Translation of SSM into R

Marie Gosme - 2020-02-21

Objectives:

- no dependency on Excel (R is open-source and cross-platform)
- possibility to run from a website
- possible to run in batch mode, and to run several simulations (i.e. several locations-climates) at the same time

Desired improvements of the code over the original SSM.xls code:

-use the same code for all crops (in excel version, the code is repeated for each crop, with subtle differences, which makes the code (i) difficult to read and understand, (ii) difficult to maintain (one has to make the same changes in several places in the code)=> crop-specific parts are explicitly identified
-add modularity (to help code maintenance and future developments).

Architecture of the model:

The model is "encapsulated" so that function definitions and model variables are not laying around in the user's workspace. The setup function creates an object (here, named MyModel, but it could be anything else), that contains an instance of the model and that can be manipulated by handler functions (defined in file handlers.R), e.g. functions to set the simulation options, to run the model for n time steps, to plot variables etc... Within the object, there are several "global" variables (named in uppercase), procedures (i.e. R functions that have no arguments and return no value, but access and modify global variables), and functions (which are called by procedures to compute different intermediary variables that are used by procedures). All intermediary variables and state variables, as well as input parameters (e.g. climate variables and crop parameters (which vary according to the crop currently present)), are saved at

each time step (the current time step is a data.frame named ALLDAYDATA, which is updated by each procedure, and at the end of the timestep, this data.frame is appended to a list (ALLSIMULATEDDATA) with one element per timestep.

