ERAHUMED DSS

Valerio Gherardi   Pablo Amador Crespo   Andreu Rico

Table of contents

Preface

The purpose of this book is to provide a comprehensive reference for the ERAHUMED Decision Support System. Here you can find the technical descriptions of the algorithms employed by the system, as well as the user manual for the accompanying software.

The Support System and, hence, this book are currently under development on Github. In particular, the `{erahumed}` R package is hosted here.

For general information on the ERAHUMED project, please refer to the official website. If you want to get in touch, you can contact any of us via e-mail:

- Andreu Rico (Coordinator)
- Pablo Amador (PhD Researcher)
- Valerio Gherardi (Software Developer)

1 Introduction

This is a book created from markdown and executable code.

See Martínez-Megías et al. (2024) for additional info.

Part I

Technical description

2 The ERAHUMED model: a bird's eye view

3 ERAHUMED model components

This Chapter provides detailed descriptions of the various components of the ERAHUMED model, briefly introduced in Chapter 2.

3.1 INP: Input Data

The purpose of this model component is simply to collect the empirical data that provides the observational input to all subsequent modeling layers. This data consists of hydrological and meteorological time series data for the Albufera lake in the desired time frame.

3.1.1 Input

The only inputs to this modeling layer are the two aforementioned time-series data-sets.

Hydrological data consists of the direct measurements of the Albufera lake's daily water levels and outflows. Since the lake has three estuaries (*Gola de Pujol*, *Gola del Perellonet* and *Gola del Perelló*), this data amounts to four daily time series. The actual dataset bundled with the ERAHUMED software was obtained from the public data repository compiled by the *Confederación Hidrográfica del Júcar* (TODO: Ref.), and is described in more detail in Appendix A.1.

Meteorological data consists of the daily measurements of precipitation and evapotranspiration per unit are, and temperature (average, maximum and minimum). This amounts to five daily the series. The actual data bundled with the ERAHUMED software was obtained (TODO: where), and is described in more detail in Appendix A.2.

The time-series inputs described above are collected in the table below (the common index t refers to the time - *i.e.* day - of observation).

Table 3.1: Observational input to the ERAHUMED model (INP model component).

Input	Units	Description
L_t	m	Lake water level
O_t^{Pujol}	m^3	*Pujol* daily outflow
$O_t^{\text{Perelló}}$	m^3	*Perelló* daily outflow

Input	Units	Description
$O_t^{\text{Perellonet}}$	m^3	*Perellonet* daily outflow
P_t	mm	Precipitation (per unit area)
ETP_t	mm	Evapotranspiration (per unit area)
$T_t^{\max}$	°C	Maximum temperature
$T_t^{\min}$	°C	Minimum temperature
T_t^{ave}	°C	Average temperature

3.1.2 Output

This modeling layer does not involve any actual computation, and its output is simply a time-series data-set obtained as the combination of the two input data-sets, *i.e.* the collection of time series of Table 3.1.

3.2 HBA: Hydrological Balance of the Albufera lake

The purpose of this model component is to compute the total daily inflow to the Albufera lake. The relevant equation expressing the hydrological balance is:

$$\text{Volume Change} = \text{Inflow} - \text{Outflow} + \text{Precipitation} - \text{Evapotranspiration} \tag{3.1}$$

where the unknown is Inflow, whereas the remaining terms are obtained from observational data, as described in the previous Section (Section 3.1).

3.2.1 Input

The inputs specific to this layer are two numerical functions converting water heights into the water *volumes* that appear in Equation 3.1. Specifically:

- The lake's *storage curve* converts the measured lake's level into a total water volume.
- The *precipitation-evapotranspiration volume function* (or, for the sake of brevity, *P-ETP function*), that converts precipitation and evapotranspiration values *per unit area* into an overall water volume difference.

