWER( 1 + growth\_activity\_lo(node,tec,year,time) , duration\_period(year) )

\* 'soft' relaxation of dynamic constraints

- ( ACT\_LO(node,tec,year,time)

\* ( POWER( 1 + soft\_activity\_lo(node,tec,year,time) , duration\_period(year) ) - 1 )

)$( soft\_activity\_lo(node,tec,year,time) )

\* optional relaxation for calibration and debugging

%SLACK\_ACT\_DYNAMIC\_LO% - SLACK\_ACT\_DYNAMIC\_LO(node,tec,year,time)

;

\*\*\*

\* .. \_equation\_activity\_soft\_constraint\_lo:

\*

\* Equation ACTIVITY\_SOFT\_CONSTRAINT\_LO

\* """"""""""""""""""""""""""""""""""""

\* This constraint ensures that the relaxation of the dynamic activity constraint does not exceed the

\* level of the activity in the previous period.

\*

\* .. math::

\* \text{ACT\_LO}\_{n,t,y,h} \leq \sum\_{y^V \leq y,m,y-1} \text{ACT}\_{n^L,t,y^V,y-1,m,h} & \text{if } y \neq \text{'first\_period'} \\

\* + \sum\_{m,y-1} \text{historical\_activity}\_{n^L,t,y-1,m,h} & \text{if } y = \text{'first\_period'}

\*

\*\*\*

ACTIVITY\_SOFT\_CONSTRAINT\_LO(node,tec,year,time)$( soft\_activity\_lo(node,tec,year,time) )..

ACT\_LO(node,tec,year,time) =L=

SUM((vintage,mode,year2)$( map\_tec\_lifetime(node,tec,vintage,year2) AND map\_tec\_act(node,tec,year2,mode,time)

AND seq\_period(year2,year) ),

ACT(node,tec,vintage,year2,mode,time) ) $ (NOT first\_period(year))

+ SUM((mode,year\_all2)$( seq\_period(year\_all2,year) ),

historical\_activity(node,tec,year\_all2,mode,time) ) $ first\_period(year)

;

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_section\_emission:

\*

\* Emission section

\* ----------------

\*

\* Auxiliary variable for aggregate emissions

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* .. \_equation\_emission\_equivalence:

\*

\* Equation EMISSION\_EQUIVALENCE

\* """""""""""""""""""""""""""""

\* This constraint simplifies the notation of emissions aggregated over different technology types

\* and the land-use model emulator. The formulation includes emissions from all sub-nodes :math:`n^L` of :math:`n`.

\*

\* .. math::

\* \text{EMISS}\_{n,e,\widehat{t},y} =

\* \sum\_{n^L \in N(n)} \Bigg(

\* \sum\_{t \in T(\widehat{t}),y^V \leq y,m,h }

\* \text{emission\_factor}\_{n^L,t,y^V,y,m,e} \cdot \text{ACT}\_{n^L,t,y^V,y,m,h} \\

\* + \sum\_{s} \ \text{land\_emission}\_{n^L,s,y,e} \cdot \text{LAND}\_{n^L,s,y}

\* \text{ if } \widehat{t} \in \widehat{T}^{LAND} \Bigg)

\*

\* .. versionchanged:: v3.11.0

\*

\* ``type\_tec`` elements that appear in either of the :ref:`mapping sets <section\_maps\_def>`

\* ``map\_shares\_commodity\_share`` or ``map\_shares\_commodity\_total`` are excluded from this equation,

\* and thus also from the domain of the ``EMISS`` :ref:`variable <section\_decision\_variable\_def>`.

\*\*\*

EMISSION\_EQUIVALENCE(node,emission,type\_tec,year)$(

NOT type\_tec\_share(type\_tec) AND NOT type\_tec\_total(type\_tec)

)..

