

WaterTAP3 Documentation

version 0.0.1

Ariel Miara, Kurban Sitterley, Michael Talmadge, Zhe Huang, Anna Evans, Jennifer Stokes-Draut, Jordan Macknick

August 10, 2021

Contents

| | |
|--------------------------------|----------|
| WaterTAP3 Documentation | 1 |
| WaterTAP3 Overview | 1 |
| WaterTAP3 Installation | 1 |
| WaterTAP3 Cost Estimates | 2 |
| WaterTAP3 Outputs | 2 |
| WaterTAP3 Impacts | 2 |
| WaterTAP3 Framework | 4 |
| Model Structure | 4 |
| Source Water | 5 |
| Treatment Train Setup | 6 |
| Treatment Train Example | 7 |
| Water Recovery | 9 |
| Constituent Removal | 10 |
| Financials | 11 |
| Costing Indices and Factors | 11 |
| Financial Basis Inputs | 12 |
| System Capital Costs | 13 |
| System Operating Costs | 14 |
| Pipe Parity Metrics | 15 |
| Unit Models | 17 |
| Alum Addition | 18 |
| Ammonia Addition | 20 |
| Anti-Scalant Addition | 22 |
| Backwash Solids Handling | 24 |
| Basic Unit | 26 |
| Brine Concentrator | 30 |
| Cartridge Filtration | 31 |
| Caustic Soda Addition | 32 |
| Generic Chemical Addition | 34 |
| Chlorination | 37 |
| Carbon Dioxide Addition | 39 |
| Coagulant Addition | 40 |
| Crystallizer | 42 |
| Deep Well Injection | 43 |
| Electrodialysis Reversal | 45 |
| Evaporation Ponds | 46 |
| Ferric Chloride Addition | 51 |
| Fixed Bed Gravity Basin | 53 |
| Fixed Bed Pressure Vessel | 54 |
| GAC - Gravity | 55 |

| | |
|-----------------------------------|----|
| GAC - Pressure Vessel | 57 |
| Hydrochloric Acid Addition | 59 |
| Iron & Manganese Removal | 61 |
| Landfill | 63 |
| Landfill ZLD | 64 |
| Lime Addition | 65 |
| Lime Softening | 67 |
| Media Filtration | 69 |
| Microfiltration | 71 |
| Municipal Drinking (Distribution) | 72 |
| Ozone + AOP | 73 |
| Reverse Osmosis | 76 |
| Sedimentation | 81 |
| Sodium Bisulfite Addition | 82 |
| Storage Tanks | 84 |
| Sulfuric Acid Addition | 85 |
| Surface Discharge | 87 |
| Seawater Intake | 89 |
| Tri-Media Filtration | 90 |
| UV + AOP | 91 |
| Water Pumping Station | 94 |
| Well Field | 96 |

| | |
|--------------|-----------|
| Index | 97 |
|--------------|-----------|

WaterTAP3 Documentation

This is the documentation for the WaterTAP3 model.

WaterTAP3 Overview

The Water Technoeconomic Assessment Pipe-Parity Platform (WaterTAP3) was developed under the National Alliance for Water Innovation (NAWI) to facilitate consistent technoeconomic assessments of desalination treatment trains. It is an analytically robust modeling tool that can be used to evaluate cost, energy, and environmental tradeoffs of water treatment technology across different water sources, sectors, and scales.

WaterTAP3 is publicly accessible and based on open-source models and data. The model is designed to be flexible and customizable to allow users to adapt or create new treatment trains and process models to assess different water treatment performance criteria.

Examples of WaterTAP3 applications include:

- Baseline the state of water treatment systems across key water-using economic sectors.
- Road-mapping critical technology development priorities to achieve pipe parity.
- Performing ongoing project evaluation for NAWI funded research.
- Conducting technoeconomic and life-cycle assessments of water technologies and systems across the water industry (in development).

WaterTAP3 Installation

WaterTAP3 is still in development so is not listed in `pip` and must be accessed by cloning the current GitHub repo available at <https://github.com/NREL/WaterTAP3>:

1. Install the Anaconda distribution of Python:

- <https://www.anaconda.com/products/individual>

2. Create new empty directory for WaterTAP3 installation:

- e.g. this might be a folder called `wt3` on your desktop

3. Open terminal and navigate to that directory:

- e.g. `cd ~/Desktop/wt3`

4. Clone the repo:

- In terminal, enter `git clone https://github.com/NREL/WaterTAP3`
- For example, if you cloned the repo into `~/Desktop/wt3`, you will have a new directory `~/Desktop/wt3/WaterTAP3`
- All the necessary python files and data files will be downloaded into the `WaterTAP3` directory.

5. Navigate to `~/WaterTAP3`. This directory contains the `watertap3.yml` file that is used to create the `watertap3` Python environment.

6. Create the `watertap3` Python environment:

- In terminal, enter `conda env create --file watertap3.yml`

7. Activate the `watertap3` Python environment:

- In terminal, enter `conda activate watertap3`

8. Navigate to `~/WaterTAP3/watertap3`. This directory contains the `setup.py` file that is used to install `watertap3` as an editable Python package.

- For example, if you cloned the repo into `~/Desktop/wt3`, your working directory should now be `~/Desktop/wt3/WaterTAP3/watertap3`.
- This is also the directory that contains this documentation.

9. Install `watertap3` as an editable Python package:

- Must be in directory containing `setup.py` file
- In terminal, enter `python -m pip install -e .`

• **NOTE: MUST INCLUDE THE PERIOD AT THE END OF THIS COMMAND**

- You should end up with new directory `watertap3.egg-info` in current directory

10. Start Jupyter Notebook app or Jupyter Lab app to access tutorial notebooks and run WaterTAP3:

- For Jupyter Notebook - `jupyter notebook`
- For Jupyter Lab - `jupyter lab`

WaterTAP3 Cost Estimates

Cost estimations are represented at the unit process level (i.e. per treatment technology in the train) and aggregated to the system-level. Estimated costs include:

- Capital investment
- Annual operation and maintenance
 - Fixed (labor, maintenance)
 - Variable (energy, chemical)

WaterTAP3 Outputs

The key performance metrics from WaterTAP3 currently include:

- Levelized Cost of Water: cost per unit of treated water
- Energy intensity: direct energy consumption per unit of treated water
- Water recovery: the percentage of water recovered for a beneficial use
- Constituent removal: the percentage of constituent mass removed from the source water

Other outputs in development include:

- Extent of alternative water and energy resources
- Life-cycle assessment of environmental impacts (e.g. greenhouse gas emissions, total water intensity)
- System resiliency and security

WaterTAP3 Impacts

The objective of WaterTAP3 is to become a standard tool to evaluate water treatment system performance across key metrics used to promote and assess pipe-parity for a range of users including industry and academia. The results from WaterTAP3 can help identify trade-offs among the different performance metrics and enable users to assess how particular technologies affect pipe-parity metrics and how improvements in one metric can affect others across a range of source water conditions and technology performance parameters.

The flexibility and comprehensive scope of WaterTAP3 make it a useful tool for industry-wide technoeconomic analyses, promoting better informed water investment decisions and technology development. The tool can be used by policymakers, planners, and others without extensive analytical experience through the publicly available graphical user interface (under development).

WaterTAP3 Framework

WaterTAP3 simulates steady-state water treatment train performance and costs including flow and constituent mass balance across unit processes, based on source water conditions, configurations of treatment technologies, and system-level techno-economic assumptions. Capital costs, operating costs, and unit performance are estimated for both individual treatment processes and the system as a whole. Users can perform new analysis either by selecting a premade train from the treatment train library or connecting any number of unit processes together in a custom configuration.

The model contains various technical and cost parameter options for several treatment technology models and a library of influent water quality characteristics for a variety of source waters. Users can customize water quality parameters to evaluate technology performance for their unique conditions. The model can be used for different assessment needs, including process simulation, optimization, uncertainty analysis, and sensitivity analysis.

Model Structure

WaterTAP3 is implemented in Python using the [Institute for the Design of Advanced Energy Systems \(IDAES\)](#) which is itself based on the Python optimization and modeling package [Pyomo](#). Though initially developed for the research of innovative energy processes, at its core the IDAES framework is used to track material and energy flows through complex systems, making it useful for tracking water flows, chemical concentrations, and other parameters (e.g. pressure, temperature) through a treatment train.

Every WaterTAP3 model `m` is instantiated as a Pyomo `ConcreteModel()` object upon which an IDAES `FlowsheetBlock()` is placed to facilitate water and material transfers through each unit process in the WaterTAP3 model. Each unit model in WaterTAP3 is an instance of the `WT3UnitProcess()` class, which has properties inherited from the IDAES `UnitModelBlockData()` class. Connections between unit models are represented by the IDAES `Arc()` class.

Much like actual treatment technologies, material and energy exchanges in WaterTAP3 are facilitated by “ports” assigned to each unit process. Each water source and unit process is added as an attribute to the flowsheet and by default are given one inlet port, one outlet port, and one waste port. Additional inlet and outlet ports are added dynamically based on the input file (`treatment_train_setup.xlsx` - discussed in a later section). Flow and mass balance is achieved across each unit process based on water recovery values and constituent removal factors.

Source Water

`data/case_study_water_sources.csv`

Source water information is required to run the model. It can be selected from a list of pre-existing case studies or entered manually using Jupyter Notebook or the input data tables in the data folder. There is no limit on the number of source water nodes for the model and the treatment train design section details how to connect source waters to the treatment train. Water flow rates (volumetric) and any constituent information required to calculate a unit process performance or cost must exist in the source water information.

The source water input dataset is arranged into the following columns:

- **units:** The units used for the constituent, such as kg/m3 (constituent concentration)
- **value:** The number of the variable of interest
- **water_type:** The type/source of the water. This is where the intake unit water_type names must match the train input water_type process parameter in the treatment train design.
- **case_study:** The treatment facility name.
- **reference:** The name of the project.
- **variable:** The name of the constituent or property of interest, such as 'flow' (required) or 'tds'.
- **scenario:** The name of the scenario that the values correspond with, otherwise the default value will be used.

Treatment Train Setup

data/treatment_train_setup.xlsx

This is the input file used to arrange and connect unit processes in the proper order and with the proper flows to create a treatment train according to the user's design.

The columns are:

- **CaseStudy:** The treatment facility name.
- **Reference:** The name of the project that is using the model.
- **Scenario:** The name of the scenario that the values correspond with, used to match water recovery and constituent removal rates specific to the case study-scenario combination.
- **Unit:** The unit process to be added to the train. The unit name in this column must match the exact naming convention of the unit process in the WaterTAP3 model.
- **Type:** The role of the unit process along the treatment train.
 - Options are “intake”, “treatment”, “use”, or “waste”
 - “intake” types are declared for units that begin the treatment train and require at least one source water.
 - “use” types are declared for units considered to have a treated water product. The volume of water that goes through these units is used to calculate LCOW. Therefore, there must be at least one unit declared as the use with a flow rate greater than zero. These units do not have an outlet or waste port connected to another unit (i.e. they are a terminal unit)
 - “treatment” types are declared for treatment units. These units can have any number of inlet, outlet, and waste ports.
 - “waste” types are declared for units that handle terminal waste streams for the facility. These units can have any number of inlet ports but don't have outlet or waste ports that connect to another unit (i.e. they are a terminal unit).
- **UnitName:** The unit process name as defined by the user.
 - Each unit name must be unique. For example, if there are two of the same unit processes under the **Unit** column (e.g. chlorination), the first chlorination unit name could be `chlorination_a` and the second could be `chlorination_b`. The unit in this column is connected to the unit defined in the **ToUnitName** column.
- **ToUnitName:** The **UnitName** for a unit process in a different row that connects to the **UnitName** for the current row. The user can include any number of destination units by including them in this column separated by a column and with *no space in between*.
 - Example: If the user wanted `media_filtration` to split outlet flow 50/50 to `anti_scalant_addition` and `chlorination_b`, they would put `anit_scalant_addition,chlorination_b` in the **ToUnitName** column for the `media_filtration` row.
 - This is better explained in the example below.
- **FromPort:** For each unit process identified in the **ToUnitName** column, there needs to be a corresponding port, either “outlet” or “waste”, that tells the model how to connect the current unit to the next unit in the train. If there is more than one port, the types are separated by a comma and *no space in between*.
 - Example: If the user wanted the treated flow from `media_filtration` to go to `anti_scalant_addition` and the waste flow to go to `backwash_solids_handling`, put `outlet,waste` in the **FromPort** column for the `media_filtration` row.

- **Parameter:** Used to characterize each unit process. All parameter formats are provided as python dictionaries, meaning the cell text must be enclosed by curly braces { }, have quotation marks around parameter names, commas between each parameter, and a colon : separating the parameter name and the value for that parameter.

- Example: If "chemical_name", and "dose" are the parameter names, the proper format is:

```
• {'chemical_name': 'Chlorine', 'dose': 5}
```

- Specifics for required and optional unit parameters for each unit are provided in the documentation for each unit model.
- If the value for an input parameter is a word (e.g. chemical name), it *must* be in single ' ' or double " " quotes. If it is a number, it *must not* be in quotes.

Treatment Train Example

Below is an example treatment train input file with various unit processes, flow splits, and waste streams.

| | CaseStudy | Reference | Scenario | Unit | Type | UnitName | ToUnitName | FromPort | Parameter |
|----|-----------|-----------|----------|-----------------------|-----------|-----------------------|---|---------------------|--|
| 1 | example | nawi | baseline | well_field | intake | well_field | media_filtration | outlet | {'water_type': ['source_water']} |
| 2 | example | nawi | baseline | media_filtration | treatment | media_filtration | cartridge_filtration,anti_scalant_addition,landfill | outlet,outlet,waste | {'split_fraction': [0.65, 0.35]} |
| 3 | example | nawi | baseline | cartridge_filtration | treatment | cartridge_filtration | decarbonator | outlet | |
| 4 | example | nawi | baseline | anti_scalant_addition | treatment | anti_scalant_addition | ro_first_stage | outlet | {'dose': 5} |
| 5 | example | nawi | baseline | reverse_osmosis | treatment | ro_first_stage | decarbonator,ro_second_stage | outlet,waste | {'erd': 'no'} |
| 6 | example | nawi | baseline | reverse_osmosis | treatment | ro_second_stage | decarbonator,landfill | outlet,waste | {'erd': 'no'} |
| 7 | example | nawi | baseline | basic_unit | treatment | decarbonator | lime_softening | outlet | {'unit_process_name': 'decarbonator'} |
| 8 | example | nawi | baseline | lime_softening | treatment | lime_softening | chlorination | outlet | {'lime': 10} |
| 9 | example | nawi | baseline | chlorination | treatment | chlorination | treated_storage | outlet | {'chemical_name': 'Chlorine'} |
| 10 | example | nawi | baseline | holding_tank | treatment | treated_storage | municipal_drinking | outlet | {'avg_storage_time': 12, 'surge_cap': 0.2} |
| 11 | example | nawi | baseline | municipal_drinking | use | municipal_drinking | | | |
| 12 | example | nawi | baseline | landfill | waste | landfill | | | |
| 13 | example | nawi | baseline | landfill | waste | landfill | | | |

- Row 2: well_field is the "intake" unit for the treatment train. All the flow from the well_field flows to media_filtration.
 - "intake" units must have a water_type in the **Parameter** column that corresponds to the proper source water found in case_study_water_sources.csv
 - The source water name must be in quotes and enclosed in brackets [] even if there is only one source water for the treatment train. This is to facilitate the inclusion of multiple source waters for a single intake.
- Row 3: Outlets for media_filtration flows to both cartridge_filtration and ro_first_stage. There is also a waste stream that flows to landfill.
 - The user can designate any split fraction they want in the **Parameter** column by using the "split_fraction" parameter.
 - Split fractions provided with "split_fraction" must be provided in brackets [] and the order of values in the split fraction correspond to the order of units in **ToUnitName**. In this case, 65% of the flow from media_filtration flows to cartridge_filtration and 35% flows to anit_scalant_addition.
 - The unit paramater "split_fraction" must be provided in **Parameter** even if the split is 50/50 (i.e. {'split_fraction': [0.5, 0.5]}).
- Row 4: Flow from cartridge_filtration flows to decarbonator.
 - Note that this flow stream is bypassing the two-stage reverse osmosis process. You can arrange flows in any configuration desired provided the input sheet is correct.

