**User guide of *SurEau- Ecos v1.0.0***

Julien Ruffault ([julien.ruff@gmail.com](mailto:julien.ruff@gmail.com))

François Pimont ([francois.pimont@inrae.fr](mailto:francois.pimont@inrae.fr))

Hervé Cochard ([herve.cochard@inrae.fr](mailto:herve.cochard@inrae.fr))

Nicolas Martin-StPaul ([nicolas.martin@inrae.fr](mailto:nicolas.martin@inrae.fr))

*To report any bugs or to ask support, please contact Julien Ruffault and Nicolas Martin-StPaul*

# Introduction

*SurEau-Ecos* is a mechanistic plant water model designed to simulate water fluxes, fuel moisture content and plant mortality at stand level according to a set of vegetation traits and soil physical properties This manual presents the essential steps to configure and run *SurEau-Ecos* in *R* according to simulation goals. Note that this manual does not aim to describe the processes implemented in *SurEau-Ecos* (this is the subject of a more specific document that will soon be available). Besides, to ease the use and configuration of *SurEau-Ecos*, the model will also be delivered as a R package.

# General organization

*SurEau-Ecos* V1.0.X can be downloaded at the following link (download the last release):

[**https://github.com/julien-ruffault/SurEau-Ecos**](https://github.com/julien-ruffault/SurEau-Ecos)

*SurEau-Ecos* is delivered as a zip file. Directories organization of files within the main directory must not be changed.

Before starting, the directory of the model (*mainDir*) should be defined, such as :

mainDir = "/Users/Name/SurEau-Ecos\_v1.0.0"

Then, source all functions in your R environment with the following line of code :

source(paste0(mainDir,'/functions/load.SurEau-Ecos.R'))

*Sureau-Ecos* can be run with the function run.SurEau-Ecos(). This function has **six** different arguments containing the required variables to run the model. All these arguments must be specified and declared in the following order. :

1. modeling options

2. simulation parameters

3. climate data

4. stand parameters

5. soil parameters

6. vegetation parameters

The R script ‘*quick\_start/example\_launcher..R*’ is an example on how to run a basic simulation of *SurEau-Ecos* (with a ‘standard’ configuration of modeling options) that can be used by the users as a template to build their own simulation.

The six input lists of *SurEau-Ecos*’s parameters and how they can be initialized are described in the following chapters. A more specific description on how vegetation and soil is represented in *SurEau-Ecos* and how to determine those parameters will soon be provided in a more specific documentation.

# Modeling options

Modeling options indicate the options for the implementation of *SurEau-Ecos* and allows the user to choose between several mechanisms affecting the behavior of the model.

The list of modeling options needed to run the fonction run.SurEau-Ecos() must be created by the function create.modeling.options(). The function takes several arguments as input (see Table 1). Note that some of these options might also change the list of climate data and the stand, soil or vegetation input parameters that are required to run the model (e.g. StomatalRegFormulation determined the type of stomatal regulation and hence the vegetation parameters that should be provided).

*Examples*

# basic configuration of *SurEau-Ecos*

modeling\_options = create.modeling.options()

# configuration to run SurEau-Ecos with a constant climate

modeling\_options = create.modeling.options(constantClimate=T)

|  |  |  |
| --- | --- | --- |
| **R parameter** | **Description** | **Comment** |
| constantClimate | a logical value indicating whether a constant climate will be used during the simulation (default = F). If set to ‘T’, the first line of the climate input file will be repeated to generate the climate data.frame in create.climate.data() |  |
| timeStepForEvapo | a numerical value (in hours) indicating the time step for the main evapotranspiration loop. Should be one of the following 1,2,4,6,8 (default = 1).  NB: This is the time step at which climate data are sampled (before being interpolated in the small time step loop). Computations becomes inaccurate for timeStepForEvapo>2h, as the range of daily variations of climate conditions and associated stomatal regulation and fluxes are not well discretized | for advanced users only |
| numericalScheme | Implicit' (default) or 'Semi-Implicit' | for advanced users only  (see Section 8) |
| compOptionsForEvapo | "Normal" (Default): adaptive time step (10, 6, 3, 1 min)  "Accurate" : fixed time step (10 s)  "Fast" : adaptive time step (1 hour, 10 min)  "Custom" : specify your small time step (parameter “customSmallTimeStepInSec”, default is 600 s =10 min) | for advanced users only (see Section 8) |
| soilEvap | A logical value to activate or deactivate the computation of soil evaporation. NB: this is computed on the large time step and might raise convergence issues on large time steps larger than 2h. | for advanced users only |
| Lcav and Tcav | Water released by cavitation is by default redistributed to adjacent cells (for respectively Leaf and Trunk apo). Default is 1 (for redistribution). 0 corresponds to absence of redistribution. | for advanced users only |
| ETPFormulation | the formulation of ETP to be used, either 'PT' (Priestley-Taylor, Default) or 'P' (Penman). | Penman formulation not implemented yet |
| RnFormulation | The method to be used to derive net radiation from global radiation, either 'Linacre' (default) or 'Linear' | the linear method is not implemented yet |
| stomatalRegFormulation | The type of regulation to be used for stomatal response to leaf symplasmic water potential, either 'Sigmoid' (default) 'PiecewiseLinear' or ‘Turgor’. |  |
| avoidWaterSoilTransfer | a logical value indicating whether the transfer of water between soil layers should be avoided by disconnecting the soil layers that get refilled from the soil-plant system (default =F). | Not implemented |
| defoliation | a logical value indicating whether trees should loose leaves when occurs.cavitation occurs of the above part of plant (default =F) . Defoliation starts only when PLC\_TL > 10% . |  |
| resetSWC | a logical value indicating whether soil layers should be refilled at the beginning of each year (default=F) |  |

# Simulation parameters

Simulation parameters indicate the specified time period and also configure the writing and format of the output simulation file. Simulation parameters must be created by the function create.simulation.parameters()(see the list of arguments in Table 2).

startYearSimulation and endYearSimulation arguments specify the time period for the simulation and must be set according to the period covered by the climate input data (see section 5).

