IRENA FlexTool

Methodology

IRENA 2020
1. Major assumptions
2. Building blocks
3. Investment run
4. Costs
5. More information
Major Assumptions
1. Perfect foresight

2. Linear model

3. Formulated with MathProg, solved with an open solver

4. Cost minimising
Perfect vs. uncertain forecast

- FlexTool has **no uncertainty** – it knows what will happen and solves a perfect dispatch.

- In reality:
  - Uncertainty forces transmission system operators to **commit more resources than needed**
  - Uncertainty means, *e.g.*, **charging and discharging of energy storages cannot be fully optimal**

- FlexTool commits:
  - Resources sufficient for perfect dispatch
  - Single upward reserve that can cover variability within model time step as well as contingencies

- **Reserves that are used to mitigate forecast errors** (longer than model time step) can be used during dispatch – FlexTool should not keep those reserved, because then they could not be used.

- **Consequences:**
  - Value of storages can be higher than can be really achieved
  - Slightly smaller costs, since the model can manage with less online units
Linear vs. mixed integer modelling

- IRENA FlexTool does **not use integer** decision variables (e.g., on/off)
  - Instead, startups have been linearised (unit can start also partially)

- It matters when:
  - Running operational optimisation for **actual system operation**
  - Comparing technologies that have distinct **start-up** characteristics (e.g., gas turbine vs. gas engine)
  - **Small systems** with only few units
  - **Large individual units** in comparison to system size
  - System **stability** requires some units to be online (can be endogenously forced in FlexTool)

- It matters a little in:
  - **High level** long-term planning
  - Systems with **flexible generation** portfolios
Solving the problem

- FlexTool uses **GNU MathProg** language to formulate the optimisation problem to a separate solver
- **Solver minimises** (or maximises) a system of linear equations
  - flexModel.mod is a MathProg file and contains the equations
- For linear problems, **open source solvers perform quite well**
  - Especially Clp (Coin-or linear programming) used by FlexTool
- In mixed-integer problems commercial solvers are orders of magnitude better than open source solvers
- Efficient solution algorithms are based on **primal simplex, dual simplex, and interior point methods**
- Genetic algorithms, AI, particle swarm, annealing, etc. methods also exist, but are much slower
- Linear problems are typically **solved to global optimum** (integer problems are typically not – defined by solution gap)
**Objective function**

**Cost minimisation**

- **Operation**
  - + fixed operation and maintenance costs
  - + variable operation and maintenance costs
  - + fuel costs of units
  - + CO2 emission costs
  - + start-up costs
  - + penalty cost for loss of load
  - + penalty cost for insufficient upward reserves
  - + penalty cost for insufficient capacity margin
  - + penalty cost for curtailment of VRE
  - + penalty cost for insufficient inertia
  - + unit investment costs
  - + storage investment costs
  - + transmission line investment costs

- **Penalties**
  - + unit investment costs
  - + storage investment costs
  - + transmission line investment costs

**Capacity**

- =
- + pre-existing capacity [units: capacity]
- + forced new capacity [units: invested_capacity]
- + invested new capacity [v_invest | v_investTransfer]

**Investment**

- + unit investment costs
- + storage investment costs
- + transmission line investment costs

**Operation Penalties**

- **Capacity**
  - =
  - + pre-existing capacity [units: capacity]
  - + forced new capacity [units: invested_capacity]
  - + invested new capacity [v_invest | v_investTransfer]

- **Objective function**

- **Capacity**
  - =
  - + pre-existing capacity [units: capacity]
  - + forced new capacity [units: invested_capacity]
  - + invested new capacity [v_invest | v_investTransfer]
Building Blocks
1. Grid

2. Node
   - Demand
   - Reserve requirements
   - Non-synchronous limit
   - Inertia limit
   - Transfer between nodes

3. Unit
   - Defining unit category
   - Upward limit
   - Online variable
   - Ramp constraint
   - Advanced features

4. Timestep
gridNodes

- Grids, nodes, and nodeGroups are defined in
  - nodeGroup
  - gridNode sheets
One of the basic building blocks, 1/2

- Grids are used to label different grids (e.g., electricity and natural gas)
  - No equations or constraints related only to grid
  - Used when presenting results

- Combination of gridNodes used in defining the model
One of the basic building blocks, 2/2

- Grids and nodes are used to model **characteristics of geographical areas**, including:
  - Demand,
  - Reserve requirements,
  - Non-synchronous limit
  - Inertia limit

- One node can be part of only one grid
  - They can cover the same geographical area, but need different names

- Nodes can be modelled individually or as a group of nodes

- Transfers between nodes allows sharing generation and reserves
Demand of each node

- Net demand in each node is a sum of demand and import
  - Annual sums are defined in **gridNode** sheet
  - Hourly values are calculated based on normalised time series

### Normalised demand of one hour:

Demand (t0001) = \( \frac{\text{ts}_{\text{energy}}(t0001)}{\text{sum}_t(\text{ts}_{\text{energy}})} \times \text{annual demand} \)
Energy balance equation

\[\text{generation from non-VRE units } [v_{\text{gen}}] + \text{generation from VRE units } [ts_{\text{cf}} \times \text{capacity}] - \text{curtailment of VRE units } [v_{\text{curtail}}] + \text{imports* } [v_{\text{transfer}} \text{ and/or } ts_{\text{import}} \text{: time series}] + \text{energy conversions to the node* } [v_{\text{convert}}] + \text{discharging of storages* } [v_{\text{gen}}] + \text{loss of load } [v_{\text{slack}}] + \text{energy demand } [v_{\text{charge}} \text{ and/or } ts_{\text{demand}} \text{: time series}] + \text{exports* } [v_{\text{transfer}} \text{ and/or } ts_{\text{import}} \text{: time series}] + \text{energy conversions from the node } [v_{\text{convert}}] + \text{charging of storages } [v_{\text{charge}}]\]