Even though the R interface to the ERAHUMED software allows arbitrary definitions, the graphical user interface assumes a linear approximation for both these functions. Therefore, the storage curve is assumed to take the form:

$$V_t = m \cdot L_t + q, \tag{3.2}$$

where V_t is the (daily) water volume of the lake, L_t the corresponding water level (*cf.* Table 3.1), and m and q are numerical coefficients[1]. Correspondingly, the P-ETP function reads:

$$\Delta V_t^{\text{P-ETP}} = \alpha \cdot \mathrm{P}_t - \beta \cdot \mathrm{ETP}_t, \tag{3.3}$$

where the left-hand side represents the relevant water volume difference, whereas the P and ETP terms in the right-hand side are the measured precipitation and evapotranspiration levels per unit area (*cf.* Table 3.1). The actual default values of m, q, α and β are documented in Appendix A.4.

3.2.2 Output

The output of this model consists of the time-series collected in the table below.

Table 3.2: HBA model component outputs

Variable	Units	Description
V_t	m^3	Lake water volume
I_t	m^3	Lake total inflow
O_t	m^3	Lake total outflow

3.2.3 Details

Using the notation introduced in the previous Subsections, we can rewrite Equation 3.1 as follows:

$$V_{t+1} - V_t = I_t - O_t + \Delta V_t^{\text{ETP}}, \tag{3.4}$$

where V_t and ΔV_t^{ETP} are computed through the storage and P-ETP functions, as in Eqs. 3.2 and 3.3. The only terms that requires further clarification in Equation 3.4 is O_t, the total outflow from the lake.

Strictly speaking, O_t is not merely the sum of O_t^{Pujol}, $O_t^{\text{Perelló}}$ and $O_t^{\text{Perellonet}}$, but is rather calculated as follows:

$$O_t = \max\left[O_t^{\text{Pujol}} + O_t^{\text{Perelló}} + O_t^{\text{Perellonet}},\ \Delta V_t^{\text{ETP}} + V_t - V_{t+1}\right]. \tag{3.5}$$

[1] In fact, for the purpose of computing volume *changes* entering the hydrological balance Equation 3.1, only the slope m is required.

The rationale is that the simple sum of estuaries outflows omits potentially important contributions from *water recirculation*, that is to say, water being pumped out from the lake for rice-field irrigation, by the so-called *tancats*. Such amount of recirculated water is hard to estimate and, in the lack of a better model, we simply assume this to be negligible, *except* when a positive amount is required by Equation 3.4 itself, due the physical constraint that $I_t \geq 0$.

Once O_t is calculated through Equation 3.5, I_t can be immediately obtained from Equation 3.4. Notice that whenever the aforementioned compensating outflow term due to water recirculation is included (which happens when the maximum in Eq. 3.5 is given by the second term), the total inflow is always estimated to be zero.

3.3 HBP: Hydrological Balance of rice Paddy clusters

This modeling layer simulates the local hydrology (*i.e.* water levels, inflows and outflows) of rice paddy clusters in the Albufera National Park. The simulation, that literally involves the generation of a synthetic data-set, is based on:

- The hydrology of the Albufera lake, which allows to constrain the total outflow of all rice paddies, assumed to be equal to the lake's total inflow (the main output of the HBA layer, see Section 3.2).
- An estimate of the fraction of the total lake's inflow that comes from each of the 26 ditches that flow into the Albufera.
- A subdivision of the rice fields surface into clusters that are assumed to be hydrologically connected with a single ditch - meaning that the sum of outflows of clusters from the same group must equal the total flow through the corresponding ditch.
- An ideal yearly management plan for the irrigation and draining of rice paddies.

3.3.1 Input

As already mentioned, the simulation is based on a spatial clustering, which is described in greater detail in Appendix A.5. For the present purposes, we stress that each cluster, consists of rice paddies of a definite rice variety, is labeled according to whether it is a *tancat* or not, and, most importantly, is assumed to be hydrologically connected with a single ditch.

The yearly management plan for these clusters, is described in Appendix A.3, and provides, for each day of the year d, the following:

- Ir_d: boolean expressing whether the cluster is supposed to be irrigated (if true) on day of year d.
- Dr_d: boolean expressing whether the cluster is supposed to be drained (if true) on day of year d.

- $\mathcal{H}_d$: ideal water level (in cm) of the cluster on day of year d.