Treatment Train Setup

- Row 5: The chemical addition unit `anti_scalant_addition` receives 35% of flow from `media_filtration`
 - This unit requires a "dose" in the **Parameter** column.
- Row 6: The permeate stream from `ro_first_stage` flows to `decarbonator` and the reject (i.e. waste) stream flows to `ro_second_stage`
 - Note that the **ToUnitName** column does not match the **Unit** column. The user can provide any name they want for the unit in **ToUnitName**, but the **Unit** entry must match *exactly* the name of the Python file for that unit (without '.py' appended)
 - In this case, because there are two `reverse_osmosis` units, they must have different names for WaterTAP3 to correct the unit flows properly.
- Row 7: The permeate stream from `ro_second_stage` also flows to `decarbonator` and the reject (i.e. waste) stream flows to `landfill`
 - Both `reverse_osmosis` units require an "erd" parameter to indicate if the model should include an energy recovery device.
- Row 8: The `decarbonator` is receiving flows from `cartridge_filtration`, `ro_first_stage`, and `ro_second_stage`.
 - The `decarbonator` is a basic unit so must have a **Parameter** "unit_process_name" that matches the name of the desired unit in `basic_unit.csv`.
 - More information on basic units is provided in the basic unit documentation.
- Row 9-11: For each of these rows, 100% of the flow from the **UnitName** is flowing to **ToUnitName**.
 - Each have entries in **Parameters** as required.
- Row 12: For this treatment train `municipal_drinking` is the "use". Any water that flows through this unit is used to calculate LCOW.
 - The **ToUnitName** is empty for "use" units since flow does not go anywhere from here. It is a terminal unit. Similarly, it does not need an entry in **FromPort**

Water Recovery

data/water_recovery.csv

Water recovery x is represented as the fraction (between zero and one) of water recovered (treated) after it passes through a unit process. Case-study based water recovery is given for certain unit processes if the case study has a unique recovery rate, otherwise default values are used.

The water balance for a single unit incorporates the water recovery:

$$Q_{in}x = Q_{out}$$

And if the overall water balance for each unit is:

$$Q_{in} = Q_{out} + Q_{waste}$$

Then the water flow for the waste stream is:

$$Q_{waste} = Q_{in}(1 - x)$$

The total system water recovery along a single stream in the train for all n units is the product of the water recovery for every unit in the stream:

$$x_{stream} = \prod_i^n x_i$$

And therefore the outlet flow for the entire stream the system water recovery multiplied by the sum of all k source flows to the system:

$$Q_{out, stream} = x_{stream} \sum_i^k Q_{in, i}$$

The columns of data/water_recovery.csv are:

- **case_study**: The treatment facility name or default
- **scenario**: The name of the scenario that the values correspond with
- **unit_process**: The name of the unit process corresponding to the recovery value.
- **recovery**: How much water is recovered by each unit process (%)
- **reference**: The source of the recovery data

Constituent Removal

data/constituent_removal.csv

Constituent removal r is represented as the fraction (between zero and one) of the mass of the constituent being removed after it passes through a unit process. Case-study based constituent removal is given for certain unit processes if the case study has a unique recovery rate, otherwise default value are used.

The constituent removal data table states how much each unit process in each treatment facility removes a given water constituent. The model uses the information about the constituents in the source water as well as the removal rates of each unit process given in this constituent removal table to calculate constituent levels in the output water.

The overall mass balance for a single constituent i flowing into a single unit is based on volumetric flow and constituent concentration c :

$$Q_{in}C_{in,i} = Q_{out}C_{out,i} + Q_{waste}C_{waste,i}$$

Any mass removed by the unit is assumed to flow out the waste outlet. Then the constituent removal is:

$$Q_{in}C_{in,i}r_i = Q_{waste}C_{waste,i}$$

And therefore the mass flow out of the unit (i.e. to the next unit process) can be calculated with:

$$Q_{out}C_{out,i} = Q_{in}C_{in,i}(1 - r_i)$$

The constituent removal input dataset is arranged into the following columns:

- **units:** The units used for the constituent, such as kg/m3 (constituent concentration)
- **calculation_type:** How the model will handle the values when the unit process changes the constituent level, ultraviolet transmittance, or pH.
 - *fractional_constituent_removal:* fractional removal
 - *absolute_value:* percent removal for ultraviolet transmittance
 - *delta_constituent_or_property:* when the pH is changed
- **unit_process:** Which unit process is removing the constituent.
- **case_study:** The treatment facility name.
- **value:** The fraction or percent of the constituent in the source water that will be removed.
- **data_source:** The source of the data values and how values were calculated.
- **constituent:** The constituent being removed as named in the model
- **scenario:** The name of the scenario that the values correspond with
- **reference:** The name of the project that is using the mode

Financials

This page describes the different costing and pipe party metrics calculated in WaterTAP3 and how they are calculated in the model.

Costing Indices and Factors

Costing indices are available in `plant_cost_indices.csv` located in the data folder.

There are four types of cost indices applied in WaterTAP3 – Capital, Catalysts and Chemicals, Labor and Consumer Price Index. These factors are used to help account for the time-value of investments and are used in the capital and operating cost calculations. The following index factors are calculated for each process model in the treatment train based on:

- The user input for Analysis Year (for Cost Indices). This value is consistent across the entire treatment train.
- The Basis Year for the specific Process Model, Catalyst, Chemical, Replacement Part or other purchased item. This value is process-model-dependent with potentially different values across unit models within a treatment train.

The four cost indices are (where Y is the year):

- Capital Cost Index Factor:

$$f_{cap} = \frac{Y_{analysis}}{Y_{model}}$$

- Chemical Cost Index Factor:

$$f_{chem} = \frac{Y_{analysis}}{Y_{chem}}$$

- Labor Cost Index Factor:

$$f_{labor} = \frac{Y_{analysis}}{Y_{model}}$$

- Other Cost Index Factor:

$$f_{other} = \frac{Y_{analysis}}{Y_{model}}$$

There are also various assumed costing factors for each case study read in from `case_study_basis.csv`:

- Electricity price:

$$P$$

- Plant capacity utilization:

$$f_{util}$$

- Land cost as percent of FCI:

$$f_{land}$$

- Working capital as percent of FCI:

$$f_{work}$$

- Salaries as percent of FCI:

f_{sal}

- Maintenance costs as percent of FCI:

f_{maint}

- Laboratory costs as percent of FCI:

f_{lab}

- Insurance/taxes as percent of FCI:

f_{ins}

- Benefits as percent of salary:

f_{ben}

- Assumed plant lifetime:

L

- Weighted Average Cost of Capital (debt interest rate):

$WACC$

Financial Basis Inputs

The case study basis input data table contains the foundational technoeconomic assumptions for the entire treatment train.

The input dataset is arranged into the following columns:

- **case_study**: The treatment facility name.
- **scenario**: The name of the scenario that the TEA values correspond with
- **value**: The number or name of the variable of interest
- **reference**: The name of the project that is using the model
- **variable**: The name of the variable of interest
 - *analysis_year*: The first year of the plant is/was in operation
 - *location_basis*: The country or U.S. state where the plant is located. Used for assigning the electricity cost [\$/kwh]. Electricity costs are provided in the data folder.
 - *plant_life_yrs*: The initial design basis for plant-life and used for life cycle analysis calculations. The default plant-life is 20 years.
 - *land_cost_percent*: The assumed cost of land as a percentage of total fixed capital investment. This is a part of the total capital investment.
 - *working_capital_percent*: The assumed cost of working capital as a percentage of total fixed capital investment. This is a part of the total capital investment.
 - *salaries_percent*: The assumed cost of salaries as a percentage of total fixed capital investment. This is a part of the fixed operating costs.
 - *employee_benefits_percent*: The assumed cost of employee benefits as a percentage of total salary cost. This is a part of the fixed operating costs.

- *maintenance_cost_percent*: The assumed cost of maintenance as a percentage of total fixed capital investment. This is a part of the fixed operating costs.
- *laboratory_fees_percent*: The assumed cost of laboratory fees as a percentage of total fixed capital investment. This is a part of the fixed operating costs.
- *insurance_and_taxes_percent*: The assumed cost of insurance and taxes as a percentage of total fixed capital investment. This is a part of the fixed operating costs.
- *default_cap_scaling_exp*: The typical value for economy-of-scale for capital equipment. This is a part of the total and fixed capital investment.
- *default_opex_scaling_exp*: The typical value for economy-of-scale for fixed plant costs. This is a part of the fixed operating costs.
- *cap_by_equity*: The percent of the capital that is financed by equity rather than debt. This is a part of the capital recovery factor and LCOW calculations.
- *debt_interest_rate*: The rate for loan financing of capital. This is a part of the capital recovery factor and LCOW calculations.
- *exp_return_on_equity*: The expected return, interest rate, or cost of capital associated with the portion of capital financed with equity.
- *default_tpec_multiplier*: The Total Purchased Equipment Cost (TPEC) to fixed capital investment. This is a part of the total and fixed capital investment calculations.
- *default_tic_multiplier*: The Total Installed Cost (TIC) to fixed capital investment. This is a part of the total and fixed capital investment calculations.

System Capital Costs

The **Total Installed Costs (TIC)** are calculated for each individual unit process within the treatment train. This represents the cost of capital equipment plus the costs associated with installing the equipment in the plant. These include any costs associated with foundations, piping, insulation, assembly, buildings, electrical systems, and instrumentation. There are four approaches to calculating TIC in WaterTAP3, depending on the process, that vary in their level of detail:

- As a function of volumetric or mass flow only:
 - e.g. Basic units
- As function volumetric or mass flow and at least one other design criteria, such as chemical additions in which a dose is a required parameter:
 - e.g. chemical additions
- Based on physical performance/attributes of unit model, such as water recovery or constituent removal:
 - e.g. UV/AOP
- Multiple cost elements summed to calculate the TIC based on physical performance/attributes of unit model, such as pump and membrane costs for RO based on feed and osmotic pressures or evaporation pond costs that are based on evaporation rates, water recovery, and other design criteria:
 - e.g. Reverse osmosis

Depending on the costs assumed to be included in the unit's capital cost calculation (one of the four methods described above), some units may require additional cost multipliers to fully represent the TIC. For these units, the

calculated capital cost is multiplied by either the **Equipment Installation Factor (EIF)** or the **Indirect Cost Factor (ICF)**. The EIF and ICF have default values of 3.4 (typical value range: 2.5-6.7) and 1.65 (typical value range: 1.2-1.7), respectively. After inclusion of either of these factors (if necessary), TIC is assumed to include indirect costs associated with constructing the process such as engineering costs, construction expenses, legal expenses, contractor fees, and contingencies. This is the **unadjusted Fixed Capital Investment**:

$$FCI_{unadj} = (EIF)TIC$$

Or:

$$FCI_{unadj} = (ICF)TIC$$

Then TIC is adjusted by the Capital Cost Index Factor (defined above) to get the FCI:

$$FCI = f_{cap}FCI_{unadj}$$

Finally, to arrive at the **Total Capital Investment (TCI)**, land costs and the working capital are added to the FCI:

$$TCI = FCI + C_{land} + C_{work}$$

Where:

$$C_{land} = f_{land}FCI$$

And:

$$C_{work} = f_{work}FCI$$

System Operating Costs

WaterTAP3 considers both variable and fixed operating costs. Variable operating costs are dependent on the flow rate and capacity utilization of each treatment technology, while fixed costs are dependent on the capital costs of the treatment facility.

Variable Operating Costs

Variable operating costs include any chemical additions, electricity costs, and other variable costs such as equipment replacements (e.g., membrane replacement costs for a reverse osmosis unit).

Chemical costs are based on the chemical dosage [kg/m³] as defined in the model or by the user for a given chemical addition. The costs of the chemicals can be found in the data folder. The annual chemical costs [\$MM/yr] are calculated as:

$$C_{chem} = \sum_k^n D_k C_k Q_{in} f_{util}$$

Where D is the dose [kg/m³] of chemical k and C is the unit cost [\$/kg] of chemical k as found in `catalysts_chemicals.csv`.

Electricity costs are based on the electricity intensity [kWh/m³] of each unit process, which is provided as a constant or calculated based on the configuration of the treatment process (see unit models for details). The annual electricity costs [\$MM/yr] are calculated as:

$$C_{elec} = \sum_k^n E_k Q_{in} f_{util} P$$

Where E is the electricity intensity [kWh/m³] for unit k and P is the price of electricity for the locale [\$/kWh].

There is also possibility for the inclusion of other operating costs that are unit specific. For most units, there are no costs included in this category.

Fixed Operating Costs

Employee salaries are calculated and scaled according to:

$$C_{sal} = f_{labor} f_{sal} FCI_{unadj}$$

Employee benefits are calculated according to:

$$C_{ben} = C_{sal} f_{ben}$$

Plant maintenance costs are calculated as:

$$C_{maint} = f_{maint} FCI$$

Plant laboratory costs are calculated as:

$$C_{lab} = f_{lab} FCI$$

Plant insurance and taxes are calculated according to:

$$C_{ins} = f_{ins} FCI$$

Total & Annual Operating Costs

The total fixed operating costs are calculated as:

$$C_{op, tot} = C_{sal} + C_{ben} + C_{maint} + C_{lab} + C_{ins}$$

And annual operating costs are:

$$C_{op, an} = C_{chem} + C_{elec} + C_{other} + C_{op, tot}$$

Pipe Parity Metrics

Levelized Cost of Water (LCOW)

The Levelized Cost Of Water (LCOW) [\$/m³] is one of the primary pipe-parity metrics provided as an output from WaterTAP3.

$$LCOW = \frac{f_{recov} TCI + C_{op, an}}{V f_{util}}$$

With the capital recovery factor:

$$f_{recov} = \frac{WACC(1 + WACC)^L}{(1 + WACC)^L - 1}$$

And V is the total volume of treated water that goes toward a beneficial use. In WaterTAP3, this is the volume of water that flows through any unit designated as a “use” in the input sheet `treatment_train_setup.xlsx`.

The individual components that sum to the total LCOW are calculated as:

$$LCOW_{TCI} = \frac{f_{recov} TCI}{V_{treat} f_{util}}$$

The electricity LCOW is calculated as:

$$LCOW_{elec} = \frac{C_{elec}}{V_{treat}f_{util}}$$

The fixed operating LCOW is calculated as:

$$LCOW_{op} = \frac{C_{op,an}}{V_{treat}f_{util}}$$

The chemical cost LCOW is calculated as:

$$LCOW_{other} = \frac{C_{chem}}{V_{treat}f_{util}}$$

The other cost LCOW is calculated as:

$$LCOW_{other} = \frac{C_{other}}{V_{treat}f_{util}}$$

The electricity intensity for the system is calculated as:

$$E_{sys} = \frac{C_{elec}}{PV_{treat}}$$

Unit Models

All unit processes achieve the mass balances described in the water recovery and constituent removal section. The calculations of water recovery and constituent removal are based on factors (in the data folder) or equations that represent some level of the physical processes that occur in the treatment process. There are four levels of detail (discussed also in the Financials section) across the unit models and the current status of the unit models is shown in the figure below.

Alum Addition

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

Unit Parameters

There is one unit parameter:

- "dose" - dose of chemical [mg/L]
- Required parameter

Capital Costs

The chemical solution flow S [gal/day] is used in a cost curve of the general form:

$$C = aS^b$$

For a single alum addition unit, $a = 15408$ and $b = 0.5479$. The full cost equation in WaterTAP3 is:

$$C_{alum} = 15408S^{0.5479}$$

This cost is then multiplied by the number of units and the EIF factor for the final FCI for the chemical addition. These parameters were determined by fitting data from FIGURE 5.5.15 - LIQUID ALUM FEED in McGivney & Kawamura (2008).

Electricity Intensity

Electricity intensity for chemical additions is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/day] of the solution necessary to achieve the desired dose:

$$M_{alum} = Q_{in}D_{alum}$$

- The volumetric flow S [gal/min] of the chemical solution, which incorporates the solution density [kg/m³] and ratio of the chemical in the solution [v/v]:

$$S = \frac{M_{alum}}{\rho_{alum}X_{alum}}$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

Then the electricity intensity is calculated as:

$$E_{alum} = \frac{0.746Sh}{3960\eta_{pump}\eta_{motor}Q_{in}}$$

Assumptions

- Number of units = 2

- Solution density [kg/m³] = 1360
- Ratio in solution = 0.5
- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

Reference

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

Alum Addition Module

```
class watertap3.wt_units.alum_addition.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity.

Parameters:

- **lift_height** (*float*) – Lift height for pump [ft]
- **pump_eff** (*float*) – Pump efficiency
- **motor_eff** (*float*) – Motor efficiency

Returns: Electricity intensity [kWh/m³]

```
    fixed_cap (unit_params)
```

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'dose': 10}

Parameters: **dose** (*float*) – Alum dose [mg/L]

Returns: Alum addition fixed capital cost [\$MM]

```
    get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

```
    solution_vol_flow ()
```

Chemical solution flow in gal/day

Parameters:

- **solution_density** (*float*) – Alum solution density [kg/m³]
- **ratio_in_solution** (*float*) – Ratio of alum in solution

Returns: Alum solution flow [gal/day]

Ammonia Addition

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

Unit Parameters

There is one unit parameter:

- "dose" - dose of chemical [mg/L]
- Required parameter

Capital Costs

The ammonia solution flow S [gal/day] is used in a cost curve of the general form:

$$C = aS^b$$

For a ammonia addition unit, $a = 6699.1$ and $b = 0.4219$. The full cost equation in WaterTAP3 is:

$$C_{amm} = 6699.1S^{0.4219}$$

This cost is then multiplied by the number of units and the EIF factor for the final FCI for the chemical addition. These parameters were determined by fitting data from FIGURE 5.5.11 - AQUA AMMONIA FEED 29% SOLUTION in McGivney & Kawamura (2008).