Output data is written in an output csv file whose path must be specified by the argument outputPath. outputResolution specifies which one of the three resolutions is chosen for output writing in file : ‘subdaily’ (i.e., similar timestep of timeStepForEvapo specified in modeling options), ‘daily’ or ‘yearly’ time scale.

The type of output that must be written is specified by the outpuType argument. By default,a “simple'' output type is chosen (‘simple\_subdaily’, ‘simple\_daily’ or ‘simple\_yearly’ type) according to the chosen resolution. Two more output types are already implemented for the sub-daily time scale, ‘diagnostic subdaily’ (which writes all possible outputs) and ‘LFMC-subdaily’ (used for fuel moisture simulation purposes). Alternatively, users can also specify their own output configuration. In that case, output names should be provided as a csv file with ‘;’ as separator and placed in the “functions/ouput\_types' directory.

**Examples**

# create simulation parameters to run SurEau-Ecos on the period from 1990 to 1992 with ‘LFMC’ output type at the subdaily time scale

output\_path = paste0(mainDir, '/quick\_start/example\_output\_subdaily.csv')

simulation\_parameters <- create.simulation.parameters(startYearSimulation = 1990,endYearSimulation = 1992, mainDir = mainDir, outputType = 'LFMC\_subdaily', outputPath = output\_path)

|  |  |  |
| --- | --- | --- |
| **R parameter** | **Description** | **Comment** |
| mainDir | Main directory of the model |  |
| startYearSimulation | a numeric indicating the starting year for the simulation (must match the dates of the input climate file) |  |
| endYearSimulation | a numeric indicating the last year for the simulation (must match the dates of the input climate file) |  |
| resolutionOutput | the resolution for the output simulation file. Must be 'subdaily' (default), 'daily' or 'yearly' |  |
| outputType | the type of output chosen. if not provided set to ‘simple\_subdaily’, ‘simple\_daily’ or ‘simple\_yearly’ according to resolutionOutput. |  |
| outputPath | the path of the output simulation file |  |
| overWrite | a logical value indicating whether the chosen output path can be overwritten if it already exists (default = F) |  |

# Climate data

Weather input data must include variables at the **daily** scale. Daily data will be disaggregated at the time step specified in modeling options (R parameter : timeStepForEvapo). Note that at this stage, **it is not possible to use subdaily data** as input in *SurEau-Ecos*. This will be fixed in the next version.

Climate data used as input in run.SurEau-Ecos.R() must have been created by the function create.climate.data(). this function takes three arguments : filePath (path a csv file containing the climate data), modeling\_options (a list containing the modeling options created with create.modeling.options()) and simulation\_parameters a list containing the simulation parameters created with create.simulation.parameters())

Weather data should be arranged in a csv file with days in rows and variables in columns, ‘;’ as field separator character and ‘.’ as decimal character. For each row the Date must be provided in the following format : “dd/mm/yyy”. Only the data on corresponding to startingYear and endYear specified in the list of simulation parameters will be retained. Table 3 indicates the symbols, units, definitions and variable name.

*example*

# load climate data from test and select the period from 2005 to 2006

climate\_path = paste0(‘mainDir’, test\_simulation/climate\_data\_test.csv’)

modeling\_options = create.modeling.options()

simulation\_parameters = create.simulation.parameters(starting\_year =2005 ,endYear = 2006)

climate\_data = create.climate.data(filePath = climate\_path, modeling\_options, simulation\_parameters)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Symbol** | **Unit** | **R parameter** | **Description** | **Comment** |
| *Date* | *-* | DATE | Date of the day ( "dd/mm/yyyy") | not needed if 'constantClimate' is set to T |
| *Tmean* | *°C* | Tair\_mean | Mean daily temperature |  |
| *Tmin* | *°C* | Tair\_min | minimum daily temperature |  |
| *Tmax* | *°C* | Tair\_max | maximum daily temperature | - |
| *Rg* | *MJ.m2* | Rg\_sum | daily global radiation | - |
| *ppt* | *mm* | PPT\_sum | precipitation | - |
| *RHmean* | *%* | RH\_mean | mean daily relative humidity | - |
| *RHmin* | *%* | RH\_min | minimum daily relative humidity | - |
| *RHmax* | *%* | RH\_max | maximum relative humidity | - |
| *u* | *m.s-1* | WS\_mean | mean daily wind speed | - |

# Soil parameters

Soil is represented as a three-layer bucket whose physical properties allow the estimation of soil water retention and hydraulic properties. Note that in *Sureau-Ecos,* the word soil refers to the depth that plant rooting systems can reach, including cracks within the bedrock. Specifying layers with an elevated rock fragment content may be important in seasonally-arid climates like the Mediterranean, because plants often extend their roots into cracks existing in the parent rock to access water during summer (Rambal 1993, Ruffault *et al.* 2013, Caceres *et al.* 2015).

By default the soil is represented by a 3-layer bucket model of the following depths : 0-0.3 m, 0.3 m - 1 m and 1 m - 4 m. Depth of soil layers can be changed when creating soil parameters (see example below) but not the number of soil layers. Variations of soil and rhizosphere conductance (ksoil), and mean soil water potential in the root zone (Psi\_soil) are calculated with the van Genuchten–Mualem equations (Mualem 1976; van Gen-uchten 1980), from the unsaturated hydraulic conductivity of the soil (Ksat), scaled to the rhizosphere according to the Gardner–Cowan formulation (Gardner 1964; Cowan 1965). Details and equations can be found in Martin-StPaul et al. (2017). Note that some of the roots parameters used for the Gardnar-Cowen formulation are defined in vegetation parameters (see section 7).

