

STICS intercrop: a work in progress for the ReMIX H2020 project

R. Vézy

2020-03-31

Contents

1 Prerequisites	5
2 Light Interception	7
2.1 Introduction	7
2.2 Plant shape	8
2.3 Light interception	11
2.4 Summary	24
2.5 Discussion and proposed modifications	24
3 Light incident to the dominated crop (trg)	25
3.1 Computing the trg incident to the dominated plant	25
3.2 Proposed solution	25
3.3 Results	26
4 Wheat EAI parameterization	29
4.1 Introduction	29
4.2 Model sensitivity to the P_sea parameter	29
4.3 Parameterization of the P_sea parameter	33
5 Plant density and equivalent plant density	35
5.1 Introduction	35
5.2 The density effect on LAI	35
5.3 The equivalent density	36
5.4 Proposition for new equivalent density formalism	37
6 Interrow spacing	41
6.1 Introduction	41
6.2 Inter-row spacing for mixed crops in the model	42
7 Design	45
7.1 Introduction	45
7.2 Designs that can be simulated	45
7.3 Designs that cannot be simulated	45

8 Issue on LAI not decreasing	47
8.1 Introduction	47
8.2 Investigating the sla hypothesis	48
8.3 Leaf senescence	48
8.4 The solution to the first issue	52
8.5 Fixing a second issue	53
9 Issue on FaPAR greater than one	57
9.1 Introduction	57
9.2 The solution to the problem	58
10 Computing plant height	59
10.1 Introduction	59
10.2 Link between height and lai and dry mass	62
10.3 Computing plant height	62
11 Developpment stages and LAI dynamic	67
11.1 Introduction	67
11.2 Checking the ilev	67
11.3 Checking laicomp	69
11.4 Comparing simulated and observed lai and masec in self-IC	69
11.5 Checking durvieI	70
12 The LAI in intercropping is too high	73
12.1 Introduction	73
12.2 Investigating the ulai hypothesis	74
12.3 Investigating the density hypothesis	75
13 Miscellaneous	79
13.1 Introduction	79
13.2 Extra code from transrad	79
13.3 Sowing density in Auzeville	79
13.4 Energy balance	79
13.5 Positive raint with zero LAI+EAII	80
13.6 Qfix decrease	80
13.7 eddensite_rac	81
14 First summary	83
14.1 Introduction	83
14.2 Model outputs	83
14.3 Model parameterization	83
15 Interception fix	87
15.1 Introduction	87
15.2 Model parameterization	88
16 Light competition effect	89

Chapter 1

Prerequisites

Each chapter of the book match a specific objective. I first introduce the subject with a brief description, try to find some solutions for the specific issues and show the results.

The code is only visible on the `html` version of the book, so please refer to this format if you need any further information (open the `index.html` file).

This book is written using the R **bookdown** package, which can be installed from CRAN or Github:

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.name/tinytex/>.

Chapter 2

Light Interception

2.1 Introduction

In STICS, the light interception is either computed using a simple Beer law assuming a homogeneous, turbid canopy, or a radiation transfer model that consider the plants leaf area index, canopy shape, height and density. The Beer law option is very simple but generally fairly accurate for high density homogeneous crops, but may rapidly yield unsatisfying predictions for stands with higher structural complexity (e.g. perennial plantations or mixed crops). For mixed crops, the model uses the radiation transfer model in a particular manner that is described in further details below.

For the case of intercrops, the model first set the taller crop as “dominant”, and the shorter crop as “dominated”. The model then compute the radiation interception of each one according to the dominancy, the structure (height, width, light extinction coefficient...) and position (interrow distance, row orientation) of the species. Of course, the dominancy of each plant species can be inverted if the dominated plant become taller than the dominant plant, and the model checks every day for such a case.

```
j=1          # julian day (DOY= Day Of Year)
ir= 1        # Interrow distance
l= 0.2       # Width of the plant crown
rap= 0.5     # Thickness to width ratio
haut= 1.      # Height of the part of the dominant plant above the dominated plant
              # = hauteurzero = P_hautbase-originehaut
              # with originehaut= dominated plant height
P_forme= 1   # Plant shape code (1= rectangle, 2= upside triangle, 3= downside tr.)
alpha= 0      # Row orientation
lat= 43.61    # Latitude in degrees
trg= 25       # Global radiation
```

```

rdif= 0.4      # Diffuse fraction of the atmospheric radiation
P_ktrou= 0.8   # Light extinction coefficient of the dominant plant
lai= 4          # Leaf area index of the dominant plant
eai= 0          # Equivalent leaf area index, = photosynthetic surface that is not fr
# leaves (e.g. wheat ears, rapeseed siliquae, pea pods or grapes during their green st

```

To facilitate understanding, we will use a common intercrop example throughout the whole document. We take a mixed crop simulated for the day of the year 1 with a global radiation of 25 MJ m⁻² day⁻¹ (= 12 MJ m⁻² day⁻¹ of PAR), and a diffuse fraction of light of 0.4, an interrow spacing of 1 meter, a shape of 1 (1= rectangle, 2= upside triangle and 3= downside rectangle), a canopy width of 0.2 meter, a canopy thickness of 0.1 meter, a row orientation of 0 relative to the North-South axis at latitude 43.61 degrees north, an LAI of 4, a light extinction coefficient of 0.8. The canopy of the dominant plant is 1 meter above the dominated plant. All code used in this document can be viewed in each section by clicking on the **Code** button on the right.

2.2 Plant shape

Each plant radiation interception is computed using an approximation of its shape (`P_forme`= 1, rectangle, = 2 upside triangle, = 3 downside triangle), leaf area index (`lai` or `p(i)%lai(ens,n)`), height, width, density at emergence and leaf area density (`dfol`).

2.2.1 Plant width computation

First, the plant width is computed using the `formplante` function. This computation is possible thanks to the relationship between two ways for computing the plant leaf area, where one of them uses the plant width:

- First, assuming the plant has a square footprint (i.e. plant projection is square), we can find the plant leaf area using the leaf area index (`lai`) and the plant density:

$$LA = \frac{(lai + laisen + eai)}{densite} \text{ which is the equivalent to: } LA = (lai + laisen + eai) \cdot interrang \cdot distrang$$

where `lai` is the leaf area index, `laisen` is the senescent leaf area index, `eai` is the equivalent leaf area of the photosynthetic organs that are not leaves (e.g. flower buds), `interrang` is the inter-row spacing and `distrang` the intra-row distance.
- Second, using the leaf area density and the plant volume such as:

$$LA = dfol \cdot largeur \cdot epaisseur \cdot profondeur$$
 for a cuboid and $LA = dfol \cdot \frac{1}{2} \cdot (largeur \cdot epaisseur \cdot profondeur)$ for a triangular prism, where `dfol` is the leaf area density, `largeur` is the **plant width** (also called `largtrans`), `epaisseur` is the plant thickness and `profondeur` is the plant depth.

Knowing both equations, and assuming the two following hypothesis:

1. The plant depth is equal to the plant width
2. The intra-row distance between two plants is equal to the plant width

We can now compute the plant width as:

$$\text{largeur} = \sqrt{\frac{(lai+laisen+eai) \cdot \text{interrang}}{dfol \cdot \text{varrapforme}}}$$

where **varrapforme** (or **raptrans** further) is the ratio between the plant thickness and width and is computed as $\text{varrapforme} = \frac{(\text{hauteur} - P_{\text{hautbase}})}{\text{largeur}}$ where **hauteur** is the total plant height, **P_hautbase** is the plant (i.e. crown) base height and **largeur** is the plant width (see fig 2.1 for more details)

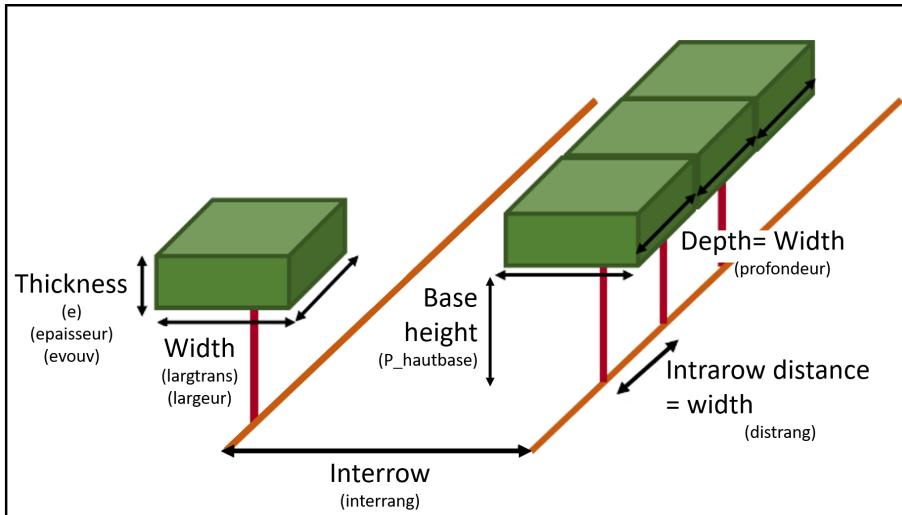


Figure 2.1: Diagram representing the different parameters used to compute plant width. The different names used in the model are shown between parenthesis

2.2.2 Plant width correction

If the dominated plant is higher than the base height of the dominant plant, the radiation interception of the dominated plant is partially reduced, by reducing the volume that can intercept light to the canopy volume above the dominated plant only. This correction is made to consider the competition for light between the two species, and is computed according to the shape of the plant.

Consequently, the model first compute the height of the dominated plant (**sc%originehaut**) by looking for the maximum height between the sunlit and shaded part of the dominated plant:

$$\text{sc%originehaut} = \max(p(i+1)\%\text{hauteur}(\text{sc\%AO}), p(i+1)\%\text{hauteur}(\text{sc\%AS}))$$

Table 2.1: Variable name modification in the formplante function

Original	Modified	Definition
hauteur	hauteur	Height
largeur	largtrans	Width
varrapforme	raptrans	Thickness/Width Ratio
enouv	enouv	Thickness

`sc%originehaut` is fixed to 0 (= soil) while computing the dominated plant.

Hence, the new thickness (`enouv`) is computed as: $enouv = largeur \cdot |varrapforme| + hauteurzero$, and is used to re-compute the shape of the plant:

- The new thickness to width ratio (`raptrans`, formerly `varrapforme`): $raptrans = \frac{enouv}{largeur}$ for rectangle shaped plants,
- The new width (`largtrans`, formerly `largeur`): $largtrans = \frac{enouv}{varrapforme}$ for upsided triangle shaped plants,
- The new width (`largtrans`, formerly `largeur`): $raptrans = \frac{enouv}{largtrans}$ for downsidized triangle shaped plants,

All variable names are changed, whether there is a correction or not. Here is a summary table:

Upside triangle is a triangle with its base at the bottom, while downside triangle is a triangle with the base at the top.

The correction of the shape of the plant can be summarised in the diagram Presented in Figure 2.2.

The correction of the shape of the plant is only used to compute the light transmitted to a plane at the dominated plant or soil height. The targeted plant interception is computed using its whole leaf area index and a light extinction coefficient.

2.2.3 Plant height

The plant height is simply computed using the plant base height, its width, and the thickness to width ratio:

$$hauteur = P_{hautbase} + largeur \cdot |varrapforme|$$

So indirectly, the plant height depends on its LAI, because the plant width is computed using it in the first place. This formalism maybe not optimal, because the plant `height` and `LAI` are not well related during advanced plant physiological stages.

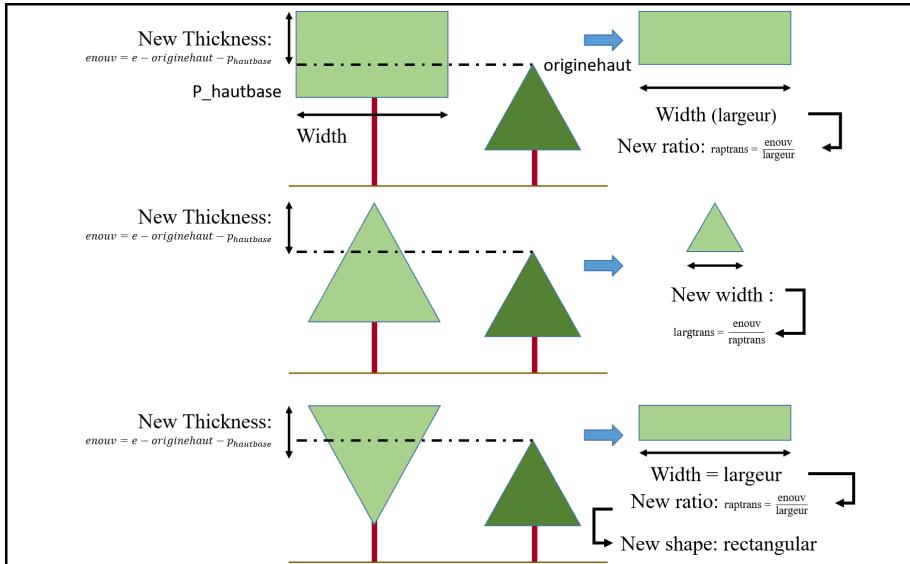


Figure 2.2: Competition for radiation interception of the dominant plant induced by a high dominated plant

2.3 Light interception

The light intercepted by a plant species is obtained by computing the light reaching a horizontal plane below its canopy, either at the height of the dominated plant (for the dominant plant) or the soil (for the dominated plant). Therefore, the light incident on this plane is either coming from:

- The incident light coming from the atmosphere, divided into two components, namely the diffuse and direct light. This light is called **rdroit** in the model,
- The light transmitted by the dominant crop, which is called **rtrans**, and that is generally of lower quality for photosynthesis. The effect of light quality is wrapped in the equivalent density formalism, see Chapter 5 for more details.

Consequently, numerous points (20, or 200 if the inter-row is lower than 1 m) are equally distributed every meter along the inter-row (*i.e.* one point every 5 cm, or every 0.5 cm with 200 points), at the height of the plane. These points are used to discretize the computation of the incident light at the surface of the plane.

Hence, the total number of points to simulate is computed using the **interval** parameter, which is equal to 200 if the inter-row is lower than 1 meter or 20 if more. It is then used to compute the total number of points to simulate as:

$$N_{points} = \frac{ir}{2} \cdot interval$$

In practice, the model really simulates only half of the inter-row, because it is considered that the other half have the same light conditions at daily time-scale. For example, if we take an interrow of 10 meters, the model simulates 100 points equally distributed from 0 to 5 meters.

Here is an example of the X position on the plane, starting from the left-hand side of the row, using an inter-row spacing of 1 meter and a plant width of 0.2 meter:

```
if(ir<1.0){
  interval = 200
}else{
  interval = 20.
}
i= 1:(ir / 2 * interval)
x= (i-1) / interval
cat(paste("x=", paste(x, collapse = " ", )))
```

x= 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45

The points are then divided into two groups :

- The sunlit points, which are located directly under the crown of the targeted crop.
- The shaded points, which are not directly under the crop crown, *i.e.* they have the sky above them.

Then the semi-hemisphere above each point is discretized onto 2 x 23 angles: 23 angles from top to right, and 23 angles from top to left. These angles are used to compute **kgdiffus**, the atmospheric diffuse radiation incident to the X point. However, if the X point is below the targeted crop canopy ($X < 1/2$), only the 23 angles from the top to the right are used to compute **kgdiffus** (considering that Xs are only computed from the left-hand plants row until the middle of the inter-row, see fig.2.3 for more details).

The model determines the two angles (θ_1 and θ_2) between which the point only receive incident light coming from the atmosphere (**rdroit**). Using these two angles (or their tangent, G), the model computes:

1. The daily direct radiation (*i.e.* cumulated hourly radiation) that is incoming only during the time period between two hours (**h1** and **h2**) when the sun angle is between θ_1 and θ_2 . Function **kgeom** called in **rtrans**.
2. The incident diffuse radiation for all angles between θ_1 and θ_2 . Function **kdiff** called in **rtrans**.
3. The light transmitted to the plane by the target crop for all angles below θ_1 or above θ_2 .

In practice, all points with an x position lower than $\frac{largeur}{2}$ are shaded, and all other are sunlit, so the model computes the diffuse

radiation coming from the atmosphere only for one quarter of the hemisphere for shaded Xs ; the other quarter will receive transmitted light because all angles are superior to θ_2 .

Main functions used are `transrad`, `rtrans`, `kdiff` and `kgeom`.

The position of θ_1 and θ_2 (and their tangent) depends from three components: the crop shape, its inter-row spacing, and the sun azimuth (see fig. 2.3).

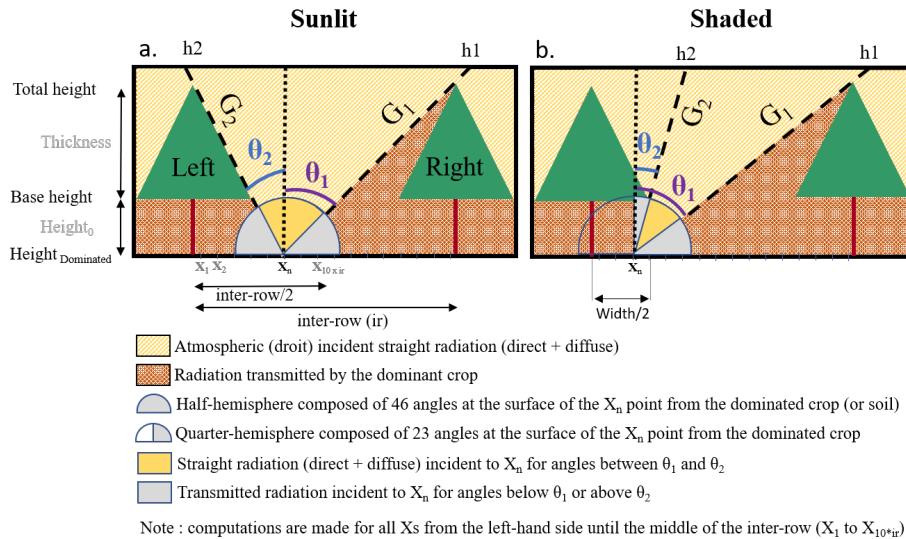


Figure 2.3: Diagram of the computation workflow of STICS for radiation interception for two X points placed above the dominated plant species. a. The X point is considered sunlit; b. The X point is considered shaded (right under the dominant plant canopy), so only the right-hand side of the semi-hemisphere is computed for atmospheric radiation

2.3.1 Incident direct radiation from the atmosphere

The incident direct radiation for each X point is computed for all angles of the hemisphere between θ_1 and θ_2 (see 2.3), and summed up to integrate the semi-hemisphere of each point.

Here is an example of the computation of the direct proportion received at each X point for a crop using the `kgeom()` function:

```
decangle= function(j){  
  theta1 = 2 * pi * (j - 80) / 365  
  theta2 = 0.034 * (sin(2 * pi * j / 365) - sin(2 * pi * 80 / 365))  
  theta = theta1 - theta2
```

```

    decangle = asin(0.3978 * sin(theta))
}

thetacrit= function(lat,j,tgh,alpha){

  acrit = 0.0
  bcrit = 0.0
  a = 0.0
  b = 0.0
  thetacr = 0.0
  hcritprec= 0
  n = 3
  dec = decangle(j)
  hprec = 0.0
  pi = 4 * atan(1.0)
  theta= rep(0,180)

  for(i in 1:(18 * n)){
    theta[i] = 10. / n * (i - 1)
    theta[i] = (theta[i] - 90) / 180 * pi
    # Sun position (h,azim)
    sinh = sin(lat) * sin(dec) + cos(lat) * cos(dec) * cos(theta[i])
    h = asin(sinh)
    cosazim = (-cos(lat) * sin(dec) + sin(lat) * cos(dec) * cos(theta[i])) / cos(h)
    cosazim = pmin(1.0,cosazim)
    if(theta[i] !=0.0){
      azim = acos(cosazim) * theta[i] / abs(theta[i])
    }else{
      azim = 0.0
    }
    if(sinh<0.0){h = 0.0}

    # Critical height
    hcrit = atan(tgh * abs(sin(azim + alpha + 0.00001)))
    # test for h = hcrit
    if (hcritprec >= hprec & hcrit <= h & i > 1){
      # Linear interpolation:
      acrit = (hcrit - hcritprec) / (theta[i] - theta[i-1])
      bcrit = hcrit - acrit * theta[i]
      a = (h - hprec) / (theta[i] - theta[i-1])
      b = h - a * theta[i]

      if(a!=acrit){thetacr = (b - bcrit) / (acrit - a)}
      hcritprec = hcrit
      hprec = h
    }
  }
}

```

```

        }
    }
    return(thetacr)
}

e = abs(l * rap) # Thickness of the plant crown
xprec= 0           # Init.
limite = l / 2

if(ir<1.0){interval = 200.}else{interval = 20.}
if(l>ir/2.){l = ir/2.}

if(e>0.0{
    limite2 = l/2 * (haut/e+1)
}else{
    P_forme = 1
}
# NB : is it sure that it is (haut/e+1) and not (haut/(e+1)) ?

# Saving some intermediate results for output here (not in the real function):
x_s= (1:(ir / 2 * interval))
Output= data.frame(x_index= x_s,
                    x_value= (x_s-1)/interval,
                    theta1= rep(NA_real_,length(x_s)),
                    theta2= rep(NA_real_,length(x_s)),
                    theta1_deg= rep(NA_real_,length(x_s)),
                    theta2_deg= rep(NA_real_,length(x_s)),
                    kg= rep(0,length(x_s)))

# Loop over Xs:

for(i in x_s){
    x = (i-1) / interval
    if(xprec<=l/2&x>l/2){ilim = i}
    xprec = x

    # kdir:

    # Rectangle:
    if(P_forme==1){
        tgh1 = (haut + e) / (ir - x - limite)
        theta1 = thetacrit(lat,j,tgh1,alpha)
        if(x>limite){
            tgh2 = (haut + e) / (x - limite)
            theta2 = thetacrit(lat,j,tgh2,alpha)
    }
}
```

```

}else if(x < limite){
    tgh2 = haut / ( -x + limite)
    theta2 = -thetacrit(lat,j,tgh2,alpha)
}else{
    # x == limite
    theta2 = 0
}
}

# Downside Triangle:
if (P_forme==2 & rap<0.){
    tgh1 = (haut + e) / (ir - x - limite)
    theta1 = thetacrit(lat,j,tgh1,alpha)
    if (x > limite){
        tgh2 = (haut + e) / (x - limite)
        theta2 = thetacrit(lat,j,tgh2,alpha)
    }else if(x < limite){
        tgh2 = (haut + e) / ( x - limite)
        theta2 = -thetacrit(lat,j,tgh2,alpha)
    }else{
        theta2 = 0
    }
}

# Upside Triangle:
if (P_forme==2&rap>0.){ then
    tgh1 = (haut + e) / (ir - x - limite)
    theta1 = thetacrit(lat,j,tgh1,alpha)
    if (x < limite2){
        if (x > limite){
            tgh2 = haut / ( x - limite)
            theta2 = thetacrit(lat,j,tgh2,alpha)
        }else if(x < limite){
            tgh2 = haut / (limite - x)
            theta2 = -thetacrit(lat,j,tgh2,alpha)
        }else{
            theta2 = 0.0
        }
    }else if(x >= limite2){
        tgh2 = (haut + e) / x
        theta2 = thetacrit(lat,j,tgh2,alpha)
    }
}

# NB: is there a missing else here ? kg is modified just after...