As explained in Appendix A.3, the management plan depends on the rice variety and on the *tancat* label of a cluster.

In addition to these rather complex inputs, the model component requires the following numerical parameters:

- k_{flow} (*Ideal flow rate*). Rate at which water flows through rice paddies when these are being simultaneously irrigated and drained, with the overall level being kept constant. Expressed in $\text{cm} \cdot \text{day}^{-1}$.
- h_{thres} (*Height threshold*). Maximum allowed water level for a cluster to be considered emptied, used in the calculation of draining/irrigation plan delays. Expressed in cm.

Finally, it should be noted that the simulation involves random sampling (see Subsection 3.3.3), so that the seed of random number generation is an additional pseudo-parameter.

3.3.2 Output

The output of this layer provides a simulation for the local hydrology of each cluster, summarized in the table below

Table 3.3: HBP model component outputs

Variable	Units	Description
$h_{c,t}$	m^3	Cluster's water height
$I_{c,t}$	m^3	Cluster's inflow
$O_{c,t}$	m^3	Cluster's outflow

3.3.3 Detail

3.3.3.1 Ditch inflows

The first step consists in breaking down the total inflow to the lake, obtained through Equation 3.1, into inflows from individual ditches. Each paddy cluster is assumed to communicate with a single ditch (*cf.* Section A.5), and the flow through a ditch is assumed to be proportional to the total area of clusters belonging to said ditch. Therefore, denoting by I_i and A_i the inflow and area of ditch i, and by I the total inflow, we compute:

$$I_i = \frac{A_i}{\sum_j A_j} \cdot I \tag{3.6}$$

3.3.3.2 Main Algorithm

In the following, we focus on the set of clusters that communicate with a specific ditch, whose inflow (estimated according to Section 3.3.3.1) we denote by Q. The essence of the algorithm is to approximate as closely as possible the cluster's *ideal* inflows and outflows, with the constraint that the sum of the actual outflows from all clusters must equal Q.

Let us start by setting up some basic notation. We denote by:

$$h_c^{\mathrm{id}}(t), \quad I_c^{\mathrm{id}}(t), \quad O_c^{\mathrm{id}}(t), \tag{3.7}$$

the ideal water level, inflow and outflow of cluster c at time (*i.e.* day) t, and by:

$$h_c^{\mathrm{re}}(t), \quad I_c^{\mathrm{re}}(t), \quad O_c^{\mathrm{re}}(t), \tag{3.8}$$

the corresponding real quantities. The area of cluster c, that provides the conversion between water volumes and column heights, is denoted by A_c. Precipitation and evapotranspiration water column values are denoted by:

$$\mathrm{P}(t), \quad \mathrm{ETP}(t). \tag{3.9}$$

Finally, we will define below a *plan delay* accumulated for cluster c at time t, denoted $D_c(t)$. This is computed recursively along the iterations of the algorithm, and its role will be clarified below.

The local balance algorithm proceeds iteratively as follows. At day t, assume that $h_c^{\mathrm{re}}(t-1)$ and $D(t-1)$ have already been computed; a single iteration consists of the following steps, which we describe in full detail below:

- ***Step 1.*** Recovering the $h_c^{\mathrm{id}}(t)$, $I_c^{\mathrm{id}}(t)$ and $O_c^{\mathrm{id}}(t)$ from the irrigation and draining plan, applying the computed plan delay $D_c(t-1)$.

2. ***Step 2.*** Computing $h_c^{\mathrm{re}}(t)$, $I_c^{\mathrm{re}}(t)$ and $O_c^{\mathrm{re}}(t)$ enforcing the constraint $\sum_c O_c^{\mathrm{re}}(t) = Q(t)$.
3. ***Step 3.*** Computing the updated plan delays $D_c(t)$.

In order to initialize the iteration, we assume that at some initial time, say $t = 0$, we have $D_c(t=0) = 0$ $h_c^{\mathrm{re}}(t=0) = h_c^{\mathrm{id}}(t=1)$.