Soil parameters should be provided as a csv file with ‘;’ as separator and ‘.’ for decimal. The following table indicates the symbols, unit and description of each soil parameter. A specific documentation will soon be provided to help parametrize soil in *SurEau-Ecos*.

*example*

soilPath=‘/quick\_start/soil\_example.csv' # path to the example soil dataset

# Create a soil with three similar soil of 0.2 meters depth each

soil\_parameters=create.soil.parameters(filePath=soilPath,depths=c(.2,.4,.6))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Symbol** | **Unit** | **R parameter** | **Description** | **Comment** |
| rfc1 | % | RFC\_1 | Rock fragment content of the 1st layer |  |
| rfc2 | % | RFC\_2 | Rock fragment content of the 2nd layer |  |
| rfc3 | % | RFC\_3 | Rock fragment content of the 3rd layer |  |
| fc | cm3.cm-3 | field\_capacity | soil water content at field capacity | Can be estimated using texture and bulk density |
| 𝛂 | MPa-1 | alpha\_vg | shape parameter (Van-Genuchten model) | “ |
| n | - | n\_vg | Pore size distribution index (Van-Genuchten model) | “ |
| I | - | I\_vg | shape parameter (Van-Genuchten model) | “ |
| Ksat | mmol.m-2.s-1.MPa-1 | Ksat | Soil hydraulic conductivity at saturaton | “ |
| sc | cm3.cm-3 | saturation\_capacity\_vg | Soil water content at saturation | “ |
| rc | cm3.cm-3 | residual\_capacity\_vg | residual soil water content | “ |
| gsoil,0 | mmol.m-2.s-1 | gSoil0 | conductance at soil saturation |  |

1. **Vegetation parameters**

Vegetation parameter in SurEau-Ecos model, are related to four different functions :

(1) Parameters related to the hydraulic scheme that computes water flows, water stocks, hydraulic conductance, plant water potential within the plant

(2) Parameter related to (i) the transpiration from the leaf to the atmosphere that are related to the leaf area (LAI) and to the stomatal and aerodynamic conductance of the crown and (ii) and from the stem to the atmosphere (related to the wood area and trunk conductance to water vapour).

(3) Parameters related to the leaf phenology determining the flushing.

(4) Parameters related to canopy microclimate (rainfall interception by the canopy, light extinction coefficient…).

An example of the parameters needed is provided in the table below and a description

|  |  |  |  |
| --- | --- | --- | --- |
| **R parameters** | **Value** | **Unit** | **Comment** |
| P50\_VC\_Leaf | -3.4 | Mpa | # P50 of the vulnerability curve of the leaves |
| slope\_VC\_Leaf | 60 | %/Mpa | # Slope of the vulnerability curve of the leaves |
| EpsilonSymp\_Leaf | 10 | Mpa | # Modulus of elasticity of the leaf symplasm (derived from PV curve) |
| PiFullTurgor\_Leaf | -2.1 | Mpa | # Osmotic potential at full turgor of the leaf symplasm (derived from PV curve) |
| ApoplasmicFrac\_Leaf | 0.4 | - | # Apoplasmic fraction of the leaf |
| P50\_VC\_Trunk | -3.4 | Mpa | # P50 of the vulnerability curve to cavitation of the xylem of the root and trunk |
| slope\_VC\_Trunk | 60 | %/Mpa | # Slope of the vulnerability curve to cavitation of the xylem of the root and trunk |
| PiFullTurgor\_Trunk | -2.1 | Mpa | # Modulus of elasticity of the root-to-trunk symplasm |
| EpsilonSymp\_Trunk | 10 | Mpa/% | # Osmotic potential at full turgor of the root-to-trunk symplasm |
| ApoplasmicFrac\_Trunk | 0.4 | - | # Apoplasmic fraction of the sapwood of the root-to-trunk |
| SymplasmicFrac\_Trunk | 0.2 | - | # Apoplasmic fraction of the sapwood of the root-to-trunk |
| LDMC | 0.5 | fraction | # Leaf or shoot dry matter content. Fraction of shoot dry mass per saturated mass. |
| Succulence | 100 | g/m2 | # Leaf or shoot water content per unit leaf area |
| LMA | 100 | g/m2 | # Leaf mass per area amount of leaf or shoot dry matter per unit leaf area. |
| kPlantInit | 0.62 | mmol/m2/s/Mpa | # K\_axial (symplasm and apoplasm included) from root tip up to leaf symplasm (included). Derived from SapFlow and Water potential |
| k\_TSymInit | 0.26 | mmol/m2/s/Mpa | # K\_Symplasm (radial) of branch+Trunk+Root on a leaf area basis derived from average tree dimension and specific conductivity. Can set a high value to neglect its effect. |
| volumeLiving\_TRB | 39,87 | (mm or sapwood litre/m2 of soil) | # Living wood water content (in trunk, branches and roots) at saturation (maximal) **per unit soil area.** This value is multiplied by ApoplasmicFrac\_Trunk and SymplasmicFrac\_Trunk to obtain the symplasmic and apoplasmic water volumes. So far this value must be estimated by the user from inventory allometric data + wood density + sapwood water content. **To be divided by LAI to compute on a leaf area basis.** |
| fRootToLeaf | 1 | - | #ratio of RAI/LAI (note that LAI is given in the Stand Parameters) |
| rootRadius | 0.0002 | m | #Root radius (used to compute root area) |
| betaRootProfile | 0.97 | - | #Parameters setting the root distribution with soil depth |
| C\_LApoInit | 1,39E-01 | mmol/m2 leaf/Mpa | #leaf apoplasmic capacitance (set a very low value if not known) |
| C\_TApoInit | 1,39E-02 | mmol/m2 leaf/Mpa | #trunk and branches apoplasmic capacitance (set a very low value if not known) |
| gCrown0 | 45 | mmol/m2/s | #conductance to water vapour of the crown (aerodynamic conductance) |
| gsMax | 200 | mmol/m2/s | #maximal stomatal conductance to water vapour (Jarvis) |
| gsNight | 20 | mmol/m2/s | #night time stomatal conductance to water vapour (Jarvis) |
| JarvisPAR | 0.006 | - | #stomatal response to light (Jarvis) |
| Tgs\_sens | 17 | - | #Stomatal sensitivity to temperature (Jarvis) |
| Tgs\_optim | 25 | - | #Optimal temperature for stomatal conductance (Jarvis) |
| P12\_gs | -2.07 | Mpa | #water potential for incipient (12%) stomatal closure |
| P88\_gs | -2.62 | Mpa | #water potential for full (88%) stomatal closure |
| turgorPressureAtGsMax | 1 | Mpa | #Turgor pressure at maximal stomatal conductance. This parameter is used to apply a stomatal regulation through turgor pressure |
| gmin20 | 4 | mmol/m2/s | # Minimal conductance to water vapour |
| TPhase\_gmin | 37.5 | °C | # Threshold temperature for phase transition of gmin |
| Q10\_1\_gmin | 1.2 | mmol/m2/s /10°C | # Q10 for gmin dependance to temperature before Tphase\_gmin |
| Q10\_2\_gmin | 4.8 | mmol/m2/s /10°C | # Q10 for gmin dependance to temperature after Tphase\_gmin |
| fTRBToLeaf | 0.8 | m2 wood exposed/m2leaf | #ratio of wood and branches area to leaf ==> to compute trunk Emin. To be estimated by the user from inventory and allometric data |
| gmin\_T | 3 | mmol/m2/s | #Trunk and branches minimal conductance |
| Foliage | Evergreen | - | #Foliage type |
| DayStart | 55 | doy | #Day to start accumulating temperature |
| Tbase | 3 | °C | #Minimum temperature to start cumulating temperature for budburst |
| Fcrit | 450 | °C | #Critical sum temperature for bud burst |
| nbdayLAI | 21 | days | #Number of days to achieve LAImax (note that LAI max is given in the stand parameters) |
| PTcoeff | 1.14 | - | #Prstley taylor coefficient for computing potential evapotranspiration (Not used in this version) |
| CanopyStorageParam | 1.5 | mm/m2 leaf | # Water storage in the canopy |
| K | 0.5 | - | # extinction coefficient to compute ETP and VPD at soil level |

