

# LBRAI2219 – Modélisation des Systèmes Biologiques

Granato Noah

26/05/2025

## Contents

<b>1 Résumé</b>	<b>2</b>
<b>2 Introduction</b>	<b>2</b>
2.1 Etat de l'art . . . . .	2
2.2 Objectifs . . . . .	2
2.3 Paramètres Sélectionnés . . . . .	3
<b>3 Matériel et méthodes</b>	<b>3</b>
3.1 Modèles utilisés . . . . .	3
3.2 Packages utilisés . . . . .	3
3.3 Fonctions et codes partagés utilisés . . . . .	4
<b>4 Schéma de Modélisation</b>	<b>5</b>
<b>5 Résultats</b>	<b>5</b>
5.1 Simulation classique de CRootBox + Marshal . . . . .	5
5.2 Simulations CRootBox + Marshal pour différentes valeurs de a et r (pour tous les types de racines) . . . . .	9
5.3 Simulations CRootBox + Marshal pour différentes valeurs de a et r (taproot) . . . . .	14
5.4 Identifier les Tact min et max . . . . .	18
5.5 Tentative de couplage entre Marshal et le modèle de culture Apsim . . . . .	21
<b>6 Conclusion générale</b>	<b>37</b>
<b>7 Références</b>	<b>38</b>

# 1 Résumé

## 2 Introduction

### 2.1 Etat de l'art

Dans le contexte actuel de changement climatique, où les épisodes de sécheresse deviennent malheureusement plus fréquents et plus intenses, améliorer la résilience des cultures est devenu un véritable défi pour l'agriculture (IPCC, 2022 ; Spinoni et al., 2018). Tester en conditions réelles toutes les options possibles (différentes variétés, architectures racinaires, types de sols, climats) prend non seulement beaucoup de temps, mais représente aussi un coût important, et reste souvent techniquement irréalisable (Messina et al., 2022).

Parmi les cultures concernées, le maïs (*Zea mays*) occupe une place centrale. C'est l'une des céréales les plus cultivées au monde, utilisée aussi bien pour l'alimentation humaine que pour l'élevage ou l'industrie (FAO, 2021). Mais c'est aussi une espèce particulièrement sensible au stress hydrique, notamment au stade de floraison (Messina et al., 2015). Son rendement peut chuter drastiquement en cas de déficit en eau, ce qui en fait une culture prioritaire à améliorer en termes de tolérance à la sécheresse (Bänziger et al., 2000). Mieux comprendre le rôle des racines dans cette réponse au stress est donc essentiel pour guider les futures sélections variétales.

C'est justement là que la modélisation prend tout son sens. En représentant le fonctionnement d'une plante ou d'un écosystème à l'aide d'un modèle mathématique ou informatique, on peut explorer rapidement une grande diversité de scénarios, combiner plusieurs paramètres, et voir lesquels influencent le plus un critère essentiel (Postma et al., 2017).

Bien sûr, les modèles ne remplacent pas les expérimentations sur le terrain. Mais ils offrent un aperçu simplifié et cohérent de la réalité, qui permet de tester des hypothèses, d'écartier certaines pistes peu prometteuses, et surtout, de se concentrer sur les solutions les plus intéressantes. En ce sens, ils deviennent de véritables outils d'orientation et de réflexion (Gonzalez et al., 2021).

Ce projet s'inscrit dans cette démarche : utiliser les modèles pour mieux comprendre ce qui, dans le système racinaire du maïs, peut réellement faire la différence face à la sécheresse.

### 2.2 Objectifs

Ce travail a pour objectif principal de comprendre comment certains traits racinaires influencent le rendement du maïs en conditions de sécheresse. Plus précisément, il s'agit d'évaluer dans quelle mesure des variations dans la structure des racines peuvent modifier la capacité de la plante à absorber l'eau, et par conséquent, affecter sa croissance et sa production de biomasse.

Pour cela, l'approche choisie repose sur une chaîne de modélisation en plusieurs étapes, allant de l'échelle de la racine à celle de la culture. À chaque étape, des paramètres spécifiques sont simulés ou calculés, afin de faire le lien entre l'architecture racinaire, le fonctionnement hydraulique de la plante, et le développement global de la culture.

L'objectif est donc double :

1. Explorer virtuellement différents scénarios racinaires, afin d'identifier les configurations qui favorisent une meilleure absorption de l'eau dans un contexte de stress hydrique.
2. Évaluer l'impact de ces configurations sur le rendement simulé, en intégrant les résultats dans un modèle de culture soumis à des conditions sèches.

## 2.3 Paramètres Sélectionnés

Parmi l'ensemble des paramètres racinaires potentiellement influents, deux ont été retenus pour cette étude : la vitesse de croissance et le rayon de la racine principale (taproot). D'autres traits comme l'angle d'élongation ou la densité de ramifications auraient également pu être pertinents, mais ces deux paramètres ont été choisis en raison de leur lien direct avec la profondeur atteinte par les racines et la capacité de transport hydraulique, deux aspects clés en conditions de stress hydrique.

Dans un contexte de sécheresse, plusieurs études ont montré que la vitesse de croissance de la racine principale joue un rôle crucial pour accéder rapidement aux couches profondes du sol, là où l'eau résiduelle reste disponible. Un enracinement rapide permet une meilleure acquisition hydrique dès les premiers stades de développement, ce qui améliore la croissance et le rendement final du maïs (Gleason et al., 2019 ; Feng et al., 2022). Ces travaux soulignent que ce trait est particulièrement déterminant en conditions de stress hydrique, où le temps pour atteindre l'eau disponible est limité.

En parallèle, le rayon de la racine principale influence directement la capacité de transport de l'eau dans la plante. Des racines plus épaisses sont associées à une plus grande conductivité hydraulique, facilitant l'approvisionnement de la plante en eau, même en sol compact ou sec (Klein et al., 2020 ; Rishmawi et al., 2023). Étudier ces deux paramètres ensemble permet donc d'explorer comment la géométrie et la dynamique de la racine pivot affectent l'efficacité d'absorption et la performance de la culture sous sécheresse.

## 3 Matériel et méthodes

### 3.1 Modèles utilisés

Ce travail repose sur l'utilisation de trois modèles complémentaires, organisés en chaîne. Chaque modèle intervient à une étape précise du raisonnement, de la représentation du système racinaire jusqu'à la simulation du rendement de la culture.

Le modèle CRootBox permet de simuler des systèmes racinaires en deux dimensions, à partir d'un ensemble de paramètres racinaires. Il reproduit la croissance des racines dans le sol au fil du temps, en tenant compte de facteurs comme leur vitesse d'élongation ou leur rayon. Grâce à lui, il est possible de visualiser différentes architectures racinaires et d'évaluer l'effet de certains traits sur la structure globale du système.

Une fois l'architecture générée, le modèle MARSHAL est utilisé pour en évaluer le fonctionnement hydraulique. Il permet d'estimer l'efficacité de chaque système racinaire simulé à capter l'eau du sol, et fournit des résultats à l'échelle de la racine entière. C'est sur cette base que l'on peut ensuite intégrer les effets racinaires dans un modèle de culture.

Enfin, un modèle inspiré d'APSIM permet de simuler la croissance du maïs au cours du temps, dans des conditions environnementales données. Ce modèle prend en compte les propriétés climatiques, le sol, et les caractéristiques de la plante, y compris sa capacité à absorber l'eau. Il permet ainsi de relier les traits racinaires simulés à une dynamique de production, et d'estimer l'effet de chaque configuration sur le rendement final.

En articulant ces trois outils, on obtient un cadre cohérent pour analyser l'impact des racines sur la performance de la culture de maïs, en particulier lorsque l'eau devient un facteur limitant.

### 3.2 Packages utilisés

Manipulation et traitement de données :

Le package tidyverse regroupe un ensemble d'outils essentiels pour la science des données, incluant notamment :

- dplyr, utilisé pour filtrer, sélectionner, transformer, regrouper et résumer des jeux de données.
- readr, pour lire efficacement les fichiers CSV.
- tidyr, pour réorganiser les données sous un format exploitable.
- data.table, qui permet une manipulation rapide et économique en mémoire, particulièrement utile pour les grands ensembles de données.
- plyr, utilisé pour appliquer des fonctions à des sous-ensembles de données (split-apply-combine).

Visualisation des résultats :

- Le package ggplot2 a été utilisé pour produire des graphiques clairs et personnalisés, représentant l'évolution des variables simulées.
- Le package viridis a fourni une palette de couleurs optimisée pour les gradients continus et perceptibles, même en impression noir et blanc.
- gridExtra et cowplot ont permis de combiner plusieurs graphiques dans une même figure et d'en améliorer la présentation.

Calculs matriciels :

Le package Matrix a été utilisé pour effectuer les opérations sur matrices creuses, nécessaires au calcul du flux d'eau dans MARSHAL.

```
#####
# 1 : LOAD THE LIBRARIES AND SOURCE FILES
#####

library(tidyverse)
library(plyr)
library(readr)
library(data.table)
library(dplyr)
library(Matrix)
library(readxl)
library(tidyr)
library(ggplot2)
library(viridis) #palette de couleurs
library(patchwork) #pour faire un patchwork des 64 graphes
library(gridExtra)
```

### 3.3 Fonctions et codes partagés utilisés

Deux fichiers R ont été chargés pour intégrer des fonctions clés :

- io\_function.R : Ce fichier contient les fonctions d'entrée/sortie liées à CRootBox. Il permet notamment de lire et écrire les fichiers de paramètres rparam et pparam, en facilitant la modification automatisée des traits racinaires.
- getSUF.R : Il regroupe l'ensemble des calculs associés à MARSHAL, notamment la simulation du fonctionnement hydraulique des systèmes racinaires : conductivité globale, absorption radiale, potentiel hydrique et flux d'eau.

Deux scripts issus du GitHub partagé dans le cadre du cours ont été adaptés pour ce projet :

- vanGenuchten.R, développé à partir d'un script Python par Mattias Van Eetvelt et Basile Delvoie, a été utilisé pour modéliser la rétention d'eau dans le sol selon la courbe de van Genuchten.
- Apsim.R, proposé par Alice Falzon et Maxime Cornez, constitue le cœur du modèle de culture utilisé ici. Il simule la dynamique de la biomasse, de l'indice foliaire (LAI), de la transpiration et de l'eau disponible dans le sol, en réponse aux conditions météorologiques et aux propriétés du sol.

```
# Custom functions
source("inputs/functions/io_function.R") # CROOTBOX
source("inputs/functions/getSUF.R") # MARSHAL
source("inputs/functions/vanGenuchten.R")
```

## 4 Schéma de Modélisation

- 1) Choix des paramètres racinaires à tester → Rayon (a) et vitesse de croissance (r) de la racine principale
- 2) Simulation racinaire (CRootBox) → Génération d'architectures racinaires selon différentes combinaisons de a et r
- 3) Évaluation hydraulique (MARSHAL) → Calcul de l'efficacité d'absorption d'eau (Krs, Tact) pour chaque architecture simulée
- 4) Analyse de sensibilité → Comparaison des résultats selon les combinaisons de a et r
- 5) Couplage avec modèle de culture (APSIM) → Simulation de la croissance du maïs selon la capacité d'absorption d'eau obtenue
- 6) Analyse des performances → Comparaison des biomasses et transpirations finales → Identification des combinaisons racinaires les plus efficaces

## 5 Résultats

### 5.1 Simulation classique de CRootBox + Marshal

Cette première étape a pour objectif de réaliser une simulation complète de référence, en utilisant les paramètres par défaut des modèles.

La simulation commence par l'utilisation de CRootBox, permettant de reproduire la croissance d'un système racinaire dans le sol. Le résultat est ensuite visualisé, ce qui donne un aperçu de la forme générale du réseau racinaire.

Dans un second temps, ce système racinaire est analysé avec MARSHAL, un modèle qui permet de simuler le fonctionnement hydraulique des racines. Cela signifie qu'on évalue comment l'eau circule dans le système racinaire, et comment elle est absorbée depuis le sol jusqu'à la plante. On obtient ainsi une estimation de l'efficacité du système à capter l'eau.

