Supporting material for CityWat: sub-models

Barnaby Dobson and Ana Mijic, Imperial College London, UK

# Summary

In this document we describe the mass balance and operation equations in CityWat for our model of London’s water cycle. Each section describes a different sub-model that can be called. We first describe the functions for the standard simulation model and then describe how different options alter these functions or add new functions. To improve readability, we have omitted unit conversions in our equations, however variable/parameter units and conversions are all explicit in the code itself.

# Urban water cycle sub-models

## Abstraction (scripts/models.py/abstraction)

This function evaluates water use restrictions and minimum required flows; and determines how much water to abstract from the River Thames, either to supply reservoir storage or direct to treatment.

### Evaluate water use restrictions and minimum required flow

Our implementation of water use restrictions is based on that presented in (Mortazavi‐Naeini et al., 2019).

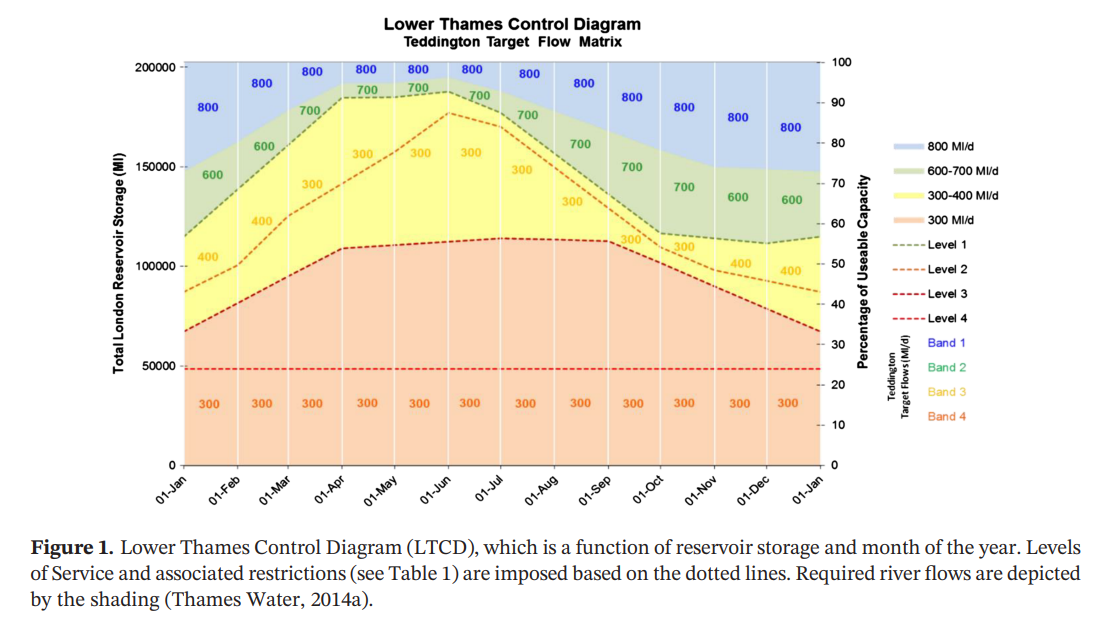


Figure 1: The Lower Thames Operating Agreement, from Figure 1 in (Mortazavi‐Naeini et al., 2019).

This diagram states the minimum required flow and level of water use restrictions depending on the time of year and total supply reservoir storage. Thus, level of restrictions can be described as follows,

Where LORt is the level of restrictions at time t, dependent on supply reservoir storage at the start of the timestep, SS,t-1, the month of the year, month­t, transformed by a function f which is depicted by the dashed lines in Figure 1. Similarly, the minimum required flow in the Thames can be described by,

Where MRFt, is the minimum required flow at time t, transformed by a function g which is depicted by the filled areas in Figure 1.

### Calculate abstraction

Once MRFt has been calculated, the amount of water available for abstraction is known, described by,

Where amax,t is the maximum abstraction at time t, Qa,t is the flow upstream of abstractions, aupstream are the abstractions that take place upstream of London’s abstraction and acap is capacity on the abstractions.

