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

R. Vezy

2018-06-21

Contents

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.

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 2, a light extinction coefficient of 0.2. 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:

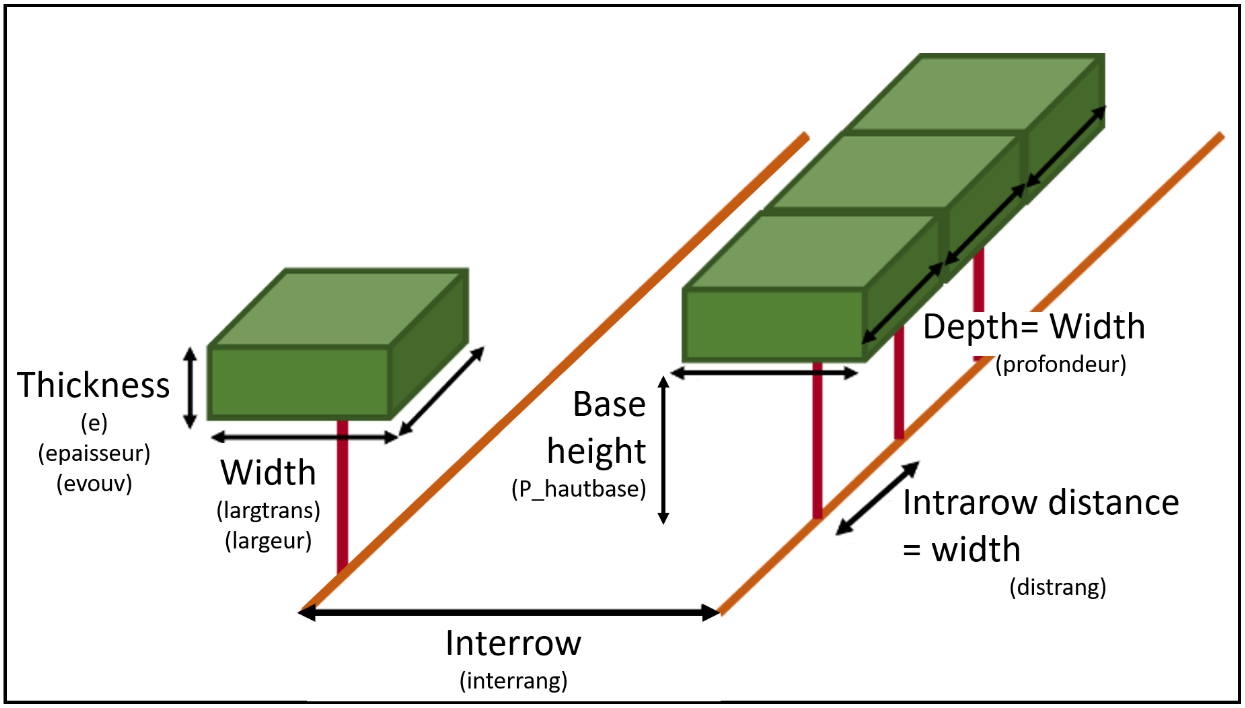


Figure 2.1: **Diagram representing the different parameters used to compute plant width. The different names used in the model are shown between parenthesis**

- 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:
  $LA = \frac{(lai+laisen+eai)}{densite}$ 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:
$largeur = \sqrt{\frac{(lai+laisen+eai) \cdot interrang}{dfol \cdot varrapforme}}$

where `varrapforme` (or `raptrans` further) is the ratio between the plant thickness and width and is computed as $varrapforme = \frac{(hauteur - P_{hautbase})}{largeur}$ 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)

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

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 dominant 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:
`sc%originehaut = max(p(i+1)%hauteur(sc%AO),p(i+1)%hauteur(sc%AS))`

> `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 downsided 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.

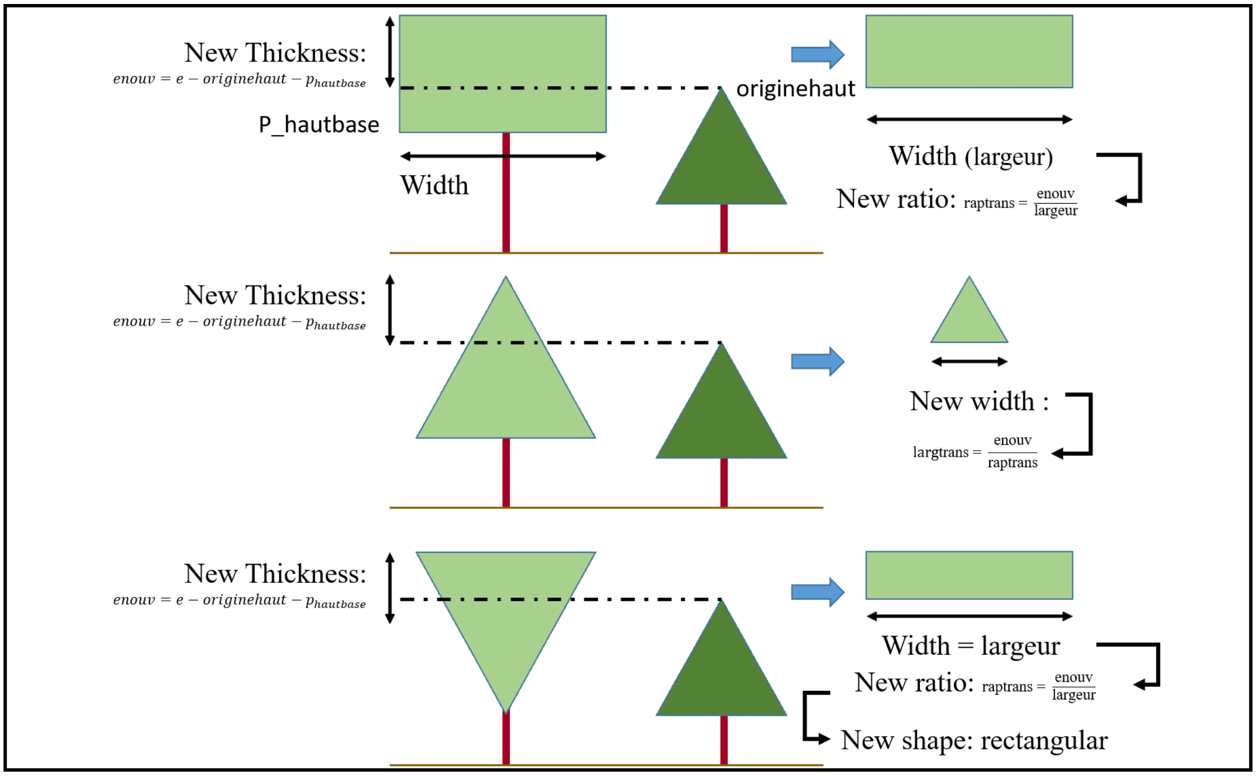


Figure 2.2: **Competition for radiation interception of the dominant plant induced by a high dominated plant**

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) of 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<l/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).

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:

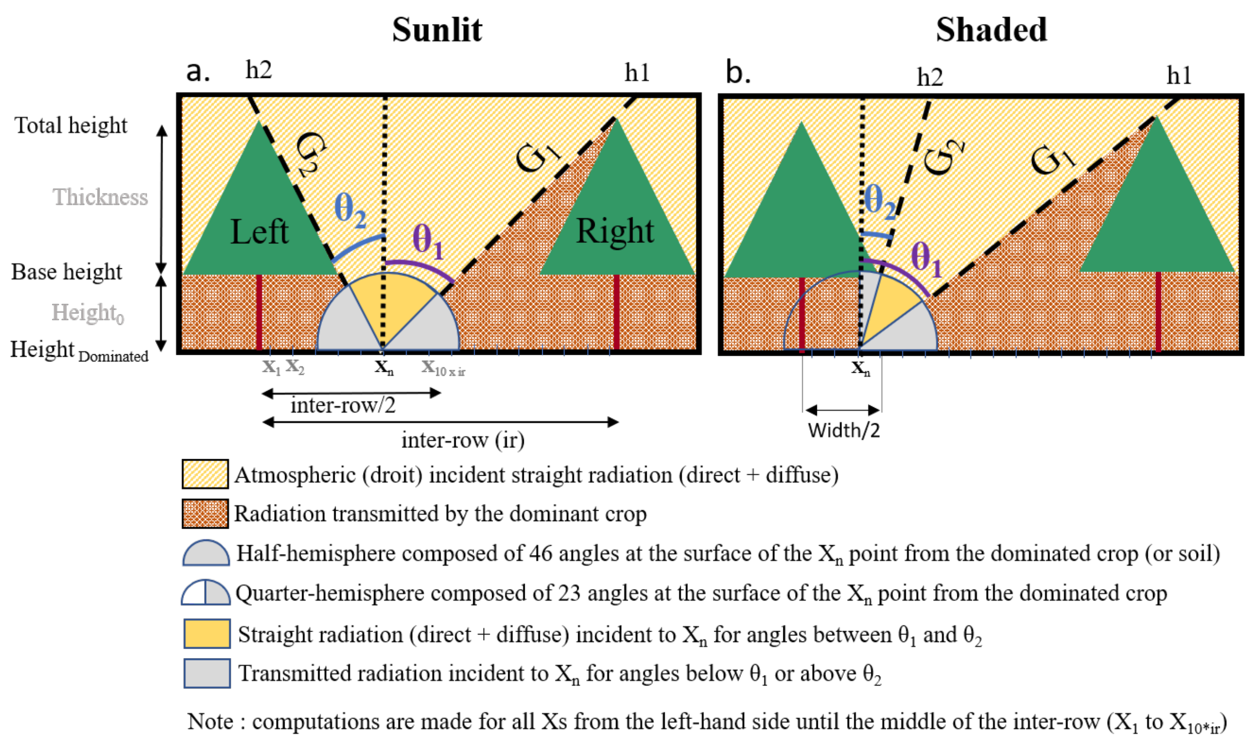


Figure 2.3: **Diagram of the computation workflow of STICS for radiation interception for two X points placed above the dominated plant species. a. The X point is considered sunlit; b. The X point is considered shaded (right under the dominant plant canopy), so only the right-hand side of the semi-hemisphere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

```
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...
  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."))
```

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

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)
SOCtab= 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(l>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$Angle==j]= SOCtab[j]
      ##########################################
    }
    ######## Not in the real function ########
    Output_diff$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]
        ##########################################
      }
      ######## Not in the real function ########
      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

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)	Light environ
1	0.00	0.7727202	0.3106046	0.4621155	Shaded
2	0.05	0.7784515	0.3279890	0.4504624	Shaded
3	0.10	0.7799044	0.3323963	0.4475082	Sunlit
4	0.15	0.7931983	0.3727197	0.4204785	Sunlit
5	0.20	0.7918188	0.3685355	0.4232833	Sunlit
6	0.25	0.7975112	0.3858020	0.4117092	Sunlit
7	0.30	0.7984328	0.3885975	0.4098354	Sunlit
8	0.35	0.8011901	0.3969610	0.4042292	Sunlit
9	0.40	0.7961262	0.3816007	0.4145254	Sunlit
10	0.45	0.7963256	0.3822058	0.4141198	Sunlit

2.3.4 Light tramsitted 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:

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

where P_{ktrou} 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:

The Total incident ratio is computed as: `Total incident ratio= rdroit+rtransmis`. It is the proportion of the incident ligth 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

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	0.7755858
rsoleil	Sunlit	0.7943134

canopy receives in average `incident_light$Total[1]*100`% of the light coming from the atmosphere: `incident_light$rdroit[1]*100`% of the light directly from the sky (either direct or diffuse light) + `incident_light$rtransmis[1]*100`% 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:

Finally, both ratios are used to compute the intercepted PAR `raint` (MJ m-2 d-1) of the target plant as:
$raint = P_{parsurrg} \cdot trg \cdot (1 - (rombre \cdot surfAO) - (rsoleil \cdot surfAS))$
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.

For our example, the crop would have intercepted NA, NA MJ m-2 day-1 of radiation, with a PAR of 12.5 W m-2, an `rombre` of 0.7755858, NA, an `rsoleil` of NA, 0.7943134 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. Substracting 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.

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.