En résumé, cette simulation de base permet de tester le bon fonctionnement du pipeline CRootBox–MARSHAL et de disposer d'une référence pour les prochaines simulations, où certains paramètres racinaires seront modifiés.

```

#####
# 2 : RUN SINGLE SIMULATION FOR DEFAULT PARAMS (CRootBox + Marshal)
#####

# We load the default parameter sets for the CRootBox simulation
rparam <- read_rparam(path = "inputs/param.rparam")
pparam <- read_pparam(path = "inputs/param.pparam")

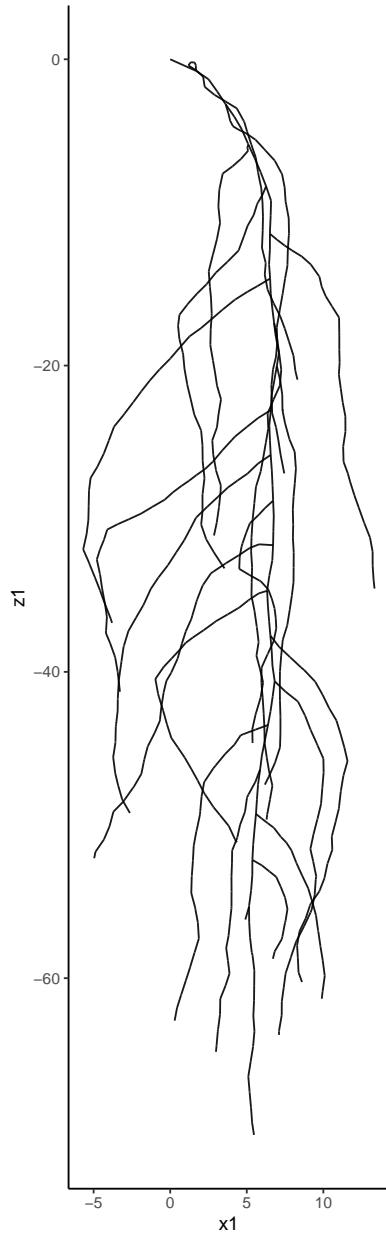
# Check the parameter files
head(rparam)

# Run crootbox
system("inputs/crootbox.exe") # Run crootbox for windows

# Get the results of the simulation
rootsystem <- fread("outputs/current_rootsystem.txt", header = T)

# Plot the root system
rootsystem %>%
  ggplot() +
  theme_classic() +
  geom_segment(aes(x = x1, y = z1, xend = x2, yend = z2), alpha=0.9) +
  coord_fixed()

```



```

# We load the default parameter sets for the MARSHAL simulation
psiCollar <- -15000
soil <- read_csv("inputs/soil.csv")
conductivities <- read_csv("inputs/conductivities.csv")

head(conductivities)
print(soil)

# Run MARSHAL
hydraulics <- getSUF(rootsystem, conductivities, soil, psiCollar)

# Map the segment-scale results on the root system
hydraulic_archi <- hydraulics$root_system
hydraulic_archi$suf <- hydraulics$suf[,1]

```

```

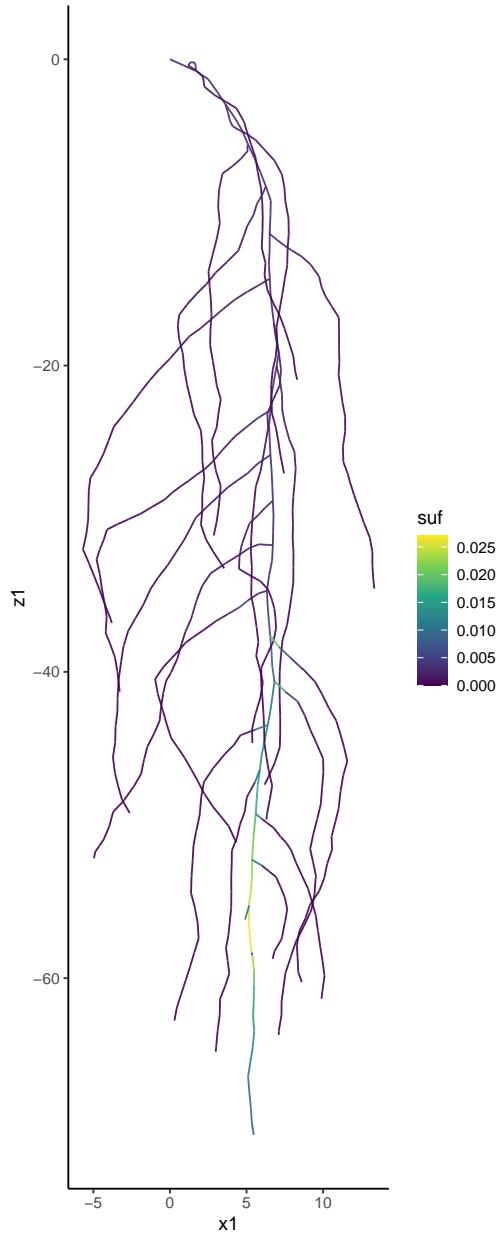
hydraulic_archi$kr <- hydraulics$kr[,1]
hydraulic_archi$kx <- hydraulics$kx[,1]
hydraulic_archi$jr <- hydraulics$jr[,1]
hydraulic_archi$jxl <- hydraulics$jxl[,1]

# Get the plant scale results

print(paste0("Root system conductivity = ",hydraulics$krs))
print(paste0("Root system transpiration = ",hydraulics$tact))

# Plot the results
hydraulic_archi %>%
  ggplot() +
  theme_classic() +
  geom_segment(aes(x = x1, y = z1, xend = x2, yend = z2, col = suf), alpha=0.9) +
  scale_color_viridis_c() +
  coord_fixed()

```



## 5.2 Simulations CRootBox + Marshal pour différentes valeurs de a et r (pour tous les types de racines)

Dans cette section, une analyse de sensibilité est menée afin d'examiner comment deux paramètres morphologiques influencent le fonctionnement hydraulique global du système racinaire. Les paramètres étudiés sont :

a : le rayon des racines (en cm), qui influence la capacité de transport de l'eau ;

r : la vitesse de croissance des racines (en cm/jour), qui détermine à quelle profondeur et avec quelle rapidité le système racinaire se développe.

Ces deux paramètres sont modifiés simultanément pour les trois types de racines simulées par CRootBox : la racine principale (taproot), les racines latérales (lateral), et les longues latérales (longlateral). L'objectif

est de tester toutes les combinaisons possibles de valeurs dans une grille de simulation (8 valeurs pour a, 8 pour r, soit 64 combinaisons au total).

Pour chaque combinaison, le pipeline suivant est appliqué :

- a) Les fichiers de paramètres de CRootBox sont mis à jour avec les valeurs de a et r pour les trois types de racines.
- b) CRootBox est exécuté pour générer une architecture racinaire complète correspondant aux paramètres définis.
- c) Le système racinaire simulé est ensuite évalué avec MARSHAL, afin de simuler son fonctionnement hydraulique.
- d) Deux indicateurs principaux sont enregistrés :
  - Krs : conductivité hydraulique totale du système racinaire,
  - Tact : transpiration potentielle de la plante.
- e) Une visualisation immédiate de la structure racinaire est générée pour chaque simulation, avec une coloration indiquant la répartition spatiale de l'absorption d'eau (SUF).

L'objectif final est d'évaluer l'effet combiné du rayon et de la vitesse de croissance des racines sur l'efficacité d'absorption d'eau, en condition de déficit hydrique, et d'identifier les configurations les plus favorables à un fonctionnement racinaire performant.

```
# (re)lis les fichiers-template pour ne pas les altérer définitivement
rparam_tpl <- read_rparam("inputs/param.rparam")
pparam_tpl <- read_pparam("inputs/param.pparam")

# grilles de paramètres (pour racine primaire uniquement)
a_vec <- round(seq(0.05, 0.50, length.out = 8), 3) # rayon racinaire [cm]
r_vec <- round(seq(1.5, 4.5, length.out = 8), 2) # vitesse croissance [cm j-1]

# conteneur résultats plante
summary_results <- data.frame()

simid <- 0
for(a in a_vec){
  for(rgr in r_vec){

    simid <- simid + 1
    message("Simulation ", simid, "/", length(a_vec)*length(r_vec),
            " ( a = ", a, " ; r = ", rgr, " )")

    #prépare les fichiers d'entrée CRootBox
    rparam <- rparam_tpl
    pparam <- pparam_tpl

    rparam$val1[rparam$param == "a"] <- a
    rparam$val1[rparam$param == "r"] <- rgr

    write_rparam(rparam, "inputs/param.rparam")
    write_pparam(pparam, "inputs/param.pparam")
  }
}
```

```

# Run CRootBox
system("inputs/crootbox.exe")

rootsys <- fread("outputs/current_rootsystem.txt")

# Run Marshal
hydro   <- getSUF(rootsys, conductivities, soil, psiCollar)

# Stocke les sorties Krs & Tact

summary_results <- rbind(summary_results,
                           data.frame(a = a,
                                       r = rgr,
                                       Krs = hydro$krs,
                                       Tact = hydro$tact)
)
}

# Visualisation immédiate du réseau racinaire

archi <- hydro$root_system
archi$suf <- hydro$suf[, 1]

g <- ggplot(archi) +
  geom_segment(aes(x = x1, y = z1,
                    xend = x2, yend = z2,
                    colour = suf),
                linewidth = 0.25) +
  scale_colour_viridis_c(option = "plasma",
                         name = "SUF",
                         guide = guide_colourbar(barwidth = 0.5,
                                                  barheight = 4)) +
  coord_fixed() +
  theme_void() +
  ggtitle(sprintf("a = %.2f cm | r = %.2f cm·j⁻¹", a, rgr))

  print(g)
}
}

```

### 5.2.1 Génération de Heatmaps(Krs et Tact)

La première map montre l'évolution de la conductivité hydraulique globale du système racinaire (Krs). On observe une tendance claire et continue à l'augmentation de Krs lorsque le rayon racinaire augmente, ainsi que lorsque la vitesse de croissance augmente. Cela signifie que des racines plus épaisses, combinées à un développement plus rapide dans le sol, permettent à la plante de créer un système racinaire plus performant du point de vue du transport d'eau.

Cette tendance est attendue : un rayon plus grand implique une meilleure conductivité axiale, tandis qu'une vitesse de croissance plus élevée favorise une exploration plus rapide et plus profonde du sol, augmentant la longueur totale du système racinaire.

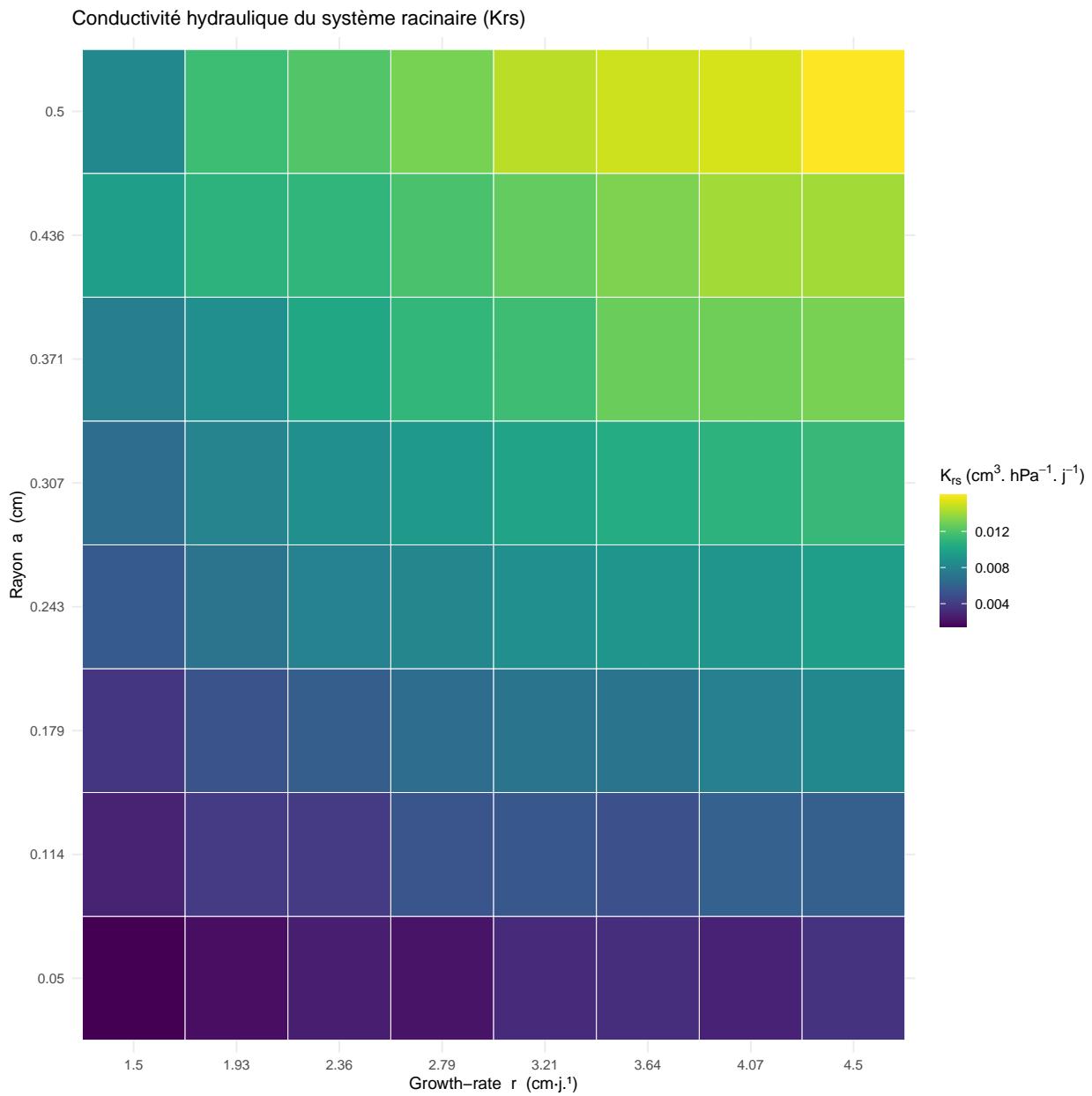
La seconde map représente la transpiration réelle simulée par MARSHAL (Tact). On retrouve ici une ten-

dance similaire à celle observée pour Krs : Tact augmente globalement avec le rayon et la vitesse de croissance. Cela confirme que des traits racinaires favorisant un système plus dense, large et profond permettent à la plante de mieux maintenir son flux d'eau, même en condition de stress hydrique.