```

Table 2.2: Example of the computation of the incident direct radiation ratio kgdirect. See introduction section for further details on the crop.

X index	X value (m)	Theta 1 (rad)	Theta 2 (rad)	Theta 1 (deg)	Theta 2 (deg)	kgdirect
1	0.00	-0.7563093	0.1163553	46.66667	83.33333	0.2850744
2	0.05	-0.7563093	0.0581776	46.66667	86.66667	0.3140484
3	0.10	-0.6981317	0.0000000	50.00000	90.00000	0.3213938
4	0.15	-0.6981317	-0.0581776	50.00000	93.33333	0.3504662
5	0.20	-0.6399541	-0.1163553	53.33333	96.66667	0.3566258
6	0.25	-0.6399541	-0.1745329	53.33333	100.00000	0.3854034
7	0.30	-0.5817764	-0.2327106	56.66667	103.33333	0.3900624
8	0.35	-0.5235988	-0.2908882	60.00000	106.66667	0.3934016
9	0.40	-0.4654211	-0.2908882	63.33333	106.66667	0.3678012
10	0.45	-0.4654211	-0.3490659	63.33333	110.00000	0.3954097

```

if(e > 2.1e-2){
  kg = cos(theta1)
  kg = cos(theta2)
}
kg = 0.5 * (cos(pi/2 + theta1) + cos(pi/2 + theta2))
kg = max(kg,0.0)

Output$theta1[Output$x_index==i]= theta1
Output$theta2[Output$x_index==i]= theta2
Output$kg[Output$x_index==i]= kg
}

Output$theta1_deg= (pi/2+Output$theta1)*180/pi
Output$theta2_deg= (pi/2-Output$theta2)*180/pi

colnames(Output)= c("X index","X value (m)","Theta 1 (rad)","Theta 2 (rad)",
                    "Theta 1 (deg)","Theta 2 (deg)","kgdirect")

kable(Output, caption= paste("Example of the computation of the incident direct",
                            "radiation ratio kgdirect. See introduction section",
                            "for further details on the crop."))

```

Where θ_1 and θ_2 are the angles visible in fig.3 and kgdirect the proportion of the semi-hemisphere that receive direct light, considering sun position throughout the day by using stand latitude, day of year and row orientation. In the model, θ_1 and θ_2 are computed in radian relative to the vertical plane above X. To provide a simpler representation, θ_1 and θ_2 are also given in degrees relative to the horizontal plane (i.e. soil or dominated plant species surface) in tab.2.2.

2.3.2 Incident diffuse radiation from the atmosphere

The incident diffuse radiation for all angles between θ_1 and θ_2 is computed using G , the apparent tangent to the considered θ . First the model computes it for θ_1 , and uses it as a threshold under which no diffuse radiation comes from the atmosphere because only transmitted radiation reaches this angle for the considered X point. Second, the model tests if the X point is below the target crop canopy, and if not it computes G for θ_2 , and applies the same methodology, with G being the upper threshold this time. This method avoids computing the left-hand quarter of the hemisphere since it receives only transmitted light necessarily if it is under the targeted crop.

Here is an example of the computation of the diffuse proportion received at each X point for the same crop as above (interrow= 1 meter, width= 0.2 meter, thickness= 0.1 meter, height= 1 meter) using the `kdif()` function:

```
# Initializations:
htab= c(9.23, 9.23, 9.23, 9.23, 9.23, 10.81, 10.81, 26.57,
       26.57, 26.57, 31.08, 31.08, 31.08, 31.08, 31.08, 47.41,
       47.41, 47.41, 52.62, 52.62, 69.16, 69.16, 69.16)
aztab= c(12.23, 59.77, 84.23, 131.77, 156.23, 36, 108, 0, 72, 144, 23.27, 48.73, 95.27,
        120.73, 167.27, 0, 72, 144, 36, 108, 0, 72, 144)
S0Ctab= c(0.0043, 0.0043, 0.0043, 0.0043, 0.0043, 0.0055,
          0.0055, 0.0140, 0.0140, 0.0140, 0.0197, 0.0197,
          0.0197, 0.0197, 0.0197, 0.0336, 0.0336, 0.0336,
          0.0399, 0.0399, 0.0495, 0.0495, 0.0495)
xprec= 0

if(ir<1.0){interval = 200.}else{interval = 20.}
if(1>ir/2.){l = ir/2.}

# Saving some intermediate results for Output_diff here (not in the real function):
x_s= (1:(ir / 2 * interval))
Output_diff= data.frame(x_index= rep(x_s,each=23),
                        x_value= rep((x_s-1)/interval,each=23),
                        Angle = rep(1:23, length(x_s)),
                        G1= rep(NA_real_,length(x_s)*23),
                        G2= rep(NA_real_,length(x_s)*23),
                        hcrit= rep(0,length(x_s)*23),
                        kgdiffus= rep(0,length(x_s)*23))

# Loop over Xs:

for(i in x_s){
  x = (i-1) / interval
  if(xprec<=l/2&x>=l/2){ilim = i}
```

```

xprec = x

# kdif:

x = min(x,ir/2)
e = abs(l*rap)
kgdiffus = 0.

# Computing for the right-hand of the row:
G1 = (haut+e)/(ir-x-l/2.) # the apparent tangent h1
for(j in 1:23){
  hcrit = atan(G1*sin(aztab[j]/180*pi))/pi*180
  if(hcrit<htab[j]){
    kgdiffus = kgdiffus+SOCtab[j]
    ##### Not in the real function #####
    Output_diff$kgdiffus[Output_diff$x_index==i&Output_diff$Angle==j]= SOCtab[j]
    #####
  }
  #####
  Output_diff$hcrit[Output_diff$x_index==i&Output_diff$Angle==j]= hcrit
  Output_diff$G1[Output_diff$x_index==i&Output_diff$Angle==j]= G1
  #####
}
# Computing for the left-hand row, if the point is not a shaded point:
if(x>l/2){
  G2 = (haut+e)/(x-l/2) # the apparent tangent h2
  for(j in 1:23){
    hcrit = atan(G2*sin(aztab[j]/180*pi))/pi*180
    if(hcrit < htab[j]){
      kgdiffus = kgdiffus+SOCtab[j]
      ##### Not in the real function #####
      Output_diff$kgdiffus[Output_diff$x_index==i&Output_diff$Angle==j]=
        Output_diff$kgdiffus[Output_diff$x_index==i&Output_diff$Angle==j]+SOCtab[j]
      #####
    }
    #####
    Output_diff$hcrit[Output_diff$x_index==i&Output_diff$Angle==j]= hcrit
    Output_diff$G2[Output_diff$x_index==i&Output_diff$Angle==j]= G2
    #####
  }
}
colnames(Output_diff)= c("X index","X value (m)","angle","G1 (tangent)",

```

```

          "G2 (tangent)", "hcrit", "kgdiffus")
if(knitr:::pandoc_to() == "html") {Output_diff} else{
  kable(Output_diff[1:10,],
        caption = paste("Computation of the diffuse light incident on each",
                        "point x. Ten first rows only, for the full data,",
                        "please refer to the html version of this book"),
        col.names = c("X index", "X value (m)", "angle", "G1 (tangent)",
                     "G2 (tangent)", "hcrit", "kgdiffus"), longtable = TRUE)
}

```

Table 2.3: Computation of the diffuse light incident on each point x. Ten first rows only, for the full data, please refer to the html version of this book

X index	X value (m)	angle	G1 (tangent)	G2 (tangent)	hcrit	kgdiffus
1	0	1	1.222222	NA	14.51577	0.000
1	0	2	1.222222	NA	46.56057	0.000
1	0	3	1.222222	NA	50.56789	0.000
1	0	4	1.222222	NA	42.35117	0.000
1	0	5	1.222222	NA	26.22654	0.000
1	0	6	1.222222	NA	35.69363	0.000
1	0	7	1.222222	NA	49.29501	0.000
1	0	8	1.222222	NA	0.00000	0.014
1	0	9	1.222222	NA	49.29501	0.000
1	0	10	1.222222	NA	35.69363	0.000

The total diffuse radiation incident to each point X is simply the cumulative of each angle:

```

suppressWarnings(suppressPackageStartupMessages(library(dplyr)))
Output_diff%>%
  group_by(`X index`)%>%
  summarise(kgdiffus= sum(kgdiffus))

kable(Output_diff,caption= paste("Example of the computation of the incident diffuse",
"radiation ratio kgdiffus. See introduction section for further details on the crop."))

```

2.3.3 Total diffuse and direct radiation from the atmosphere incident to each X point

The total radiation from the atmosphere incident to each X point is computed by using the incident diffuse (`rdif`) and direct (`rdirect`) radiation coming from the atmosphere weighted respectively by the previously computed `kgdiffus` and `kgdirect` as follow: $rdroit = (kgdiffus \cdot rdif) + (kgdirect \cdot rdirect)$

Table 2.4: Example of the computation of the incident diffuse radiation ratio kgdiffus. See introduction section for further details on the crop.

X index	kgdiffus
1	0.3489
2	0.3489
3	0.3489
4	0.4061
5	0.3864
6	0.3864
7	0.3864
8	0.4023
9	0.4023
10	0.3624

2.3.4 Light transmited by the dominant crop to the X points

The light transmitted by the dominant crop to the dominated crop (`rtransmis`) is computed using the total radiation from the atmosphere incident to each X point (`rdroit`) and the effective targeted plant leaf area index as follow:

$$rtransmis = (1.0 - rdroit) \cdot e^{-P_{ktro} \cdot (lai + eai)}$$

where P_{ktro} is the targeted plant light extinction coefficient, `lai` the plant leaf area index, and `eai` the equivalent leaf area index, which represent the photosynthetic surface that is not from leaves (e.g. wheat ears, rapeseed pods, pea pods or grapes during their green stage).

2.3.5 Total light incident to X points

The total light incident to each X point is the sum of the three components: atmospheric diffuse light, atmospheric direct light and transmitted light by the dominant crop. Taking our previous example again, this computation leads to:

```
rdirect= 1-rdif
rdroit= (Output_diff$kgdiffus*rdif)+(Output$kg*rdirect)
rtransmis = (1.0 - rdroit) * (exp(-P_ktroc * (lai + eai)))

incident_light=
  data.frame(Output[,1:2],Total=rdroit+rtransmis, rdroit= rdroit,
            rtransmis= rtransmis,
            light_env= ifelse(Output$`X value (m)` <limite,"Shaded","Sunlit"))
kable(incident_light,
      col.names = c("X index","X value (m)","Total incident ratio",
                  "Atmospheric ratio (rdroit)","Transmitted ratio (rtransmis)" ),
```

Table 2.5: Example of the computation of the total incident radiation ratio. See introduction section for further details on the crop.

X index	X value (m)	Total incident ratio	Atmospheric ratio (rdroit)	Transmitted ratio (rtransmis)
1	0.00	0.3387059	0.3106046	0.0281
2	0.05	0.3553817	0.3279890	0.0273
3	0.10	0.3596093	0.3323963	0.0273
4	0.15	0.3982891	0.3727197	0.0255
5	0.20	0.3942753	0.3685355	0.0255
6	0.25	0.4108381	0.3858020	0.0250
7	0.30	0.4135196	0.3885975	0.0249
8	0.35	0.4215422	0.3969610	0.0249
9	0.40	0.4068080	0.3816007	0.0251
10	0.45	0.4073885	0.3822058	0.0251

```
"Light environment"),
caption=paste("Example of the computation of the total incident radiation ratio.",
             "See introduction section for further details on the crop."))

```

The Total incident ratio is computed as: $\text{Total incident ratio} = \text{rdroit} + \text{rtransmis}$. It is the proportion of the incident light from the atmosphere that reach the point on the plane below the plant canopy. This light either comes directly from the sky (rdroit) or passing through the plant canopy (rtransmis). Consequently, the first point X of our example that is right under the plant canopy receives in average 33.8705892% of the light coming from the atmosphere: 31.0604617% of the light directly from the sky (either direct or diffuse light) + 2.8101275% of the light that is transmitted by the plant above.

2.3.6 Total radiation incident to the plane

After computing the total incident ratio for each point, the model averages the values between the so-called sunlit and shaded components. Each point positioned right under the plant canopy is considered shaded, and all other is considered sunlit (see fig. 2.3). using our previous example, this computation gives:

```
incident_light %>%
  group_by(light_env) %>%
  summarise(Total= mean(Total)) %>%
  transmute(Variable= c("rombre", "rsoleil"), light_env, Total)

kable(incident_light,
      col.names = c("STICS variable name", "Light environment",

```

Table 2.6: Example of the computation of the total incident radiation ratio for each light environment for a horizontal plane below the targeted plant canopy. See introduction section for further details on the crop.

STICS variable name	Light environment	Total incident light ratio
rombre	Shaded	0.3470438
rsoleil	Sunlit	0.4015338

```
"Total incident light ratio"),
caption= paste("Example of the computation of the total incident radiation",
             "ratio for each light environment for a horizontal plane below",
             "the targeted plant canopy. See introduction section for further",
             "details on the crop."))
```

Finally, both ratios are used to compute the intercepted PAR `raint` (MJ m⁻² d⁻¹) of the target plant as:

$$raint = P_{parsurrg} \cdot \text{trg} \cdot (1 - (rombre \cdot \text{surfAO}) - (rsoleil \cdot \text{surfAS}))$$

where $P_{parsurrg}$ is a coefficient to compute PAR (Photosynthetically Active Radiation) from the global radiation, `trg` is the active radiation (either global radiation or radiation transmitted below the dominant crop), `rombre` and `rsoleil` are the ratio of incident light for shaded and sunlit components of the plane, and `surfAO` and `surfAS` are the relative surfaces (0 to 1) of the shaded and sunlit components of the plane. The computation of `trg` is further described in Chapter 3.

```
library(tidyverse)
library(magrittr)
P_parsurrg= 0.5
incident_light%>%spread(light_env,Total)
surfAO = 1/ir
surfAO = min(surfAO,1.0)
surfAS = 1.0 - surfAO
raint = P_parsurrg * trg * (1 - (incident_light$Shaded[1] * surfAO) -
                           (incident_light$Sunlit[2]* surfAS))
```

For our example, the crop would have intercepted 7.62 MJ m⁻² day⁻¹ of radiation, with a PAR of 12.5 W m⁻², an `rombre` of 0.3470438, an `rsoleil` of 0.4015338 and a relative surface of 0.2 and 0.8 for the shaded and sunlit component respectively.

The relative surfaces (0 to 1) of the shaded and sunlit components of the plane (`surfAO` and `surfAS`) that are computed during the dominant plant computation are then used as the shaded and sunlit surfaces for the dominated plant.

2.4 Summary

The interception of the targeted plant is obtained by:

1. Computing the light that is incident at a horizontal plane at the height of the dominated plant (or the soil) and,
2. Subtracting this incident light to the global PAR, which gives the PAR intercepted by the target plant.

This process is applied iteratively to the dominant and the dominated plant to compute both species PAR interception while taking account for their respective structure (shape, height, width...).

2.5 Discussion and proposed modifications

The dominated plant interception is computed in two separate computations, one for the sunlit component, and one for the shaded component, and interception is then weighted by their relative surface. The dominant plant is considered having 100% of its surface that is sunlit, so all this computation is made only for its sunlit part.

The radiation above the dominant plant (`trg`) is the global radiation, but the radiation above the dominated plant (also `trg`) is computed as: $trg = trg \cdot rsoleil$ where `rsoleil` is the average proportion of light intercepted by the sunlit area of the plane below the dominant plant.

This computation is only right for the sunlit component of the dominated plant. A modification is proposed, discussed and tested on Chapter 3.

Chapter 3

Light incident to the dominated crop (trg)

3.1 Computing the trg incident to the dominated plant

For the moment, STICS compute the radiation incident above the dominated (or associated) plant as:

Which means that trg for the dominated plant ($i > 1$) is computed as the global atmospheric radiation (`trg_bak` here), reduced by the average proportion of light transmitted by the sunlit area of the plane below the dominant plant (and above the dominated plant). This computation does not consider that the average proportion of light incident above the shaded part of the dominated plant (`rombre`) is different from `rsoleil`. See Chapter 2 for more details.

3.2 Proposed solution

We propose to change this computation to take the relevant incident light according to the light regime of the dominated plant under computation:

With this new computation, the radiation incident above the dominated plant depends from the component under consideration (shaded or sunlit), and is computed using the geometry of the dominant plant (for atmospheric+transmitted light computation).

3.3 Results

A comparison of the two was made using the sticRs package, from which a summary plot was extracted. The results are shown in Figure 3.1.

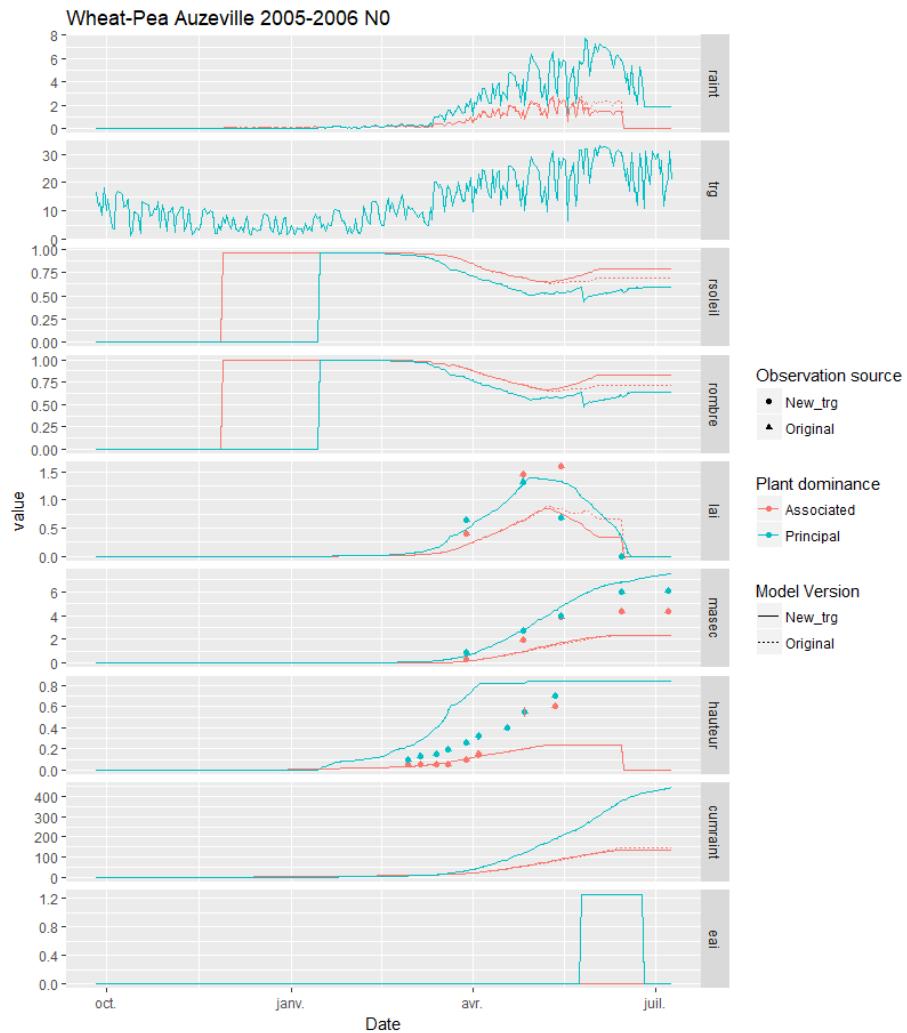


Figure 3.1: trg comparison

The comparison between both indicated that the dominated plant intercepted more PAR with the original computation (`raint`), due to its wrong light regime (`rsoleil` for both AS and AO). While the dry mass and height of the dominated plant did not change, its LAI was previously higher on the end of the rotation, which increased the `rsoleil` and `rombre` of the ground (visible as associated

ones). These simulations also showed that the wheat (dominant) **eai** was highly overestimated, which will be fixed in the next simulations.