Electricity Intensity

Electricity intensity for chemical additions is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/day] of the solution necessary to achieve the desired dose:

$$M_{amm} = Q_{in}D_{amm}$$

- The volumetric flow S [gal/min] of the chemical solution, which incorporates the solution density [kg/m³] and ratio of the chemical in the solution [v/v]:

$$S = \frac{M_{amm}}{\rho_{amm}X_{amm}}$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

Then the electricity intensity is calculated as [kWh/m³]:

$$E_{amm} = \frac{0.746Sh}{3960\eta_{pump}\eta_{motor}Q_{in}}$$

Assumptions

- Number of units = 2

- Solution density [kg/m³] = 1360
- Ratio in solution = 0.5
- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

Reference

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

Sodium Bisulfite Module

```
class watertap3.wt_units.ammonia_addition.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity.

Parameters:

- **lift_height** (*float*) – Lift height for pump [ft]
- **pump_eff** (*float*) – Pump efficiency
- **motor_eff** (*float*) – Motor efficiency

Returns: Electricity intensity [kWh/m³]

```
fixed_cap (unit_params)
```

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'dose': 10}

Fixed capital for ammonia addition is a function of ammonia dose, ammonia solution flow, and the number of units.

Parameters: **dose** (*float*) – Ammonia dose [mg/L]

Returns: Ammonia addition fixed capital cost [\$MM]

```
get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

```
solution_vol_flow ()
```

Chemical solution flow in gal/day

Parameters:

- **solution_density** (*float*) – Solution density [kg/m³]
- **ratio_in_solution** (*float*) – Ratio of chemical in solution

Returns: Ammonia solution flow [gal/day]

Anti-Scalant Addition

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

Unit Parameters

There is one unit parameter:

- "dose" - dose of chemical [mg/L]
- Required parameter

Capital Costs

The model assumes Hydrazine as the anti-scalant. This unit is based off of costing parameters for sulfuric acid.

The anti-scalant solution flow S [gal/day] is used in a cost curve of the general form:

$$C = aS^b$$

For a single anti-scalant addition unit, $a = 900.97$ and $b = 0.6179$. The full cost equation in WaterTAP3 is:

$$C_{anti} = 900.97S^{0.6179}$$

This cost is then multiplied by the number of units and the EIF factor for the final FCI for the chemical addition. These parameters were determined by fitting data from FIGURE 5.5.11 - SULFURIC ACID FEED in McGivney & Kawamura (2008).

Electricity Intensity

Electricity intensity for chemical additions is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/day] of the solution necessary to achieve the desired dose:

$$M_{anti} = Q_{in}D_{anti}$$

- The volumetric flow S [gal/min] of the chemical solution, which incorporates the solution density [kg/m³]:

$$S = \frac{M_{anti}}{\rho_{anti}}$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

Then the electricity intensity is calculated as [kWh/m³]:

$$E_{anti} = \frac{0.746Sh}{3960\eta_{pump}\eta_{motor}Q_{in}}$$

Assumptions

- Number of units = 2
- Solution density [kg/m³] = 1021
- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

Reference

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

Anti-Scalant Module

```
class watertap3.wt_units.anti_scalant_addition.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity.

Parameters:

- **lift_height** (*float*) – Lift height for pump [ft]
- **pump_eff** (*float*) – Pump efficiency
- **motor_eff** (*float*) – Motor efficiency

Returns: Electricity intensity [kWh/m³]

```
fixed_cap (unit_params)
```

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'dose': 10}

Returns: Anti-scalant addition fixed capital cost [\$MM]

```
get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

```
solution_vol_flow ()
```

Chemical solution flow in gal/day

Parameters: **solution_density** (*float*) – Solution density [kg/m³]

Returns: Anti-scalant solution flow [gal/day]

Backwash Solids Handling

The backwash solids handling unit model represents the handling of the waste/backwash streams from filtration processes (e.g. tri-media filtration). It is typically recycled back into the treatment train or sent to a final waste unit.

Unit Parameters

There is one parameter for backwash solids handling:

* "recovery" - proportion of flow recycled back to treatment technology

- Required parameter
- Value between 0 and 1

Capital Costs

The capital costs are a function of mass flow [kg/hr] from Figure 5.7.1 in McGivney & Kawamura (2008):

$$C_{bw} = 9.76 \frac{M_{in}}{1577255}^{0.918}$$

The mass flow in is the sum of all the constituents flowing into the unit:

$$C_{in} = \sum_i^n C_i$$

Then, the density of the solution is [kg/m³]:

$$\rho_{in} = 0.6312(C_{in}) + 997.86$$

And mass flow is determined with [kg/hr]:

$$M_{in} = \rho_{in} Q_{in}$$

Electricity Intensity

Electricity intensity for backwash solids handling is based off the pump used. The calculation includes:

- Lift height [ft]:

$$h$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

- And the influent flow in [gal/min] and [m³/hr]:

$$Q_{gpm}, Q_{m3hr}$$

Then the electricity intensity is calculated as:

$$E_{bw} = \frac{0.746 Q_{gpm} h}{3960 \eta_{pump} \eta_{motor} Q_{m3hr}}$$

Assumptions

- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

References

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

Unit Template

```
class watertap3.wt_units.backwash_solids_handling.UnitProcess(*args, **kwds)

    get_costing(unit_params=None, year=None)
        Initialize the unit in WaterTAP3.
```

Basic Unit

Capital costs for the units defined as “basic” are based entirely on either the volumetric or mass flow into the unit. All of the data for each of the basic units is read in from the `basic_unit.csv` located in the data folder.

Unit Parameters

- "unit_process_name" - unit process name that is being represented as a basic unit.
 - Required parameter
 - Must be enclosed in double " " or single ' ' quotes
 - Must match *exactly* the entry in the `unit_process` column in `basic_unit.csv`

basic_unit.csv

This .csv file contains all of the data used to calculate fixed capital costs and electricity intensity for all basic units. The columns are:

- `unit_process` - unit process name in WaterTAP3
- `flow_basis` - flow basis from costing source [m3/hr]
- `cap_basis` - B fixed capital investment from costing source [\$MM]
- `cap_exp` - x fixed capital investment scaling exponent
- `elect` - electricity intensity for unit [kWh/m3]
- `year` - costing basis year
- `kind` - determines if unit is based on water flow or mass flow

Capital Costs

Flow Based

Capital costs for basic units based on water flow are calculated with the general form:

$$C_{basic} = Bf^x$$

The ratio of the unit water flow to the basis water flow is the scaling factor f used to scale the unit costs to the basis costs:

$$f = \frac{Q_{in}}{Q_{basis}}$$

Mass Based

For those units based on mass flow, the mass flowing into the unit must be determined. First we determine the concentration flowing into the unit [kg/m3], calculated as the summation of the concentration of all constituents entering the unit:

$$C_{in} = \sum_i^n c_i$$

Then, we estimate the density of the solution [kg/m3] from Bartholomew & Mauter (2019):

$$\rho_{in} = 0.6312(C_{in}) + 997.86$$

Mass flow [kg/hr] is determined with:

$$M_{in} = \rho_{in} Q_{in}$$

Capital costs for basic units based on mass flow are calculated with the general form:

$$C_{basic} = B f^x$$

The ratio of the unit mass flow to the basis mass flow is the scaling factor f used to scale the unit costs to the basis costs:

$$f = \frac{M_{in}}{M_{basis}}$$

Electricity Intensity

Electricity intensity for basic units is read directly from the `elect` column in `basic_unit.csv` (see above) and does not scale with flow (i.e. it is a fixed value).

List of Basic Units

- Treatment Technologies:
 - ABMET Intermediate Pumps
 - ABMET Interstage Pumps
 - Aeration Basins
 - Air Floation
 - Anaerobic Digestion Oxidation
 - Bio-Active Filtration
 - Bioreactor
 - Bioreactor BW Pump
 - Bioreactor Feed Pump
 - Bioreactor Effluent Pump
 - Blending Reservoir
 - Buffer Tank
 - Conventional Activated Sludge (CAS)
 - Clarifier
 - Decarbonators
 - Dissolved Air Flotation (DAF)
 - Drainage Sump Pumps
 - Filter Presses
 - Intrusion Mitigation
 - Membrane Bioreactors (MBR)
 - Microscreen Filtration
 - Nanofiltration (NF)
 - pH Adjustment
 - Raw Water Pumps
 - Screens
 - Separators
 - Settling Ponds
 - Settling Tanks
 - SMP
 - Transfer Pumps
 - Tramp Oil Tanks
 - Ultrafiltration (UF)
 - Ultratiltration Feed Pumps
 - WAIV
 - Walnut Shell Filter
- Uses/Waste Streams & Other:

- Agriculture
- Cooling Supply
- Discharge
- Industrial
- Intrusion Mitigation
- Injection Wells
- Irrigation
- Mining
- Municipal WWTP
- Passthrough

References

Bartholomew, T. V. and M. S. Mauter (2019).
“Computational framework for modeling membrane processes without process and solution property simplifications.”
Journal of Membrane Science 573: 682-693.

Basic Unit Module

```
class watertap3.wt_units.basic_unit.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity for basic units.

Returns: Electricity intensity [kWh/m3]

```
    fixed_cap ()
```

Parameters: **flow_in** (*float*) – Flow in to basic unit [m3/hr]

Brine Concentrator

The brine concentrator represents a thermal (evaporative brine concentrator) based on WateReuse Foundation (2008).

Unit Parameters

None.

Capital Costs

Capital costs for brine concentrators are a function of influent TDS, water recovery, and flow in. The regression is based on data found in Tables 5.1 and A2.3 found in WateReuse Foundation (2008).

$$C_{brine} = 15.1 + 3.02 \times 10^{-4}(C_{TDS}) - 18.8(x_{wr}) + 8.08 \times 10^{-2}(Q_{in})$$

Electricity Intensity

Electricity intensity is a function of the same variables and uses the same reference.

$$E_{brine} = 9.73 + 1.1 \times 10^{-4}(C_{TDS}) + 10.4(x_{wr}) + 3.83 \times 10^{-5}(Q_{in})$$

Reference

Mickley, Michael C. (2008)
 “Survey of High-Recovery and Zero Liquid Discharge Technologies for Water Utilities”
 WateReuse Foundation
 ISBN: 978-1-934183-08-3

Brine Concentrator Module

```
class watertap3.wt_units.brine_concentrator.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity for brine concentrator. :param tds_in: TDS concentration in to brine concentrator [mg/L] :type tds_in: float :param water_recovery: Water recovery for the brine concentrator :type water_recovery: float :param flow_in: Water flow in to brine concentrator [m3/hr] :type flow_in: float :return: Electricity intensity [kWh/m3]

```
    fixed_cap ()
```

Fixed capital for brine concentrator.

Parameters:

- **tds_in** (*float*) – TDS concentration in to brine concentrator [mg/L]
- **water_recovery** (*float*) – Water recovery for the brine concentrator
- **flow_in** (*float*) – Water flow in to brine concentrator [m3/hr]

Returns: Fixed capital cost for brine concentrator [\$MM]

```
    get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

Cartridge Filtration

Unit Parameters

None

Capital Costs

The capital costs are a function of flow [MGD] with cost curve parameters from McGivney & Kawamura (2008):

$$C_{cart} = 0.72557Q_{in}^{0.5862}$$

Electricity Intensity

Electricity intensity for cartridge filtration is fixed at 2E-4 kWh/m3 and is taken from Bukhary, et al. (2019).

References

CAPITAL

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

ELECTRICITY

Bukhary, S., Batista, J., Ahmad, S. (2019).
An Analysis of Energy Consumption and the Use of Renewables for a Small Drinking Water Treatment Plant.
Water, 12(1), 1-21.

Cartridge Filtration Module

```
class watertap3.wt_units.cartridge_filtration.UnitProcess(*args, **kwds)
```

```
    get_costing(unit_params=None, year=None)
        Initialize the unit in WaterTAP3.
```

Caustic Soda Addition

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

Unit Parameters

There is one unit parameter:

- "dose" - dose of chemical [mg/L]
- Required parameter

Capital Costs:

The caustic soda flow S [gal/day] is used in a cost curve of the general form:

$$C = aS^b$$

For a single caustic soda addition unit, $a = 2262.8$ and $b = 0.6195$. The full cost equation in WaterTAP3 is:

$$C_{NaOH} = 2262.8S^{0.6195}$$

This cost is then multiplied by the number of units and the EIF factor for the final FCI for the chemical addition. These parameters were determined by fitting data from FIGURE 5.5.12b - SODIUM HYDROXIDE FEED in McGivney & Kawamura (2008).

Electricity Intensity

Electricity intensity for chemical additions is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/day] of the solution necessary to achieve the desired dose:

$$M_{NaOH} = Q_{in}D_{NaOH}$$

- The volumetric flow 'S' [gal/min] of the chemical solution, which incorporates the solution density [kg/m³]:

$$S = \frac{M_{NaOH}}{\rho_{NaOH}X_{NaOH}}$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

Then the electricity intensity is calculated as [kWh/m³]:

$$E_{NaOH} = \frac{0.746Sh}{3960\eta_{pump}\eta_{motor}Q_{in}}$$

Assumptions

- Number of units = 2
- Solution density [kg/m³] = 1021
- Ratio in solution = 50%
- Lift height [ft] = 100
- Pump efficiency = 0.9

- Motor efficiency = 0.9

Reference

William McGivney & Susumu Kawamura (2008)
 Cost Estimating Manual for Water Treatment Facilities
 DOI:10.1002/9780470260036

Caustic Soda Module

```
class watertap3.wt_units.caustic_soda_addition.UnitProcess (*args, **kwargs)
```

elect ()

Electricity intensity.

Parameters:

- **lift_height** (*float*) – Lift height for pump [ft]
- **pump_eff** (*float*) – Pump efficiency
- **motor_eff** (*float*) – Motor efficiency

Returns: Electricity intensity [kWh/m3]

fixed_cap (unit_params)

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'dose': 10}

Parameters: **dose** (*float*) – Caustic dose [mg/L]

Returns: Caustic soda addition fixed capital cost [\$MM]

get_costing (unit_params=None, year=None)

Initialize the unit in WaterTAP3.

solution_vol_flow ()

Chemical solution flow in gal/day

Parameters:

- **solution_density** (*float*) – Solution density [kg/m3]
- **ratio_in_solution** (*float*) – Ratio of chemical in solution

Returns: Caustic soda solution flow [gal/day]

Generic Chemical Addition

This unit is for a generic chemical addition that does not have a specific unit model in WaterTAP3.

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

You can add any chemical you want to `catalyst_chemicals.csv` located in the data folder. The entry must include a price per kg and a year for the unit price.

Unit Parameters

There are two unit parameters:

- "chemical_name" - the name of the chemical to be added:
 - Required parameter
 - Must be in single ' ' or double " " quotes
 - Must match exactly an entry in `catalyst_chemicals.csv`. If it is not in that file, users can easily add a chemical by providing an entry for each column.
- "dose" - chemical dose [mg/L]:
 - Required parameter
 - Dose of chemical to be added

Capital Costs

The generic chemical addition module is based off costing parameters for sulfuric acid.

The chemical solution flow S [gal/day] is used in a cost curve of the general form:

$$C = aS^b$$

For a single unit, $a = 900.97$ and $b = 0.6179$. The full cost equation in WaterTAP3 is:

$$C_{chem} = 900.97S^{0.6179}$$

This cost is then multiplied by the number of units and the EIF factor for the final FCI for the chemical addition. These parameters were determined by fitting data from FIGURE 5.5.11 - SULFURIC ACID FEED in McGivney & Kawamura (2008).

Electricity Intensity

Electricity intensity for chemical additions is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/day] of the solution necessary to achieve the desired dose:

$$M_{chem} = Q_{in}D_{chem}$$

- The volumetric flow S [gal/min] of the chemical solution, which incorporates the solution density [kg/m³]:

$$S = \frac{M_{chem}}{\rho_{chem}}$$

- The pump and motor efficiencies:

$$\eta_{\text{pump}}, \eta_{\text{motor}}$$

Then the electricity intensity is calculated as [kWh/m³]:

$$E_{\text{chem}} = \frac{0.746Sh}{3960\eta_{\text{pump}}\eta_{\text{motor}}Q_{\text{in}}}$$

Assumptions

- Number of units = 2
- Solution density [kg/m³] = 1000
- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

Reference

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

Chemical Addition Module