3.3.3.2.1 Step 1: ideal balance

The management data-set described Section A.3 provides the ideal water level for each cluster and for every day of the year. Denoting by $d(t)$ the day of the year corresponding to t, the relevant entry of the management data-set is that corresponding to day:

$$d'_c(t) = d(t) - D_c(t-1), \tag{3.10}$$

where $D_c(t-1)$ is the accumulated plan delay for thi cluster. Therefore we set:

$$h_c^{\mathrm{id}}(t) = \text{Planned water level on day of year } d'_c(t). \tag{3.11}$$

To compute ideal inflow and outflow, we require:

$$h_c^{\mathrm{id}}(t) = \max(0,\, h_c^{\mathrm{re}}(t-1) + \mathrm{P}(t) - \mathrm{ETP}(t)) + I_c^{\mathrm{id}}(t) - O_c^{\mathrm{id}}(t), \tag{3.12}$$

where we assume $I_c^{\mathrm{id}}(t) > 0$ and $O_c^{\mathrm{id}}(t) > 0$. Clearly, Equation 3.12 alone does not individually specify $I_c^{\mathrm{id}}(t)$ and $O_c^{\mathrm{id}}(t)$, but only their difference $\Delta_c(t) = I_c^{\mathrm{id}}(t) - O_c^{\mathrm{id}}(t)$. In order to fix both these quantities, we first define[2]:

$$\begin{aligned}(I_c^{\mathrm{id}}(t))^{(0)} &= \begin{cases} k & \text{cluster planned to be in flux on day of year } d'_c(t) \\ 0 & \text{otherwise} \end{cases}, \\ (O_c^{\mathrm{id}}(t))^{(0)} &= (I_c^{\mathrm{id}}(t))^{(0)} - \Delta_c(t).\end{aligned} \tag{3.13}$$

and, in order to ensure that flows are positive[3], we finally set:

$$\begin{aligned}O_c^{\mathrm{id}}(t) &= \max\{(O_c^{\mathrm{id}}(t))^{(0)},\, 0\} \\ I_c^{\mathrm{id}}(t) &= O_c^{\mathrm{id}}(t) + \Delta_c^{\mathrm{id}}(t)\end{aligned} \tag{3.14}$$

[2] The condition is again evaluated using the the management data-set, where the relevant variables are the `irrigation` and `draining` columns. The cluster is understood to be in flux if both `irrigation` and `draining` are `TRUE`.

[3] The positivity of $I_c^{\mathrm{id}}(t)$ below is ensured by the fact that $(O_c^{\mathrm{id}}(t))^{(0)} \geq -\Delta_c(t)$ by construction.

3.3.3.2.2 Step 2: real balance

Real flows are obtained from ideal ones (Equation 3.14) in such a way to satisfy the constraint:

$$\sum_{c} O_c^{\mathrm{re}}(t) = Q(t), \tag{3.15}$$

where the right-hand side is the total ditch flow computed earlier (Section 3.3.3.1). At each time-step t, the cluster's index set is randomly permuted[4], and the real flows are calculated as:

$$\begin{aligned} O_c^{\mathrm{re}}(t) &= \min\{O_c^{\mathrm{re}}(t),\, Q(t) - \sum_{c'<c} O_{c'}^{\mathrm{re}}(t)\} + \frac{\max\{0, Q(t) - \sum_{c'} O_{c'}^{\mathrm{re}}(t)\}}{\text{number of clusters}}, \\ I_c^{\mathrm{re}}(t) &= \max\{I_c^{\mathrm{id}}(t) - O_c^{\mathrm{id}}(t) + O_c^{\mathrm{re}}(t),\, 0\} \end{aligned} \tag{3.16}$$

In words, clusters are emptied in a random order within the allowed capacity of the corresponding ditch (*i.e.* its actual total flow) - if the sum of ideal outflows is less than capacity, the remaining outflow is equally shared among clusters. Using Equation 3.16, we finally determine the real water level achieved as:

$$h_c^{\mathrm{re}}(t) = \max(0,\, h_c^{\mathrm{re}}(t-1) + \mathrm{P}(t) - \mathrm{ETP}(t)) + I_c^{\mathrm{re}}(t) - O_c^{\mathrm{re}}(t), \tag{3.17}$$

to be compared with Equation 3.12.