EMISS(node,emission,type\_tec,year)

=E=

SUM(location$( map\_node(node,location) ),

\* emissions from technology activity

SUM((tec,vintage,mode,time)$( cat\_tec(type\_tec,tec)

AND map\_tec\_act(location,tec,year,mode,time) AND map\_tec\_lifetime(location,tec,vintage,year) ),

emission\_factor(location,tec,vintage,year,mode,emission) \* ACT(location,tec,vintage,year,mode,time) )

\* emissions from land use if 'type\_tec' is included in the dynamic set 'type\_tec\_land'

+ SUM(land\_scenario$( type\_tec\_land(type\_tec) ),

land\_emission(location,land\_scenario,year,emission) \* LAND(location,land\_scenario,year) )

) ;

\*\*\*

\* Bound on emissions

\* ^^^^^^^^^^^^^^^^^^

\*

\* .. \_emission\_constraint:

\*

\* Equation EMISSION\_CONSTRAINT

\* """"""""""""""""""""""""""""

\* This constraint enforces upper bounds on emissions (by emission type). For all bounds that include multiple periods,

\* the parameter :math:`\text{bound\_emission}\_{n,\widehat{e},\widehat{t},\widehat{y}}` is scaled to represent average annual

\* emissions over all years included in the year-set :math:`\widehat{y}`.

\*

\* The formulation includes historical emissions and allows to model constraints ranging over both the model horizon

\* and historical periods.

\*

\* .. math::

\* \frac{

\* \sum\_{y' \in Y(\widehat{y}), e \in E(\widehat{e})}

\* \begin{array}{l}

\* \text{duration\_period}\_{y'} \cdot \text{emission\_scaling}\_{\widehat{e},e} \cdot \\

\* \Big( \text{EMISS}\_{n,e,\widehat{t},y'} + \sum\_{m} \text{historical\_emission}\_{n,e,\widehat{t},y'} \Big)

\* \end{array}

\* }

\* { \sum\_{y' \in Y(\widehat{y})} \text{duration\_period}\_{y'} }

\* \leq \text{bound\_emission}\_{n,\widehat{e},\widehat{t},\widehat{y}}

\*

\*\*\*

EMISSION\_CONSTRAINT(node,type\_emission,type\_tec,type\_year)$is\_bound\_emission(node,type\_emission,type\_tec,type\_year)..

SUM( (year\_all2,emission)$( cat\_year(type\_year,year\_all2) AND cat\_emission(type\_emission,emission) ),

duration\_period(year\_all2) \* emission\_scaling(type\_emission,emission) \*

( EMISS(node,emission,type\_tec,year\_all2)$( year(year\_all2) )

+ historical\_emission(node,emission,type\_tec,year\_all2) )

)

/ SUM(year\_all2$( cat\_year(type\_year,year\_all2) ), duration\_period(year\_all2) )

=L= bound\_emission(node,type\_emission,type\_tec,type\_year) ;

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_section\_landuse\_emulator:

\*

\* Land-use model emulator section

\* -------------------------------

\*

\* Bounds on total land use

\* ^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* .. \_equation\_land\_constraint:

\*

\* Equation LAND\_CONSTRAINT

\* """"""""""""""""""""""""

\* This constraint enforces a meaningful result of the land-use model emulator,

\* in particular a bound on the total land used in |MESSAGEix|.

\* The linear combination of land scenarios must be equal to 1.

\*

\* .. math::

\* \sum\_{s \in S} \text{LAND}\_{n,s,y} = 1

\*

\*\*\*

LAND\_CONSTRAINT(node,year)$( SUM(land\_scenario$( map\_land(node,land\_scenario,year) ), 1 ) ) ..

SUM(land\_scenario$( map\_land(node,land\_scenario,year) ), LAND(node,land\_scenario,year) ) =E= 1 ;

\*\*\*

\* Dynamic constraints on land use

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\* These constraints enforces upper and lower bounds on the change rate per land scenario.

\*

\* .. \_equation\_dynamic\_land\_scen\_constraint\_up:

\*

\* Equation DYNAMIC\_LAND\_SCEN\_CONSTRAINT\_UP

\* """"""""""""""""""""""""""""""""""""""""

\*

\* .. math::

\* \text{LAND}\_{n,s,y}

\* \leq & \text{initial\_land\_scen\_up}\_{n,s,y}

\* \cdot \frac{ \Big( 1 + \text{growth\_land\_scen\_up}\_{n,s,y} \Big)^{|y|} - 1 }

\* { \text{growth\_land\_scen\_up}\_{n,s,y} } \\

\* & + \big( \text{LAND}\_{n,s,y-1} + \text{historical\_land}\_{n,s,y-1} \big)

\* \cdot \Big( 1 + \text{growth\_land\_scen\_up}\_{n,s,y} \Big)^{|y|}

\*

\*\*\*

DYNAMIC\_LAND\_SCEN\_CONSTRAINT\_UP(node,land\_scenario,year)$( map\_land(node,land\_scenario,year)

AND is\_dynamic\_land\_scen\_up(node,land\_scenario,year) )..