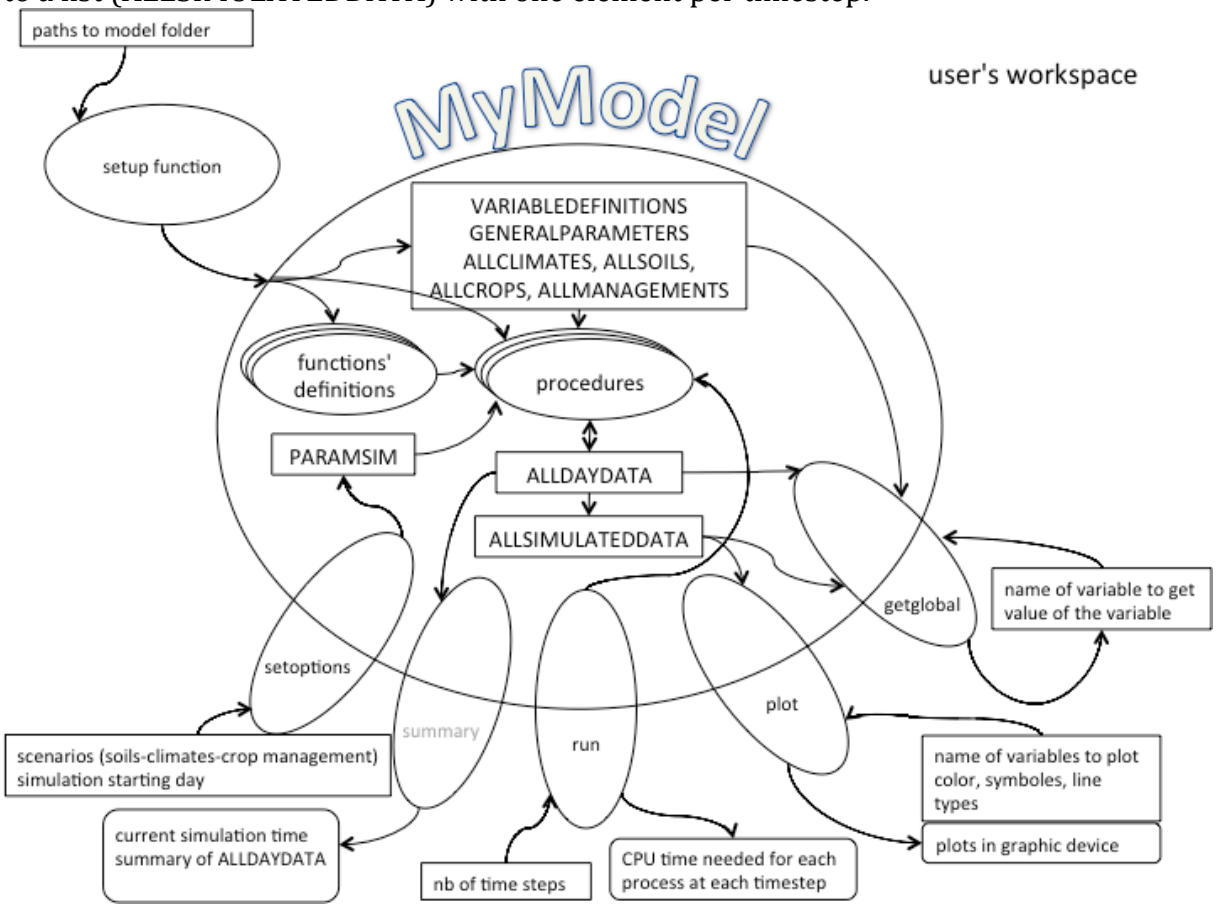

Figure 1: the setup function takes as argument the path to the model folder and returns an R object with its own environment, separated from the user's workspace. This object (here called "MyModel") can be manipulated by handler functions, for example MyModel$run(100) will run the model for 100 timesteps. Ovals: functions; squares: vectors, dataframes or lists; rounded squares: outputs to the console or graphic device

Naming conventions:

UPPERCASE: "global" (i.e. within the model) variables, accessible to all procedures
lowercase: local (i.e. within each function or procedure) variable
#variable names
#iXXXX: Input variable (e.g. iPr : Precipitation)
#pXXXX: Parameter (e.g. pPhyllochron)
#cXXXX: Computed variable (e.g. cDeltaLAI)
#sXXXX: State variable (e.g. sLAI)
#function names
#fXXXX: Function (e.g. fPhotosynthesis)
#rXXXX: pRocedure (e.g. rUpdateLAI)
#eXXXX: read input scenarios from External files (e.g. eReadInInputs)
#mXXXX: Model interface, i.e. handler functions (e.g. mRun, mPlotDynamics)

Model algorithm:

- At the beginning of the simulation, a first element (corresponding to day 0) is created in ALLSIMULATEDDATA, with state variables initialised at 0.

- At the beginning of each timestep, a new ALLDAYDATA data.frame is created, with initial values as specified in ALLVARIABLES (read from sheet "savedeachday" of the excel file allvariables.xlsx, which contains the definition, unit, original SSM name and initial value of all variables, usually NA for computed variables), then state variables (name starting with s) are initialised at their value from the previous time step. Then the procedure (name starting with r) corresponding to each module is called sequentially.
- In each procedure, local computed variables (name starting with c) are created using different functions (name starting with f), using as arguments either state variables values from the previous time step (i.e. from ALLSIMULATEDDATA[[length(ALLSIMULATEDDATA)]]), or computed variables from the current time step (i.e. from ALLDAYDATA). At the end of the procedure, ALLDAYDATA is updated with all state variables (which will be used to initialise their values for the next time step) as well as parameters and computed variables (to allow access to intermediary variables when analysis the model outputs).
- Each function corresponds to the computation of one intermediary variable in the SSM code, we tried to keep the code as similar to the original SSM code as possible...
- ... except that we use "**filters**" to know to which cases to apply the function or procedure: in the original SSM code, the conditions were hard-coded for each crop (e.g. for wheat:

```
bdBRV = bdEMR              'Beg. response to vernalization
bdTRV = bdSEL              'End of response to vernalization
If CBD >= bdBRV And CBD <= bdTRV .... do vernalisation calculations
```