Chapter 4

Wheat EAI parameterization

4.1 Introduction

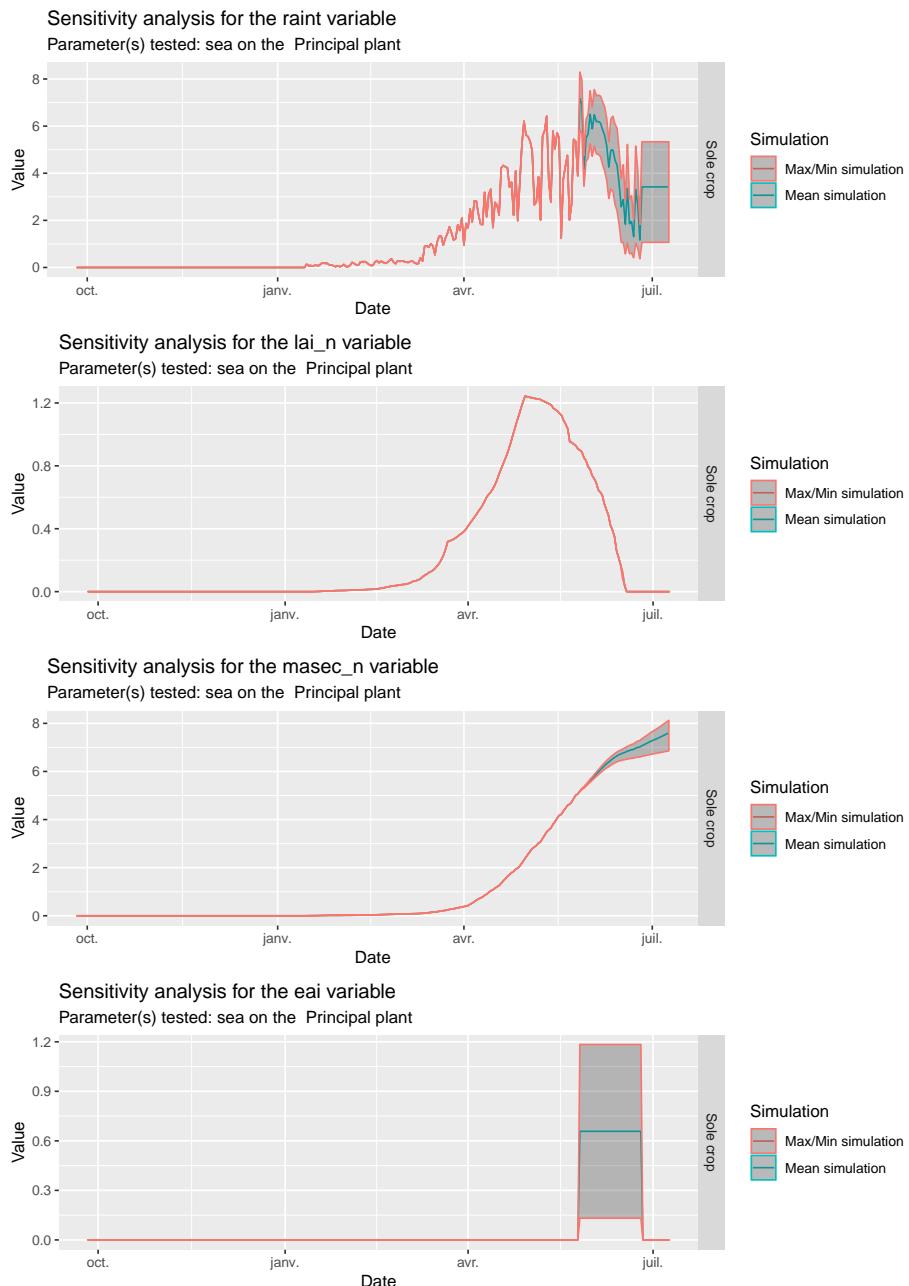
The simulations outputs from Chapter 3 have shown that the wheat ears equivalent photosynthetic surface area (`eai`) was probably overestimated by a 3-4 factor. This surface is computed using the fruit dry mass (`maenfruit`) and a parameter (`P_sea`) as follow:

$$eai = P_{sea} \cdot \frac{maenfruit}{100}$$

This parameter was originally equal to 100, meaning that each gram of fruit dry mass gave 1 m² of equivalent LAI. This value was probably overestimated, but before re-parameterizing the plant, we first needed to know the model sensitivity to this parameter.

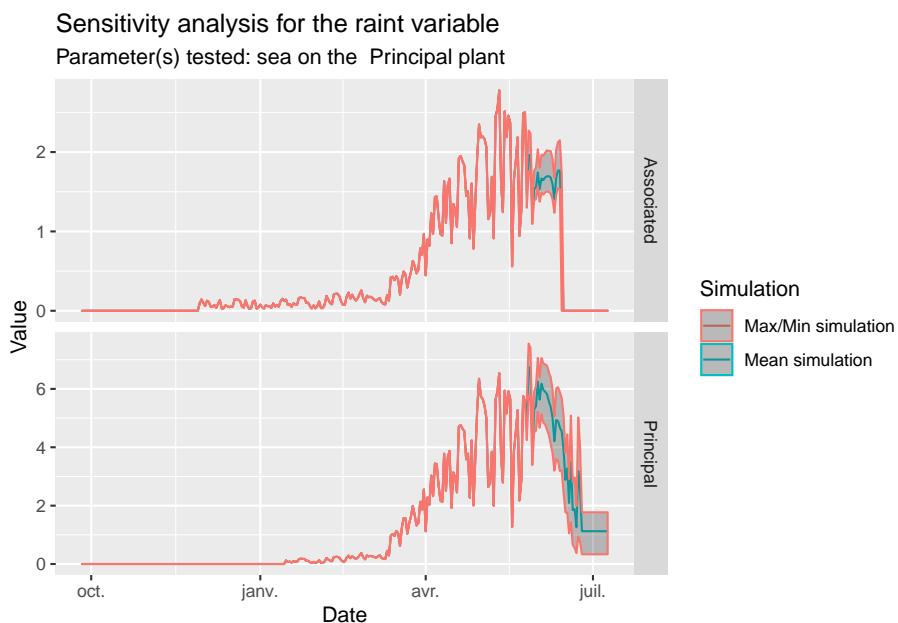
4.2 Model sensitivity to the `P_sea` parameter

The model sensitivity was assessed using the sticRs package using values from 10 to 90 for the `P_sea` parameter. The results are shown below (see `html` version of this book for interactivity):

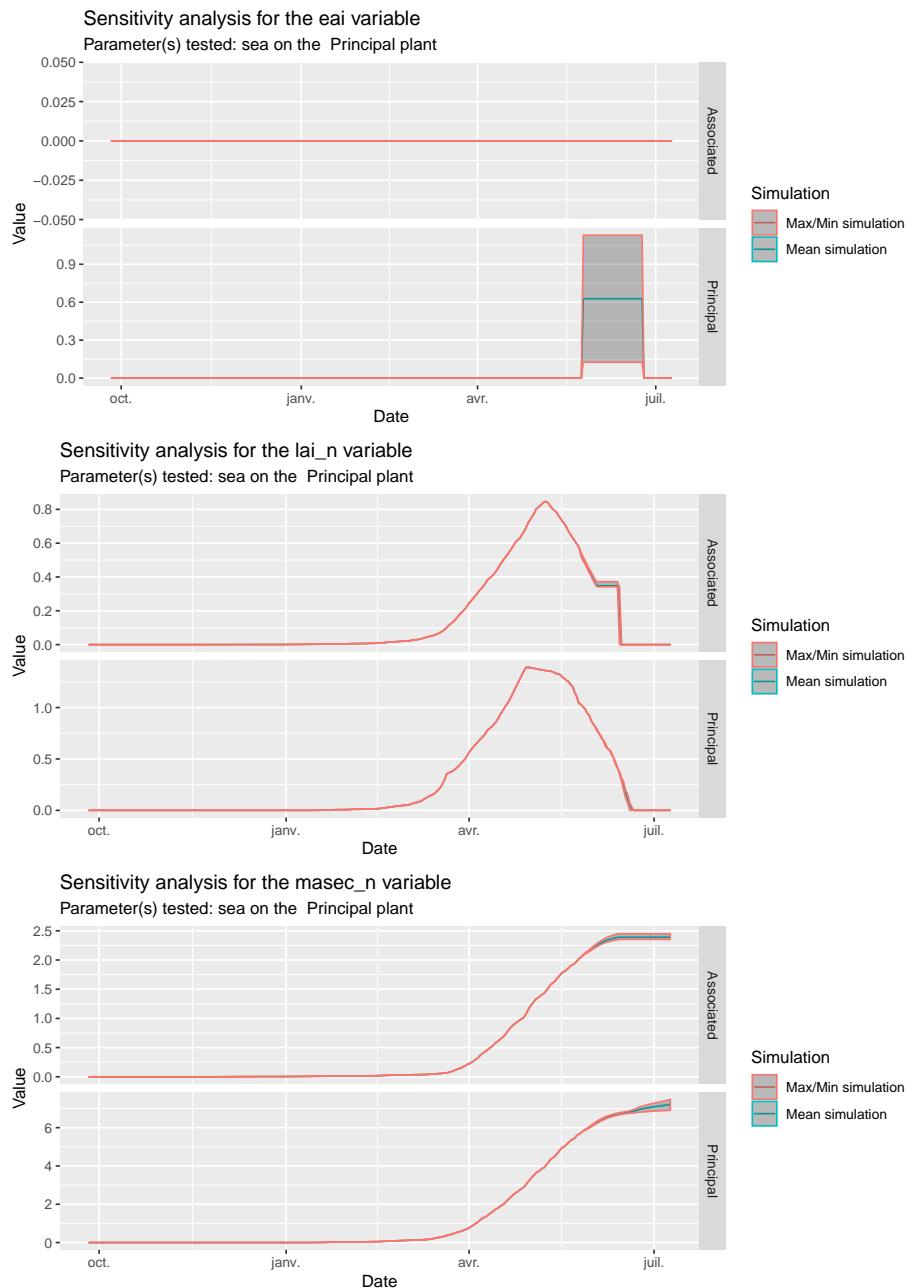


The results show that the higher the `P_sea`, the higher the `eai`, and hence increasing the light interception (`raint`), which induced a higher aboveground dry mass (`masec`). The plant LAI is not influenced by the `P_sea` though, because there is no retroactions on this variable.

In the case of intercrops, the P_sea parameter can influence the dominated plant also, because less light is transmitted when P_sea is high :



The other variables are not much impacted though:



4.3 Parameterization of the P_{SEA} parameter

Knowing that the key outputs from the model are not much impacted by the P_{SEA} parameter as soon as its value is not unreasonable, we tested several values of the parameter (*i.e.* 20, 50 and 80) and compared the outputs of the model against measurements for several types of management: Wheat in sole crop, Pea in sole crop, Wheat in self-intercrop, Pea in self-intercrop, and a Wheat-Pea intercrop. The results are available on this page. In short, the value we kept for the wheat was 20.

Chapter 5

Plant density and equivalent plant density

5.1 Introduction

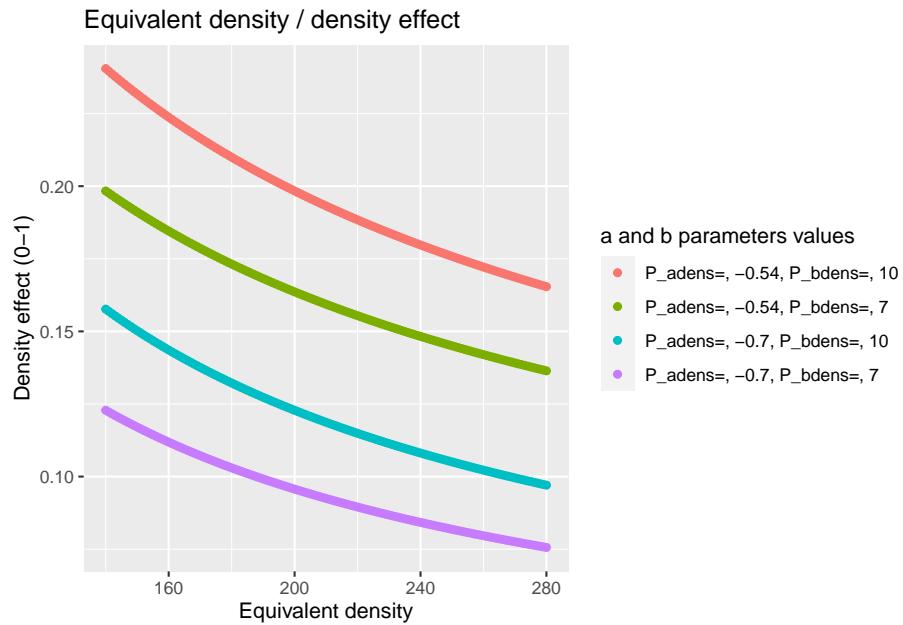
The plant density, which is related to the interrow distance, seems to be an important formalism to describe the crop, and particularly for mixed crops. Several computations are made to represent plant competition in the STICS model, making the density effect a complex process. Lets describe each step of the process to have a clearer representation in mind.

5.2 The density effect on LAI

In the model, the plant density is taken as a negative effect upon the LAI growth as soon as a threshold of LAI is reached. This threshold ($P_{laicomp}$) represents the moment when the leaf surface of a plant start becoming competitive for light against another plant (from the same species or not). So whenever the LAI is higher than $P_{laicomp}$, the effect of the density ($efdensite$) become closer to 0 (the effect is null when equal to 1, and maximum at 0). This effect is computed as: $ef_{densite} = \min \left\{ 1.0 ; e^{P_{adens} \cdot \frac{\log(densiteequiv)}{P_{bdens}}} \right\}$ or more simply:
 $ef_{densite} = \min \left\{ 1.0 ; \left(\frac{densiteequiv}{P_{bdens}} \right)^{P_{adens}} \right\}$

Replace the equation in the model to simplify too ?

Here is a plot representing the density effect along the equivalent density:



So the higher the density, the higher the negative effect on LAI.

5.3 The equivalent density

In sole crops, the density effect is straightforward. However, under the case of mixed crops, the density effect can be higher for the dominated plant compared to its equivalent in sole crops. Indeed, a pea in sole crop would have a given competition with other close plants, but a different one when mixed with wheat, where the same density of wheat can give higher competition effect for light because it is taller.

Then the density effect is computed as an equivalent density instead (`densiteequiv`), that can differ from the sowing density for the dominated crop to increase the negative effect of `efdensite` compared to a sole crop.

The previous implementation in STICS was simple. As soon as a plant become dominated, it had an increased equivalent density compared to its actual density (*e.g.* doubled). After some discussion with the STICS intercrop team, this formalism was considered not suitable for two reasons:

- Whenever a plant become shorter its density drastically increases regardless of the height difference. We propose to modify the formalism to consider the height difference explicitly while computing the equivalent density.
- For the moment, the competition induced by the density of the other

species (*i.e.* interspecies competition) was re-included in the equivalent density computation using the `P_bdensp1` and `P_bdensp2` parameters, which come from the sole crop formalism, where it is used to consider intra-species competition effect. The equivalent density was originally set up to consider the decrease in light quality when a plant has another plant above it. Adding another computation of competition based on the density of the other plant could be redundant, because it is already considered during the light interception (see Chapter 6 and 2). Consequently, we propose to remove this computation from the equivalent density formalism, and to let the `P_bdensp1` and `P_bdensp2` parameters being used to compute the `efdensite` variable only.

5.4 Proposition for new equivalent density formalism

5.4.1 Definition of the different propositions

The user will now have four options to compute the equivalent density, depending on the value of the `P_code_calc_equi_density` parameter:

- (1): Previous computation : $density_{Equivalent} = density_{p2} + \frac{density_{p1}}{P_{bdensp1}} \cdot P_{bdensp2}$
- (2): Improved computation to define the equivalent density as a function of the height difference between the plants as follow: $density_{Equivalent} = \begin{cases} \Delta_{height} > hauteur_{threshold} & density_{p2} + \frac{density_{p1}}{P_{bdensp1}} \cdot P_{bdensp2} \\ \Delta_{height} < hauteur_{threshold} & density_{p2} + slope \cdot abs|\Delta_{height}| \end{cases}$
with $diffx = \frac{density_{p1}}{P_{bdensp1}} \cdot P_{bdensp2}$ and $slope = \frac{diffx}{hauteur_{threshold}}$
- (3): Improved computation to define the equivalent density as a function of the height difference between the plants, and using different parameters for intraspecific and interspecific competition, with simpler meanings. This formalism would give no increase in equivalent density until a given threshold (`hauteur_threshold`), after which a progressive increase in the equivalent density would happen until the maximum allowed. In this computation, the slope would be computed by the model, and the equivalent density would be computed as follows: $density_{Equivalent} = \begin{cases} \Delta_{height} < hauteur_{threshold} & density_{p2} \\ \Delta_{height} > hauteur_{threshold} & b + \Delta_{height} \cdot slope \\ \Delta_{height} > hauteur_{threshold_2} & Max_{equDens} \end{cases}$
with $slope = \frac{Max_{equDens}}{hauteur_{threshold_2} - hauteur_{threshold}}$. This formalism would include three parameters:
 - `hauteur_threshold`: the difference in height below which no com-

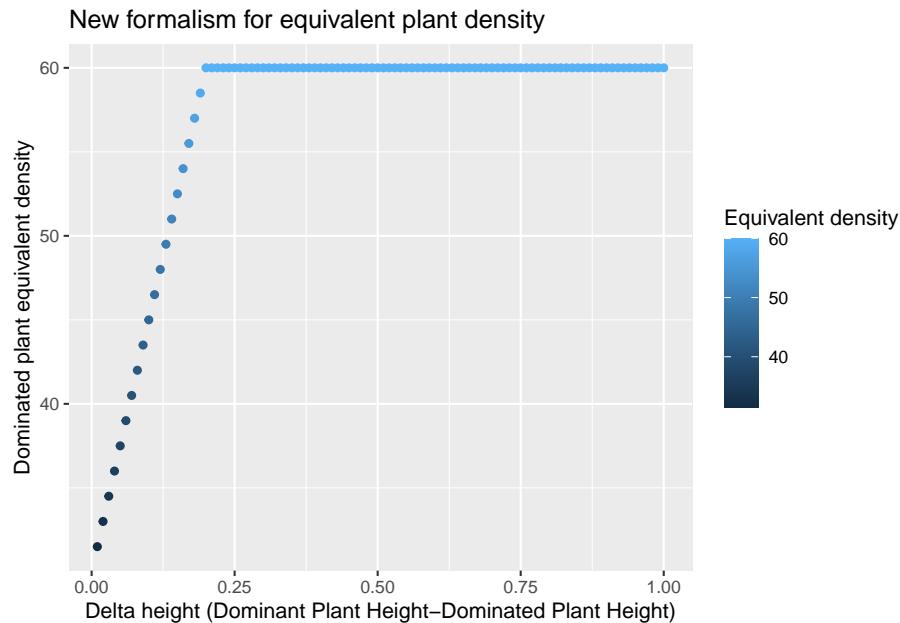
petition is occurring

- `hauteur_threshold_2`: the difference in height below which a progressive competition is occurring
- `Max_equDens`: the maximum equivalent density allowed (generally the sole crop density).
- Any other value: the equivalent density is equal to the true plant density

5.4.2 Visual representation of the options

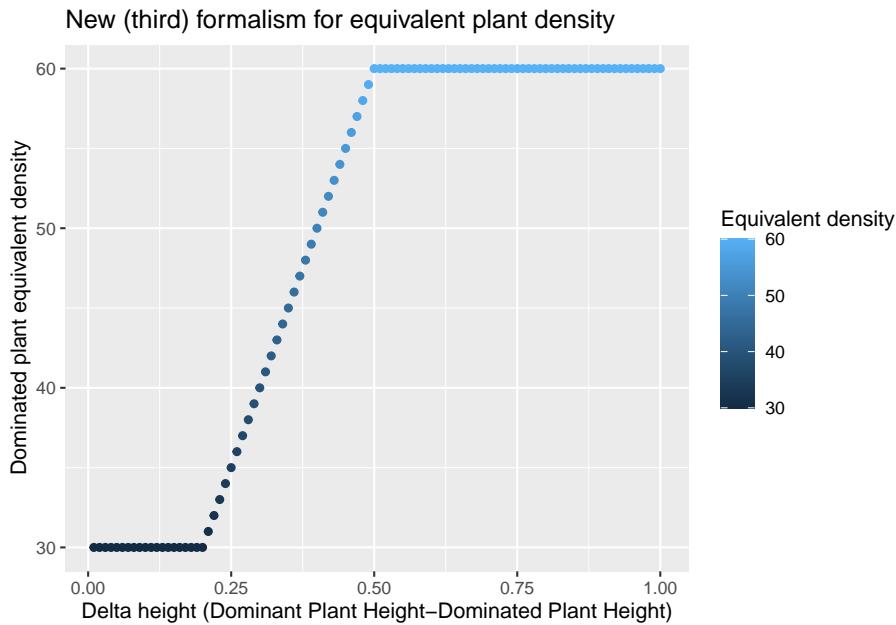
The two new formalisms are both linking the equivalent density to the difference of height between the two plants, but using different parameters and outputs.

Here is a visual representation of the equivalent density when `P_code_calc_equi_density=2` for a wheat-pea intercrop with a plant density of 140 for the wheat as the principal species and 30 for the pea as the associated species:



This option has several implications in the model, notably that the dominated plant is less impacted by the competition with the dominant plant when both have approximately the same height.

Here is a visual representation of the same output when `P_code_calc_equi_density=3`:



This formalism considers no interspecific competition for the quality of light while the difference in height is low, and then an increased competition until a maximum is reached.

5.4.3 Comparison between options

A comparison of the different formalisms was made using the sticRs package, from which a summary plot was extracted, and the results are shown in Figure 5.1.

It is not very clear from the figure, but the second and third options (*i.e.* `P_code_calc_equi_density= 2` and `P_code_calc_equi_density= 3`) for the equivalent density computation are both less severe than the previous one (`P_code_calc_equi_density= 1`) for the dominated plant. The second one has effect on the equivalent effect for limited time only (dashed red curve), while the third one show no increase in equivalent density all along the crop life because the height difference remains below the first threshold. Consequently, the dominated plant intercepts more light (see `cumraint` and `fapar`) and has a higher LAI using the third option.

