THÈSE

Pour l'obtention du grade de
DOCTEUR DE L'UNIVERSITÉ DE POITIERS
UFR des sciences fondamentales et appliquées
Unité de recherche pluridisciplinaire prairies et plantes fourragères - URP3F
(Diplôme National - Arrêté du 25 mai 2016)

École doctorale : Chimie, Écologie, Géosciences et AgroSciences Théodore Monod (Poitiers)
Secteur de recherche : Physiologie Végétale

Présentée par :
Arthur Couturier

Impact de la diversité architecturale intraspécifique chez le ray-grass anglais sur la variabilité du Rc:Rs directionnel

Directeur(s) de Thèse :
Abraham Escobar-Gutiérrez, Elzbieta Frak, Didier Combes

Soutenue le 07 novembre 2022 devant le jury

Jury :

Président	**Sophie Camélio**	Professeure, ENSIP, Université de Poitiers
Rapporteur	**Nathalie Leduc**	Professeur, IRSH, Institut Agro, Université d'Angers
Rapporteur	**Pascal Barla**	Chargé de recherche INRIA, Manao, Talence
Membre	**Frédéric Boudon**	Chargé de recherche INRIA, Cirad, AGAP, Montpellier

Pour citer cette thèse :
Arthur Couturier. *Impact de la diversité architecturale intraspécifique chez le ray-grass anglais sur la variabilité du Rc:Rs directionnel* [En ligne]. Thèse Physiologie Végétale. Poitiers : Université de Poitiers, 2022. Disponible sur Internet <http://theses.univ-poitiers.fr>

THÈSE

pour l'obtention du Grade de

DOCTEUR DE L'UNIVERSITÉ DE POITIERS

(Faculté des Sciences Fondamentales et Appliquées)

(Diplôme National – Arrêté du 25 mai 2016)

École Doctorale :

Chimie Ecologie Géosciences Agrosciences « Rosalind Franklin »

Secteur de Recherche :

Biologie des populations et écologie

Présentée par :

Arthur Couturier

Impact de la diversité architecturale intraspécifique chez le ray-grass anglais sur la variabilité du R:FR directionnel.

Directeur de Thèse :

Abraham Escobar-Gutiérrez

Co-encadrants :

Elzbieta Frak

Didier Combes

Soutenue le 07 novembre 2022

devant la Commission d'Examen

JURY

Nathalie LEDUC	Rapporteuse	Professeur d'Université d'Angers
Pascal BARLA	Rapporteur	Chargé de Recherche, INRIA
Sophie CAMELIO	Examinatrice	Professeur d'Université, ENSIP
Michael CHELLE	Examinateur	Directeur de Recherche, INRAE
Fréderic BOUDON	Examinateur	Chargé de Recherche, INRIA
Abraham ESCOBAR GUTIERREZ	Directeur de Thèse	Directeur de Recherche, INRAE
Elzbieta FRAK	Co-Directrice	Chargée de Recherche, INRAE
Didier COMBES	Co-Directeur	Ingénieur de Recherche, INRAE

Arthur COUTURIER, 2022. **Impact de la diversité architecturale intraspécifique chez le ray-grass anglais sur la variabilité du R:FR directionnel.**

INRAE, UR4, URP3F, Equipe d'Ecophysiologie des Plantes Fourragères. BP 6, F-86600 Lusignan, France.

Résumé

Pour mieux comprendre et modéliser l'influence du signal R:FR sur le développement d'une plante au sein d'un couvert quel que soit le stade de développement des plantes voisines il faut connaître sa variabilité. Dans ce travail de thèse, il s'agissait plus particulièrement de prendre en compte l'aspect directionnel de ce signal ici appelé zeta (ζ) avec un double objectif : i) caractériser sa variabilité dans le rayonnement réfléchi dans un faible angle de vue à partir de plantes de ray-grass d'architectures contrastées et ii) tester la capacité des modèles existants à rendre compte de cette variabilité. Pour cela un dispositif de mesure du rayonnement directionnel dans un faible angle de vue a été conçu de façon à capter le signal renvoyé par des plantes et/ou organes de petite taille. Des expérimentations ont été mises en place afin de caractériser le ζ correspondant au rayonnement réfléchi par les plantes dans différentes configurations de densité, de distance entre la plante et le capteur et de conditions de ciel. Nos résultats montrent une forte variabilité de ζ directionnel essentiellement liée à l'état de développement des plantes et aux conditions de ciel. Les modèles testés ne rendent compte de cette variabilité que partiellement et nécessitent une meilleure prise en compte des sources de variabilité du ζ directionnel identifiées par la mesure. Toutefois, ces résultats de modélisation du ζ directionnel restent encourageants et de plus amples tests d'amélioration doivent être réalisés afin d'aboutir à un outil permettant de mieux comprendre les mécanismes d'intégration des signaux par les plantes et donc de la photomorphogenèse.

Mots clés : Interactions lumineuses, Modélisation, FSPM, Capteur directionnel, Qualité de la lumière, Photomorphogenèse, Prairies, Lolium perenne

---

Arthur COUTURIER, 2022. **Impact of intraspecific architectural diversity in rye-grass on directional R:FR variability.**

Abstract

To better understand and model the influence of the R:FR signal on the development of a plant within a canopy, regardless of the development stage of neighbouring plants, it is necessary to know its variability. In this thesis, the aim was more specifically to take into account the directional aspect of this signal, here called zeta (ζ), with a twofold objective: i) to characterise its variability in the radiation reflected at a low angle of view from ryegrass plants with contrasting architectures and ii) to test the capacity of existing models to account for this variability. For this purpose, a device for measuring directional radiation at a small angle of view was designed to capture the signal reflected by small plants and/or organs. Experiments were set up to characterise the ζ corresponding to the radiation reflected by plants in different configurations of density, distance between the plant and the sensor and sky conditions. Our results show a high variability of directional ζ mainly related to the developmental state of the plants and the sky conditions. The models tested only partially account for this variability and require a better consideration of the sources of variability in directional ζ identified by the measurement. However, these modelling results of the directional ζ remain promising and further improvement tests need to be carried out in order to come up with a tool to better understand the mechanisms of signal integration by plants and thus of photomorphogenesis.

Remerciements

Merci à la Fondation Xavier-Bernard et à la région Nouvelle Aquitaine pour avoir financé cette thèse.

Je tiens à remercier mes encadrants pour leur suivi quasi quotidien au cours de cette thèse : Abraham Escobar-Gutiérrez pour ses conseils ; Elzbieta Frak pour ses encouragements constants et sa capacité à voir le meilleur de chaque situation ; et Didier Combes pour m'avoir poussé tout au long de ma thèse et de m'avoir constamment fait des retours pertinents sur mon travail. Votre encadrement m'a permis d'aller jusqu'au bout et je sais que je n'ai pas été le plus facile des doctorants.

Jean-Louis Durand qui a permis l'instruction de cette thèse et Philippe Barre pour sa finalisation administrative.

Je tiens également à remercier mon comité de thèse pour leurs retours et leur bienveillance : Sabine Demotes-Mainard, Angélique Christophe et Christian Fournier.

Merci à Pascal Vernoux et surtout à Frank Gelin pour leurs capacités incroyables d'adaptation et de création de dispositif permettant de réaliser nos expérimentations, même pour nos idées les plus farfelues et pour m'avoir initié au babyfoot.

Merci à Cédric Perrot et Eric Roy pour leur imagination débordante capable de répondre à tous nos besoins expérimentaux.

Merci à Annie Eprinchard et Marianne Van Peteghem pour m'avoir plus qu'aidé sur les différentes manips.

Merci à Isabelle Boissou, Nathalie Bonnet et Sophie Bergeronneau pour leur savoir-faire administratif et leur patience.

Merci aux doctorant·e·s que j'ai croisés durant ces années et avec qui j'ai pu partager mes joies et mes angoisses : Simon Rouet, Sylvain Edouard, Vincent Colas, Victoria Acker. Merci à Mau pour les discussions en voiture et aux pauses cafés. Merci à Stéphanie Mahieu pour les découvertes BD et le covoiturage.

Merci à Baptistin Bodin pour son travail réalisé en stage et son aide apportée à mes expérimentations. Merci à l'ensemble des stagiaires et mains d'œuvre que j'ai rencontrés au cours de ma thèse et à leur bonne humeur.

Merci à l'ensemble des membres de l'écophy pour toutes ces pauses cafés et plus largement un énorme merci à l'ensemble des collègues de Lusignan.

Merci à l'ensemble des intervenant·e·s des différentes formations et séminaires que j'ai suivis durant cette thèse.

Merci à mes colocs : Justine, Valentine, Tom, Maxime, Noémie, Clé, vous avez été mes rayons de soleil au cours de ces années. Merci Béa pour ton soutien infaillible et les raclées que tu m'a mises au babyfoot. Merci aux copaines du gang des colocs pour tous les moments partagés ensemble. Merci à tout·e·s les camarades de luttes.

Merci à ma famille pour leur compréhension et leur patience.

Liste des abréviations

DOY jour de l'année

FOV champ de vision (°)

FR rouge sombre, longueurs d'ondes proches de 730 nm

FSPM modèle de plante structure-fonction

LAD densité de la surface foliaire

LAI indice de surface foliaire

LED diode électroluminescente

LIDF distribution de l'inclinaison des feuilles

PAR rayonnement photosynthétiquement actif, entre 400 nm et 700 nm

PO propriétés optiques (absorbance, réflectance, transmittance)

PPFD densité de flux de photons ($\mu mol.m^{-2}.s^{-1}$)

TrPPFD densité de flux de photons transmis (%)

R rouge clair, longueurs d'ondes proches de 660 nm

R:FR rapport entre le rayonnement compris entre 600 nm et 700 nm et le rayonnement compris entre 700 nm et 800 nm

SAS syndrome de l'évitement de l'ombre

SOC standardized overcast sky, modèle de ciel

SVC surface de plante capturée par le modèle de capteur directionnel simple

UOC uniform overcast sky, modèle de ciel

UTC temps universel coordonné

VGL modèle Virtual GrassLand

ζ rapport entre le rayonnement compris entre 655 nm et 665 nm et le rayonnement compris entre 725 nm et 735 nm

Table des matières

I. Introduction....7

II. État de l'art....11

1. Les prairies....12
1.1. Culture des prairies en France....12
1.2. Impact des pratiques agricoles sur la diversité des prairies temporaires....13
1.3. Morphologie des espèces fourragères....14

2. Les interactions plante-environnement....18
2.1. Interactions plante-plante....18
2.2. Interactions lumineuses....19
2.3. Interactions plante-environnement dans les prairies semées....22

3. La photomorphogenèse....23
3.1. Perception de l'environnement lumineux par les plantes via les photorécepteurs....23
3.2. Photomorphogenèse des plantes soumises à des variations de FR....25
3.3. Anticipation....28
3.4. Photomorphogenèse dans les prairies....28

4. Mesure de la photomorphogenèse....29
4.1. Mesure de la morphogenèse....29
4.2. Mesure de l'environnement lumineux....29
4.3. Conditions environnementales de la mesure du R:FR....30
4.4. Méthode de mesure du R:FR....31

5. Modélisation des interactions plantes-environnement....34
5.1. Représentation de la photomorphogenèse dans la modélisation....34
5.2. Modélisation de la mesure des signaux....36

6. Problématique....37

III. Matériels et Méthodes....39

1. Modèles de lumière....40
1.1. CARIBU....40
1.2. RIRI....42
1.3. MALIA....43
1.4. Construction du ciel dans les différents modèles....44
1.5. Synthèse des modèles....46
1.6. Calcul du ζ à partir du rayonnement transmis....47

2. Mesure et modélisation du rayonnement directionnel....48
2.1. Dispositifs de mesure directionnelle....48
2.2. Modélisation d'un capteur directionnel....52

3. Acquisition de l'architecture des plantes....57
3.1. Scanner laser....57
3.2. Reconstruction des surfaces....59

4. Expérimentations....60
4.1. ArchiMono....60
4.2. ArchiDuo....61
4.3. Cinétique journalière du ζ....62
4.4. Mesure de peuplement artificiels....63
4.5. Traitement des spectres mesurés....64

5. Évaluation de la sensibilité de la morphogenèse au ζ simulé à partir de deux modèles de rayonnement....65
5.1. Approche....65
5.2. Cadre d'utilisation des modèles de transfert radiatif pour le calcul dynamique du ζ....65
5.3. Modèle de plante individuel individu-centré: Virtual GrassLand....68
5.4. Évaluation de la sensibilité des réponses morphogénétiques à la méthode de simulation du ζ....70
IV. Résultats....71
1. Introduction....72
2. Sensibilité des modèles radiatif à la simulation du ζ....72
2.1. Effet des paramètres d'entrée....72
2.2. Effet de la méthode de modélisation des rediffusions sur le calcul du ζ....75
2.3. Stade de développement et architecture des plantes en interaction avec le ζ....77
2.4. Effet des valeurs de ζ simulées sur la taille finale des organes simulés....78
2.5. Sensibilité de la morphogenèse à la modélisation du ζ....80
2.6. Comparaison de la croissance des plantes simulées avec des mesures....81
3. Variabilité du signal directionnel....85
3.1. Evolution du ζ en fonction de la distance....85
3.2. Evolution du ζ en fonction de l'heure de mesure....88
3.3. Evolution du ζ en fonction du voisinage....90
4. Modélisation de la mesure du rayonnement directionnel....92
4.1. Modélisation simple du rayonnement directionnel....92
4.2. Comparaisons entre les simulations obtenues avec les modèles directionnels et les mesures....96
5. Application de diverses conditions à la modélisation de la mesure du rayonnement directionnel....117
V. Discussion....120
1. Introduction....121
2. Mesure et variabilité du signal ζ pour une plante dans un couvert....121
2.1. Directionnalité du signal....121
2.2. Distance et stade de développement....124
2.3. Évolution du ζ en fonction de l'heure de mesure....128
3. Capacité des modèles à simuler le ζ de manière directionnelle....131
3.1. Création du capteur....131
3.2. Difficulté de représentation de la mesure dans les modèles....137
3.3. Limites des modèles de ciel utilisés....139
4. Sensibilité des modèles de transfert radiatifs à la simulation du ζ....140
5. Intégration du signal....142
VI. Conclusion et Perspectives....145
VII. Bibliographie....147
VIII. Annexes....158

I. Introduction

Les prairies semées et permanentes sont une source de production de fourrage peu onéreux utilisées pour l'alimentation animale. Ces systèmes agraires représentent en France plus de 1/3 des surfaces agricoles et présentent de multiples avantages. En effet les prairies assurent un bon rendement de nourriture de qualité au cours du temps pour un faible impact environnemental, l'apport en intrant dans les prairies étant très faible (Durand *et al.* 2013). Elles participent à la préservation de la biodiversité grâce à la richesse de leur composition, aussi bien inter que intraspécifique. Elles contribuent au stockage de carbone lié à la pérennité des plantes qui composent ces systèmes. Les prairies semées sont pour la plupart composées d'un mélange de légumineuses et de graminées, cette association permet de garantir un rendement important et stable au cours du temps. L'ensemble des avantages des prairies en font un système agricole incontournable dans la chaîne de production alimentaire. La compréhension du fonctionnement des plantes présentes dans le couvert prairial est importante pour améliorer et adapter les prairies aux nouveaux contextes et plus particulièrement aux modifications de l'environnement imposées par le changement climatique.

L'une des voies de recherche pour améliorer la compréhension du fonctionnement des prairies correspond à l'étude expérimentale et de modélisation des interactions plantes-environnement. Dans un couvert végétal, les plantes subissent les variations de multiples paramètres environnementaux telles que la température, l'eau et la lumière. La lumière est un facteur déterminant de la croissance des plantes de par son effet sur les échanges gazeux (photosynthèse), cependant les interactions avec la végétation ne se limitent pas uniquement à la quantité de lumière reçue par les plantes mais comprennent également la qualité de la lumière. En effet, certaines longueurs d'onde indépendamment de l'effet quantitatif du rayonnement sur la photosynthèse ont un impact sur la morphogenèse des plantes au travers de photorécepteurs présents dans les différents organes des plantes. Parmi ces photorécepteurs on trouve les cryptochromes, qui absorbent les longueurs d'onde dans le bleu (400 nm-500 nm), et les phytochromes qui absorbent les longueurs d'onde situées entre 600 nm et 800 nm. Les phytochromes régulent différents processus de croissance des organes, en particulier leur nombre, leur taille et leur vitesse de croissance. L'ensemble de ces processus modulés par les signaux lumineux correspond à la photomorphogenèse.