- in SSM.R, these conditions are part of the crop parameterisation:

vernalisation.filter = "is.after('emergence', 0) & is.before('stemElongation', 0)" which means that vernalisation is applied only to wheat crops that are already emerged but haven't started the stem elongation stage. The syntax of is.before and is.after is the following: is.before(stage, bd=-Inf), which means that if bd is not specified, the stage itself is not included (is.before('tillering') means that it happens strictly before the beginning of tillering stage, while is.before('tillering', 5) means that it happens until the 5th biological day within the tillering stage. For is.after, the syntax is is.after(stage, bd=Inf), which means that if bd is not specified, the stage itself is not included (is.after('tillering') means that it happens strictly after the end of tillering stage, while is. after ('tillering', 5) means that it happens starting on the 5th biological day within the tillering stage. Note that the stage names are not hard-coded: they come from the names given to the thresholds for each stage provided in the crop parameter file.

Structure of the global variables

PARAMSIM

```
List of 8
 $ simustart    : Date[1:1], format: "1997-11-01"
 $ cases        :'data.frame':     2 obs. of  4 variables:
  ..$ climatename: chr [1:2] "Ain Hamra - Meknes" "Ain Hamra - Meknes"
  ..$ soilname   : chr [1:2] "325_-35" "325_-35"
  ..$ lat        : num [1:2] 35 45
  ..$ long       : num [1:2] -5 -5
 $ directory    : chr "/Users/user/Documents/b_maison/congeMat/D4DECLIC/runSSM"
```

$ climateformat: chr "standardSSM"
$ cropformat : chr "standardSSM"
$ soilformat : chr "standardSSM"
$ managformat : chr "standardSSM"
$ Neffect : logi TRUE

simustart is the day when the simulation should start (used to find the climate data for this day) ; cases is a data.frame with n rows (=n cases= n combinations of climate, soil, geographic coordinates (and in the future crop management) ; directory is the path to the folder where your input folder, containing your input (soil, climate, crop parameter, crop management) files are placed (and where the output folder will be created to store graphs or export files) ; climateformat, cropformat, soilformat and managformat are the formats in which these input files are (for now only "standardSSM" is available, it was planned to include other formats to allow automatic link with external climate and soil databases, but not done yet) ; Neffect is a boolean indicating which version of the model to use: with or without N effect on LAI decrease (for now, only the version with N effect is coded).

ALLDAYDATA

'data.frame': 2 obs. of 111 variables:
$ iDate : Date, format: "1997-11-04" "1997-11-04"
$ sAccumulatedLeafDryMatter : num 0 0
$ sAccumulatedStemDryMatter : num 0 0
$ sAccumulatedVegDryMatter : num 0 0
$ sLAI : logi NA NA
$ sMainstemNodeNumber : num 1.72 1.72
$ sPlantdensity : num 280 280
$ sCrop : chr "WHEAT" "WHEAT"
$ sCultivar : chr "durum wheat" "toto"
$ sBiologicalDay : num 2.88 2.88
$ sGrowthStage : chr "germination" "germination"
$ sThermalUnite : num 79.2 79.2
$ sRootFrontDepth : num 0 0
etc... (all variables listed in sheet "savedeachday" of file allvariables.xlsx)
rows are cases (=combinations of climate-soil-management)

ALLSIMULATEDDATA

list of n data.frames (ALLDAYDATA at the end of the time step, one for each timestep (plus day 0)).

VARIABLEDEFINITIONS

this data.frame is read from sheet "savedeachday" in file "allvariables.xlsx" from the model folder.
'data.frame': 111 obs. of 9 variables:
$ name : chr "sAccumulatedLeafDryMatter" "sAccumulatedStemDryMatter" "sAccumulatedVegDryMatter" "sLAI" ...
$ typeinthemodel : chr "stateVariable" "stateVariable" "stateVariable" "stateVariable" ...