\* share of land scenario in

LAND(node,land\_scenario,year) =L=

\* initial 'new' land used for that type (compounded over the duration of the period)

initial\_land\_scen\_up(node,land\_scenario,year) \* (

( ( POWER( 1 + growth\_land\_scen\_up(node,land\_scenario,year) , duration\_period(year) ) - 1 )

/ growth\_land\_scen\_up(node,land\_scenario,year) )$( growth\_land\_scen\_up(node,land\_scenario,year) )

+ ( duration\_period(year) )$( NOT growth\_land\_scen\_up(node,land\_scenario,year) )

)

\* expansion of land scenario from previous period

+ SUM((year\_all2)$( seq\_period(year\_all2,year) ),

( LAND(node,land\_scenario,year\_all2)$( model\_horizon(year\_all2) )

+ historical\_land(node,land\_scenario,year\_all2) )

\* POWER( 1 + growth\_land\_scen\_up(node,land\_scenario,year) , duration\_period(year) )

)

\* optional relaxation for calibration and debugging

%SLACK\_LAND\_SCEN\_UP% + SLACK\_LAND\_SCEN\_UP(node,land\_scenario,year)

;

\*\*\*

\* .. \_equation\_dynamic\_land\_scen\_constraint\_lo:

\*

\* Equation DYNAMIC\_LAND\_SCEN\_CONSTRAINT\_LO

\* """"""""""""""""""""""""""""""""""""""""

\*

\* .. math::

\* \text{LAND}\_{n,s,y}

\* \geq & - \text{initial\_land\_scen\_lo}\_{n,s,y}

\* \cdot \frac{ \Big( 1 + \text{growth\_land\_scen\_lo}\_{n,s,y} \Big)^{|y|} - 1 }

\* { \text{growth\_land\_scen\_lo}\_{n,s,y} } \\

\* & + \big( \text{LAND}\_{n,s,y-1} + \text{historical\_land}\_{n,s,y-1} \big)

\* \cdot \Big( 1 + \text{growth\_land\_scen\_lo}\_{n,s,y} \Big)^{|y|}

\*

\*\*\*

DYNAMIC\_LAND\_SCEN\_CONSTRAINT\_LO(node,land\_scenario,year)$( map\_land(node,land\_scenario,year)

AND is\_dynamic\_land\_scen\_lo(node,land\_scenario,year) )..

\* share of land scenario in

LAND(node,land\_scenario,year) =G=

\* initial 'new' land used for that type (compounded over the duration of the period)

- initial\_land\_scen\_lo(node,land\_scenario,year) \* (

( ( POWER( 1 + growth\_land\_scen\_lo(node,land\_scenario,year) , duration\_period(year) ) - 1 )

/ growth\_land\_scen\_lo(node,land\_scenario,year) )$( growth\_land\_scen\_lo(node,land\_scenario,year) )

+ ( duration\_period(year) )$( NOT growth\_land\_scen\_lo(node,land\_scenario,year) )

)

\* reduction of land scenario from previous period

+ SUM((year\_all2)$( seq\_period(year\_all2,year) ),

( LAND(node,land\_scenario,year\_all2)$( model\_horizon(year\_all2) )

+ historical\_land(node,land\_scenario,year\_all2) )

\* POWER( 1 + growth\_land\_scen\_lo(node,land\_scenario,year) , duration\_period(year) )

)

\* optional relaxation for calibration and debugging

%SLACK\_LAND\_SCEN\_LO% - SLACK\_LAND\_SCEN\_LO(node,land\_scenario,year)

;

\*\*\*

\* These constraints enforces upper and lower bounds on the change rate per land type

\* determined as a linear combination of land use scenarios.