* can contain losses
Reserve requirements: Static reserves

- **Static reserves** are predefined time series that need to be activated
  - For single node or node group

- If static reserves are activated, every node and node_group requires own matching time series

- **Dynamic reserves** are calculated based on generating units (see next slide), but these need also to be activated
Reserve requirements: Dynamic reserves

- **Dynamic reserve** can be defined for units
  - Reserve increase ratio in unit sheet
  - By default used with VRE generation
  - When unit generates, it increases the reserve need
  - *E.g.*, 10 MW of wind power is defined to need 1 MW reserves

- **Dynamic reserve** is not additional to static, model checks every hour (stricter requirements)
Single node reserve requirement:
+ sum of reserves from the units in the node \( \sum (v_{\text{reserve}}(\text{node}, \text{unit}, t)) \)
+ reserve from VRE units \( v_{\text{reserveVRE}}(\text{node}, t) \)
+ lack of reserve penalty variable \( v_{\text{reserveSlack}}(\text{node}, t) \)
\( \geq \)
+ reserve requirement for the node

Static
- [gridNode: use ts_reserve]
- [ts_reserve_node: time series]

Dynamic:
- [gridNode: use dynamic reserve]
- Sum (units: \( v_{\text{gen}}(\text{node}, \text{unit}, t) \)) \( \times \) [units: reserve_increase_ratio]
Upward reserve requirement: groups of nodes

NodeGroup reserve requirement:
+ sum of reserves from the units in the nodes
+ reserve from VRE units
+ lack of reserve penalty variable
 >=
+ reserve requirement for the node

NodeGroup 'reserveNodes' = Union (Node A, Node B, Node C)

Static
- [nodeGroup: use ts_reserve]
- [ts_reserve_nodeGroup: time series]

Dynamic:
- [nodeGroup: use dynamic reserve]
- Sum (units: \[v_gen(node, unit, t)\] × [units: reserve_increase_ratio]
Further reserve limits

**VRE upward reserve:**
+ reserve from VRE \([v\_reserveVRE]\) <=
+ VRE curtailment \([v\_curtail]\) × reserve contribution [unit_type: max_reserve (0-1)]
Maximum non-synchronous share, 1/2

- **Maximum non-synchronous shares** activated in master sheet
- Defined for single node or node group
- Units are flagged synchronous (0) or non-synchronous (1) in unitType sheet

### Master Sheet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>co2_cost</td>
<td>10</td>
</tr>
<tr>
<td>loss_of_load_penalty</td>
<td>10000</td>
</tr>
<tr>
<td>loss_of_reserves_penalty</td>
<td>20000</td>
</tr>
<tr>
<td>lack_of_inertia_penalty</td>
<td>30000</td>
</tr>
<tr>
<td>curtailment_penalty</td>
<td>20</td>
</tr>
<tr>
<td>lack_of_capacity_penalty</td>
<td>5000</td>
</tr>
<tr>
<td>time_in_years</td>
<td>1.000</td>
</tr>
<tr>
<td>time_period_duration</td>
<td>60</td>
</tr>
<tr>
<td>reserve_duration</td>
<td>0.50</td>
</tr>
<tr>
<td>use_capacity_margin</td>
<td>1</td>
</tr>
<tr>
<td>use_online</td>
<td>1</td>
</tr>
<tr>
<td>use_ramps</td>
<td>0</td>
</tr>
<tr>
<td>use_non_synchronous</td>
<td>1</td>
</tr>
<tr>
<td>use_inertia_limit</td>
<td>0</td>
</tr>
<tr>
<td>mode_invest</td>
<td>0</td>
</tr>
<tr>
<td>mode_dispatch</td>
<td>1</td>
</tr>
<tr>
<td>print_duration</td>
<td>0</td>
</tr>
<tr>
<td>print_durationRamp</td>
<td>0</td>
</tr>
<tr>
<td>print_unit_results</td>
<td>0</td>
</tr>
</tbody>
</table>

### UnitType Sheet

- ST_coal
- Engine_gas
- CC_oil
- ST_bio
- Hydro_RES
- Hydro_ROR
- wind
- PV
- battery

### GridNode Sheet

- Grid node
- Node A
- Node B
- Node C
- Node D

### NodeGroup Sheet

- Mainland
- 0.80
Maximum non-synchronous share, 2/2

+ non-synchronous generation $[v_{\text{gen}}]$
+ non-synchronous VRE generation $[\text{ts\_cf: time series}] \times \text{capacity}$
  - curtailment of VRE units $[v_{\text{curtail}}]$
+ non-synchronous conversion $[v_{\text{convert}}]$
+ HVDC transfer into the node $[v_{\text{transfer}}]$
+ discharging of non-synch. storages $[v_{\text{gen}}]$
\leq\)
maximum non synchronous share $[\text{nodeGroup: non synchronous share}]$
\
\times (\)
  + energy demand $[v_{\text{charge}} \text{ and/or ts\_demand: time series}]$
  + exports - imports $[v_{\text{transfer}} \text{ and/or ts\_import: time series}]$
  + energy conversions from the node $[v_{\text{convert}}]$
  + charging of storages $[v_{\text{charge}}]$
  - loss of load $[v_{\text{slack}}]$
)
Minimum inertia limit, 1/2

- **Minimum inertia limit** needs to be activated from master sheet in input data (use_inertia_limit = 1)
- Defined only for node groups
- Inertia constant for each unit defined in unitType (MWs/MW)