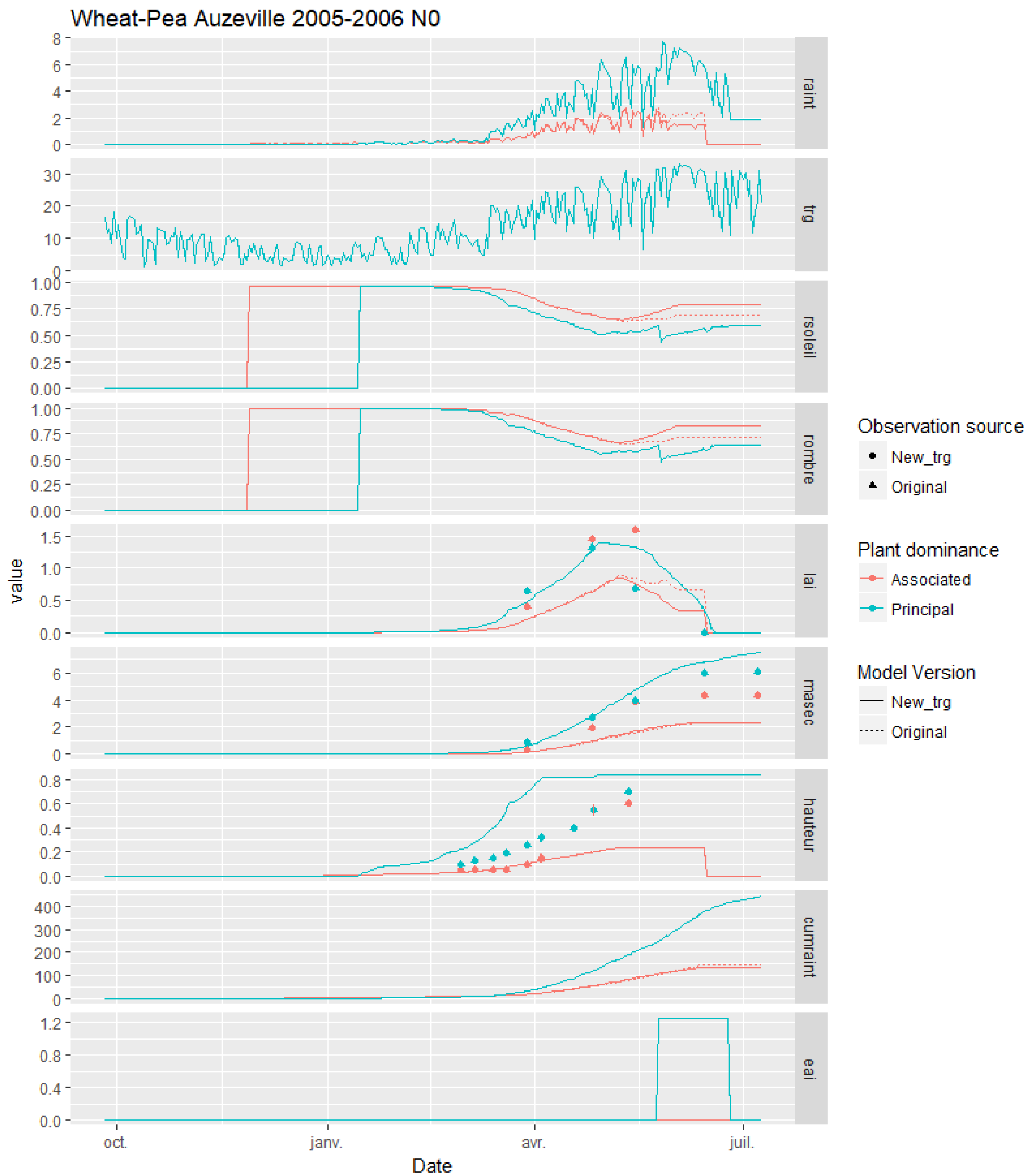


Figure 3.1: trg comparison

Chapter 4

Wheat EAI parameterization

4.1 Introduction

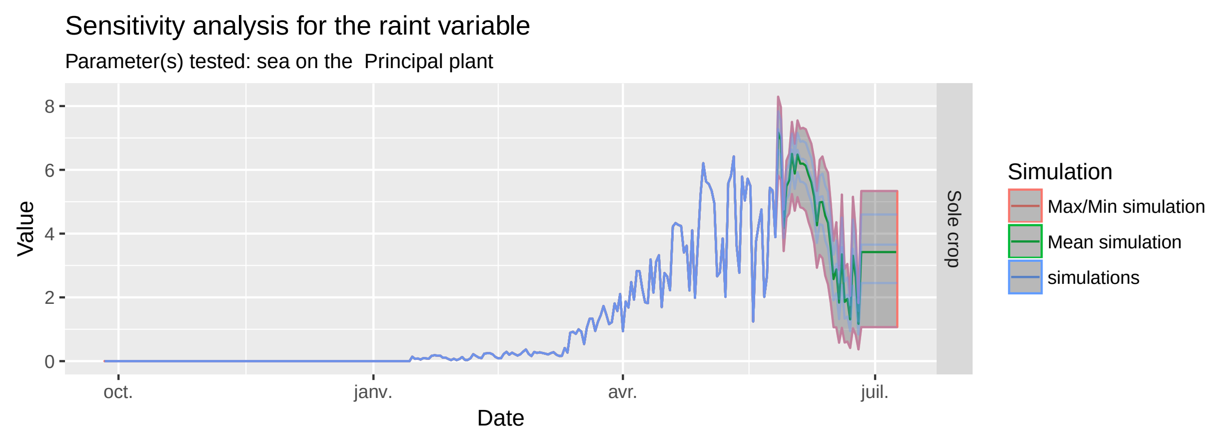
The simulations outputs from Chapter 3 have shown that the wheat ears equivalent photosynthetic surface area (`eai`) was probably overestimated by a 3-4 factor. This surface is computed using the fruit dry mass (`maenfruit`) and a parameter (`P_sea`) as follow:

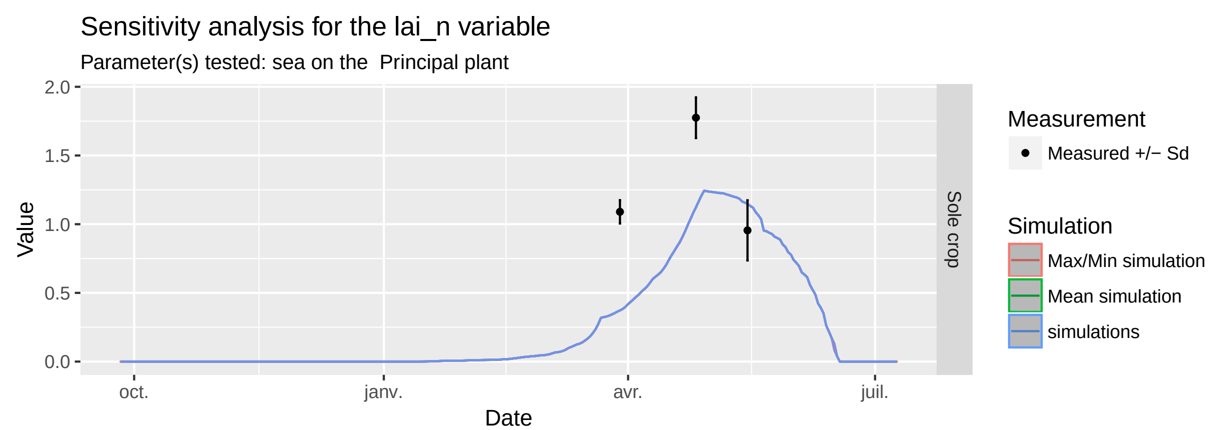
$eai = P_{sea} \cdot \frac{maenfruit}{100}$

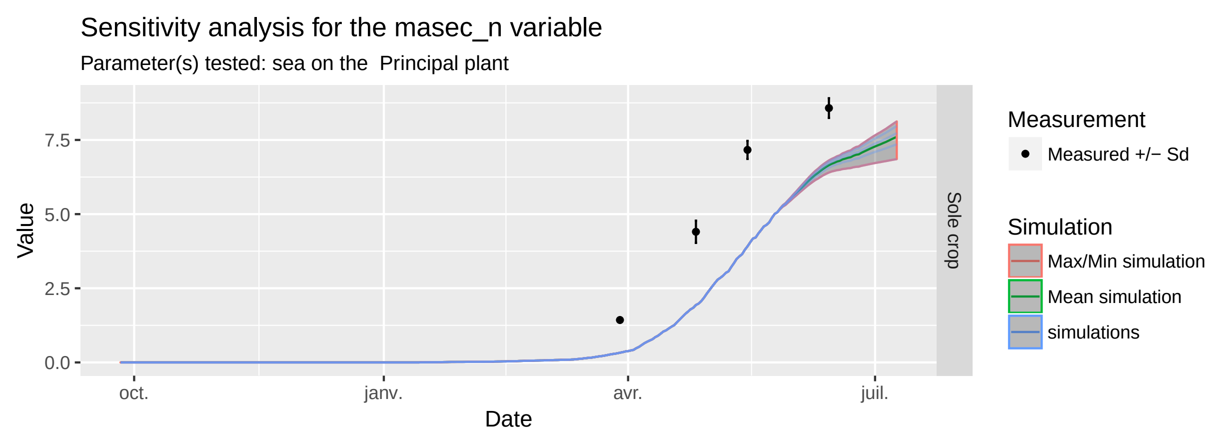
This parameter was originally equal to 100, meaning that each gram of fruit dry mass gave 1 m2 of equivalent `LAI`. This value was probably overestimated, but before re-parameterizing the plant, we first needed to know the model sensitivity to this parameter.

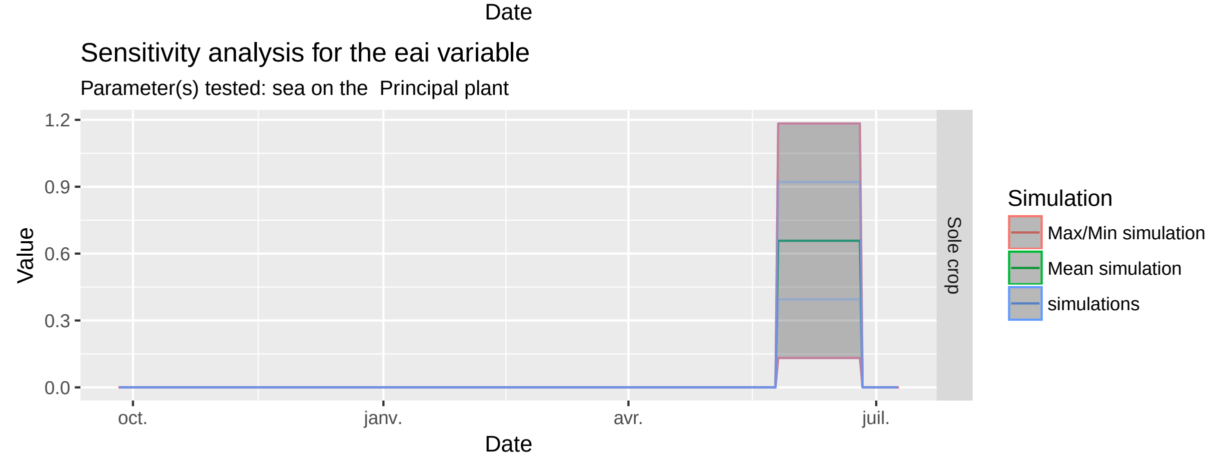
4.2 Model sensitivity to the `P_sea` parameter

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):


Sensitivity analysis for the raint variable
Parameter(s) tested: sea on the Principal plant
Value
0
2
4
6
8
oct.
janv.
avr.
juil.
Date
Sole crop
Simulation
Max/Min simulation
Mean simulation
simulations