\*

\* .. \_equation\_dynamic\_land\_type\_constraint\_up:

\*

\* Equation DYNAMIC\_LAND\_TYPE\_CONSTRAINT\_UP

\* """"""""""""""""""""""""""""""""""""""""

\*

\* .. math::

\* \sum\_{s \in S} \text{land\_use}\_{n,s,y,u} &\cdot \text{LAND}\_{n,s,y}

\* \leq \text{initial\_land\_up}\_{n,y,u}

\* \cdot \frac{ \Big( 1 + \text{growth\_land\_up}\_{n,y,u} \Big)^{|y|} - 1 }

\* { \text{growth\_land\_up}\_{n,y,u} } \\

\* & + \Big( \sum\_{s \in S} \big( \text{land\_use}\_{n,s,y-1,u}

\* + \text{dynamic\_land\_up}\_{n,s,y-1,u} \big) \\

\* & \quad \quad \cdot \big( \text{LAND}\_{n,s,y-1} + \text{historical\_land}\_{n,s,y-1} \big) \Big) \\

\* & \quad \cdot \Big( 1 + \text{growth\_land\_up}\_{n,y,u} \Big)^{|y|}

\*

\*\*\*

DYNAMIC\_LAND\_TYPE\_CONSTRAINT\_UP(node,year,land\_type)$( is\_dynamic\_land\_up(node,year,land\_type) )..

\* amount of land assigned to specific type in current period

SUM(land\_scenario$( map\_land(node,land\_scenario,year) ),

land\_use(node,land\_scenario,year,land\_type) \* LAND(node,land\_scenario,year) ) =L=

\* initial 'new' land used for that type (compounded over the duration of the period)

initial\_land\_up(node,year,land\_type) \* (

( ( POWER( 1 + growth\_land\_up(node,year,land\_type) , duration\_period(year) ) - 1 )

/ growth\_land\_up(node,year,land\_type) )$( growth\_land\_up(node,year,land\_type) )

+ ( duration\_period(year) )$( NOT growth\_land\_up(node,year,land\_type) )

)

\* expansion of previously used land of this type from previous period and upper bound on land use transformation

+ SUM((year\_all2)$( seq\_period(year\_all2,year) ),

SUM(land\_scenario$( map\_land(node,land\_scenario,year) ),

( land\_use(node,land\_scenario,year\_all2,land\_type)

+ dynamic\_land\_up(node,land\_scenario,year\_all2,land\_type) )

\* ( LAND(node,land\_scenario,year\_all2)$( model\_horizon(year\_all2) )

+ historical\_land(node,land\_scenario,year\_all2) )

\* POWER( 1 + growth\_land\_up(node,year,land\_type) , duration\_period(year) )

)

)

\* optional relaxation for calibration and debugging

%SLACK\_LAND\_TYPE\_UP% + SLACK\_LAND\_TYPE\_UP(node,year,land\_type)

;

\*\*\*

\* .. \_equation\_dynamic\_land\_type\_constraint\_lo:

\*

\* Equation DYNAMIC\_LAND\_TYPE\_CONSTRAINT\_LO

\* """"""""""""""""""""""""""""""""""""""""

\*

\* .. math::

\* \sum\_{s \in S} \text{land\_use}\_{n,s,y,u} &\cdot \text{LAND}\_{n,s,y}

\* \geq - \text{initial\_land\_lo}\_{n,y,u}

\* \cdot \frac{ \Big( 1 + \text{growth\_land\_lo}\_{n,y,u} \Big)^{|y|} - 1 }

\* { \text{growth\_land\_lo}\_{n,y,u} } \\

\* & + \Big( \sum\_{s \in S} \big( \text{land\_use}\_{n,s,y-1,u}

\* + \text{dynamic\_land\_lo}\_{n,s,y-1,u} \big) \\

\* & \quad \quad \cdot \big( \text{LAND}\_{n,s,y-1} + \text{historical\_land}\_{n,s,y-1} \big) \Big) \\

\* & \quad \cdot \Big( 1 + \text{growth\_land\_lo}\_{n,y,u} \Big)^{|y|}

\*

\*\*\*

DYNAMIC\_LAND\_TYPE\_CONSTRAINT\_LO(node,year,land\_type)$( is\_dynamic\_land\_lo(node,year,land\_type) )..