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>co2_cost</td>
<td>10</td>
</tr>
<tr>
<td>loss_of_load_penalty</td>
<td>10000</td>
</tr>
<tr>
<td>loss_of_reserves_penalty</td>
<td>20000</td>
</tr>
<tr>
<td>lack_of_inertia_penalty</td>
<td>30000</td>
</tr>
<tr>
<td>curtailment_penalty</td>
<td>20</td>
</tr>
<tr>
<td>lack_of_capacity_penalty</td>
<td>5000</td>
</tr>
<tr>
<td>time_in_years</td>
<td>1000</td>
</tr>
<tr>
<td>time_period_duration</td>
<td>60</td>
</tr>
<tr>
<td>reserve_duration</td>
<td>0.50</td>
</tr>
<tr>
<td>use_capacity_margin</td>
<td>1</td>
</tr>
<tr>
<td>use_online</td>
<td>1</td>
</tr>
<tr>
<td>use_ramps</td>
<td>0</td>
</tr>
<tr>
<td>use_non_synchronous</td>
<td>1</td>
</tr>
<tr>
<td>use_inertia_limit</td>
<td>0</td>
</tr>
<tr>
<td>nodeGroup_sheet</td>
<td></td>
</tr>
</tbody>
</table>

- **nodeGroup sheet**: Table showing node group and inertia limit
- **unitType sheet**: Table showing unit type and inertia constant

Master sheet
Minimum inertia limit, 2/2

+ online capacity of conventional online units
  × inertia constant
+ generation of conventional units without online
  × inertia constant
+ generation from VRE units
  × inertia constant
+ lack of inertia penalty variable
>=
+ inertia limit in MWs

[v_online] [unit_type: inertia constant]
[v_gen] [unit_type: inertia constant]
[ts Cf: time series] × capacity [unit_type: inertia constant]
[v_inertiaSlack]
[nodeGroup: inertia limit]

Can be applied only to group of nodes [nodeGroup]
Transfers, 1/2

- **Transfers** between nodes are defined in nodeNode sheet
  - Both nodes have to be from the same grid
  - Existing transfer links can have different capacity to different direction
  - Future investments will always have equal capacity to both directions
Transfers, 2/2

- **Transfer with losses** requires at least two variables
  - A linear equation with ‘loss x transfer’ would mean that in the other direction the loss is actually a gain
- The loss can be used to make the *model ‘leak’*
  - Instead of curtailing VRE, the model can dissipate energy by transferring in two directions at once
  - Can be controlled only with a binary variable (not allowed in FlexTool)
- Hence, **three variables**: transfer, transfer rightward and transfer leftward
  - Transfer does not contain loss
  - Transfer rightward allows losses and helps to limit the leakage
  - Transfer leftward helps to limit the leakage further

![Diagram](image)
Transfer losses (balance equation)

- **Two nodes**: left and right (a node-node link is established with one direction only)

- When transferring from **left to right**:
  - Left node: transfer deducted from node balance
  - Right node: transfer minus loss added to the node balance

- When transferring from **right to left**:
  - Left node: transfer minus loss added to the node balance
  - Right node: transfer deducted from node balance

Balance leftward node

\[
\text{Balance leftward node}
= + \text{transfer} \quad [v_{\text{transfer}}(g,n,n_{\text{right}},t)]
- \text{loss} \times (1 - \text{[nodeNode: loss]})
+ \text{rightward transfer} \quad [v_{\text{transferRightward}}(g,n,n_{\text{right}},t)]
\]

Balance rightward node

\[
\text{Balance rightward node}
= + \text{transfer} \quad [v_{\text{transfer}}(g,n_{\text{left}},n,t)]
- \text{rightward transfer} \quad [v_{\text{transferRightward}}(g,n_{\text{left}},n,t)]
\times \text{loss} \quad \times \text{[nodeNode: loss]}
\]

When rightward: CANCELS EACH OTHER

When leftward: ZERO
**Transfer constraints**

**Tie transfers together**

\[
\begin{align*}
\text{transfer} & \quad [v_{\text{transfer}}(g,n_{\text{left}},n_{\text{right}},t)] \\
= & \\
\text{rightward transfer} & \quad [v_{\text{transferRightward}}(g,n_{\text{left}},n_{\text{right}},t)] \\
- \text{leftward transfer} & \quad [v_{\text{transferLeftward}}(g,n_{\text{left}},n_{\text{right}},t)]
\end{align*}
\]

**Limit rightward transfer**

\[
\begin{align*}
\text{transfer} & \quad [v_{\text{transfer}}(g,n_{\text{left}},n_{\text{right}},t)] \\
\leq & \\
\text{capacity} & \quad \text{[see orange box]}
\end{align*}
\]

**Limit rightward transfer again**

\[
\begin{align*}
\text{transfer rightward} & \quad [v_{\text{transferRightward}}(g,n_{\text{left}},n_{\text{right}},t)] \\
\leq & \\
\text{capacity} & \quad \text{[see orange box]}
\end{align*}
\]

**Rightward capacity**

\[
\begin{align*}
\text{capacity} & \quad = \\
\text{pre-existing leftward transfer capacity} & \quad \text{[nodeNode: cap_rightward]} \\
\text{forced new capacity} & \quad \text{[nodeNode: invested_capacity]} \\
\text{invested new capacity} & \quad \text{[v_investTransfer]}
\end{align*}
\]

And same for leftward transfers!!!
Concluding remarks on transfers

- Transfer with losses works, but the model can leak

- In normal circumstances, the model does not leak (why waste energy?), but the model can use leakage instead of VRE curtailment (if curtailment has a penalty cost)
  - E.g.,
    - Rightward transfer = 100 MW
    - Leftward transfer = -100 MW
    - → Transfer = 0
    - Loss = 5%
    - → Leakage = 100 MW \times 5\% = 5 MW

- Leakage shown in Summary sheet of Results (‘Model leakage TWh/a’)
One of the basic building blocks

- **Units** are used to model
  - Power plants,
  - Storages,
  - Inflow units, *e.g.*, hydro power,
  - VRE units, *e.g.*, wind and solar,
  - Scheduled run units,
  - Conversion units (*e.g.*, power to heat), etc.