Cette thèse porte principalement sur l'analyse du signal lumineux R:FR réfléchi par les plantes. Il est caractérisé par le rapport des flux de photons mesurés pour les longueurs d'ondes correspondant aux pics d'absorption des phytochromes situés à 660 nm et 730 nm. Dans ce travail ce rapport est nommé ζ (zêta) (Monteith 1976). Ce signal intervient dans toutes les étapes de

croissance des plantes, depuis la germination jusqu'à la sénescence. La variabilité du ζ dans le rayonnement incident a été caractérisée au cours de la deuxième moitié du XX siècle. Au sein du couvert, cette variabilité a été caractérisée plus tardivement (fin des années 80). La mesure du ζ y a majoritairement été réalisée de manière hémisphérique, permettant une mesure simple et unique du rayonnement dans un couvert. Les mesures hémisphériques du ζ ont permis d'établir des lois de réponses entre des valeurs de signal mesurées et les réponses observées. Ces mesures hémisphériques permettent d'obtenir une valeur de ζ provenant de l'ensemble des directions. Cette méthode de mesure ne permet cependant pas de caractériser la variabilité du signal en lien avec la présence des plantes voisines dans des situations de couvert peu développé. En effet les signaux lumineux sont modifiés par les plantes en fonction de leur architecture et des propriétés optiques de leurs organes. Dans les premières phases de développement du couvert, pour mettre en évidence une modification de ζ en lien avec la présence de plantes voisines, des mesures directionnelles des signaux lumineux sont nécessaires.

Un moyen d'accéder au ζ perçu par les plantes dans un couvert est la modélisation. En effet, les modèles sont capables de simuler le rayonnement reçu pour un nombre de mesure théoriquement infini de l'environnement lumineux d'une plante au sein du couvert quelles que soient les conditions (position du soleil, météo) et le nombre de point de mesure. Les modèles de plante structure-fonction permettent une simulation du rayonnement reçu par les organes, toutefois ces outils ne sont actuellement pas configurés pour la simulation directionnelle des signaux lumineux.

Dans cette thèse, nous avons cherché à modéliser l'environnement lumineux directionnel provenant d'une plante ou d'un groupe de plante. Pour cela des modèles de transfert radiatifs ont été choisis et adaptés pour la simulation du ζ caractérisant le rayonnement directionnel. Un dispositif de mesures directionnelles a été développé, ce qui a permis de réaliser des expérimentations dont les résultats ont servi à caractériser la variabilité du ζ directionnel et à évaluer la qualité de simulation des modèles de transfert radiatif.

Le premier chapitre de cette thèse présente l'**État de l'art** en partant du contexte des prairies, puis en présentant les travaux antérieurs sur la mesure du rayonnement lumineux dans le cadre de l'étude de la photomorphogenèse, suivie d'une présentation des différentes méthodes de modélisation de la lumière dans les modèles de plante structure-fonction. Cette partie se conclue par la présentation de la problématique traitée dans cette thèse.

Le second chapitre est consacrée au **Matériels et Méthodes**. Cette partie présente les dispositifs de mesure de l'environnement lumineux directionnel, l'adaptation des modèles de transfert radiatif à la simulation de l'environnement lumineux directionnel et le protocole expérimental utilisé pour les différentes expérimentations.

Le chapitre **Résultats** porte sur la présentation de résultats de simulations et des mesures du ζ de flux directionnel en fonction de différentes conditions.

Les principaux résultats sont ensuite mis en avant dans le chapitre **Discussion** qui met en perspective nos observations avec celles de la littérature concernant les variations du ζ (hémisphérique et directionnel). Les résultats de simulation obtenues avec les modèles de transfert radiatif sont discutés afin d'identifier les sources d'erreurs possibles entre la mesure et la modélisation.

Enfin, le chapitre **Conclusion et Perspectives** synthétise l'apport de ce travail de thèse dans le contexte de la recherche sur la photomorphogenèse et propose différentes suites possibles à ce travail.

II. État de l’art

1. Les prairies

1.1. Culture des prairies en France

Les prairies constituent un élément important dans l'agriculture, en Europe elles représentent 35 % de la surface agricole utile (SAU) (Huyghe et al. 2014) et entre 20 % et 25 % de la SAU en France. Elles constituent un maillon central dans le système de polyculture-élevage de par leur production d'aliment peu onéreux, avec un faible apport d'intrants et assurant d'importantes fonctions écosystémiques (Huyghe et al. 2015). Les prairies participent à la séquestration du carbone atmosphérique par la production de biomasse. En Europe la capture du CO2 réalisée par les prairies est plus importante que celle réalisée par les forêts (Mauchamp et al. 2020). Elles participent également à améliorer la pérennité de la parcelle et la stabilité des sols en réduisant les risques d'érosion liés au système racinaire des espèces végétales. En plus de réduire les risques d'érosion, la présence d'un couvert végétal tel que celui des prairies permet de retarder les lessivage en retenant les éléments du sol pouvant contaminer les nappes phréatiques (en particulier les nitrates), les prairies permettent donc une régulation de la qualité des eaux. Cependant, les services écosystémiques apportés par les prairies peuvent être fortement réduits par des pratiques agricoles intensives telles que des fauches trop fréquentes ou une sur-fertilisation des sols. En effet, cela a pour conséquence de transformer les prairies en sources de CO2 et le travail du sol les rend plus sensibles à l'érosion et aux écoulements (Mauchamp et al. 2020). Pour ces raisons, l'évolution des pratiques agricoles vers une culture des prairies sans apport d'intrants permet d'améliorer leur bilan atmosphérique.

Deux catégories de prairies peuvent être distinguées en fonction de leur mode de culture : les prairies permanentes et les prairies semées. Une prairie permanente est un espace enherbé subissant peu de pratiques agricoles, n'ayant pas été ressemé depuis au moins 5 ans et qui ne fait pas partie d'une rotation culturale. Une prairie temporaire est une culture ayant entre 1 et 5 ans et faisant partie d'une rotation culturale, ce type de prairie peut être régulièrement fauché et/ou être utilisé comme pâturage.

Les prairies semées sont génétiquement hétérogènes et accueillent une très grande diversité inter et intraspécifique, principalement composées de graminées (poaceae) et de légumineuses qui forment un couvert végétal dense (plus de 1000 plantes.m-2) (Gillet 1980). Entre 1960 et 2010, la surface des prairies permanentes a diminué de 25 % alors que la surface occupée par les prairies

temporaires a augmenté de 83 % (Huyghe et al. 2015). L'augmentation de surface des prairies temporaires peut s'expliquer par un recul des cultures pures au profit de surfaces en mélanges et associations. En effet, l'association graminées-légumineuse permet de réduire l'utilisation des intrants (engrais et produits phytosanitaires) grâce à la fixation d'azote des légumineuses, d'améliorer les performances agronomiques grâce au rendement énergétique des graminées et au rendement protéique des légumineuses et permet d'assurer une production fourragère stable tout au long de l'année grâce à la forte pérennité des graminées (Louarn et al. 2010; Gastal et al. 2012; Prieto et al. 2015).

1.2. Impact des pratiques agricoles sur la diversité des prairies temporaires

La première pratique agricole ayant un impact sur la diversité des prairies temporaires est la diversité initiale lors du choix des graines utilisées pour le semis. La diversité semée initiale a un impact sur les interactions plante-plante car celles-ci diffèrent selon les espèces présentes dans la prairie, où le choix des légumineuses associées aux graminées a un impact particulièrement important sur la dynamique d'installation des espèces prairiales au cours du temps (Louarn et al. 2010). Par exemple, dans le cas d'une association trèfle-blanc/graminées en condition de faibles intrants azotés, la compétition pour cette ressource conduit à une alternance cyclique entre les légumineuses qui enrichissent le sol en N et les graminées, meilleures compétitrices pour la lumière (Harper 1977; Turkington 1989; Thornley et al. 1995; Loiseau et al. 2001). Dans le cas d'une association luzerne/graminées, la compétition pour la lumière est plus forte et la luzerne est généralement dominante sur les graminées les premières années (Chamblee and Collins 1988) jusqu'à ce que les graminées colonisent l'espace et provoquent une réduction de la densité de luzerne dans la prairie.

La fauche régulière des prairies semées modifie drastiquement la structure physique ainsi que le microclimat des plantes (Fig. 1) ce qui a pour conséquence de modifier l'ensemble des interactions entre les plantes. Ce microclimat est caractérisé par la lumière, la température, la disponibilité en eau et la concentration en CO2 dans le couvert. Après la fauche, l'ensemble des organes du couvert se retrouve à la même hauteur, modifiant la dominance des espèces du couvert. La capacité de repousse des plantes, en particulier la remobilisation de leurs ressources et la préservation des méristèmes, est déterminante pour la compétition entre les individus et donc pour

la diversité du couvert. La morphologie des plantes présentes dans le couvert joue donc un rôle important dans l'acquisition des ressources et par la suite dans la diversité résultante.

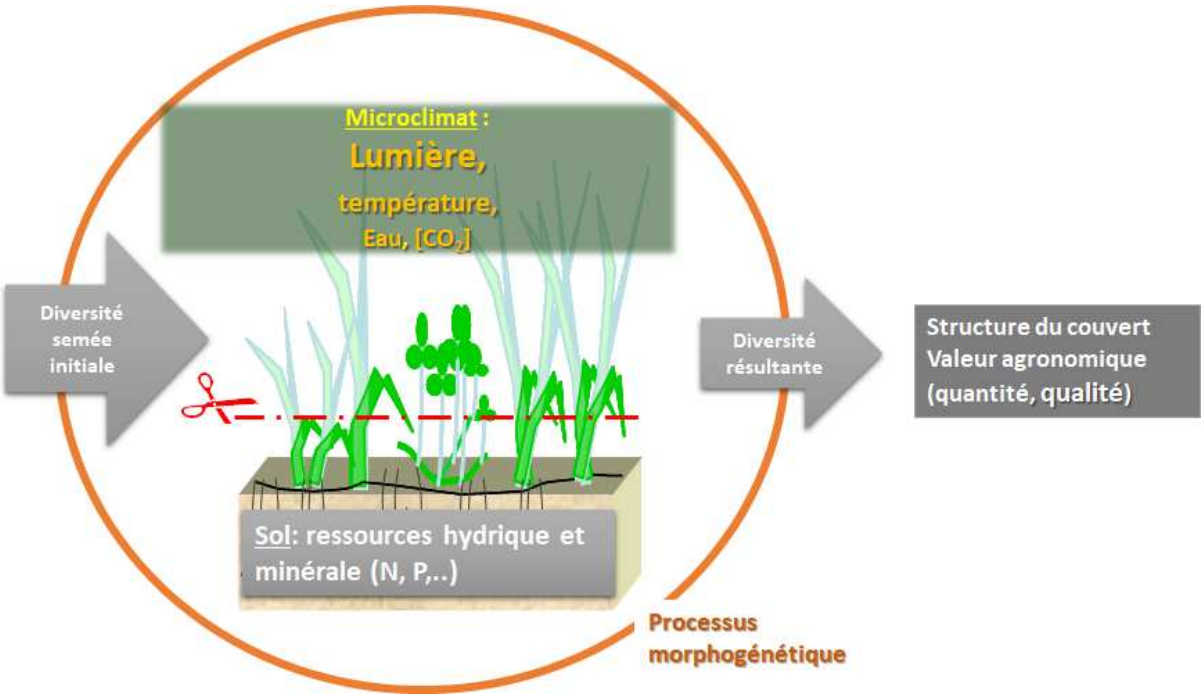


Figure 1: Evolution de la diversité d'une prairie et de son microclimat durant son cycle de vie.

1.3. Morphologie des espèces fourragères

1.3.1. Les graminées

Les graminées sont des monocotylédones dont les phytomères (unité d'organisation d'une plante) sont définis par des entre-nœuds auxquels sont attachées des feuilles et des bourgeons associés, ces entre-nœuds sont extrêmement réduits en phase végétative (Gillet 1980). Chez les graminées fourragères, les bourgeons situés à l'aisselle des feuilles sont regroupés dans une zone appelée le plateau de tallage situé à quelques millimètres au-dessus du sol (Fig. 2).

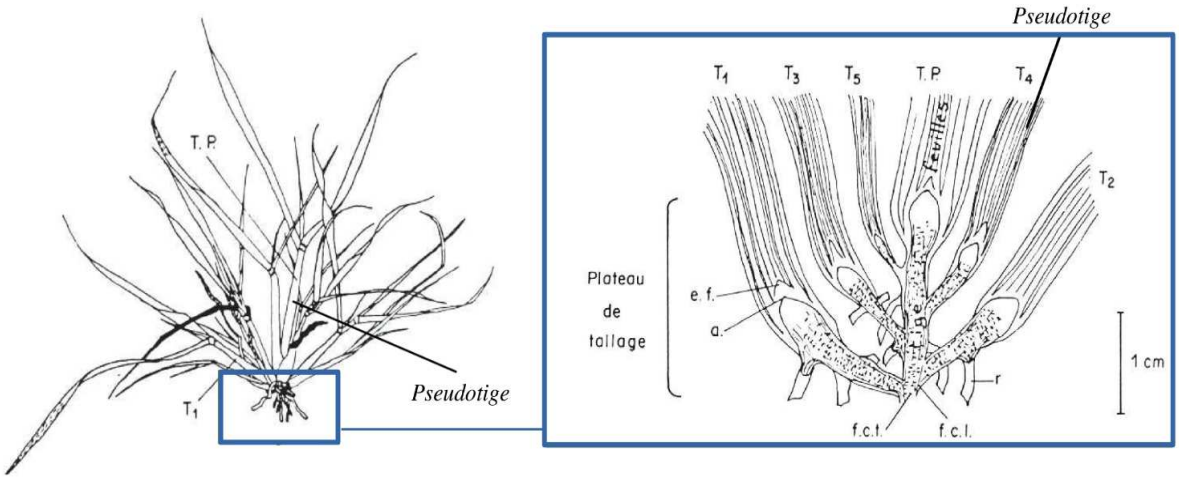


Figure 2: Dessin d'une touffe de ray-grass et de son plateau de tallage, repris de . Six talles développées sont visibles. e.f : ébauche foliaire, a, apex, r racines. T.P Talle primaire.

Le développement des phytomères aboutit à la formation des talles constituées d'une série de feuilles qui s'enroulent les unes dans les autres autour de l'apex (Fig. 3). La croissance des feuilles (composées d'un limbe et d'une gaine reliés par une ligule) peut être résumée au processus d'allongement où la production de nouvelles feuilles et de nouvelles talles vont dépendre de la durée de cet allongement avec des règles de coordination précises (Verdenal et al. 2008). Ainsi, le bourgeon d'une nouvelle feuille apparaît lorsque la pointe de la feuille précédente sort de la gaine et un nouveau bourgeon de talle apparaît lorsque la 3e feuille de cet axe est émise (Skinner and Nelson 1995; Lafarge and Durand 2011).

La croissance des graminées est identique dans son fonctionnement quelle que soit l'espèce, cependant une forte diversité de forme peut être observée entre et au sein des espèces (Fig. 4). Les graminées peuvent ainsi produire un grand nombre de talles avec des petites feuilles ou produire de longues feuilles avec moins de ramifications. L'émission des talles étant liée à l'émission des feuilles, le nombre de ramification produite est donc corrélé au temps d'élongation foliaire.