$ typeR : chr "numeric" "numeric" "numeric" "numeric" ...
$ module : chr "DMDistribution" "DMDistribution" "DMDistribution" "LAI" ...
$ unit : chr "g m-2 " "g m-2" "g m-2" "m2 m-2" ...
$ definition : chr "Accumulated leaf dry matter " "Accumulated Steam dry matter " "Accumulated vegetative (leaf + stem) dry matter " "Leaf area index " ...
$ translationSSM : chr "WLF" "WSF" "WVEG" "LAI" ...
$ french : chr "Stock de matière sèche dans les feuilles" "Stock de matière sèche dans les parties végétatives" "Stock de matière sèche dans les tiges" "Index de surface foliaire" ...
$ defaultInitialvalue: num 0 0 0 0 0 0 0 0 0 0 ...

ALLCROPS

list (one element per crop.cultivar) of crop parameters, organised by modules. This list is created by reading all sheets of file crops.xlsx in the input folder of the simulation folder (NOT the model folder, which contains example input files, which are not read).
List of 3
$ WHEAT.durum wheat:List of 14
..$ name : chr "WHEAT.durum wheat"
..$ thresholds :List of 9
.. ..$ germination : num 6
.. ..$ emergence : num 5
.. ..$ tillering : num 8
.. ..$ stemElongation: num 6
.. ..$ Booting : num 6
.. ..$ earing : num 15
.. ..$ anthesis : num 43
.. ..$ maturation : num 8
.. ..$ senescence : num Inf
..$ waterstress :List of 1
.. ..$ filter: chr "is.after('emergence', 0) & is.before('senescence', 10)"
..$ DMDistribution:List of 12
.. ..$ filter : chr NA
.. ..$ pFractionCropMassTranslocateble : num 0.22
.. ..$ pGrainConversionCoefficient : num 1
.. ..$ pParCoefLeafAeraHigh : num 0.3
.. ..$ pParCoefLeafAeraLow : num 0.6
.. ..$ pParCoefLeafAeraTerminationLeafGrow : num 0.1
.. ..$ pPotentialSlopeHarvestIndex : num 0.008
.. ..$ pPotentialSlopeHarvestIndexCriticalPoint1: num 0
.. ..$ pPotentialSlopeHarvestIndexCriticalPoint2: num 600
.. ..$ pPotentialSlopeHarvestIndexCriticalPoint3: num 1200
.. ..$ pPotentialSlopeHarvestIndexCriticalPoint4: num 3200
.. ..$ pTotalCropMassFromCoefLeafAreaLowHigh : num 160
..$ DMProduction :List of 7
.. ..$ filter : chr "is.after('germination') & is.before('anthesis', 41,5)"
.. ..$ pKPAR : num 0.65
.. ..$ plethalRUE : num 35
.. ..$ pRadEffiencyOptimal: num 2.2
.. ..$ pTbasRUE : num 0

```
.. ..$ pTopt1RUE         : num 15
.. ..$ pTopt2RUE         : num 22
..$ LAI_Mainstem  :List of 5
.. ..$ filter               : chr "is.after('germination', 0) & is.before('Booting', 0)"
.. ..$ pcoefPlantLeafNumberNode: num 1
.. ..$ pExpPlantLeafNumberNode : num 2.5
.. ..$ pPhyllochron          : num 118
.. ..$ pSpecificLeafArea      : num 0.02
..$ LAI_Senescence:List of 7
.. ..$ filter                : chr NA
.. ..$ pFreezeFracLeafDestruction: num NA
.. ..$ pFreezeThresholdTemp     : num NA
.. ..$ pHeatFracLeafDestruction  : num 0.1
.. ..$ pHeatThresholdTemp       : num 30
.. ..$ pSpecLeafNGreenLeaf      : num 1.8
.. ..$ pSpecLeafNSenescenceLeaf  : num 0.4
..$ phenology     :List of 5
.. ..$ filter     : chr NA
.. ..$ pTbasedev  : num 0
.. ..$ pTlethaldev: num 40
.. ..$ pTopt1dev  : num 27.5
.. ..$ pTopt2dev  : num 27.5
..$ photoperiod   :List of 3
.. ..$ filter              : chr "is.after('emergence', 0) & is.before('stemElongation', 0)"
.. ..$ pCriticalPhotoPerdiod  : num 14
.. ..$ pPhotoPeriodSensitivity: num 0.17
..$ rRootDepth    :List of 3
.. ..$ filter              : chr "is.after('emergence',0) & is.before('anthesis',5)"
.. ..$ pMaxDepthWaterExtraction: num 1000
.. ..$ pPotentialRootGrowth    : num 30
..$ rWaterBudget  :List of 2
.. ..$ filter                      : chr NA
.. ..$ pTranspirationEfficiencyLinkedToCO2: num 5.8
..$ vernalisation :List of 7
.. ..$ filter               : chr "is.after('emergence', 0) & is.before('stemElongation', 0)"
.. ..$ pTbaseVernalization     : num -1
.. ..$ pTlethalVernalization    : num 12
.. ..$ pTopt1Vernalization     : num 0
.. ..$ pTopt2Vernalization     : num 8
.. ..$ pVDSAT                : num 50
.. ..$ pVernalizationSensitivity: num 0.002
..$ LAI_Secondary :List of 1
.. ..$ filter: chr "is.after('Booting', 0) & is.before('earing', 5)"
..$ DM_SeedGrowing:List of 1
.. ..$ filter: chr NA
 $ WHEAT.toto      :List of 14
idem
$ Chickpea.Ghab2   :List of 14
idem
```