```
class watertap3.wt_units.chemical_addition.UnitProcess (*args, **kwds)
```

elect (unit_params)
Electricity intensity.

Parameters:

- **lift_height** (*float*) – Lift height for pump [ft]
- **pump_eff** (*float*) – Pump efficiency
- **motor_eff** (*float*) – Motor efficiency

Returns: Electricity intensity [kWh/m³]

fixed_cap (unit_params)

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'dose': 10}

Fixed capital for chemical addition is a function of chemical dose, chemical solution flow, and the number of units.

Parameters:

- **chemical_name** (*str*) – Chemical name to be used.
- **dose** (*float*) – Dose of chemical [mg/L]

Returns: Chemical addition fixed capital cost [\$MM]

get_costing (unit_params=None, year=None)
Initialize the unit in WaterTAP3.

solution_vol_flow ()
Chemical solution flow in gal/day

Parameters: **solution_density** (*float*) – Solution density [kg/m³]

Returns: Chemical solution flow [gal/day]

Chlorination

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

Unit Parameters

There are two unit parameters:

- "chemical_name" - the name of the chemical for chlorination:
 - Required parameter
 - Must be in single ' ' or double " " quotes
- "dose" - chlorination dose [mg/L]:
 - Optional parameter
 - Default value is 9.5 mg/L (calculated below)

Capital Costs

Chlorination capital costs are a function of the applied chlorine dose and the flow using data in Table 3.23 of the Texas Water Development Board (2016).

The chlorine dose is calculated from:

$$D_{Cl} = C + rt + \frac{Ct}{t}$$

- C = Chlorine demand [mg/L]
- r = Chlorine decay rate [mg/Lhr]; default = 3
- t = Contact time [hr]; default = 1.5
- Ct = Desired Ct [mg*min/L]; default = 450

Then, using the data provided in Table 3.23, cost data is read in based on the dose and fit to the general form based on flow [MGD]:

$$C = aQ^b$$

In other words, values of a and b will depend on the dose used for the unit. Once a and b are known, the capital costs for chlorination are:

$$C_{Cl} = aQ_{in}^b$$

Electricity Intensity

Electricity intensity for chlorination is fixed at 5E-5 kWh/m³ and is taken from Bukhary, et al. (2019).

Assumptions

According to the reference, capital costs only include chemical feed equipment and assume there is sufficient contact time downstream of the chlorine feed point.

References

CAPITAL

[User's Manual for Integrated Treatment Train Toolbox - Potable Reuse \(IT3PR\) Version 2.0](#)

Steinle-Darling, E., Salveson, A., Russel, C., He, Q., Chiu, C., Lesan, D.

Texas Water Development Board

December 2016

ELECTRICITY

Bukhary, S., Batista, J., Ahmad, S. (2019).

An Analysis of Energy Consumption and the Use of Renewables for a Small Drinking Water Treatment Plant. *Water*, 12(1), 1-21.

Chlorination Module

```
class watertap3.wt_units.chlorination.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity.

Returns: Electricity intensity [kWh/m3]

```
    fixed_cap (unit_params)
```

Parameters: **unit_params** – Unit parameters from input sheet.

Returns:

```
    get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

Carbon Dioxide Addition

Unit Parameters

None

Capital Costs

The capital costs are a function of flow [MGD] in McGivney & Kawamura (2008):

$$C_{co2} = 0.464Q_{in}^{0.7}$$

Electricity Intensity

Electricity intensity is fixed at 0.01 kWh/m3 from _____.

References

CAPITAL

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

ELECTRICITY

Carbon Dioxide Addition Module

```
class watertap3.wt_units.co2_addition.UnitProcess (*args, **kwds)
```

```
    elect ()
        Electricity intensity.
```

Returns: Electricity intensity [kWh/m3]

```
    fixed_cap ()
        "unit_params" are the unit parameters passed to the model from the input sheet as a Python dictionary.
```

EXAMPLE: {'dose': 10}

Fixed capital for alum addition is a function of alum dose, alum solution flow, and the number of units.

Assumptions:

- Number of units = 2

Returns: Alum addition fixed capital cost [\$MM]

```
    get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

Coagulant Addition

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

Unit Parameters

There is one unit parameter:

- "dose" - dose of chemical [mg/L]
- Required parameter

Capital Costs

The coagulant flow S [gal/day] is used in a cost curve of the general form:

$$C = aS^b$$

For a single alum addition unit, $a = 15408$ and $b = 0.5479$. The full cost equation in WaterTAP3 is:

$$C_{coag} = 15408S^{0.5479}$$

This cost is then multiplied by the number of units and the EIF factor for the final FCI for the chemical addition. These parameters were determined by fitting data from FIGURE 5.5.15 - LIQUID ALUM FEED in McGivney & Kawamura (2008).

Electricity Intensity

Electricity intensity for chemical additions is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/day] of the solution necessary to achieve the desired dose:

$$M_{coag} = Q_{in}D_{coag}$$

- The volumetric flow S [gal/min] of the chemical solution, which incorporates the solution density [kg/m³]:

$$S = \frac{M_{coag}}{\rho_{coag}X_{coag}}$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

Then the electricity intensity is calculated as [kWh/m³]:

$$E_{coag} = \frac{0.746Sh}{3960\eta_{pump}\eta_{motor}Q_{in}}$$

Assumptions

- Number of units = 2

- Solution density [kg/m3] = 1360
- Ratio in solution = 50%
- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

Reference

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

Coagulant Addition Module

```
class watertap3.wt_units.coagulant_addition.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity.

Parameters:

- **lift_height** (*float*) – Lift height for pump [ft]
- **pump_eff** (*float*) – Pump efficiency
- **motor_eff** (*float*) – Motor efficiency

Returns: Electricity intensity [kWh/m3]

```
    fixed_cap (unit_params)
```

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'dose': 10}

Fixed capital for alum addition is a function of alum dose, alum solution flow, and the number of units.

Returns: Alum addition fixed capital cost [\$MM]

```
    get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

```
    solution_vol_flow ()
```

Chemical solution flow in gal/day

Parameters:

- **solution_density** (*float*) – Solution density [kg/m3]
- **ratio_in_solution** (*float*) – Ratio of chemical in solution

Returns: Alum solution flow [gal/day]

Crystallizer

The crystallizer represents a thermal crystallizer and is based on WaterReuse Foundation (2008).

Unit Parameters

None.

Capital Costs

Capital costs for crystallizers are a function of influent TDS, water recovery, and flow in.

The regression is based off of data found in Tables A2.1 and A2.3 found in WaterReuse Foundation (2008).

$$C_{cryst} = 1.41 - 7.11 \times 10^{-7}(C_{TDS}) + 1.45(x_{wr}) + 0.56(Q_{in})$$

Electricity Intensity

Electricity intensity is a function of the same variables and uses the same reference.

$$E_{cryst} = 56.7 + 1.83 \times 10^{-5}(C_{TDS}) - 9.47(x_{wr}) - 8.63 \times 10^{-4}(Q_{in})$$

Reference

Mickley, Michael C. (2008)
 “Survey of High-Recovery and Zero Liquid Discharge Technologies for Water Utilities”
 WaterReuse Foundation
 ISBN: 978-1-934183-08-3

Crystallizer Module

```
class watertap3.wt_units.crystallizer.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity for crystallizer. :param tds_in: TDS concentration in to crystallizer [mg/L] :type tds_in: float
 :param water_recovery: Water recovery for the brine concentrator :type water_recovery: float :param flow_in:
 Water flow in to crystallizer [m3/hr] :type flow_in: float :return: Electricity intensity [kWh/m3]

```
    fixed_cap ()
```

Fixed capital for crystallizer.

Parameters:

- **tds_in** (*float*) – TDS concentration in to crystallizer [mg/L]
- **water_recovery** (*float*) – Water recovery for the brine concentrator
- **flow_in** (*float*) – Water flow in to crystallizer [m3/hr]

Returns: Fixed capital cost for crystallizer [\$MM]

```
    get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

Deep Well Injection

Deep well injection is used to dispose of waste streams.

Unit Parameters

Deep well injection has two parameters:

- "lift_height" - lift height for injection pump [ft]:
 - Optional parameter
 - Default value is 400 ft
- "pipe_distance" - pipe distance from facility to deep well injection site [mi]
 - Required parameter

Capital Costs

Capital costs for deep well injection are based off of the costs for the Kay Baily Hutchinson (KBH) deep well injection site. Costing calculation is split into well construction and pipe construction. From the KBH data, well construction is \$16.9 MM. Piping cost assumes an 8 in diameter pipe, and is calculated as:

$$C_{pipe} = 0.28x_{pipe}$$

The total fixed cost is then calculated by scaling with KBH flow according to:

$$C_{dwi} = (C_{well} + C_{pipe}) \left(\frac{Q_{in}}{Q_{KBH}} \right)^{0.7}$$

Electricity Intensity

Electricity intensity for deep well injection is based off the pump used. The calculation includes:

- Lift height [ft]:
 h
- The pump and motor efficiencies:
 $\eta_{pump}, \eta_{motor}$
- And the influent flow in [gal/min] and [m³/hr]:
 Q_{gpm}, Q_{m3hr}

Then the electricity intensity is calculated as [kWh/m³]:

$$E_{dwi} = \frac{0.746 Q_{gpm} h}{3960 \eta_{pump} \eta_{motor} Q_{m3hr}}$$

Assumptions

- Lift height [ft] = 100

- Pump efficiency = 0.9
- Motor efficiency = 0.9

Reference

Module

```
class watertap3.wt_units.deep_well_injection.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity for deep well injection [kWh/m3]

Parameters: **lift_height** (*float*) – Lift height for pump [ft]

Returns: Electricity intensity [kWh/m3]

```
    fixed_cap (unit_params)
```

Fixed capital cost for deep well injection.

Parameters:

- **unit_params** (*dict*) – Input parameter dictionary from input sheet.

- **lift_height** (*float*) – Lift height for pump [ft]

- **pipe_distance** (*float*) – Piping distance to deep well injection site

Returns: Fixed capital cost for deep well injection [\$MM]

Electrodialysis Reversal

Unit Parameters

None.

Capital Costs

Capital costs for electrodialysis reversal are based on those from the Irwin case study.

$$C_{EDR} = 31 \frac{Q_{in}}{946}$$

Electricity Intensity

Electricity intensity is a function of TDS [mg/L] into the unit and based off of a regression of data from Baker (2004):

$$E_{EDR} = 0.2534 + 5.149 \times 10^{-4} C_{tds}$$

References

ELECTRICITY

Richard W. Baker (2004)
 “Membrane Technology and Applications, Second Edition”
 DOI:10.1002/0470020393

Electrodialysis Reversal Module

```
class watertap3.wt_units.electrodialysis_reversal.UnitProcess(*args, **kwds)
```

```
get_costing(unit_params=None, year=None)
    Initialize the unit in WaterTAP3.
```

Evaporation Ponds

Evaporation ponds are commonly used to further concentrate membrane brine to reduce the volume needed for disposal.

The generalized approach is as follows:

1. Calculate the evaporation rate based on environmental conditions.
2. Determine area required for evaporation ponds using a mass balance approach.
3. Calculate capital costs based off of area and costing approach selected by user.

Unit Parameters

The evaporation pond model draws from several sources and includes several optional parameters for user input. Further detail on these parameters is provided below:

- "approach" - costing approach used for the model (more detail is provided below on each approach):
 - Optional parameter
 - Options are "wt3", "zld", and "lenntech"
 - Default approach is "wt3" if no input given
 - Must be enclosed in double " " or single ' ' quotes
- "evap_method" - method used to calculate evaporation rate
 - Optional parameter
 - Two options are "turc" and "jensen"
 - Defaults to "jensen" if no input given
 - Must be enclosed in double " " or single ' ' quotes
- "air_temp" - air temperature for evaporation rate calculation [C]
 - Optional parameter
 - Default value is 20 deg C
 - Note: Must provide both "air_temp" and "solar_rad" together or default values for both will be used.
- "solar_rad" - incident solar radiation for evaporation rate calculation [mJ/cm2]
 - Optional parameter
 - Default value is 25 mJ/cm2
 - Note: Must provide both "air_temp" and "solar_rad" together or default values for both will be used.
- "humidity" - humidity for use in calculation of ratio to adjust pure water evaporation rate to saline water evaporation rate
 - Optional parameter
 - Default value is 0.5 (i.e. 50% humidity)

- Note: Must provide both "humidity" and "wind_speed" together or default values for both will be used.
- "wind_speed" - wind speed for use in calculation of ratio to adjust pure water evaporation rate [m/s]
 - Optional parameter
 - Default value is 5 m/s
 - Note: Must provide both "humidity" and "wind_speed" together or default values for both will be used.
- "liner_thickness" - thickness of liner used for calculation of cost per acre [mil]
 - Optional parameter
 - Default value is 50 mil
 - Note that 1 mil = 1/1000 inches
 - Note: Must provide "liner_thickness", "land_cost", "land_clearing_cost", and "dike_height" together or default values for all will be used.
- "land_cost" - cost to purchase land for evaporation pond [\$/acre]
 - Optional parameter
 - Default value is \$5,000/acre
 - Note: Must provide "liner_thickness", "land_cost", "land_clearing_cost", and "dike_height" together or default values for all will be used.
- "land_clearing_cost" - cost to clear land for evaporation pond [\$/acre]
 - Optional parameter
 - Default value is \$1,000/acre
 - Note: Must provide "liner_thickness", "land_cost", "land_clearing_cost", and "dike_height" together or default values for all will be used.
 - Typical costs for different types of land cover (Bureau of Reclamation, 2006):
 - brush = \$1,000/acre
 - sparsley wooded = \$2,000/acre
 - medium wooded = \$4,000/acre
 - heavily wooded = \$7,000/acre
- "dike_height" - height of dikes for evaporation pond [ft]
 - Default value is 8 ft
 - Typical dike heights are 4-12 ft (from Bureau of Reclamation reference)
 - Note: Must provide "liner_thickness", "land_cost", "land_clearing_cost", and "dike_height" together or default values for all will be used.

Evaporation Rate Calculation

The evaporation rate is R is dependent on site specific meteorological conditions and the salinity of the water. Salinity has the effect of lowering the evaporation rate, and because evaporation ponds are commonly used to concentrate brine streams, an evaporation rate calculated for pure water must be adjusted downward.

WaterTAP3 uses one of two regressions from each of the two references below to estimate the evaporation rate of pure water under the given meteorological conditions. Both are functions of air temperature T and solar irradiance J . For Jensen & Haise (1963) [mm/day]:

$$R_{pure} = R_{j\&h} = 0.41/(0.025T + 0.078)$$

And for Turc (1961) [mm/day]:

$$R_{pure} = R_{turb} = \frac{0.313T(J + 2.1)}{T + 15}$$

In WaterTAP3, the water in the pond is assumed to be saline. The evaporation rate of saline water can be estimated to be 70% that of pure water (WateReuse Foundation, 2008). Thus, this calculated evaporation rate is multiplied by 0.7 to arrive at the estimated evaporation rate for saline water:

$$R_{saline} = 0.7R_{pure}$$

Pond Area Calculation

For mass balance purposes in WaterTAP3, the flow of evaporated water is the considered to be the flow out of the unit. To accomodate a given water recovery, the area of the pond is calculated as:

$$A_{pond} = \frac{Q_{out}}{R_{saline}}$$

Capital Costs

The user can choose one of three costing approaches for evaporation ponds in WaterTAP3 that can be provided as an option in `unit_params` under "approach":

1. "wt3" - default approach if the user does not provide one. Incorporates an adjusted pond area and more in-depth costing function. Based on Bureau of Reclamation reference below.
2. "zld" - only considers area and assumes \$0.3M/acre. Based on WateReuse reference below.
3. "lenntech" - capital cost determined purely from flow. Based on Lenntech reference below.

WT3 Approach:

The WT3 approach is the default approach and uses a regression for total pond area from the Bureau of Reclamation reference below. After calculation of the required pond area based on flow above, if this approach is used the pond area is adjusted upwards to incorporate the additional area needed for dikes:

$$A_{adj} = 1.2(A_{pond})(1 + 0.155 \frac{h_{dike}}{\sqrt{A_{pond}}})$$

Then, the cost per acre [\$/acre] is determined that incorporates "liner_thickness", "land_cost", "land_clearing_cost", and "dike_height":

$$C_{acre} = 5406 + 465(z_{liner}) + 1.07(c_{land}) + 0.931(c_{clear}) + 217.5(h_{dike})$$

Thus, using this approach capital costs for evaporation ponds are calculated as:

$$C_{evap} = A_{adj}C_{acre}$$

ZLD Approach:

The ZLD approach is named from the WaterReuse document it was adapted from. Using this approach, the unadjusted pond area is used. The cost per acre is assumed to be \$0.3MM. Thus, the capital costs are calculated as:

$$C_{evap} = A_{pond} \times 0.3$$

Lenntech Approach:

This approach is based entirely on flow in [m3/d] to the evaporation pond and does not include the calculation for evaporation pond area. It assumes an evaporation rate of 1 m/yr:

$$C_{evap} = 0.031Q_{in}^{0.7613}$$

Note that while the reference mentions salt concentrations, land and earthwork costs, and liner costs, it is unclear how these are incorporated into the cost curve above.

Electricity Cost

WaterTAP3 does not include any electricity intensity for evaporation ponds.

References**EVAPORATION RATE**

Turc, L. (1961)

“Water requirements assessment of irrigation, potential evapotranspiration: Simplified and updated climatic formula.”

Annales Agronomiques, 12, 13-49.

Jensen, M.E., Haise, H.R. (1963)

“Estimating evapotranspiration from solar radiation.”

Proceedings of the American Society of Civil Engineers

Journal of the Irrigation and Drainage Division, vol. 89, pp. 15-41.

COSTING

U.S. Bureau of Reclamation (2006)

Mickley, Michael C.

“Membrane Concentrate Disposal: Practices and Regulation”

Chapter 10: Evaporation Pond Disposal

WaterReuse Foundation (2008)

Mickley, Michael C.

“Survey of High-Recovery and Zero Liquid Discharge Technologies for Water Utilities”

ISBN: 978-1-934183-08-3

Lenntech.com

<https://www.lenntech.com/Data-sheets/Brine-Evaporation-Ponds.pdf>

Evaporation Pond Module