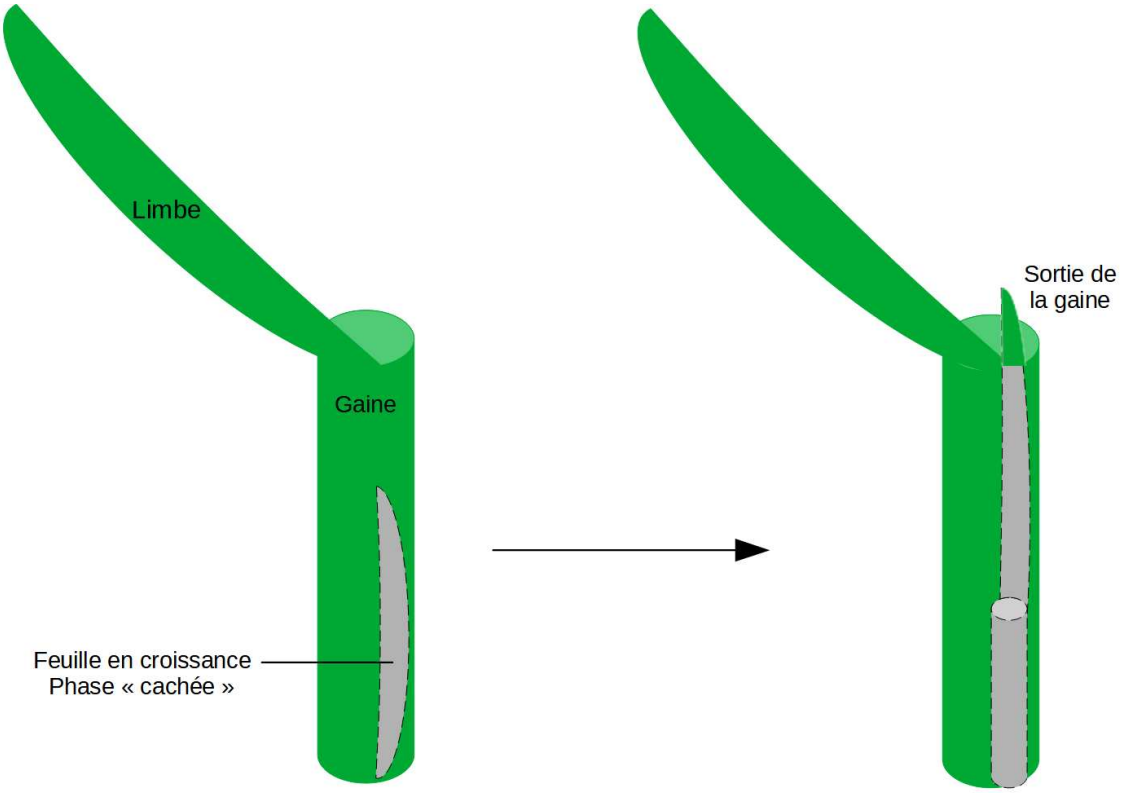


Figure 3: Schéma du développement des feuilles de graminées. La phase « cachée » de la croissance d'une feuille correspond au stade de croissance où elle pousse dans la gaine mais n'est pas encore observable.



Figure 4: Illustration de la variabilité génétique interspécifique de l'appareil aérien de graminées prairiales, repris de . De gauche à droite : ray-grass anglais gazonnant, ray-grass anglais fourrager, dactyle (Dactylis glomerata L.) et fétuque (Festuca arundinacea S.). Des caractères comme le nombre de talles, la longueur de feuilles et leur géométrie sont très variables et conduisent à une grande diversité architecturale inter et intraspécifique des graminées.

1.3.2. Les légumineuses

Les légumineuses sont des dicotylédones ayant une organisation architecturale différente selon les espèces. Deux principaux profils de croissance peuvent être décrits : un profil « rampant » formant des stolons (par exemple le trèfle blanc) et un profil « dressé » formant de longues ramifications en hauteur (Fig. 5A). Contrairement aux graminées, les légumineuses présentent des phytomères accessibles à l'observation car ceux-ci sont formés successivement le long des axes de la plante (Fig. 5B). Un phytomère de légumineuse est composé d'un entre-nœud, d'un pétiole, d'une feuille composée de folioles et d'un bourgeon se formant à l'aisselle du pétiole et de l'entre-nœud (Fig. 5C). Contrairement aux graminées, l'apparition des fleurs ne conduit pas à la disparition des axes.

La morphogenèse des légumineuses a un impact sur leur croissance dans le couvert, car les légumineuses ayant une structure formant des stolons sont moins dominantes et plus assujetties à la compétition pour la lumière, tandis que les légumineuses se développant verticalement ont tendance à être plus dominantes et donc subissant moins la compétition pour la lumière.

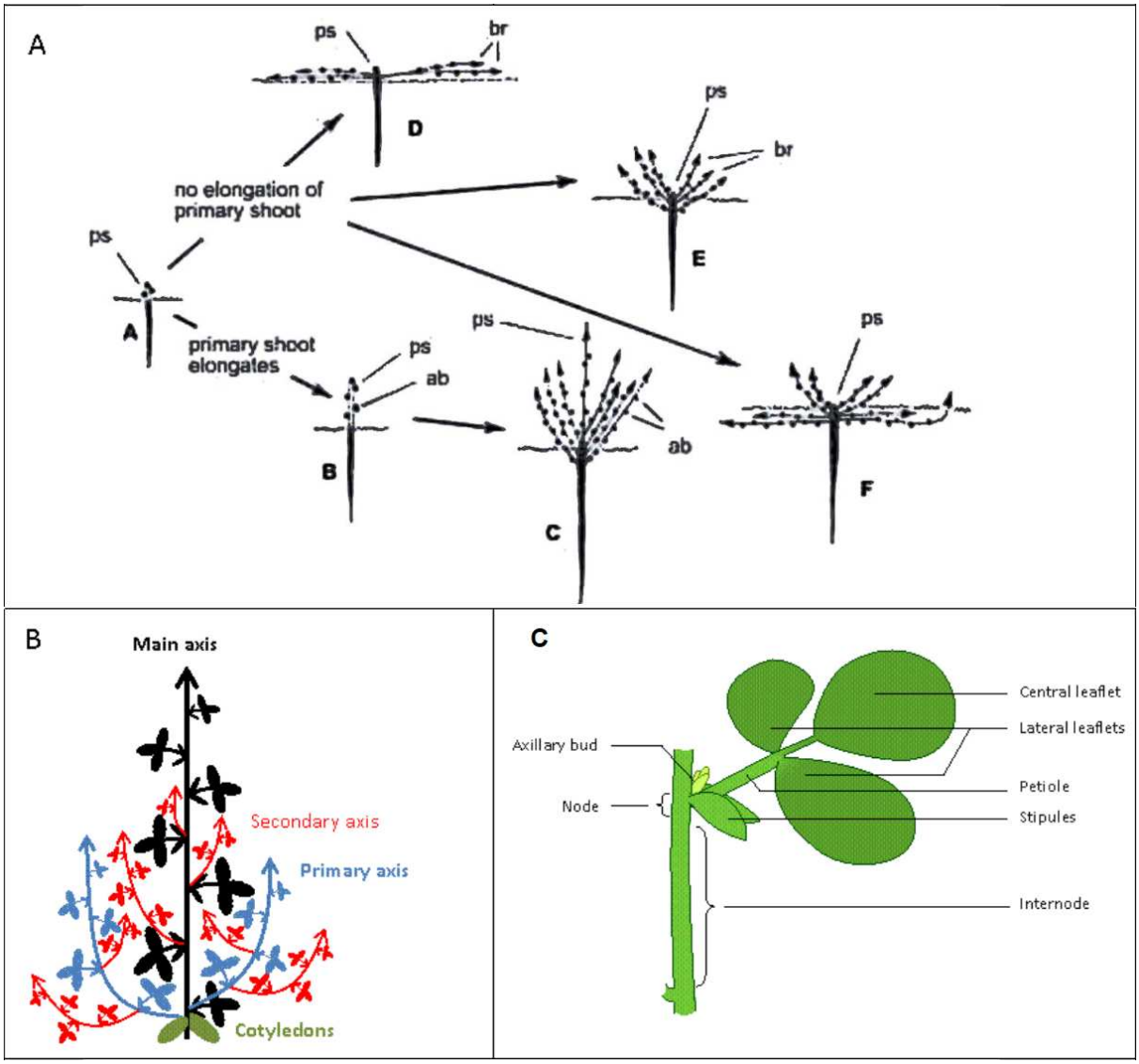


Figure 5: Développement des légumineuses. (A) Représentation schématique du développement de la morphologie de différentes Légumineuses fourragères . ps= primary shoot (axe principal), ab= axillary buds (bourgeons axillaires), br=branches. A = plantule; B, C = luzerne ; D = trèfle blanc et trèfle souterrain ; E = lotier et trèfle violet ; F = trèfle Kura. (B) Classification générale des axes utilisés pour décrire les légumineuses (C) Organisation des organes d'un phytomère de légumineuse .

2. Les interactions plante-environnement

2.1. Interactions plante-plante

Dans un couvert végétal, les interactions entre les plantes et leur environnement jouent un rôle déterminant sur leur croissance et leur survie. Différents types d'interaction ont lieu au sein d'un couvert végétal et peuvent être décrits par l'approche 4C : la complémentarité, la compensation, la coopération et la compétition (Justes et al. 2021).

La complémentarité correspond à des besoins en ressources différents entre des espèces en croissance dans un même milieu. La complémentarité peut également être pour une même ressource si les espèces n'ont pas besoin de cette ressource au même moment de leur croissance.

La compensation correspond à l'occupation de l'espace par une espèce lorsqu'une autre espèce ne parvient pas à se maintenir dans le milieu, généralement à cause d'un stress biotique ou abiotique.

La coopération correspond aux interactions où l'une des espèces présente modifie l'environnement de manière bénéfique pour une autre espèce. Par exemple, dans le cas des prairies semées une coopération existe entre les légumineuses et les graminées grâce à l'enrichissement du sol en ressources azotées qui est utilisé par les graminées.

La compétition désigne les interactions où une espèce présente possède une meilleure capacité à acquérir des ressources limitées au détriment des autres espèces. Les principales interactions de compétitions sont pour les ressources suivantes : la lumière, les ressources hydriques et les éléments nutritifs du sol.

Ces quatre types d'interaction entre les plantes ne sont pas exclusifs et plusieurs types d'interactions peuvent avoir lieu au sein d'un même couvert végétal.

La lumière, en l'absence d'autres contraintes liées à l'acquisition des autres ressources (eau/azote/éléments minéraux), constitue la force motrice de la croissance des plantes. L'étude des interactions plante-lumière est donc primordiale pour comprendre son impact sur la morphogenèse des plantes et par conséquent sur l'élaboration du rendement des cultures en générales et des prairies semées en particulier. Pour cette raison cette thèse s'intéresse principalement aux interactions entre les plantes et leur environnement lumineux.

2.2. Interactions lumineuses

La capacité des plantes à intercepter la lumière dépend de leur architecture aérienne :de la surface des organes , de la distribution spatiale de ces organes, de leur orientation ainsi que de la direction du soleil. Une fois le rayonnement intercepté, son devenir va dépendre des propriétés optiques de l'organe à absorber transmettre ou réfléchir ce rayonnement.

Ainsi dans un couvert, une plante reçoit de la lumière provenant de trois sources (Fig. 6) :

- **Le rayonnement direct**, c'est-à-dire la lumière provenant directement du Soleil.
- **Le rayonnement diffus**, issu de la diffraction de la lumière par les molécules présentent dans l'atmosphère et ne provenant pas d'une source précise, contrairement au rayonnement direct.
- **La lumière rediffusée** par l'environnement proche (plantes voisines, sol, bâtiments…), qui comprend à la fois le rayonnement réfléchi et le rayonnement transmis par cet environnement

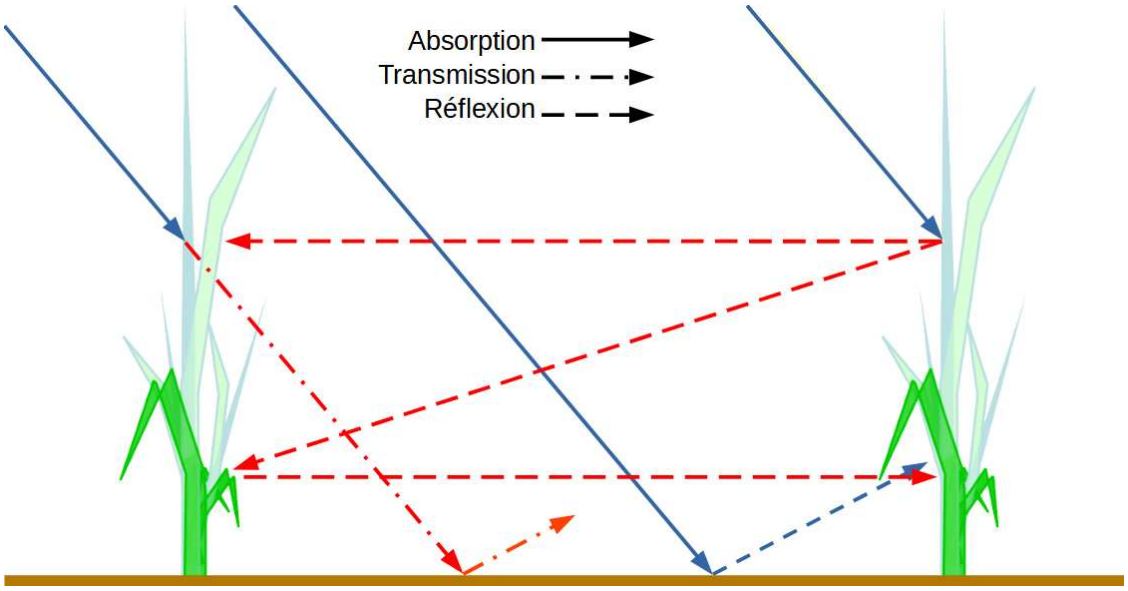


Figure 6: Schéma d'interaction des plantes avec l'environnement lumineux au sein d'un couvert.

Lorsque la lumière arrive sur la surface d'un organe de plante capable de réaliser la photosynthèse, cet organe absorbe les longueurs d'ondes situées dans le PAR (Photosynthetically Active Radiation) entre 400 nm et 700 nm et transmet et/ou réfléchi les autres longueurs d'ondes provenant du rayonnement incident. Les longueurs d'ondes absorbées, réfléchies et transmises dépendent des propriétés optiques (PO) des différents organes de la plante (épaisseur, structure,

pilosité…) (Mazliak 1982) et de l'angle d'incidence de la lumière sur ces organes (Seavers and Smith 1997).

Les interactions plantes-lumière interviennent dès l'apparition des premiers organes végétaux. Les photorécepteurs présents dans ces organes permettent d'analyser l'environnement lumineux résultant de ces interactions (quantité, qualité et l'asymétrie spatiale de la lumière), mais également la périodicité du signal liée à la variabilité saisonnière du rayonnement incident (Mazliak 1982).

La quantité de rayonnement est exprimée en densité de flux de photons (PPFD) et correspond à la quantité d'énergie reçue par les plantes. La notion de quantité de lumière interceptée par les plantes est généralement définie comme la quantité d'énergie dans le domaine du PAR utilisée par les plantes pour la photosynthèse qui correspond au flux de photons des longueurs d'onde situées entre 400 nm et 700 nm. Aujourd'hui, la définition du domaine du PAR est cependant discutée et certains scientifiques considèrent un élargissement nécessaire au proche infrarouge (Zhen and Bugbee 2020).

La qualité ou composition spectrale de la lumière décrit les longueurs d'ondes que les plantes sont capables de percevoir, et qui dépassent le domaine du visible. En effet, certains photorécepteurs des plantes opèrent dans le domaine des ultraviolets (à partir de 280 nm pour les récepteurs d'UVB) et jusqu'à 800 nm (proche infrarouge pour les phytochromes). La qualité de la lumière est modifiée à l'intérieur du couvert en fonction des propriétés optiques des différents organes végétaux, les plantes réfléchissent majoritairement des longueurs d'onde correspondant au domaine du proche infrarouge aussi appelé rouge sombre (Fig. 7).

La périodicité a un impact sur les plantes sensibles à la durée du jour, par exemple le déclenchement de la floraison chez les plantes de jour court ou de jour long. En milieu naturel cette périodicité dépend de la latitude et de la période de l'année mais peut facilement être reproduite en condition contrôlée.

Enfin, **l'asymétrie spatiale** décrit la direction selon laquelle la lumière incidente arrive sur les organes des plantes et peut modifier la direction de croissance des organes ou peut entraîner leur réorientation. La position des organes des plantes dans l'espace ainsi que la surface qu'ils occupent jouent donc un rôle sur les interactions plante-lumière, l'architecture des plantes est donc un élément important à prendre en compte pour la compréhension de ces interactions.

Le voisinage végétal joue un rôle déterminant dans les variations quantitative (ombrage) et qualitative perçu par un individu au sein d'un couvert. Ces quatre composantes (qualité, quantité, périodicité et asymétrie) de la lumière utilisée comme source d'information par les plantes sont distinctes les unes des autres, en particulier la qualité et la quantité de lumière qui ont des effets distincts sur la croissance des plantes (Sparkes and King 2008). L'ensemble des modifications de croissance liées aux interactions plante-lumière basées sur les variations spectrales de cette dernière est défini par la photomorphogenèse.