- Units are modelled slightly differently in dispatch and invest modes. Invest mode simplifies equations, because it is much slower to solve.
Main constraints and equations for units in dispatch mode are

- Upward limit of generation, reserve provision, and storage charge/discharge \(^{(always \ on)}\)
  - Four unit categories: generating unit, inflow unit without storage, VRE unit, conversion unit

- Online variable \((activated \ by \ user)\). If activated, units can have
  - Start-up costs,
  - Minimum uptime and downtime, and
  - Efficiency loss with partial load

- Ramp constraint \((activated \ by \ user)\)

- Costs related to unit operations \((always \ on, \ but \ parameters \ can \ have \ 0 \ value)\)
  - Variable costs: fuel, variable O&M, CO\(_2\) cost, startup costs
  - Fixed costs: fixed O&M
Unit category

- **Four unit categories:** generating unit, inflow unit without storage, VRE unit, conversion unit
  - FlexTool decides unit category based on the unit input defined in “units” sheet
  - Input options: fuel, cf profile, inflow, input grid/node, none

<table>
<thead>
<tr>
<th>unitGroup</th>
<th>unit_input</th>
<th>fuel</th>
<th>cf_profile</th>
<th>inflow</th>
<th>input_grid</th>
<th>input_node</th>
<th>output_grid</th>
<th>output_node</th>
<th>capacity (MW)</th>
<th>max. inv. (MV)</th>
<th>storage (kWh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_Coal</td>
<td>fuel</td>
<td>coal</td>
<td></td>
<td>inflow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wind</td>
<td>cf_profile</td>
<td>wind</td>
<td>wind_A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Battery</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hydro_RES</td>
<td></td>
<td></td>
<td>nodeB_RES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15000</td>
</tr>
<tr>
<td>Hydro_ROR</td>
<td></td>
<td></td>
<td>nodeB_ROR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVcharger</td>
<td></td>
<td></td>
<td>EV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **ST_Coal:** Input from fuel -> generating unit
- **Wind:** input from cf_profile -> VRE unit
- **Battery:** no input + storage -> generating unit
- **Hydro_RES:** inflow + storage -> generating unit
- **Hydro_ROR:** inflow, but no storage -> inflow unit without storage
- **EVcharger:** input from node -> conversion unit

units sheet
Upward limit: generating units

**Generating units:**
- + generation \([v_{gen}]\)
- + reserve \([v_{reserve}]\)
- - charge \([v_{charge}]\)
- <=
- + capacity OR [see orange box]
- online capacity \([v_{online}]\)

**Generating mode:**
- Max capacity
- OR online capacity
- + reserves
- generation

**Storage charging mode:**
- Max capacity
- + reserves
- - charge

**Demand increase:**
- Max capacity
- “charge”
Model syntax example (upward limit for online units)

\[
\begin{align*}
\text{s.t. upwardLimitOnline } &\{(g,n,u,t) \in \text{gnut} : (g,n,u) \in \text{gnu_gen } \&\& \ u \in \text{unit_online}\} : \\
&+ v_{\text{gen}[g,n,u,t]} \\
&+ (\text{if } (g,n,u) \in \text{gnu_reserve then } v_{\text{reserve}[g,n,u,t]}) \\
&- (\text{if } (g,n,u) \in \text{gnu_storage_charging then } v_{\text{charge}[g,n,u,t]}) \\
&\leq \\
&+ v_{\text{online}[g,n,u,t]} \\
\end{align*}
\]
Upward limit: inflow no storage, VRE

Simplified unit categories to allow faster solve time

**Inflow but no storage:**

+ generation \([v_{\text{gen}}]\)
+ reserve \([v_{\text{reserve}}]\)
\(\leq\)
+ inflow time series \([ts_{\text{inflow}}: \text{series}]\)

**VRE units:**

+ curtail \([v_{\text{curtail}}]\)
\(\leq\)
+ capacity factor \([ts_{\text{cf}}: \text{time series}]\)
\(\times\) capacity \([\text{see orange box}]\)

**Capacity**

=  
+ pre-existing capacity \([\text{units: capacity}]\)
+ forced new capacity \([\text{units: invested_capacity}]\)
+ invested new capacity \([v_{\text{invest}} | v_{\text{investTransfer}}]\)
Upward limit: conversion units

- Single variable $v_{\text{convert}}$ presents both directions of the conversion.
- In the input node, the energy consumption is equal to $v_{\text{convert}} / \text{efficiency}$.
- The output node energy yield is just $v_{\text{convert}}$.
- Efficiency can be a time series.
- Maximum capacity is limited on the input side (units sheet capacity affects input; e.g., heat pump with 100 MW capacity and 2.5 COP can generate 250 MW heat, but can be affected by efficiency time series).
- No startups or online for conversion units.