\* amount of land assigned to specific type in current period

SUM(land\_scenario$( map\_land(node,land\_scenario,year) ),

land\_use(node,land\_scenario,year,land\_type) \* LAND(node,land\_scenario,year) ) =G=

\* initial 'new' land used for that type (compounded over the duration of the period)

- initial\_land\_lo(node,year,land\_type) \* (

( ( POWER( 1 + growth\_land\_up(node,year,land\_type) , duration\_period(year) ) - 1 )

/ growth\_land\_lo(node,year,land\_type) )$( growth\_land\_lo(node,year,land\_type) )

+ ( duration\_period(year) )$( NOT growth\_land\_lo(node,year,land\_type) )

)

\* expansion of previously used land of this type from previous period and lower bound on land use transformation

+ SUM((year\_all2)$( seq\_period(year\_all2,year) ),

SUM(land\_scenario$( map\_land(node,land\_scenario,year) ),

( land\_use(node,land\_scenario,year\_all2,land\_type)

+ dynamic\_land\_lo(node,land\_scenario,year\_all2,land\_type) )

\* ( LAND(node,land\_scenario,year\_all2)$( model\_horizon(year\_all2) )

+ historical\_land(node,land\_scenario,year\_all2) )

\* POWER( 1 + growth\_land\_lo(node,year,land\_type) , duration\_period(year) )

)

)

\* optional relaxation for calibration and debugging

%SLACK\_LAND\_TYPE\_LO% - SLACK\_LAND\_TYPE\_LO(node,year,land\_type)

;

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_section\_of\_generic\_relations:

\*

\* Section of generic relations (linear constraints)

\* -------------------------------------------------

\*

\* This feature is intended for development and testing only - all new features should be implemented

\* as specific new mathematical formulations and associated sets & parameters!

\*

\* Auxiliary variable for left-hand side

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* .. \_equation\_relation\_equivalence:

\*

\* Equation RELATION\_EQUIVALENCE

\* """""""""""""""""""""""""""""

\* .. math::

\* \text{REL}\_{r,n,y} = \sum\_{t} \Bigg(

\* & \ \text{relation\_new\_capacity}\_{r,n,y,t} \cdot \text{CAP\_NEW}\_{n,t,y} \\[4 pt]

\* & + \text{relation\_total\_capacity}\_{r,n,y,t} \cdot \sum\_{y^V \leq y} \ \text{CAP}\_{n,t,y^V,y} \\

\* & + \sum\_{n^L,y',m,h} \ \text{relation\_activity}\_{r,n,y,n^L,t,y',m} \\

\* & \quad \quad \cdot \Big( \sum\_{y^V \leq y'} \text{ACT}\_{n^L,t,y^V,y',m,h}

\* + \text{historical\_activity}\_{n^L,t,y',m,h} \Big) \Bigg)

\*

\* The parameter :math:`\text{historical\_new\_capacity}\_{r,n,y}` is not included here, because relations can only be active

\* in periods included in the model horizon and there is no "writing" of capacity relation factors across periods.

\*\*\*

RELATION\_EQUIVALENCE(relation,node,year)..

REL(relation,node,year)

=E=

SUM(tec,

( relation\_new\_capacity(relation,node,year,tec) \* CAP\_NEW(node,tec,year)

+ relation\_total\_capacity(relation,node,year,tec)

\* SUM(vintage$( map\_tec\_lifetime(node,tec,vintage,year) ), CAP(node,tec,vintage,year) )

)$( inv\_tec(tec) )

+ SUM((location,year\_all2,mode,time)$( map\_tec\_act(location,tec,year\_all2,mode,time) ),

relation\_activity(relation,node,year,location,tec,year\_all2,mode)

\* ( SUM(vintage$( map\_tec\_lifetime(location,tec,vintage,year\_all2) ),

ACT(location,tec,vintage,year\_all2,mode,time) )

+ historical\_activity(location,tec,year\_all2,mode,time) )

)

) ;

\*\*\*

\* Upper and lower bounds on user-defined relations

\* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

\*

\* .. \_equation\_relation\_constraint\_up:

\*

\* Equation RELATION\_CONSTRAINT\_UP

\* """""""""""""""""""""""""""""""

\* .. math::

\* \text{REL}\_{r,n,y} \leq \text{relation\_upper}\_{r,n,y}

\*\*\*

RELATION\_CONSTRAINT\_UP(relation,node,year)$( is\_relation\_upper(relation,node,year) )..

