Solution Exercise week 1 – Hydrologic Balance

Exercise 1, on paper

Task: make a sketch of this hydrologic system, highlighting the appropriate control volume and the relevant fluxes. Write down a mass-balance equation for this system.

Solution: the only input to the system is given by infiltration, which in turn is assumed to be equal to precipitation (no surface runoff is produced). The outputs are the evapotranspiration flux, which returns water to the atmosphere, and the bottom leakage, which drains water to deeper aquifers. All fluxes are expressed as area-specific, hence their units are lenght over time $[L/T]$, while storage is in length units $[L]$. The balance equation for this system reads:

$$\frac{dS(t)}{dt} = IN(t) - OUT(t) = P(t) - ET(t) - L(t)$$

With initial condition $S(t=0) = S_0$. Since the storage can be expressed as the product between the available pore volume (nZ, [L]) and the soil saturation (s, [-]), the balance can be also expressed as:

$$nZ\frac{ds(t)}{dt} = P(t) - ET(t) - L(t)$$

with some initial condition $s(t=0) = s_0$

Task: discretize the mass-balance equation. Then, compute the evolution of the soil water storage $S(t)$ over time, knowing that the initial soil moisture is $s_0 = 0.4$, soil depth is $Z = 1000\,mm$, porosity is $n = 0.35$ and using the mean hydrologic fluxes reported in Table 1.

Solution: as the considered fluxes represent an average value during the timestep, the soil storage evolution can be computed by implementing the discretized problem:
`S(i+1)=S(i)+(P(i)-ET(i)-L(i))*dt`, for `i=1:5` starting with `S(1)=s_0*Z*n=140` mm. Note that the computed S values refer to the beginning of the timestep. The solution is reported in Table 1.

timestep	-	1	2	3	4	5	6
time	[d]	0	0.5	1.0	1.5	2.0	2.5
$S(t)$	[mm]	140	139.4	138.5	137.8	136.9	136.0

Table 1: Result for the water storage $S(t)$, rounded to the first decimal digit

Exercise 2, on a computer

1. **Question:** over how much time were those variables recorded?
   Solution: The file includes 8760 entries at hourly regular timesteps $\Delta t = 1/24$ days, hence it has $8760/24 = 365$ days, i.e. 1 year of data.

2. **Task:** make plots of the imported fluxes as a function of time. Report the fluxes in mm/d and time as the number of days elapsed since the first measurement.
Solution: the simplest way to solve this exercise is to create a new x-axis which expresses time in days, for example `t=(1:N)*dt`. See Figure (1) for a plot example. It is VERY important to report correct labels for both the x-axis and the y-axis.

3. **Task:** compute the mean and the standard deviation of each variable.
Solution:
P: mean=3.18 mm/d, std=13.28 mm/d
ET: mean=1.50 mm/d, std= 1.98 mm/d
L: mean=1.76 mm/d, std= 2.95 mm/d

   Questions: what is the difference between using `std(`*variable*`,0)` and `std(`*variable*`,1)`? (If you are using Python (NumPy/pandas): what settings correspond to the sample vs. population standard deviation, and which are the defaults?) How many rainy hours are there in the dataset? What is the average precipitation on a rainy hour?
Solution: for a complete answer, check the Matlab documentation. In `std(`*x*`,0)` (the default), one computes the 'sample standard deviation', so the normalization factor is N-1. In `std(`*x*`,1)` one computes the 'population standard deviation', so the normalization factor is N. The difference becomes negligible for large samples ($1/N \approx 1/(N-1)$), but it can be important for small samples, like the ones of Assignment 1!
(For Python, in NumPy, np.std with `ddof=0` (default) computes the population standard deviation (normalization N); `ddof=1` gives the sample standard deviation (normalization N-1). In pandas, `Series/DataFrame.std` defaults to `ddof=1` (sample); set `ddof=0` for the population version. As above, the difference shrinks for large N but can be important for small samples.)
A simple way to count the number or rainy days is to use *logical expressions*. The command `P>0` returns a vector of the same size as `P`, where for each entry of `P` there is a `true` (or 1) if the logical condition is true (i.e. if `P>0`), while there is `false` (or 0) if the logical condition is not true (i.e. if `P<=0`). When logical indexing is used, Matlab operations are only performed over the `true` (or 1) values. Hence, `length(P(P>0))` gives the length of vector `P` over rainy hours only. The number of rainy hours in the dataset is 1210 and the average precipitation on a rainy hour is 23.00 mm/d (equivalent to 23.00/24=0.96 mm/h).