Conversion upward limit:
- + convert $[v_{\text{convert}}]$.
- + reserve (to output node) $[v_{\text{reserve}}]$
- <=
- + capacity [see orange box]
- OR online capacity $[v_{\text{online}}]$

Converting to output:
- Max capacity
- + reserves
- conversion
Reserve provision

- User defines reserve capabilities of generation and conversion units in input data file, sheet `unit_type`
Maximum reserve provision: generating units

**Reserve limit**

\[ v_{\text{reserve}} \leq \text{max. reserve} \times \text{capacity} \]

OR online \[ v_{\text{online}} \]

**Capacity**

\[ \text{Capacity} = \]

\[ \text{pre-existing capacity} + \text{forced new capacity} + \text{invested new capacity} \]

\[ \text{[units: capacity]} \]

\[ \text{[units: invested_capacity]} \]

\[ \text{[v_invest | v_investTransfer]} \]
Maximum reserve provision: conversion units

Conversion units **convert energy** from one type to another, *e.g.*, electricity to hydrogen or EV charger (grid electricity to car electricity).

In g,n,u,t language, conversion unit changes energy from one grid to another.

This is different from an unit that changes energy from one node to another (transfer link).

**Limit for providing reserve when converting from electricity:**

\[+ \text{reserve (input node)} \leq + \text{convert} \times \text{max\_reserve}\]

- \([v\_\text{reserve}(g,n\_\text{input},u,t)]\)
- \([v\_\text{convert}]\)
- \([\text{unit\_type: max\_reserve}]\)

**Limit for providing reserve when converting to electricity:**

\[+ \text{reserve (output node)} \leq + \text{capacity} \times \text{max\_reserve}\]

- \([v\_\text{reserve}(g,n\_\text{output},u,t)]\)
- \([\text{see orange box}]\)
- \([\text{unit\_type: max\_reserve}]\)
Inflow but no storage:

+ generation \[v_{gen}\]
+ reserve \[v_{reserve}\]
<=
+ inflow time series \[ts_{inflow}: \text{series}\]
Storages

- Deciding parameter is ‘storage (MWh)’ in units sheet
  - If ‘storage (MWh)’ has a positive value, unit has storage → Hydro_RES is storage unit and hydro_ROR is not
  - More details of storage can be given in unit_type sheet, e.g., charge efficiency (eff charge), storage losses (self discharge loss), and discharge efficiency (efficiency)
Time jump for storages

- It is important to maintain **storage chronology** when using jumps in time.

- In full year run, hours follow other and storages are optimised correctly.

- When using jumps, FlexTool has to follow storage values over the jumps.

- Storage value optimisation requires somewhat complicated equations when time jumps are allowed, see following slides.
Storage balance equation

\[
\text{state (t)} = \text{state (t-1)} + \frac{v_{\text{charge}}}{\text{eff}_{\text{charge}}} + \text{inflow} - v_{\text{gen}} \times \text{efficiency} - v_{\text{spill}} - v_{\text{state}} \times \text{self discharge loss}
\]
Additional storage constraints

- Why?
  - Otherwise the model can curtail through losses without limit (linear storage model can generate and charge at the same time)

### Charging limit:
+ charge \[ v_{charge} \] 
\leq 
+ capacity OR online \[ \text{units: capacity} / v_{online} \]

### Generating limit:
+ generation \[ v_{gen} \] 
\leq 
+ capacity OR online \[ \text{units: capacity} / v_{online} \]

### Storage start state
+ state in first time step \[ v_{state} \] 
= 
+ parameter storage start \[ \text{units: storage_start} \]

### Storage finish state
+ state in the last time step \[ v_{state} \] 
= 
+ parameter storage finish \[ \text{units: storage_finish} \]
Activating online variable: related parameters

- User activates **online variables** in input data, sheet master (use_online = 1)
- Minimum load, efficiencies, startup costs, and uptime constraints are unit type parameters (input data, sheet unit_type)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>co2_cost</td>
<td>10</td>
</tr>
<tr>
<td>loss_of_load_penalty</td>
<td>10000</td>
</tr>
<tr>
<td>loss_of_reserves_penalty</td>
<td>20000</td>
</tr>
<tr>
<td>lack_of_inertia_penalty</td>
<td>30000</td>
</tr>
<tr>
<td>curtailment_penalty</td>
<td>5000</td>
</tr>
<tr>
<td>time_in_years</td>
<td>1.000</td>
</tr>
<tr>
<td>time_period_duration</td>
<td>60</td>
</tr>
<tr>
<td>reserve_duration</td>
<td>0.50</td>
</tr>
<tr>
<td>use_capacity_margin</td>
<td>1</td>
</tr>
<tr>
<td>use_online</td>
<td>1</td>
</tr>
<tr>
<td>use_ramps</td>
<td>0</td>
</tr>
<tr>
<td>use_non_synchronous</td>
<td>1</td>
</tr>
<tr>
<td>use_inertia_limit</td>
<td>0</td>
</tr>
<tr>
<td>mode_invest</td>
<td>0</td>
</tr>
<tr>
<td>mode_dispatch</td>
<td>1</td>
</tr>
<tr>
<td>print_duration</td>
<td>0</td>
</tr>
<tr>
<td>print_durationRamp</td>
<td>0</td>
</tr>
<tr>
<td>print_unit_results</td>
<td>0</td>
</tr>
</tbody>
</table>

**master sheet**

**unit_type sheet**
## Start-up and online variables

### Startup

<table>
<thead>
<tr>
<th>Term</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>startup(unit, t)</td>
<td>[v_{\text{startup}}]</td>
</tr>
<tr>
<td>( \geq )</td>
<td></td>
</tr>
<tr>
<td>online(unit, t)</td>
<td>[v_{\text{online}}]</td>
</tr>
<tr>
<td>- online(unit, previous ( t ))</td>
<td>[v_{\text{online}}]</td>
</tr>
</tbody>
</table>

- Activating **online variable** increases costs
  - Start up costs (default value = 0)
  - Increased fuel consumption of online units (default value = full load efficiency -> no increase in fuel consumption)