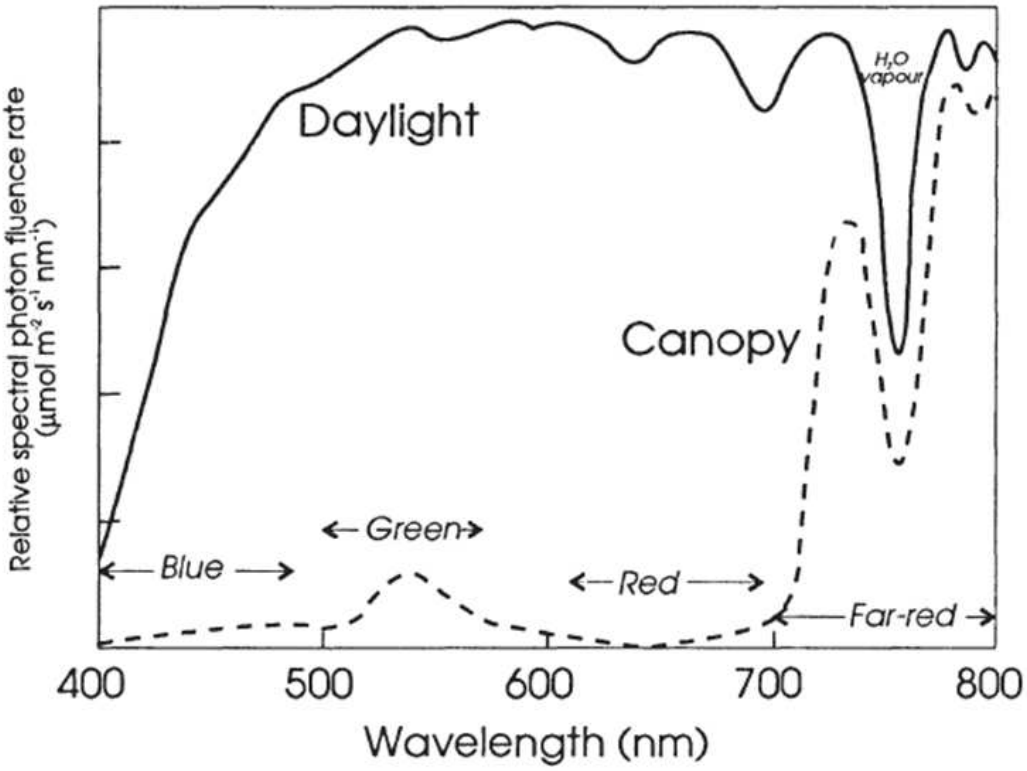


Figure 7: Spectre lumineux mesuré au dessus (—) et à l'intérieur d'un couvert végétal (– –), repris de .

2.3. Interactions plante-environnement dans les prairies semées

Dans le cas des prairies temporaires, le semis est effectué à la même date pour l'ensemble des plantes du couvert, ainsi la croissance de l'ensemble des individus composant la parcelle peut être simultanée ou alors décalée. Dans les deux situations, durant les premiers stades de croissance, on suppose qu'aucune compétition pour les ressources n'est encore mise en place entre les individus : les individus ne reçoivent pas d'ombre de la part des plantes avoisinantes et les ressources hydriques et minérales du sol sont suffisantes pour les besoins de ces plantes. Cependant, les plantes peuvent détecter la présence de plantes voisines au travers des signaux lumineux échangés par les premiers organes en émergence, qui agissent ainsi comme signal précoce de détection des plantes voisines (Ballaré et al. 1987).

3. La photomorphogenèse

La photomorphogenèse correspond aux réponses développementales liées à l'interaction des plantes avec la lumière, certains auteurs ont restreint ce processus aux réponses développementales liées à un stimulus lumineux non directionnel et non périodique et indépendantes de la photosynthèse (Mohr 1972).

3.1. Perception de l'environnement lumineux par les plantes via les photorécepteurs

La lumière agit comme un signal provoquant des réponses morphogénétiques. La lumière doit être interceptée par un site de perception qui, selon les longueurs d'ondes, va déclencher différentes cascades de transduction du signal puis de réponses. Ces sites de perception correspondent à la présence de différents photorécepteurs qui vont répondre à des longueurs d'ondes qui leur sont propres.

3.1.1. Sites de perception

La localisation des sites de perception est distincte selon les plantes (Fig. 8).

Chez les monocotylédones, les principaux sites de perceptions du R (Rouge clair) et du FR (Rouge sombre) ont été localisés au niveau de l'apex (Wan and Sosebee 1998), de la gaine (Cordukes and Fisher 1974; Skinner and Simmons 1993; Li et al. 2001), et des feuilles en croissance (Skinner and Simmons 1993) (Fig. 8).

Chez les dicotylédones, les principaux sites de perceptions du R et du FR ont été localisés sur les feuilles en cours de développement (Heraut-Bron et al. 2000; Hay et al. 2001; Héraut-Bron et al. 2001), sur les pétioles (Thompson 1995), sur les entre-nœuds (Lecharny 1979; Ballaré et al. 1990), sur les apex (Robin et al. 1994), sur les cotylédons ainsi que sur l'hypocotyle (Franssen and Bruinsma 1981; Procko et al. 2014).

La localisation et le nombre de sites de perceptions sont donc liés à l'architecture des plantes. La localisation des sites de perception évolue avec la taille de la plante chez les dicotylédones, car ils suivent le développement foliaire via les feuilles en développement et les entre-nœuds, tandis que chez les monocotylédones leurs sites de perception restent localisés à la base de la plante, au niveau de l'apex et des gaines.

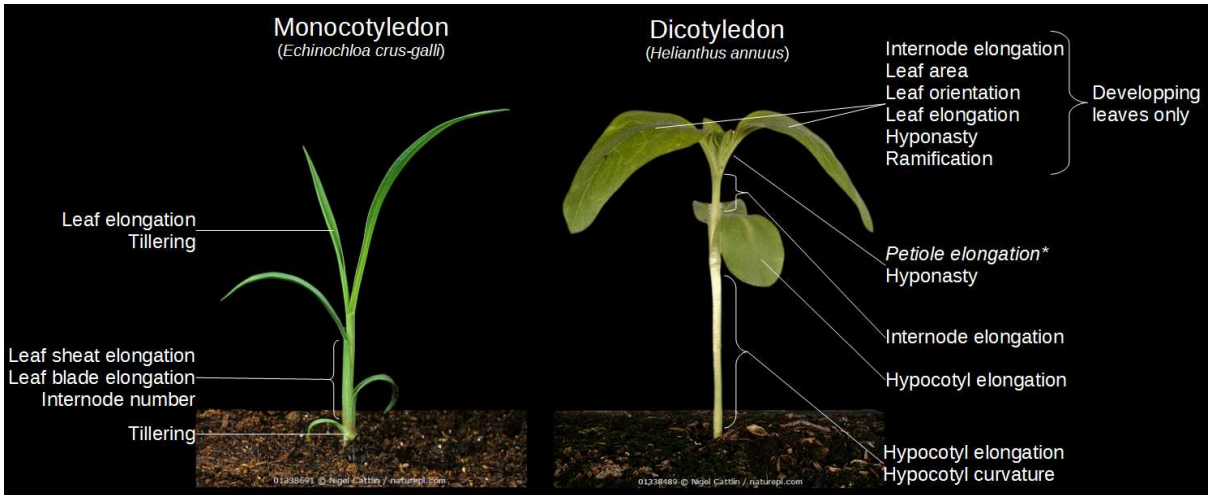


Figure 8: Localisation des potentiels sites de perceptions du FR mis en évidence dans la littérature et leurs effets respectifs sur la morphogenèse. Photos prises par Nigel Cattlin.

Trois catégories de photorécepteurs ont été identifiées (Fraser et al. 2016). Les phytochromes, qui absorbent des longueurs d'onde allant du bleu au proche infrarouge et dont les pics d'absorption se situent dans le domaine du rouge clair (R, 660 nm) et du rouge sombre (FR, 730 nm). Les cryptochromes, qui absorbent principalement des UVA et le bleu dont les pics d'absorption se situent respectivement autour de 380 nm et 450 nm. Enfin, les photorécepteurs UVR8, qui absorbent principalement dans les longueurs d'onde des UVB (280 nm-315 nm).

3.1.2. Les phytochromes

Les phytochromes sont des photorécepteurs qui jouent un rôle important par rapport aux variations de la qualité de lumière en lien avec le voisinage. Les phytochromes existent sous 2 formes : une forme active (PFR) dont le maximum d'absorption est à 730 nm et une forme inactive (PR) dont le maximum d'absorption se situe à 660 nm (Mazliak 1982). Ces maximums d'absorption pour chaque forme correspondent aux longueurs d'onde provoquant le passage d'une forme à une autre. Les phytochromes sont synthétisés sous la forme inactive (Fig. 9), une augmentation de R (600-700 nm) entraîne une conversion de la forme inactive vers une forme intermédiaire à faible durée de vie (P695) puis passe sous forme active PFR, à l'inverse une exposition de la forme active du phytochrome à des longueurs d'onde proche de 730 nm entraîne une conversion du PFR vers une forme intermédiaire avant de passer sous forme PR à l'obscurité (Fig. 9). Ce mécanisme de transition entre les deux formes des phytochromes est appelé photoconversion. Le rapport entre la teneur en PFR et la teneur totale en phytochrome (PFR+PR) constitue le photoéquilibre des phytochromes, ce rapport permet de caractériser l'action des phytochromes sur la morphogenèse (Morgan and Smith 1976).

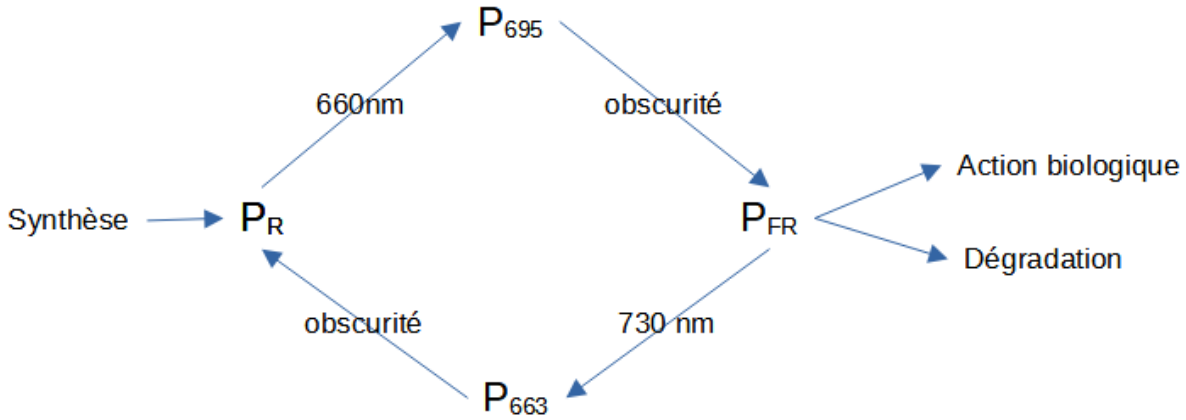


Figure 9: Formes des phytochromes en fonction des longueurs d'onde absorbées. P663 et P695 sont des formes intermédiaires à faible durée de vie, repris de .

3.1.3. Les cryptochromes

Les cryptochromes sont des photorécepteurs ayant un pic d'absorption autour de 450 nm. L'exposition à cette longueur d'onde déclenche l'activation des cryptochromes qui entraîne une cascade de transcription responsable des réponses de photomorphogenèse liées à la lumière bleue. En parallèle, l'activation des cryptochromes provoque une boucle d'inactivation produisant des inhibiteurs des cryptochromes permettant de réguler leur action (Wang and Lin 2020). De nombreuses réponses à la lumière impliquant les cryptochromes ont été identifiées, parmi les réponses de morphogenèse on trouve l'inhibition de l'allongement de l'hypocotyle (Ahmad and Cashmore 1993), la stimulation de la croissance des cotylédons (Lin et al. 1998) et la régulation des réponses à l'ombrage (Pedmale et al. 2016).

3.1.4. Les photorécepteurs UVR8

Les photorécepteurs UVR8 sont des récepteurs encore peu étudiés (Jenkins 2014). Les UVR8 sont principalement responsables de réponses liées à la protection des plantes aux UV, en déclenchant des mécanismes de réparation de l'ADN. Ces photorécepteurs sont également impliqués dans la morphogenèse, en régulant la croissance de l'hypocotyle, la croissance des feuilles, l'allongement des tiges et la ramification (Hectors et al. 2007; Wargent et al. 2009).

3.2. Photomorphogenèse des plantes soumises à des variations de FR

L'effet du voisinage se traduit par des modifications locales de flux de photons R et de FR. avec une augmentation relative de FR ce qui conduit automatiquement à la diminution du rapport

R:FR (ζ). Les réponses liées à une diminution de ζ peuvent être morphologiques (modifications à l'échelle des organes), anatomiques (modifications à l'échelle des cellules des organes), nutritionnelles (modifications des processus d'assimilation de stockage des ressources hydriques et minérales) et physiologiques (modification des processus physiologiques) (Waddington et al. 1992).

Ces réponses peuvent être observées à tous les stades de développement et chez toutes les espèces de plantes, aussi bien chez les monocotylédones que chez les dicotylédones (Demotes-Mainard et al. 2016). Une augmentation de la proportion de FR dans le couvert provoque une diminution de ramifications produites par les plantes (Fig. 10), en réduisant le nombre de bourgeons produits ou en retardant leur débourrage (Casal et al. 1987; Wan and Sosebee 1998; Finlayson et al. 2010).

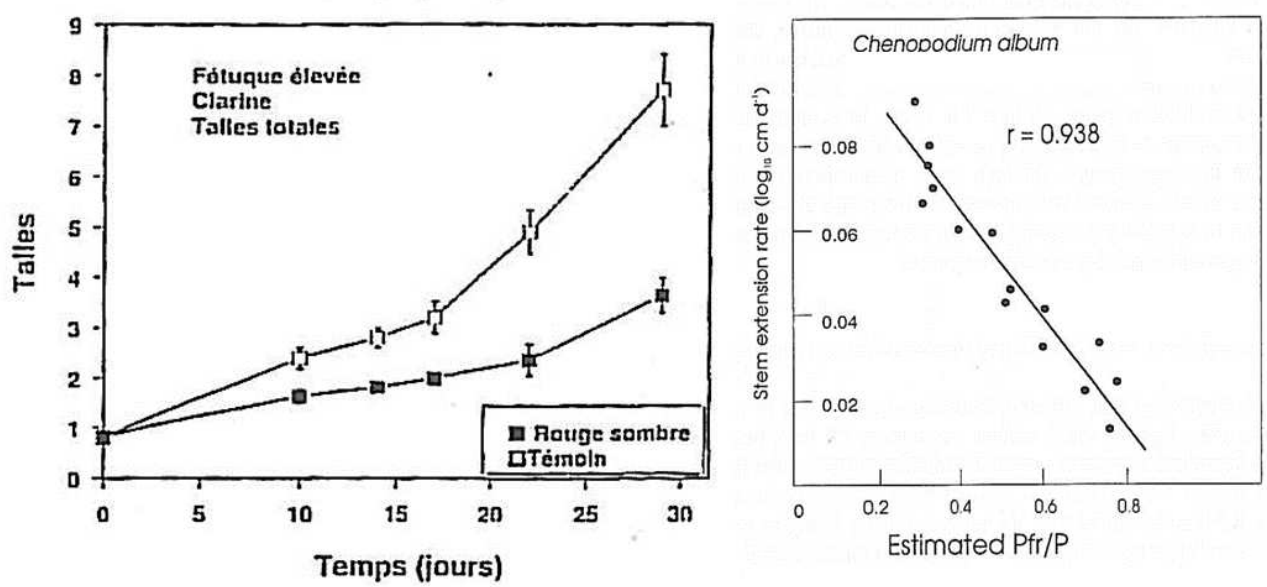


Figure 10: Gauche : Impact de la lumière rouge sombre sur la ramification chez la fétuque, repris de . Droite : Relation entre la vitesse d'allongement de la tige chez l'Ansérine blanche et le photoéquilibre des phytochromes, repris de .

À de jeune stade de développement des plantes, une réponse de croissance de l'hypocotyle peut être observée si celui-ci est éclairé par du FR (Casal 2012). L'augmentation de la taille des tiges est également une réponse photomorphogénétique typique des plantes, cet allongement est à la fois observable par la longueur finale des organes (Kasperbauer 1971; Morgan and Smith 1978; Reed et al. 1993; Cerny et al. 2003) mais également par l'augmentation de la vitesse de croissance (Morgan and Smith 1976). De la même manière, une exposition des plantes à du FR provoque une l'élongation des entre-nœuds (Whitelam and Johnson 1982; Ballaré et al. 1987; Rajapakse and Kelly 1994; Libenson et al. 2002; Heo et al. 2002), des pétioles (Holmes and Smith 1977; Whitelam

and Johnson 1982; Kozuka et al. 2005; Sasidharan et al. 2010) et des feuilles (Casal et al. 1987; Skinner and Simmons 1993).