1. **Parameters related to the hydraulic scheme**

SurEau-Ecos relies on the SurEau principles (Martin-StPaul et al 2017, 2020; Cochard 2019, Cochard et al, 2021), and is based on a representation of the plant as different organs that have an apoplasm and a symplasm. What we call a **compartment** is an apoplasm or a symplasm within an organ. Roughly, the apoplasm is considered to be made of cell-walls and xylem lumen and the symplasm is made of living cells and phloem. During daily and seasonal fluctuation of water status (i.e. *water potential* in SurEau-Ecos) both contribute to desiccation through different mechanisms: the symplasm is affected by cell elasticity and osmotic pressure whereas the apoplasm is affected by cavitation which leads to xylem embolism. Cavitation has two effects: it produces a decrease of the hydraulic conductance of the compartment thereby affecting water flows, and it also produces a release of water that decreases the moisture content of the tissue and that limits water stress by dampening temporally the water potential decline.

Contrasting with SurEau.C, which includes 4 different organs (roots, trunk, branches, leaves) -- and thus 8 different compartment -- SurEau-Ecos is simplified and has only **two main organs** : (1) the **canopy leaves**, and (2) a **trunk** which in fact refers to the sapwood of the branches, the trunk and the roots.

Thus, SurEau-Ecos considers four compartments : leaf apoplasm and leaf symplasm, trunk apoplasm and trunk symplasm. For each of these compartments the model solves a water balance equation at each time step (not presented here). The water balance accounts for (1) the exchange between a compartment and the adjacent compartment – that are computed from conductance and water potential gradients with the adjacent compartments-- (2) the transpiration and water leaks (i.e. water fluxes going out of the plant) that are specific to the symplasm and (3) internal changes of water content related to capacitance and (4) the water release from cavitation which occurs only in the apoplasm.

A set of traits allows to define the symplasm and apoplasm water relation. The fraction of symplasm and apoplasm is defined by the apoplasmic fraction for the leaves and the trunk. The symplasm is characterized by the pressure volume curve parameters (Epsilon\_Symp, PiFullTurgor) and the apoplasm by the vulnerability curve to cavitation (P50, and Slope).

>To simplify the parameterization of the hydraulic patchway, mainly one **total plant hydraulic conductance per unit leaf area** is required. However, kPlantInit is internally distributed among plant organs (k-root-to-trunk-apoplasm and k-trunk-to-leaf-apoplasm and k-leaf-symplasm) assuming that 50% of the resistance is belowground in accordance with the root distribution between the three soil layers, and 50% is located aboveground and mostly in the leaf symplasm. To parameterize the root area a root-shoot is used and the beta factor from (Jackson et al 1996) allows to set a vertical distribution with soil depth. In addition, the model requires a value of conductance per unit leaf area of the trunk symplasm (k\_TSymInit). This value allows to regulate the exchange between the symplasmic water content of the trunk and the ascendant water flow. If, which would be normal, you don’t know what to set for k\_TSymInit, keep the proposed value that you can eventually rescale with your leaf area index. You can also increase this value to neglect its effect.