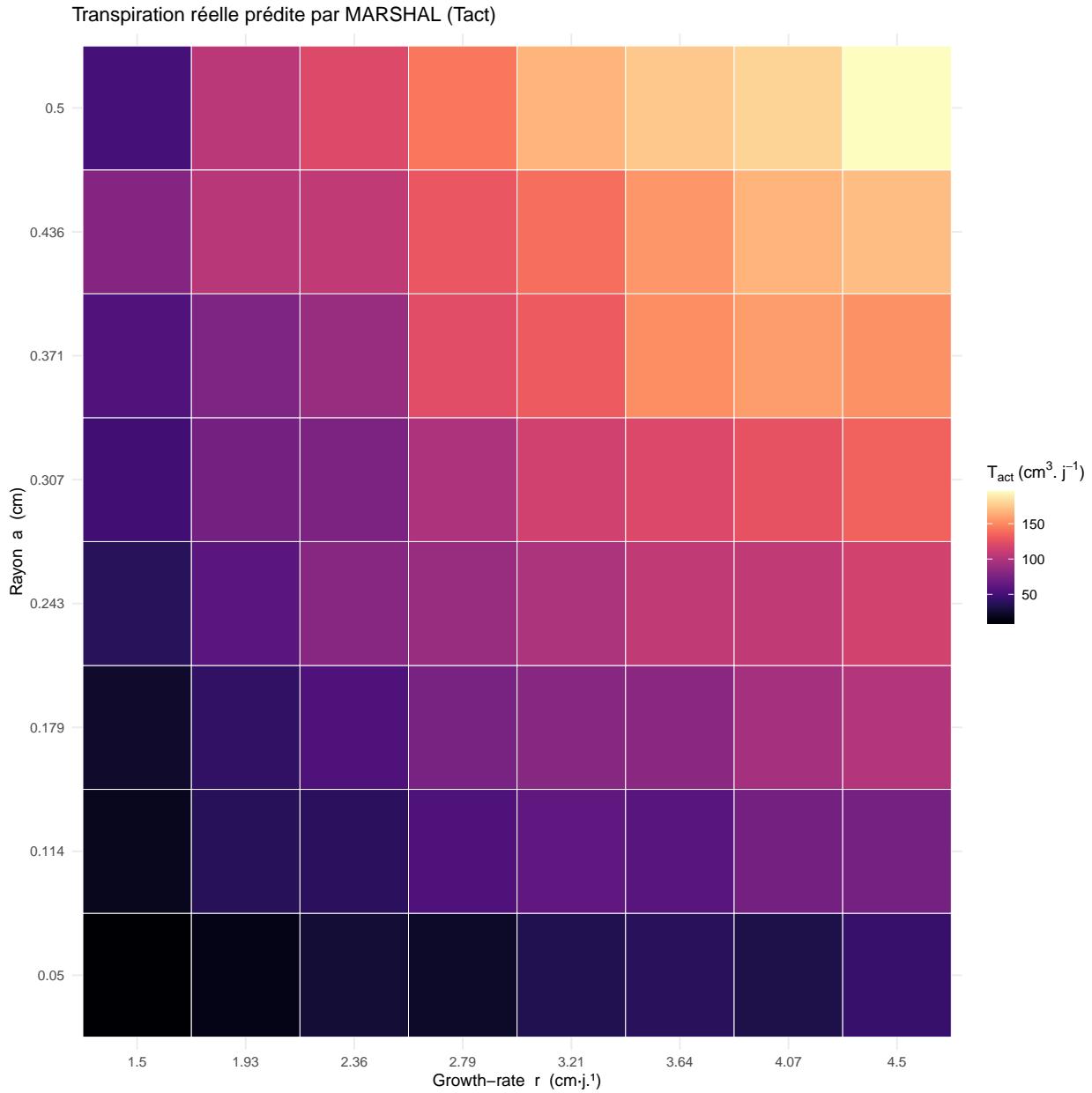
Ces résultats sont cohérents et soulignent que l'effet combiné d'un développement racinaire rapide et de racines de plus grand calibre améliore à la fois la conductivité hydraulique et la capacité d'absorption d'eau de la plante. Ces deux traits apparaissent donc comme des leviers importants à considérer dans une optique d'optimisation du système racinaire pour la tolérance à la sécheresse.

```
#####
# HEAT-MAPS (Krs et Tact)
#####
summary_results$a_f <- factor(summary_results$a,
                                levels = sort(unique(summary_results$a)))
summary_results$r_f <- factor(summary_results$r,
                                levels = sort(unique(summary_results$r)))

# --- Conductivité racinaire -----
ggplot(summary_results, aes(x = r_f, y = a_f, fill = Krs)) +
  geom_tile(color = "white") +
  scale_fill_viridis(
    name = expression(K[rs] ~"(cm"^-3*". hPa"^-1*". j"^-1*)"),
    option = "viridis") +
  labs(x = "Growth-rate r (cm·j⁻¹)",
       y = "Rayon a (cm)",
       title = "Conductivité hydraulique du système racinaire (Krs)") +
  theme_minimal()
```



```
# --- Transpiration réelle ---
ggplot(summary_results, aes(x = r_f, y = a_f, fill = Tact)) +
  geom_tile(color = "white") +
  scale_fill_viridis(
    name = expression(T[act] ~ "(cm" ^ 3 * ". j" ^ -1 * ")"),
    option = "magma") +
  labs(x = "Growth-rate r (cm·j-1)",
       y = "Rayon a (cm)",
       title = "Transpiration réelle prédicta par MARSHAL (Tact)") +
  theme_minimal()
```



### 5.3 Simulations CRootBox + Marshal pour différentes valeurs de $a$ et $r$ (tap-root)

Ce code reprend le même principe que l'analyse précédente : une grille de simulations est réalisée en faisant varier deux paramètres racinaires clés: le rayon ( $a$ ) et la vitesse de croissance ( $r$ ). Ceci est fait afin d'en observer l'effet sur le fonctionnement hydraulique de la plante.

La différence ici est que ces deux paramètres sont modifiés uniquement pour la racine principale (taproot), tandis que les racines latérales et longues latérales conservent leurs valeurs par défaut. Cela permet d'isoler l'effet des traits de la racine pivot sur l'architecture globale du système racinaire et sur ses performances hydrauliques.

```

# (re)lis les fichiers-template pour ne pas les altérer définitivement
rparam_tpl <- read_rparam("inputs/param.rparam")
pparam_tpl <- read_pparam("inputs/param.pparam")

# grilles de paramètres (pour racine primaire uniquement)
a_vec <- round(seq(0.05, 0.50, length.out = 8), 3)    # rayon racinaire [cm]
r_vec <- round(seq(1.5, 4.5, length.out = 8), 2)    # vitesse croissance [cm j-1]

# conteneur résultats plante
summary_results <- data.frame()

simid <- 0
for(a in a_vec){
  for(rgr in r_vec){

    simid <- simid + 1
    message("Simulation ", simid, "/", length(a_vec)*length(r_vec),
            " ( a = ", a, " ; r = ", rgr, " )")

    #prépare les fichiers d'entrée CRootBox
    rparam <- rparam_tpl
    pparam <- pparam_tpl

    rparam$val1[rparam$name == "Taproot" & rparam$param == "a"] <- a
    rparam$val1[rparam$name == "Taproot" & rparam$param == "r"] <- rgr

    write_rparam(rparam, "inputs/param.rparam")
    write_pparam(pparam, "inputs/param.pparam")

    # Run CRootBox
    system("inputs/crootbox.exe")

    rootsys <- fread("outputs/current_rootsystem.txt")

    # Run Marshal
    hydro <- getSUF(rootsys, conductivities, soil, psiCollar)

    # Stocke les sorties Krs & Tact

    summary_results <- rbind(summary_results,
                               data.frame(a = a,
                                          r = rgr,
                                          Krs = hydro$krs,
                                          Tact = hydro$tact))
  }
}

# Visualisation immédiate du réseau racinaire

archi <- hydro$root_system
archi$suf <- hydro$suf[, 1]

g <- ggplot(archi) +

```

```

    geom_segment(aes(x = x1, y = z1,
                      xend = x2, yend = z2,
                      colour = suf),
                  linewidth = 0.25) +
    scale_colour_viridis_c(option = "plasma",
                           name = "SUF",
                           guide = guide_colourbar(barwidth = 0.5,
                                                   barheight = 4)) +
    coord_fixed() +
    theme_void() +
    ggtitle(sprintf("a = %.2f cm | r = %.2f cm·j⁻¹", a, rgr))

  print(g)
}
}

```

### 5.3.1 Génération de Heatmaps(Krs et Tact)

Les deux heatmaps, représentant respectivement la conductivité hydraulique racinaire (Krs) et la transpiration réelle simulée (Tact), montrent un constat similaire : les variations sont faibles, peu structurées, et aucune tendance nette ne se dégage en fonction des paramètres testés (a et r).

Cela indique que modifier uniquement le rayon et la vitesse de croissance de la racine principale (taproot) n'a qu'un impact limité sur les performances hydrauliques globales de la plante. Bien que la taproot soit essentielle pour l'ancrage initial et l'exploration verticale du sol, elle ne suffit pas à elle seule à améliorer de manière significative l'absorption d'eau, en particulier dans un réseau racinaire plus complexe.

Ces résultats mettent en évidence le rôle central des racines secondaires (latérales et longues latérales), qui assurent la majeure partie de la surface d'échange avec le sol et garantissent une meilleure connectivité hydraulique. Leur contribution semble décisive dans la capacité de la plante à capter l'eau, surtout en condition de stress hydrique.