The abstraction is then made to be the maximum beneficial abstraction, i.e. the abstraction that does not draw river flow below MRFt, does not cause reservoirs to spill, nor results in an oversupply to the water treatment works. This can be described as,

Where atarg,t is the maximum allowable and beneficial river abstraction on a given timestep, t, Ss,cap is the storage capacity of supply reservoirs, DFWTW,t-1 is the freshwater treatment works demand on the previous timestep, and agw,targ is the target groundwater abstraction on a given day.

Once the target abstraction has been calculated it is sent to satisfy freshwater treatment plant demand and fill up reservoirs, as follows,

Where aFWTW,t is the river abstraction direct to freshwater treatment, aS,t is the river abstraction to reservoirs.

### Update storage and flow state variables

Finally, the supply reservoir storage’s volume can be updated as follows,

And the river flow downstream of abstraction can also be updated,

## Releases (scripts/models.py/release)

This function determines how much water should be sent to freshwater treatment works and what combination of groundwater, river water and reservoir water should be used.

We calculate the ideal reservoir release, assuming groundwater is set at its target abstraction,

We then evaluate where storage is in relation to the Thames storage control curves (the Level 1 restriction curve depicted in Figure 1) if the target release was to be made,

Where SCCt is the volume of storage above the control curve on a given day, h is a function that transforms that is the storage value at the control curve on a given month.

To avoid drawing down the reservoir below the control curve, groundwater abstractions may be increased up to their maximum,

Where agw,cap is the maximum possible daily groundwater abstraction. Once groundwater abstractions have been calculated, reservoir releases must make up the remaining freshwater treatment works demand,

And reservoir storage updated,

## Calculate consumer demand (scripts/models.py/calculate\_consumer\_demand)

Before updating freshwater treatment demand, we must know how much water is required by consumers. This function applies water use restrictions, seasonal demand profiles and calculates how much of outdoor water demand is satisfied by rainfall.

### Baseline demand

The model first calculates a ‘baseline demand’, that can be thought of as the average annual demand,

### Water use restrictions and seasonal profile

Then, restrictions are implemented. Each level of restrictions reduces demand by a different percentage, as described in Figure 2.

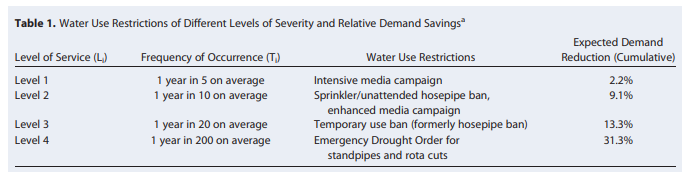


Figure 2: Demand reductions that result from water use restrictions, Table 1 in (Borgomeo et al., 2016)

Thus, the percentage reduction can be described by,

Where k transforms the level of restrictions according to Figure 2. Thus, by multiplying baseline demand with a seasonal demand profile and percentage reductions, the total demand on a given day can be defined by,

Where *l* defines a seasonal profile of water demand.

### Outdoor water demand

We have assumed that outdoor water demand is simply a percentage of Dh,tot,t. Thus, the amount of demand that can be supplied by rainfall is described by,

Where Pt is the precipitation on a given day, Agardens is the area of gardens in the model and Rrain is the percentage of demand that is outdoors (and thus can be satisfied by rainfall).

Thus, the demand from water consumers can be described by,

## Calculate distribution demand (scripts/models.py/calculate\_distribution\_demand)

Once consumer demand has been calculated, distribution demand can simply be represented by including leakage on top of consumer demand as follows,

## Freshwater treatment (scripts/models.py/freshwater\_treatment)

With distribution demand and the freshwater treatment input known, the freshwater treatment output can be calculated and its demand updated.