> A set of traits are used to inform the water volume in the apoplasm and the symplasm of each organ (Leaves and trunk). The leaf canopy water content is computed internally by using the LAI (which is set in the *stand parameters*) in combination with the LDMC and LMA of the species. For the trunk water content, however, the user has to set himself a value of water in *mm.* This value should be derived from his own knowledge of the stand and the species. Typically, this can be derived from inventory + allometric relationships + density + sapwood moisture content estimates.

1. **Parameters related to the transpiration**

Leaf stomatal transpiration is computed using the Jarvis model (other options are coming), minimal transpiration of the leaf is computed from leaf minimal conductance and its relationship to temperature (a two phase Q10 proposed by Cochard 2020).

Parameters that are used to compute the transpiration at the leaf level include: the maximal stomatal conductance of the species, the parameters of stomatal response to light and to temperature (from the Jarvis model). By default the stomatal conductance response to water deficit is derived from an empirical sigmoidal curve that is parameterized using the water potential causing 12% loss of conductance (P12gs) and the water potential causing 88% loss of conductance (P88gs). Other options can be used including a control by leaf symplasm turgor loss (which is computed with the pressure volume curves parameters of the leaf symplasm) or a piecewise function. See the Modellinng\_option to set the stomatal regulation. Note that the leaf area index (LAI) which is used to upscale leaf transpiration at the stand level is set in the *stand parameters*.

Parameters used to compute the trunk transpiration includes the minimal conductance of the trunk to water vapour (gmin\_T) and a scaling coefficient allowing to compute the the branch and trunk area exposed to air from the LAI (fTRBToLeaf). This can be estimated by the user or left as a default.

1. **Parameters related to the plant phenology**

The model uses a simple, one phase, temperature forcing phenology model. The principle is to accumulate degrees every day until a threshold for budburst. This option is used only for deciduous species. There are four parameters including : Foliage (evergreen vs deciduous), the base temperature to start accumulating degree (Tbase), the critical sum of temperature for budburst (Fcrit) and a day of the year when temperature start to be cumulated (dayStart). In any case, once a budburst has occurred, the leaf will grow to reach maximal LAI in a preset time (parameter nbdayLAI, in days), independently of the climate or the species (this will be refined later on).

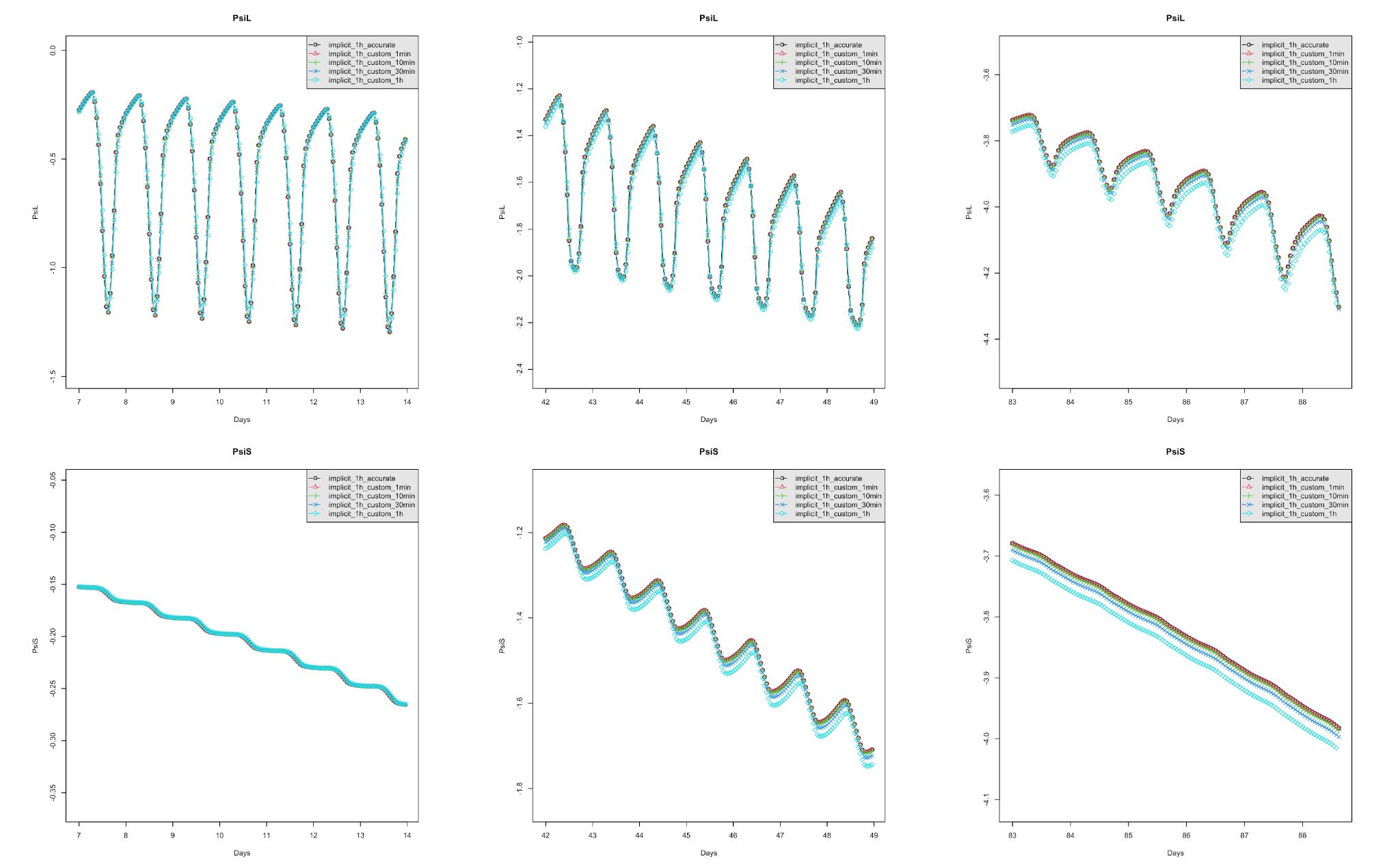
1. **Parameters related to microclimate of the canopy**