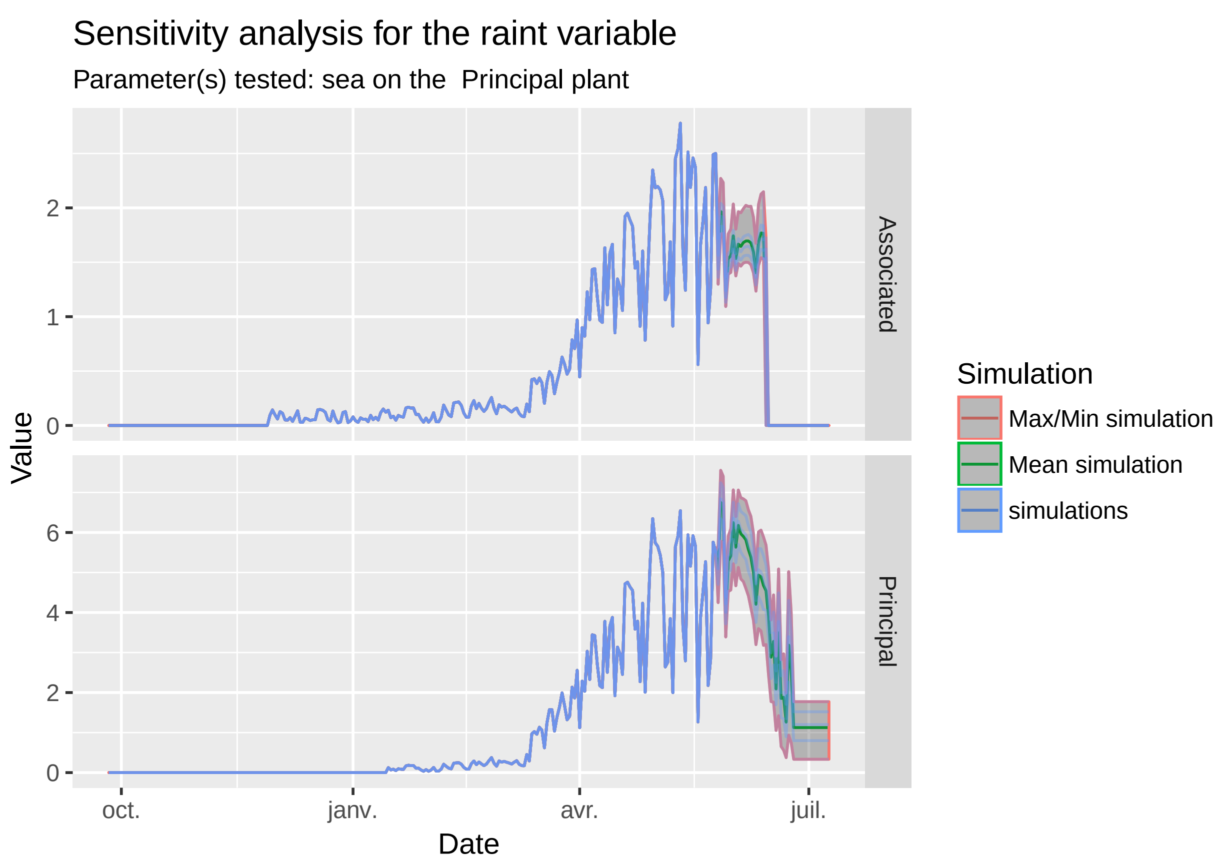

Sensitivity analysis for the lai_n variable
Parameter(s) tested: sea on the Principal plant
Value
0.0
0.5
1.0
1.5
2.0
oct.
janv.
avr.
juil.
Date
Sole crop
Measurement
Measured +/− Sd
Simulation
Max/Min simulation
Mean simulation
simulations


Sensitivity analysis for the masec_n variable
Parameter(s) tested: sea on the Principal plant
Value
0.0
2.5
5.0
7.5
oct.
janv.
avr.
juil.
Date
Sole crop
Measurement
Measured +/− Sd
Simulation
Max/Min simulation
Mean simulation
simulations

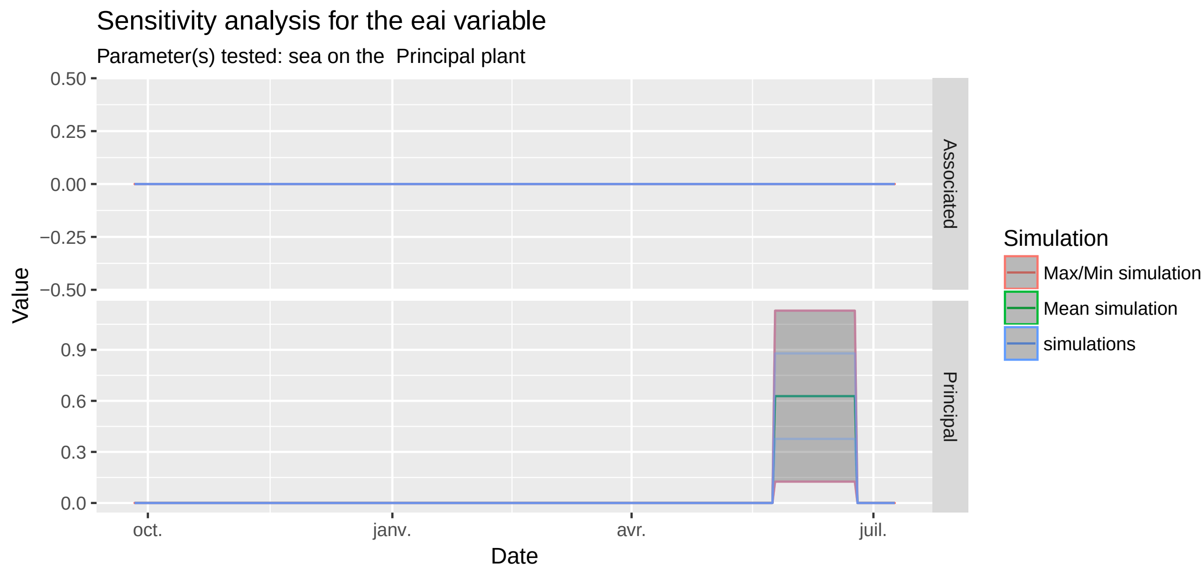

Sensitivity analysis for the eai variable
Parameter(s) tested: sea on the Principal plant
Value
0.0
0.3
0.6
0.9
1.2
oct.
janv.
avr.
juil.
Date
Sole crop
Simulation
Max/Min simulation
Mean simulation
simulations

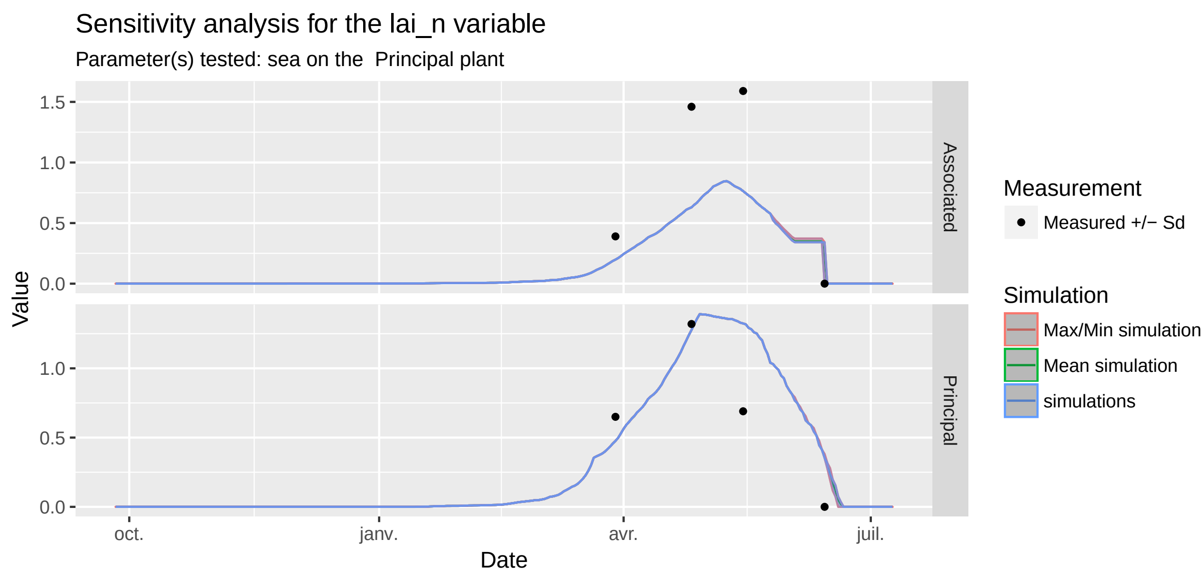
The results show that the higher the P_sea, the higher the eai, and hence increasing the light interception (raint), which induced a higher aboveground dry mass (masec). The plant LAI is not influenced by the P_sea though, because there is no retroactions on this variable.