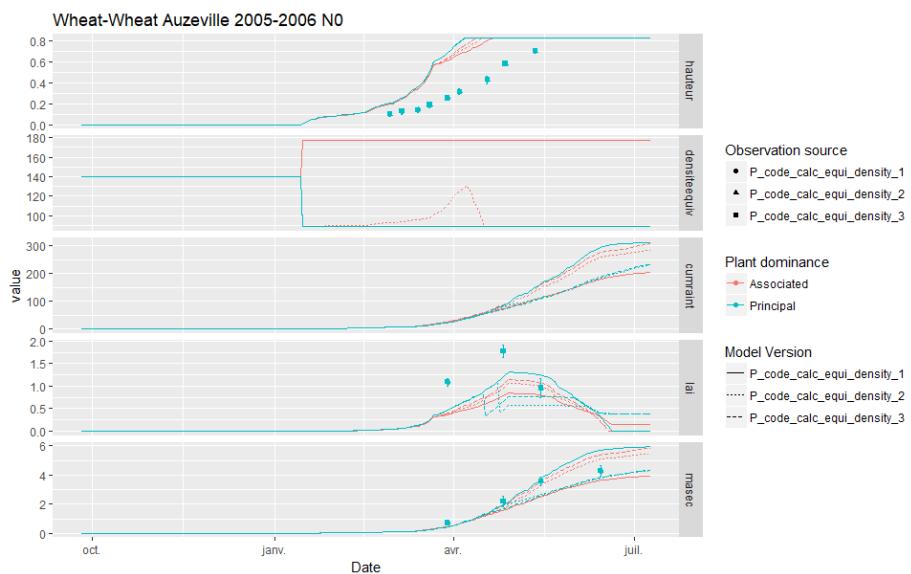


Figure 5.1: Comparing options for the equivalent density computation

Chapter 6

Interrow spacing

6.1 Introduction

The inter-row is the distance between two rows of the same plant species. Figure 6.1 shows a simple design with a field with two plant species sowed with the same inter-row.

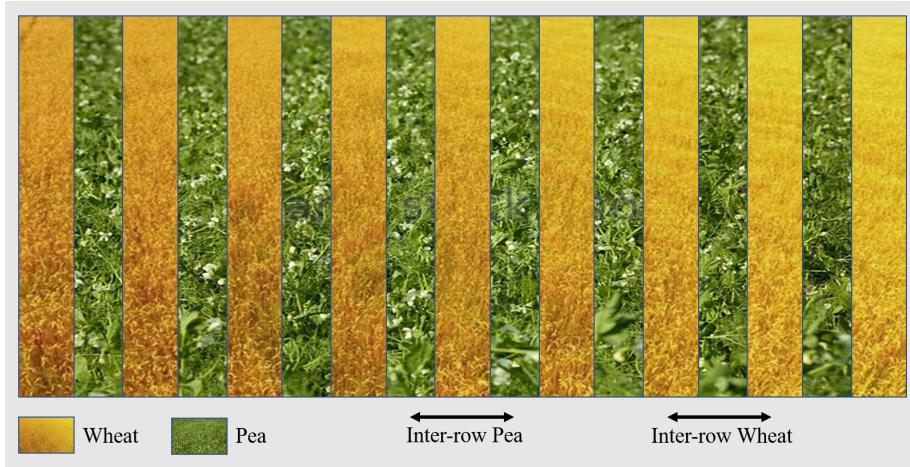


Figure 6.1: Pea-Wheat intercrop using the same inter-row spacing for the two crops

So far, so good. Now what happens if we set a different inter-row spacing for the two species ?

6.2 Inter-row spacing for mixed crops in the model

Whereas the model as a notion of the position of the plants along the interrow considering the same plant species (*i.e.* Principal or Associated), it does not explicitly position the different plant species between each other. Indeed, the light interception of the dominant plant is first computed using its geometry and a plane at the height of the dominated plant. Then, the light interception of the dominated plant is computed using the average light incident on the previous plane (separated between shaded and sunlit component), the plant geometry, and a second plane right above the soil. So when computing its light interception, the dominated plant do not consider at all the dominant plant interrow spacing, but only the light it transmits. Of course the dominant plant interrow spacing does impact the light that is transmitted to the dominated plant, but it is not an explicit description of the interrow spacing.

Figure 6.2 shows a depiction of how the model describe the interrow for intercrops.

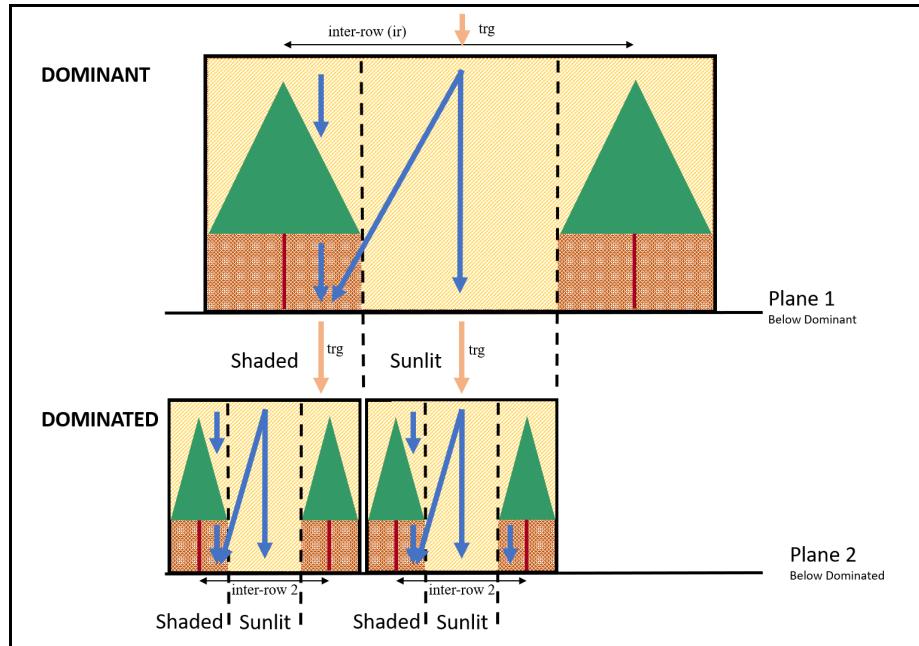


Figure 6.2: Interrow effect on light interception for intercrops in the STICS model

The model does effectively compute half the interrow light interception only, because it is assumed that the other half have the same light regime at daily time-scale. First the interrow of the dominant plant is used to position the

left and right dominant plants. The model then computes the global radiation (**trg**) that is transmitted to the Plane 1, which is then used to compute the light intercepted by the dominant plant. The light transmitted to the Plane 1 is divided into two light regimes: a shaded component (surface right under the dominant plant canopy) and a sunlit component (the opposite).

Second, the light incident on the Plane 1 is used as the **trg** for the dominated plant, and the same computation than for the dominant plant is performed for the shaded and sunlit components of the dominated plant, and then integrated at species-scale using both light interception and relative surface.

To conclude on this point, we see that the interrow spacing can only impact the light interception as a density effect (more plants per m², closer intra-species canopy), but not as a pure geometrical effect.

For more details on how the **trg** is computed, see Chapter 3, and for more details on light interception, see Chapter 2.

Chapter 7

Design

7.1 Introduction

It is difficult to understand well which cropping design (i.e. species arrangement) can be simulated using the STICS model formalisms. Based on the previous information from Chapter 2, 3, and 6, we present some use-cases where the model can be applied according to its formalisms, and where it cannot. The list of designs proposed here is not exhaustive, and the user should always think about the relevance of using STICS to model a particular design.

7.2 Designs that can be simulated

Figure 7.1 shows the depiction of some of the cropping designs that possibly can be simulated by the STICS model. The list is not exhaustive, but gives an overall look on the possibilities:

The list will be extended soon, and the different assumptions and domain of validity for each will be detailed too.

7.3 Designs that cannot be simulated

Figure 7.2 shows a design that cannot be simulated by the STICS intercrop model as is:

Indeed, strips implies that interspecies light competition is only present on the border of each strip, making the Dominant/Dominated paradigm unrealistic for this design. The model could simulate large strips as two separated crops though.

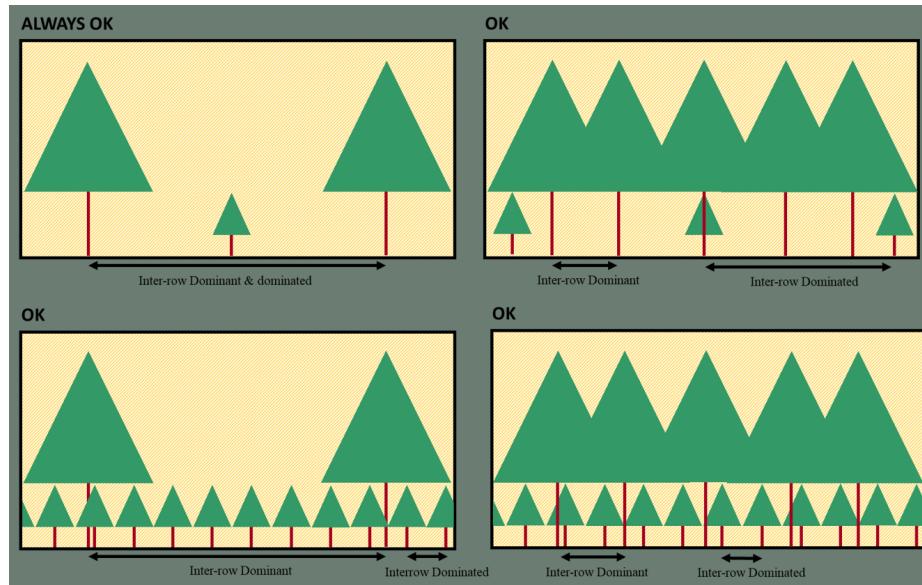


Figure 7.1: Depiction of the potentially adapted intercrop designs for simulation using the STICS model.

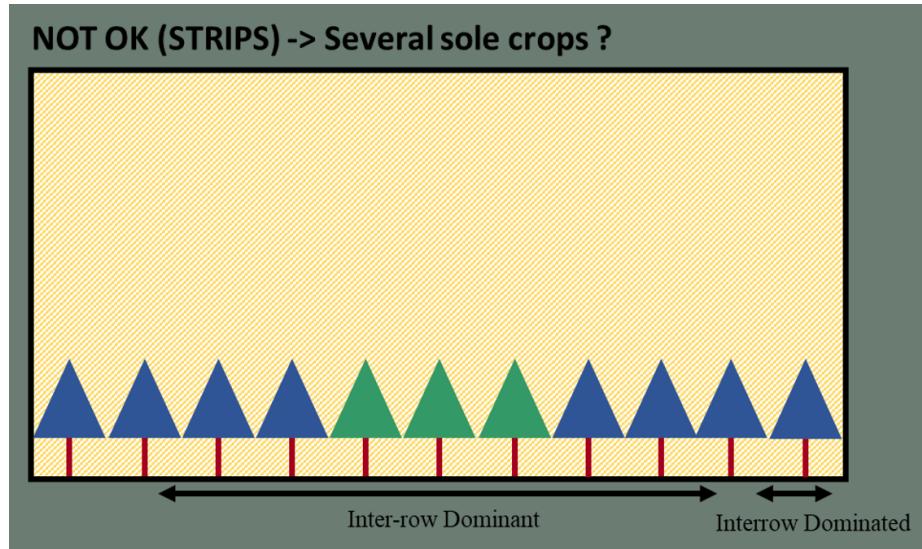


Figure 7.2: Depiction of the intercrop designs not adapted for simulation using the STICS model as is.

Chapter 8

Issue on LAI not decreasing

8.1 Introduction

The LAI is not decreasing on the end of the crop cycle for the dominant plant (Figure 8.1).

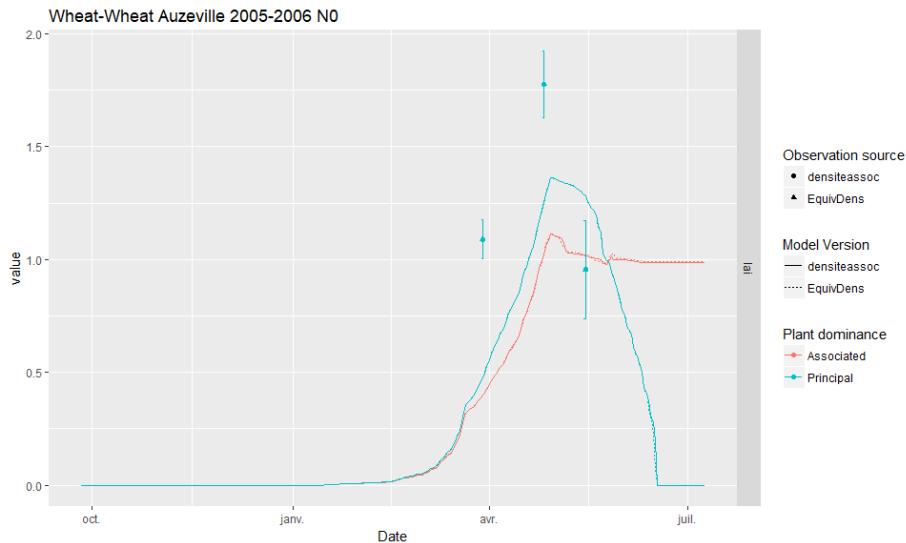


Figure 8.1: Leaf Area Index in a self-intercrop of Wheat-Wheat

This issue can have different origins. The main hypothesis are:

- The leaf surface may be too high compared to their mass, coming from an abnormally high `s1a`

- A too low senescence, coming itself from a too long leaf lifespan, which could arise from a too low nitrogen mobilization (increase the lifespan), and/or a too low water stress (water stress can decrease the life span).

8.2 Investigating the sla hypothesis

Figure 8.2 shows that both plants (*i.e.* Principal and Associated) have exactly the same SLA during the growing cycle. This discard the possibility of a problem coming from a wrong SLA because it would be much higher for the plant with the constant LAI at the end of the cycle.

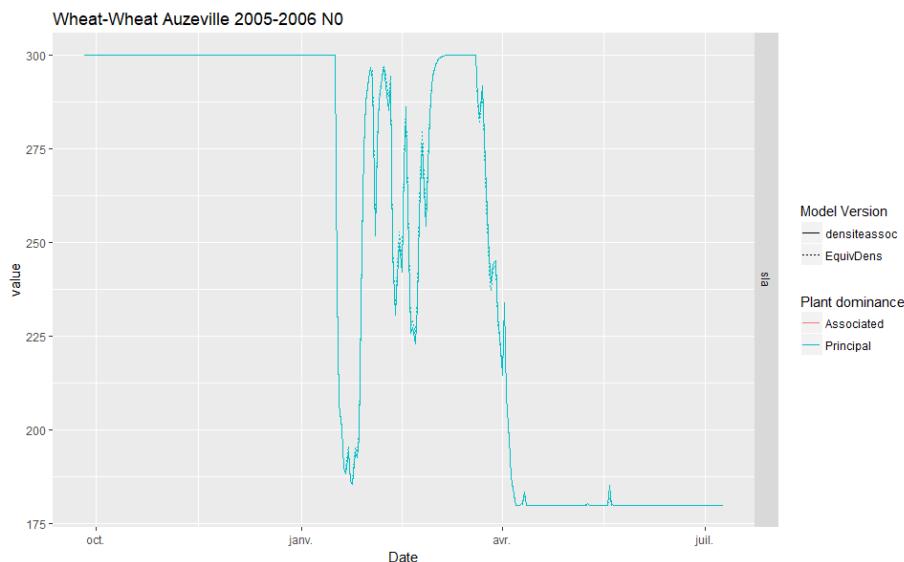


Figure 8.2: Specific Leaf Area in a self-intercrop of Wheat-Wheat

8.3 Leaf senescence

Leaf senescence is actually much lower for the plant that has a non-decreasing LAI (Figure 8.3)

The next step is to find out why leaf senescence is too low for this plant.

8.3.1 Leaf lifespan

The Figure 8.4 shows that the leaf lifespan is not much different between both plants either, rejecting our hypothesis of a too long lifespan.

However, printing the lifespan for each light environment (*i.e.* shaded, sunlit) showed that the lifespan of the shaded part of the dominated plant was equal

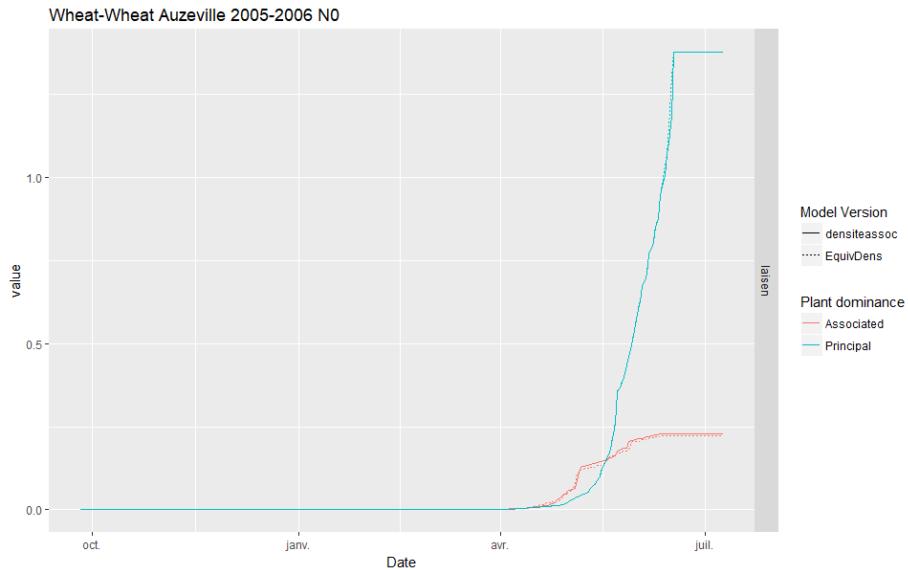


Figure 8.3: Leaf Senescence in a self-intercrop of Wheat-Wheat

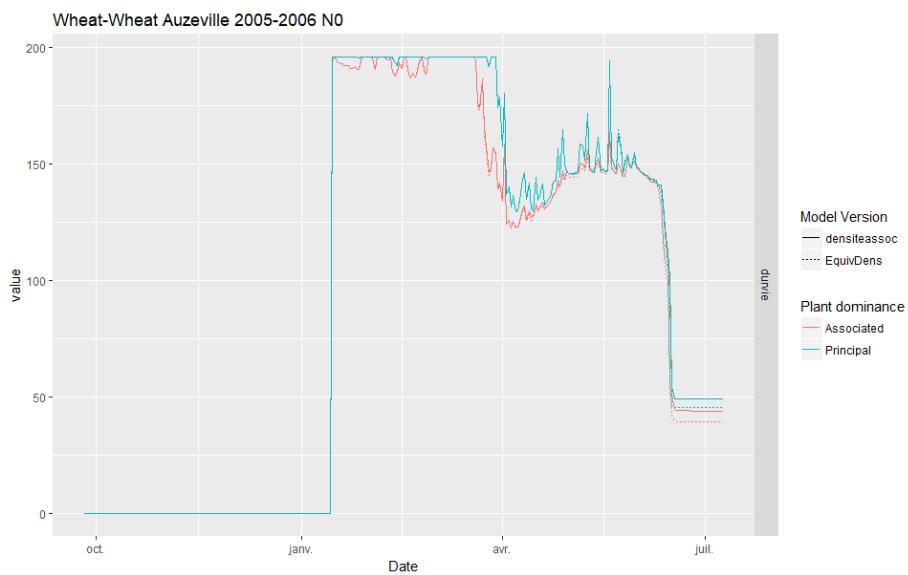


Figure 8.4: Leaf Lifespan in a self-intercrop of Wheat-Wheat

to 0 most of the time, making `nsencours` jump from `nlev` to the first day when senescence is computed (here at day 208) at once (Figure 8.5).

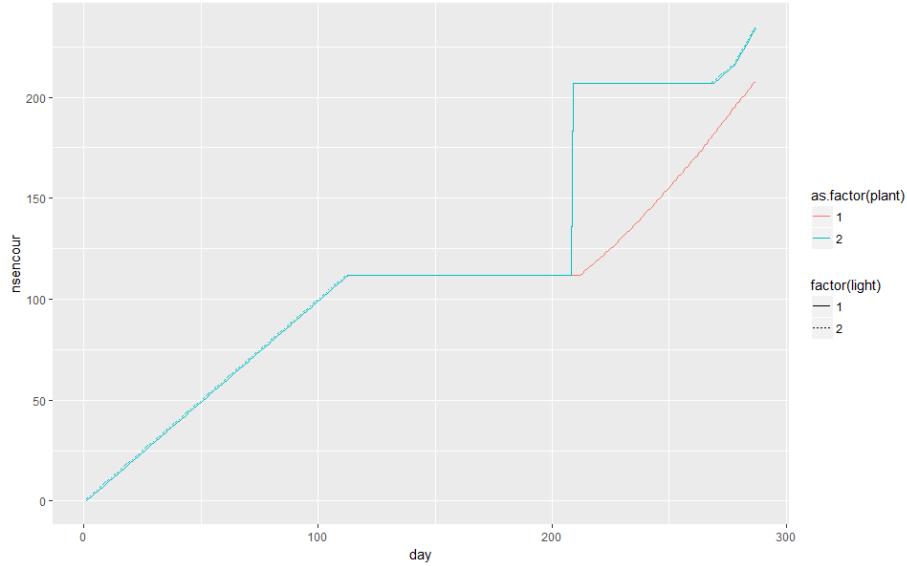


Figure 8.5: Leaf (lai) appearence day (`nsencour`) compared to the day (`day`) they become senescent (self-intercrop of Wheat-Wheat)

The leaf lifespan (`durvie(i)`) was computed as:

The bug came from the computation of `durvie(i)` that was initialized to 0, and always equal to 0 afterward until the `fin` was reached.

A correction is proposed to fix the bug, which make the same computation except on the first day on which the senescence is computed (`nsencour==nlev`):

The new computation fix the bug, making `nsencour` more progressive as for the dominated plant (Figure 8.6).

8.3.2 Stress indices

The senescence is computed using leaf life span, but also two different stress indices: `senfac` for the water stress and `innsenes` for the nitrogen stress. Figure 8.7 shows that the plant with non-decreasing LAI has a higher stress compared with the other plant, rejecting again the hypothesis of a too low stress indices.