REL(relation,node,year)

%SLACK\_RELATION\_BOUND\_UP% - SLACK\_RELATION\_BOUND\_UP(relation,node,year)

=L= relation\_upper(relation,node,year) ;

\*\*\*

\* .. \_equation\_relation\_constraint\_lo:

\*

\* Equation RELATION\_CONSTRAINT\_LO

\* """""""""""""""""""""""""""""""

\* .. math::

\* \text{REL}\_{r,n,y} \geq \text{relation\_lower}\_{r,n,y}

\*\*\*

RELATION\_CONSTRAINT\_LO(relation,node,year)$( is\_relation\_lower(relation,node,year) )..

REL(relation,node,year)

%SLACK\_RELATION\_BOUND\_LO% + SLACK\_RELATION\_BOUND\_LO(relation,node,year)

=G= relation\_lower(relation,node,year) ;

\*----------------------------------------------------------------------------------------------------------------------\*

\*\*\*

\* .. \_gams-storage:

\*

\* Storage section

\* ---------------

\*

\* MESSAGEix offers a set of equations to represent a wide range of storage solutions flexibly.

\* Storage solutions are modeled as "technologies" that can be used to store a "commodity" (e.g., water, heat, electricity, etc.)

\* and shift it over sub-annual time slices within one model period. The storage solution presented here has three

\* distinct parts: (i) Charger: a technology for charging a commodity to the storage container,

\* for example, a pump in a pumped hydropower storage (PHS) plant. (ii) Discharger: a technology

\* to convert the stored commodity to the output commodity, e.g., a turbine in PHS.

\* (iii) Storage container: a device for storing a commodity over time, such as a water reservoir in PHS.

\* If desired, the user can combine charger and discharger parts into one technology, using two different "modes" of operation

\* for that technology like turbo-machinery in PHS. This way the capacity related information, like investment cost, lifetime, capacity factor, etc.,

\* will be defined only for one technology (i.e., charger-discharger), as opposed to modeling these two parts separately.

\*

\* .. figure:: ../../\_static/storage.png

\*

\* Storage equations

\* ^^^^^^^^^^^^^^^^^

\* The content of storage device depends on three factors: charge or discharge in

\* one time slice (represented by `Equation STORAGE\_CHANGE`\_), linked to the state of charge in the previous

\* time slice and storage losses between these two consecutive time slices (represented by `Equation STORAGE\_BALANCE`\_).

\* Moreover, the storage device can be optionally filled with an initial value as percentage of its capacity (see more details under `Equation STORAGE\_BALANCE\_INIT`\_).

\* Another option is to link a commodity for maintaining the operation of storage device over time (see `Equation STORAGE\_INPUT`\_).

\*

\* .. \_equation\_storage\_change:

\*

\* Equation STORAGE\_CHANGE

\* """""""""""""""""""""""

\* This equation shows the change in the content of the storage container in each

\* sub-annual time slice. This change is based on the activity of charger and discharger

\* technologies connected to that storage container. The notation :math:`S^{\text{storage}}`

\* represents the mapping set `map\_tec\_storage` denoting charger-discharger

\* technologies connected to a specific storage container in a specific node and

\* storage level. Where:

\*

\* - :math:`t^{C}` is a charging technology and :math:`t^{D}` is the corresponding discharger.

\* - :math:`h-1` is the time slice prior to :math:`h`.

\* - :math:`l^{T}` is `lvl\_temporal`, i.e., the temporal level at which storage is operating

\* - :math:`m^{S}` is `mode` of operation for storage container technology

\* .. math::

\* \text{STORAGE\_CHARGE}\_{n,t,m^s,l,c,y,h} =

\* \sum\_{\substack{n^L,m,h-1 \\ y^V \leq y, (n,t^C,t,l,y) \sim S^{\text{storage}}}} \text{output}\_{n^L,t^C,y^V,y,m,n,c,l,h-1,h}

\* \cdot & \text{ACT}\_{n^L,t^C,y^V,y,m,h-1} \\

\* - \sum\_{\substack{n^L,m,c,h-1 \\ y^V \leq y, (n,t^D,t,l,y) \sim S^{\text{storage}}}} \text{input}\_{n^L,t^D,y^V,y,m,n,c,l,h-1,h}

\* \cdot \text{ACT}\_{n^L,t^D,y^V,y,m,h-1} \quad \forall \ t \in T^{\text{STOR}}, & \forall \ l \in L^{\text{STOR}}

\*\*\*

STORAGE\_CHANGE(node,storage\_tec,mode,level\_storage,commodity,year,time)$sum(

(tec,mode2,lvl\_temporal), map\_tec\_storage(node,tec,mode2,storage\_tec,mode,level\_storage,commodity,lvl\_temporal) ) ..