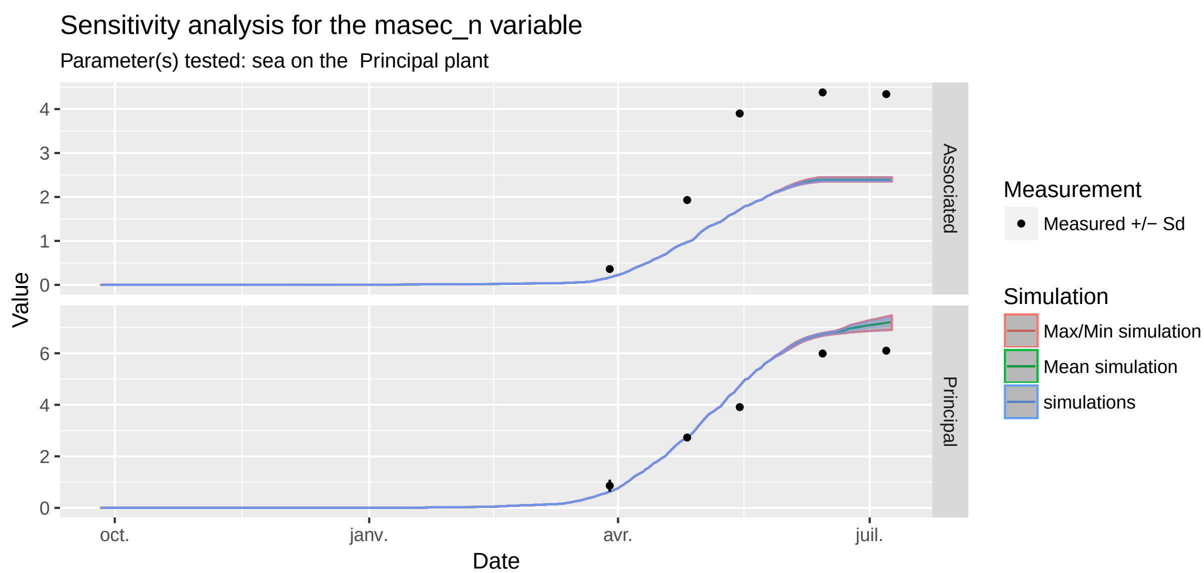
In the case of intercrops, the P_sea parameter can influence the dominated plant also, because less light is transmitted when P_sea is high :



The other variables are not much impacted though:


Sensitivity analysis for the eai variable
Parameter(s) tested: sea on the Principal plant
Value
0.50
0.25
0.00
−0.25
−0.50
0.9
0.6
0.3
0.0
Associated
Principal
oct.
janv.
avr.
juil.
Date
Simulation
Max/Min simulation
Mean simulation
simulations


Sensitivity analysis for the lai_n variable
Parameter(s) tested: sea on the Principal plant
Value
1.5
1.0
0.5
0.0
Associated
Principal
oct.
janv.
avr.
juil.
Date
Measurement
Measured +/− Sd
Simulation
Max/Min simulation
Mean simulation
simulations


Sensitivity analysis for the masec_n variable
Parameter(s) tested: sea on the Principal plant
Value
4
3
2
1
0
6
Associated
Principal
oct.
janv.
avr.
juil.
Date
Measurement
Measured +/− Sd
Simulation
Max/Min simulation
Mean simulation
simulations

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

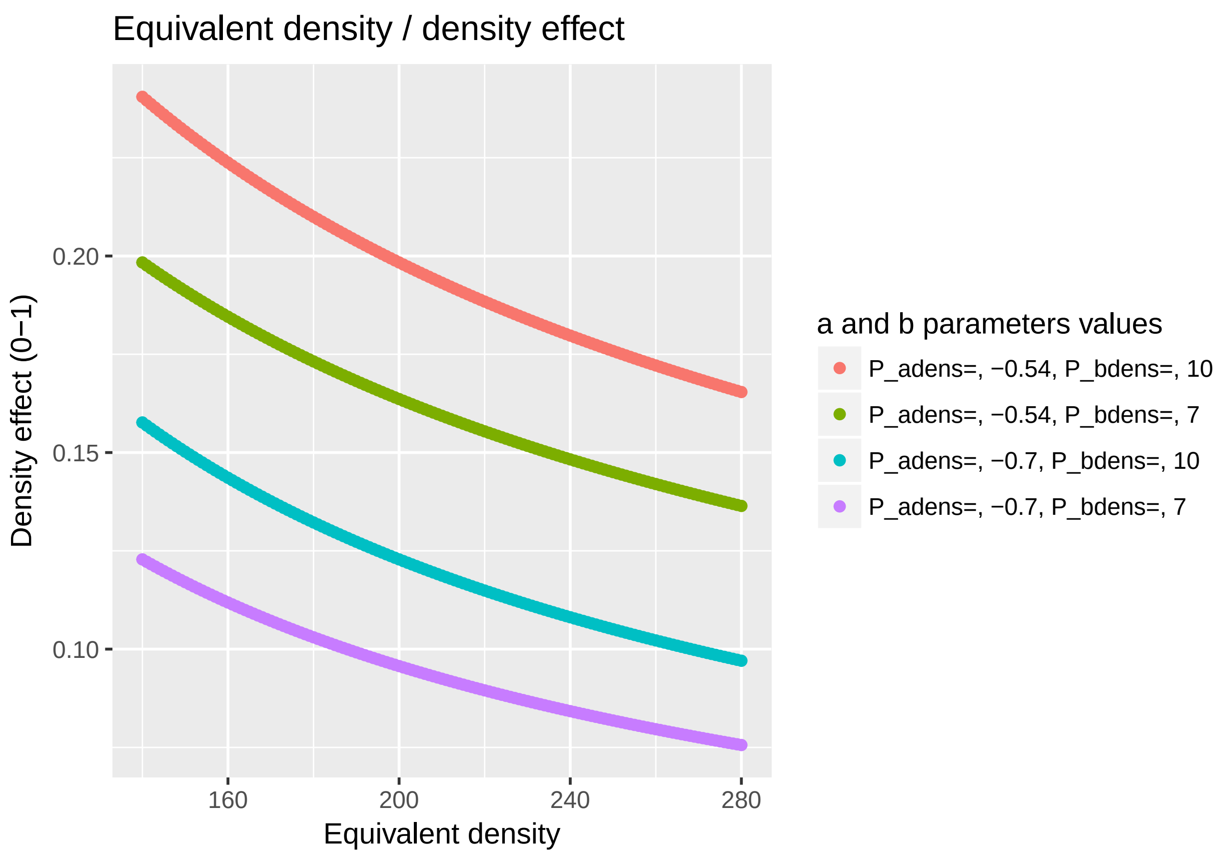
The plant density, which is related to the interrow distance, seems to be an important formalism to describe the crop, and particularly for mixed crops. Several computations are made to represent plant competition in the STICS model, making the density effect a complex process. Lets describe each step 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: $ef_{densite} = \min\left\{1.0\ ;\ (\frac{densiteequiv}{P_{bdens}})^{P_{adens}}\right\}$

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.