### Online capacity is constrained

<table>
<thead>
<tr>
<th>Term</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>online</td>
<td>[v_{\text{online}}]</td>
</tr>
<tr>
<td>( \leq )</td>
<td></td>
</tr>
<tr>
<td>capacity</td>
<td>[see orange box]</td>
</tr>
</tbody>
</table>

### Capacity

\[
\text{Capacity} = \\
+ \text{pre-existing capacity} \quad [\text{units: capacity}] \\
+ \text{forced new capacity} \quad [\text{units: invested_capacity}] \\
+ \text{invested new capacity} \quad [v_{\text{invest}} | v_{\text{investTransfer}}]
\]
Minimum uptime and downtime

**Minimum uptime**

+ online \[ v_{\text{online}}(g,n,u,t) \]

\[ \geq \]

- sum of capacity started up
  during minimum uptime \[ \sum (t_>=t-\text{unittype:}\text{min}\_\text{uptime} & t_<t) \]
  \[ v_{\text{startup}}(g,n,u,t_) \]

**Minimum downtime**

+ online \[ v_{\text{online}}(g,n,u,t) \]

\[ \leq \]

+ capacity [see orange box]

- sum of capacity started up
  during minimum downtime \[ \sum (t_>=t+1 & t_<t+1+\text{unittype:}\text{min}\_\text{downtime}) \]
  \[ v_{\text{startup}}(g,n,u,t_) \]

- Online variable is linear
  - Not binary
  - These limits apply to started up quantity (in MWs)
Minimum online and generation, maximum generation

**Minimum online**

+ online \[ v_{\text{online}}(g,n,u,t) \] >= + min. online limit [capacity * ts_unit: min_online]

**Minimum generation**

+ generation \[ v_{\text{gen}}(g,n,u,t) \] >= + min. generation limit [capacity * ts_unit: min_generation]

**Maximum generation**

+ generation \[ v_{\text{gen}}(g,n,u,t) \] <= + max. generation limit [capacity * ts_unit: max_generation]
Unit constraints: minimum load

Minimum load:
+ generation \[ v_{\text{gen}} \]
\[ \geq \]
+ online \[ v_{\text{online}} \]
\[ \times \text{min.load} \]
\[ \text{[unit_type: min_load]} \]

Example unit:
100 MW_fuel max capacity, Min load 0.33
Efficiency 0.4, efficiency at min load 0.35

<table>
<thead>
<tr>
<th>V_{online}</th>
<th>V_{\text{gen}, \text{max}}</th>
<th>Eff. (%)</th>
<th>V_{\text{gen}, \text{min}}</th>
<th>Eff. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 MW</td>
<td>100 MW</td>
<td>40%</td>
<td>33 MW</td>
<td>35%</td>
</tr>
<tr>
<td>60 MW</td>
<td>60 MW</td>
<td>40%</td>
<td>20 MW</td>
<td>35%</td>
</tr>
<tr>
<td>33 MW</td>
<td>33 MW</td>
<td>40%</td>
<td>11 MW</td>
<td>35%</td>
</tr>
<tr>
<td>0 MW</td>
<td>0 MW</td>
<td>-</td>
<td>0 MW</td>
<td>-</td>
</tr>
</tbody>
</table>
Activating ramp constraint: related parameters

- User activates **ramp constraint** in input data, sheet master (use_ramps = 1)
- Also adds rampRoom figures to the results

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>co2_cost</td>
<td>10</td>
</tr>
<tr>
<td>loss_of_load_penalty</td>
<td>10000</td>
</tr>
<tr>
<td>loss_of_reserves_penalty</td>
<td>20000</td>
</tr>
<tr>
<td>lack_of_inertia_penalty</td>
<td>30000</td>
</tr>
<tr>
<td>curtailment_penalty</td>
<td>20</td>
</tr>
<tr>
<td>lack_of_capacity_penalty</td>
<td>5000</td>
</tr>
<tr>
<td>time_in_years</td>
<td>1.00</td>
</tr>
<tr>
<td>time_period_duration</td>
<td>60</td>
</tr>
<tr>
<td>reserve_duration</td>
<td>0.50</td>
</tr>
<tr>
<td>use_capacity_margin</td>
<td>1</td>
</tr>
<tr>
<td>use_online</td>
<td>1</td>
</tr>
<tr>
<td>use_ramps</td>
<td>0</td>
</tr>
<tr>
<td>use_non_synchronous</td>
<td>1</td>
</tr>
<tr>
<td>use_inertia_limit</td>
<td>0</td>
</tr>
<tr>
<td>mode_invest</td>
<td>0</td>
</tr>
<tr>
<td>mode_dispatch</td>
<td>1</td>
</tr>
<tr>
<td>print_duration</td>
<td>0</td>
</tr>
<tr>
<td>print_durationRamp</td>
<td>0</td>
</tr>
<tr>
<td>print_unit_results</td>
<td>0</td>
</tr>
</tbody>
</table>

**master sheet**

**unit_type sheet**
Unit ramp constraint:
+ reserve \[v_{\text{reserve}}(t)]
+ generation \[v_{\text{gen}}(t)]
\leq
+ generation in the previous time step \[v_{\text{reserve}}(t-1)]
+ upward ramp capability \[\text{unit}_\text{type: ramp_up (0-1)}\]
\times\ capacity \[\text{units: capacity} + v_{\text{invest}}\]

- Similar for downward ramp
- Also ramp constrained:
  - storage units
  - demand increasing units
  - conversion units
- Charging can also be ramp constrained
- For storage units maximum upward ramp could be from full charging to full discharging
  \((2 \times \text{capacity})\)
Time series constraints for units

- User can give constraints to units as time series
  - Units need to be flagged to use time series in units sheet
  - Time series are given in ts_unit sheet

- Possible time series are
  - efficiency (works as efficiency at unit_type sheet, but has separate value for each hour)
  - fix_generation, min_generation, max_generation (these fix or limit the generation. Use values from 0-1 as a share of max generation)
  - min_online (this sets a minimum value for unit online variable, see slide 51. Use values from 0-1.)