```
class watertap3.wt_units.evaporation_pond.UnitProcess (*args, **kwds)
```

```
    elect ()
```

WaterTAP3 has no electricity intensity associated with evaporation ponds.

evaporation_rate (unit_params, t)

Calculation of evaporation rate [gpm/acre]

Parameters:

- **unit_params** (*dict*) – Input dictionary from input sheet.
- **t** (*int*) – Time indexing variable to use in Var()
- **evap_method** (*str*) – Evaporation rate method
- **humidity** (*float*) – Humidity expressed as decimal for evaporation rate calculation
- **wind_speed** (*float*) – Wind speed for evaporation rate calculation [m/s]
- **air_temp** (*float*) – Air temperature for evaporation rate calculation [C]
- **solar_rad** (*float*) – Incident solar radiation for evaporation rate calculation [mJ/m2]

Returns:

evaporation_rate_regress (t)

NOTE: THIS FUNCTION IS NOT USED IN THE CURRENT RELEASE OF WaterTAP3

Calculates evaporation rate based on air temperature, TDS in, humidity, and wind speed.

fixed_cap (unit_params)

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

Evaporation ponds can have many unit_params

EXAMPLE: {'approach': 'wt3', 'area': 3500, 'humidity': 0.75, 'wind_speed': 10}

Parameters:

- **unit_params** (*dict*) – Input parameter dictionary from input sheet.
- **liner_thickness** (*float*) – Liner thickness [mil]
- **land_cost** (*float*) – Cost of land for evaporation pond [\$/acre]
- **land_clearing_cost** (*float*) – Cost to clear land for evaporation pond [\$/acre]
- **dike_height** (*float*) – Height of dikes [ft]

Returns: Fixed capital cost for evaporation ponds [\$MM]

get_costing (unit_params=None, year=None)

Initialize the unit in WaterTAP3.

Ferric Chloride Addition

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

Unit Parameters

There is one unit parameter:

- "dose" - dose of chemical [mg/L]
- Required parameter

Capital Costs

The ferric chloride solution flow S [gal/day] is used in a cost curve of the general form:

$$C = aS^b$$

For a single ferric chloride addition unit, $a = 34153$ and $b = 0.319$. The full cost equation in WaterTAP3 is:

$$C_{ferric} = 34153S^{0.319}$$

This cost is then multiplied by the number of units and the EIF factor for the final FCI for the chemical addition. These parameters were determined by fitting data from FIGURE 5.5.13 - FERRIC CHLORIDE FEED 42% SOLUTION in McGivney & Kawamura (2008).

Electricity Intensity

Electricity intensity for chemical additions is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/day] of the solution necessary to achieve the desired dose:

$$M_{ferric} = Q_{in}D_{ferric}$$

- The volumetric flow S [gal/min] of the chemical solution, which incorporates the solution density [kg/m3]:

$$S = \frac{M_{ferric}}{\rho_{ferric}X_{ferric}}$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

Then the electricity intensity is calculated as [kWh/m3]:

$$E_{ferric} = \frac{0.746Sh}{3960\eta_{pump}\eta_{motor}Q_{in}}$$

Assumptions

- Number of units = 2

- Solution density [kg/m³] = 1460
- Ratio in solution = 42%
- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

Reference

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

Ferric Chloride Addition Module

```
class watertap3.wt_units.ferric_chloride_addition.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity.

Parameters:

- **lift_height** (*float*) – Lift height for pump [ft]
- **pump_eff** (*float*) – Pump efficiency
- **motor_eff** (*float*) – Motor efficiency

Returns: Electricity intensity [kWh/m³]

```
fixed_cap (unit_params)
```

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'dose': 10}

Fixed capital for ferric chloride addition is a function of ferric dose, ferric solution flow, and the number of units.

Returns: Ferric addition fixed capital cost [\$MM]

```
get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

```
solution_vol_flow ()
```

Chemical solution flow in gal/day

Parameters:

- **solution_density** (*float*) – Solution density [kg/m³]
- **ratio_in_solution** (*float*) – Ratio of chemical in solution

Returns: Ferric solution flow [gal/day]

Fixed Bed Gravity Basin

The fixed bed gravity basin unit in WaterTAP3 is based off of a regression of several runs using EPA's Work Breakdown Structure-Based (WBS EPA) model. These models each have some "standard designs" that make default assumptions and span a range of flows 0.03 MGD to 75 MGD (USEPA (2019)).

The approach for the WT3 model is to regress the total capital cost output from the EPA model vs. flow for each of the EPA model standard designs with the fixed bed gravity basin option. A similar approach is taken to determine electricity intensity.

Both sets of data are fit to a power curve. Then a and b are used with the flow for the model to determine capital and electricity intensity.

Unit Parameters

None.

Capital Costs

The `cap_total` column is read in from the `cost_curves.csv` and is fit to a power curve. Then a and b from that regression is used with the unit flow [m³/hr] to determine capital costs:

$$C_{fbgb} = aQ_{in}^b$$

Electricity Intensity

The `electricity_flow` column is read in from the `cost_curves.csv` and is fit to a power curve. Then a and b from that regression is used with the unit flow [m³/hr] to determine electricity intensity:

$$E_{fbgb} = aQ_{in}^b$$

Chemical Use

The WBS EPA model includes costs for acetic acid, phosphoric acid, iron chloride, activated carbon, sand, and anthracite. The average is taken for the range of flows and assumed to be the "dose" for the particular chemical/material.

References

US Environmental Protection Agency (2019)

"Work Breakdown Structure-Based Cost Model for Biological Drinking Water Treatment"

<https://www.epa.gov/sites/production/files/2019-07/documents/wbs-biotreat-documentation-june-2019.pdf>

<https://www.epa.gov/sdwa/drinking-water-treatment-technology-unit-cost-models>

Unit Template

```
class watertap3.wt_units.fixed_bed_gravity_basin.UnitProcess(*args, **kwargs)
```

```
    elect ()
```

```
        Electricity intensity for fixed bed gravity basin [kWh/m3] :return:
```

```
    fixed_cap ()
```

```
        Returns: Fixed capital for fixed bed gravity basin [$MM]
```

```
    get_costing (unit_params=None, year=None)
```

```
        Initialize the unit in WaterTAP3.
```

Fixed Bed Pressure Vessel

The fixed bed gravity basin unit in WaterTAP3 is based off of a regression of several runs using EPA's Work Breakdown Structure-Based (WBS EPA) model. These models each have some "standard designs" that make default assumptions and span a range of flows 0.03 MGD to 75 MGD (USEPA (2019)).

The approach for the WT3 model is to regress the total capital cost output from the EPA model vs. flow for each of the EPA model standard designs with the fixed bed pressure vessel option. A similar approach is taken to determine electricity intensity.

Both sets of data are fit to a power curve. Then a and b are used with the flow for the model to determine capital and electricity intensity.

Unit Parameters

None

Capital Costs

The `cap_total` column is read in from the `cost_curves.csv` and is fit to a power curve. Then a and b from that regression is used with the unit flow [m³/hr] to determine capital costs:

$$C_{fbpv} = aQ_{in}^b$$

Electricity Intensity

The `electricity_flow` column is read in from the `cost_curves.csv` and is fit to a power curve. Then a and b from that regression is used with the unit flow [m³/hr] to determine electricity intensity:

$$E_{fbpv} = aQ_{in}^b$$

Chemical Use

The WBS EPA model includes costs for acetic acid, phosphoric acid, iron chloride, activated carbon, sand, and anthracite. The average is taken for the range of flows and assumed to be the "dose" for the particular chemical/material.

References

US Environmental Protection Agency (2019)

"Work Breakdown Structure-Based Cost Model for Biological Drinking Water Treatment"

<https://www.epa.gov/sites/production/files/2019-07/documents/wbs-biotreat-documentation-june-2019.pdf>

<https://www.epa.gov/sdwa/drinking-water-treatment-technology-unit-cost-models>

Unit Template

```
class watertap3.wt_units.fixed_bed_gravity_basin.UnitProcess(*args, **kwargs)
```

```
    elect ()
```

```
        Electricity intensity for fixed bed gravity basin [kWh/m3] :return:
```

```
    fixed_cap ()
```

```
        Returns: Fixed capital for fixed bed gravity basin [$MM]
```

```
    get_costing (unit_params=None, year=None)
```

```
        Initialize the unit in WaterTAP3.
```

GAC - Gravity

The gravity GAC unit in WaterTAP3 is based off of a regression of several runs using EPA's Work Breakdown Structure-Based (WBS EPA) model. These models each have some "standard designs" that make default assumptions and span a range of flows 0.03 MGD to 75 MGD (USEPA (2019)).

The approach for the WT3 model is to regress the total capital cost output from the EPA model vs. flow for each of the EPA model standard designs with the gravity GAC option. A similar approach is taken to determine electricity intensity.

Both sets of data are fit to a power curve. Then a and b are used with the flow for the model to determine capital and electricity intensity.

Unit Parameters

Anion exchange has one parameter:

- "ebct" - empty bed contact time for the unit [min]:
 - Required parameter
 - There are different costing data for 0-60 min and >60 min

Capital Costs

The `cap_total` column is read in from the `cost_curves.csv` based on user-defined EBCT and is fit to a power curve. Then a and b from that regression is used with the unit flow [m3/hr] to determine capital costs:

$$C_{gac} = aQ_{in}^b$$

Electricity Intensity

The `electricity_flow` column is read in from the `cost_curves.csv` based on user-defined EBCT and is fit to a power curve. Then a and b from that regression is used with the unit flow [m3/hr] to determine electricity intensity:

$$E_{gac} = aQ_{in}^b$$

Chemical Use

The WBS EPA model includes costs for activated carbon. The average is taken for the range of flows and assumed to be the "dose" for the particular chemical/material.

References

US Environmental Protection Agency (2017)
 "Work Breakdown Structure-Based Cost Model for Granular Activated Carbon Drinking Water Treatment"
https://www.epa.gov/sites/production/files/2019-03/documents/wbs-gac-documentation-dec-2017_v2.pdf
<https://www.epa.gov/sdwa/drinking-water-treatment-technology-unit-cost-models>

Unit Template

```
class watertap3.wt_units.gac_pressure_vessel.UnitProcess(*args, **kwargs)

    elect ()
        Electricity intensity for pressure vessel GAC [kWh/m3]:return:

    fixed_cap (unit_params)
```

Returns: Fixed capital for pressure vessel GAC [\$MM]

get_costing(unit_params=None, year=None)
Initialize the unit in WaterTAP3.

GAC - Pressure Vessel

The GAC pressure vessel in WaterTAP3 is based off of a regression of several runs using EPA's Work Breakdown Structure-Based (WBS EPA) model. The WBS EPA models each have some "standard designs" that make default assumptions (USEPA (2019)) and span a range of flows 0.03 MGD to 75 MGD.

The approach for the WT3 model is to regress the total capital cost output from the EPA model vs. flow for each of the EPA model standard designs with the fixed bed gravity basin option. A similar approach is taken to determine electricity intensity.

Both sets of data are fit to a power curve. Then a and b are used with the flow for the model.

Unit Parameters

Anion exchange has one parameter:

- "ebct" - empty bed contact time for the unit [min]:
 - Required parameter
 - There are different costing data for 0-60 min and >60 min

Capital Costs

The `cap_total` column is read in from the `cost_curves.csv` based on user-defined EBCT and is fit to a power curve. Then a and b from that regression is used with the unit flow [m3/hr] to determine capital costs:

$$C_{gac} = aQ_{in}^b$$

Electricity Intensity

The `electricity_flow` column is read in from the `cost_curves.csv` based on user-defined EBCT and is fit to a power curve. Then a and b from that regression is used with the unit flow [m3/hr] to determine electricity intensity:

$$E_{gac} = aQ_{in}^b$$

Chemical Use

The WBS EPA model includes costs for activated carbon. The average is taken for the range of flows and assumed to be the "dose" for the particular chemical/material.

References

US Environmental Protection Agency (2017)
 "Work Breakdown Structure-Based Cost Model for Granular Activated Carbon Drinking Water Treatment"
https://www.epa.gov/sites/production/files/2019-03/documents/wbs-gac-documentation-dec-2017_v2.pdf
<https://www.epa.gov/sdwa/drinking-water-treatment-technology-unit-cost-models>

Unit Template

```
class watertap3.wt_units.gac_pressure_vessel.UnitProcess(*args, **kwargs)

    elect ()
        Electricity intensity for pressure vessel GAC [kWh/m3] :return:

    fixed_cap (unit_params)
```

Returns: Fixed capital for pressure vessel GAC [\$MM]

get_costing(unit_params=None, year=None)
Initialize the unit in WaterTAP3.

Hydrochloric Acid Addition

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

Unit Parameters

There is one unit parameter:

- "dose" - dose of chemical [mg/L]
- Required parameter

Capital Costs

The hydrochloric acid solution flow S [gal/day] is used in a cost curve of the general form:

$$C = aS^b$$

For a single hydrochloric acid addition unit, $a = 900.97$ and $b = 0.6179$. The full cost equation in WaterTAP3 is:

$$C_{HCl} = 900.97S^{0.6179}$$

This cost is then multiplied by the number of units and the EIF factor for the final FCI for the chemical addition. These parameters were determined by fitting data from FIGURE 5.5.11 - SULFURIC ACID FEED in McGivney & Kawamura (2008).

Electricity Intensity

Electricity intensity for chemical additions is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/day] of the solution necessary to achieve the desired dose:

$$M_{HCl} = Q_{in}D_{HCl}$$

- The volumetric flow S [gal/min] of the chemical solution, which incorporates the solution density [kg/m³]:

$$S = \frac{M_{HCl}}{\rho_{HCl}}$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

Then the electricity intensity is calculated as [kWh/m³]:

$$E_{HCl} = \frac{0.746Sh}{3960\eta_{pump}\eta_{motor}Q_{in}}$$

Assumptions

- Number of units = 2

- Solution density [kg/m3] = 1490
- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

Reference

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

Hydrochloric Acid Addition Module

```
class watertap3.wt_units.hydrochloric_acid_addition.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity.