En plus d'avoir un impact sur la taille des organes, les réponses de photomorphogenèse sont également impliquées dans la réorientation des organes, en particulier dans la réorientation des feuilles en modifiant leur orientation par rapport au sol (Tao et al. 2008) et en modifiant leur orientation horizontalement (Maddonni et al. 2002).

Les réponses engendrées par l'action des phytochromes sous l'effet du voisinage peuvent correspondre à l'une des trois stratégies de cohabitation décrites chez les plantes (Novoplansky 2009; Gruntman et al. 2017).

La confrontation correspond aux réponses ayant pour but de conférer une dominance compétitive, généralement associée à une croissance verticale.

La tolérance correspond aux mécanismes visant à maximiser les performances des plantes sous des conditions de lumière dégradées par les plantes voisines.

L'évitement correspond aux modifications d'architecture latérales dont le but est de minimiser les interactions de compétition.

La stratégie de réponse d'un individu peut varier dans le temps, selon l'évolution des conditions lumineuses. Bien que ces différentes stratégies soient identifiées et distinctes dans leur réponse, cette distinction n'est pas utilisée dans la littérature et l'ensemble des réponses morphogénétiques d'une plante en réponse aux signaux lumineux liés aux individus voisins sont regroupés sous le terme « Syndrome d'évitement de l'ombre » (SAS). Le SAS correspond donc à l'ensemble des changements architecturaux d'une plante en lien avec les signaux lumineux transmis par les plantes voisines (Casal 2012). Il regroupe à la fois les réponses liées à un ombrage direct et donc à une diminution du PAR et les modifications de la qualité de lumière au sein du couvert. L'utilisation du concept de SAS a été principalement faite pour comprendre les interactions entre les plantes dans un couvert développé ou pour étudier les réponses d'une plante en développement dans un couvert déjà installé (en forêt par exemple). Ainsi, les études décrivant le SAS permettent uniquement de comprendre des mécanismes de réponse dans un contexte de couvert développé et donc de forte compétition, cependant l'étude de la photomorphogenèse dans un couvert en cours de développement dans lequel les plantes au stade précoce ne se font pas d'ombre directe reste encore peu étudiée.

3.3. Anticipation

Les signaux que les plantes reçoivent au travers de la qualité et de la quantité de lumière sont distincts (Sparkes and King 2008). En effet, la qualité de la lumière peut être modifiée avant que les plantes ne se fassent une ombre mutuelle. Ainsi, les variations de R et de FR dans un couvert peuvent agir comme un signal précoce de la compétition capable de déclencher des réponses de morphogenèse (Ballaré et al. 1987, 1990). Par exemple, des réponses photomorphogénétiques ont pu être observées chez des plants de radis sauvage (Raphanus raphanistrum) exposés à des plantes voisines situées jusqu'à 1 m de distance (Hewitson et al. 2000). Ce phénomène d'anticipation met en lumière les besoins de caractériser les interactions entre les plantes avant que celles-ci ne provoquent de diminution du PAR dans le couvert, cette caractérisation peut être effectuée par une mesure de la photomorphogenèse couplée à l'utilisation de modèles capable de rendre compte des interactions plante-plante et représentant l'architecture 3D des plantes.

3.4. Photomorphogenèse dans les prairies

L'étude de la photomorphogenèse dans les prairies nécessite de comprendre les interactions entre l'architecture des plantes et la lumière. Les sites d'interactions des légumineuses sont facilement accessibles, les variations de taille des organes en lien avec le voisinage peuvent donc être facilement reliées aux variations de lumière perçues dans le couvert. Les graminées présentent un arrangement spatial des organes qui leur est propre, et l'étude de cette architecture est un élément clé pour la compréhension des interactions lumineuses ayant lieu au sein d'un couvert végétal. Le site de perception responsable du contrôle du tallage est difficilement accessible et le délai entre l'apparition des nouvelles feuilles sur le plateau de tallage et leur observation en dehors de la gaine ajoute des difficultés dans la compréhension de leur photomorphogenèse. L'étude de la photomorphogenèse des graminées doit prendre en compte un décalage entre la réception d'un signal lumineux par les sites de perception et l'observation des effets de ce signal, pour les réponses ayant lieu durant les phases de croissances cachées, c'est-à-dire entre l'émission d'une feuille et son émergence.

4. Mesure de la photomorphogenèse

Pour pouvoir comprendre les réponses de photomorphogenèse il faut à la fois être capable d'évaluer les conditions lumineuses dans lesquelles se situe une plante et de mesurer les

modifications de l'architecture des plantes. Pour remplir ces deux conditions, plusieurs méthodes de mesures existent.

4.1. Mesure de la morphogenèse

L'une des premières difficultés de la caractérisation de la photomorphogenèse est la mesure de l'architecture des plantes. En effet, l'architecture prend en compte la forme, l'orientation et la position de l'ensemble des organes. Sa mesure dans un couvert est à la fois contraignante en termes de temps, en termes de méthode puisque la manipulation des plantes modifie leur arrangement dans l'espace et en termes d'impact sur le développement des plantes lié à ces manipulations. En raison de ces contraintes, la photomorphogenèse est majoritairement évaluée à la récolte du couvert (Holmes and Smith 1977; Casal et al. 1985; Kwesiga and Grace 1986; Davis and Simmons 1994; Lee et al. 1996; Sparkes and King 2008) ce qui permet une mesure simple et précise des organes mais ignore l'agencement des organes dans l'espace et donc l'architecture des plantes du couvert. Une autre méthode d'estimation de la structure du couvert passe par la mesure du LAI (Ballaré et al. 1987; Smith et al. 1990), qui permet de prendre en compte l'évolution du couvert au cours du temps mais, contrairement à la mesure des organes à la récolte, ignore les spécificités topologiques des plantes présentes dans le couvert.

4.2. Mesure de l'environnement lumineux

La réponse des plantes au voisinage correspond à l'évaluation de l'impact des variations de R et de FR sur les phytochromes. Les réponses de photomorphogenèse sont liées aux variations du photoéquilibre des phytochromes (Morgan and Smith 1976, 1978; Smith and Holmes 1977). Cependant, ce photoéquilibre des phytochromes ne peut être quantifié que par un dosage nécessitant des mesures destructives en broyant les échantillons pour mesurer la concentration de chaque forme du phytochrome. Afin de simplifier la mesure du photoéquilibre des phytochromes une estimation est faite à partir de l'environnement lumineux en faisant le ratio entre le flux de photon dans le domaine du R et le flux de photon dans le domaine du FR (ratio R:FR), cette estimation suit une relation présentée en figure 11. Ainsi, le ratio R:FR est une variable environnementale qui permet de décrire la réponse des plantes à leur environnement lumineux.

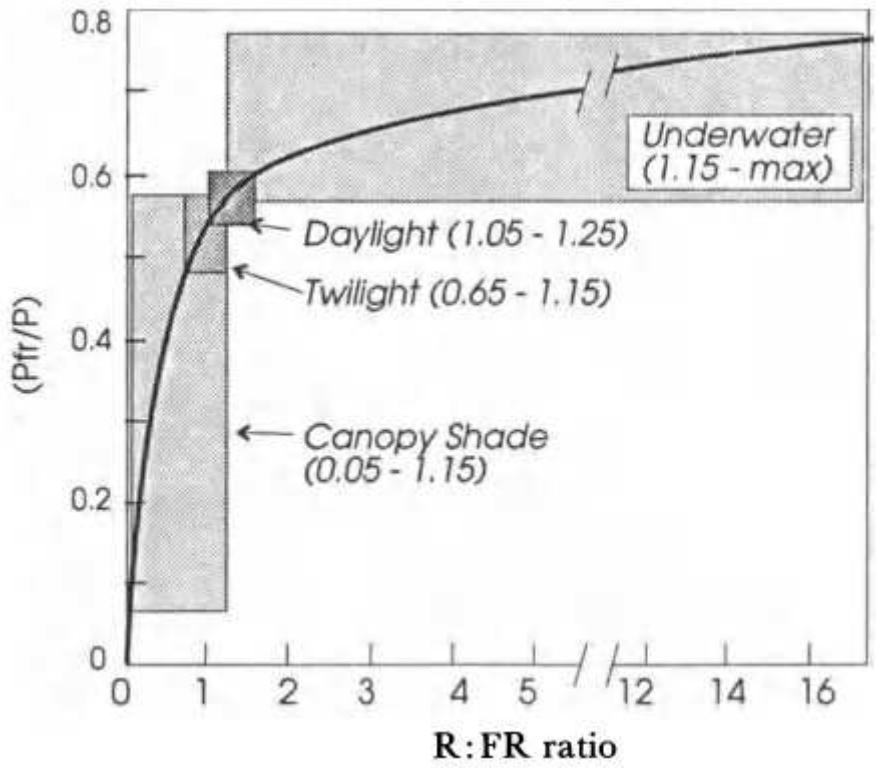


Figure 11: Relation entre le R:FR et le photoéquilibre des phytochromes, repris de .

Le R:FR peut être calculé en faisant le rapport du flux de photon des longueurs d'ondes comprises entre 600 nm et 700 nm par le flux de photon des longueurs d'ondes comprises entre 700 nm et 800 nm. ζ (zêta) est un autre descripteur de ce rapport proposé par Monteith (1976), il se réfère aux pics d'absorption des phytochromes et correspond au rapport suivant :

$$\zeta = \frac{rayonnement\ entre\ 655\,nm\ et\ 665\,nm}{rayonnement\ entre\ 725\,nm\ et\ 735\,nm}$$

Afin d'étudier l'effet du ζ sur la photomorphogenèse, il faut d'abord caractériser sa variabilité et pour cela ce signal doit être mesuré. L'emploi de spectroradiomètre couplé à un capteur permet une caractérisation fine.

4.3. Conditions environnementales de la mesure du R:FR

La mesure de l'impact du R:FR sur la morphogenèse a principalement été évaluée en utilisant ses variations naturelles, par exemple en fonction de la densité du couvert (Kasperbauer 1971; Kasperbauer and Karlen 1994), ou en modifiant artificiellement sa valeur dans le rayonnement

incident. Cette dernière méthode repose sur la modification du R:FR par l'application d'une lumière incidente avec un R:FR connu sur une plante ou un organe, soit en utilisant des filtres (Lecharny 1979; Casal et al. 1987; Ugarte et al. 2010), soit en ajoutant une lumière FR supplémentaire à l'aide de lumière incandescente (Ballaré et al. 1989) ou de LED (Sparkes and King 2008; Pantazopoulou et al. 2017). Cependant, ces deux approches présentent des limitations concernant la caractérisation du ratio réellement perçu en relation avec les réponses morphogénétiques observées. En particulier, elles ignorent toutes deux les changements locaux de R:FR au cours du développement de la plante, induits par la diffusion du FR par les plantes voisines (lorsqu'elles sont testées sur plus d'une plante individuelle). De manière générale, cela soulève la question de la perception de la lumière fluctuante caractérisée par des R:FR variables sur la morphogenèse des plantes, qui est encore difficile à évaluer. Par conséquent, l'utilisation de FSPM pour simuler la lumière caractérisée par un R:FR aux sites de perception à chaque stade du développement est nécessaire pour une meilleure compréhension des réponses des plantes à ce signal.

4.4. Méthode de mesure du R:FR

Que le R:FR soit naturel ou modifié artificiellement, sa valeur peut être caractérisée en mesurant l'environnement lumineux à l'aide d'un capteur hémisphérique (Casal et al. 1987; Davis and Simmons 1994; Wherley et al. 2005) ou directionnel (Ballaré et al. 1987; Evers et al. 2006; Chelle et al. 2007) (Fig. 12).

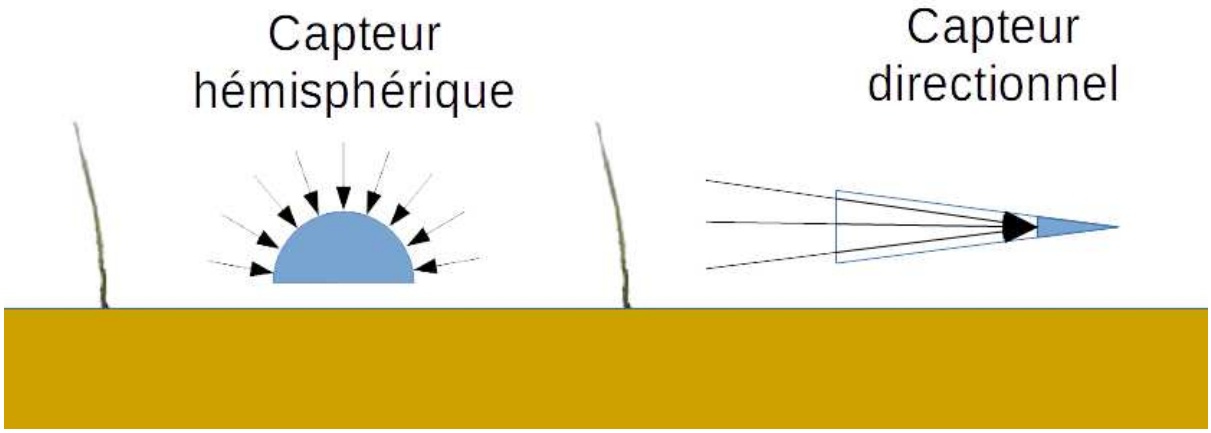


Figure 12: Différence de mesure entre les capteurs hémisphériques et les capteurs directionnels.

La majorité des recherches sur l'impact du R:FR sur le développement des plantes utilise une mesure hémisphérique de l'environnement lumineux. Ces capteurs hémisphériques permettent de mesurer le spectre de lumière provenant de toutes les directions (mesure à 360°) sur un plan et permettent une évaluation des conditions de qualité de lumière à l'intérieur d'une canopée. À des stades précoces, la canopée d'un couvert est très faible, les mesures hémisphériques intègrent donc principalement le spectre du ciel et la contribution du rayonnement des plantules est faible en proportion de rayonnement reçu, n'occupant qu'une faible partie de la zone mesurée par le capteur. Des mesures réalisées par Sattin et al. (1994) et Ballaré et al. (1987) ont mis en évidence que la diminution du R:FR dans le couvert n'était détectable qu'à partir d'un seuil de LAI alors que des variations de ce ratio étaient mesurables par des capteurs ayant un champ de vision restreint (Fig. 13). De plus, les mesures réalisées à l'aide d'un capteur hémisphérique présentent des variations en fonction des conditions météorologiques (Chelle et al. 2007). Pour limiter ces changements les mesures hémisphériques sont généralement réalisées sous un ciel bleu à midi (Kruk et al. 2006; Zhu et al. 2014), réduisant également l'ombre portée au sein du couvert, car l'élévation du soleil y est maximale.

