**Tutorial 1**

In this tutorial we did:

1. Build a toy model of a real reservoir
2. Apply a water balance assuming simple operating rules
3. Visualise results

Important: all the data presented here is NOT supposed to be 100% accurate. This is a *model inspired from* the real-life case of the Conowingo Dam, rather than a *model of* the Conowingo Dam. For instance, note that we neglect evaporation and seepage losses in the whole tutorial; we also don’t include the Muddy Run pumped storage facility in the model.

*Solutions in italics.*

**Part 0. Preparation**

Conowingo Dam is introduced by the tutorial video introduction (see Blackboard). The video presents the dam and its uses. The tutorial folder comprises data (under the form of Google spreadsheet) and uncompleted Matlab (.m) files.

First let’s create a dedicated folder for tutorials for this module, and a subfolder for Tutorial 1.

Then, let’s download the data (as an Excel file, .xlsx) and the Matlab files, and save them in the same dedicated folder. It is important to set up a dedicated workspace for any modelling work.

Now let’s look at a few preliminary questions to understand better how to manipulate the data.

Q1. Open the “Reservoir characteristics” sheet of the Excel file. Also open “reservoir\_shape.m” in Matlab. You’ll see there are two lines to complete (lines 40 and 43) using the annotations of line 8-9 of the Excel sheet. Complete them to get the plots relating reservoir head, reservoir lake surface area, and reservoir volume.

*See Matlab file. The assumption rests on the relationship between head and surface area, and this constrains “Volume = f(Head)” which is important to relate water fluxes (and stocks) to reservoir levels. Surface area would have been important if we had factored in evaporation from the lake surface.*

Q2. Maximal release through the turbines is indicated in B5 of the “Reservoir characteristics” sheet. What does that imply for the combined turbine and generator efficiency?

*To generate power at 100% of installed capacity, we need maximal release through the turbines and a full reservoir (nominal head). Then we have where is the installed capacity (in W) and directly converts from potential energy into electric energy: it accounts for turbine and generator efficiency. This gives = 0.853 with g=9.81m/s2 and =1000 kg/m3 (conversion from MW to W needed).*

*This value is close to what we would have obtained assuming respective turbine and generator efficiencies of 0.90 and 0.95.*

Q3. Now let’s learn to manipulate Excel files in Matlab. Look up the “Data retrieval” part of “reservoir\_shape.m” (lines 12-26), and follow the questions in order:

1. Can you adapt this command to get the demand data in the command line interface?

*demands = xlsread(‘Conowingo data.xlsx’, 'Demands', 'B2:E13');*

1. How do you get the minimum release for environmental flows in April without ever reporting the demand value you read in the Excel sheet manually? (and in m3/s instead of cfs?)

*b = demands(4,4) \* 0.3048^3;*

1. Reading both the “Demands” and “Flow data” Excel sheets and storing them in separate variables, can you use the month value in the “Flow data” sheet to retrieve the environmental flow requirement on day 4236 (still without manual reporting from the Excel sheet).

*flows = xlsread(‘Conowingo data.xlsx”, 'Flow data’);*

*c =demands(flows(4236,5), 4) \* 0.3048^3;*

You can look up the Matlab help: can you locate function xlsread in there?

* The “Help” button is at the right of the “HOME” tab. Click it and then type “xlsread” in the search bar at the top of the pop-up window.

**Part 1. Basic water balance**

We will now carry out the water balance for Conowingo Dam, using our data and a simple release rule called SOP (Standard Operating Policy). In the SOP, we try to match the release requirements to meet environmental flows as much as we can; if the reservoir is full, we release more to avoid overfilling the reservoir.

We do that from the main Matlab routine “main.m” (Section “Part 1”) and the water balance function it calls.

Q1. A Matlab structure storing reservoir characteristics is set up. But the units need to be changed so everything uses m, m3 and s as units (note: m3 is the volume during a time step, here m3 and m3/day are used interchangeably).

*See main.m.*

Q2. First we need to prepare the flows and demand data. Use what you learned from Q3 of Part 0 to fill in that part as required (main.m).

Note that demands are divided between demand from downstream (environmental flows) and demands that are met by direct withdrawals from the reservoir (other demands).

*See solution version main\_solutions.m, lines 30 to 41.*

Q3. Now we can complete the water balance function in a separate file named “water\_balance\_basic.m”. Follow instructions so that you get plots of storage and outflows that match the result figures for Part 1.

*See water\_balance\_basic.m*

**Part 2. Water balance refinements**

Copy function “water\_balance\_basic.m” into a new file (and function) “water\_balance\_inter.m” and take the following steps.

An important approximation when building the model above is that in reality, intakes for domestic and industrial water uses are at a certain height, meaning that water can only be withdrawn if lake levels are high enough.

Q1. First, assume that all three sources have an intake at the same level of 100 ft. How can you modify the water balance to only withdraw water if the level is high enough?

(Hint: the “main.m” file

*Implemented in file “water\_balance\_inter.m”. Structure “reservoir” that is an argument to the function is given new fields to 1) compute head, and 2) have the intakes integrated.*

*Head is computed based on availability before at-site withdrawals, and withdrawals only happen if levels are high enough. To compute head, it is necessary to go back to the equations linking head, volume and surface area to eliminate the latter. Indeed, we have head H, storage S, and area A, and:*

*1) we have a linear relationship between A and H in G5 of spreadsheet "reservoir characteristics", with A = A\_full \* (H-H\_empty)/(H\_full-H\_empty)*

*2) from H5 same spreadsheet, since the volume is half the volume of a rectangular reservoir: S= 1/2\* (H-H\_empty)\*A or A=2\*S/(H-H\_empty)*

*﻿So eliminating A from (1) and (2) we have: H-H\_empty = sqrt(2\*S\*(H\_full-H\_empty)/A\_full*

*which directly leads to the formula in the solution for the water balance.*

Q2. Now, let us consider the actual intake heights:

* 103.5 ft for the Peach Bottom nuclear plant
* 100.5 ft for the Chester water supply
* 91.5 ft for the Baltimore water supply

Can you integrate those three conditions to the water balance? Make a copy of “water\_balance\_inter/.m” called “water\_balance\_final.m”

(hint: for argument “local\_demand” use an array with three columns instead of one)

*A loop in the final water balance, on the number of intakes, is what is needed here (see water\_balance\_final). It relies on defining “reservoir.demand\_intake\_level” with the intakes in the same order as the columns in the “Demands” spreadsheet.*

**Part 3. Visualise results**

Q1. By completing unfinished lines, compute hydropower production in MWh 1. as a time series to plot (hint: copy and paste from figure templates from part 1), and 2. as an annual average.

*See main.m*

*To get the average hydropower production in W, we need to convert release back to m3/s, hence the 86400 factor in main.m (number of seconds in a day). Then dividing by 1E6 to convert to MW, and multiply par 24 to convert into hydropower production over all 24h in the day.*

Q2. Uncomment the rest of the code and look at the Figures. What do they mean in terms of uses?

*In-class. For the last figure, note that all demands have failure events where water levels are below the intake… see Tutorial 2 to compute this.*