3.3.3.2.3 Step 3: updating the plan delay

The purpose of the plan delay $D_c(t)$ is to allow all clusters to be emptied as required by the ideal management plan, which may be hindered on the originally scheduled days by the first of Equation 3.16, since this sets to zero the real outflows for some clusters whenever the ditch's flow is saturated.

The updated value $D_c(t)$ is obtained as follows. If $d(t)$ (the *actual* day of year) is outside of the window $W =$ [20th of April, 15th of October], we reset all $D_c(t) = 0$. Otherwise, if $h_c^{\mathrm{id}}(t) > 0$ or $h_c^{\mathrm{re}}(t) < H_{\mathrm{thres}}$, the plan delay is unchanged for this cluster: $D_c(t) = D_c(t-1)$. Finally, if $h_c^{\mathrm{id}}(t) = 0$ but $h_c^{\mathrm{re}}(t) > H_{\mathrm{thres}}$, we set $D_c(t) = D_c(t-1) + 1$.

[4]With some abuse of notation, we assume the indexes c and c' in Equation 3.16 to be sorted according to this random permutation.

3.4 CA: Chemical Applications

TBD.

3.5 CT: Chemical Transport

This model component describes the evolution of chemical masses in the three compartments of foliage, water and sediment. This is obtained by solving the system of differential equations that describes the dynamics of chemicals, through a semi-analytic approach with suitable approximations and simplifications.

3.5.1 Input

TBD.

3.5.2 Output

TBD.

3.5.3 Details

The evolution of masses in the foliage, water and sediment compartments is described by the following system of differential equations:

$$\begin{aligned}
\frac{\mathrm{d}m_f}{\mathrm{d}t} &= -(k_f + w)m_f + a_f \\
\frac{\mathrm{d}m_w}{\mathrm{d}t} &= -(k_w + d_w + s + \frac{O}{V})m_w + d_s m_s + w m_f + a_w - \sigma(\frac{m_w}{\rho V}) \\
\frac{\mathrm{d}m_s}{\mathrm{d}t} &= (d_w + s)m_w - (k_s + d_s)m_s + a_s + \sigma(\frac{m_w}{\rho V})
\end{aligned} \tag{3.18}$$

where:

- $a_{f,w,s}$ are the mass application rates of the chemical in the three compartments,
- $k_{f,w,s}$ are the degradation rates of the chemical in the three compartments,
- d_w and d_s are the water-sediment diffusion rates,
- s is the settling rate,
- w is the washout rate,
- O is the outflow rate of the rice field,

- V is the volume of water in the rice field,
- ρ is the chemical solubility in water,
- $\sigma(x)$ is a function (not further specified, see below) that grows quickly for $x > 1$, and vanishes for $x \leq 1$. This function accounts for solubility. (TODO: motivate the inclusion of exactly these processes)

Strictly speaking, all these terms have instantaneous time dependence[5]. Apart from making the system 3.18 hard to attack by analytic means, such a dependence is troubling because we don't have access to the exact time dependence of the majority of these terms (*e.g.* outflow, volume or chemical applications), our input consisting of simple daily average/cumulative values. On the other hand, if all the terms were constant, and if we could neglect σ, the solution of Equation 3.18 would be immediate, as the corresponding system becomes a linear ODE with constant coefficients, in addition whose eigenvalues and eigenvectors can be computed explicitly.

What we use in practice is an intermediate semi-analytic approach that allows us to compute daily values of $m_{f,w,s}$, which we can briefly summarize as follows. The time evolution of $m_{f,w,s}$ in a daily time-step (from t to $t+1$, say, assuming t is measured in days) is obtained through the following four consecutive stages:

1. We compute the exact evolution of $m_{f,w,s}$ according to the linear ODE obtained by disregarding the processes of outflow, chemical application and solubility, and using daily constant values for all the remaining constants involved (described in more detail below).
2. We compute the m_w losses due to outflow as if they happened instantaneously after the processes computed in the previous step took place, with the water volume of the rice paddy varying from $V(t) + O(t)$ to $V(t)$[6]
3. We compute the mass applications again as instantaneous, after the losses in the water compartment due to outflows took place.
4. We compare the resulting m_w with the maximum amount allowed by the solubility in water, that is $m_w^{\max}(t) = \rho V(t)$ and transfer any excess to the sediment compartment (m_s).