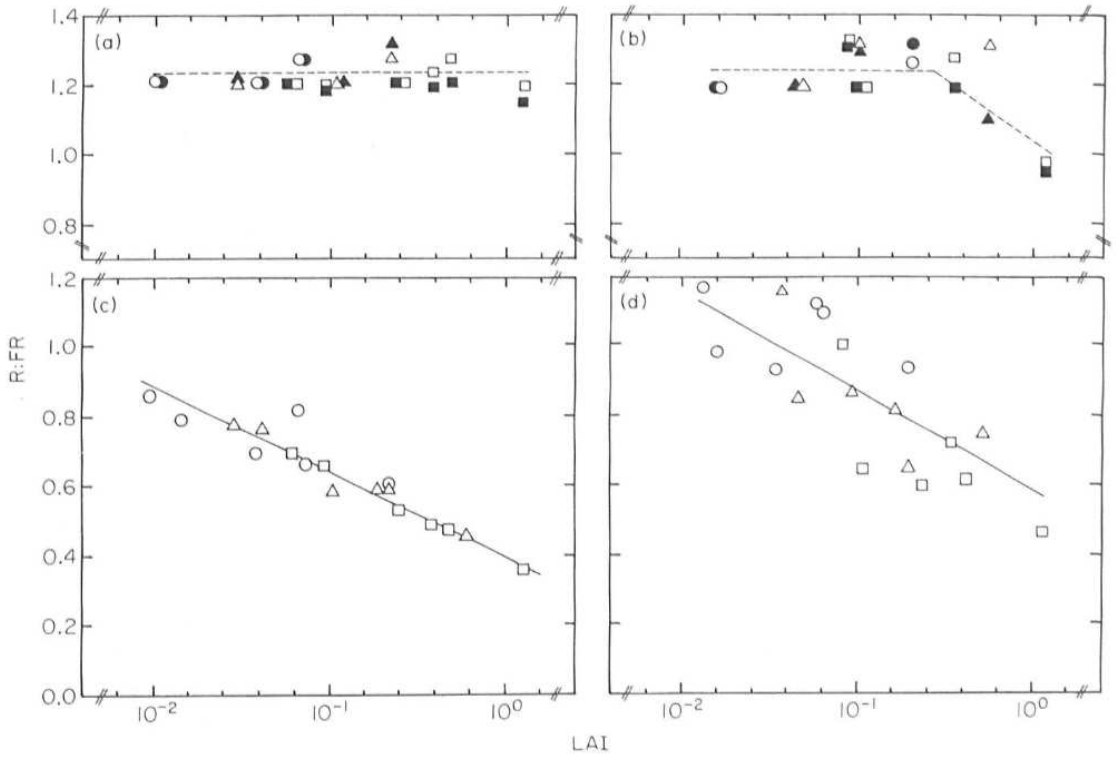


Figure 13: Relation entre le ζ et le LAI dans un couvert de Datura ferox. Les mesures sont faites à midi (a, c) ou en fin d'après-midi (b, d) en utilisant un capteur hémisphérique (a, b) ou un cylindre intégrant (c, d). Repris de

Les capteurs directionnels sont définis par un champ de vision ou FOV (Field Of View) qui correspond à l'espace contenu dans un cône d'angle connu. Ils permettent une mesure plus fine de l'environnement lumineux en capturant précisément la lumière renvoyée par les organes des plantes, cette mesure directionnelle permet ainsi de prendre en compte la géométrie des organes et l'architecture des plantes mesurées. L'une des raisons de la faible utilisation des capteurs directionnels est liée au fait que cette technologie n'est disponible que depuis relativement récemment par rapport à l'ensemble des travaux concernant les effets photomorphogénétiques du R:FR. La première tentative de mesurer de l'environnement lumineux à l'aide d'un capteur directionnel a été réalisée par Ballaré et al. (1987). Dans cette expérimentation le capteur utilisé est un capteur hémisphérique ayant été modifié pour que la lumière soit mesurée à 360° mais uniquement dans une hauteur de cône de 10°, désigné comme « cylindre intégrant » (Fig. 14). L'utilisation de capteurs directionnel a pu mettre en évidence une diminution du ζ de la lumière émise par les plantes dès les premières phases du développement qui n'étaient pas observables en utilisant des capteurs hémisphériques (Fig. 13)(Ballaré et al. 1987, 1990).

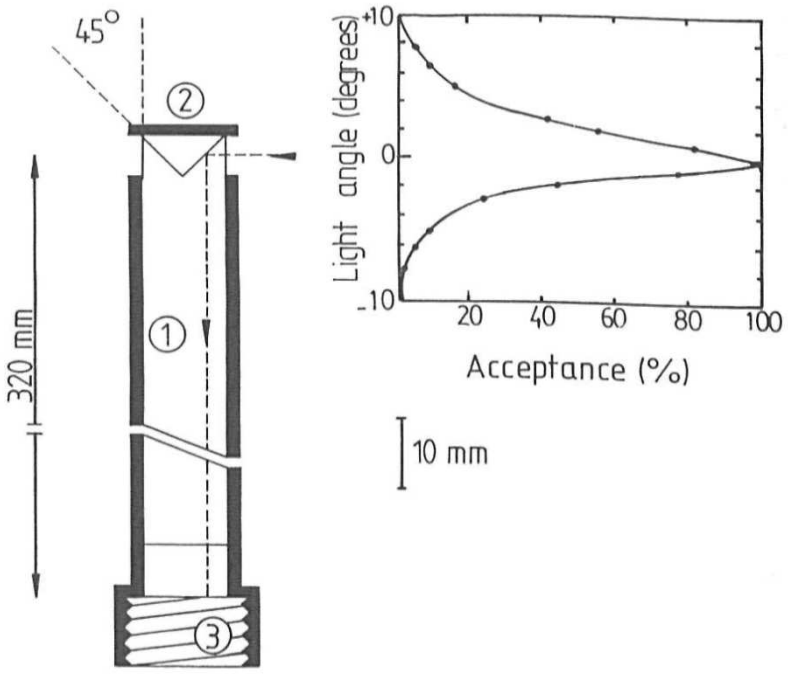


Figure 14: Schéma du cylindre d'intégration créé par comme première approche de mesure dans un angle de vue restreint. (1) Barre d'acrylique transparent, (2) Revêtement noir, (3) Assemblage pour la sonde du spectroradiomètre. La figure de droite montre l'angle d'ouverture du cylindre.

Bien que le « champ de vision des phytochromes » soit encore inconnus, il est nécessaire d'utiliser des outils capables de mesurer les variations de ζ les plus minimes afin de comprendre les mécanismes de photomorphogenèse dans un couvert prairial aux stades précoces de développement. Afin de déterminer à quel FOV les mesures d'environnement lumineux réalisées avec un capteur directionnel devaient être effectuées, des tests de champ de vision ont été effectuées (Fig. 15, données non publiées). Ces mesures ont mis en évidence que l'utilisation d'un capteur directionnel ayant un FOV de 3° permettait de détecter la présence de plantes faiblement développées.

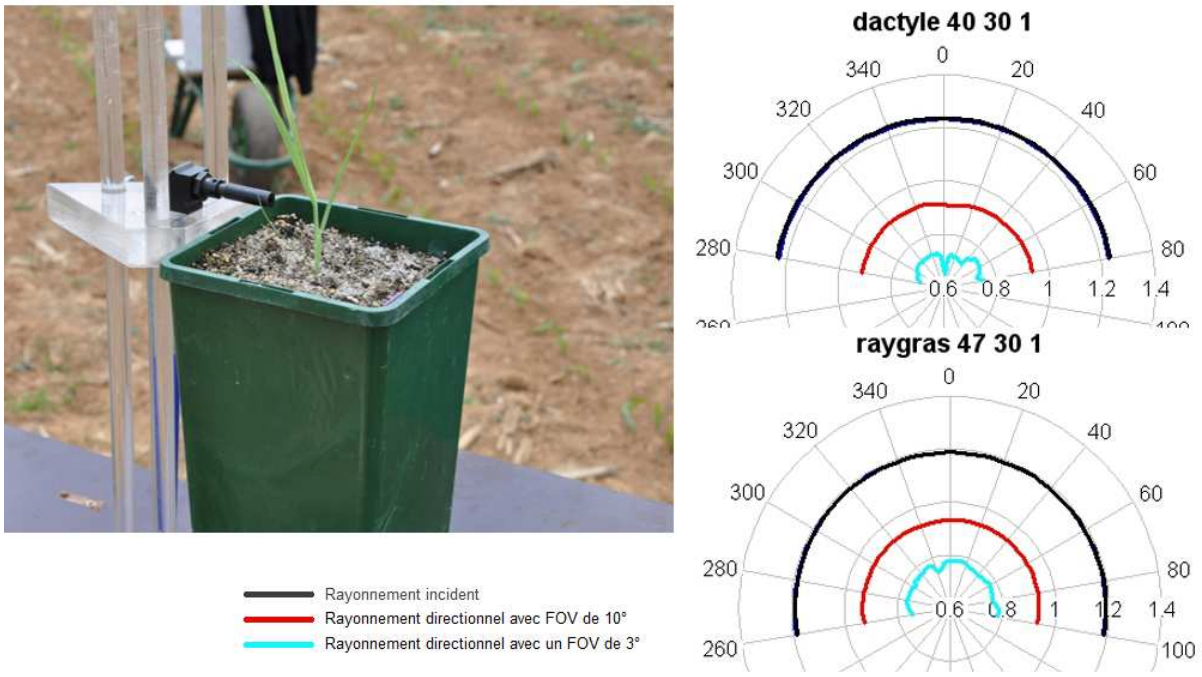


Figure 15: Évaluation du champ de vision nécessaire à la détection d'une graminée isolée peu développée. Le ζ a été mesuré pour des jeunes pousses de dactyle et de ray-gras à des angles de vue de 3° et de 10°.

5. Modélisation des interactions plantes-environnement

5.1. Représentation de la photomorphogenèse dans la modélisation

La modélisation de la photomorphogenèse nécessite la prise en compte de l'architecture des plantes durant le développement du couvert. Cette représentation 3D des plantes est permise par les modèles fonctionnels plante-structure (FSPM) qui représentent explicitement la structure spatiale des plantes (Vos et al. 2010). Afin de modéliser la photomorphogenèse, cette représentation explicite de la structure de la plante est couplée à un modèle de lumière permettant de simuler le R:FR ou ζ et ses variations au sein du couvert (Gautier et al. 2000; Evers et al. 2007; Kahlen et al. 2008; Barillot et al. 2014).

On peut distinguer 3 approches de modélisations de la lumière dans les FSPM.

L'approche de radiosité est une méthode projective basée sur le calcul de la projection des composants explicites de la végétation (feuilles, tige, etc.) sur un plan perpendiculaire à la direction du rayonnement incident. Couplée aux FSPM, cette méthode fournit plus d'informations sur l'interception de la lumière à l'échelle de l'organe que les modèles radiatifs basés sur l'approche du milieu turbide.

Depuis les travaux pionniers de Gautier et al. (2000) sur le trèfle blanc, de nouveaux modèles ont été publiés qui simulent la lumière caractérisée par R:FR et les réponses associées des plantes. Par exemple, on peut trouver des études concernant les effets de la lumière avec différentes valeurs de R:FR sur : le tallage du blé (Evers et al. 2007) : la longueur finale des entre-nœuds du concombre (Kahlen et al. 2008; Kahlen and Stützel 2011) ; la longueur des entre-nœuds du pois (Barillot et al. 2014) ou des légumineuses fourragères (Louarn and Faverjon 2018). Cette méthode de radiosité permet de traiter la diffusion de la lumière par les phytoéléments et nécessite un temps de calcul important.

L'approche des lancers de rayons, qui consiste à faire partir un nombre de rayons défini pour chaque pixel d'une scène finale vers les objets de la scène. Les rayons sont ensuite réfléchis ou transmis selon les propriétés optiques de ces objets jusqu'à ce que les rayons rencontrent un objet considéré comme étant la source lumineuse (généralement le soleil).

L'approche turbid-medium consiste à considérer la structure de la canopée végétale comme un milieu turbide à une seule couche. Cette méthode considère la canopée végétale comme une couche homogène horizontale composée d'éléments orientés. Les éléments végétaux sont supposés avoir une dispersion aléatoire et une taille négligeable, occupant de manière homogène un volume donné. À partir de ces approximations, l'atténuation du rayonnement dans la zone de calcul suit la loi de Beer-Lambert. Cette approche peut également être réalisée en prenant en compte l'hétérogénéité spatiale et la dispersion non aléatoire des organes. L'approche turbid-medium ne prend donc pas en compte l'ensemble de l'architecture des plantes contrairement aux approches radiosité et lancer de rayons.

5.2. Modélisation de la mesure des signaux

La majorité des recherches sur la photomorphogenèse sont basées sur l'utilisation de capteurs hémisphériques. Pour cette raison, les modèles existants capables de modéliser la photomorphogenèse utilisent également une représentation hémisphérique des capteurs.

Une première approche de modélisation de la mesure directionnelle des signaux lumineux a été effectuée par Chelle et al. (2007). Dans cette étude, l'effet de l'orientation de capteurs hémisphériques (capteur positionné au niveau du sol et orienté vers le ciel ou vers le couvert) sur la mesure du ζ est évaluée pour des conditions de ciel clair et de ciel couvert pour 2 densités (Fig. 16). Ces simulations ont mis en évidence une détection de diminution du ζ plus importante lorsque les capteurs sont orientés vers le couvert.

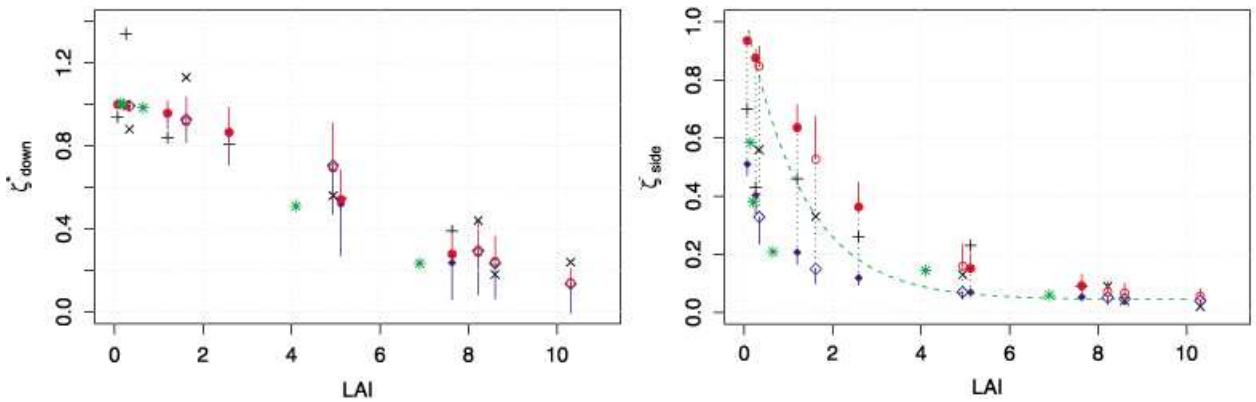


Figure 16: Comparaison des valeurs observées et simulées de ζ juste au-dessus du sol (gauche) pour le flux descendant et (droite) pour le flux latéral venant du nord. Les valeurs moyennes simulées de ζ sont représentées par des cercles pour les conditions standard de ciel couvert (SOC) et par des diamants pour des conditions de ciel clair entre 12 h et 14 h UTC. Les symboles pleins et ouverts correspondent à des densités de plantes faibles et élevées, respectivement. Les lignes en pointillés reliant les cercles et les diamants représentent la plage de variation de la moyenne simulée ζ entre les conditions de ciel clair et de ciel couvert. Les barres d'erreur supérieures et inférieures représentent l'écart-type du ζ simulé avec la mesure. Repris de .

L'effet de l'angle de vue des capteurs a également été évalué et des capteurs directionnels ont donc été modélisés pour des FOV de 80° et de 40° et dont les résultats ont été comparés avec un capteur hémisphérique (Fig. 16). Cette modélisation met en évidence une plus forte sensibilité de détection de la diminution du ζ en lien avec l'augmentation du LAI lorsque l'on réduit le FOV des capteurs, cependant le ζ simulé par les 3 capteurs est similaire pour des conditions de ciel clair autour du midi solaire.

6. Problématique

La compétition pour la lumière est très présente dans les systèmes prairiaux (Wedin and Tilman 1993; Lemaire et al. 2005). Dans le contexte actuel de changement climatique la diversité génétique représentée par des mélanges de variétés (en monospécifique) et/ou d'espèces (mélanges binaires ou multi-espèces) est utilisée comme levier potentiel d'adaptation des prairies semées aux contraintes imposées par ce changement (Violle et al. 2015). Cette diversité génétique se traduit par une diversité de phénotypes représentés par des plantes à architectures contrastées pouvant exacerber les phénomènes de compétition pour les ressources, lumière en premier lieu. La compréhension des interactions lumineuses ayant lieu au sein d'un couvert prairial pourrait permettre un meilleur contrôle de la qualité des fourrages produits par celui-ci.

Pour mieux évaluer les effets de la compétition pour la lumière entre les plantes il est nécessaire d'analyser jusqu'à quel degré une plante est capable de détecter la présence de plantes voisines avant la diminution quantitative de cette ressource et donc dès les premiers stades de développement.

En effet, dans ces conditions la diversité génétique résultante va dépendre de l'évolution de la structure des plantes, c'est-à-dire de la morphogenèse. L'impact des plantes voisines sur cette morphogenèse ne doit donc pas être seulement étudiée une fois la compétition installée mais aussi et surtout dès les phases précoces de développement des couverts. L'analyse des signaux lumineux qui parviennent aux plantes et en particulier le ζ est donc nécessaire et l'approche directionnelle du signal semble la plus appropriée compte tenu de la taille des plantes et la finesse de leurs organes.

Afin de décrire avec précision l'environnement lumineux des plantes aux premiers stades de leur développement, davantage d'études sur l'impact du ζ sur la photomorphogenèse doivent être menées avec des capteurs directionnels pour améliorer sa modélisation de ce signal et des réponses.

Dans une optique de compréhension des interactions plante-plante au travers des signaux lumineux dans un couvert, l'utilisation d'un modèle existant capable de simuler des capteurs hémisphériques pose la question de l'impact du mode de simulation du ζ sur la simulation de la morphogenèse.

La volonté de comprendre la photomorphogenèse en se rapprochant le plus possible des variations de l'environnement lumineux expérimenté par les plantes tend vers l'utilisation de capteur directionnel. Les modèles de rayonnement surfacique existants ne permettant pas pour le

moment la modélisation d'un capteur directionnel. Pour ces modèles la création d'un capteur directionnel est nécessaire, on peut cependant s'interroger sur leur capacité à simuler le ζ directionnel et sur la sensibilité de ces simulations aux paramètres d'entrée.