Such parameters helps to compute the water balance of the stand, they include a canopy storage parameter (CanopyStorageParameter) to compute the water intercepted by the canopy when it rains, an extinction coefficient to compute ETP at the soil level (*k*), the priestley-taylor coefficient (PTCoeff) for computing potential evapotranspiration (not used in this version)

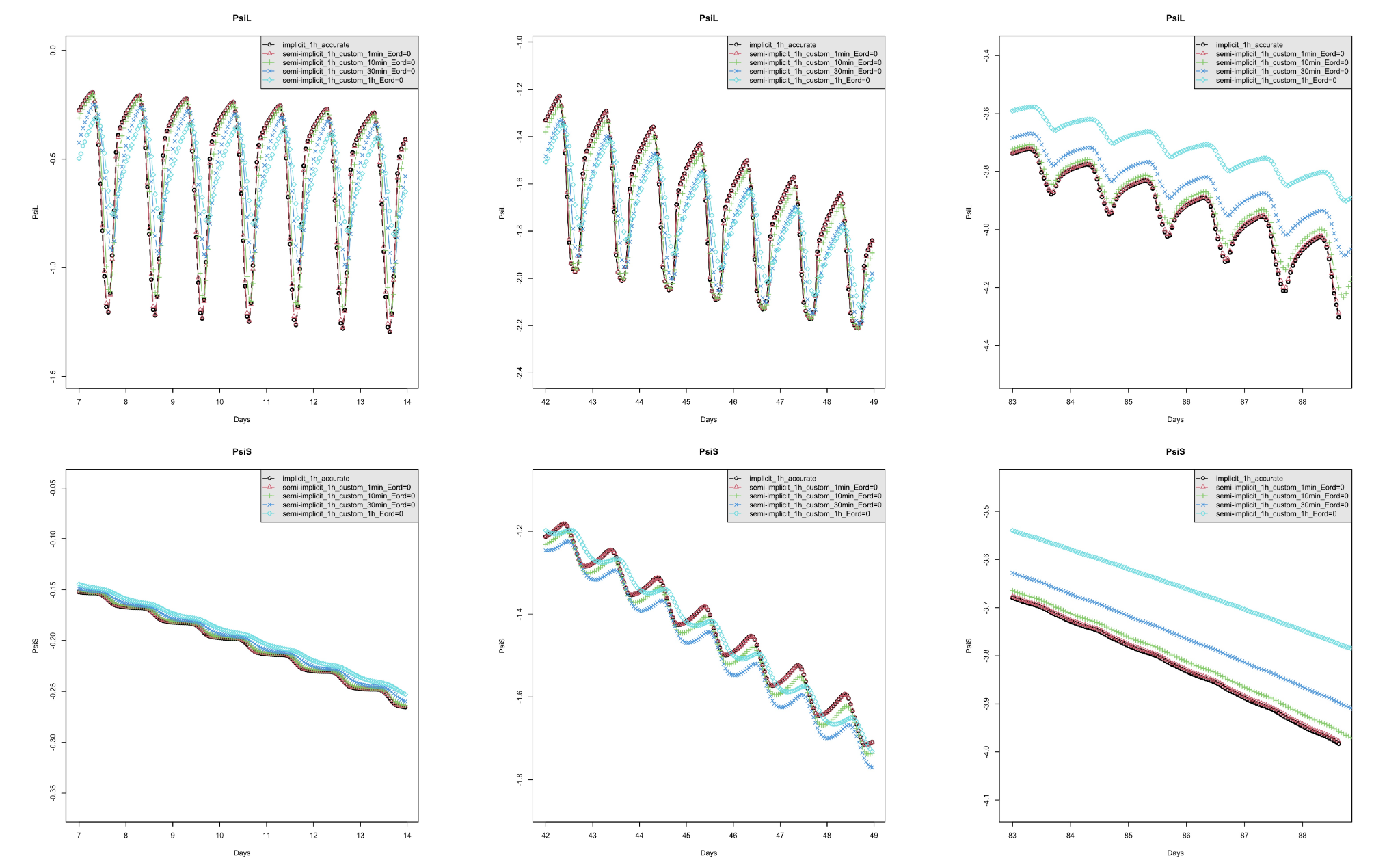
1. Numerical schemes and stability analyses

We did not implement an explicit numerical scheme in Sureau-Ecos, as done in the C Sureau version (Cochard *et al.* 2021), which estimates water fluxes from the current values of water potentials, because the computational costs would have been too prohibitive for the R language (on the order of 1 day for a very basic simulation) as the CFL constraint imposes a time step on the order of 10 ms (See theoretical documentation). Instead, we rely on an implicit scheme, which estimates water fluxes from the values of water potential taken at the next time step (still unknown), hence requiring to solve a (linearized) system of equations, in which water potentials at the next time steps are the unknowns (see theoretical documentation).