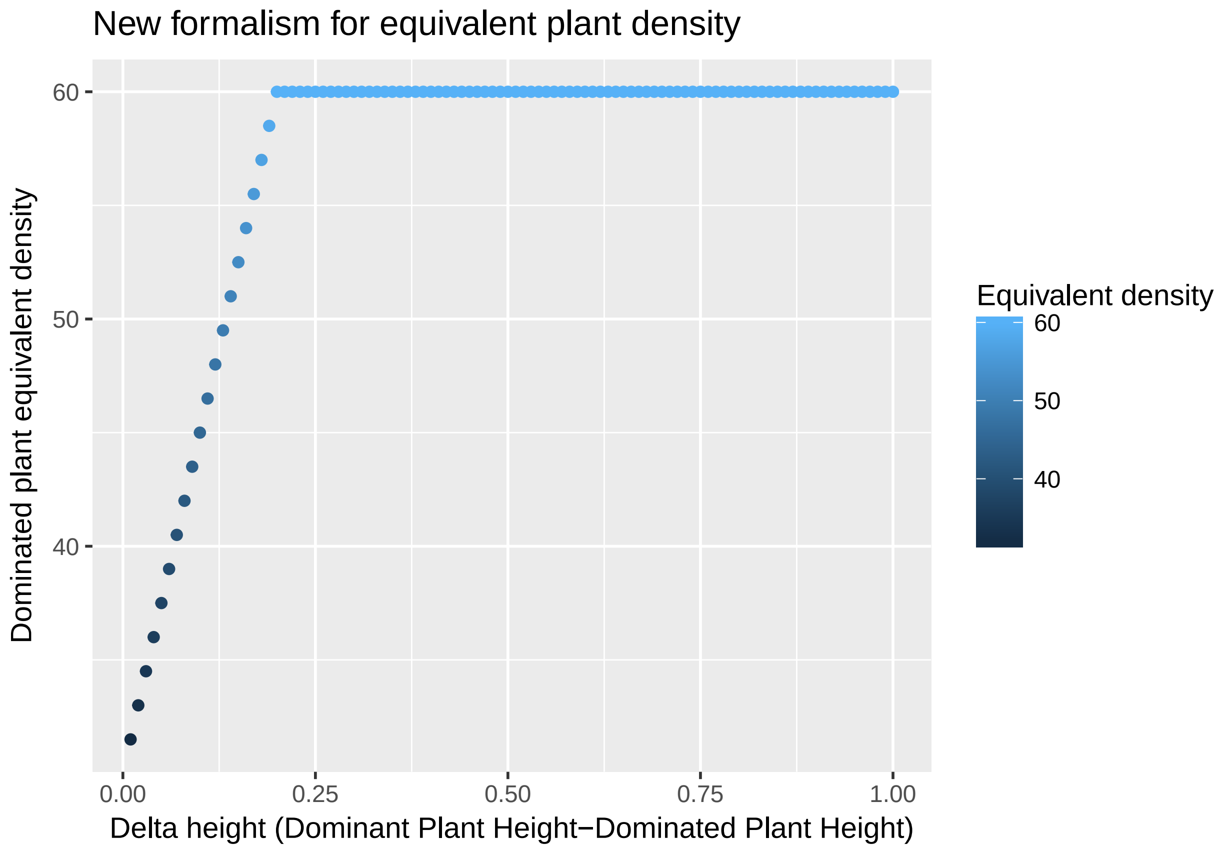
Then the density effect is computed as an equivalent density instead (`densiteequiv`), that can differ from the sowing density for the dominated crop to increase the negative effect of `efdensite` compared to a sole crop.

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, Sebastian Munz modified the STICS code to implement a new formalism to define the equivalent density as a function of the height difference between the plants as follow:

$$density_{Equivalent} = \begin{cases} \Delta_{height} > hauteur_{threshold} & density_{p2} + \frac{density_{p1}}{P_{bdensp1}} \cdot P_{bdensp2} \\ \Delta_{height} < hauteur_{threshold} & density_{p2} + slope \cdot abs\left|\Delta_{height}\right| \end{cases}$$

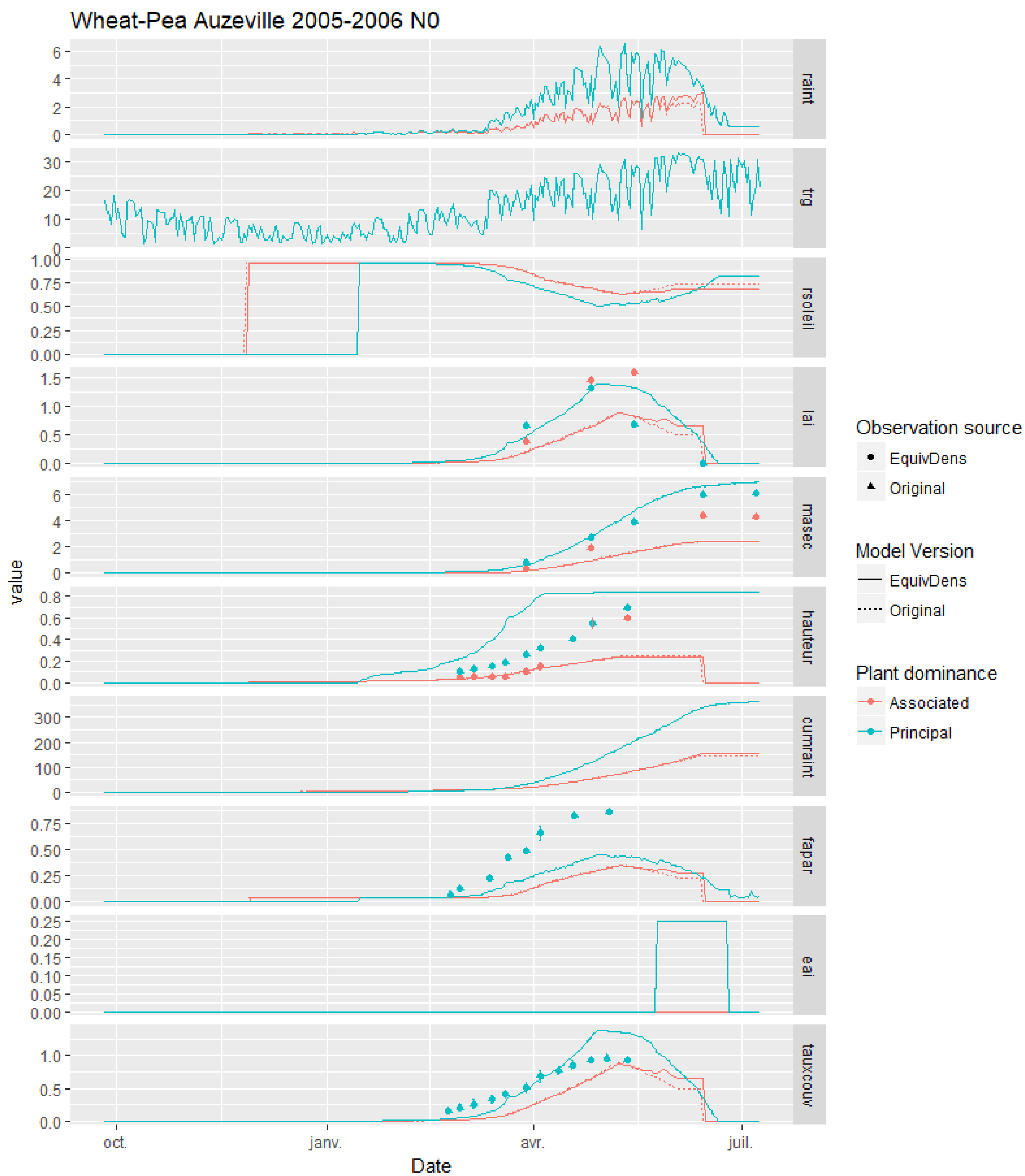
with $diffx = \frac{density_{p1}}{P_{bdensp1}} \cdot P_{bdensp2}$ and $slope = \frac{diffx}{hauteur_{threshold}}$

Here is an exemple with 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:



The new formalism 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.

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



The results showed that the new equivalent density computation is less severe for the dominated plant compared to the previous formalism. Consequently, the dominated plant intercepts more light (see `cumraint` and `fapar`) and has a higher `LAI`.

Now the next step is to parameterize well the parameters for the wheat and the pea.