Pour répondre à ces questions, cette thèse propose en premier lieu de caractériser par la mesure la variabilité du signal ζ. Le ζ a été calculé à partir du rayonnement renvoyé par des plants de ray-grass d'architecture contrastée dans différentes configurations afin d'évaluer si la distance, l'architecture (l'état de développement) ou encore les conditions météorologiques influencent le signal qui se propage de manière directionnelle. En second lieu, des approches de modélisation directionnelle du ζ ont été testées et évaluées à partir des mesures précédemment décrites.

III. Matériels et Méthodes

1. Modèles de lumière

Afin de modéliser la photomorphogenèse dans des modèles FSP, le calcul du ζ intercepté par les organes des plantes est nécessaire afin estimer l'importance de ce processus.

1.1. CARIBU

CARIBU est une interface logicielle développée par Chelle et al. (2004) permettant de relier le modèle d'échange radiatifs CANESTRA à des modèles de L-systèmes. Le modèle CANESTRA est basé sur le calcul de la lumière incidente interceptée par une surface donnée dans un couvert végétal par une méthode projective (Chelle et al. 1998; Chelle and Andrieu 1998) effectué par les processeurs. Dans cette méthode, la structure de la plante est explicitement décrite en représentant individuellement chaque organe par un ensemble de triangles. Chaque triangle est projeté parallèlement à la direction de la lumière sur un plan perpendiculaire à la direction d'incidence. Chaque triangle projeté sur le plan est représenté par un ensemble de pixels (Fig.17).

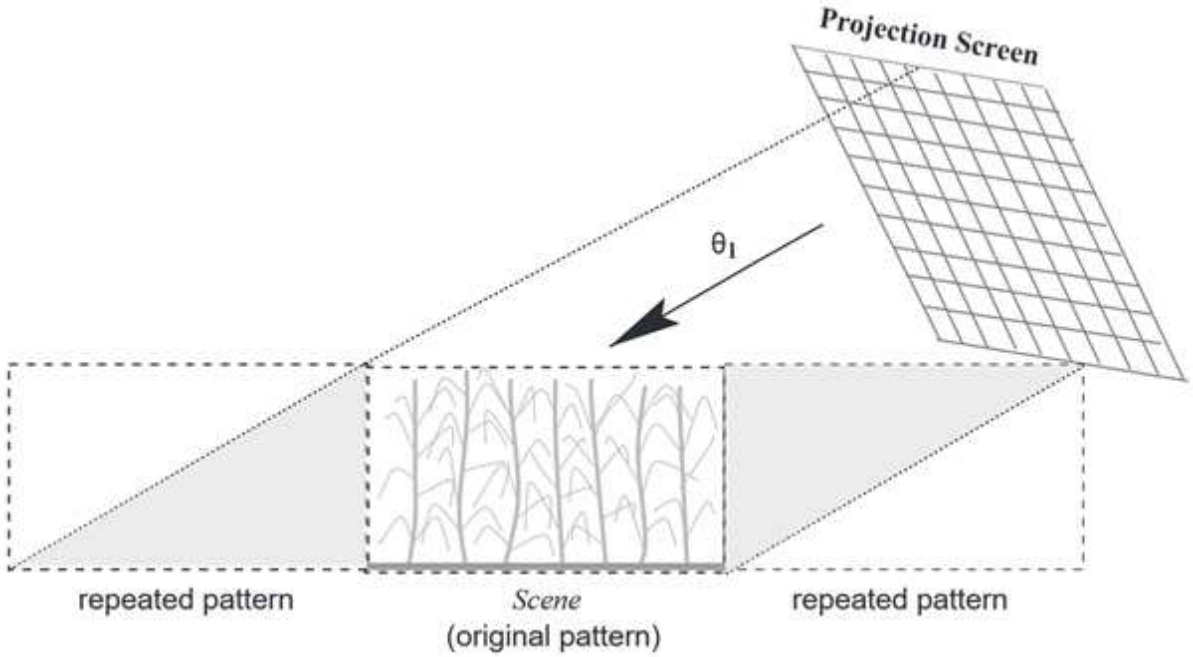


Figure 17: Schéma de calcul du rayonnement dans CARIBU, repris de . Le pattern de la scène peut être représenté de manière infini.

L'information portée par chaque pixel est mise à jour en comparant la distance des triangles au plan, afin de prendre en compte les chevauchements. Lorsque l'ensemble des triangles est projeté sur le plan, leur irradiance est estimée en utilisant l'irradiance d'un plan horizontal au-dessus de la

canopée. L'irradiance est une grandeur physique qui définit la quantité de rayonnement reçue par une surface, exprimée en W.m-2.

1.1.1. Paramètre d'entrée du modèle

Lors de la construction de la scène, le modèle CARIBU récupère la scène OpenAlea ainsi que les données de ciel. En complément, les propriétés optiques de l'ensemble des éléments de la scène ainsi que le pattern, qui définit la zone de la scène sur lequel les calculs du modèle doivent être effectués, sont demandés par le modèle. C'est à partir de cette scène au format CARIBU que les calculs de rayonnement sont effectués.

1.1.2. Calcul du rayonnement dans la scène

Avant de calculer le rayonnement de la scène, le pattern sur lequel sont effectués les calculs peut être répété de manière infinie pour reproduire un large couvert et supprimer les effets de bord. Ce calcul peut être réalisé de manière directe, c'est-à-dire que seules les interceptions de premier ordre sont calculées, ou avec la prise en compte des rediffusions. Si les simulations prennent en compte les rediffusions, les triangles présents dans la scène doivent chacun posséder des propriétés optiques de réflectance et de transmittance qui peuvent être identiques ou différentes pour les deux faces d'un même triangle.

Le calcul du rayonnement se fait par une projection sur un écran dont la taille en pixels est définie dans les paramètres d'entrée du modèle. La résolution optimale de l'écran est choisie en comparant les résultats à ceux obtenus à l'aide du modèle de lancer de rayons Parcimonie (PARticle traCINg in the canOPY) Monte Carlo ray tracing (Chelle 1996). Les interactions radiatives aux limites latérales de la canopée végétale sont traitées en considérant une répétition infinie et périodique de cette canopée avec les mêmes propriétés structurelles (Chelle et al. 1998).

CARIBU permet également de simuler le rayonnement intercepté par un capteur virtuel, qui correspond à un ensemble de triangles interceptant la lumière à un endroit donné de la scène mais qui n'interfère pas sur les interactions lumineuses (rediffusions). Les capteurs virtuels sont définis par un dictionnaire python qui contient les coordonnées cartésiennes de chaque triangle pour chacun de ses sommets. Ainsi, CARIBU génère des sorties au format csv contenant le rayonnement intercepté pour chaque triangle ou par un ensemble de triangles pour les longueurs d'onde spécifiées dans les entrées du modèle.

1.2. RIRI

RIRI (Radiation Interception in Row Intercropping) est un modèle de radiosité basé sur une approche turbid-medium (Sinoquet and Bonhomme 1992). Dans ce modèle, la structure de la canopée des plantes est décrite comme un ensemble de voxels contigus (Fig. 18).

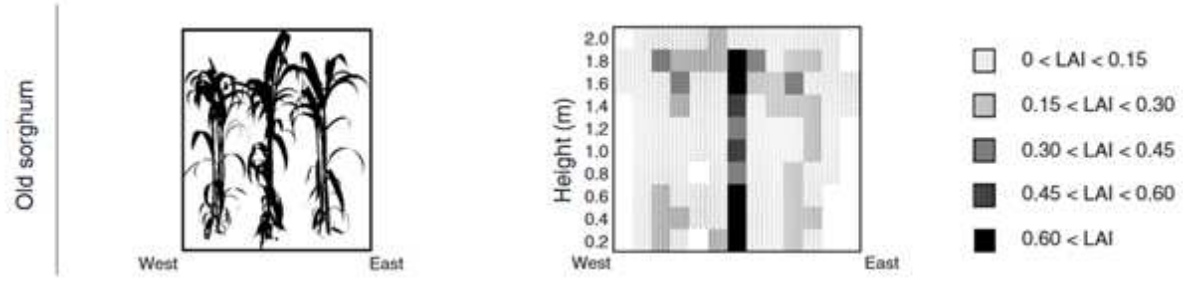


Figure 18: Représentation 3D spatialement explicite de canopée (gauche) et sa structure correspondante considérés comme des voxels contigus contenant l'information du LAI (droite). Repris de .

Le modèle RIRI a été implémenté en Python sur la plateforme OpenAlea (Pradal et al. 2008). Dans le cadre de ce travail, le modèle RIRI est couplé au FSPM L-py (Boudon et al. 2012) qui permet une représentation de l'architecture aérienne basée sur les L-systèmes (Prusinkiewicz and Lindenmayer 1990). Cette intégration de RIRI dans le modèle L-py permet de calculer l'interception du PAR par les différentes composantes de la canopée. Ce modèle turbid-medium nécessite des données météorologiques pour chacun des pas de temps de la simulation, ainsi qu'une représentation discrétisée à l'échelle du voxel de la distribution de la densité de la surface foliaire et de la distribution de l'angle des feuilles dans une grille 3D pour chaque plante à exécuter. La discrétisation permet de résumer en une seule valeur l'ensemble du rayonnement intercepté par le voxel au cours d'une journée. De plus, la taille des voxels utilisée pour les calculs est définie dans les entrées du modèle, avec une taille par défaut de 4 × 4 × 2 cm.

1.2.1. Calcul du rayonnement

Une approche probabiliste est utilisée pour calculer l'interception de la lumière : l'extinction du rayonnement sur son chemin entre le ciel et un point à l'intérieur de la canopée est calculée selon la loi de Beer-Lambert, qui considère une dispersion spatiale aléatoire des feuilles et une distribution uniforme de l'azimut des feuilles. Au sein de chaque voxel, les feuilles présentes sont caractérisées par deux paramètres structurels : la fonction de distribution de l'inclinaison des feuilles (LIDF) et la densité de la surface foliaire (LAD). La LIDF peut être définie dans les paramètres d'entrée comme distincte pour chaque voxel ou égale à la LIDF de la canopée entière. Dans le cas

où la scène simulée est infinie, les voxels contigus sont répétés de façon périodique et infinie avec les mêmes propriétés que le modèle initial. Le PAR intercepté est calculé selon le pas de temps du modèle pour chaque élément de la scène.

1.3. MALIA

MALIA est un moteur de rendu open-source développé par l'équipe MANAO du LABRI de Bordeaux (INRIA) (https://pacanows.gitlabpages.inria.fr/MRF/main.md.html). Ce moteur de rendu utilise à la fois le lancer de rayons, qui permet le calcul de rayonnement, et l'intégration de Monte-Carlo, qui permet d'obtenir le chemin parcouru par chaque rayon, pour obtenir des images RGB ou spectrales d'une scène 3D (Fig. 19).

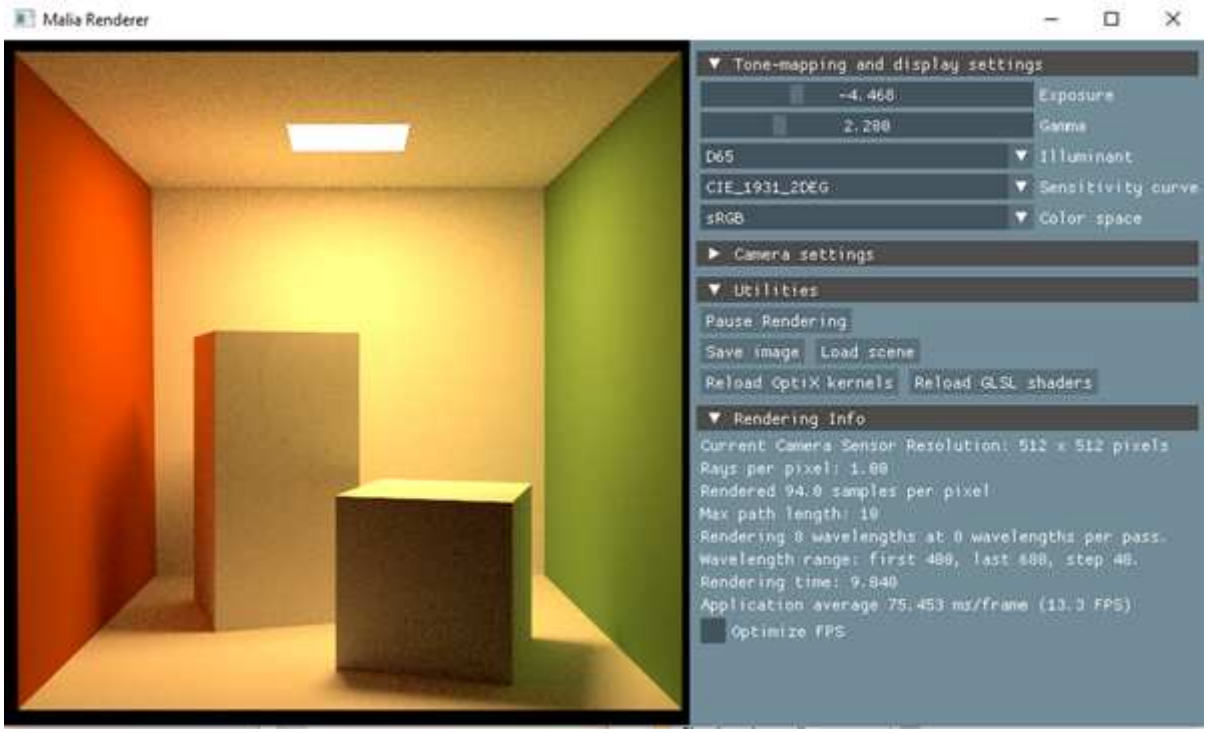


Figure 19: Interface du logiciel MALIA.

Les sorties de MALIA sont sous le format d'une image spectrale contenant le rayonnement intercepté par chaque pixel pour chacune des longueurs d'onde. La conservation du rendu sous ce format permet de réaliser un travail post-traitement sans avoir besoin de relancer la simulation.

Malia offre également comme avantage la possibilité de tenir compte des propriétés non lambertiennes des feuilles, une implémentation simple de nouveaux modèles de ciel et la possibilité d'obtenir les trajets optiques.

1.3.1. Paramètre d'entrée du modèle

Afin de fonctionner, MALIA nécessite un fichier de scène et un fichier de caméra qui doivent être en format XML. Le fichier de la scène contient le choix du modèle de ciel, la position du soleil, le spectre de l'environnement lumineux, le chemin de fichier de chaque objet 3D, leur position cartésienne dans la scène ainsi que les PO correspondants à chacun de ces objets. Le fichier de caméra peut contenir plusieurs caméras. Pour chacune d'entre elles, le fichier doit préciser sa position cartésienne, le point vers lequel la caméra est orientée, son champ de vision (en °) et sa résolution en pixels qui doit être au minimum de 500 × 500.

1.3.2. Calcul du rayonnement

Étant basé sur le lancer de rayons, le modèle MALIA ne permet pas de simulation sans rediffusion. Le spectre lumineux du ciel est indiqué dans les paramètres d'entrée du modèle, allant de 300 à 800 nm, ainsi que le nombre de rayons par pixel. Ces rayons partent d'une ou de plusieurs caméras, selon les besoins (ici une seule caméra est utilisée), et seuls les rayons atteignant la source lumineuse (Soleil) par rediffusion sont pris en compte pour le calcul du spectre de chaque pixel du rendu. Ce calcul prend en compte les propriétés optiques de chaque élément de la scène.

1.4. Construction du ciel dans les différents modèles

Pour construire le ciel utilisé par ces modèles de lumière, différentes variables d'entrées sont utilisées.

La position du soleil, est calculée à partir de la date, de l'heure et des coordonnées géographique du point d'observation. Pour le site de Lusignan, les coordonnées utilisées ont pour latitude 46.4036827 et pour longitude 0.0787160.

Le type de rayonnement est également pris en compte, les simulations peuvent être effectuées avec un rayonnement direct, reproduisant un ciel clair, ou diffus, représentant un ciel couvert.

Pour les modèles CARIBU et RIRI, le rayonnement clair est réalisé avec 2 modèles de ciel. Le modèle Standardized Overcast sky (SOC), pour lequel le rayonnement provient à la fois du ciel et du soleil avec un poids plus important pour le soleil (Rosen and Hooper 1989). Une discrétisation du ciel avec six directions, correspondant à la lumière incidente quotidienne d'un ciel diffus, est définie par défaut pour le modèle SOC. Selon (Dulk 1989) une discrétisation du ciel à six directions est suffisante pour rendre compte des interactions plante-rayonnement. Le rayonnement diffus est

réalisé avec le modèle de ciel Uniform Overcast sky (UOC) pour lequel la distribution du rayonnement est uniforme sur l'ensemble du ciel (Rosen and Hooper 1989).