Enfin, il convient de souligner que dans les simulations où tous les types de racines sont modifiés ensemble, les paramètres a et r ont été appliqués uniformément à toutes les racines, ce qui reste une simplification. En réalité, chaque type de racine possède des propriétés propres, et il serait pertinent d'explorer, dans de futurs scénarios, des combinaisons de paramètres différencier et plus biologiquement réaliste, reflétant les vitesses de croissance et les rayons spécifiques à chaque type de racine.

```

#####
# HEAT-MAPS (Krs et Tact)
#####
summary_results$a_f <- factor(summary_results$a,
                                 levels = sort(unique(summary_results$a)))
summary_results$r_f <- factor(summary_results$r,
                                 levels = sort(unique(summary_results$r)))

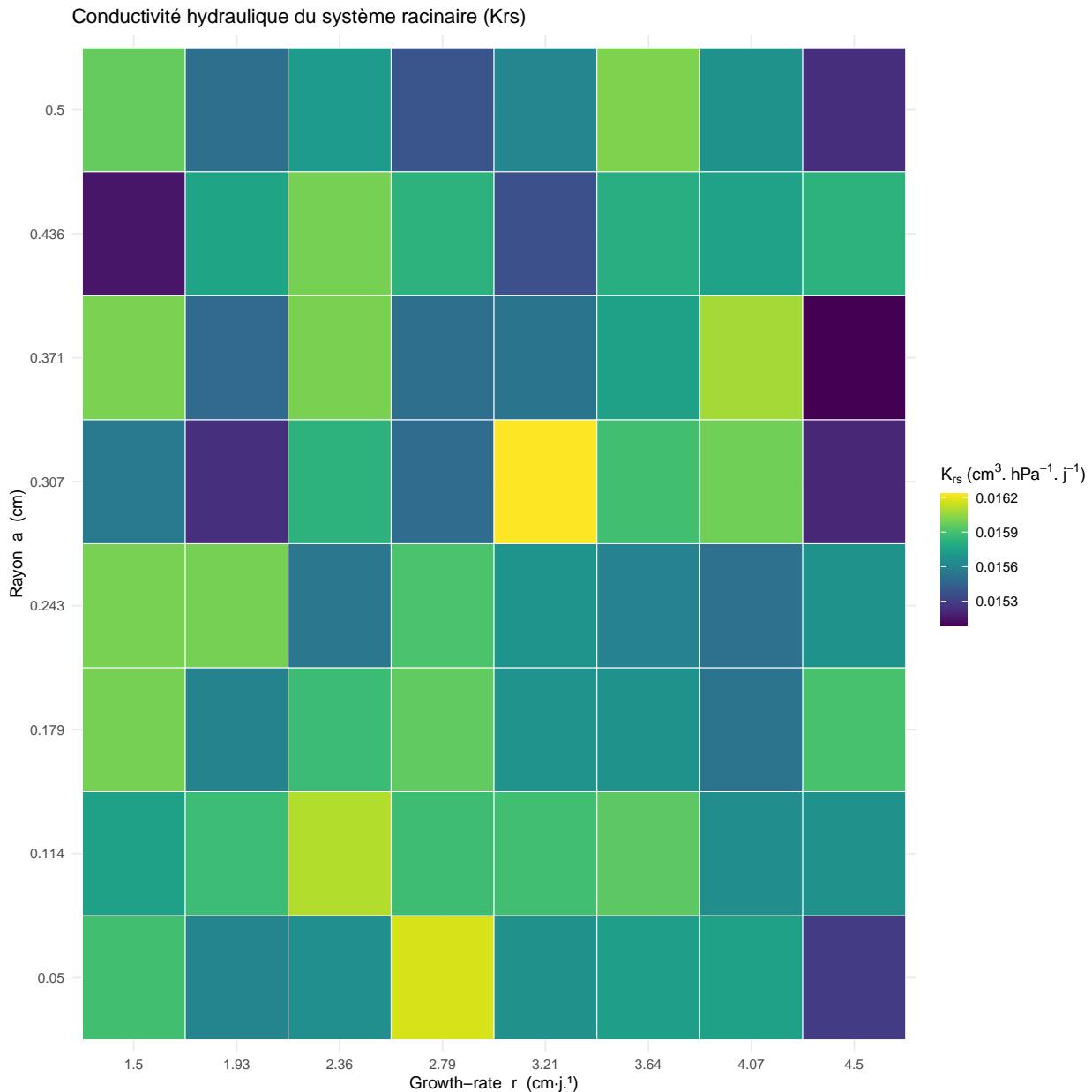
# --- Conductivité racinaire -----
ggplot(summary_results, aes(x = r_f, y = a_f, fill = Krs)) +
  geom_tile(color = "white") +
  scale_fill_viridis(
    name = expression(K[rs] ~ "(cm" ^ 3 * ". hPa" ^ -1 * ". j" ^ -1 * ")"),
    option = "viridis") +
  labs(x = "Growth-rate r (cm·j⁻¹)",
       y = "Rayon a (cm)",
       title = "Conductivité hydraulique racinaire")

```

```

    title = "Conductivité hydraulique du système racinaire (Krs)" +
theme_minimal()

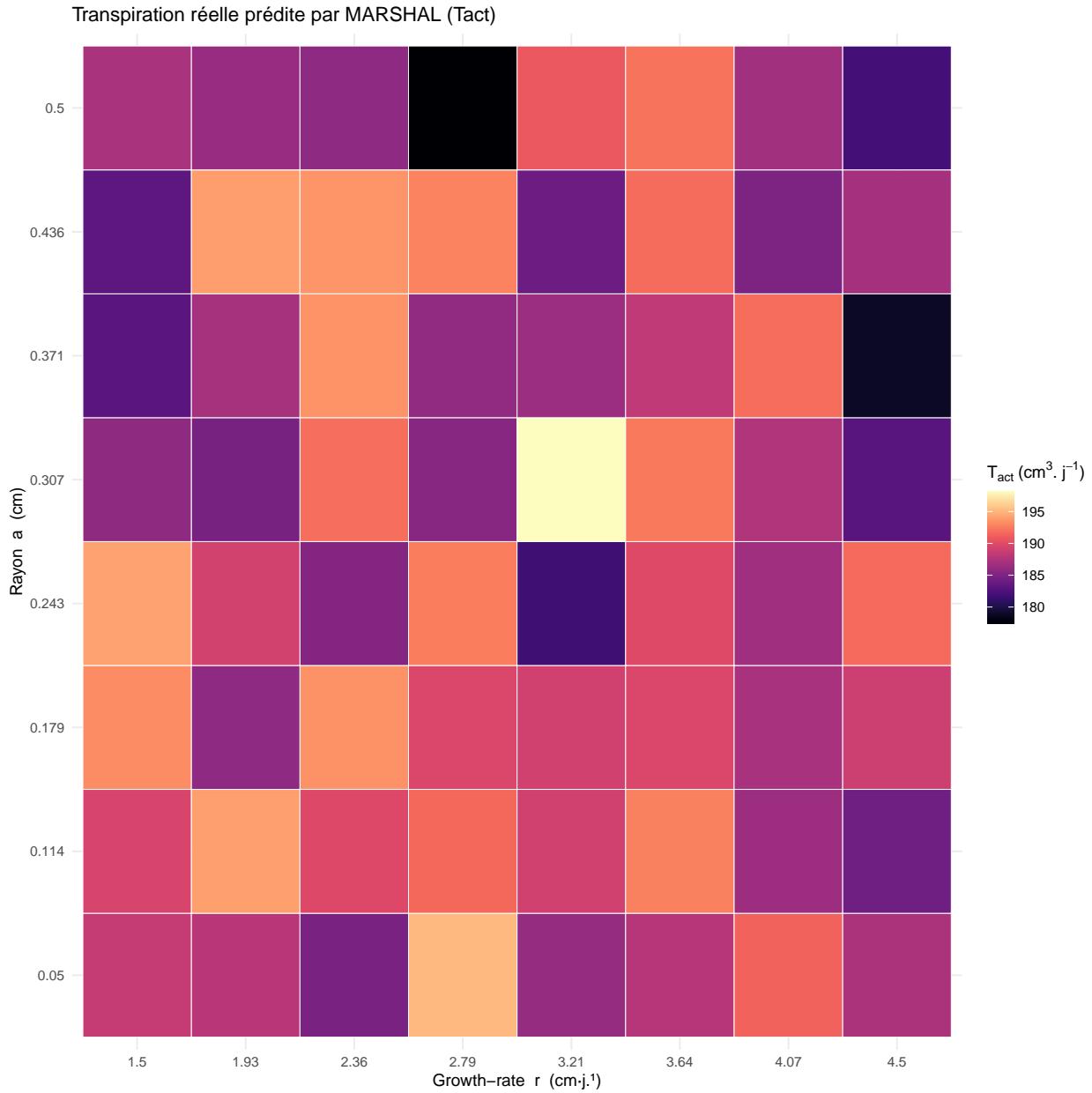
```



```

# --- Transpiration réelle ---
ggplot(summary_results, aes(x = r_f, y = a_f, fill = Tact)) +
  geom_tile(color = "white") +
  scale_fill_viridis(
    name = expression(T[act] ~ "(cm" ^ 3 * ". j" ^ -1 * ")"),
    option = "magma") +
  labs(x = "Growth-rate  $r$  (cm·j-1)",
       y = "Rayon  $a$  (cm)",
       title = "Transpiration réelle prédicta par MARSHAL (Tact)") +
  theme_minimal()

```



#### 5.4 Identifier les Tact min et max

Dans cette partie du travail, l'objectif est d'analyser visuellement deux architectures racinaires extrêmes, correspondant aux valeurs minimales et maximales de transpiration réelle (Tact) obtenues dans les simulations, afin d'en évaluer les implications en situation de sécheresse.

En effet, si une transpiration élevée peut refléter une bonne absorption d'eau, elle peut également être préjudiciable dans un contexte de disponibilité hydrique limitée, en entraînant un épuisement plus rapide des ressources du sol. Inversement, une transpiration trop faible peut traduire un fonctionnement inefficace. Il est donc pertinent de comparer ces deux extrêmes pour mieux comprendre l'effet des traits racinaires sur l'équilibre hydrique de la plante.

Plus précisément, on visualise ici :

- d'une part, l'architecture racinaire ayant permis la transpiration réelle (Tact) la plus élevée,
- d'autre part, celle ayant conduit à la valeur la plus faible de Tact.

Pour cela, deux simulations sont relancées avec les paramètres correspondants ( $a$  et  $r$ ), dans un scénario où seule la taproot est modifiée. Les structures racinaires obtenues sont ensuite affichées côte à côté, colorées en fonction de la répartition du SUF, ce qui permet d'observer visuellement l'intensité d'absorption d'eau dans chaque segment.

La figure ci-dessus compare ainsi :

- à gauche, le cas Tact MAX, obtenu avec une racine principale fine et une vitesse de croissance élevée ;
- à droite, le cas Tact MIN, correspondant à une racine principale plus épaisse et plus lente.

-> (Ces résultats ont été obtenu pour un run mais change d'un run à l'autre)

On observe que les deux architectures restent structurellement proches, avec des différences visuelles peu marquées. Cette faible sensibilité morphologique et fonctionnelle se traduit aussi par des valeurs de Tact relativement proches, malgré des paramètres très contrastés.

De plus, les résultats peuvent varier d'une exécution à l'autre, ce qui renforce l'idée que modifier uniquement les caractéristiques de la taproot n'a qu'un impact limité sur le fonctionnement global du système racinaire. Ces observations confirment les tendances identifiées dans les heatmaps précédentes : une amélioration significative de la performance hydraulique nécessiterait probablement une adaptation conjointe des traits racinaires sur l'ensemble du système, et non sur la taproot seule.

```
## lignes "meilleur" et "pire" Tact
best_row <- summary_results[ which.max(summary_results$Tact), ] # Tact MAX
worst_row <- summary_results[ which.min(summary_results$Tact), ] # Tact MIN

simulate_archi <- function(a, rgr, tact_label, label_txt){
  # (a) prépare les inputs
  rparam <- rparam_tpl
  pparam <- pparam_tpl
  rparam$val1[rparam$name == "Taproot" & rparam$param == "a"] <- a
  rparam$val1[rparam$name == "Taproot" & rparam$param == "r"] <- rgr
  write_rparam(rparam, "inputs/param.rparam")
  write_pparam(pparam, "inputs/param.pparam")

  # (b) CRootBox + MARSHAL
  system("inputs/crootbox.exe")
  rootsys <- fread("outputs/current_rootsystem.txt")
  hydro   <- getSUF(rootsys, conductivities, soil, psiCollar)

  # (c) data.frame pour ggplot
  archi <- hydro$root_system
  archi$suf <- hydro$suf[, 1]

  g <- ggplot(archi) +
    geom_segment(aes(x = x1, y = z1,
                     xend = x2, yend = z2,
                     colour = suf),
                 linewidth = 0.30) +
    scale_colour_viridis_c(option = "plasma", name = "SUF") +
```

```

coord_fixed() +
theme_void() +
ggtitle(sprintf("%s (%.1f cm³·j⁻¹)\n a = %.2f cm | r = %.2f cm·j⁻¹",
label_txt, tact_label, a, rgr))

list(plot = g,
      rootsys = rootsys,
      suf = archi$suf)
}

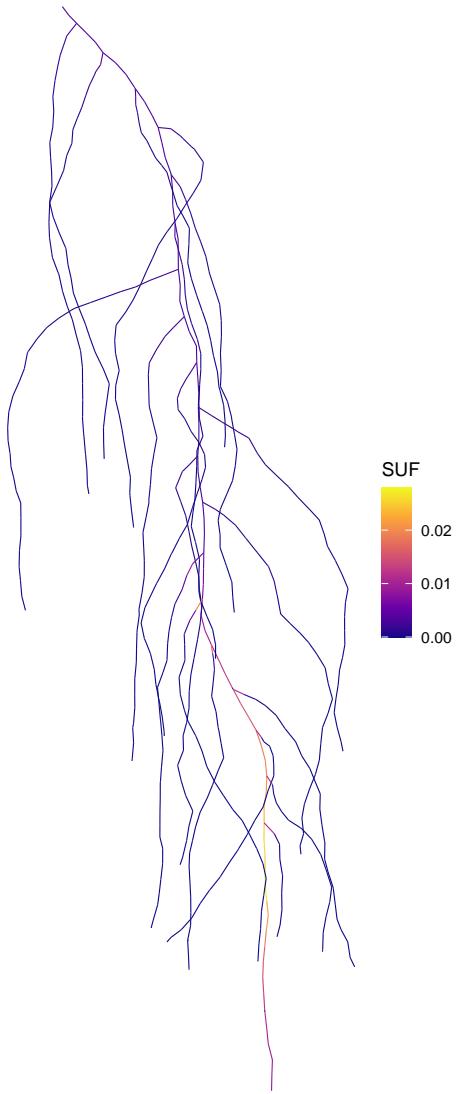
## 7.3 exécute pour Tact MAX et Tact MIN
sim_max <- simulate_archi(best_row$a , best_row$r ,
                           tact_label = best_row$Tact,
                           label_txt = "Tact MAX")
sim_min <- simulate_archi(worst_row$a, worst_row$r,
                           tact_label = worst_row$Tact,
                           label_txt = "Tact MIN")

## 7.4 palette commune
suf_range <- range(c(sim_max$suf, sim_min$suf), na.rm = TRUE)
sim_max$plot <- sim_max$plot +
  scale_colour_viridis_c(option = "plasma", limits = suf_range, name = "SUF")
sim_min$plot <- sim_min$plot +
  scale_colour_viridis_c(option = "plasma", limits = suf_range, name = "SUF")

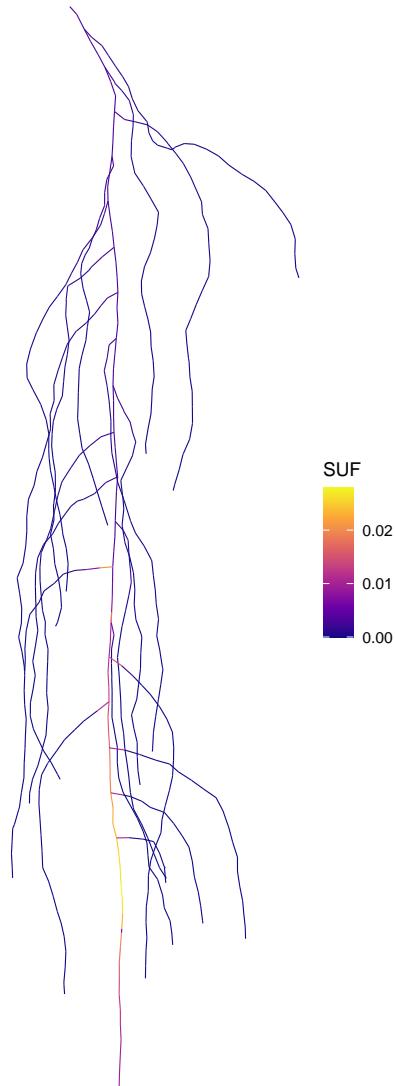
## 7.5 affichage côte à côte
grid.arrange(sim_max$plot, sim_min$plot, ncol = 2)