ALLCLIMATES

data.frame with one row for each date in each climate. It is created by reading all sheets of file climates.xlsx in the input folder of the simulation folder (NOT the model folder, which contains example input files, which are not read).

'data.frame': 15336 obs. of 8 variables:
$ YEAR : num 1995 1995 1995 1995 1995 ...
$ DOY : num 1 2 3 4 5 6 7 8 9 10 ...
$ iRSDS : num 11.3 9.1 11.7 13.1 7.3 12 11.4 14 13.9 13.5 ...
$ iTASMax : num 21.2 12.8 15 12.8 13.5 15.4 12.8 12.8 14.2 13.5 ...
$ iTASMin : num 7.8 8.2 4.2 5.5 4 7.8 6.4 0.1 -2.3 2.7 ...
$ iPr : num 0 2 0 0 0 0 0 0 0 0 ...
$ date : Date, format: "1995-01-01" "1995-01-02" "1995-01-03" ...
$ climatename: chr "Ain Hamra - Meknes" "Ain Hamra - Meknes" "Ain Hamra - Meknes" "Ain Hamra - Meknes" ...

ALLSOILS

not done yet.
the watermodule is being coded, for now it uses a list named PARAMSSOILS, but it might change in the future
List of 7
$ pNLayer : num [1:3] 2 2 3
$ pDrainLayer : num [1:3] 2 2 3
$ pSoilAlbedo : num [1:3] 0.12 0.13 0.15
$ U : logi NA
$ pSoilCurveNumber: num [1:3] 60 70 80
$ pVPDcoef : num [1:3] 0.65 0.65 0.65
$ paramlayers :List of 3
..$:'data.frame': 2 obs. of 6 variables:
.. ..$ layer : int [1:2] 1 2
.. ..$ pLayerThickness: num [1:2] 300 700
.. ..$ pSaturation : num [1:2] 0.36 0.4
.. ..$ pFieldCapacity : num [1:2] 0.24 0.25
.. ..$ pWiltingPoint : num [1:2] 0.1 0.12
.. ..$ pSoilDryness : num [1:2] 0.03 0.04
..$:'data.frame': 2 obs. of 6 variables:
.. ..$ layer : int [1:2] 1 2
.. ..$ pLayerThickness: num [1:2] 200 800
.. ..$ pSaturation : num [1:2] 0.36 0.4
.. ..$ pFieldCapacity : num [1:2] 0.24 0.25
.. ..$ pWiltingPoint : num [1:2] 0.1 0.14
.. ..$ pSoilDryness : num [1:2] 0.04 0.04
..$:'data.frame': 3 obs. of 6 variables:
.. ..$ layer : int [1:3] 1 2 3
.. ..$ pLayerThickness: num [1:3] 300 200 400
.. ..$ pSaturation : num [1:3] 0.36 0.4 0.38
.. ..$ pFieldCapacity : num [1:3] 0.24 0.25 0.22
.. ..$ pWiltingPoint : num [1:3] 0.1 0.12 0.09
.. ..$ pSoilDryness : num [1:3] 0.03 0.04 0.035