- The use of all these are demonstrated in template.xlsm
Units with two outputs

Fuel use:

\[ + \text{fuel}_\text{use} \quad [v_{fueluse}] \]
\[ = \]
\[ + 1^{st} \text{output} \times \text{slope} \quad [v_{gen(g,node1,u,t)}] \]
\[ + \text{online} \times \text{section} \quad [v_{online}] \]
\[ + 2^{nd} \text{output} \]
\[ \times \text{fuel}_\text{use}_\text{increase} \quad [\text{units: fuel}_\text{use}_\text{increase}] \]

'No load' fuel use (section)
Units with two outputs

Fixed output ratio
\[ v_{\text{gen}}(g, \text{node2}, u, t) = v_{\text{gen}}(g, \text{node1}, u, t) \times \text{eq\_co-efficient} + \text{eq\_constant} \]

Output 2 cannot provide reserve (use output 1 for electricity)
Units with two outputs

Less than output ratio
+ output to node 2 \( [v_{\text{gen}}(g,\text{node2},u,t)] \)
\( \leq \)
+ output to node 1 \( [v_{\text{gen}}(g,\text{node1},u,t)] \)
\( \times \) co-efficient [units: output2_lt_coeff]
+ constant [units: output2_lt_constant]

Upper limit for 2\(^{nd}\) output:
+ output to node 2 \( [v_{\text{gen}}(g,\text{node2},u,t)] \)
\( \leq \)
+ ratio between outputs [units: output2_max_capacity]
\( \times \) 1\(^{st}\) output online \( [v_{\text{online}}(g,\text{node1},u,t)] \)
OR 1\(^{st}\) output capacity [see orange box]

Greater than output ratio
+ output to node 2 \( [v_{\text{gen}}(g,\text{node2},u,t)] \)
\( \geq \)
+ output to node 1 \( [v_{\text{gen}}(g,\text{node1},u,t)] \)
\( \times \) co-efficient [units: output2_gt_coeff]
+ constant [units: output2_gt_constant]
Units with two outputs

Fixed output ratio

\[ + \text{ output to node 2 } \quad [v_{\text{gen}}(g, \text{node2}, u, t)] \]

\[ = \]

\[ + \text{ output to node 1 } \quad [v_{\text{gen}}(g, \text{node2}, u, t)] \]

\[ \times \text{ co-efficient } \quad [\text{units: output2_eq_coeff}] \]

\[ + \text{ constant } \quad [\text{units: output2_eq_constant}] \]

Upper limit for 2nd node:

\[ + \text{ output to node 2 } \quad [v_{\text{gen}}(g, \text{node2}, u, t)] \]

\[ \leq \]

\[ + \text{ max_output_2_ratio } \quad [\text{units: output2_max_capacity}] \]

\[ \times 1^{\text{st}} \text{ output online } \quad [v_{\text{online}}(g, \text{node1}, u, t)] \]

OR 1st output capacity [see orange box]
Invest run

- **Invest mode** is activated from master sheet in input data (mode_invest = 1)
  - Both invest and dispatch can be active, or only either

- **Capacity margin** approximates reserves during invest run

### Master Sheet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>co2_cost</td>
<td>10</td>
</tr>
<tr>
<td>loss_of_load_penalty</td>
<td>10000</td>
</tr>
<tr>
<td>loss_of_reserves_penalty</td>
<td>20000</td>
</tr>
<tr>
<td>lack_of_inertia_penalty</td>
<td>30000</td>
</tr>
<tr>
<td>curtailment_penalty</td>
<td>20</td>
</tr>
<tr>
<td>lack_of_capacity_penalty</td>
<td>5000</td>
</tr>
<tr>
<td>time_in_years</td>
<td>1.000</td>
</tr>
<tr>
<td>time_period_duration</td>
<td>60</td>
</tr>
<tr>
<td>reserve_duration</td>
<td>0.50</td>
</tr>
<tr>
<td>use_capacity_margin</td>
<td>1</td>
</tr>
<tr>
<td>use_online</td>
<td>1</td>
</tr>
<tr>
<td>use_ramps</td>
<td>0</td>
</tr>
<tr>
<td>use_non_synchronous</td>
<td>1</td>
</tr>
<tr>
<td>use_inertia_limit</td>
<td>0</td>
</tr>
<tr>
<td>mode_invest</td>
<td>0</td>
</tr>
<tr>
<td>mode_dispatch</td>
<td>1</td>
</tr>
<tr>
<td>print_duration</td>
<td>0</td>
</tr>
<tr>
<td>print_durationRamp</td>
<td>0</td>
</tr>
<tr>
<td>print_unit_results</td>
<td>0</td>
</tr>
</tbody>
</table>

### NodeGroup Sheet

<table>
<thead>
<tr>
<th>nodeGroup</th>
<th>capacity margin (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mainLand</td>
<td>120</td>
</tr>
</tbody>
</table>

### GridNode Sheet

<table>
<thead>
<tr>
<th>grid</th>
<th>node</th>
<th>nodeGroup</th>
<th>demand (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>elec</td>
<td>nodeA</td>
<td>mainLand</td>
<td>35</td>
</tr>
<tr>
<td>elec</td>
<td>nodeB</td>
<td>mainLand</td>
<td>10</td>
</tr>
<tr>
<td>elec</td>
<td>nodeC</td>
<td>mainLand</td>
<td>20</td>
</tr>
<tr>
<td>elec</td>
<td>nodeD</td>
<td>mainLand</td>
<td>5</td>
</tr>
</tbody>
</table>
Capacity margin