Parameters:

- **lift_height** (*float*) – Lift height for pump [ft]
- **pump_eff** (*float*) – Pump efficiency
- **motor_eff** (*float*) – Motor efficiency

Returns: Electricity intensity [kWh/m3]

```
    fixed_cap (unit_params)
```

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'dose': 10}

Returns: HCl addition fixed capital cost [\$MM]

```
    get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

```
    solution_vol_flow ()
```

Chemical solution flow in gal/day

Parameters: **solution_density** (*float*) – Solution density [kg/m3]

Returns: HCl solution flow [gal/day]

Iron & Manganese Removal

The Fe/Mn removal unit in WaterTAP3 is based off of the dual media filtration schematic in the lenntech reference using costing data in McGivney & Kawamura (2008).

Unit Parameters

None

Capital Costs

The coagulation/flocculation unit in WaterTAP3 includes costing for filtration media, backwash system, and air blower.

$$C_{Fe/Mn} = (C_{filt} + C_{bw} + nC_{blow}) \frac{Q_{in}^{0.7}}{4732}$$

Filtration capital is a function of media surface area and calculated with:

$$C_{filt} = 21377 + 38.319A$$

Backwash capital is also a function of media surface area:

$$C_{bw} = 92947 + 292.44A$$

The blower capital is assumed.

Assumptions

There are six units:

$$n = 6$$

The filter surface area is 6243 ft²:

$$A = 6243$$

The air/water ratio in the blower is 0.001 [v/v]:

$$r = 0.001$$

The capital for the air blower is \$100,000:

$$C_{blow} = 100000$$

Electricity Intensity

The total electricity intensity for Fe/Mn removal is from the blower [kWh/m³]:

$$E_{Fe/Mn} = \frac{P_{blow}}{Q_{in}}$$

Where blower power is calculated with [hp]:

$$p_{rm} = 147.8q_{air}$$

And the air flow rate is [m³/hr]:

$$q_{air} = Q_{in}r$$

References

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

Schema of an iron removal system
<https://www.lenntech.com/schema-of-an-iron-removal-system.htm>

Unit Template

```
class watertap3.wt_units.iron_and_manganese_removal.UnitProcess(*args, **kwds)

    get_costing(unit_params=None, year=None)
        Initialize the unit in WaterTAP3.
```

Landfill

Unit Basics

This unit is a terminal unit in WaterTAP3 and represents the cost of sending residual solids to landfill.

Unit Parameters

None.

Capital Costs

Landfill costs are a function of mass flow into the unit:

$$C_{lf} = \frac{M_{in}}{100000} 0.7$$

The mass flow is calculated by first calculating the total concentration flowing into the unit:

$$C_{in} = \sum_i^n c_i$$

Then, we estimate the density of the solution [kg/m³]:

$$\rho_{in} = 0.6312(C_{in}) + 997.86$$

Mass flow [kg/hr] is determined with:

$$M_{in} = \rho_{in} Q_{in}$$

Electricity Intensity

There are no electricity costs associated with landfill in WaterTAP3.

References

Landfill Module

```
class watertap3.wt_units.landfill1.UnitProcess (*args, **kwds)
```

```
    get_costing (unit_params=None, year=None)
        Initialize the unit in WaterTAP3.
```

Landfill ZLD

Unit Basics

This unit is a terminal unit in WaterTAP3 and represents the cost of sending residual solids to landfill.

Unit Parameters

None.

Capital Costs

Landfill ZLD costs are a function of mass flow into the unit:

$$C_{lf} = \frac{M_{in}}{302096} 0.7$$

The mass flow is calculated by first calculating the total concentration flowing into the unit:

$$C_{in} = \sum_i^n c_i$$

Then, we estimate the density of the solution [kg/m³]:

$$\rho_{in} = 0.6312(C_{in}) + 997.86$$

Mass flow [kg/hr] is determined with:

$$M_{in} = \rho_{in} Q_{in}$$

Electricity Intensity

There are no electricity costs associated with landfill ZLD in WaterTAP3.

References

Mickley, Michael C. (2008)
 “Survey of High-Recovery and Zero Liquid Discharge Technologies for Water Utilities”
 WaterReuse Foundation
 ISBN: 978-1-934183-08-3

Landfill ZLD Module

```
class watertap3.wt_units.landfill_zld.UnitProcess(*args, **kwds)
```

```
    get_costing(unit_params=None, year=None)
        Initialize the unit in WaterTAP3.
```

Lime Addition

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

Unit Parameters

- "lime" - dose of lime [mg/L]
- Required parameter

Capital Costs

The lime solution mass flow M [lb/day] is used in a cost curve of the general form:

$$C = aM^b$$

For a single lime addition unit, $a = 16972$ and $b = 0.5435$. The full cost equation in WaterTAP3 is:

$$C_{lime} = 16972S^{0.5435}$$

This cost is then multiplied by the number of units and the EIF factor for the final FCI for the chemical addition. These parameters were determined by fitting data from FIGURE 5.5.9 - LIME FEED in McGivney & Kawamura (2008).

Electricity Intensity

Electricity intensity for chemical additions is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/hr] of the solution necessary to achieve the desired dose:

$$M_{lime} = Q_{in}D_{lime}$$

- The volumetric flow S [m³/hr] of the chemical solution, which incorporates the solution density [kg/m³]:

$$S = \frac{M_{lime}}{\rho_{lime}}$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

Then the electricity intensity is calculated as [kWh/m³]:

$$E_{lime} = \frac{0.746Sh}{3960\eta_{pump}\eta_{motor}Q_{in}}$$

Assumptions

- Number of units = 2
- Solution density [kg/m³] = 1250
- Lift height [ft] = 100

- Pump efficiency = 0.9
- Motor efficiency = 0.9

Reference

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

Lime Addition Module

```
class watertap3.wt_units.lime_addition.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity.

Parameters:

- **lift_height** (*float*) – Lift height for pump [ft]
- **pump_eff** (*float*) – Pump efficiency
- **motor_eff** (*float*) – Motor efficiency

Returns: Electricity intensity [kWh/m3]

```
    fixed_cap (unit_params)
```

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'lime': 10}

Fixed capital for lime addition is a function of lime dose, lime solution flow, and the number of units.

Parameters: **lime** (*float*) – Lime dose [mg/L]

Returns: Lime addition fixed capital cost [\$MM]

```
    get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

```
    solution_vol_flow ()
```

Chemical solution flow in gal/day

Parameters: **solution_density** (*float*) – Solution density [kg/m3]

Returns: Lime solution vol. flow [gal/day], Lime solution mass flow [lb/day]

Lime Softening

Unit Parameters

There is one unit parameter:

* "lime" - lime dose for unit [mg/L]:

- Required parameter

Capital Costs

The capital costs are a function of flow [m³/hr] with cost curve parameters from McGivney & Kawamura (2008):

$$C_{lime} = 0.0704Q_{in}^{0.7306}$$

Electricity Intensity

Electricity intensity for lime softening is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/hr] of the solution necessary to achieve the desired dose:

$$M_{lime} = Q_{in}D_{lime}$$

- The volumetric flow S [m³/hr] of the chemical solution, which incorporates the solution density [kg/m³]:

$$S = \frac{M_{lime}}{\rho_{lime}}$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

Then the electricity intensity is calculated as [kWh/m³]:

$$E_{lime} = \frac{0.746Sh}{3960\eta_{pump}\eta_{motor}Q_{in}}$$

Assumptions

- Solution density [kg/m³] = 1250
- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

References

Minnesota Rural Water Association, Chapter 16 Lime Softening

<https://www.mrwa.com/WaterWorksMnl/Chapter%2016%20Lime%20Softening.pdf>

Lime Softening Module

```
class watertap3.wt_units.lime_softening.UnitProcess (*args, **kwds)
```

elect ()

Electricity intensity for chemical additions is a function of lift height, pump efficiency, and motor efficiency.

Returns: Electricity intensity [kWh/m3]

fixed_cap (unit_params)

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'lime': 10}

Fixed capital for lime addition is a function of lime dose, lime solution flow, and the number of units.

Parameters: **lime** (*float*) – Lime dose [mg/L]

Returns: Lime addition fixed capital cost [\$MM]

get_costing (unit_params=None, year=None)

Initialize the unit in WaterTAP3.

solution_vol_flow ()

Determine alum solution flow rate in gal / day

Returns: Lime solution flow [gal/day]

Media Filtration

Unit Parameters

None

Capital Costs

Capital costs for media filtration includes the cost of the filter and the cost of the backwash system (McGivney & Kawamura (2008)).

$$C_{mf} = C_{filter} + C_{bw}$$

The cost of the filter is a function of the surface area and the number of units:

$$C_{filter} = (21377 + 38.319A)n$$

With the filter surface area calculated with the assumed filtration rate [ft²]:

$$A = \frac{Q_{in}}{v}$$

The cost of the backwash system is also a function of filter surface area:

$$C_{bw} = 92947 + 292.44A$$

Assumptions

There are six units:

$$n = 6$$

The filtration rate is 10 m/hr:

$$v = 10$$

Electricity Intensity

Electricity intensity is fixed at 0.00015 kWh/m³ from Bukhary et al. (2019).

References

CAPITAL

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

ELECTRICITY

Bukhary, S., et al. (2019).
“An Analysis of Energy Consumption and the Use of Renewables for a Small Drinking Water Treatment Plant.”
Water 12(1).

Media Filtration Module

```
class watertap3.wt_units.media_filtration.UnitProcess (*args, **kwds)
```

`get_costing` (unit_params=None, year=None)
Initialize the unit in WaterTAP3.

Microfiltration

Unit Parameters

None

Capital Costs

Capital costs for microfiltration is based entirely on flow [MGD] from Table 3.20 in the Texas Water Board reference:

$$C_{MF} = 2.5Q_{in}$$

Electricity Intensity

Electricity intensity is fixed at 0.18 kWh/m³ from Plappally & Lienhard (2012).

References

CAPITAL

User's Manual for Integrated Treatment Train Toolbox - Potable Reuse (IT3PR) Version 2.0
 Steinle-Darling, E., Salveson, A., Russel, C., He, Q., Chiu, C., Lesan, D.
 Texas Water Development Board
 December 2016

ELECTRICITY

Plappally, A. K. and J. H. Lienhard V (2012).
 "Energy requirements for water production, treatment, end use, reclamation, and disposal."
Renewable and Sustainable Energy Reviews 16(7): 4818-4848.

Microfiltration Module

```
class watertap3.wt_units.microfiltration.UnitProcess(*args, **kwargs)
```

```
    get_costing(unit_params=None, year=None)
        Initialize the unit in WaterTAP3.
```

Municipal Drinking (Distribution)

This unit is a terminal unit in WaterTAP3 and represents the initial cost to get product water to the distribution system.

Unit Parameters

None

Capital Costs

The capital costs are a function of flow [MGD]:

$$C_{md} = 0.0403Q_{in}^{0.8657}$$

Electricity Intensity

Electricity intensity for municipal drinking is based off the pump used. The calculation includes:

- Lift height [ft]:

$$h$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

- And the influent flow in [gal/min] and [m3/hr]:

$$Q_{gpm}, Q_{m3hr}$$

Then the electricity intensity is calculated as:

$$E_{bw} = \frac{0.746Q_{gpm}h}{3960\eta_{pump}\eta_{motor}Q_{m3hr}}$$

Assumptions

- Lift height [ft] = 300
- Pump efficiency = 0.9
- Motor efficiency = 0.9

References

Voutchkov, N. (2018).
Desalination Project Cost Estimating and Management.
<https://doi.org/10.1201/9781351242738>

Municipal Drinking (Distribution) Module

```
class watertap3.wt_units.municipal_drinking.UnitProcess (*args, **kwds)
```

```
    get_costing (unit_params=None, year=None)
        Initialize the unit in WaterTAP3.
```

Ozone + AOP

The Ozone unit in WaterTAP3 is used to calculate costs for both Ozone and Ozone+AOP systems.

Unit Parameters

Unit parameters are read in from the `unit_params` dictionary given in the input sheet.

- "aop" - boolean to indicate if the unit includes AOP costs:
 - Required parameter
 - If `True`, oxidant costs are included
 - If `False`, oxidant costs are excluded
- "contact_time" - contact time with ozone [min]
 - Required parameter
- "ct" - Concentration * time (Ct) target [mg/(L*min)]
 - Required parameter
- "mass_transfer" - mass transfer coefficient for ozone contactor
 - Required parameter
- "chemical_name" - oxidant chemical name
 - Required parameter if `aop = True`
 - Must match exactly the chemical name in `chemical_name.csv`

Capital Costs

The Ozone/AOP unit in WaterTAP3 is used for both Ozone only units and Ozone+AOP units. For this reason costs are broken up between the ozone system and the oxidant injection system.

Ozone Capital Costs

Ozone capital costs in WaterTAP3 are a function of the ozone contact time, Ct, ozone mass transfer, ozone dose, TOC concentration, flow in, and the ozone/TOC ratio.

TOC concentration and flow in are taken from the model. Contact time, mass transfer, and Ct are taken from user input. Ozone demand is calculated from Texas Water Development Board (2016):

$$D_{O3} = \frac{t_{O3} C_{TOC} Ct}{\eta_{O3}}$$

Ozone dose is then used in a regression derived from data found in Table 3.24 from Texas Water Development Board (2016).

Oxidant Capital Costs

The unit can accept any chemical name and dose, but the cost curve used was developed specifically for Hydrogen Peroxide in Texas Water Development Board (2016):

$$C_{ox} = 12285_{ox}^{0.2277}$$

Oxidant costs are calculated are based off the oxidant solution flow [lb/day]:

$$S_{ox} = Q_{in}D_{ox}$$

Where the oxidant dose is calculated with [kg/m3]:

$$D_{ox} = 0.5Xc_{toc}$$

And the ozone/TOC ratio X is calculated with:

$$X = 1 + \frac{t_{O_3}Ct}{c_{toc}}$$

Electricity Intensity

Electricity intensity is a function of the ozone flow [lb/day] and water flow into unit [m3/hr] from Mundy et al. (2018):

$$E_{O_3} = 5 \frac{Q_{O_3}}{Q_{in}}$$

References

COSTING

User's Manual for Integrated Treatment Train Toolbox - Potable Reuse (IT3PR) Version 2.0

Steinle-Darling, E., Salvesson, A., Russel, C., He, Q., Chiu, C., Lesan, D.

Texas Water Development Board

December 2016

ELECTRICITY

Mundy, B., et al. (2018).

"A Review of Ozone Systems Costs for Municipal Applications. Report by the Municipal Committee – IOA Pan American Group."

Ozone: Science & Engineering 40(4): 266-274.

Ozone + AOP Module

```
class watertap3.wt_units.ozone_aop.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity for Ozone/Ozone AOP unit.

Parameters: **ozone_flow** (*float*) – Flow of ozone from ozone generator [lb/day]

Returns: Ozone/Ozone AOP electricity intensiy [kWh/m3]

```
    fixed_cap (unit_params)
```

Fixed capital for Ozone/Ozone AOP unit.

Parameters:

- **unit_params** (*dict*) – Input parameter dictionary from input sheet
- **toc_in** (*float*) – TOC concentration into unit [mg/L]
- **aop** (*bool*) – Boolean to indicate if unit is AOP or not.
- **contact_time** (*float*) – Ozone contact time [min]
- **ct** (*float*) – Concentration * time (Ct) target [mg/(L*min)]
- **mass_transfer** (*float*) – Mass transfer coefficient
- **chemical_name** (*str*) – Name of oxidant used if unit is AOP

Returns: Fixed capital cost for Ozone/Ozone AOP [\$MM]

get_costing (unit_params=None, year=None)

Initialize the unit in WaterTAP3.

solution_vol_flow ()

Determine oxidant solution flow rate [gal/day]

Returns: Oxidant solution flow [gal/day]

Reverse Osmosis

The capital cost, O&M costs, electricity consumption, TDS removal, and water recovery of the reverse osmosis unit model are based on a basic representation of the physical performance of the process. This includes calculations of osmotic pressure and mass balance across the membrane. The feed pressure (i.e. pump power required) and membrane area are estimated based on optimizing for LCOW, unless specified otherwise. All other variables in the reverse osmosis model have default assumptions, provided below, that factor into calculating the water recovery, TDS removal, pure water flux, and osmotic pressure.

Unit Parameters

The user only has to specify whether or not to include an energy recovery device (ERD).

- "erd" - whether or not to include an ERD
 - Required parameter
 - Options are "yes" or "no"

Unit Constraints

The reverse osmosis unit in WaterTAP3 is broken up into a feed, permeate, and retentate IDAES `Block()` to track the different mass, flow, and pressure changes for each stream.

Mass and Pressure Constraints

The unit only uses the TDS concentration into the unit to estimate the total concentration of mass going into each reverse osmosis unit.

Thus, for each block (feed, permeate, retentate), some or all of the following mass and/or energy balance equations apply.