We implement two implicit numerical schemes. The default version is a fully-implicit scheme (numericalScheme=”Implicit”), which is specific to Sureau-Ecos. The second one, called semi-implicit one (numericalScheme=“Semi-Implicit”), is inspired from the work of Xu *et al.* (2016) and Tuzet *et al.* (2017). The latter scheme assumes that water fluxes can be estimated from the water potential of the next time step for the cell of interest, but for the current time step for adjacent cells. This approximation allows to overcome the CFL limitation of an explicit code, but is less accurate than the “Implicit scheme”, so that smaller time steps are required for the convergence of the model. Numerical explorations shows that the semi-implicit scheme requires time steps on the order of 1 min (which is slightly slower than described in Xu *et al.* (2016) which stated that 10 minutes was enough), whereas the time step can be generally larger than 30 minutes with the implicit scheme (see fig 8.1 for Implicit and 8.2 for Semi-Implicit). Also the survival time is quite sensitive to time steps with the Semi-Implicit scheme (increasing from 89 to 96 days), whereas they remain equal to 89 days for the different time steps tested with the Implicit scheme, including the largest one (1h, so that small and large time steps are identical).

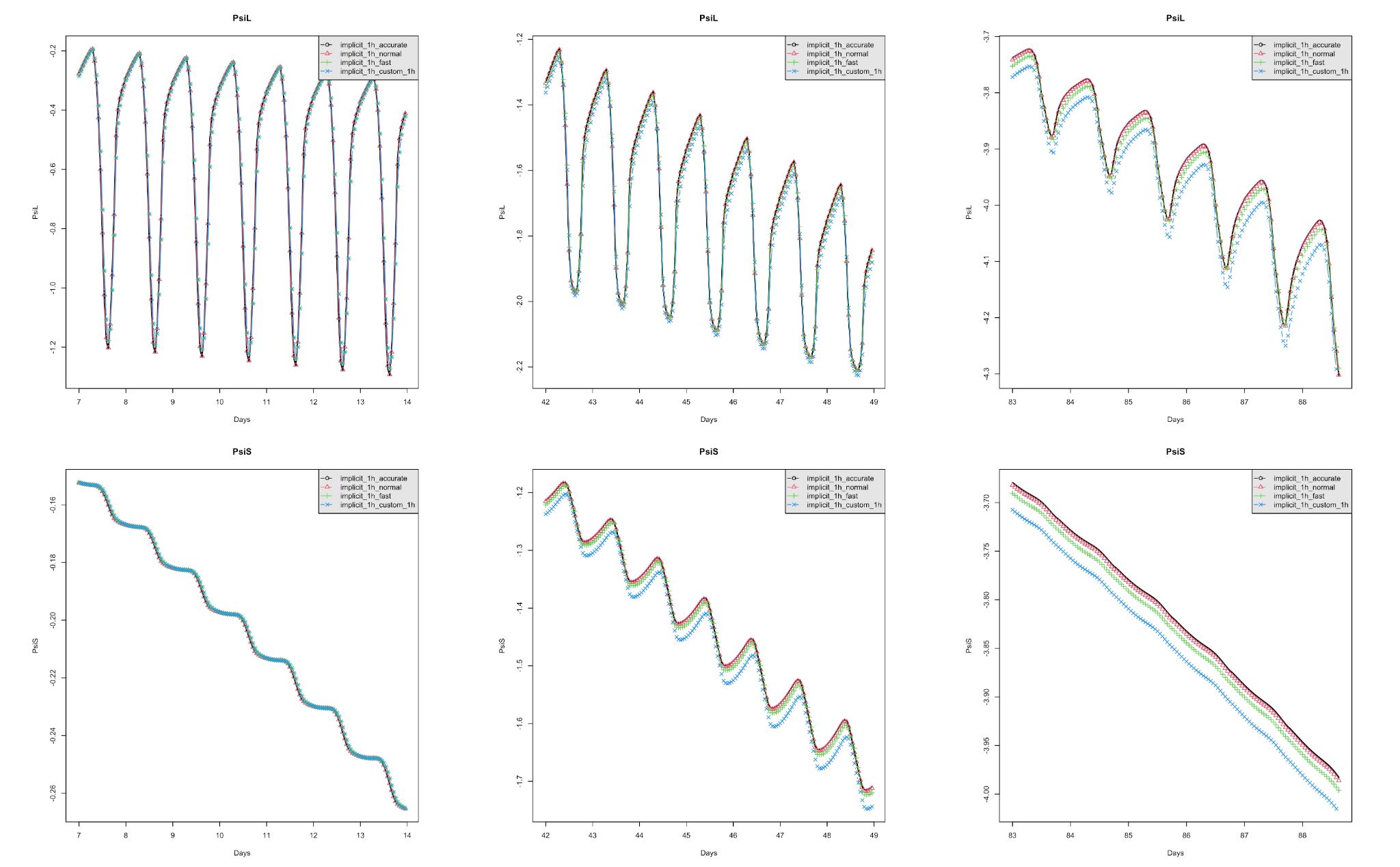


**Fig. 8.1.** Numerical stability of the Implicit scheme: “implicit\_1h\_accurate” is here for reference, as computed with a very small small time step (10 s); the other lines correspond to small time steps ranging from 1 min to 1h. The stability is very good up to 30 min. In particular, the survival time remains equal to 89 days for the reference and all the time steps up to 1h with the “Implicit scheme”.