   Task: compute the total volume of precipitation, evapotranspiration and leakage that crosses the control volume.
Solution: as fluxes are considered constant during a timestep, one simply needs to sum them up. The only important thing here is to remember that fluxes are in mm/d but the timestep is hours, so fluxes have to be multiplied by the timestep Δt (otherwise the result is wrong by an unacceptable factor of 24). Total P = 1159.6 mm, total ET = 545.8 mm, total L = 643.6 mm.

   Question: does the balance close (meaning total inputs ≈ total outputs)?
Solution: as total P - total ET - total L = -29.80 mm we can say that the balance closes almost perfectly. Indeed, the missing 29.80 mm is less than 3% of total precipitation.

   Optional Question: Knowing that precipitation data has been taken from a city in canton Valais, is it more likely that it comes from Sion or from Brig? Why?
Solution: A total annual precipitation of roughly 1160 mm is not compatible with the relatively dry climate of Sion, that receives on average less than 600 mm per year, while it is fully compatible with the climate of Brig (and many other parts of Switzerland), where mean annual precipitation is about 1100 mm per year.

4. **Task:** compute and plot the evolution of the soil storage $S(t)$ over time by implementing and

solving a hydrologic balance. The initial soil moisture is $s_0 = 0.4$. Assume n and Z equal to those in exercise 1.
Solution: as the considered fluxes are constant during the timestep, the soil moisture evolution can be computed by implementing the discretized problem:
`S(i+1)=S(i)+(P(i)-ET(i)-L(i))*dt`, for `i=1:N-1` starting with `S(1)=s_0*Z*n`. Note again that the computed S values refer to the beginning of the timestep. An example of the result is shown in Figure 2.

Question: How often is $S(t)$ lower than 100 mm?
Solution: The condition $S < 100$ holds over 445 (non-consecutive) hours.