```

Tact MAX ( $198.1 \text{ cm}^3 \cdot \text{j}^{-1}$ )  
 $a = 0.31 \text{ cm} \quad | \quad r = 3.21 \text{ cm} \cdot \text{j}^{-1}$



Tact MIN ( $177.4 \text{ cm}^3 \cdot \text{j}^{-1}$ )  
 $a = 0.50 \text{ cm} \quad | \quad r = 2.79 \text{ cm} \cdot \text{j}^{-1}$



```
## 7.6 réseaux racinaires réutilisables
rootsys_max <- sim_max$rootsys    # pour le cas Tact MAX
rootsys_min <- sim_min$rootsys    # pour le cas Tact MIN
```

## 5.5 Tentative de couplage entre Marshal et le modèle de culture Apsim

Dans cette section, l'objectif est de préparer un couplage entre le modèle hydraulique MARSHAL et un modèle de culture APSIM. Le lien possible entre ces deux modèles repose sur une variable commune : la transpiration de la plante.

MARSHAL simule la transpiration réelle (Tact) que la plante peut soutenir, en fonction de l'architecture racinaire et du potentiel hydrique du sol ( $\Psi$ ). Or, les modèles culturaux comme APSIM expriment l'état hydrique du sol en stock d'eau disponible, et non en  $\Psi$ . Il est donc nécessaire de convertir ce stock en un

potentiel compatible avec MARSHAL, à l'aide de la courbe de Van Genuchten, qui permet d'établir une correspondance entre teneur en eau et potentiel hydrique, en fonction du type de sol. Cette méthode est inspirée des scripts partagés sur le GitHub du cours.

Méthodologie L'approche suivie ici consiste à :

- a) Choisir une valeur de stock d'eau initial correspondant à une situation de déficit hydrique (ici, 9 cm).
- b) Estimer la teneur en eau moyenne associée à ce stock sur une certaine profondeur.
- c) Inverser la fonction de Van Genuchten pour en déduire le potentiel hydrique  $\Psi$  correspondant.
- d) Répéter cette opération pour trois types de sol contrastés : sableux, limoneux et argileux, à partir des paramètres proposés par Carsel & Parrish (1988).

Résultats La figure obtenue illustre la relation entre le stock d'eau initial et le potentiel hydrique  $\Psi$ , pour chaque type de sol. On observe une grande sensibilité de cette relation à la texture du sol. Dans les sols argileux ou limoneux, une réduction modérée du stock d'eau entraîne une augmentation très forte de  $\Psi$ , avec des valeurs parfois irréalistes. Dans ces cas, le stock de 9 cm (ligne pointillée bleue) ne se retrouve même plus dans la plage représentée, car il correspondrait à des potentiels excessivement négatifs, peu exploitables par les racines.

En revanche, dans le sol sableux, la courbe reste plus progressive et stable. Le même stock d'eau initial y donne un  $\Psi$  situé dans une plage cohérente avec les simulations MARSHAL, ce qui rend le couplage numériquement possible et biologiquement réaliste.

```
# Define soils params
soil_params_limoneux <- list(
  theta_s = 0.43,
  theta_r = 0.078,
  alpha   = 0.036,
  n       = 1.56
)

soil_params_argileux <- list(
  theta_s = 0.38,
  theta_r = 0.098,
  alpha   = 0.008,
  n       = 1.09
)

soil_params_sableux <- list(
  theta_s = 0.43,
  theta_r = 0.045,
  alpha   = 0.145,
  n       = 2.68
)

soil_params <- list(
  Limoneux = soil_params_limoneux,
  Argileux = soil_params_argileux,
  Sableux  = soil_params_sableux
)

#default psi collar
```

```

psi0 = -15000
theta0 = vanGenuchten(psi0, soil_params_sableux, unit='hPa')

print(paste0("Theta0 = ", theta0))

# stock d'eau initial dans le modèle de culture pour une situation en déficit hydrique
stock_initial_deficit = 9

# Gamme de stocks d'eau initiaux (en cm) pour le plot
stocks <- seq(45, 1, length.out = 50)

df <- map_dfr(names(soil_params), function(type) {
  par      <- soil_params[[type]]
  theta0  <- vanGenuchten(psi0, par, unit = "hPa")
  theta1  <- (stocks - 20 * theta0) / 100
  psi1    <- sapply(theta1, function(t)
    if(t > par$theta_r) inverse_vangenuchten(t, par, "hPa") else NA)
  data.frame(stock = stocks, psi = psi1, sol = type)
})

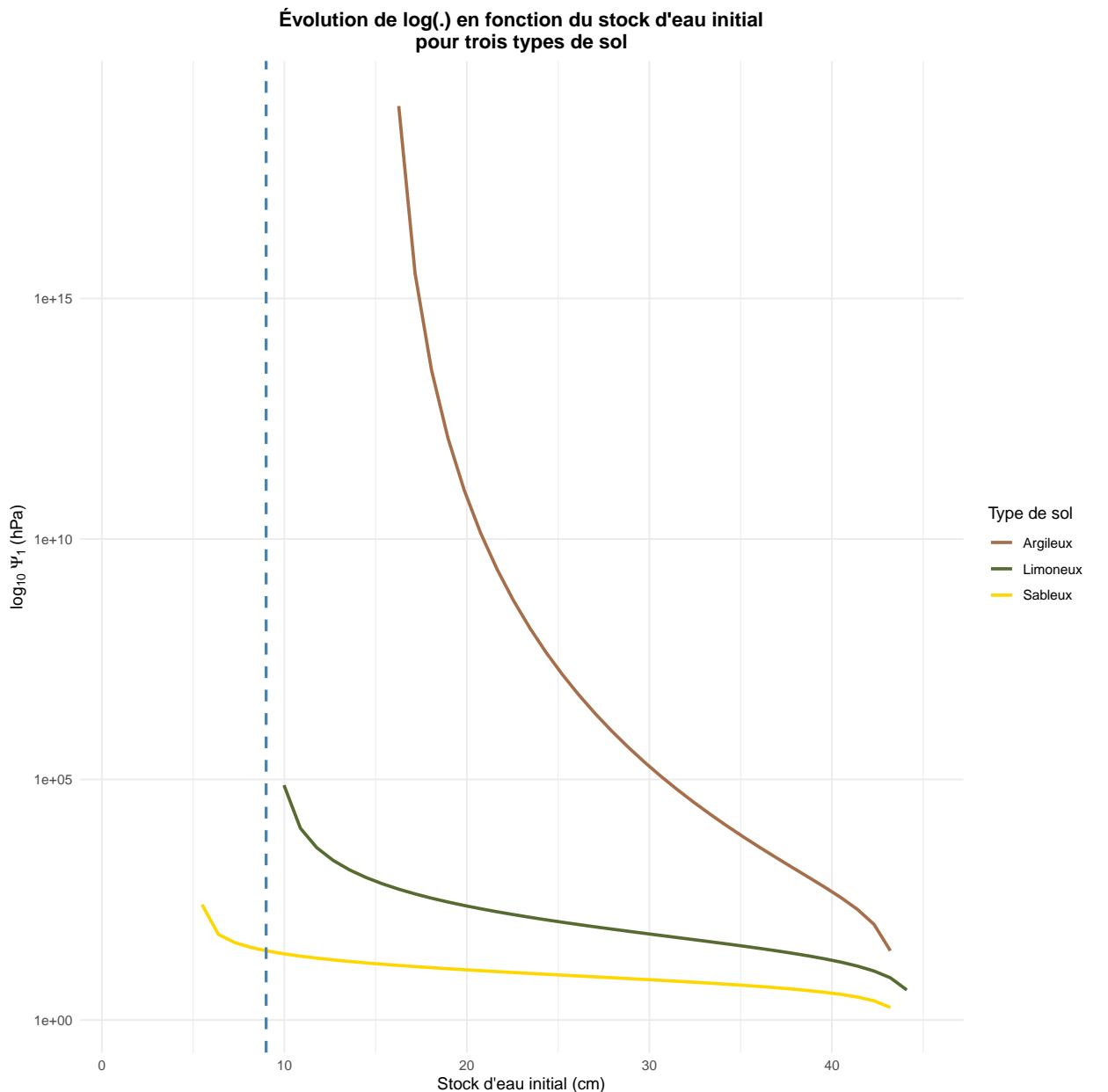
ggplot(df, aes(x = stock, y = psi, colour = sol)) +
  geom_line(linewidth = 1, na.rm = TRUE) +
  # échelle logarithmique explicite
  scale_y_log10(
    name = expression(log[10]~Psi[1]~"(hPa)"),
    breaks = 10^(seq(0, 19, by = 5)),    # repères tous les 5 ordres de grandeur
    minor_breaks = NULL
  ) +
  # couleurs (sable = jaune, limon = olive, argile = clay-brown)
  scale_colour_manual(
    name   = "Type de sol",
    values = c(
      Sableux  = "#FFD700",
      Limoneux = "#556B2F",
      Argileux = "#A66E4A"
    )
  ) +
  # ligne verticale pour le stock du scénario déficit
  geom_vline(xintercept = stock_initial_deficit,
             linetype   = "dashed",
             colour     = "steelblue",
             linewidth  = 0.9) +
  labs(
    x      = "Stock d'eau initial (cm)",
    title = "Évolution de log(Ψ) en fonction du stock d'eau initial\npour trois types de sol"
  ) +
  theme_minimal() +
  theme(

```

```

    plot.title = element_text(face = "bold", hjust = 0.5)
)

```



- Résultats :

La figure obtenue illustre la relation entre le stock d'eau initial et le potentiel hydrique  $\Psi$ , pour chaque type de sol. On observe une grande sensibilité de cette relation à la texture du sol. Dans les sols argileux ou limoneux, une réduction modérée du stock d'eau entraîne une augmentation très forte de  $\Psi$ , avec des valeurs irréalistes. Dans ces cas, le stock de 9 cm (ligne pointillée bleue) ne se retrouve même plus dans la plage représentée.

En revanche, dans le sol sableux, la courbe reste plus progressive et stable. Le même stock d'eau initial y donne un  $\Psi$  situé dans une plage cohérente avec les simulations MARSHAL, ce qui rend le couplage numériquement possible.