\* change in the content of storage in the examined time slice

STORAGE\_CHARGE(node,storage\_tec,mode,level\_storage,commodity,year,time) =E=

\* increase in the content of storage due to the activity of charging technologies

SUM( (location,vintage,tec,mode2,time2,time3,lvl\_temporal)$(

map\_tec\_lifetime(node,tec,vintage,year) AND map\_temporal\_hierarchy(lvl\_temporal,time,time3

)$map\_tec\_storage(node,tec,mode2,storage\_tec,mode,level\_storage,commodity,lvl\_temporal) ),

output(location,tec,vintage,year,mode2,node,commodity,level\_storage,time2,time)

\* duration\_time\_rel(time,time2) \* ACT(location,tec,vintage,year,mode2,time2) )

\* decrease in the content of storage due to the activity of discharging technologies

- SUM( (location,vintage,tec,mode2,time2,time3,lvl\_temporal)$(

map\_tec\_lifetime(node,tec,vintage,year) AND map\_temporal\_hierarchy(lvl\_temporal,time,time3

)$map\_tec\_storage(node,tec,mode2,storage\_tec,mode,level\_storage,commodity,lvl\_temporal) ),

input(location,tec,vintage,year,mode2,node,commodity,level\_storage,time2,time)

\* duration\_time\_rel(time,time2) \* ACT(location,tec,vintage,year,mode2,time2) );

\*\*\*

\* .. \_equation\_storage\_balance:

\*

\* Equation STORAGE\_BALANCE

\* """"""""""""""""""""""""

\*

\* This equation ensures the commodity balance of storage technologies, where the commodity is shifted between sub-annual

\* time slices within a model period. If the state of charge of storage is set exogenously in one time slice through

\* :math:`\storageinitial\_{ntlcyh}`, the content from the previous time slice is not carried over to this time slice.

\*

\* .. math::

\* \STORAGE\_{ntmlcyh} =\ & \STORAGECHARGE\_{ntmlcyh} \\

\* & + \STORAGE\_{ntmlcy(h-1)} \cdot (1 - \storageselfdischarge\_{ntmly(h-1)}) \\

\* \forall\ & t \in T^{\text{STOR}}, l \in L^{\text{STOR}}, \storageinitial\_{ntmlcyh} = 0

\*\*\*

STORAGE\_BALANCE(node,storage\_tec,mode,level,commodity,year,time2,lvl\_temporal)$ (

SUM((tec,mode2), map\_tec\_storage(node,tec,mode2,storage\_tec,mode,level,commodity,lvl\_temporal) )

\* AND NOT storage\_initial(node,storage\_tec,mode,level,commodity,year,time2)

)..

\* Showing the the state of charge of storage at each time slice

STORAGE(node,storage\_tec,mode,level,commodity,year,time2) =E=

\* change in the content of storage in the examined time slice

+ STORAGE\_CHARGE(node,storage\_tec,mode,level,commodity,year,time2)

\* storage content in the previous subannual time slice

+ SUM(time$map\_time\_period(year,lvl\_temporal,time,time2),

STORAGE(node,storage\_tec,mode,level,commodity,year,time)

\* considering storage self-discharge losses due to keeping the storage media between two subannual time slices

\* (1 - storage\_self\_discharge(node,storage\_tec,mode,level,commodity,year,time) ) ) ;

\*\*\*

\* .. \_equation\_storage\_balance\_init:

\*

\* Equation STORAGE\_BALANCE\_INIT

\* """""""""""""""""""""""""""""

\*

\* Where :math:`\storageinitial\_{ntlyh}` has a non-zero value, this equation ensures that the amount of commodity stored

\* at the end of a sub-annual time slice is equal or greater than the initialized content of storage in the following time slice.