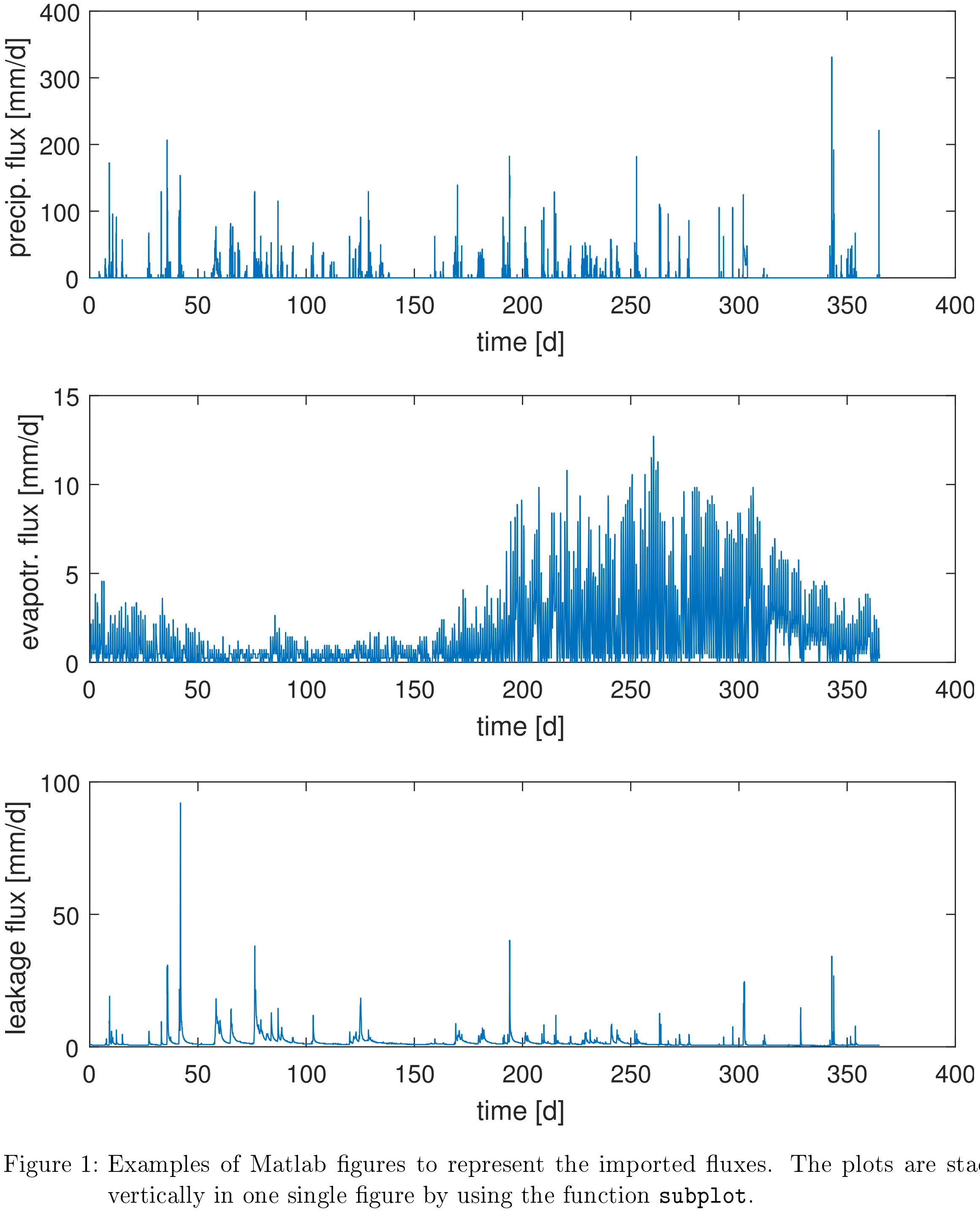

Figure 1: Examples of Matlab figures to represent the imported fluxes. The plots are stacked vertically in one single figure by using the function `subplot`.

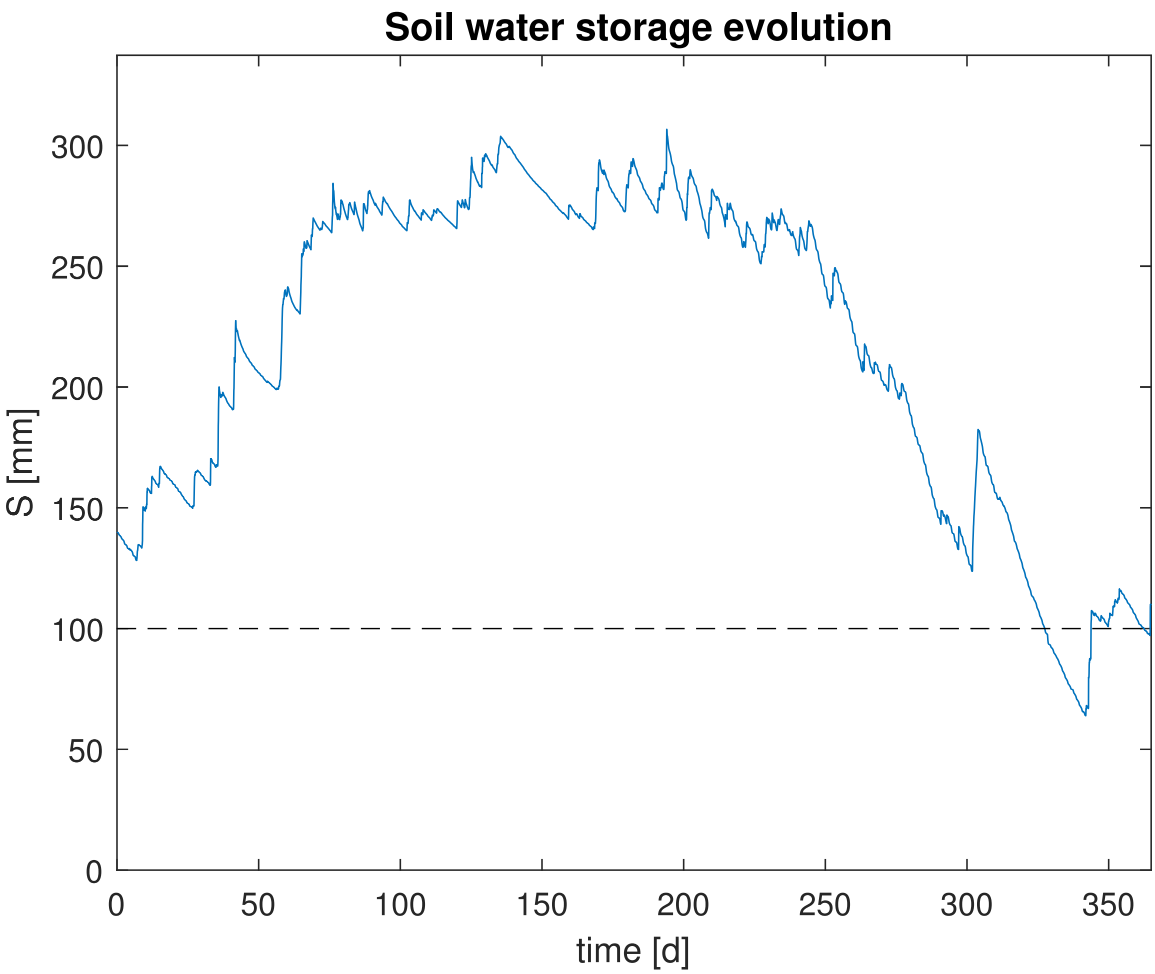

Figure 2: plot of soil moisture evolution with highlighted the level $S = 100$ mm.