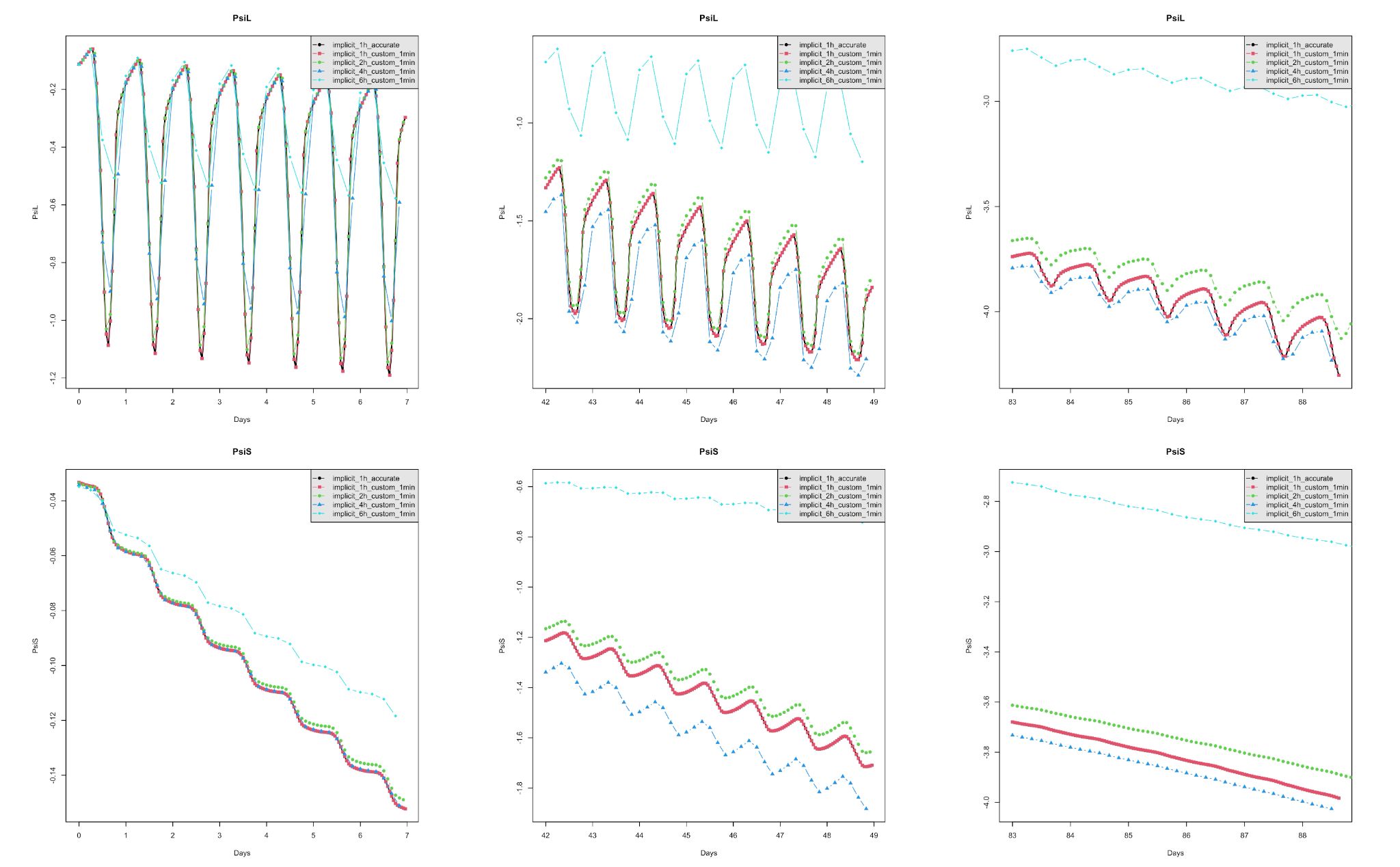
**Fig. 8.2.** Same as 8.1 for the Semi-Implicit scheme. The convergence requires 1 min. In particular, the survival time increases from 89 days for the reference to 96 days for the 1h resolution.

In order to optimize the code, an adaptive time step has been implemented. Indeed, some processes such as stomatal regulation require smaller time steps than others, but are not at play during the whole simulation. In the *“Normal”* mode (Default), the initial small time steps are at 10 minutes, but thoas are automatically and gradually refined up to 1 min in periods of intense regulation changes. This mode generally gives similar results as the “Accurate” mode, which has a fixed time step at 10 s, and which is used as reference in numerical stability studies, at a price of a computational cost up to 40 times higher. As the *“Normal”* mode, the *“Fast”* mode also relies on an adaptive time step, but starting computation on a 1h-base and refining it up to 10 minutes. It is on the order of 3 times faster than the normal mode and can typically be used for explanatory analyses, as minor differences may appear with respect to the reference. Finally, the *“Custom”* mode allows the user to set a -fixed- small time step in s (parameter smallTimeStepInSec, by default =600s). A value of 60 s, is typically on the order of 7 times slower than *“Normal”* mode. The performance of these different modes is shown in Fig. 8.3.



**Fig. 8.3.** Numerical stability of the different modes with the Implicit scheme (1h accurate is here for reference, the other line corresponding to predefined time step mode). The “Normal” mode is very accurate and the “Fast” mode is very good.

We recommend not to change the large time step much (keep 1h or eventually 2h), which is used for climate sampling (also called climate disaggregation). Indeed, the lack of discretisation of climate data quickly leads to large discrepancies in simulations, due to the nonlinearity at play in the models (especially the stomatal regulation). An example of sensitivity to the large time step (for a fixed small time step at 1 minute, in order to isolate the effect of the climate discretisation) is shown in Fig. 8.4. The lack of discretisation is especially visible in the early time steps, before leading to large discrepancies later in the simulation. Time step of 1h is recommended, especially because the impact of this in terms of computational cost is fairly marginal (on the order of 20%).



**Fig. 8.4.** Numerical stability of the large time step with the Implicit scheme (1h accurate is here for reference, the other line corresponding to different large time steps, the small time step being always 1min). From this result, a value of 1h is recommended.

REFERENCE

To be filled up.