The total concentration of TDS flowing into the unit is [kg/m³]:

$$c_{tot} = 0.6312c_{tds} + 997.86$$

The mass concentration of water into the block is [kg/m³]:

$$c_w = c_{tot} - c_{tds}$$

The mass flow of water into the block is [kg/s]:

$$M_w = c_w Q_{in}$$

The mass flow of TDS into the block is [kg/s]:

$$M_{TDS} = c_{TDS} Q_{in}$$

The mass fraction of TDS into the block is:

$$X_{TDS} = \frac{M_{TDS}}{M_w + M_{TDS}}$$

The mass fraction of water into the block is:

$$X_w = \frac{M_w}{M_w + M_{TDS}}$$

The overall mass balance around water flow is:

$$M_{w,f} = M_{w,p} + M_{w,r}$$

And the overall mass balance around TDS is:

$$M_{tds,f} = M_{tds,p} + M_{tds,r}$$

Feed Block

The osmotic coefficient is calculated according to Bartholomew & Mauter (2019):

$$\sigma = 4.92x_{TDS}^2 + 0.0889x_{TDS} + 0.918$$

Then the osmotic pressure [bar] is calculated as:

$$P_{osm} = \frac{8.45 \times 10^7 \sigma x_{TDS}}{1 - x_{TDS}}$$

The flux is calculated from the feed stream as:

$$J_w = \rho_w K_w (P_{feed} - P_{atm} - 0.5P_d) - (0.5(P_{osm,f} + P_{osm,r}))$$

The pressure drop is assumed to be 3 bar.

Permeate Stream

The equation for mass into permeate is slightly different than for feed or retentate streams Bartholomew & Mauter (2019):

$$c_{tot,p} = 756c_{tds} \times 10^{-6} + 995$$

The concentration coming out of the unit is:

$$c_{tds,out} = c_{tot,p} x_{tds,p}$$

The mass flow of water is determined from the membrane area and pure water flux:

$$M_{w,p} = J_w A$$

The mass flow of TDS is determined from the membrane area and the salt permeability coefficient:

$$M_{TDS,p} = 0.5AK_s(c_{tds,f} + c_{tds,r})$$

The pressure on the permeate side is assumed to be atmospheric:

$$P_p = P_{atm}$$

Capital Costs

The capital costs for reverse osmosis has several components:

1. Pump capital cost
2. Membrane capital cost
3. Pressure vessel and & rack capital cost

4. Energy recovery device (optional)

The total capital costs for RO are calculated as follows, which each component described below:

$$C_{RO} = C_{pump} + C_{mem} + C_{erd} + 3.3C_{pv,r}$$

The value is then multiplied by the Indirect Cost Factor (IDF = 1.65) to arrive at the final fixed capital for reverse osmosis.

Pump Capital Costs

The pump power is calculated as:

$$p_{pump} = \frac{Q_{in} P_f}{\eta_p}$$

The pump cost is then:

$$C_{pump} = p_{pump} \frac{53}{10^5 \times 3600}^{0.97}$$

Membrane Capital Costs

The cost per meter squared of membrane area is \$30 assumed from EPA documentation below. The membrane costs are calculated as:

$$C_{mem} = A \times 30$$

Pressure Vessel & Rack Capital Costs

The pressure vessel cost is a function of area and the number of vessels per membrane area (assumed to be 0.025 vessels/m²) and the vessel cost (assumed to be \$1000) based on the default EPA assumptions:

$$C_{pv} = A * 0.025$$

The rack capital costs assumes 2 trains, 150 ft start, and 5 ft per additional vessel:

$$C_{rack} = (150 + (x_{add} A 0.025)) 33 n_{trains}$$

And the total for this component is:

$$C_{pv,r} = C_{pv} + C_{rack}$$

Energy Recovery Device

The capital costs for the ERD option is taken from Bartholomew & Mauter (2019) and is based on the volumetric mass flow into the ERD device [kg/hr]:

$$M_{tot,r} = M_{H2O,r} + M_{tds,r}$$

And the capital costs are calculated:

$$C_{erd} = 3134.8 M_{tot,r}^{0.58}$$

Electricity Intensity

The electricity intensity is a function of the pump power, ERD power, and the flow into the unit.

The pump power is calculated as [kW]:

$$p_{\text{pump}} = \frac{Q_{\text{in}} P_f}{\eta_p}$$

The ERD power is function of the retentate flow (flow out of the unit), retentate pressure, and the ERD efficiency [kW]:

$$p_{\text{erd}} = \frac{Q_{\text{out}} (P_f - 1)}{\eta_{\text{erd}}}$$

And the electricity intensity for the unit is calculated as [kWh/m³]:

$$E_{\text{RO}} = \frac{p_{\text{pump}} - p_{\text{erd}}}{Q_{\text{in}}}$$

Membrane Replacement Rate & Chemical Cost

The membrane replacement rate is included in the other variable operating costs for reverse osmosis.

Membrane replacement costs (assumed to be 25% of area per year):

$$C_{\text{replace}} = 0.25 C_{\text{mem}}$$

Chemical costs are assumed to equal 1% of the capital cost.

References

Bartholomew, T. V. and M. S. Mauter (2019).

“Computational framework for modeling membrane processes without process and solution property simplifications.” *Journal of Membrane Science* 573: 682-693.

Lu, Y.-Y., et al. (2007).

“Optimum design of reverse osmosis system under different feed concentration and product specification.” *Journal of Membrane Science* 287(2): 219-229.

US Environmental Protection Agency (2019)

“Work Breakdown Structure-Based Cost Model for Reverse Osmosis/Nanofiltration Drinking Water Treatment” <https://www.epa.gov/sites/production/files/2019-07/documents/wbs-ronf-documentation-june-2019.pdf>

Reverse Osmosis Module

```
class watertap3.wt_units.reverse_osmosis.UnitProcess(*args, **kwds)
```

```
    elect (t)
```

Parameters: **t** (*int*) – Indexing variable for Pyomo Var()

Returns:

```
    fixed_cap (t, b_cost)
```

Parameters:

• **t** (*int*) – Indexing variable for Pyomo Var()

• **b_cost** (*object*) – Costing block for unit.

Returns: Fixed capital costs for reverse osmosis [\$MM]

```
    get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

Sedimentation

Unit Parameters

There is one parameter:

- "settling_velocity" - v the assumed settling velocity for the unit [m/s]:
- Required parameter

Capital Costs

The capital costs are a function of basin surface area:

$$C_{sed} = 13572 A_{basin}^{0.3182}$$

Basin surface area is calculated as [ft²]:

$$A_{basin} = \frac{Q_{in}}{v}$$

Electricity Intensity

There are no electricity costs associated with sedimentation in WaterTAP3.

References

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities
DOI:10.1002/9780470260036

Sedimentation Module

```
class watertap3.wt_units.sedimentation.UnitProcess(*args, **kwds)
```

```
get_costing(unit_params=None, year=None)
    Initialize the unit in WaterTAP3.
```

Sodium Bisulfite Addition

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

Unit Parameters

There is one unit parameter:

- "dose" - dose of chemical [mg/L]
- Required parameter

Capital Costs

Costing parameters for sodium bisulfite addition are taken from sulfuric acid. The sodium bisulfite solution flow S [gal/day] is used in a cost curve of the general form:

$$C = aS^b$$

For a single sodium bisulfite addition unit, $a = 900.97$ and $b = 0.6179$. The full cost equation in WaterTAP3 is:

$$C_{bisulf} = 900.97S^{0.6179}$$

This cost is then multiplied by the number of units and the EIF factor for the final FCI for the chemical addition. These parameters were determined by fitting data from FIGURE 5.5.11 - SULFURIC ACID FEED in McGivney & Kawamura (2008).

Electricity Intensity

Electricity intensity for chemical additions is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/day] of the solution necessary to achieve the desired dose:

$$M_{bisulf} = Q_{in}D_{bisulf}$$

- The volumetric flow S [gal/min] of the chemical solution, which incorporates the solution density [kg/m³]:

$$S = \frac{M_{bisulf}}{\rho_{bisulf}}$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

Then the electricity intensity is calculated as [kWh/m³]:

$$E_{bisulf} = \frac{0.746Sh}{3960\eta_{pump}\eta_{motor}Q_{in}}$$

Assumptions

- Number of units = 2
- Solution density [kg/m³] = 1480
- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

Reference

William McGivney & Susumu Kawamura (2008)

Sodium Bisulfite Module

```
class watertap3.wt_units.sodium_bisulfite_addition.UnitProcess(*args, **kwargs)
```

elect ()

Electricity intensity.

Parameters:

- **lift_height** (*float*) – Lift height for pump [ft]
- **pump_eff** (*float*) – Pump efficiency
- **motor_eff** (*float*) – Motor efficiency

Returns: Electricity intensity [kWh/m3]

fixed_cap (unit_params)

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'dose': 10}

Fixed capital for sodium bisulfite addition is a function of sodium bisulfite dose, sodium bisulfite solution flow, and the number of units.

Parameters: **dose** (*float*) – Sodium bisulfite dose [mg/L]

Returns: Sodium bisulfite addition fixed capital cost [\$MM]

get_costing (unit_params=None, year=None)

Initialize the unit in WaterTAP3.

solution_vol_flow ()

Chemical solution flow in gal/day

Parameters: **solution_density** (*float*) – Solution density [kg/m3]

Returns: Sodium bisulfite solution flow [gal/day]

Storage Tanks

Unit Parameters

- "avg_storage_time" - storage time for volume calculation [hr]
 - Required parameter
- "surge_cap" - desired surge capacity for volume calculation
 - Required parameter
 - Can be zero if no surge capacity is desired

Capital Costs

Storage tank capital costs are calculated as a function of the volume of storage [m3] required, flow in [m3/hr], storage duration [hr], and surge capacity needed:

$$V = Q_{in}t(1 + x)$$

The storage volume is used to calculate capital costs (\$MM):

$$C_{tank} = 1.48 \times 10^{-4} V^{1.014}$$

The data to make the regression is from Loh et al. (2002).

Electricity Intensity

There are no electricity costs associated with storage tanks in WaterTAP3.

Reference

Loh, H. P., Lyons, J., White, C.W. (2002)
DOE/NETL-2002/1169 - Process Equipment Cost Estimation Final Report.
United States: N. p., 2002. Web. doi:10.2172/797810.
<https://www.osti.gov/servlets/purl/797810>

Storage Tank Module

```
class watertap3.wt_units.holding_tank.UnitProcess (*args, **kwds)
```

```
    elect ()
```

WaterTAP3 has no electricity intensity associated with storage tanks.

```
    fixed_cap (unit_params)
```

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'avg_storage_time': 24, 'surge_cap': 0.2}

Parameters:

- **avg_storage_time** – storage time in hours
- **surge_cap** – is the surge capacity used for calculating storage volume, expressed as a fraction of the total flow (e.g. for 20% surge capacity, use 0.2)

Returns: Fixed capital for storage tanks.

```
    get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

Sulfuric Acid Addition

Costs for chemical additions are based on the chemical dose required to treat the water and the inlet flow to the unit.

Unit Parameters

There is one unit parameter:

- "dose" - dose of chemical [mg/L]
- Required parameter

Capital Costs

The sulfuric acid solution flow S [gal/day] is used in a cost curve of the general form:

$$C = aS^b$$

For a single sulfuric acid addition unit, $a = 900.97$ and $b = 0.6179$. The full cost equation in WaterTAP3 is:

$$C_{sulf} = 900.97S^{0.6179}$$

This cost is then multiplied by the number of units and the EIF factor for the final FCI for the chemical addition. These parameters were determined by fitting data from FIGURE 5.5.11 - SULFURIC ACID FEED in McGivney & Kawamura (2008).

Electricity Intensity

Electricity intensity for chemical additions is based off the pump used to inject the chemical solution, the chemical solution flow rate, and the influent flow rate. The calculation includes:

- Lift height [ft]:

$$h$$

- The mass flow rate [kg/day] of the solution necessary to achieve the desired dose:

$$M_{sulf} = Q_{in}D_{sulf}$$

- The volumetric flow S [gal/min] of the chemical solution, which incorporates the solution density [kg/m³]:

$$S = \frac{M_{sulf}}{\rho_{sulf}}$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

Then the electricity intensity is calculated as [kWh/m³]:

$$E_{sulf} = \frac{0.746Sh}{3960\eta_{pump}\eta_{motor}Q_{in}}$$

Assumptions

- Number of units = 2
- Solution density [kg/m³] = 1781
- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

Reference

William McGivney & Susumu Kawamura (2008)
Cost Estimating Manual for Water Treatment Facilities

Sulfuric Acid Addition Module

```
class watertap3.wt_units.sulfuric_acid_addition.UnitProcess (*args, **kwds)
```

```
    elect ()
```

Electricity intensity.

Parameters:

- **lift_height** (*float*) – Lift height for pump [ft]
- **pump_eff** (*float*) – Pump efficiency
- **motor_eff** (*float*) – Motor efficiency

Returns: Electricity intensity [kWh/m3]

```
fixed_cap (unit_params)
```

“unit_params” are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'dose': 10}

Returns: Sulfuric acid addition fixed capital cost [\$MM]

```
get_costing (unit_params=None, year=None)
```

Initialize the unit in WaterTAP3.

```
solution_vol_flow ()
```

Chemical solution flow in gal/day

Parameters: **solution_density** (*float*) – Solution density [kg/m3]

Returns: Sulfuric acid solution flow [gal/day]

Surface Discharge

Unit Basics

This unit is a terminal unit in WaterTAP3 and represents the cost of discharging waste streams to surface water. This is also the assumed destination for the waste from any unit process if no waste stream is specified in the input sheet.

Unit Parameters

There are two parameters:

- "pipe_distance" - distance for piping to discharge to surface water body [mi]:
 - Optional parameter
 - If included in parameters, adds pipe construction cost to capital cost.
- "pump" - whether or not to include pumping electricity costs:
 - Optional parameter
 - Options are "yes" and "no"
 - If "yes", includes electricity calculation.

Capital Costs

The capital costs are a function of flow [m³/hr] and can include piping costs:

$$C_{surf} = 35 \frac{Q_m}{10417}^{0.873} + C_{pipe}$$

Piping cost assumes an 8 in diameter pipe, and is calculated as:

$$C_{pipe} = 0.28x_{pipe}$$

If the "pipe_distance" unit parameter is not included:

$$C_{\{pipe\}} = 0$$

Electricity Intensity

Electricity intensity for surface discharge (if included) is based off the pump used. The calculation includes:

- Lift height [ft]:

$$h$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

- And the influent flow in [gal/min] and [m³/hr]:

$$Q_{gpm}, Q_{m3hr}$$

Then the electricity intensity is calculated as:

$$E_{surf} = \frac{0.746Q_{gpm}h}{3960\eta_{pump}\eta_{motor}Q_{m3hr}}$$

Assumptions

- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

References

Unit Template

```
class watertap3.wt_units.unit_template.UnitProcess (*args, **kwds)
```

```
    elect ()  
        Docstrings go here.
```

Returns:

```
    fixed_cap ()  
        Docstrings go here.
```

Returns:

```
    get_costing (unit_params=None, year=None)  
        Initialize the unit in WaterTAP3.
```

Seawater Intake

This is the intake unit process for seawater cases.

Unit Parameters

None.

Capital Costs

The capital costs are a function of flow [MGD] are from Voutchkov (2018):

$$C_{sw} = 2.15 \times 10^{-4} Q_{in}^{0.888803}$$

Electricity Intensity

Electricity intensity for seawater intake is based off the pump used. The calculation includes:

- Lift height [ft]:

$$h$$

- The pump and motor efficiencies:

$$\eta_{pump}, \eta_{motor}$$

- And the influent flow in [gal/min] and [m3/hr]:

$$Q_{gpm}, Q_{m3hr}$$

Then the electricity intensity is calculated as:

$$E_{surf} = \frac{0.746 Q_{gpm} h}{3960 \eta_{pump} \eta_{motor} Q_{m3hr}}$$

Assumptions

- Lift height [ft] = 100
- Pump efficiency = 0.9
- Motor efficiency = 0.9

References

Voutchkov, N. (2018).
Desalination Project Cost Estimating and Management.
<https://doi.org/10.1201/9781351242738>

Seawater Intake Module

```
class watertap3.wt_units.sw_onshore_intake.UnitProcess(*args, **kwargs)
```

```
get_costing(unit_params=None, year=None)
    Initialize the unit in WaterTAP3.
```

Tri-Media Filtration

Unit Parameters

None

Capital Costs

The capital costs are a function of flow [MGD] from the Texas Water Development Board (2016):

$$C_{tri} = 0.72557Q_{in}^{0.5862}$$

Electricity Intensity

Electricity intensity is fixed at 0.00045 kWh/m3 from Bukhary et al. (2019).