ALLMANAGEMENTS

It is created by reading soils.xlsx in the input folder of the simulation folder (NOT the model folder, which contains example input files, which are not read).
List of possible crop managements plans (named "row XX" based on the row number of <--- MangRowNo in the excel file).
List of 3
 $ row 13:List of 11
  ..$ dfCode   :'data.frame':   1 obs. of 2 variables:
  .. ..$ Code   : chr "ROTATION_BLE"
  .. ..$ Description:: num 0
  ..$ dfSowing   :'data.frame':   1 obs. of 11 variables:
  .. ..$ FixFind: num 4
  .. ..$ Fyear : num 2007
  .. ..$ yrno   : num 1
  .. ..$ SimDoy : num 20
  .. ..$ Fpdoy : num 288
  .. ..$ Lpdoy : num 20
  .. ..$ StopDAP: num 280
  .. ..$ SowTmp : chr "-"
  .. ..$ SowWat : num 0.3
  .. ..$ Pden   : num 280
  .. ..$ STBLW : num 0.01
  ..$ nitrogenScenario: num 2
  ..$ nitrogenNumber : num 1
  ..$ nitrogenDatetype: num 1
  ..$ nitrogendf   :'data.frame':   1 obs. of 4 variables:
  .. ..$ NapplNumber: num 1
  .. ..$ DAPorCBD   : num 1
  .. ..$ amount   : num 5
  .. ..$ FracVol   : num 7
  ..$ waterLevel   : num 0
  ..$ waterScenario   : num 2
  ..$ waterNumber   : num 0
  ..$ waterDatetype   : num 0
  ..$ waterdf   :'data.frame':   0 obs. of 2 variables:
  .. ..$ DAPorCBDorDOY: logi(0)
  .. ..$ amount   : logi(0)
 $ row 36:List of 11
idem
 $ row 59:List of 11
idem

GENERALPARAMETERS

this data.frame is read from sheet " generalPhysicalParameters" in file "allvariables.xlsx" from the model folder.
'data.frame': 5 obs. of 9 variables:
 $ name   : chr "pMinimalSoilEvaporation" "pSoilWettingWaterQuantity" "pCanopyExtinctionCoefficient" "pCropAlbedo" ...
 $ typeinthemodel   : chr "parameter" "parameter" "parameter" "parameter" ...

```
 $ typeR           : chr  "numeric" "numeric" "numeric" "numeric" ...
 $ module          : chr  "rWaterBudget" "rWaterBudget" "rWaterBudget" "rWaterBudget"
...
 $ unit            : chr  "mm" "mm" "-" "-" ...
 $ definition      : chr  "Minimal Soil Evaporation" "amount of rainfall and/or irrigation
required to wet the top soil layer to return soil evaporation from Stage II to Stage I"
"canopy extinction coefficient" "Crop albedo" ...
 $ translationSSM  : chr  "EOSMIN" "WETWAT" "KET" "CALB" ...
 $ french          : chr  "evaporation minimale du sol" NA NA NA ...
 $ defaultInitialvalue: num  1.5 10 0.5 0.23 0.48
```