\* The values in parameter :math:`\storageinitial\_{ntlyh}` are percentages showing

\* a fraction of installed capacity of storage device (container) that can be filled initially.

\*

\* .. math::

\* \STORAGE\_{ntmlcy(h-1)} \geq & \storageinitial\_{ntmlcyh} \cdot \text{duration\_time}\_{h} \cdot \text{capacity\_factor}\_{n,t,y^V,y,h} \cdot \text{CAP}\_{n,t,y^V,y} \\

\* \quad \forall \ t \ \in \ T^{\text{INV}}, \forall\ & \storageinitial\_{ntmlcyh} \neq 0

\*\*\*

STORAGE\_BALANCE\_INIT(node,storage\_tec,mode,level,commodity,year,time,time2)$ (

SUM((tec,mode2,lvl\_temporal), map\_tec\_storage(node,tec,mode2,storage\_tec,mode,level,commodity,lvl\_temporal)

AND map\_time\_period(year,lvl\_temporal,time,time2) )

AND storage\_initial(node,storage\_tec,mode,level,commodity,year,time2) )..

\* Showing the state of charge of storage at a time slice prior to a time slice that has initial storage content

STORAGE(node,storage\_tec,mode,level,commodity,year,time) =G=

\* Initial content of storage in the examined time slice as a percentage multiplier in available capacity of storage

storage\_initial(node,storage\_tec,mode,level,commodity,year,time2)

\* SUM(vintage$( map\_tec\_lifetime(node,storage\_tec,vintage,year) ), capacity\_factor(node,storage\_tec,vintage,year,time2)

\* CAP(node,storage\_tec,vintage,year) / duration\_time(time2) )

;

\*\*\*

\* .. \_equation\_storage\_input:

\*

\* Equation STORAGE\_INPUT

\* """"""""""""""""""""""""""""

\*

\* This equation links :math:`\STORAGE` to an input commodity to maintain the activity (:math:`\ACT`) of each active storage \*container\* technology

\* :math:`t`. This input commodity is distinct from the stored commodity. For example, in a pumped hydro storage solution, a user can link heating

\* for keeping the stored water warm. In this case, the input commodity is not a function of charge or discharge, but the amount of stored media in the container over time.

\* Therefore, the input commodity specified here is distinct from the one stored and discharged by \*(dis)charge\* technologies :math:`t^C,t^D` appearing in

\* :ref:`equation\_storage\_change`.

\*

\* .. math::

\* \STORAGE\_{ntmlcy^Ah} =\ & \sum\_{\{n^Ly^Vh^O \vert K\}} \durationtimerel\_{hh^O} \cdot \ACT\_{n^Lty^Vy^Amh^O} \\

\* \forall\ & n,t,l,c,m,y^A,h \vert t \in T^{\text{STOR}} \\

\* K:\ & \\text{input}\_{n^Lty^Vy^Amn^Oclhh^O} \neq 0

\*

\*\*\*

STORAGE\_INPUT(node,storage\_tec,level,commodity,level\_storage,commodity2,mode,year,time)$

( map\_time\_commodity\_storage(node,storage\_tec,level,commodity,mode,year,time) AND

SUM( (tec,mode2,lvl\_temporal), map\_tec\_storage(node,tec,mode2,storage\_tec,mode,level\_storage,commodity2,lvl\_temporal) ) ) ..

\* Connecting an input commodity to maintain the operation of storage container over time (optional)

STORAGE(node,storage\_tec,mode,level\_storage,commodity2,year,time) =E=

SUM( (location,vintage,time2)$(map\_tec\_lifetime(node,storage\_tec,vintage,year)$(

input(location,storage\_tec,vintage,year,mode,node,commodity,level,time,time2) ) ),

duration\_time\_rel(time,time2) \* ACT(location,storage\_tec,vintage,year,mode,time) )

;

\*----------------------------------------------------------------------------------------------------------------------\*

\* model statements \*

\*----------------------------------------------------------------------------------------------------------------------\*

Model MESSAGE\_LP / all / ;

MESSAGE\_LP.holdfixed = 1 ;

MESSAGE\_LP.optfile = 1 ;

MESSAGE\_LP.optcr = 0 ;