In the following, we describe in full detail the computations involved in these four steps.

3.5.3.1 Step 1. Linear ODE evolution (physico-chemical processes)

Disregarding outflow, mass application and solubility, we get a linear ODE of the form:

[5] This observation also includes pure physico-chemical "constants" such as degradation rates, where the time dependence would stem from *temperature* dependence.

[6] Recall that, in the notation of TODO, $V(t)$ denotes the final volume on day t.

$$
\begin{aligned}
\frac{\mathrm{d}m_f}{\mathrm{d}t} &= \gamma m_f, \\
\frac{\mathrm{d}m_w}{\mathrm{d}t} &= a_{ww}m_w + a_{ws}m_s + w m_f, \\
\frac{\mathrm{d}m_s}{\mathrm{d}t} &= a_{sw}m_w + a_{ss}m_s,
\end{aligned}
\tag{3.19}
$$

with:

$$
\begin{aligned}
\gamma &= -(k_f + w), \\
a_{ww} &= -(k_w + d_w + s) \\
a_{ws} &= d_s \\
a_{sw} &= d_w + s \\
a_{ss} &= -(k_s + d_s)
\end{aligned}
\tag{3.20}
$$

where the various physico-chemical parameters are assumed to be constant during a daily time-step (see Sec. TODO for the actual numerical values).

The evolution of m_f is easily obtained:

$$
m_f(t) = e^{\gamma t} m_f(0). \tag{3.21}
$$

Plugging this into the three-dimensional system 3.19 we obtain a reduced two-dimensional system for the water-sediment compartments:

$$
\begin{aligned}
\frac{\mathrm{d}m_w}{\mathrm{d}t} &= a_{ww}m_w + a_{ws}m_s + w e^{\gamma t} m_f(0), \\
\frac{\mathrm{d}m_s}{\mathrm{d}t} &= a_{sw}m_w + a_{ss}m_s,
\end{aligned}
\tag{3.22}
$$

which is solved explicitly with standard methods[7].

[7]Letting $x = (m_w,\, m_s)^T$ and $b = w(m_f(0), 0)^T$, the system can be rewritten in the form:

$$
\dot{x} = Ax + e^{\gamma t} b
$$

where

$$
A = \begin{pmatrix} a_{ww} & a_{ws} \\ a_{sw} & a_{ss} \end{pmatrix}
$$

The general solution reads:

$$
x(t) = e^{At}x(0) + (e^{At} - e^{\gamma t} I)(A - \gamma I)^{-1} b
$$

where I is the identity matrix. Notice that the exponential can be computed explicitly from the eigenvalues of A:

$$
e^{At} = e^{\lambda_+ t} P_+ + e^{\lambda_- t} P_-,
$$

with:

$$\lambda_{\pm} = \frac{\mathrm{Tr}A}{2}\sqrt{\left(\frac{\mathrm{Tr}A}{2}\right)^2 - \det A}$$

and

$$P_{\pm} = \pm\frac{1}{\lambda_+ - \lambda_-}(A - \lambda_{\mp})$$

Part II

User Manual

4 The ERAHUMED DSS User Interface

5 The {erahumed} R package

References

Martínez-Megías, Claudia, Alba Arenas-Sánchez, Diana Manjarrés-López, Sandra Pérez, Yolanda Soriano, Yolanda Picó, and Andreu Rico. 2024. "Pharmaceutical and Pesticide Mixtures in a Mediterranean Coastal Wetland: Comparison of Sampling Methods, Ecological Risks, and Removal by a Constructed Wetland." *Environmental Science and Pollution Research* 31 (10): 14593–609.

A Input Data

A.1 Hydrological data

A.2 Meteorological data

A.3 Albufera Rice Paddies Management

A.4 Storage curve and P-ETP function

A.5 Definition of rice clusters