8.3.3 Over-fertilization

The problem could arise from a nitrogen over-fertilization (`inn`) that could increase lifespan, but the Figure 8.8 shows the contrary, with a lower `inn` for the plant with non-decreasing LAI.

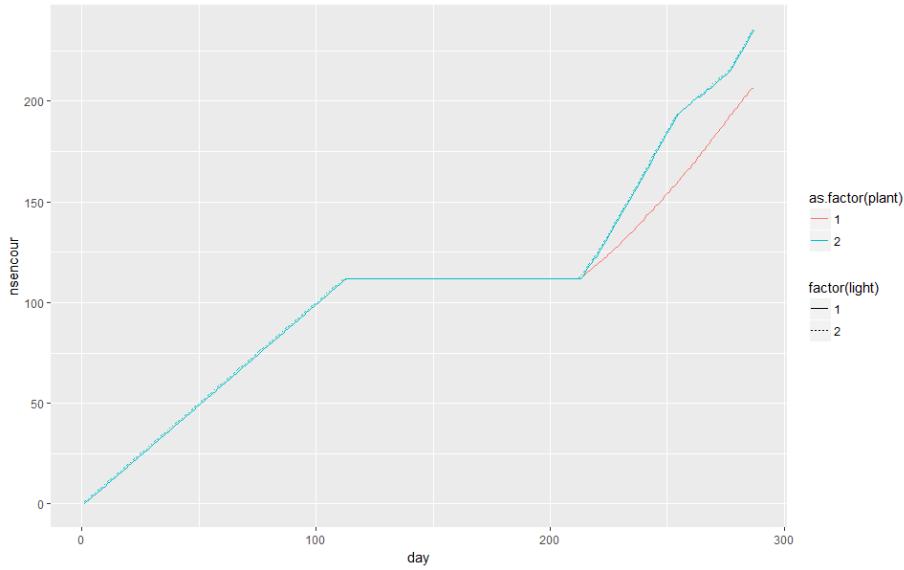


Figure 8.6: Leaf (lai) appearence day (*nsencour*) compared to the day (*day*) they become senescent using the new computation of durvie (self-intercrop of Wheat-Wheat)

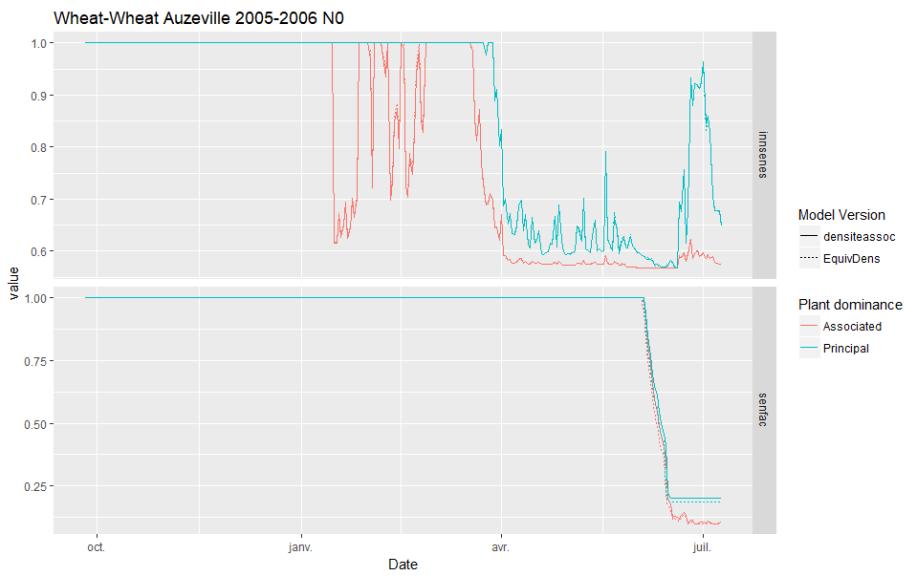


Figure 8.7: Stress index to compute leaf lifespan in a self-intercrop of Wheat-Wheat

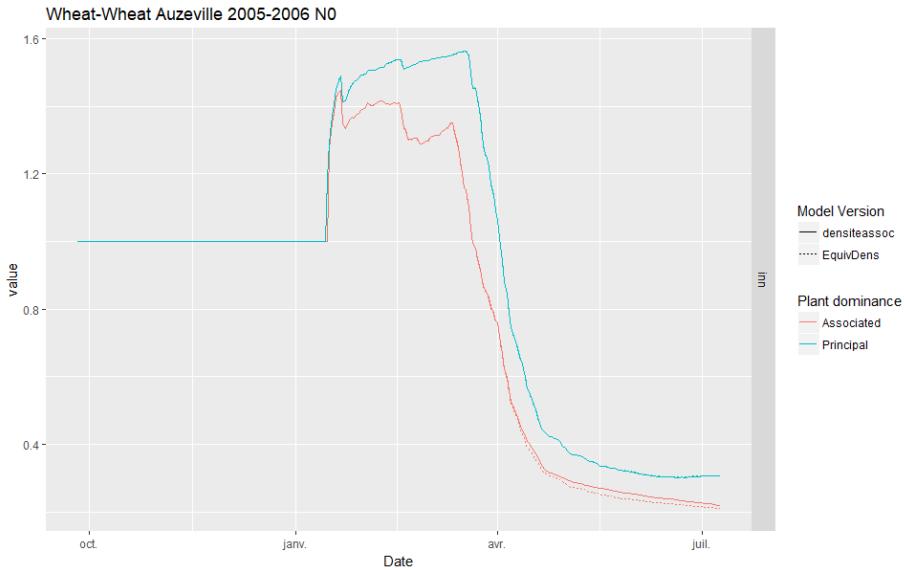


Figure 8.8: Nitrogen over-fertilization in a self-intercrop of Wheat-Wheat

8.3.4 De-activating water and nitrogen stresses

To check if the senescence is still related to water and nitrogen stresses, Figure 8.9 shows a simulation where both effects are removed (`P_codeh2oact= 2` and `P_codeinact= 2`).

Removing the stress do not change the fact that one plant has a constant LAI at the end of the crop cycle. This proves that the problem does not arise from a limitation in water or nitrogen.

8.4 The solution to the first issue

Three main variables were badly used to compute the senescence: `dltams`, `deltai` and `pfeuilverte`. These variables correspond to the daily increment in dry mass, daily increment in lai, and the green leaves proportion. The model used the values from previous computations kept in memory to compute the different components of the senescence for each light environment of the plant (*i.e.* sunlit or shaded) separately.

Unfortunately, the dominance of the plant can change over time, so a dominant plant with no shaded component can have one afterwards. If a dominant plant becomes dominated, the model was trying to use `dltams`, `deltai` and `pfeuilverte` of the shaded part of the plant that all were equal to 0 (as their initialization) because they were not computed before, making no senescence at all for the shaded part of the plant.

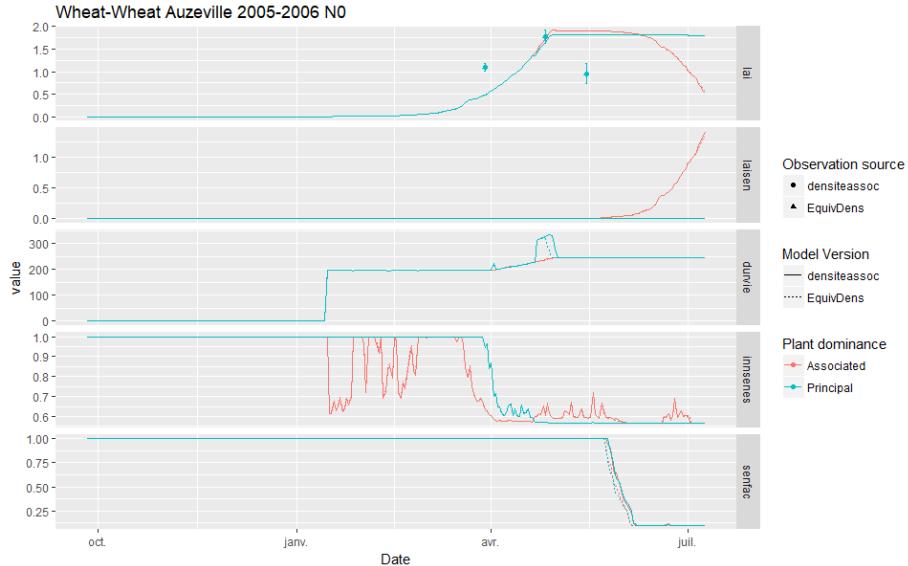


Figure 8.9: Simulation of a Wheat-Wheat self-intercrop with no nitrogen and water stresses

We propose to replace the use of a separated shaded and sunlit `dltams`, `deltai` and `pfeuilverte` by their total at plant level, weighted by their relative surface under shaded and sunlit environment. For example, the `deltai` input to the `senesce` function would be `p(i)%deltai(0,1:sc%nbjmax)*p(i)%surf(ens)` instead of `p(i)%deltai(ens,1:sc%nbjmax)`, with `p(i)%deltai(0,n) = p(i)%deltai(sc%AS,n)* p(i)%surf(sc%AS) + p(i)%deltai(sc%A0,n)* p(i)%surf(sc%A0)`. This new computation has the advantage of considering all the lai that was incremented before, with a drawback being not to consider the previous relative surface of the plant.

Figure 8.10 shows the resulting outputs, which are more realistic.

8.5 Fixing a second issue

After addressing the first issue, I added the new formalism to compute the equivalent density (see Chapter 5). While studying the effect of this new formalism, I stumbled upon the same apparent issue : the `lai` was not decreasing at the end of the crop cycle. This issue came from the fact that the sharing of the `deltai` for the dominated plant between the shaded and sunlit components was too simple (see previous paragraph). Indeed, the `deltai` was weighted according to their respective surface of the current day (`n`), but the senescence can happen on different days (and it is) because both light components do not experience the same temperature conditions, so the surface effectively used to weight `deltai`

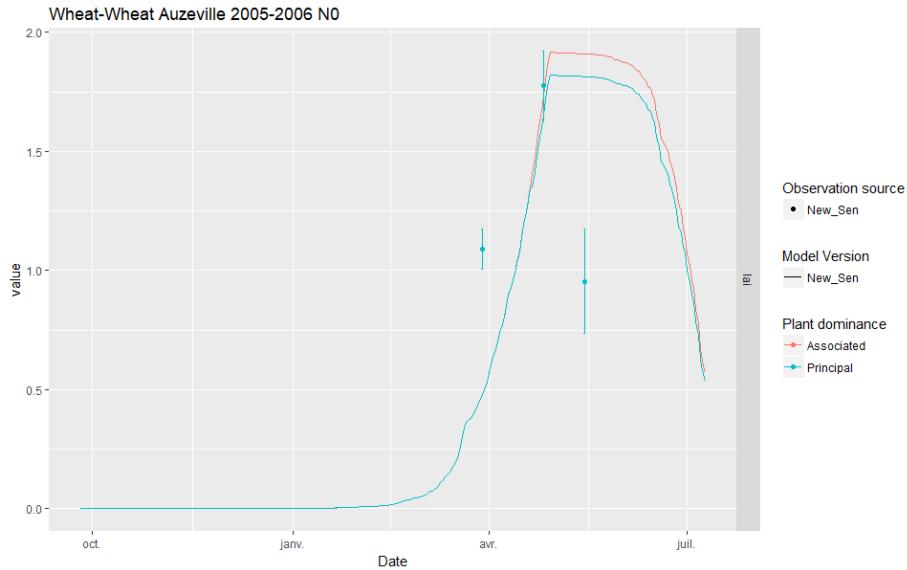


Figure 8.10: Simulation of a Wheat-Wheat self-intercrop leaf area index using the modified STICS version

can be different from one day to another, making either the total dominated plant `deltai` too high or too low, as seen in Figure 8.11, which shows an `lai` that do not go down to 0 at the end of the crop cycle.

The issue was partially fixed by adding a temporary variable (`deltai_sencour`) to keep trace of how much and when the senescent `lai` was created, to remove it from the overall pool (new variable, `deltai_tmp`) used by the shaded component afterwards. The simulations outputs using the modified STICS are visible in Figure 8.12.

This same issue was also found happening for the senescent mass, and the code was modified accordingly to correct it. However, the problem is not entirely solved, because the `lai` and `mafeuilleverte` are not going down to 0 at the end of the crop cycle yet.

This came from the way the model compute the cumulative sunlit and shaded `deltai`, `dltaisen`, `dltaisens` and `pfeuilverte` at the plant level that is not compatible with their new computation anymore, because they already are weighted for plant level. For example, we computed `dltaisens` :

So now it is simply computed as:

Figure 8.13 shows the results given by the modified STICS, which finally gives a `lai` and a `mafeuilleverte` equal to 0 at the end of the cycle.

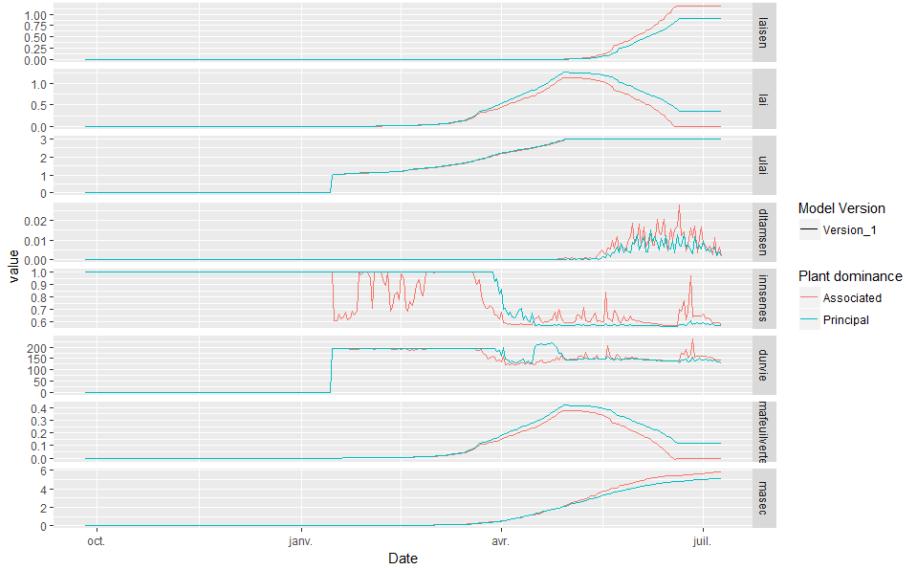


Figure 8.11: Simulation of a Wheat-Wheat self-intercrop using STICS showing an issue in the simulation of the lai at the end of the crop cycle

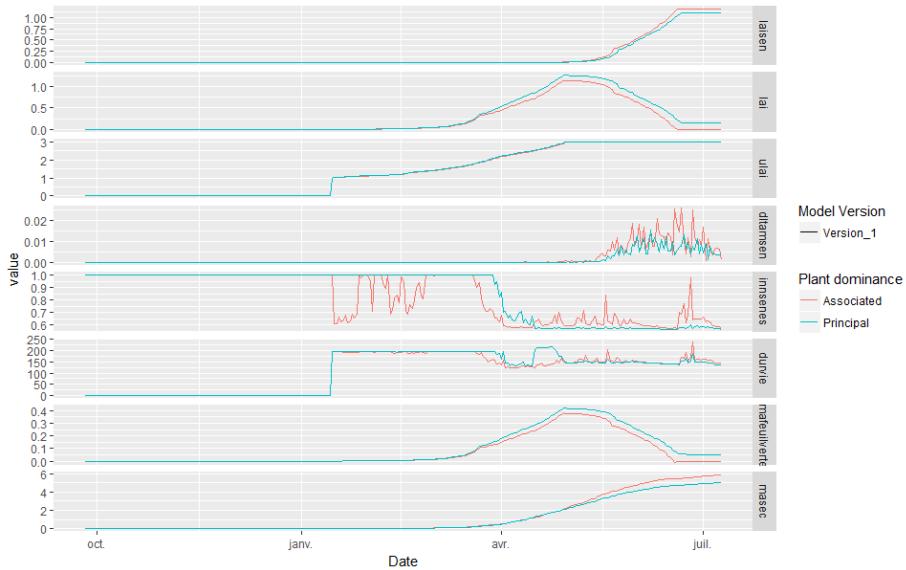


Figure 8.12: Partially fixed issue in the simulation of the lai at the end of the crop cycle

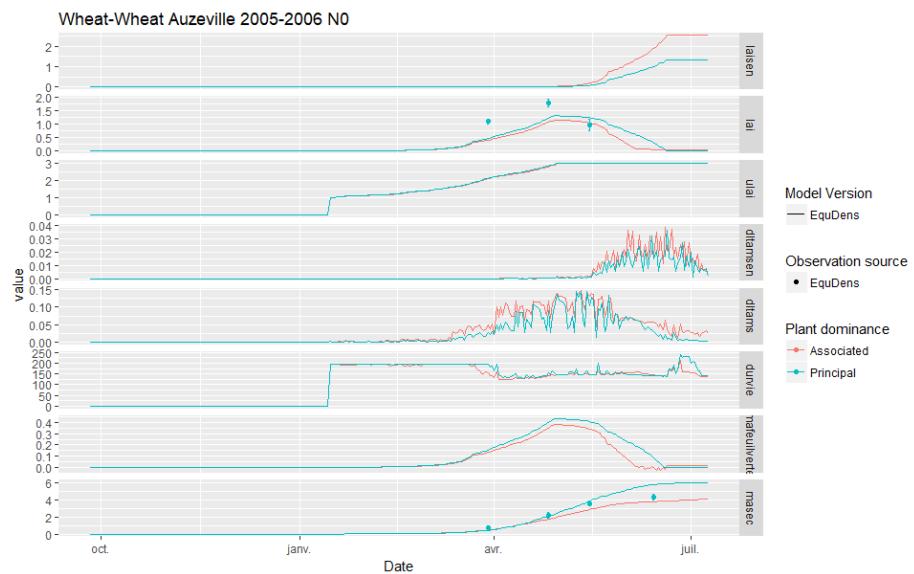


Figure 8.13: Fixed issue in the simulation of the masec and lai at the end of the crop cycle

Chapter 9

Issue on FaPAR greater than one

9.1 Introduction

The fraction of absorbed PAR (`fapar`) should have values between 0 (no absorption) and 1 (full absorption). The STICS model computed values greater than 1 sometimes (Figure 9.1), which is problematic.

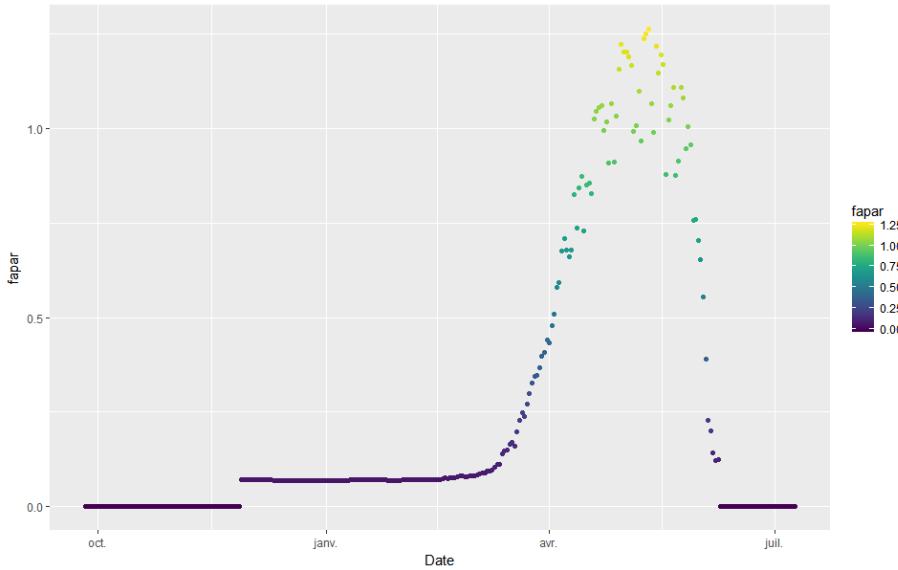


Figure 9.1: Leaf Area Index in a self-intercrop of Wheat-Wheat

9.2 The solution to the problem

The **fapar** is computed in the **raytrans** subroutine, which only uses the PAR transmitted by the dominant plant to a plane above the dominated plant components. This PAR has lower values compared to the atmospheric PAR (*i.e.* from the sky), leading to greater **fapar**. We propose to recompute the **fapar** after the **biomaer** subroutine to use the atmospheric PAR instead, which is what the user would expect.

The new fapar is now computed as $p(i)\%fapar(ens) = p(i)\%raint(ens) / (\text{trg_bak} * \text{pg}\%\text{P_parsurrg})$, and gives better results (Figure 9.2).

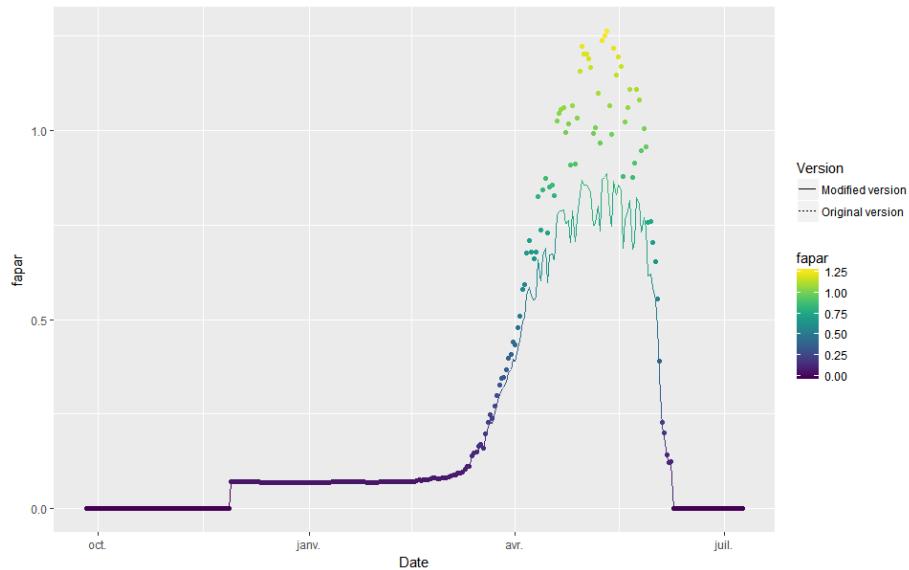


Figure 9.2: Leaf Area Index in a self-intercrop of Wheat-Wheat

Chapter 10

Computing plant height

10.1 Introduction

Plant height is not very well simulated for the moment. It is computated using the plant width, which depends itself on the plant lai ($\text{lai} + \text{eai} + \text{laisen}$). Please see Chapter 2 for more details. Figure 10.1 shows the simulated height of wheat in Auzeville, and the variables from which it is computed.