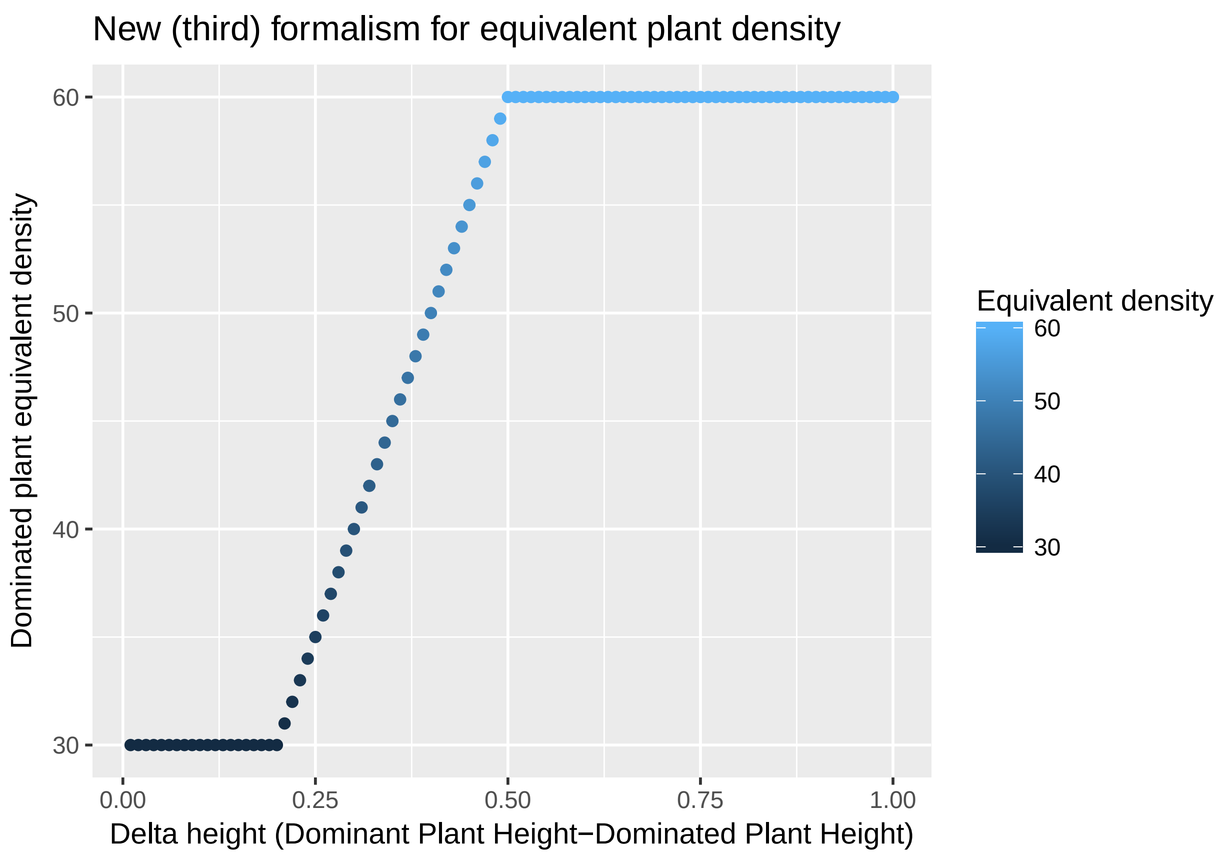
5.4 Proposition for a newer equivalent density

E.Justes, proposed a new (third) equivalent density computation that would have no increase in equivalent density until a given threshold (`hauteur_threshold_1`), 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 resulting equivalent density would be as follows:

$$density_{Equivalent} = \begin{cases} \Delta_{height} < hauteur_{threshold_1} & density_{p2} \\ \Delta_{height} > hauteur_{threshold_1} & b + \Delta_{height} \cdot slope \\ \Delta_{height} > hauteur_{threshold_2} & Max_{equDens} \end{cases}$$

with $slope = \frac{Max_{equDens}}{hauteur_{threshold_2} - hauteur_{threshold_1}}$

Here is the resulting plot for the same Wheat-Pea intercrop:



This formalism would include three parameters:

- hauteur_threshold_1: the difference in height below which no competition is occuring
- hauteur_threshold_2: the difference in height below which a progressive competition is occuring
- Max_equDens: the maximum equivalent density allowed.

5.5 Proposition to remove the old explicit interspecies density competition effect

For the moment, the competition induced by the density of the other species 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 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.

We now have to test this proposition.

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.

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 dominatd 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.

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 golbal 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.

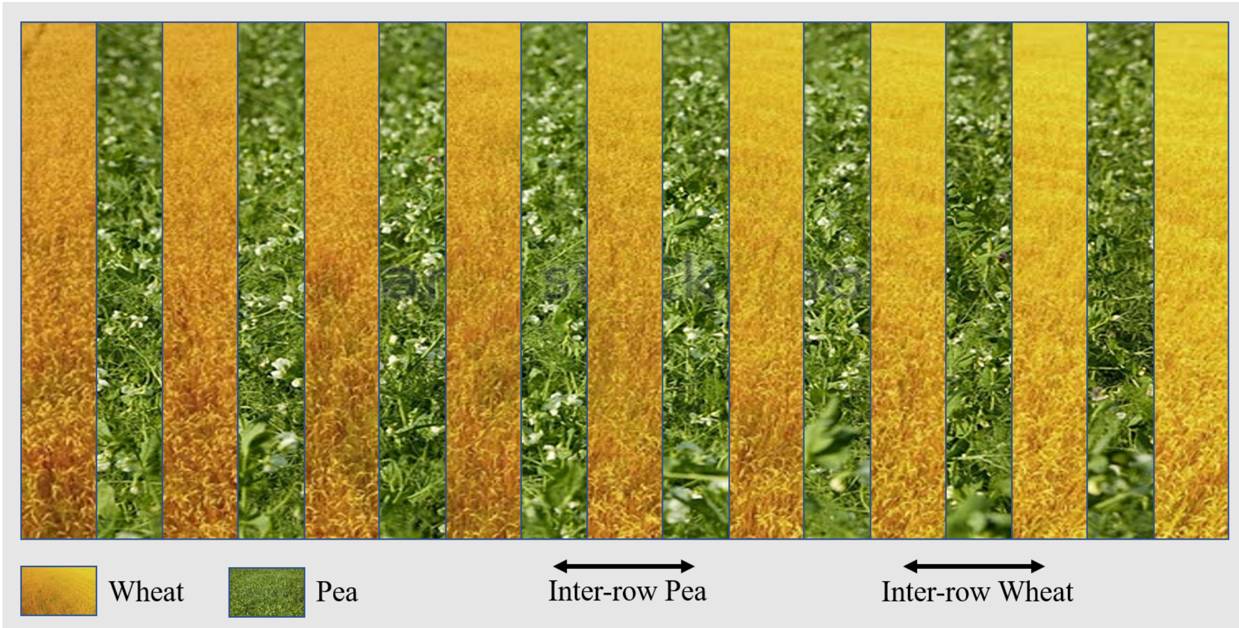


Figure 6.1: Pea-Wheat intercrop using the same inter-row spacing for the two crops