Structure of the files in the model folder

- allvariables.xlsx : excel file with the name, module where it is used, definition, unit, original SSM name and initial value of all variables
- exampleRunSSM.R : R script giving an example of how to run the model (and the definition of the setup function : this is the script that the user will run in the console
- externalFilesReadingSSM.R : function definitions for reading in external files (does not actually read anything, the reading is done when the handler function mRun() is run for the first time)
- functionsSSM.R : definition of all functions and procedures that are at the core of the SSM model (to be done: split this file by module to facilitate maintenance)
- handlersSSM.R : definition of handler functions (mRun, mPlotDynamics...)
- headersSSM.R : loading of required packages
- HousekeepingFunctionsSSM.R : definitions of functions that are not specific of SSM but are usefull for the model algorithm.
- initialisationEnvironmenVariablesSSM.R : creation of all the encapsulated "global" variables.
- input : folder with example input files
    - climates.xlsx : excel file with one sheet per possible climate (same content as in SSM.xlsx)
    - crops.xlsx : excel file with one sheet per species, one column per cultivar (same content as in SSM.xlsx except extra parameters at the bottom: thresholds for the different stages (obtained with a formula using the bdXXX parameters, warning: for the formula to give R-readable values, the decimal separator in excel must be the point, not coma as is the default in French language setting) and filters (text in R language that will be parsed by the model) for the different modules... see "filters" bullet point in the "Model algorithm" section.
    - managementPlans.xlsx not used yet (but read anyway): excel file with one sheet per crop Management plan (rule for sowing and sowing density, nitrogen and water applications): (same content as in SSM.xlsx).
    - soils.xlsx excel file with one sheet per possible soil (same content as in SSM.xlsx)
- LICENSE: GNU licence, I don't know why it's here
- progR_essaiSSMfiltres.R : to be removed, it's a remnant of old stuff
- README.md : file used by github to display on https://github.com/mariegosme/SSM.R

- SSMWaterModule.R : temporary file with the water module partially coded... will eventually be merged into functionsSSM.R (or kept as separate file if modules are separated into different files).

Function setup() sources, locally within itself, the files headersSSM.R, initialisationEnvironmenVariablesSSM.R, externalFilesReadingSSM.R, HousekeepingFunctionsSSM.R, functionsSSM.R, handlersSSM.R in this order.

How to run the model

1) run the function definition of function setup

2) build the model

```
mymodel<-setup("/Users/user/Documents/b_maison/congeMat/D4DECLIC/SSM/")
```

3) prepare cases (these will be the rows of ALLSIMULATEDDATA):

```
mycases<-data.frame(climatename="Ain Hamra - Meknes", soilname="325_-35",
lat=c(35, 45), long=-5)
#climatename: one of the sheet names of file climates (if climate in standard SSM
format)
#soilname: one of the sheet names of file soils (if in standard SSM format)
#to do: define crop rotation and management (once management procedure is
completed)
rownames(mycases)<-c("Meknes35degres", "Meknes45degres") #these will be the
cases names used in the plots, outputs etc...
```

4) define the options of the simulation

```
paramsim<-list(
 simustart=as.Date("1997-11-01"), #date of start of the simulation
 cases=mycases, #cases (e.g. combinations of soil-climate-management-location)
 directory="/Users/user/Documents/b_maison/congeMat/D4DECLIC/runSSM",#folder
where your input (with climates and soils files) and output folders are
 climateformat="standardSSM",
 cropformat="standardSSM",
 soilformat="standardSSM",
 managformat="standardSSM",
 Neffect=TRUE
)
```

5)use this to set the simulation options

```
mymodel$setoptions(paramsim)
```

6)run the model for 4 timesteps

```
mymodel$run(4)
```

7)plot the dynamics of some variables

```
dynamiques<-mymodel$plot(c("iTASMin", "iTASMax", "iRSDS"),
        colors=c(iTASMin="blue", iTASMax="red", iRSDS="black"),
whatcolors="variables",
        linetypes=c(iTASMin=1, iTASMax=1, iRSDS=2), whatlinetypes="variables",
       symbols=c(Meknes35degres=1, Meknes45degres=8), whatsymbols="cases")
```