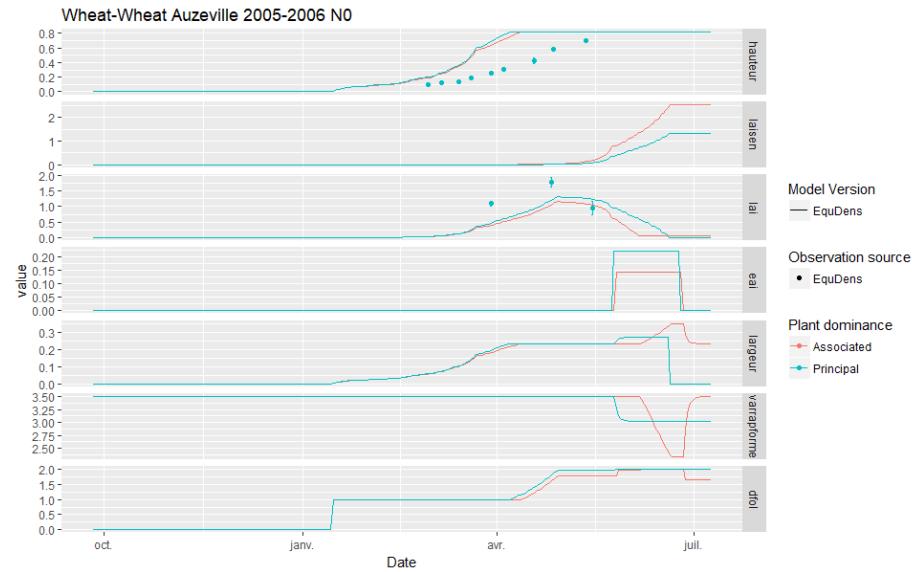
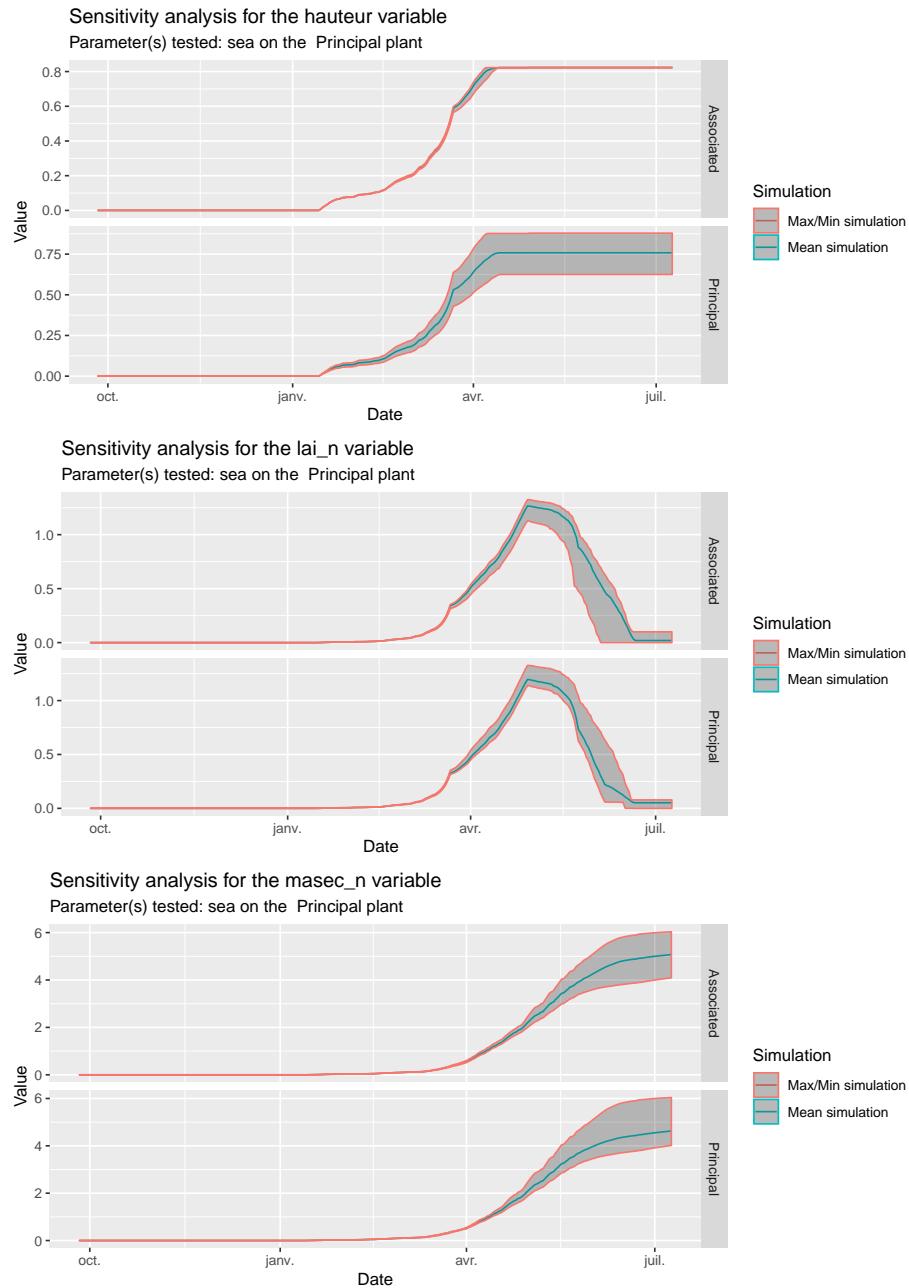


Figure 10.1: Simulated height compared to the observed values, and variables from which it is computed from, in a self-intercrop of Wheat-Wheat

The plant `height` is important in the case of intercrops because it has high impact on the light interception of both species. Indeed, a sensitivity analysis was performed using the `sticRs` package on the parameter that links the `lai` to the `height`: `P_rapforme`. The results are shown below (see `html` version of this book for interactivity):



As we can see on the above plots, as the plant **height** is affected, its **lai** is also impacted afterwards. We can also see that no values of **P_rapforme** gave satisfactory plant **height**, so maybe the link between the plant **lai** and the plant **height** is not as straightforward as previously thought, and maybe there

is another variable that could be better correlated to it.

10.2 Link between height and lai and dry mass

Using the observations from Auzeville, we plotted the evolution of the plant **height** through time along the **lai** and the **masec** (*i.e.* dry mass) to see if and how they are correlated. The Figure 10.2 shows this evolution with standardized and linearly interpolated observed values for all variables to make it easier to compare.

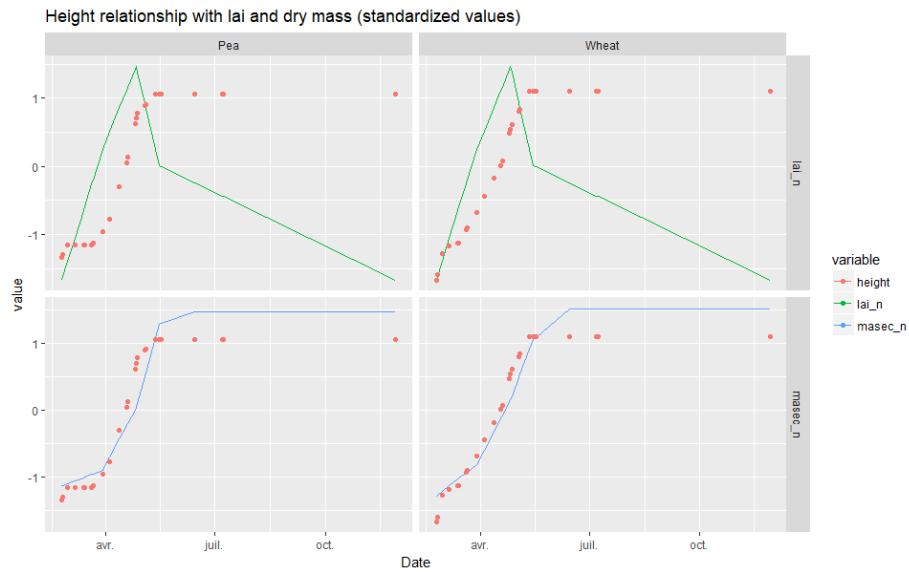


Figure 10.2: Observed plant height compared to the lai and dry mass for both wheat and pea in sole crop.

The Figure 10.2 shows that the plant **height** is better correlated to the **masec** than to the **lai**. We propose to either use the **masec** or **somcour** to compute it rather than the **lai**.

10.3 Computing plant height

10.3.1 Observations

The plant **height** could be computed by either using the plant dry mass (**masec**) or more simply the development stage, derived from cumulated degree days (**somcour**). The second method could be much simpler to parameterize from field, however, both plants experience the same microclimate in STICS for intercrops, so a plant growing dominated by another plant could have the same

cumulative temperature than a plant in sunlit conditions. Hence, this method would not consider the competition effect in interspecies for `height`, so it cannot be implemented as long as both plants have the same `somcour`.

To document further this phenomenon, Figure 10.3 shows that a Pea grown in intercrop with a wheat has a lower `height` than a Pea grown in solecrop, and that the dry mass is a good proxy to predict this behaviour.

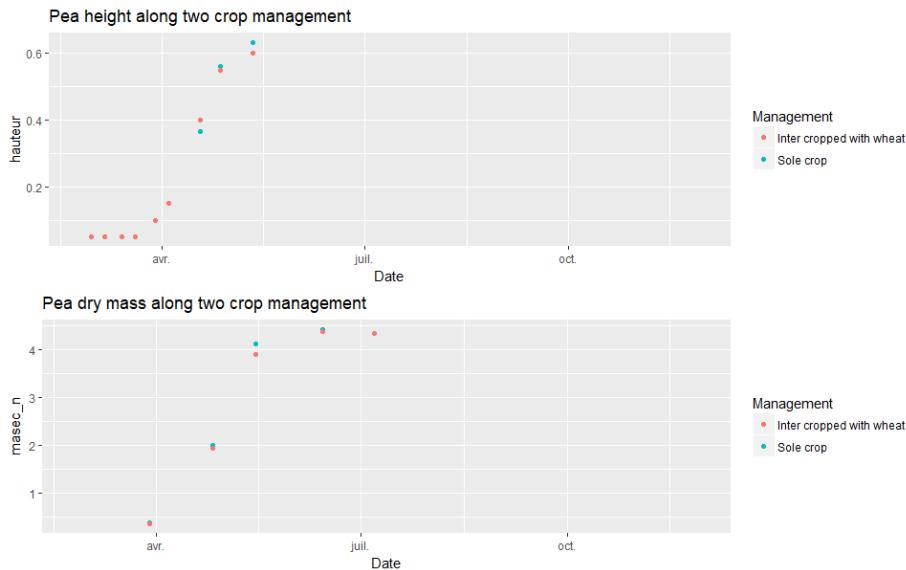


Figure 10.3: Observed Pea height and dry mass for a Pea grown in sole crop or in intercrop with wheat.

Hence, we will implement a computation of the plant `height` using the plant dry mass (`massec`).

10.3.2 Stresses

Water and nitrogen stresses can both influence the `height` of a plant. Figure 10.4 shows that the maximum `height` is reduced for the wheat grown under nitrogen stress (*i.e.* no nitrogen fertilization). However, the `height` and `massec` relationship are very close for both management, only the dry mass is higher for fertilized wheat, which gives a higher `height` in the end.

Consequently, height is indeed influenced by nitrogen stress, but it appears from our data that it is only through the `height~massec_n` relationship. Consequently, the `height` computation shouldn't be corrected by any nitrogen indices, because this stress is already accounted for in the dry mass computation.

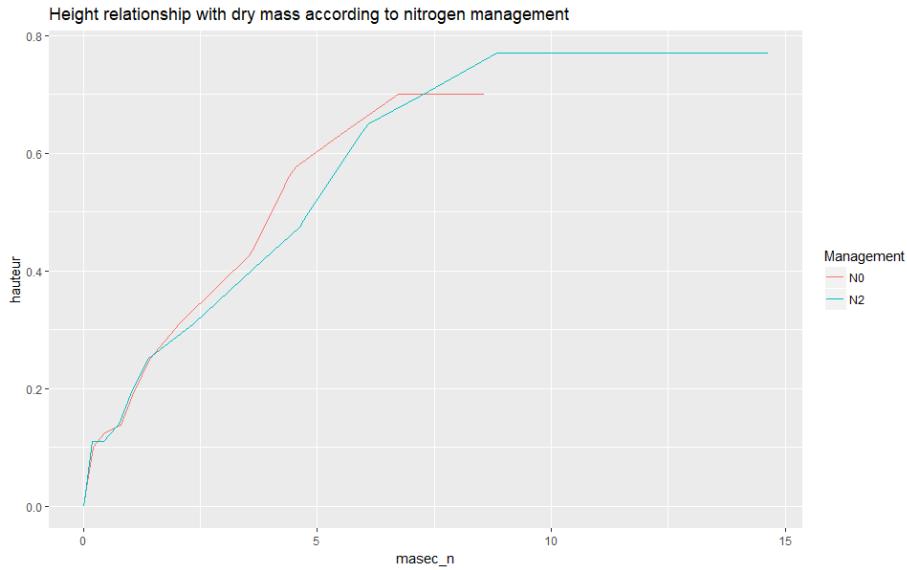


Figure 10.4: Observed relationship between height and dry mass for a wheat grown in sole crop.

10.3.3 Computation

The new computation is implemented in STICS as a new subroutine (`formplante_mas`) that is called instead of `formplante` if the newly defined `P_code_shape` parameter is equal to 2. This subroutine is largely based on the previous one, conserving all side effects of `P_hautmax`, `P_hautmaxtec`, `P_largtec` and `originehaut` but add a computation of the plant height using an allometric relationship with the dry mass as follow :

$$\text{hauteur} = P_{\text{hautbase}} + P_{\text{hautK}} \cdot \text{masec}^{P_{\text{hautA}}}$$

The `P_hautbase` parameter is a parameter from `formplante`, and is used as the intercept of the relationship, *i.e* the base height when the dry mass is equal to 0. `P_hautK` is a new parameter that defines the constant of proportionality that reflects the importance of the relationship, and `P_hautA` the scaling exponent, or the allometric coefficient. Allometric relationships are simple yet powerfull laws used to describe diverse relationships for wide range of situations. Further documentation can be found on the wikipedia webpage.

10.3.4 Results

Two simulations of a wheat in sole crop and a pea in sole crop were run using STICS to asses the potential of the new computation for height modelling, and the outputs are displayed in Figure 10.5.

The results show that using `masec` instead of the `lai` to model plant height

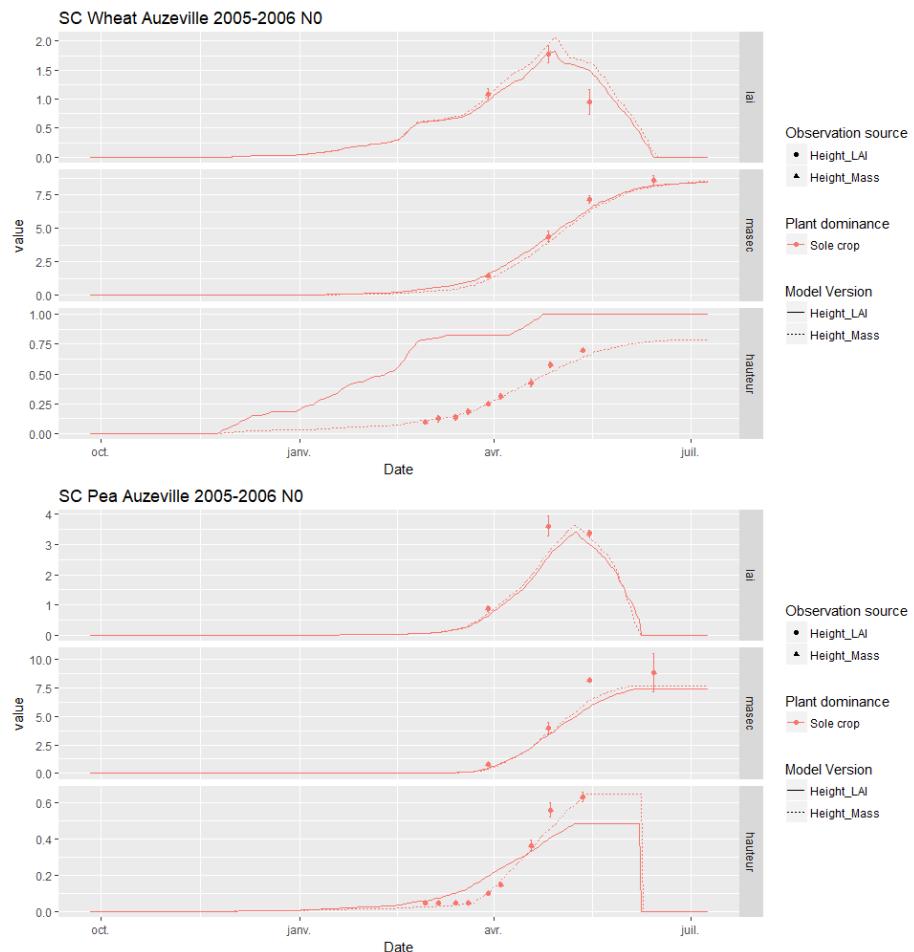


Figure 10.5: STICS simulation output for a simulation on two sole crops: wheat and pea

gives better results. The model will now be used with this formalism.

10.3.5 Stop height~masec link at stage X

The height~masec link can stop existing as soon as a plant reach a certain development stage. For example, wheat stops growing in height from the moment it is flowering.

Hence, a new parameter (`P_stage_const_height`) was added to have the choice to break this link at any developing stage from this list :

- “lax”: maximum leaf area index, end of leaf growth.
- “sen”: start of leaf senescence
- “flo”: flowering
- “drp”: start of grain filling
- “mat”: physiological maturity
- “rec”: harvest

Figure 10.6 shows the new results, with `P_stage_const_height= flo` for both plants.

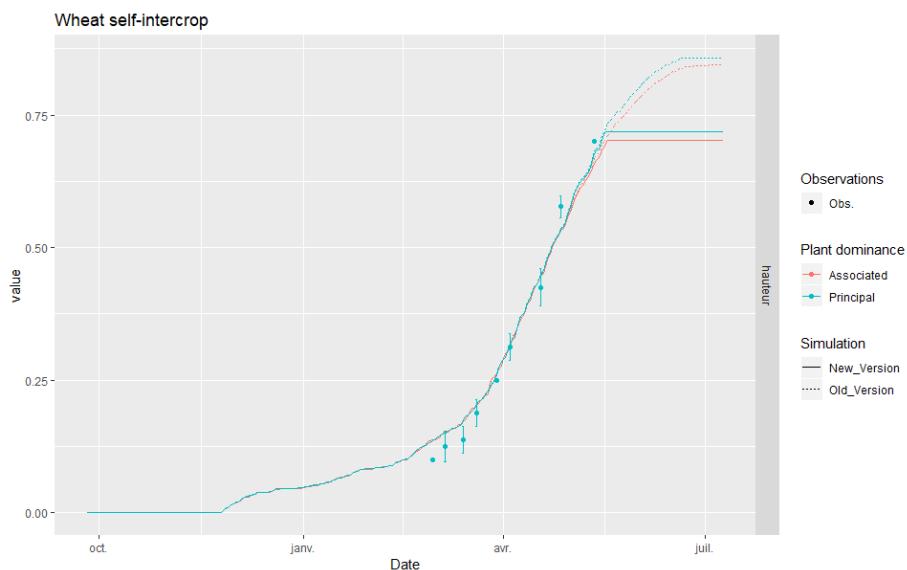


Figure 10.6: STICS simulation output for a simulation on two sole crops: wheat and pea

Chapter 11

Developpment stages and LAI dynamic

11.1 Introduction

Bedoussac (2009) showed the different development stages for the crops, which are reproduced in Figure 11.1).

The STICS simulations are not in adequation with these results for the moment.

11.2 Checking the ilev

The date of emergence is set using the `ilev` parameter. This parameter had a value of 380 in the STICS input files.

The Fig. 3 of Bedoussac (2009) shows that `ilev` should occur 151 cumulated Celsius degrees per day after sewing, but a simple computation showed us that this conditions is met on the 329th day already, and not on the 380th as previously parameterized (see computation below).