- Limites et choix du sol :

Cette expérience montre que le couplage MARSHAL–APSIM est théoriquement envisageable via la transpiration, à condition de pouvoir convertir correctement les états hydriques. Toutefois, en raison de la forte sensibilité du potentiel  $\Psi$  au stock d'eau dans certains sols, le lien peut devenir instable ou non réaliste. C'est pourquoi, pour la suite du projet, le choix a été fait de travailler exclusivement avec un sol sableux, dont les propriétés permettent un couplage entre les modèles.

### 5.5.1 Lier les sorties MARSHAL et Apsim via la transpiration

Dans cette partie, l'objectif est de lier les sorties du modèle hydraulique MARSHAL au modèle de culture simplifié inspiré d'APSIM, en utilisant comme variable commune la transpiration journalière.

Le code utilisé est une version adaptée du script partagé sur le GitHub du cours pour APSIM. La seule modification apportée concerne l'ajout de deux arguments supplémentaires dans la fonction de simulation : “use\_marshal = FALSE, rootsystem = NULL”

Ces options permettent d'activer ou non l'appel à MARSHAL, et de spécifier le système racinaire utilisé dans la simulation.

Une difficulté majeure réside dans le fait que les deux modèles expriment la transpiration dans des unités différentes :

- MARSHAL donne une transpiration en  $\text{cm}^3/\text{plante/jour}$  ;
- APSIM l'exprime en  $\text{mm/jour}$  ( $\text{d'eau par m}^2$  de surface).

Pour convertir les sorties de MARSHAL dans le format d'APSIM, on utilise la densité de 8 plantes/ $\text{m}^2$ , ce qui donne :

$$\text{Transpiration}_{\text{mm/jour}} = \text{Transpiration}_{\text{cm}^3/\text{plante/jour}} \times 0.008$$

La valeur de 8 plantes par mètre carré est issue des recommandations agronomiques publiées par GÖWEIL, qui préconise une densité de semis comprise entre 6 et 9 plantes/ $\text{m}^2$  selon les conditions hydriques.

La simulation a été menée selon deux approches :

Sans MARSHAL : le modèle de culture calcule la transpiration comme le minimum entre l'offre et la demande en eau, en fonction du sol, de la radiation et du déficit de pression de vapeur (VPD).

Avec MARSHAL : pour chaque jour, le stock d'eau disponible dans le sol est converti en un potentiel hydrique ( $\Psi$ ) via l'inversion de la fonction de Van Genuchten. Ce  $\Psi$  est ensuite injecté dans MARSHAL, qui retourne une transpiration réelle (Tact), convertie en  $\text{mm/jour}$  pour être utilisée dans le modèle culturale.

Dans un couplage plus complet, il serait logique de retenir la plus limitante des deux à chaque pas de temps, ce qui reviendrait à prendre le minimum entre la transpiration potentielle climatique et la capacité réelle d'absorption de la plante. Cependant, dans cette première approche, nous avons choisi de les comparer séparément, en simulant deux rendements indépendants. Cela permet d'évaluer l'effet direct des contraintes racinaires sur la croissance, à climat et sol identiques.

```
#####
# Chargement des paramètres
#####

# Paramètres de la culture -----
culture_params <- list(
```

```

RUE           = 1.25,   # Radiation Use Efficiency [g/MJ]
TEc          = 9,       # Coefficient d'efficience de la transpiration
VPR          = 20,      # Vitesse production de racines [mm/jour]
CroissPotLAI = 0.1,    # Croissance potentielle du LAI
k             = 0.45,    # Coefficient d'extinction de la lumière
LAI_initial   = 1.5,
Biomasse_initiale = 45
)

# Paramètres du sol (3 horizons, épaisseur, etc.) ----

soil_params <- data.frame(
  Horizon     = c(1, 2, 3),
  Epaisseur   = c(300, 300, 300), # mm
  li          = c(50, 50, 50), # Limite inférieure d'eau
  ls          = c(100, 100, 100), # Limite supérieure d'eau
  #es         = c(100, 100, 100), # Eau disponible dans sol (= sw)
  es_h        = c(40, 30, 30), # Eau disponible dans sol par horizon ; es_h1+es_h2+es_h3 = es
  kl          = c(0.07, 0.07, 0.07) # Taux d'extraction [mm/jour]
)

# Expansion foliaire (table O/D - CEF)-----

expansion_foliaire <- data.frame(
  OD  = c(0.5, 1.5, 4),
  CEF = c(0, 1, 1)
)

# Paramètres météo (ex. VPD_Frac)-----

meteo_params <- list(
  VPD_Frac = 1.0
)

# Vérification-----

# Affiche la liste des paramètres de la culture
culture_params

# Affiche le data frame du sol
soil_params

# Affiche la table d'expansion foliaire
expansion_foliaire

#####
# Facteurs externes
#####

facteur_externe <- data.frame(

```

```

# Jours (30 à 60)
Jours = c(
  30, 31, 32, 33, 34, 35,
  36, 37, 38, 39, 40, 41,
  42, 43, 44, 45, 46, 47,
  48, 49, 50, 51, 52, 53,
  54, 55, 56, 57, 58, 59, 60
),

# Radiation (MJ/m2)
Radiation = c(
  27, 27, 14, 24, 23, 21,
  23, 25, 17, 14, 26, 26,
  10, 26, 30, 27, 27, 29,
  27, 26, 25, 23, 14, 25,
  22, 20, 22, 24, 28, 25, 25
),

# Tmax (°C)
Tmax = c(
  32.3, 31.0, 26.6, 26.0, 26.6, 29.5,
  30.8, 32.5, 32.3, 25.2, 27.8, 28.2,
  27.3, 28.6, 28.6, 28.3, 27.6, 31.0,
  35.0, 34.3, 31.2, 32.7, 29.9, 30.8,
  31.2, 28.4, 27.7, 31.4, 33.0, 33.2, 32.7
),

# Tmin (°C)
Tmin = c(
  16.4, 15.8, 15.6, 10.0, 11.7, 13.0,
  16.5, 13.8, 16.7, 16.7, 15.8, 12.8,
  17.3, 11.3, 13.7, 13.4, 13.7, 12.2,
  14.7, 20.4, 15.5, 17.9, 18.4, 16.0,
  16.1, 18.0, 14.9, 16.0, 16.2, 17.2, 18.3
),

# VPDoobs (hPa)
VPDoobs = c(
  19, 17, 16, 14, 12, 15,
  18, 19, 23, 21, 15, 14,
  20, 16, 17, 17, 15, 15,
  16, 17, 18, 21, 22, 17,
  18, 20, 18, 19, 19, 20, 21
)

# Affiche la liste pour vérifier
print(facteur_externe)

# Fonction pour calculer saturated vapour pressure
svp <- function(T) { # saturated vapour pressure [kPa]
  6.1078 * exp(17.269 * T / (237.3 + T)) * 0.10
}

```

```
}
```

```
# Application à Tmax
facteur_externe$svpTmax <- svp(facteur_externe$Tmax) # pression saturante à Tmax

# Application à Tmin
facteur_externe$svpTmin <- svp(facteur_externe$Tmin) # pression saturante à Tmin

# Choisir la valeur de VPDfrac (par défaut on prend 0.75)
VPDfrac <- 0.75

# Fonction pour calculer VPDcalc
calc_VPDcalc <- function(svpTmax, svpTmin, VPDfrac) {
  VPDfrac * (svpTmax - svpTmin)*10
}

# Ajout de la colonne VPDcalc
facteur_externe$VPDcalc <- calc_VPDcalc(
  svpTmax = facteur_externe$svpTmax,
  svpTmin = facteur_externe$svpTmin,
  VPDfrac = VPDfrac
)

#####
# Modèle
#####

simulate <- function(facteur_externe, soil_params, culture_params, expansion_foliaire, use_marshall = FALSE)
n_days <- nrow(facteur_externe)

# Vecteurs pour l'eau disponible par horizon
ES1 <- numeric(n_days)
ES2 <- numeric(n_days)
ES3 <- numeric(n_days)

# Initialisation des réserves d'eau pour chaque horizon
ES1[1] <- soil_params$es_h[1]
ES2[1] <- soil_params$es_h[2]
ES3[1] <- soil_params$es_h[3]

# Initialisation de la LAI dynamique
LAI_dyn <- numeric(n_days)
LAI_dyn[1] <- culture_params$LAI_initial

# Préparation d'un data frame pour stocker les résultats totaux
results <- data.frame(
  Jour = facteur_externe$Jours,
  Tot_ES = numeric(n_days),
  Pot_Supply = numeric(n_days),
  Pot_Demand = numeric(n_days),
  Transpiration = numeric(n_days),
  LAI = numeric(n_days)
```

```

)
results$LAI[1] <- LAI_dyn[1]

# Fonction pour calculer l'effet d'expansion foliaire
leaf_exp_effect <- function(sdratio, expansion_foliaire) {
  # Si le ratio est en-deçà du premier seuil, on renvoie la première valeur,
  # si au-dessus du dernier, on renvoie la dernière,
  # sinon on effectue une interpolation linéaire.
  if(sdratio <= expansion_foliaire$OD[1]) {
    print(expansion_foliaire$OD[1])
    print(expansion_foliaire$CEF[1])
    return(expansion_foliaire$CEF[1])
  } else if (sdratio >= tail(expansion_foliaire$OD,1)) {
    return(tail(expansion_foliaire$CEF,1))
  } else {
    for(j in 1:(nrow(expansion_foliaire)-1)){
      if(sdratio >= expansion_foliaire$OD[j] && sdratio < expansion_foliaire$OD[j+1]){
        return(expansion_foliaire$CEF[j] +
               (sdratio - expansion_foliaire$OD[j]) *
               (expansion_foliaire$CEF[j+1] - expansion_foliaire$CEF[j]) /
               (expansion_foliaire$OD[j+1] - expansion_foliaire$OD[j]))
      }
    }
  }
}

# Boucle de simulation pour chaque jour
for (i in 1:n_days) {
  DAS <- facteur_externe$Jours[i]
  # La profondeur racinaire effective est limitée par la somme des épaisseurs
  profondeur_totale <- sum(soil_params$Epaisseur)
  rdepth <- min(DAS * culture_params$VPR, profondeur_totale)

  # Calcul des offres potentielles pour chaque horizon
  of1 <- ifelse(rdepth >= soil_params$Epaisseur[1], 1, rdepth / soil_params$Epaisseur[1]) * ES1[i] *
  of2 <- ifelse(rdepth <= soil_params$Epaisseur[1], 0,
                ifelse(rdepth > soil_params$Epaisseur[1] + soil_params$Epaisseur[2],
                       1, (rdepth - soil_params$Epaisseur[1]) / soil_params$Epaisseur[2])) * ES2[i] *
  of3 <- ifelse(rdepth <= (soil_params$Epaisseur[1] + soil_params$Epaisseur[2]), 0,
                (rdepth - soil_params$Epaisseur[1] - soil_params$Epaisseur[2]) / soil_params$Epaisseur[2])

  Pot_Supply <- of1 + of2 + of3

  # Calcul de l'effet lumineux : LI = 1 - exp(-k * LAI_dyn)
  li <- 1 - exp(-culture_params$k * LAI_dyn[i])

  # Calcul de la demande potentielle
  rad <- facteur_externe$Radiation[i]
  VPDcalc <- facteur_externe$VPDcalc[i]
  Pot_Demand <- rad * li * culture_params$RUE * (VPDcalc / 10) / culture_params$TEc

  if(use_marshall) {
    stock_eau = ES1[i] + ES2[i] + ES3[i]
  }
}

```

```

psi0 = -15000
theta0 = vanGenuchten(psi0, soil_params_sableux, unit='hPa')
theta1 = (stock_eau-200*theta0)/1000
psi1 = inverse_vangenuchten(theta1, soil_params_sableux, unit='hPa')

soil_day <- data.frame(
  id  = 1:4,
  z   = c(0, -40,-80,-120),
  psi = c(psi0, -psi1, -psi1, -psi1)
)

hydraulics <- getSUF(rootsystem, conductivities, soil_day, psiCollar)

transpiration = hydraulics$tact #cm^3/plante.jour
transpiration = transpiration * 0.008 #mm/jour
} else {
  # Transpiration journalière APSIM
  transpiration <- min(Pot_Supply, Pot_Demand)
}

# Stockage des résultats du jour
results$Tot_ES[i] <- ES1[i] + ES2[i] + ES3[i]
results$Pot_Supply[i] <- Pot_Supply
results$Pot_Demand[i] <- Pot_Demand
results$Transpiration[i] <- transpiration
results$LAI[i] <- LAI_dyn[i]

# Calcul du ratio offre/demande (si demande > 0)
sdratio <- if (Pot_Demand > 0) Pot_Supply / Pot_Demand else 0

# Calcul de l'effet d'expansion foliaire
if (is.na(sdratio)) {
  leaf_effect <- 0 #Si sdratio vaut NaN, on set leaf_effect à 0
} else {
  leaf_effect <- leaf_exp_effect(sdratio, expansion_foliaire)
}

# delta LAI = leaf_effect * CroissPotLAI
delta_LAI <- leaf_effect * culture_params$CroissPotLAI

# Mise à jour de LAI pour le jour suivant
if(i < n_days) {
  LAI_dyn[i + 1] <- LAI_dyn[i] + delta_LAI
}

# Mise à jour des réserves d'eau pour le jour suivant
if (i < n_days) {
  if(Pot_Supply > 0) {
    ES1[i + 1] <- ES1[i] - (of1 / Pot_Supply) * transpiration
    ES2[i + 1] <- ES2[i] - (of2 / Pot_Supply) * transpiration
    ES3[i + 1] <- ES3[i] - (of3 / Pot_Supply) * transpiration
  } else {
}

```

```

        ES1[i + 1] <- ES1[i]
        ES2[i + 1] <- ES2[i]
        ES3[i + 1] <- ES3[i]
    }
}

return(results)
}

results_APSSIM <- simulate(facteur_externe, soil_params, culture_params,
                           expansion_foliaire,
                           use_marshal = FALSE)

results_MARSHAL <- simulate(facteur_externe, soil_params, culture_params,
                           expansion_foliaire,
                           use_marshal = TRUE,
                           rootsystem = rootsys_min)

```

### 5.5.2 Résultats comparés – Simulation MARSHAL vs APSSIM (cas Min)

La figure ci-dessus compare les résultats issus du modèle de culture avec MARSHAL et sans MARSHAL (version APSSIM), en conditions identiques de climat et de sol. Quatre indicateurs clés sont représentés : l'eau disponible, la transpiration cumulée, l'indice de surface foliaire (LAI) et la biomasse cumulée.