For each time step:

Capacity margin can be applied either to nodes or to groups of nodes.
Limits for capacity investment

- FlexTool allows following definitions to investments
  - Max invest (MW or MWh) – maximum investment allowed, works only with invest mode
  - Min invest (MW or MWh) – minimum investment required, works only with invest mode, unit groups only
  - Invested capacity (MW) – predefined invested capacity, works both dispatch and invest mode
  - Invested storage (MWh) – predefined invested storage, works both dispatch and invest mode

Define fixed kW/kWh ratio for storages in unitType sheet
  - Two of the three should be provided: [inv. cost MW], [inv. cost MWh] and [fixed kW/kWh ratio]

Multiple constraints
- User can define multiple investment constraints
- FlexTool will always follow all constraints
- However, conflicting constraints will make the model infeasible (crash)
- See slide comments for examples
Costs
Objective function: minimise costs

- Each model run minimises costs of following equation:

\[
\text{Capacity} = \\
\text{fixed} \text{ operation and maintenance costs} \\
\text{variable} \text{ operation and maintenance costs} \\
\text{fuel} \text{ costs of units} \\
\text{CO2} \text{ emission costs} \\
\text{start-up} \text{ costs} \\
\text{penalty cost for loss of load} \\
\text{penalty cost for insufficient upward reserves} \\
\text{penalty cost for insufficient capacity margin} \\
\text{penalty cost for curtailment of VRE} \\
\text{penalty cost for insufficient inertia} \\
\text{unit investment} \text{ costs} \\
\text{storage investment} \text{ costs} \\
\text{transmission line investment} \text{ costs}
\]

- [units: capacity]
- [units: invested_capacity]
- [v_invest | v_investTransfer]

- [capacity \times \text{unittype: fixed_cost}]
- [v_gen | v_charge | v_convert \times \text{unittype: O&M_cost}]
- [v_fuelUse \times \text{fuel: fuel_price}]
- [v_fuelUse \times \text{fuel: CO2_content} \times \text{master: CO2_cost}]
- [v_startup \times \text{unittype: startup_cost}]
- [v_slack \times \text{master: loss_of_load_penalty}]
- [v_reserveSlack \times \text{master: loss_of_reserves_penalty}]
- [v_capacitySlack \times \text{master: lack_of_capacity_penalty}]
- [v_curtail \times \text{master: curtailment_penalty}]
- [v_inertiaSlack \times \text{master: lack_of_inertia_penalty}]
- [v_invest \times \text{unit_type: inv.cost_kW} \times \text{annuity}]
- [v_investStorage \times \text{unit_type: inv.cost_kWh} \times \text{annuity}]
- [v_investTransfer \times \text{nodeNode: inv.cost_kW} \times \text{annuity}]
Cost parameters

- Cost parameters are defined in:
  - **master sheet**: Penalties
  - **fuel sheet**: Operation
  - **unitType sheet**: Operation + investment
  - **nodeNode sheet**: Operation + Investment
Operation costs

- **Fixed operation costs:**
  - Fixed O&M

- **Variable operation costs:**
  - Variable O&M, fuel, CO₂ cost, startup costs
  - Scaled to annual level if running less than full year

- **Scaling to annual costs** is done based on
  - Amount of active timesteps
  - Total number of timesteps
  - Time_in_years parameter at master sheet

**Fixed costs =**
+ capacity \( \times \) fixed O&M

**Capacity =**
+ pre-existing capacity [units: capacity]
+ forced new capacity [units: invested_capacity]
+ invested new capacity [\(v_{\text{invest}} | v_{\text{investTransfer}}\)]

**Variable costs =**
(+ generation \( \times \) O&M cost
  + fuel use \( \times \) fuel price
  + charge \( \times \) O&M cost
  + fuel use \( \times \) fuel CO₂ content \( \times \) CO₂ cost
  + number of start ups \( \times \) capacity \( \times \) startup cost
) \( \times \) scaled to annual
Penalties costs

- The model tries to avoid very high **penalty values**
  - Seeing loss of load in result is a sign of significant flexibility issue
  - Curtailment penalty should be much lower than loss of load penalty. Default value is 20, but it could also be close to zero.

Penalties =
- penalty cost for **loss of load** $[v_{slack} \times \text{master: loss_of_load_penalty}]$
- penalty cost for **insufficient upward reserves** $[v_{reserveSlack} \times \text{master: loss_of_reserves_penalty}]$
- penalty cost for **insufficient capacity margin** $[v_{capacitySlack} \times \text{master: lack_of_capacity_penalty}]$
- penalty cost for **curtailment of VRE** $[v_{curtail} \times \text{master: curtailment_penalty}]$
- penalty cost for **insufficient inertia** $[v_{inertiaSlack} \times \text{master: lack_of_inertia_penalty}]$
**Investment costs**

**In objective function:**

+ **unit investment costs**
  \[ [v_{\text{invest}}] \times \text{inv.cost\_kW} \times \text{annuity} \]

+ **storage investment costs**
  \[ [v_{\text{investStorage}}] \times \text{inv.cost\_kWh} \times \text{annuity} \]

+ **transmission line investment costs**
  \[ [v_{\text{investTransfer}}] \times \text{nodeNode: inv.cost\_kW} \times \text{annuity} \]
More information

- More info from FlexTool methodology report

IRENA FlexTool Support: Flextool@irena.org