Figure 11.2 and Figure ?? shows a comparison of some STICS outputs for a wheat self-intercrop and a wheat sole crop respectively, with `ilev` set either at 380 like before or corrected at 329. The code used to make this output is as follows:

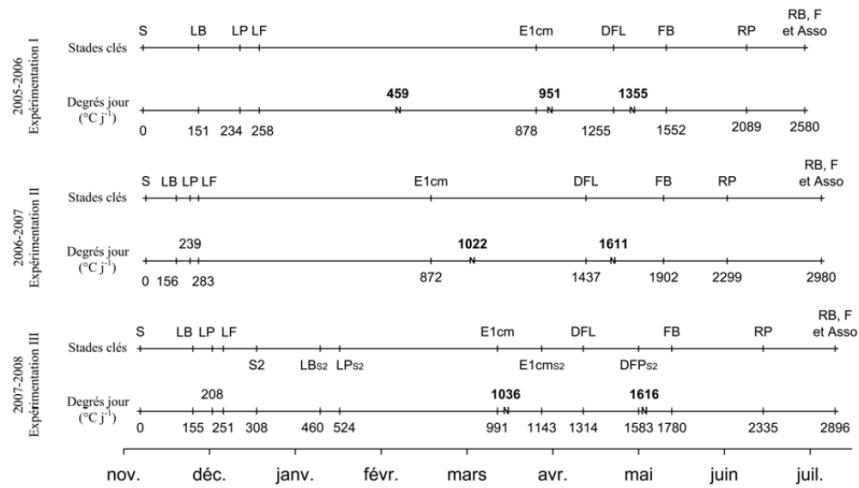


Figure 3 Stades clés et apports d'azote (N) lors des trois années d'expérimentation (2005-2006, 2006-2007 et 2007-2008) représentés sur une échelle calendaire couplée à une échelle de temps thermique (base 0 °C) : semis (S), semis décalé (S2), levée blé dur (LB), levée pois d'hiver (LP), levée féverole d'hiver (LF), stade épis 1cm du blé dur (E1cm, Zadoks 30), début floraison légumineuse (DFL), floraison blé dur (FB, Zadoks 69), récolte du pois d'hiver à maturité physiologique (RP) et récolte du blé dur (RB), de la féverole d'hiver (F) et des associations (Asso) à maturité physiologique du blé dur.

Figure 11.1: Development stages as measured in Auzeville

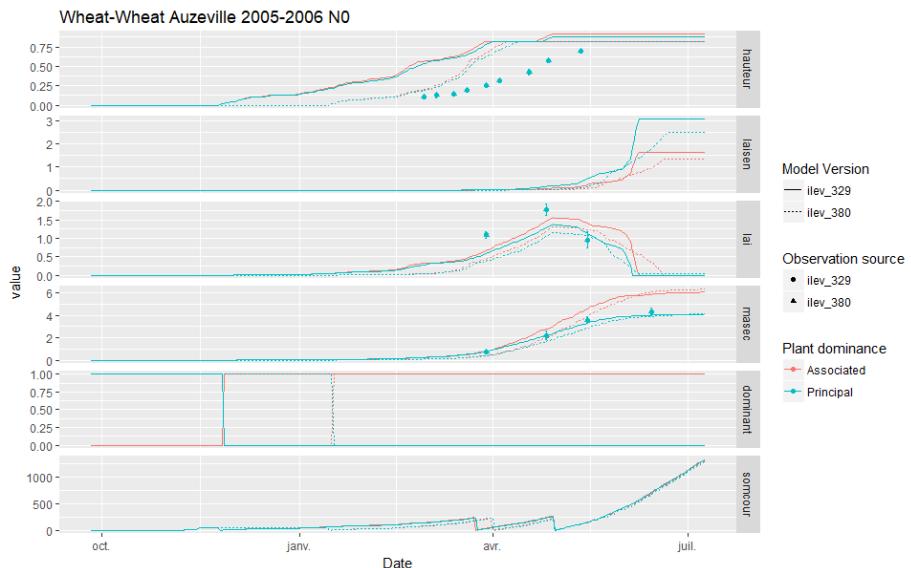
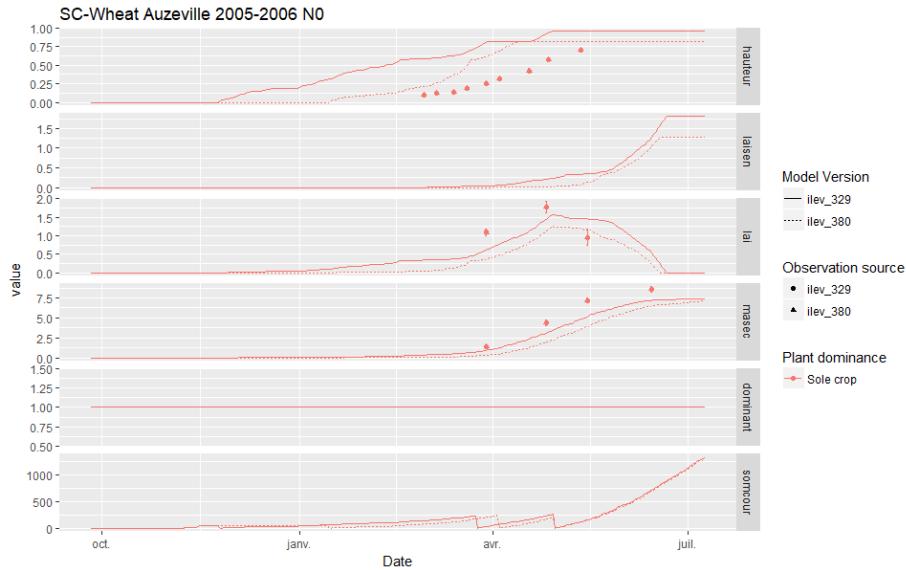


Figure 11.2: STICS outputs comparison with ilev= 380 and corrected to 329 on a wheat-wheat intercropping simulation experiment (Auzeville 2005-2006)



The new parameter values better describe the observations for the development stages, and give simulated `lai` that are closer to the observations. However, the plant height is now increasing way too soon compared to the observations.

These simulations were made using a version of STICS that uses the `lai` as a proxy for plant height. The model will soon be modified to compute it using other variables.

11.3 Checking laicomp

The `laicomp` parameter is used to stop the exponential rate of `lai` growth, and is set to 0.304 for the moment. This value could be too low compared to actual conditions, so we increased the value to 0.6 and compared the outputs in Figure 11.4).

The new parameterization of `laicomp` gives simulated `lai` that are closer to the observations when coupling with the new parameterization of `ilev`. These two new values (`ilev= 329` and `laicomp= 0.6`) will be kept for future simulations.

11.4 Comparing simulated and observed lai and masec in self-IC

The simulated `masec` for each plant of wheat or pea in self-intercrop (*i.e* half the density) was rightfully compared to half the plot-scale observations. However, the `lai` was wrongly compared to the full observation value, whereas it should be compared to half its value likewise, because the full value has to be compared to the total `lai`, that is computed by summing both plants `lai`.

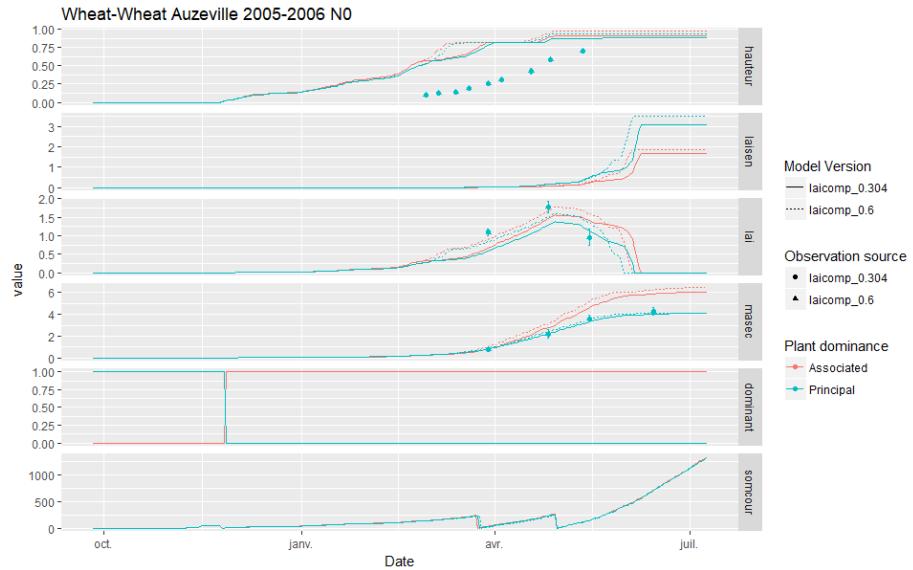


Figure 11.3: STICS outputs comparison with $\text{laicomp} = 0.304$ and corrected to 0.6 on a wheat-wheat intercropping simulation experiment (Auzeville 2005-2006)

The observations were then divided by two to fit the `lai` simulations outputs that correspond to each plant, so to half values. The new outputs are shown in Figure ??). The modification tells us that the plant `lai` is overestimated in self-intercrop.

11.5 Checking durvieI

The lifespan of the early leaves is expressed as a proportion of the lifespan of the late leaves, and is called `durvieuI`. This parameter help control the steepness of the decreasing part of the `lai` through leaf mortality (*i.e.* senescence).

It appears that the leaf senescence is not well simulated in the wheat sole crop, because the `lai` do not drop as fast as in the observations. We propose to decrease the value of `durvieI` from 0.8 to 0.6. A comparison of the effect of changing this value is shown in Figure 11.5) for sole crop and for 11.6) self-intercrop.

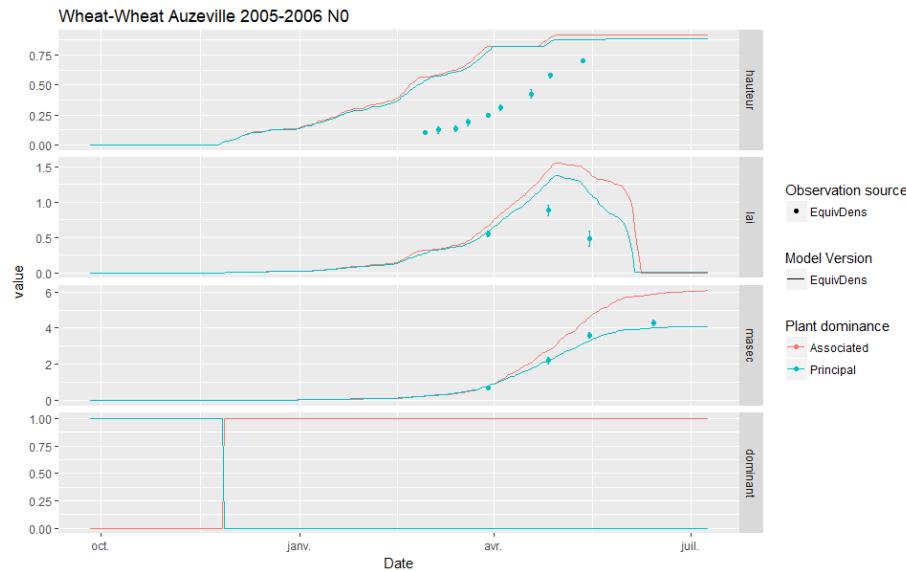


Figure 11.4: STICS outputs comparison with half lai values on a wheat-wheat intercropping simulation experiment (Auzeville 2005-2006)

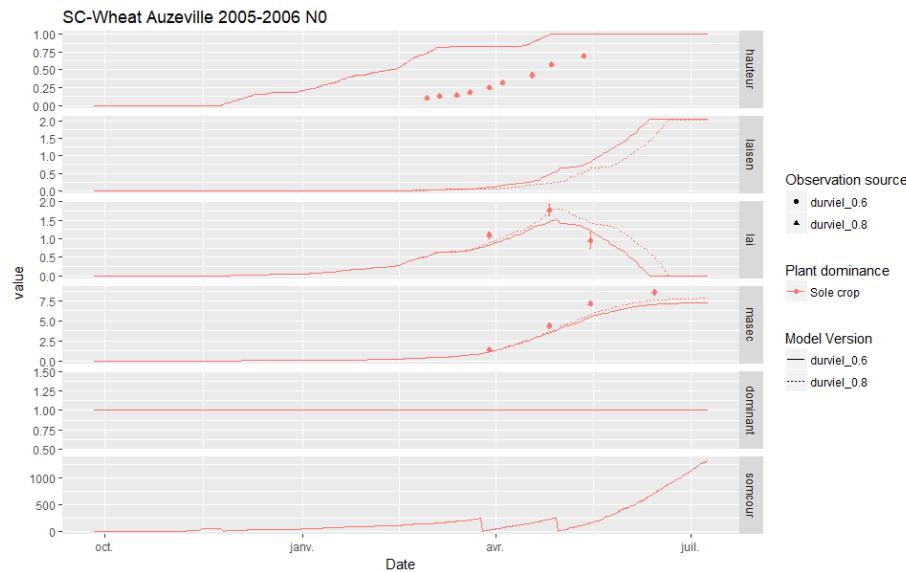


Figure 11.5: STICS outputs comparison with $\text{durvieI} = 0.8$ compared to 0.6 on a wheat sole crop simulation experiment (Auzeville 2005-2006)

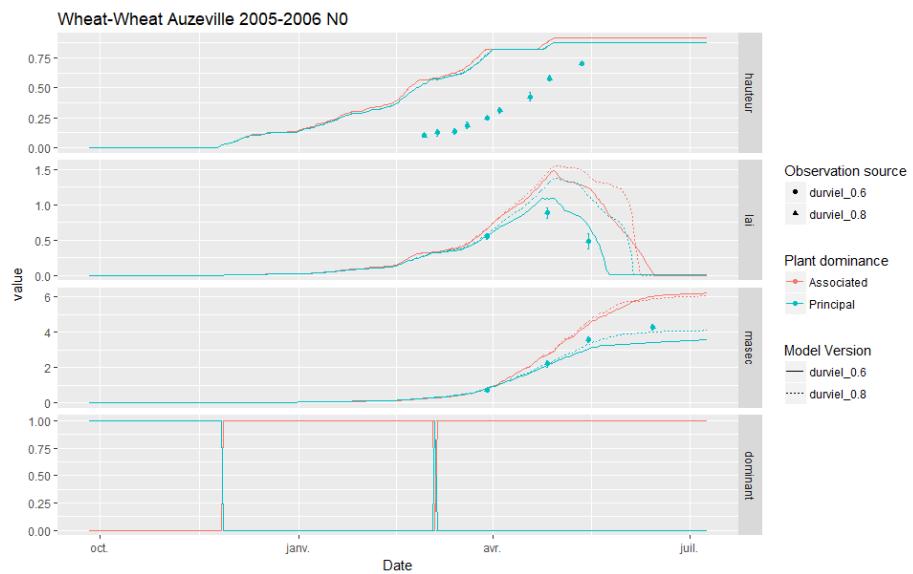


Figure 11.6: STICS outputs comparison with $\text{durvieI} = 0.8$ compared to 0.6 on a wheat-wheat intercropping simulation experiment (Auzerville 2005-2006)

Chapter 12

The LAI in intercropping is too high

12.1 Introduction

The LAI is too high compared to observations for intercropping. This is especially visible while making a self-intercrop (*i.e* wheat-wheat or pea-pea) simulation compared to a sole crop, as shown in Figure 12.1.

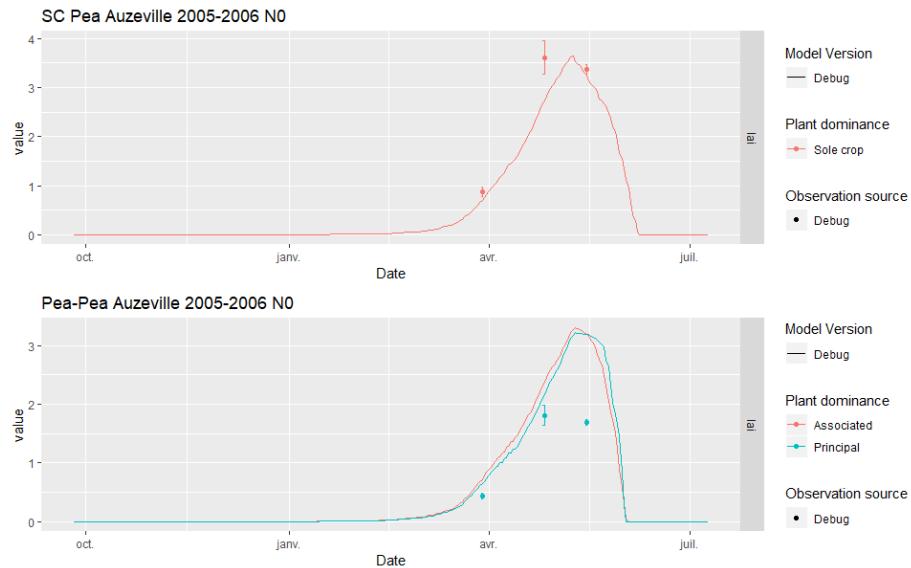


Figure 12.1: Leaf Area Index in a Pea sole crop and in a self-intercrop of Pea-Pea

This issue probably rise from the computation of `deltai`, and the main hypothesis are:

- `P_dlaimax` is not well parameterized.
- `ulai` is not well simulated in intercrop.
- `efdensite` and `densite` are not well simulated. The value of the density used in the `deltai` equation in the self-intercrop should be half the value of the sole crop.

The first hypothesis is rejected because the `lai` simulation is close to the measurements in the sole crop, so it should perform equally in intercrop because `P_dlaimax` should have the same value in sole crop and in intercrop.

12.2 Investigating the `ulai` hypothesis

Figure 12.2 shows that both the sole crop (SC) and self-intercrop (IC) simulations yield approximately the same `ulai` throughout the crop cycle. This discard the second hypothesis.

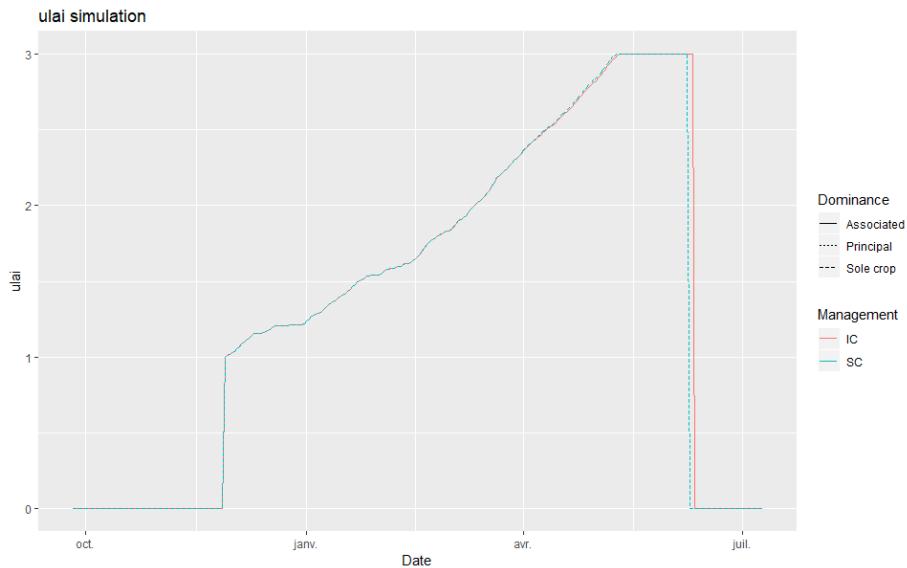


Figure 12.2: Simulated `ulai` in a self-intercrop of Pea-Pea compared to a sole crop of Pea

12.3 Investigating the density hypothesis

Figure 12.3 shows that the density in the Pea self-intercropping is higher than expected. The issue probably come from this.

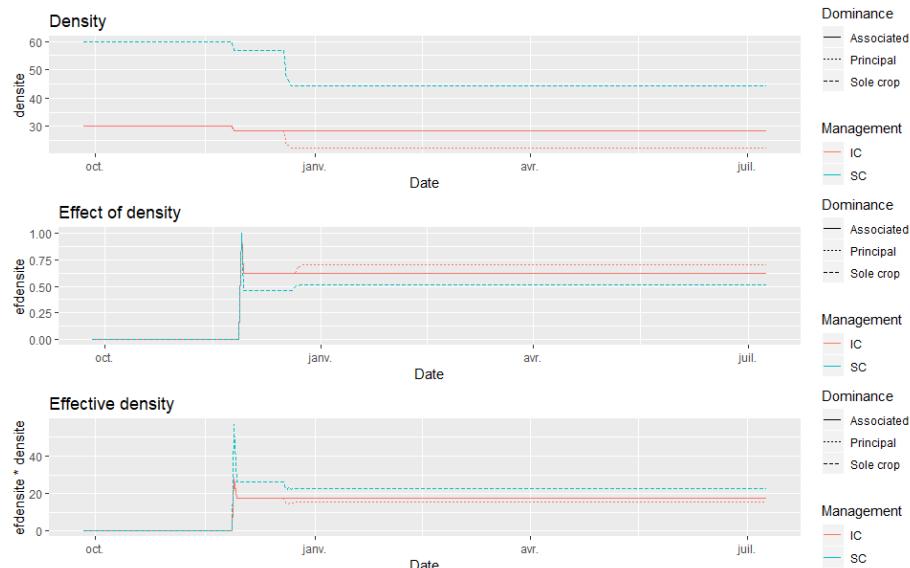


Figure 12.3: Simulated densite, effdensite and the resulting density from their multiplication in a self-intercrop of Pea-Pea compared to a sole crop of Pea

Furthermore, we can see that the plant density is reduced twice for both Pea in SC and for the dominant plant in IC. The first is because some plants are lost in the first growing stage (`densite_lev`), and the second because of frost damage. However, the frost does not affect the associated plant density in IC. This does not reflect reality and should be changed. The error came from a piece of code in the `senescen` subroutine where the computation differed between dominant and dominated plants.

The code was changed into only one condition for both plants:

SHA: f525211036a7b4b683252093187298324f4cd80b

The code modification lead to better results, with an effect of frost that is equal between dominant and dominated plants as seen in Figure 12.4.