- 1) Eau disponible On observe une diminution progressive de l'eau disponible dans les deux cas. Cependant, le modèle APSSIM “classique” consomme l'eau plus rapidement, ce qui s'explique par une transpiration plus élevée (voir graphique suivant). À l'inverse, la simulation avec MARSHAL conserve plus d'eau dans le sol.
- 2) Transpiration cumulée La transpiration calculée à partir de MARSHAL est systématiquement plus faible que celle d'APSSIM, et ce tout au long de la simulation. ça reflète la limitation hydraulique imposée par la structure racinaire, qui empêche la plante d'atteindre tout le potentiel “climatique” de transpiration. En d'autres termes, la plante ne peut pas tirer autant d'eau qu'elle le voudrait, car son système racinaire ne le permet pas.
- 3) LAI (Index foliaire) Jusqu'au jour 45, les deux courbes sont proches. Ensuite, le LAI augmente légèrement plus dans MARSHAL. Cela s'explique par une meilleure conservation de l'eau dans le sol, qui réduit le stress hydrique au fil du temps. Ainsi, malgré une transpiration plus faible (moins de), la plante maintient une croissance foliaire active, ce qui est cohérent avec une stratégie efficace en condition de sécheresse.
- 4) Biomasse cumulée De manière similaire, la biomasse est d'abord équivalente, puis devient plus élevée avec MARSHAL. Ce résultat confirme qu'une plante qui respire moins peut finalement croître davantage, en particulier quand les ressources en eau sont limitées. Cela illustre l'intérêt de modéliser la contrainte racinaire dans les simulations de croissance en contexte de stress.

Limites du modèle Plusieurs hypothèses simplificatrices doivent être soulignées :

Le système racinaire utilisé dans MARSHAL est fixe tout au long de la simulation. Or, dans la réalité, les racines se développent au cours du temps, ce qui augmenterait progressivement la capacité d'absorption. Comme CRooBox ne permet pas de simuler cette dynamique de manière continue, la transpiration calculée par MARSHAL est probablement sous-estimée.

Dans cette simulation, nous avons choisi de comparer séparément les deux modèles, sans appliquer de logique combinée. Pourtant, comme mentionné précédemment, un couplage plus complet prendrait le minimum entre la demande atmosphérique et la capacité racinaire. Mais ici, ce choix ne changerait rien : MARSHAL est toujours plus limitant que la version APSIM, donc c'est MARSHAL qui serait sélectionné dans un couplage intégré.

```

calc_outputs <- function(res){
  res$I <- 1 - exp(-culture_params$k * res$LAI)

  res$Biomasse_1 <- (res$Pot_Supply * culture_params$TEc) /
    (facteur_externe$VPDcalc / 10)

  res$Biomasse_2 <- facteur_externe$Radiation *
    culture_params$RUE * res$I

  res$O_D <- ifelse(res$Pot_Demand==0, 0,
    res$Pot_Supply / res$Pot_Demand)

  res$Biomasse_reelle <- ifelse(res$O_D > 1,
    res$Biomasse_2, # limitation lumière
    res$Biomasse_1) # limitation eau

  res$Biomasse_cumulee <- culture_params$Biomasse_initiale +
    cumsum(res$Biomasse_reelle)

  res$Transp_cum <- cumsum(res$Transpiration)

  res
}

results_APSIM <- calc_outputs(results_APSIM)
results_MARSHAL <- calc_outputs(results_MARSHAL)

#####
# GRAPHIQUES COMPARATIFS (APSIM vs MARSHAL)
#####

par(mfrow = c(2,2)) # 4 graphiques

## Eau disponible
plot(results_APSIM$Jour, results_APSIM$Tot_ES, type = "o", pch = 16,
  col = "blue", xlab = "Jours", ylab = "Eau disponible (mm)",
  main = "Eau disponible")
lines(results_MARSHAL$Jour, results_MARSHAL$Tot_ES,
  type = "o", pch = 17, col = "red")
legend("topright", legend=c("APSIM", "MARSHAL"),
  pch=c(16,17), col=c("blue","red"), bty="n")

## Transpiration cumulée
plot(results_APSIM$Jour, results_APSIM$Transp_cum, type = "o", pch = 16,
  col = "blue", xlab = "Jours", ylab = "Transp. cumulée (mm)",
  main = "Transpiration cumulée")
lines(results_MARSHAL$Jour, results_MARSHAL$Transp_cum,
  type = "o", pch = 17, col = "red")

```

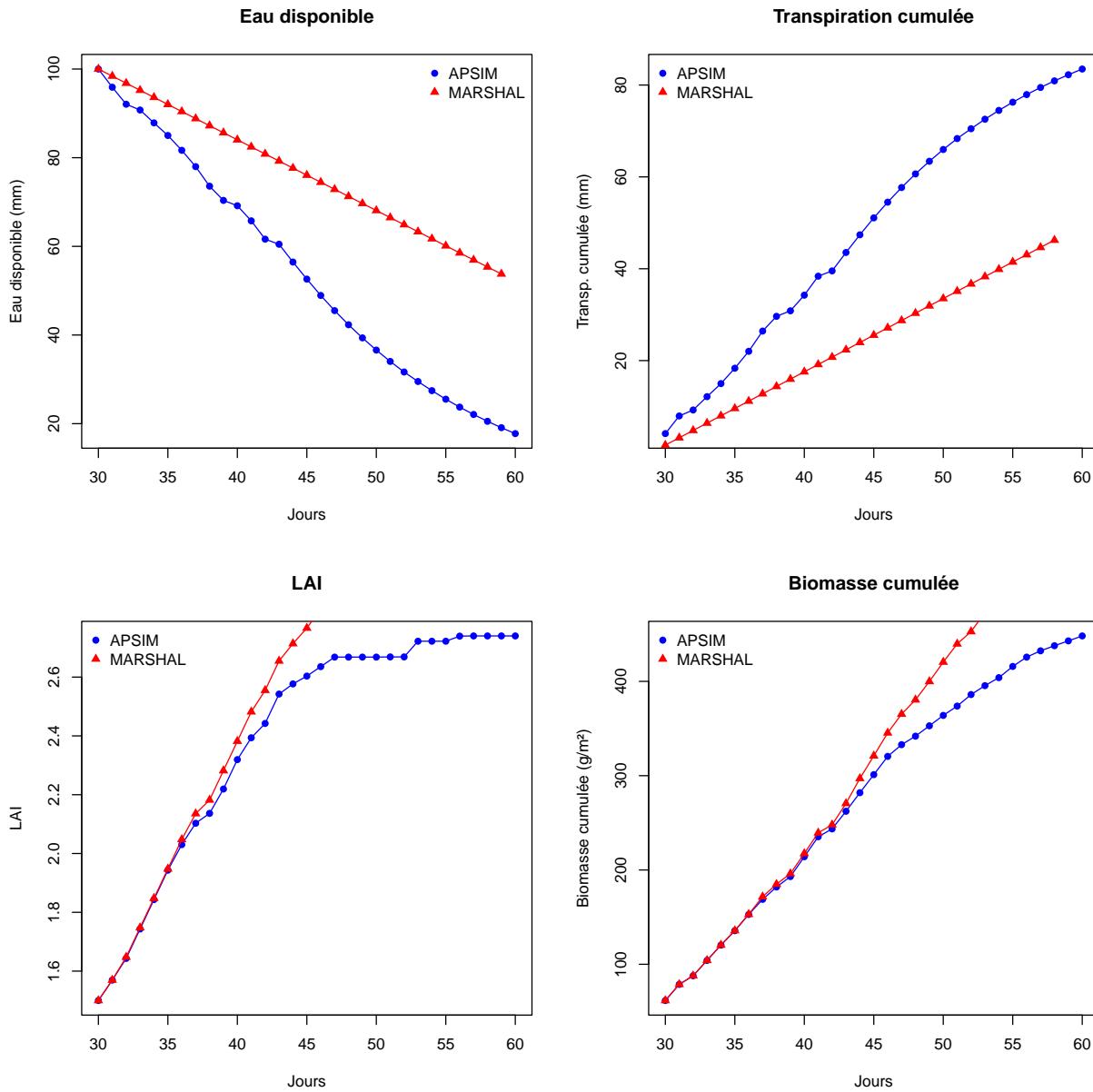
```

legend("topleft", legend=c("APSIM", "MARSHAL"),
      pch=c(16,17), col=c("blue","red"), bty="n")

## LAI
plot(results_APSSIM$Jour, results_APSSIM$LAI, type = "o", pch = 16,
      col = "blue", xlab = "Jours", ylab = "LAI", main = "LAI")
lines(results_MARSHAL$Jour, results_MARSHAL$LAI,
      type = "o", pch = 17, col = "red")
legend("topleft", legend=c("APSIM", "MARSHAL"),
      pch=c(16,17), col=c("blue","red"), bty="n")

## Biomasse cumulée
plot(results_APSSIM$Jour, results_APSSIM$Biomasse_cumulee, type = "o", pch = 16,
      col = "blue", xlab = "Jours", ylab = "Biomasse cumulée (g/m²)",
      main = "Biomasse cumulée")
lines(results_MARSHAL$Jour, results_MARSHAL$Biomasse_cumulee,
      type = "o", pch = 17, col = "red")
legend("topleft", legend=c("APSIM", "MARSHAL"),
      pch=c(16,17), col=c("blue","red"), bty="n")

```



```
par(mfrow = c(1,1)) # réinitialise la disposition graphique
```

### 5.5.3 Résultats comparés – Simulation MARSHAL vs APSIM (cas Max)

Cette seconde simulation reprend le même protocole de couplage, mais en utilisant la racine simulée avec la transpiration maximale (Tact MAX) dans MARSHAL. L'objectif est de voir si une architecture racinaire modifie significativement la croissance par rapport à la simulation précédente (faite avec Tact MIN).

Les résultats observés ici sont très similaires à ceux obtenus dans la simulation précédente :

- la transpiration cumulée reste légèrement plus faible dans MARSHAL que dans APSIM, mais la différence est minime,
- la biomasse finale et le LAI suivent quasiment les mêmes trajectoires dans les deux approches,

- la réduction de l'eau disponible dans le sol est très comparable.

Ce résultat est logique et attendu, car dans les simulations de MARSHAL précédentes, les différences entre la racine “Tact MIN” et “Tact MAX” étaient déjà faibles, du fait que seuls les paramètres de la taproot étaient modifiés. Or, comme cela a été montré plus tôt, la taproot seule n'a pas un impact suffisant sur l'efficacité hydraulique globale du système racinaire pour entraîner des différences marquées en termes de transpiration ou de rendement.