It is simplest to start by calculating the processing losses as the percentage of freshwater treatment input (i.e. the sum of river and groundwater abstractions, and reservoir releases),

Thus, the treatment output to service reservoirs and direct to distribution can be calculated,

We first send the treatment output to fill reservoirs and then send the remainder to distribution (since the distribution function can draw from service reservoirs anyway),

## Distribution (scripts/models.py/distribution)

## Household output (scripts/models.py/calculate\_household\_output)

## Urban runoff (scripts/models.py/urban\_runoff)

## Sewerage (scripts/models.py/sewerage)

## Combined sewer overflow (scripts/models.py/cso)

## Wastewater treatment (scripts/models.py/wastewater\_treatment)

## Water quality (scripts/models.py/water\_quality)

# Options

## Rainwater harvesting (scripts/models.py/calculate\_consumer\_demand and urban\_runoff)

## Wastewater reuse (scripts/models.py/wastewater\_reuse)

## Abstraction (scripts/models.py/abstraction)

# Summary of variables

## State variables

All state variables are defined on a given day, t.

|  |  |
| --- | --- |
| Symbol | Description |
| SS,t | Storage in supply reservoirs |
| montht | Month |
| LORt | Level of restrictions |
| MRFt | Minimum required flow |
| atarg,t | Maximum allowable and beneficial river abstraction |
| DFWTW,t | Demand at freshwater treatment works |
| aFWTW,t | River abstraction direct to freshwater treatment |
| aS,t | River abstraction direct to supply reservoirs |
| Qd,t | Flow downstream of abstractions |
| uFWTW,targ,t | Target release from supply reservoirs to freshwater treatment |
| agw,t | Groundwater abstractions direct to freshwater treatment |
| uFWTW,t | Release from supply reservoirs to freshwater treatment |
| Dh,base,t | Baseline demand |
| Rreduction,t | Percentage to reduce demand by due to water use restrictions |
| Dh,tot,t | Total water demand on a given day |
| Sh,rain,t | Amount of demand supplied by precipitation |
| Dh,eff,t | Effective demand required by consumers from distribution network |
| Ddist,t | Treated water required by distribution network from service reservoirs |
| LFWTW,t | Processing losses during freshwater treatment |
| OFWTW,t | Output of freshwater treatment works |
| SDres,t | Storage in service reservoirs |
| uDres,t | Release from freshwater treatment works to service reservoirs |
|  |  |
|  |  |

## Parameters

|  |  |
| --- | --- |
| Symbol | Description |
| acap | Max river abstraction on a given day |
| aupstream | Constant daily abstraction upstream of London abstractions |
| Ss,cap | Storage capacity in supply reservoirs |
| agw,targ | The target daily abstraction from groundwater sources |
| Nhouseholds | Number of households covered by the model |
| Dhousehold | Per household consumption per day |
| Dnon\_household | Total water demand not in households |
| Agarden | Area of gardens in London |
| Atotal | Area of London |
| Rrain | Proportion of demand satisfiable by rainfall |
| Rleak,dist | Proportion of distribution throughput that becomes leaked water |
| Rleak,FWTW | Proportion of freshwater treatment input that becomes leaked water |
| SDres,cap | Storage capacity in service reservoirs |
|  |  |

## Input data

|  |  |
| --- | --- |
| Symbol | Description |
| Qa,t | Average flow over a given day upstream of abstractions |
| Pt | Precipitation amount over London |
|  |  |

# References

Borgomeo, E., Mortazavi-Naeini, M., Hall, J. W., O’Sullivan, M. J., & Watson, T. (2016). Trading-off tolerable risk with climate change adaptation costs in water supply systems. *Water Resources Research*, *52*(2), 622–643. https://doi.org/10.1002/2015WR018164

Mortazavi‐Naeini, M., Bussi, G., Elliott, J. A., Hall, J. W., & Whitehead, P. G. (2019). Assessment of risks to public water supply from low flows and harmful water quality in a changing climate. *Water Resources Research*, 2018WR022865. https://doi.org/10.1029/2018WR022865