However, an evaluation of the model on Auzeville data showed that the problem of a high `lai` persisted, so the issue did not come from this bug alone. More investigation showed that `effdensite` was lower in self-intercrops compared to sole crop simulations, because the equivalent density of both dominant and dominated plants were set equal to the plant density or higher for the dominated plant if the difference in height is large (see Chapter 5). However, the plant

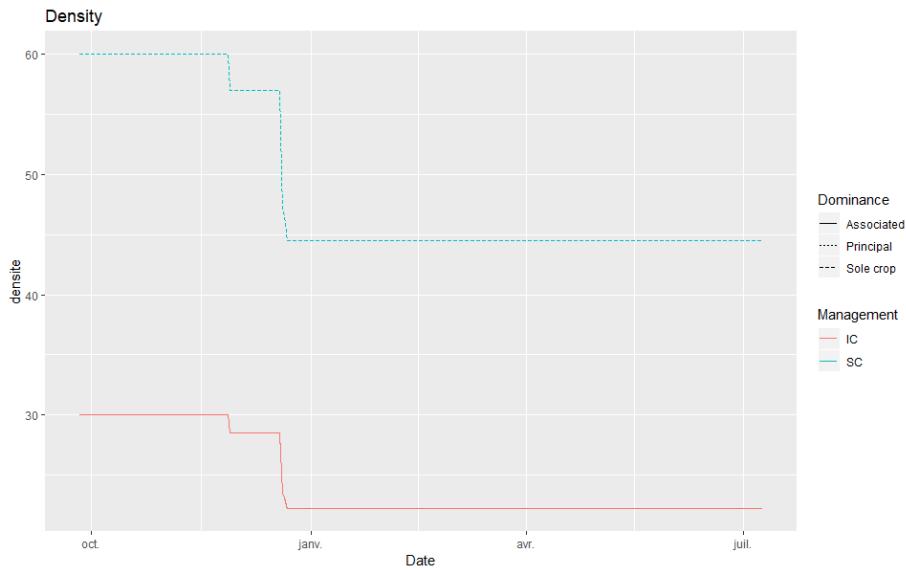


Figure 12.4: Simulated density in a self-intercrop of Pea-Pea compared to a sole crop of Pea. NB: Both plants density have the same value in the self-intercrop (half the value of sole crop)

equivalent density should refer to the plant density as if it were in sole crop to have the same density effect while using sole crop parameters for `efdensite`, and not to be equal to the half-density, as it is for the moment.

The equivalent density computation was then modified to match a density equivalent to a sole crop density, that is to say, twice the value of the sowing density in intercrop, *i.e.* the same density as if the other plant was from the same species.

As expected, the code modification led to the simulation of the same `efdensite` in SC and in IC for both plants, as seen in Figure 12.5.

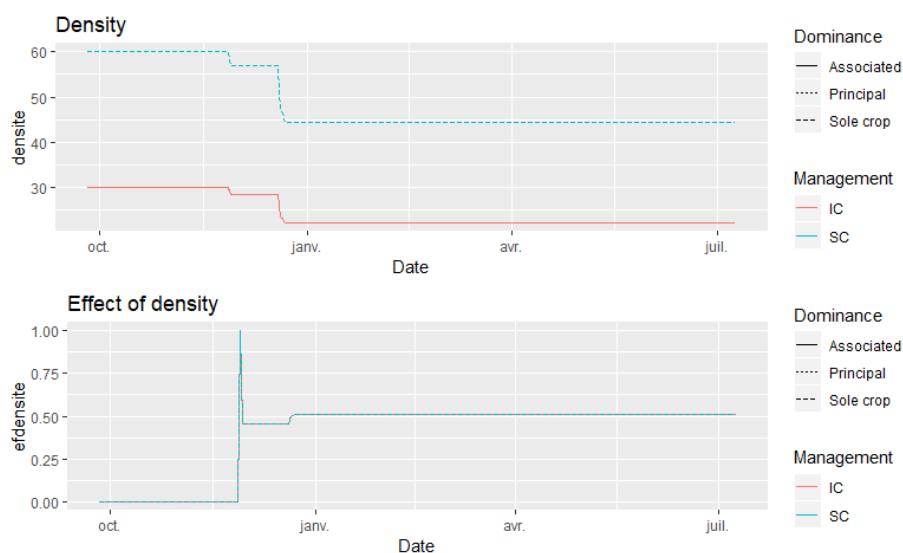


Figure 12.5: Simulated density and density effect in a self-intercrop of Pea-Pea compared to a sole crop of Pea. NB: Both plants density have the same value in the self-intercrop (half the value of sole crop), all plants/managements have the same value of edfesnites.

Chapter 13

Miscellaneous

13.1 Introduction

This chapter keep trace of several little issues resolved rapidly and with no high significance.

13.2 Extra code from transrad

Some pointless code was present in the `kgeom` subroutine in the `transrad.f90` file. The code assignated values to the `kg` variable twice in a row, and then `kg` was always recomputed anyway:

We propose to remove the `if...endif` code bloc. The effect of the removal of this part of code was tested against the Auzeville dataset, and as expected no change was observed in the outputs.

13.3 Sowing density in Auzeville

The sowing density in Auzeville was set to 168 for the wheat at half-density in intercropping. We change that to the real value of 140 instead.

13.4 Energy balance

A new variable was added in STICS to keep track of the plant layers net radiation to make an energy balance. This variable is called `rnet_plant`, and is computed in the `shutwall` subroutine. It is computed using the radiation intercepted by each plant layer. The net radiation is computed as:

$$rnet = rnet_{plant_1} + rnet_{plant_2} + rnet_{soil}$$

Now it can be checked ad hoc using the new outputs, such as shown in Figure 13.1 that shows as expected that the net radiation output from STICS (*i.e.* `rnet`) is equal to the sum of the net radiation of the dominant (`p(1)%rnet_plant`) and dominated (`p(2)%rnet_plant`) plant as well as the net radiation from the soil (`rnets`).

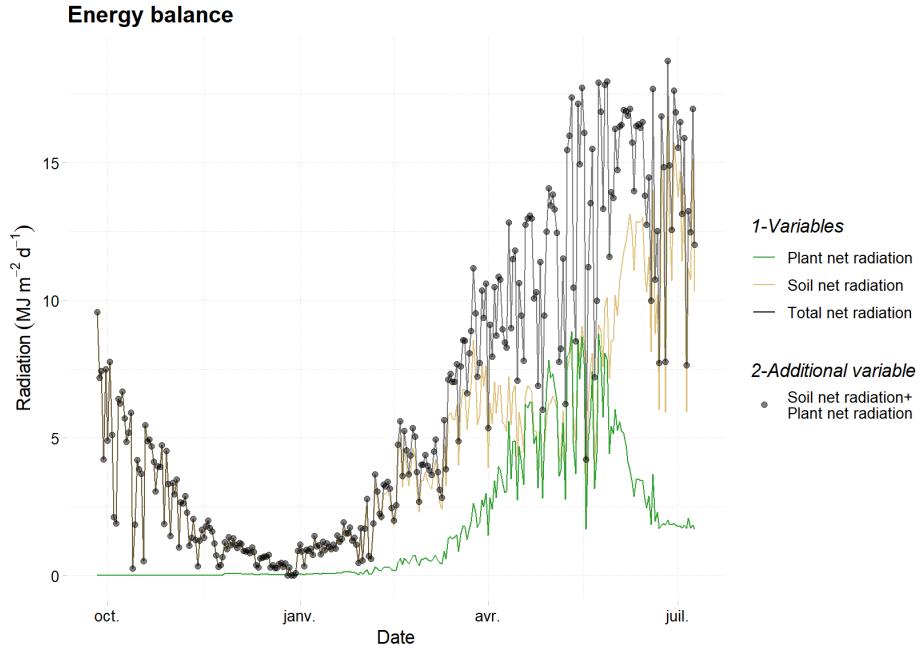


Figure 13.1: Net radiation computed in the model and out of the model

13.5 Positive raint with zero LAI+EAI

raint was not equal to 0 (and so `fapar`) when the `lai+eai` was equal to 0, instead it remained at the last computed value when `lai+eai` was >0 . A little modification of the model was made to fix the bug as seen in Figure 13.2. Note that a little modification on the computation of leaf senescence fix also a bug for the `lai` at the end (see `e027b8efd93fe77aa7ac7242ce3496589003eb41`).

13.6 Qfix decrease

S. Munz discovered that Qfix decreased abruptly in some years in intercropping. This effect should not be possible because Qfix is the cumulative fixed nitrogen. The issue was corrected in commit `4f9d92595c870d9b08f3f0b986cbb321630ad031`. The results are shown in figure 2.1.

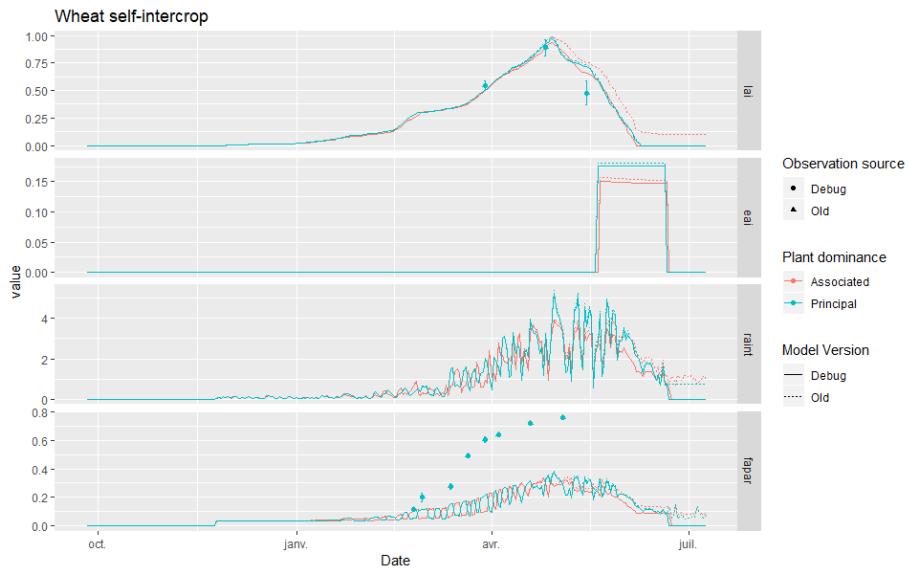


Figure 13.2: Bug fix for raint when lai+eai=0

13.7 eddensite_rac

The effect of the density on the roots (`eddensite_rac`) was computed using the current plant density, but for intercrops it should use `densiteequiv` because the parameters are fitted using a sole crop equivalent density. This computation was fixed in commit `ba45c6c73d1bb9a081393e84f648a9c90bf47428`.

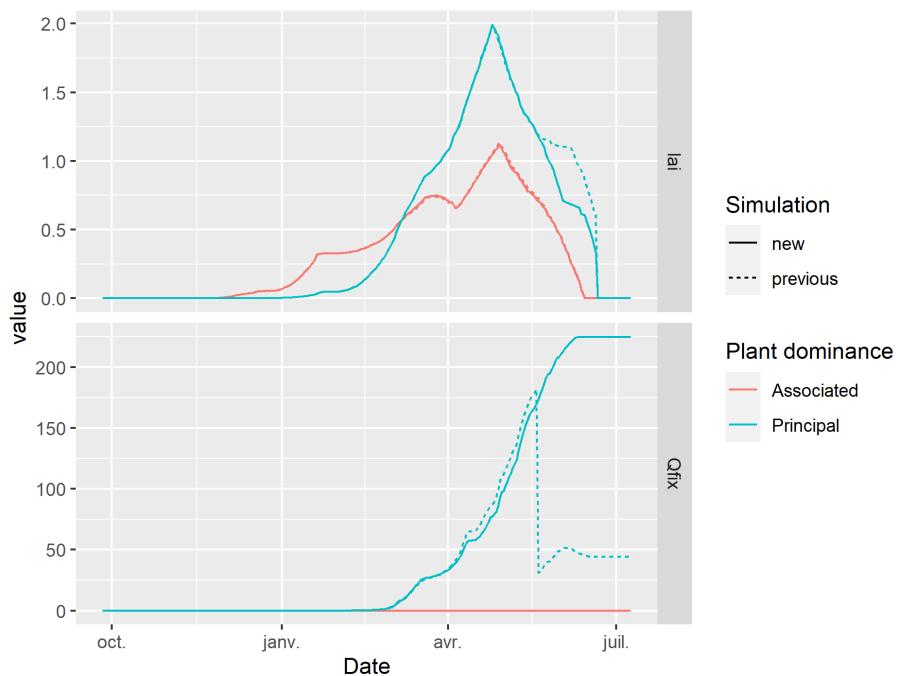


Figure 13.3: Issue with Qfix decreasing abruptly for Pea. Principal plant: Pea, Dominated: Wheat. Simulation for 2007, low nitrogen (N0)

Chapter 14

First summary

14.1 Introduction

A lot of code and parameterization has changed since the beginning of this book. Here is a little summary of all changes, starting with the outputs we have at this stage, followed with a step-by-step parameterization help for the new formalism.

14.2 Model outputs

The model was tested on a series of experimentations on Wheat and Pea from Auzeville. It consists on a set of observations made on either sole crops or intercrops, and the model was run on each case to evaluate its simulation outputs. The model was also tested to simulate a sole crop as an intercrop with both plants from the same species and variety.

The results of this evaluation are available on this separate page. You can consider this page as a snapshot of the evaluation of the model version used as : 31 mars, 2020. This version of the model can be retrieved from my GIT repository using this SHA: `c9638aafb924949b630d273608d2b3c45369cd0c`.

14.3 Model parameterization

14.3.1 Height

The height can be computed using two different paradigms:

- The previous one, that uses the plant `lai` and foliar density (which depends also on the `lai`) to compute the plant width, and then uses a ratio to retrieve the plant height.

- The new one that simply links the plant dry mass (`masec(n)`) to its height using an allometric relationship.

The computation is chosen using the `P_code_shape`. If `P_code_shape` is equal to 2, the model will use the second one (using `masec`), and for any other value, it will use the first one.

The allometric equation is simply defined as:

$$\text{hauteur} = P_{\text{hautbase}} + P_{\text{hautK}} \cdot \text{masec}^{P_{\text{hautA}}}$$

So it has two new parameters: `P_hautK` and `P_hautA`. This equation is well known across a wide range of fields, can describe a large panel of relationships, is robust, and is quite simple to parameterize. In the case of STICS, the user should use sole crop data only for parameterization, because the hauteur~masec relationship is hard-coded for this situation. In the case of intercrop, the model will adapt the computation accordingly.

For our case experiment in Auzeville, these parameters were fitted thanks to the `nls` function in R using observations for each species independently.

At first, we were afraid that the relationship would differ according to the nitrogen management, but only the dry mass was impacted by the nitrogen treatments, not the height~masec relationship. However, the relationship changed a little between sole crops and intercrops, but was still robust enough to use it as is as shown in the results from this separate page, also shown in the first paragraph.

14.3.2 Equivalent density

The equivalent density is the density the plant would have if the other plant was of the same species, *i.e.* if the field was a sole crop. This equivalent density is used to compute the plant `LAI`. It is computed in three different ways in STICS, that are all described in Chapter 5. The third computation is re-explained here for ease:

The third formalism is chosen by setting `P_code_calc_equi_density=3`. It is an improved computation to define the equivalent density as a function of the height difference between the plants, and using different parameters for intraspecific and interspecific competition, with simpler meanings. This formalism gives an equivalent density equal to the double of the true density to match a density equivalent to a sole crop, and increases this equivalent density until a given threshold (`hauteur_threshold`), after which a progressive increase in the equivalent density happen until the maximum allowed. In this computation, the slope is computed by the model, and the equivalent density is computed as follows:

$$\text{density}_{\text{Equivalent}} = \begin{cases} \Delta_{\text{height}} < \text{hauteur}_{\text{threshold}} & \text{density}_{p2} \\ \Delta_{\text{height}} > \text{hauteur}_{\text{threshold}} & b + \Delta_{\text{height}} \cdot \text{slope} \\ \Delta_{\text{height}} > \text{hauteur}_{\text{threshold}_2} & \text{Max}_{\text{equDens}} \end{cases}$$

with $slope = \frac{Max_{equDens}}{hauteur_{threshold_2} - hauteur_{threshold}}$.

Hence, this formalism include three parameters (named here as used in the model):

- $P_hauteur_threshold$: the difference in height below which no competition is occurring
- $P_hauteur_threshold_2$: the difference in height below which a progressive competition is occurring
- P_eqdens_max : the maximum density allowed in intercrop, *i.e.* generally the sole crop density.

These three parameters are quite simple to parameterize also. The first one should be set to represent the moment the height difference between two plants start increasing competition. We set this parameter to 0.2 meters, because we considered no competition increase before a difference of 20 centimeters between both plants. The second threshold was set to 0.8 meters, because we considered the competition would increase linearly until this difference in height, reaching its maximum at this point. The **Max_eqdens** was set to 240 for wheat and 60 for Pea. These densities correspond to a full density in a sole crop.

Note: the increased equivalent density with height or whatever formalism used is only there to represent competitions comming from other processes than light interception, as it is already accounted for by the light interception subroutines. The model must be well parameterized for the light interception first, using no increase in equivalent density first (*e.g.* by setting a high $P_hauteur_threshold$), and then the equivalent density formalism can be used to better match the plants competition if needed.

Chapter 15

Interception fix

15.1 Introduction

When simulating a self-intercrop, *i.e.* a monocrop simulated as if it was an intercrop, both plants outputs were quite different.

This was arising from the computation of the light competition between the two plants: one was considered dominant over the other one, so the received by the dominated plant was only the light transmitted by the other.

This computation needed to change, so we included two cases to compute the light incident to the plants:

1. The dominant plant is considered above the dominated plant (dominancy case): $p(i-1)\%rsoleil$ represents the fraction of the radiation incoming right above the plant i , *i.e.* the light transmitted to the associated plant, for the sunlit part (`rombre` is used for the shaded one). So `trg_bak * p(i-1)\%rsoleil` is the actual amount of radiation transmitted to the sunlit part of the associated plant, and `trg_bak * p(i-1)\%rombre` is the actual amount of radiation transmitted to the shaded part of the associated plant.
2. There is no clear dominancy because both plants have approx. the same height (Delta height < `hauteur_threshold`): the `trg` is not transmitted from one plant to the other, but shared between both plants. In that case, the competition for light arises only through the LAI of each plant.

The new version of the model integrating this code can be retrieved from the repository using this SHA: `591cd8db5aeaf8db2cdb5b62008c54d04d833df8`.

15.2 Model parameterization

The model does not need any further parameterization because this new formalism uses one parameter that is already available: `hauteur_threshold`.

Chapter 16

Light competition effect

The competition between the plant species in intercropping was so much mixed between several computations that it became difficult to parameterize the models, or even to grasp the functioning of the model. We propose a brand new approach to clarify and simplify the computations, while including a new process in the model: the shoot elongation.

As explained in chapter 2, STICS considers two kinds of competitions for light: the intra-species competition, and the inter-species competition. The model computes a competition for light between both species using the light interception module, and a second competition using the plant density. Until now the density effect represented both intra and inter-species competition because it was computed using an “equivalent” density that increased with the difference in height between the plants (see 5). This equivalent plant density was difficult to explain, and difficult to parameterize because it was used to compute **efdensite** (the density effect) with the same equation than for a sole crop, where this density effect is only an intra-specific competition.

The model should simulate approximately the same results simulating a sole crop or the same crop twice in intercrop (we call it a self-intercrop). It was difficult make the model do it because the same equations were used for different purposes, so the parameterization of the model did not stand well.

We propose to remove the complicated computation of the equivalent density because the competition for light between the two plants is already considered in the light interception module. We argue that the density computation should only be used for the same purpose as for the sole crop (intra-specific competition) to correctly use the **adens** and **bdens** parameters that are parameterized for a sole crop. The equivalent density would be used only as a way to match the density in intercrop to the equivalent density it would have in the sole crop, i.e. densite^2 (densite : sowing density). The density effect (**efdensite**) would then only represent the intra-row competition, and the inter-row competition

would be computed using the light interception module.

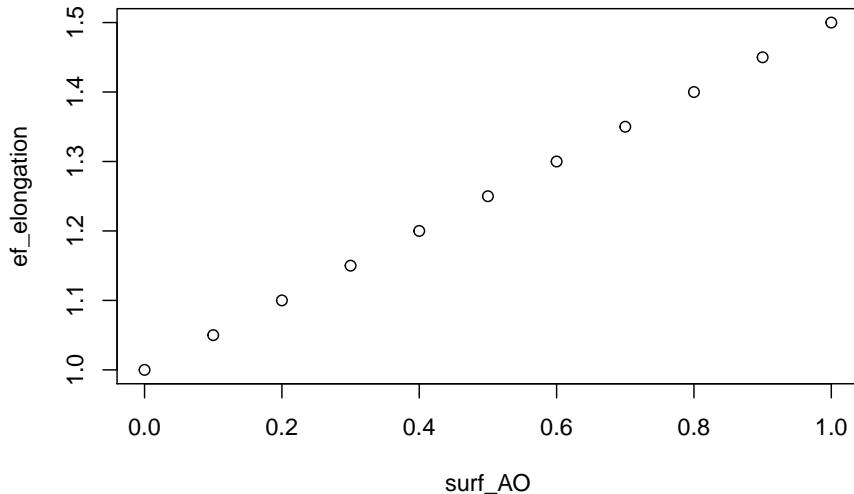
It would greatly simplify the processes involved in intercropping, reducing the number of parameters with a complex meaning : `hauteur_threshold2`, `eqdens_max1`, `eqdens_max2` and `code_calc_equi_density`. `> P_hauteur_threshold` is still needed to know when we consider the plants are starting to be in competition for light.

We also want to add another important process to the model: the elongation of the shoot with increasing shade. We propose to add a computation for the shoot elongation effect (called `ef_elongation`) that would be applied to increase the plant height with increasing shading. This computation adds only one parameter that represent the effect on plant height with the maximum shading. This parameter is called `P_elongation`.

The computation is as follow:

with `surf_AO` the relative surface of the plant that is shaded (0.0: full sun, 1.0: fully shaded), `P_elongation` the elongation effect when the plant is fully shaded (1.0 - n), and `ef_elongation` the resulting elongation effect that is applied to compute the plant height. `ef_elongation` linearly increases with increasing shading.

The following plot shows the elongation effect with `P_elongation= 1.5` for an increasingly shaded plant.



The elongation effect is then applied to the computation of the height as follows:

By modifying our previous approach, we hope to simplify the computation of the competition for light between plants, and to integrate a potentially essential process that is shoot elongation. We can summarize the new approach as follows:

- The intra-species competition is computed as it is in sole-crops using **efdensite**
- The inter-species competition is computed using the light interception module
- The plant elongation is also computed using the light interception module with the relative shaded surface of the plant as a proxy for light quality.