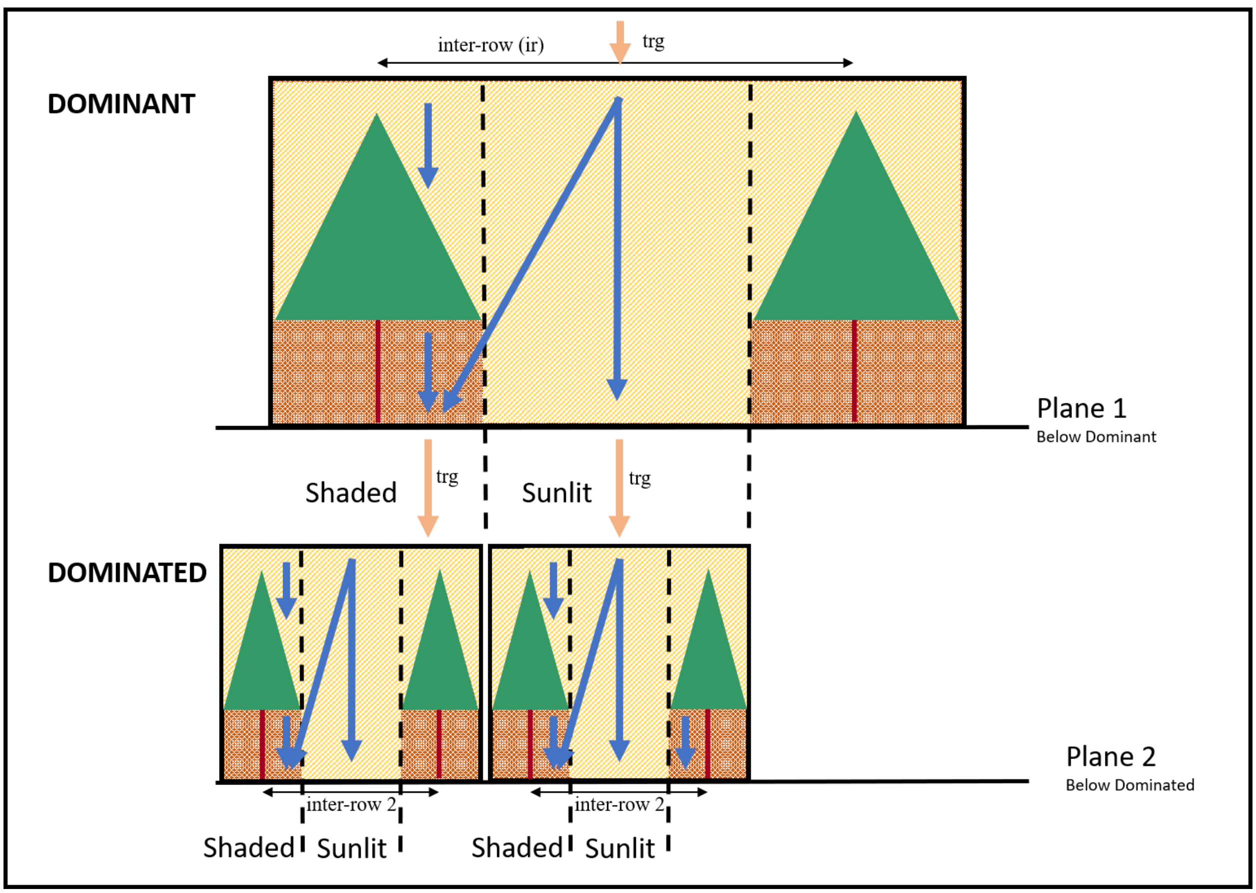


Figure 6.2: Interrow effect on light interception for intercrops in the STICS model

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 were the model can be applied according to its formalisms, and were 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 vailidity 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 unreallistic for this design. The model could simulate large strips as two separated crops though.

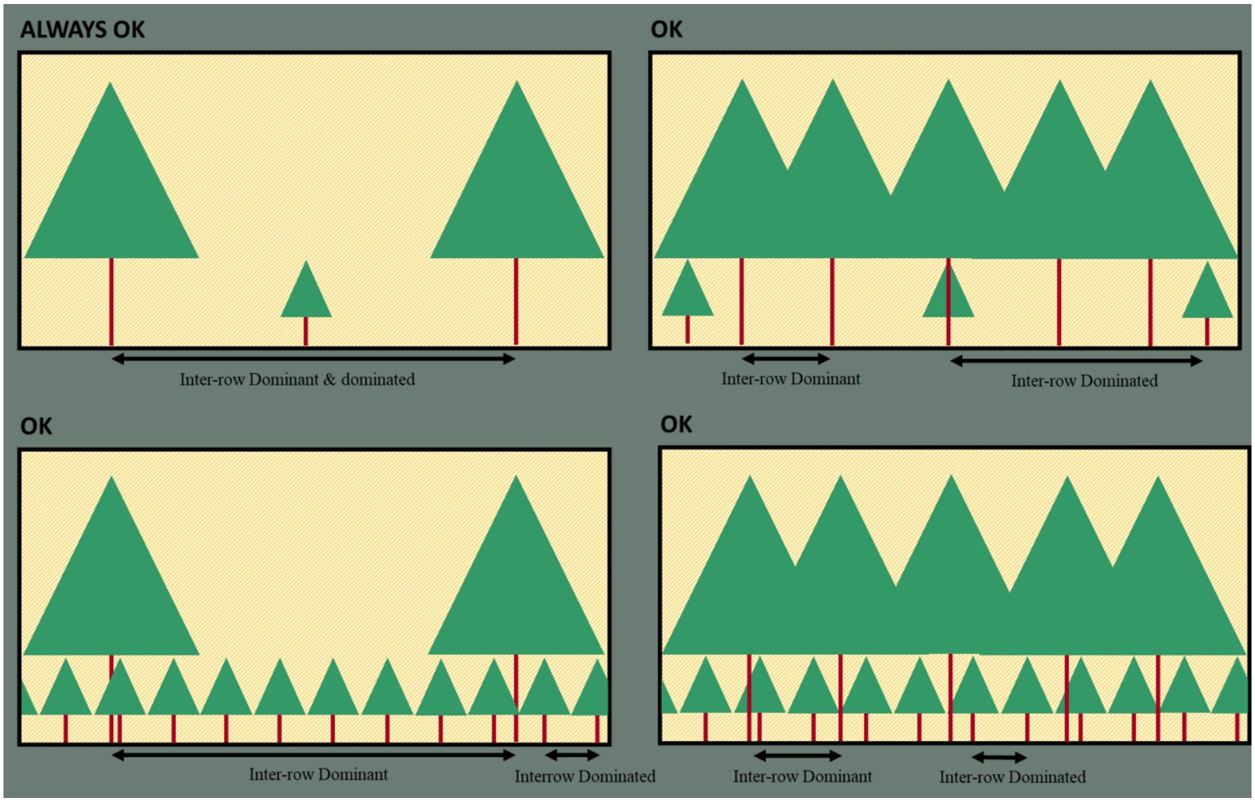


Figure 7.1: Depiction of the potentially adapted intercrop designs for simulation using the STICS model.

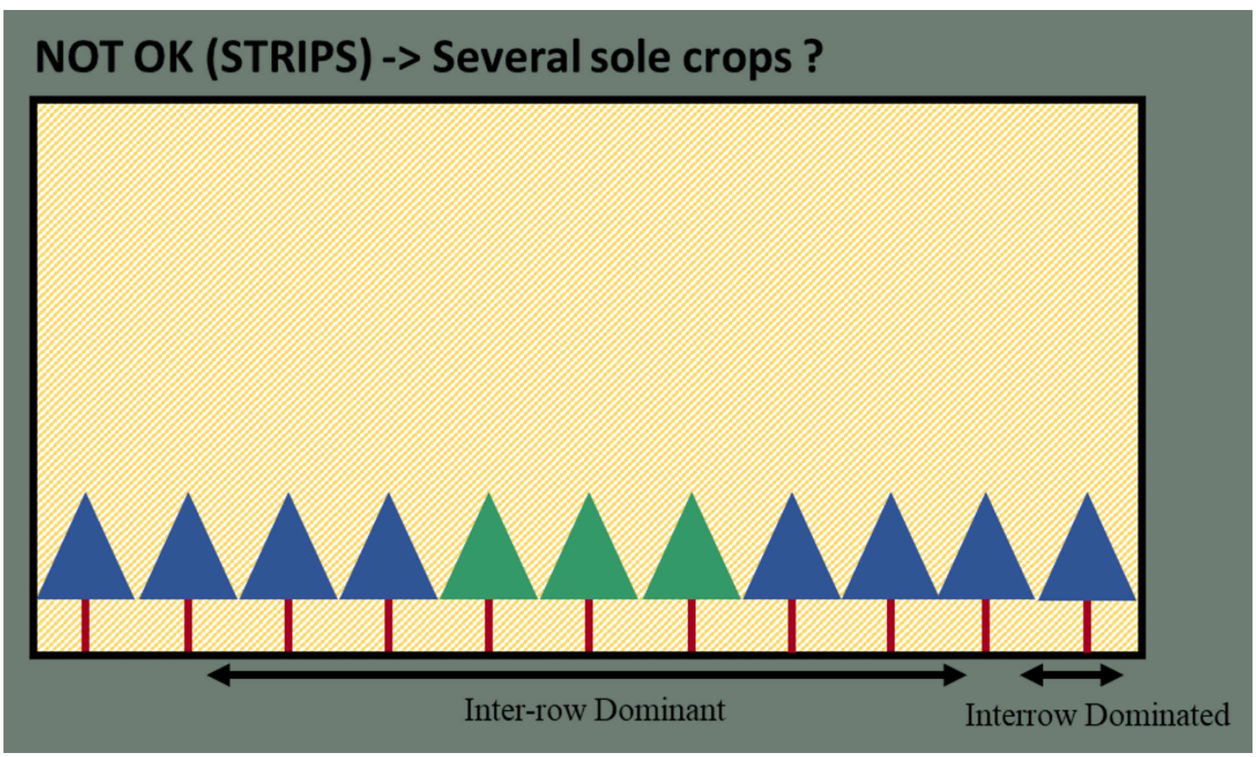


Figure 7.2: Depiction of the intercrop designs not adapted for simulation using the STICS model as is.