Cette simulation confirme que la structure racinaire, pour influencer significativement la croissance en condition de sécheresse, doit être modifiée dans son ensemble. Modifier uniquement la racine principale ne suffit pas à créer un avantage clair dans le modèle de culture. Les tendances restent stables entre les deux cas.

```

results_APSIM2 <- simulate(facteur_externe, soil_params, culture_params,
                           expansion_foliaire,
                           use_marshall = FALSE)

results_MARSHAL2 <- simulate(facteur_externe, soil_params, culture_params,
                           expansion_foliaire,
                           use_marshall = TRUE,
                           rootsystem = rootsys_max)

calc_outputs <- function(res){
  res$I <- 1 - exp(-culture_params$k * res$LAI)

  res$Biomasse_1 <- (res$Pot_Supply * culture_params$TEc) /
    (facteur_externe$VPDcalc / 10)

  res$Biomasse_2 <- facteur_externe$Radiation *
    culture_params$RUE * res$I

  res$O_D <- ifelse(res$Pot_Demand==0, 0,
                     res$Pot_Supply / res$Pot_Demand)

  res$Biomasse_reelle <- ifelse(res$O_D > 1,
                                res$Biomasse_2, # limitation lumière
                                res$Biomasse_1) # limitation eau

  res$Biomasse_cumulee <- culture_params$Biomasse_initiale +
    cumsum(res$Biomasse_reelle)

  res$Transp_cum <- cumsum(res$Transpiration)

  res
}

results_APSIM2 <- calc_outputs(results_APSIM2)
results_MARSHAL2 <- calc_outputs(results_MARSHAL2)

#####
# GRAPHIQUES COMPARATIFS (APSIM vs MARSHAL)
#####

par(mfrow = c(2,2)) # 4 graphiques

```

```

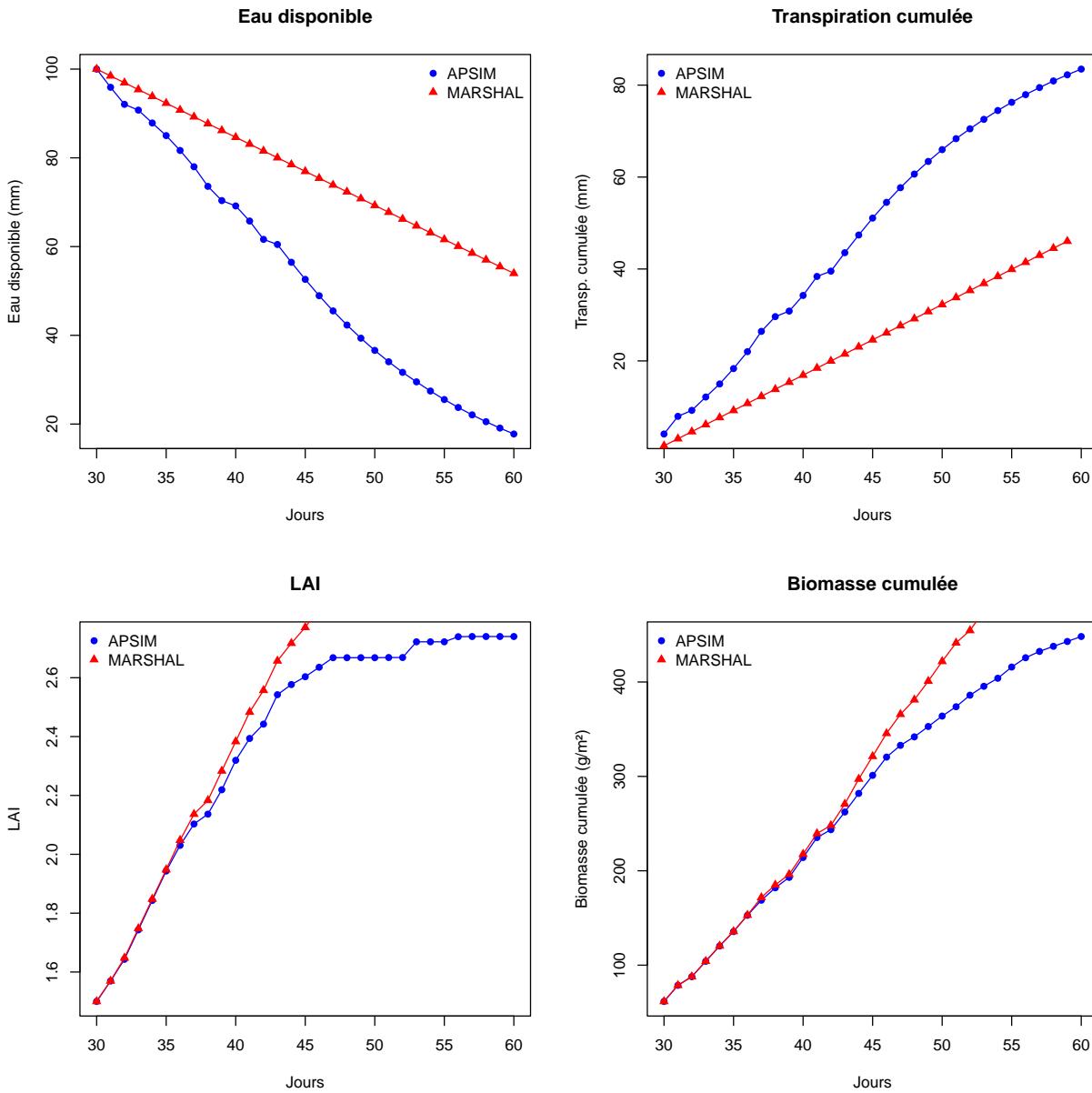
## Eau disponible
plot(results_APSSIM2$Jour, results_APSSIM2$Tot_ES, type = "o", pch = 16,
      col = "blue", xlab = "Jours", ylab = "Eau disponible (mm)",
      main = "Eau disponible")
lines(results_MARSHAL2$Jour, results_MARSHAL2$Tot_ES,
      type = "o", pch = 17, col = "red")
legend("topright", legend=c("APSSIM", "MARSHAL"),
      pch=c(16,17), col=c("blue","red"), bty="n")

## Transpiration cumulée
plot(results_APSSIM2$Jour, results_APSSIM2$Transp_cum, type = "o", pch = 16,
      col = "blue", xlab = "Jours", ylab = "Transp. cumulée (mm)",
      main = "Transpiration cumulée")
lines(results_MARSHAL2$Jour, results_MARSHAL2$Transp_cum,
      type = "o", pch = 17, col = "red")
legend("topleft", legend=c("APSSIM", "MARSHAL"),
      pch=c(16,17), col=c("blue","red"), bty="n")

## LAI
plot(results_APSSIM2$Jour, results_APSSIM2$LAI, type = "o", pch = 16,
      col = "blue", xlab = "Jours", ylab = "LAI", main = "LAI")
lines(results_MARSHAL2$Jour, results_MARSHAL2$LAI,
      type = "o", pch = 17, col = "red")
legend("topleft", legend=c("APSSIM", "MARSHAL"),
      pch=c(16,17), col=c("blue","red"), bty="n")

## Biomasse cumulée
plot(results_APSSIM2$Jour, results_APSSIM2$Biomasse_cumulee, type = "o", pch = 16,
      col = "blue", xlab = "Jours", ylab = "Biomasse cumulée (g/m²)",
      main = "Biomasse cumulée")
lines(results_MARSHAL2$Jour, results_MARSHAL2$Biomasse_cumulee,
      type = "o", pch = 17, col = "red")
legend("topleft", legend=c("APSSIM", "MARSHAL"),
      pch=c(16,17), col=c("blue","red"), bty="n")

```



```
par(mfrow = c(1,1))    # réinitialise la disposition graphique
```

## 6 Conclusion générale

L'objectif de ce travail était d'explorer l'influence de la structure racinaire sur le rendement d'une culture de maïs en conditions de sécheresse, en couplant un modèle racinaire hydraulique (MARSHAL) avec un modèle de culture inspiré d'APSIM.

En comparant les simulations avec et sans prise en compte de la structure racinaire, on observe que les rendements restent globalement similaires, en particulier jusqu'au jour 45. Ce n'est qu'en fin de simulation que la version couplée à MARSHAL montre une biomasse légèrement supérieure, malgré une transpiration systématiquement plus faible tout au long de la période. Cela s'explique par une consommation d'eau plus

modérée, qui permet à la plante de maintenir sa croissance plus longtemps en fin de stress, sans épuiser les réserves trop rapidement.

Entre les deux extrêmes testés dans MARSHAL (la racine avec Tact MIN et celle avec Tact MAX), les différences sont minimes, tant en termes de transpiration que de rendement. Ce résultat est cohérent avec ce qui a été observé dans les simulations hydrauliques précédentes : en effet, modifier uniquement la racine principale (taproot) n'est pas suffisant pour générer une variation marquée du fonctionnement global du système racinaire. L'effet reste limité, ce qui explique l'impact très modéré sur la croissance simulée.

Ces résultats montrent que le couplage MARSHAL–APSIM fonctionne techniquement et permet d'explorer des interactions racines/croissance. Mais ils soulignent aussi que pour observer un effet significatif du système racinaire sur le rendement, il faut agir à l'échelle de l'ensemble du réseau racinaire, et non uniquement sur un sous-ensemble de traits.

## 7 Références

- Bänziger, M., Edmeades, G. O., Beck, D., & Bellon, M. (2000). *Breeding for drought and nitrogen stress tolerance in maize: From theory to practice*. CIMMYT, Mexico. <https://repository.cimmyt.org/xmlui/handle/10883/551>
- FAO. (2021). *World Food and Agriculture – Statistical Yearbook 2021*. Food and Agriculture Organization of the United Nations, Rome. <https://doi.org/10.4060/cb4477en>
- Feng, X., Jia, L., Cai, Y., Guan, H., Zheng, D., Xu, P., ... & Liu, Z. (2022). ABA-inducible DEEPER ROOTING 1 improves adaptation of maize to water deficiency. *Plant Biotechnology Journal*, 20(10), 1882–1894. <https://doi.org/10.1111/pbi.13890>
- Gleason, S. M., Cooper, M., Wiggans, D. R., Bliss, C. A., Romay, M. C., Gore, M. A., ... & Comas, L. H. (2019). Stomatal conductance, xylem water transport, and root traits underpin improved performance under drought and well-watered conditions across a diverse panel of maize inbred lines. *Field Crops Research*, 234, 80–93. <https://doi.org/10.1016/j.fcr.2019.02.001>
- Gonzalez, A., Boudsocq, S., & Page, M. (2021). Modéliser pour comprendre et prévoir les interactions sol-plante-atmosphère. *Innovations Agronomiques*, 84, 1–12. <https://hal.inrae.fr/hal-03113862>
- GÖWEIL. (s.d.). *Maïs ensilé en période de sécheresse – ce qu'il faut savoir*. <https://www.goeweil.com/fr/detail/mais-ensile-secheresse/>
- IPCC. (2022). *Climate Change 2022: Impacts, Adaptation and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press. <https://www.ipcc.ch/report/ar6/wg2>
- Klein, S. P., Schneider, H. M., Perkins, A. C., Brown, K. M., & Lynch, J. P. (2020). Multiple integrated root phenotypes are associated with improved drought tolerance. *Plant Physiology*, 183(3), 1011–1025. <https://doi.org/10.1104/pp.20.00211>
- Lynch, J. P. (2013). Steep, cheap and deep: an ideotype to optimize water and N acquisition by maize root systems. *Annals of Botany*, 112(2), 347–357. <https://doi.org/10.1093/aob/mcs293>
- Messina, C. D., Hammer, G. L., McLean, G., Cooper, M., van Oosterom, E. J., & Tardieu, F. (2022). Modelling crop improvement in a world of elevated CO<sub>2</sub> and climate change. *Field Crops Research*, 277, 108421. <https://doi.org/10.1016/j.fcr.2021.108421>
- Messina, C. D., Technow, F., Tang, T., Totir, R., Gho, C., & Cooper, M. (2015). Leveraging biological insight and environmental variation to improve phenotypic prediction: integrating crop growth models (CGM) with whole genome prediction (WGP). *European Journal of Agronomy*, 70, 70–82. <https://doi.org/10.1016/j.eja.2015.06.003>

Postma, J. A., Kuppe, C., Owen, M. R., Mellor, N., Griffiths, M., Bennett, M. J., & Lynch, J. P. (2017). OpenSimRoot: widening the scope and application of root architectural models. *New Phytologist*, 215(3), 1274–1286. <https://doi.org/10.1111/nph.14641>

Rishmawi, L., Bauget, F., Protto, V., Bauland, C., Nacry, P., & Maurel, C. (2023). Natural variation of maize root hydraulic architecture underlies highly diverse water uptake capacities. *Plant Physiology*, 193(2), 1406–1420. <https://doi.org/10.1093/plphys/kiad213>

Spinoni, J., Naumann, G., Vogt, J., & Barbosa, P. (2018). The biggest drought events in Europe from 1950 to 2012. *Geophysical Research Letters*, 42(13), 5485–5493. <https://doi.org/10.1002/2015GL063891>

Carsel, R. F., & Parrish, R. S. (1988). Developing joint probability distributions of soil water retention characteristics. *Water Resources Research*, 24(5), 755–769. <https://doi.org/10.1029/WR024i005p00755>