References

CAPITAL

[User's Manual for Integrated Treatment Train Toolbox - Potable Reuse \(IT3PR\) Version 2.0](#)

Steinle-Darling, E., Salveson, A., Russel, C., He, Q., Chiu, C., Lesan, D.

Texas Water Development Board

December 2016

ELECTRICITY

Bukhary, S., et al. (2019).

"An Analysis of Energy Consumption and the Use of Renewables for a Small Drinking Water Treatment Plant." *Water* 12(1).

Tri-Media Filtration Module

```
class watertap3.wt_units.tri_media_filtration.UnitProcess (*args, **kwds)
```

```
    get_costing (unit_params=None, year=None)
        Initialize the unit in WaterTAP3.
```

UV + AOP

The UV+AOP module is used to determine costs for both UV disinfection and UV + AOP units.

Capital costs for UV systems in WaterTAP3 are significantly influenced by UV Dose, UVT, and flow. For most accurate capital cost, include facility data with input parameters.

Unit Parameters

This module has one required parameter and three optional parameters for the input sheet:

- "aop" - boolean indicating if unit includes AOP or not:
 - Required parameter
 - If `True`, include chemical costs
- "uvt_in" - assumed ultraviolet transmittance (UVT) into unit:
 - Optional parameter
 - Default value is 0.9
 - Must be between 0.5 and 0.99 in increments of 0.01
- "uv_dose" - Reduction Equivalent Dose (RED) for unit:
 - Optional parameter
 - Default value is 100 mJ/cm²
 - Must be <1200 in increments of 10
- "chemical_name" - name of chemical used for oxidant
 - Optional parameter
 - No default value
 - Can be any chemical but must match *exactly* chemical entry in `catalysts_chemicals.csv`, e.g. "Hydrogen_Peroxide"

Capital Costs

Capital costs for UV + AOP includes cost for the UV system and for the oxidant injection system:

$$C_{UV + AOP} = C_{UV} + C_{ox}$$

UV Capital Costs

Capital costs are a function of flow, UV dose, and UV Transmission (UVT). The WaterTAP3 model uses data Table 3.22 in Texas Water Development Board (2016) to interpolate cost values for:

- UVT between 0.5 and 0.99 in increments of 0.01.
- UV Dose between 10 and 1,200 mJ/cm² in increments of 10.

For the given UVT and UV Dose, WaterTAP3 retrieves cost data for flows between 1 and 25 MGD from `data/uv_cost_interp.csv`. Using this data, a cost curve is fit to a power curve. Once *a* and *b* are known, capital costs for the UV system are determined:

$$C_{UV} = aQ_{in}^b$$

Oxidant Capital Costs

The unit can accept any chemical name and dose, but the cost curve used was developed specifically for Hydrogen Peroxide.

Oxidant costs are calculated with:

$$C_{ox} = 1228(Q_{in} * D_{ox})^{0.2277}$$

Where the oxidant dose [kg/m3] is taken from the input parameter "dose".

If "aop" is False in the unit parameters:

$$C_{ox} = 0$$

Electricity Intensity

Electricity intensity for chlorination is fixed at 0.1 kWh/m3 and is taken from USEPA (2010).

References

CAPITAL

User's Manual for Integrated Treatment Train Toolbox - Potable Reuse (IT3PR) Version 2.0

Steinle-Darling, E., Salveson, A., Russel, C., He, Q., Chiu, C., Lesan, D.

Texas Water Development Board

December 2016

ELECTRICITY

US Environmental Protection Agency (2010)

"Evaluation of Energy Conservation Measures for Wastewater Treatment Facilities"

<https://nepis.epa.gov/Exe/ZyPURL.cgi?Dockkey=P1008SBM.TXT>

UV + AOP Module

```
class watertap3.wt_units.uv_aop.UnitProcess(*args, **kwds)
```

```
fixed_cap(unit_params)
```

"unit_params" are the unit parameters passed to the model from the input sheet as a Python dictionary.

EXAMPLE: {'aop': True, 'uv_dose': 350, 'dose': 5, 'chemical_name': 'Hydrogen Peroxide'}

Parameters:

- **aop** (*bool*) – (**required**) Boolean that determines if UV is used with AOP. Must be either True or False
- **uvt_in** (*float*) – (**optional**, default is 0.9) UV transmission (UVT) into unit
- **uv_dose** (*float*) – (**optional**, default is 100) Reduction Equivalent Dose (RED) [mJ/cm2]
- **dose** (*float*) – (**optional**, no default) Dose for oxidant (if AOP) [mg/L]
- **chemical_name** (*str*) – (**optional**, default is 'Hydrogen Peroxide') Name of oxidant used for AOP.

Returns: Fixed capital for UV or UV+AOP unit [\$MM]

get_costing (unit_params=None, year=None)

Initialize the unit in WaterTAP3.

solution_vol_flow ()

Determine oxidant solution flow rate in gal / day

Returns: Oxidant solution flow [gal/day]

uv_regress ()

Determine a, b costing parameters as a function of flow, UVT, and UV dose for unit.

Parameters:

- **flow_in** (*float*) – Volumetric flow into unit [MGD]
- **uvt_in** (*float*) – UV transmission (UVT) into unit
- **uv_dose** (*float*) – UV dose used by the unit [mg/L]

Returns: a, b

Water Pumping Station

Unit Basics

This is a pump unit in WaterTAP3.

Unit Parameters

There are two unit parameters:

- "pump_type" - the type of water pumping station:
 - Required parameter
 - Options are "raw" or "treated"
 - Different costing values are used for each option
- "pump_power" - pump power if available [hp]:
 - Optional parameter

Capital Costs

Depending on the value for "pump_type", different cost curves are used for the general form with flow in [MGD] from McGivney & Kawamura (2008):

$$C_{wps} = aQ_{in}^b$$

For "pump_type" = "raw":

- $a = 19370.36$
- $b = 0.9149$

For "pump_type" = "treated":

- $a = 40073.43$
- $b = 0.8667$

Electricity Intensity

If there is no input for "pump_power", electricity intensity is a function of flow [gpm]:

$$E_{wps} = 0.1024 \frac{Q_{in}}{440.3}$$

Otherwise, the input for "pump_power" is converted from horsepower to kW and electricity intensity is:

$$E_{wps} = \frac{p_{pump}}{Q_{in}}$$

References

William McGivney & Susumu Kawamura (2008)

Water Pumping Station Module

```
class watertap3.wt_units.water_pumping_station.UnitProcess (*args, **kwds)
```

```
    get_costing (unit_params=None, year=None)  
        Initialize the unit in WaterTAP3.
```

Well Field**Unit Parameters**

None.

Capital Costs

The capital costs are a function of flow [m3/hr] from Voutchkov (2018):

$$C_{tri} = 4731.6Q_{in}^{0.9196}$$

Electricity Intensity

There are no electricity costs associated with well field in WaterTAP3.

References

Voutchkov, N. (2018).
Desalination Project Cost Estimating and Management.
<https://doi.org/10.1201/9781351242738>

Well Field Module

```
class watertap3.wt_units.well_field.UnitProcess (*args, **kwds)
```

```
    get_costing (unit_params=None, year=None)  
        Initialize the unit in WaterTAP3.
```

Index

E

[elect\(\) \(watertap3.wt_units.alum_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.ammonia_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.anti_scalant_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.basic_unit.UnitProcess method\)](#)
[\(watertap3.wt_units.brine_concentrator.UnitProcess method\)](#)
[\(watertap3.wt_units.caustic_soda_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.chemical_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.chlorination.UnitProcess method\)](#)
[\(watertap3.wt_units.co2_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.coagulant_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.crystallizer.UnitProcess method\)](#)
[\(watertap3.wt_units.deep_well_injection.UnitProcess method\)](#)
[\(watertap3.wt_units.evaporation_pond.UnitProcess method\)](#)
[\(watertap3.wt_units.ferric_chloride_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.fixed_bed_gravity_basin.UnitProcess method\) \[1\]](#)
[\(watertap3.wt_units.gac_pressure_vessel.UnitProcess method\) \[1\]](#)
[\(watertap3.wt_units.holding_tank.UnitProcess method\)](#)
[\(watertap3.wt_units.hydrochloric_acid_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.lime_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.lime_softening.UnitProcess method\)](#)
[\(watertap3.wt_units.ozone_aop.UnitProcess method\)](#)
[\(watertap3.wt_units.reverse_osmosis.UnitProcess method\)](#)
[\(watertap3.wt_units.sodium_bisulfite_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.sulfuric_acid_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.unit_template.UnitProcess method\)](#)
[evaporation_rate\(\) \(watertap3.wt_units.evaporation_pond.UnitProcess method\)](#)
[evaporation_rate_regress\(\) \(watertap3.wt_units.evaporation_pond.UnitProcess method\)](#)

F

[fixed_cap\(\) \(watertap3.wt_units.alum_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.ammonia_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.anti_scalant_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.basic_unit.UnitProcess method\)](#)
[\(watertap3.wt_units.brine_concentrator.UnitProcess method\)](#)
[\(watertap3.wt_units.caustic_soda_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.chemical_addition.UnitProcess method\)](#)
[\(watertap3.wt_units.chlorination.UnitProcess method\)](#)

(watertap3.wt_units.co2_addition.UnitProcess method)
 (watertap3.wt_units.coagulant_addition.UnitProcess method)
 (watertap3.wt_units.crystallizer.UnitProcess method)
 (watertap3.wt_units.deep_well_injection.UnitProcess method)
 (watertap3.wt_units.evaporation_pond.UnitProcess method)
 (watertap3.wt_units.ferric_chloride_addition.UnitProcess method)
 (watertap3.wt_units.fixed_bed_gravity_basin.UnitProcess method) [1]
 (watertap3.wt_units.gac_pressure_vessel.UnitProcess method) [1]
 (watertap3.wt_units.holding_tank.UnitProcess method)
 (watertap3.wt_units.hydrochloric_acid_addition.UnitProcess method)
 (watertap3.wt_units.lime_addition.UnitProcess method)
 (watertap3.wt_units.lime_softening.UnitProcess method)
 (watertap3.wt_units.ozone_aop.UnitProcess method)
 (watertap3.wt_units.reverse_osmosis.UnitProcess method)
 (watertap3.wt_units.sodium_bisulfite_addition.UnitProcess method)
 (watertap3.wt_units.sulfuric_acid_addition.UnitProcess method)
 (watertap3.wt_units.unit_template.UnitProcess method)
 (watertap3.wt_units.uv_aop.UnitProcess method)

G

get_costing() (watertap3.wt_units.alum_addition.UnitProcess method)
 (watertap3.wt_units.ammonia_addition.UnitProcess method)
 (watertap3.wt_units.anti_scalant_addition.UnitProcess method)
 (watertap3.wt_units.backwash_solids_handling.UnitProcess method)
 (watertap3.wt_units.brine_concentrator.UnitProcess method)
 (watertap3.wt_units.cartridge_filtration.UnitProcess method)
 (watertap3.wt_units.caustic_soda_addition.UnitProcess method)
 (watertap3.wt_units.chemical_addition.UnitProcess method)
 (watertap3.wt_units.chlorination.UnitProcess method)
 (watertap3.wt_units.co2_addition.UnitProcess method)
 (watertap3.wt_units.coagulant_addition.UnitProcess method)
 (watertap3.wt_units.crystallizer.UnitProcess method)
 (watertap3.wt_units.electrodialysis_reversal.UnitProcess method)
 (watertap3.wt_units.evaporation_pond.UnitProcess method)
 (watertap3.wt_units.ferric_chloride_addition.UnitProcess method)
 (watertap3.wt_units.fixed_bed_gravity_basin.UnitProcess method) [1]
 (watertap3.wt_units.gac_pressure_vessel.UnitProcess method) [1]
 (watertap3.wt_units.holding_tank.UnitProcess method)
 (watertap3.wt_units.hydrochloric_acid_addition.UnitProcess method)
 (watertap3.wt_units.iron_and_manganese_removal.UnitProcess method)
 (watertap3.wt_units.landfill.UnitProcess method)

(watertap3.wt_units.landfill_zld.UnitProcess method)
 (watertap3.wt_units.lime_addition.UnitProcess method)
 (watertap3.wt_units.lime_softening.UnitProcess method)
 (watertap3.wt_units.media_filtration.UnitProcess method)
 (watertap3.wt_units.microfiltration.UnitProcess method)
 (watertap3.wt_units.municipal_drinking.UnitProcess method)
 (watertap3.wt_units.ozone_aop.UnitProcess method)
 (watertap3.wt_units.reverse_osmosis.UnitProcess method)
 (watertap3.wt_units.sedimentation.UnitProcess method)
 (watertap3.wt_units.sodium_bisulfite_addition.UnitProcess method)
 (watertap3.wt_units.sulfuric_acid_addition.UnitProcess method)
 (watertap3.wt_units.sw_onshore_intake.UnitProcess method)
 (watertap3.wt_units.tri_media_filtration.UnitProcess method)
 (watertap3.wt_units.unit_template.UnitProcess method)
 (watertap3.wt_units.uv_aop.UnitProcess method)
 (watertap3.wt_units.water_pumping_station.UnitProcess method)
 (watertap3.wt_units.well_field.UnitProcess method)

S

solution_vol_flow() (watertap3.wt_units.alum_addition.UnitProcess method)
 (watertap3.wt_units.ammonia_addition.UnitProcess method)
 (watertap3.wt_units.anti_scalant_addition.UnitProcess method)
 (watertap3.wt_units.caustic_soda_addition.UnitProcess method)
 (watertap3.wt_units.chemical_addition.UnitProcess method)
 (watertap3.wt_units.coagulant_addition.UnitProcess method)
 (watertap3.wt_units.ferric_chloride_addition.UnitProcess method)
 (watertap3.wt_units.hydrochloric_acid_addition.UnitProcess method)
 (watertap3.wt_units.lime_addition.UnitProcess method)
 (watertap3.wt_units.lime_softening.UnitProcess method)
 (watertap3.wt_units.ozone_aop.UnitProcess method)
 (watertap3.wt_units.sodium_bisulfite_addition.UnitProcess method)
 (watertap3.wt_units.sulfuric_acid_addition.UnitProcess method)
 (watertap3.wt_units.uv_aop.UnitProcess method)

U

UnitProcess (class in watertap3.wt_units.alum_addition)
 (class in watertap3.wt_units.ammonia_addition)
 (class in watertap3.wt_units.anti_scalant_addition)
 (class in watertap3.wt_units.backwash_solids_handling)
 (class in watertap3.wt_units.basic_unit)
 (class in watertap3.wt_units.brine_concentrator)

(class in watertap3.wt_units.cartridge_filtration)
(class in watertap3.wt_units.caustic_soda_addition)
(class in watertap3.wt_units.chemical_addition)
(class in watertap3.wt_units.chlorination)
(class in watertap3.wt_units.co2_addition)
(class in watertap3.wt_units.coagulant_addition)
(class in watertap3.wt_units.crystallizer)
(class in watertap3.wt_units.deep_well_injection)
(class in watertap3.wt_units.electrodialysis_reversal)
(class in watertap3.wt_units.evaporation_pond)
(class in watertap3.wt_units.ferric_chloride_addition)
(class in watertap3.wt_units.fixed_bed_gravity_basin) [1]
(class in watertap3.wt_units.gac_pressure_vessel) [1]
(class in watertap3.wt_units.holding_tank)
(class in watertap3.wt_units.hydrochloric_acid_addition)
(class in watertap3.wt_units.iron_and_manganese_removal)
(class in watertap3.wt_units.landfill)
(class in watertap3.wt_units.landfill_zld)
(class in watertap3.wt_units.lime_addition)
(class in watertap3.wt_units.lime_softening)
(class in watertap3.wt_units.media_filtration)
(class in watertap3.wt_units.microfiltration)
(class in watertap3.wt_units.municipal_drinking)
(class in watertap3.wt_units.ozone_aop)
(class in watertap3.wt_units.reverse_osmosis)
(class in watertap3.wt_units.sedimentation)
(class in watertap3.wt_units.sodium_bisulfite_addition)
(class in watertap3.wt_units.sulfuric_acid_addition)
(class in watertap3.wt_units.sw_onshore_intake)
(class in watertap3.wt_units.tri_media_filtration)
(class in watertap3.wt_units.unit_template)
(class in watertap3.wt_units.uv_aop)
(class in watertap3.wt_units.water_pumping_station)
(class in watertap3.wt_units.well_field)
uv_regress() (watertap3.wt_units.uv_aop.UnitProcess method)