Pour le modèle MALIA, le ciel est modélisé à partir du modèle de ciel de Preetham, qui représente un ciel clair. Ce modèle génère un rendu de ciel dans un espace RGB à partir de la position du soleil, une conversion vers un format spectral est effectuée à partir de la formule de la distribution spectrale relative d'un illuminant proposée par Wyszecki & Stiles (2000).

La turbidité de l'atmosphère est également utilisée comme entrée du modèle de Preetham et al. (1999). La turbidité correspond à la teneur en microparticules dans un fluide (ici l'air) modifiant les rediffusions. Cette variable n'a pas été mesurée lors des différentes expérimentations, sa valeur par défaut dans le modèle est utilisée.

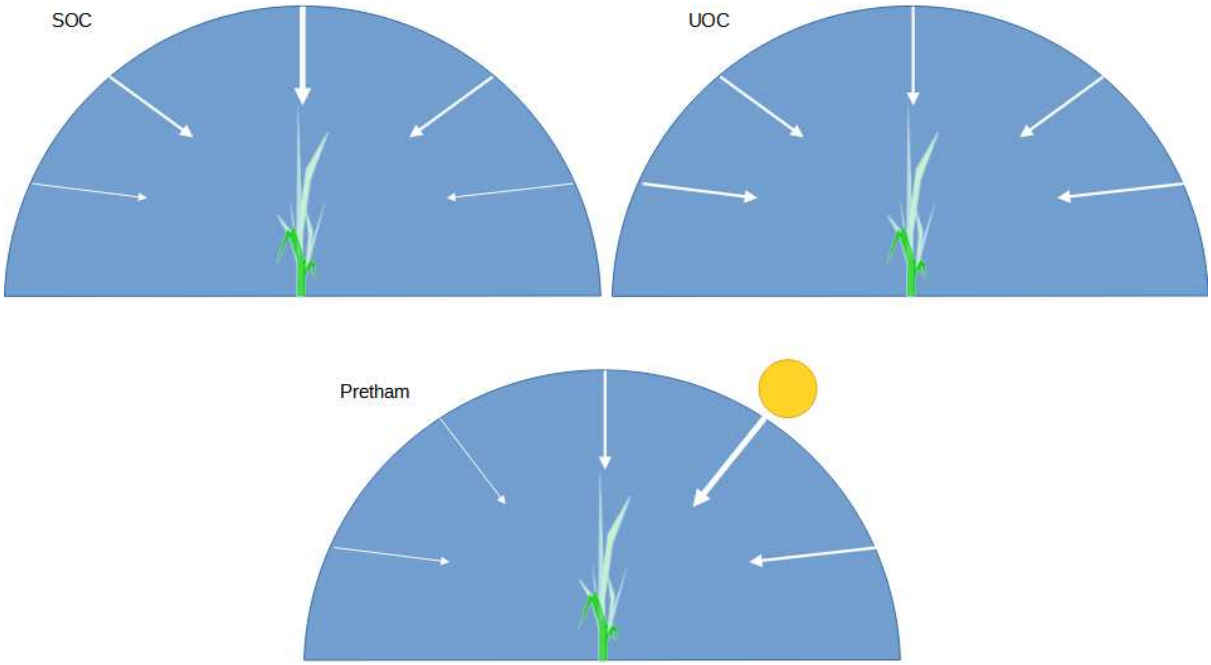


Figure 20: Schéma du fonctionnement des différents modèles de ciels utilisés. Le modèle de ciel SOC accorde un poids plus important au zénith quelque soit l'heure, le modèle UOC est homogène entre les directions quelque soit l'heure et le modèle de Pretham accorde un poids plus important aux rayons proches de la position du soleil selon l'heure de mesure.

1.5. Synthèse des modèles

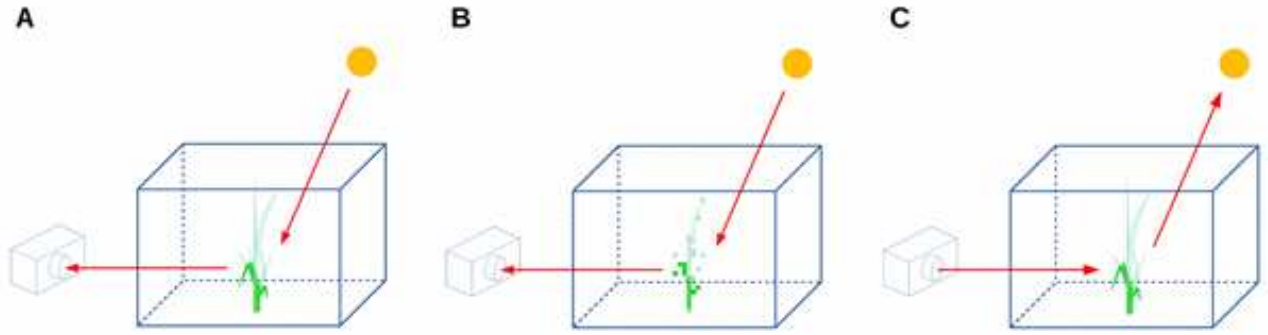


Figure 21: Schéma du calculs de rayonnement dans les différents modèles de lumière utilisés. La scène sur laquelle les calculs sont effectuée est représentée par un parallélépipède rectangle bleu. Le chemin parcouru par les rayons est représenté par les flèches rouges. La source lumineuse est représentée par le disque jaune. Le capteur est représenté par une caméra. A) Parcours d'un rayon dans le modèle CARIBU, le rayonnement est calculé à l'échelle des triangles. B) Parcours d'un rayon dans le modèle RIRI, le rayonnement est calculé à l'échelle des voxels. C) Parcours d'un rayon dans le modèle MALIA, le rayonnement est calculé à l'échelle des triangles.

Les différences entre les 3 modèles de lumière sont résumées dans le tableau de synthèse suivant :

	CARIBU	**RIRI**	**MALIA**
Méthode de calcul	Radiosité et Turbid-Medium* (*pas utilisé dans cette thèse)	Turbid-Medium	Lancer de rayons
Représentation de l'architecture	Représentation 3D de chaque organe	Angle d'orientation des feuilles (LIDF) + Surface foliaire (LAD)	Représentation 3D de chaque organe
Prise en compte de la superposition des organes	Oui	Non	Oui
Modèle de ciel	SOC UOC	SOC UOC	Preetham
Calculs	Processeurs	Processeurs	Carte graphique

En raison de leur construction, les modèles CARIBU et MALIA permettent d'obtenir le rayonnement reçu par un point précis de la scène, contrairement à RIRI. Pour cette raison, seuls CARIBU et MALIA sont utilisés pour la création de capteur de rayonnement directionnel (élément développé en partie III.2.2). De plus, MALIA offre comme avantage technique supplémentaire: l'utilisation de la carte graphique pour les calculs de rayonnement, permettant un rendu rapide.

1.6. Calcul du ζ à partir du rayonnement transmis

Le calcul des rediffusions du rayonnement intercepté par les organes des plantes est lourd en temps de calcul dans les modèles CARIBU et RIRI. Une méthode permettant de simplifier ce calcul est l'utilisation de la relation fonctionnelle proposée par Escobar-Gutiérrez et al. (2009), qui permet de calculer le ζ à partir du pourcentage de PPFD transmis (TrPPFD) à l'échelle des organes (Fig. 22). Pour être couplée aux modèles de ciel cette méthode nécessite la première étape de calcul du rayonnement intercepté, c'est-à-dire l'ensemble de l'interception par les composants de la scène avant rediffusion. La suppression du calcul des rediffusions dans les simulations est un gain de temps non négligeable qui rend l'utilisation de ces modèles plus accessibles.

Le ζ calculé à partir du TrPPFD suit la fonction de saturation non linéaire proposée et validée par Escobar-Gutiérrez et al. (2009) (Fig.22):

$$\zeta = \left[c^b + \left(d^b - c^b \right) \frac{1 - e^{-\alpha (TrPPDF - x1)}}{1 - e^{-\alpha (x2 - x1)}} \right]^{\frac{1}{b}} + \varepsilon i$$

Où :

- a = 3,09 et b = 1,59 définissent la forme de la courbe
- x1 = 0 et x2 = 2 sont respectivement les premières et dernières valeurs du domaine TrPPFD pour lequel le modèle est ajusté
- c = 0 et d = 1,12 sont les valeurs de ζ en x1 et x2 respectivement.

Cette relation entre ζ et TrPPFD utilise uniquement le TrPPFD comme paramètre physique. Par conséquent les sources de variations du ζ sont supposées être les mêmes que celles du TrPPFD (densité, propriétés optiques des organes, architecture).

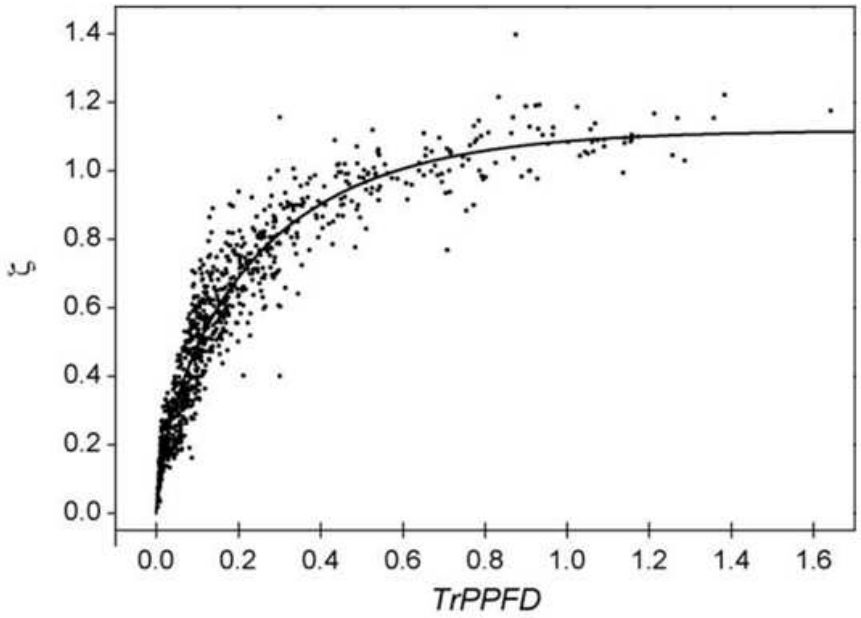


Figure 22: Relation entre le PPFD transmis et le ζ, repris de (Morgan and Smith 1978; Gillet 1980; Mazliak 1982; Ballaré et al. 1987; Smith et al. 1990; Kendrick and Kronenberg 1994; Varlet-Grancher et al. 1995; Chelle et al. 1998, 2007; Thomas 2003; Combes et al. 2008; Verdenal 2009; Escobar-Gutiérrez et al. 2009; Faverjon 2018).

2. Mesure et modélisation du rayonnement directionnel

2.1. Dispositifs de mesure directionnelle

2.1.1. Banc Spectral

Afin de mesurer la lumière réfléchie et transmise par les plantes, l'environnement lumineux est mesuré à l'aide d'un spectroradiomètre MayaPro2000 (MayP112806) relié à un capteur directionnel ayant un FOV de 3°. Le capteur du spectromètre Maya est fixé à une hauteur de 10 cm sur un dispositif nommé « banc spectral » (Fig. 23). Le banc spectral est un dispositif qui permet un déplacement horizontal d'un support mobile pouvant contenir un pot d'un diamètre de 5 cm, la gamme de distance entre le bord du pot et le capteur allant de 2 cm à 60 cm. Ce dispositif, nécessitant un support pour ses différents éléments, est principalement utilisé en intérieur (laboratoire ou sous serre).

Le spectromètre Maya est relié à un ordinateur enregistrant en temps réel l'environnement lumineux compris entre 300 nm et 1100 nm.

Une vérification de l'angle d'ouverture du capteur a été réalisée afin de s'assurer que la mesure du rayonnement soit restreinte dans un champ de vision équivalent à un angle solide de 3°. Cette vérification a été effectuée connectant la fibre à un illuminateur afin de projeter la lumière sur un plan. L'angle d'ouverture est obtenu par calcul trigonométrique entre la surface projetée et la distance au plan (calcul de la tangente de l'angle). Une fois le capteur installé sur le banc spectral la direction du capteur a été vérifiée en installant une mire sur le support de pot du banc spectral et en observant l'évolution de la tache lumineuse émise par un illuminateur en fonction de la distance (Fig. 24).

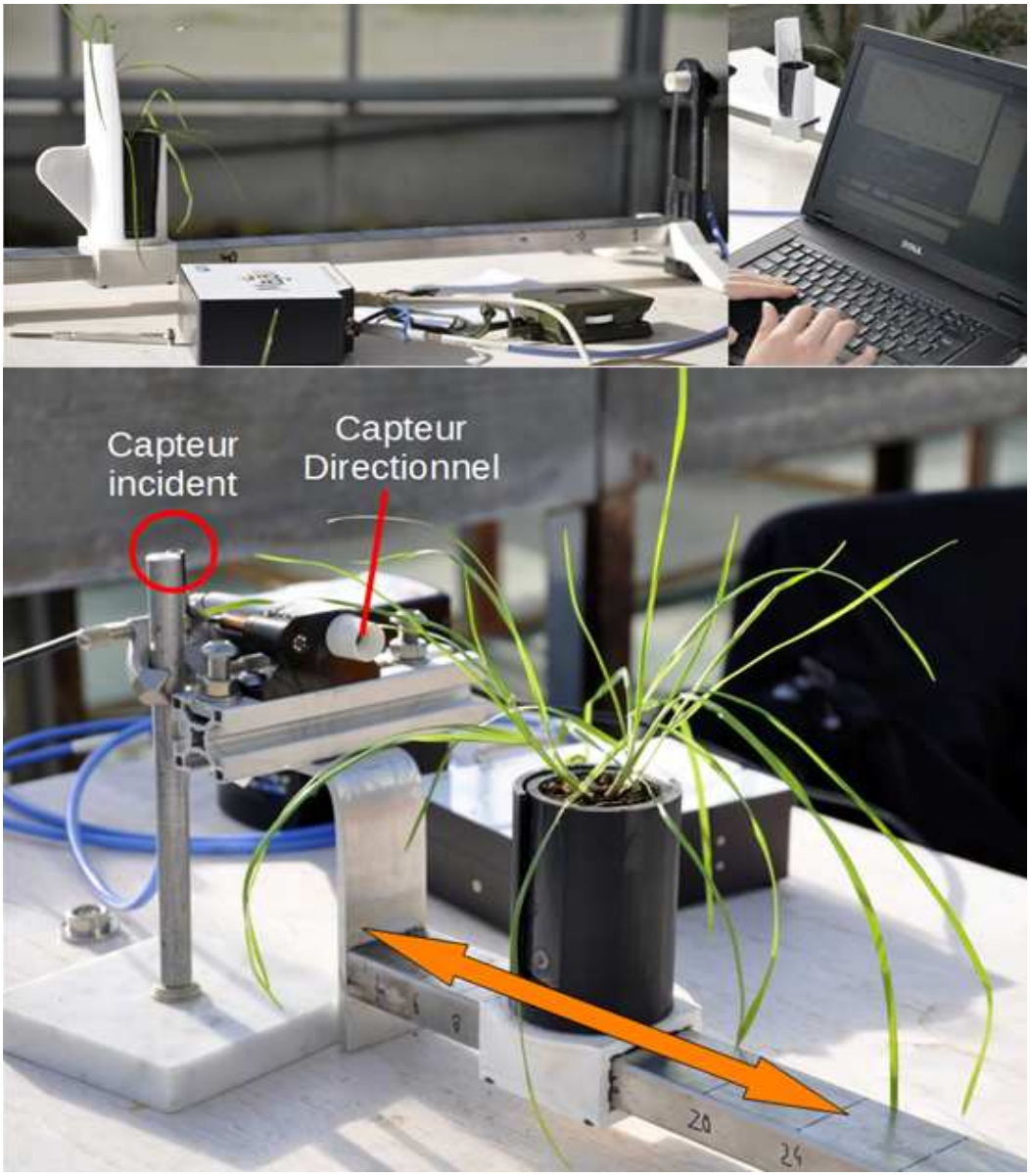


Figure 23: Dispositif du banc spectral. Le pot est positionné sur un support permettant de rapprocher ou d'éloigner la plante du capteur directionnel. Les données sont enregistrées en simultanées sur ordinateur.

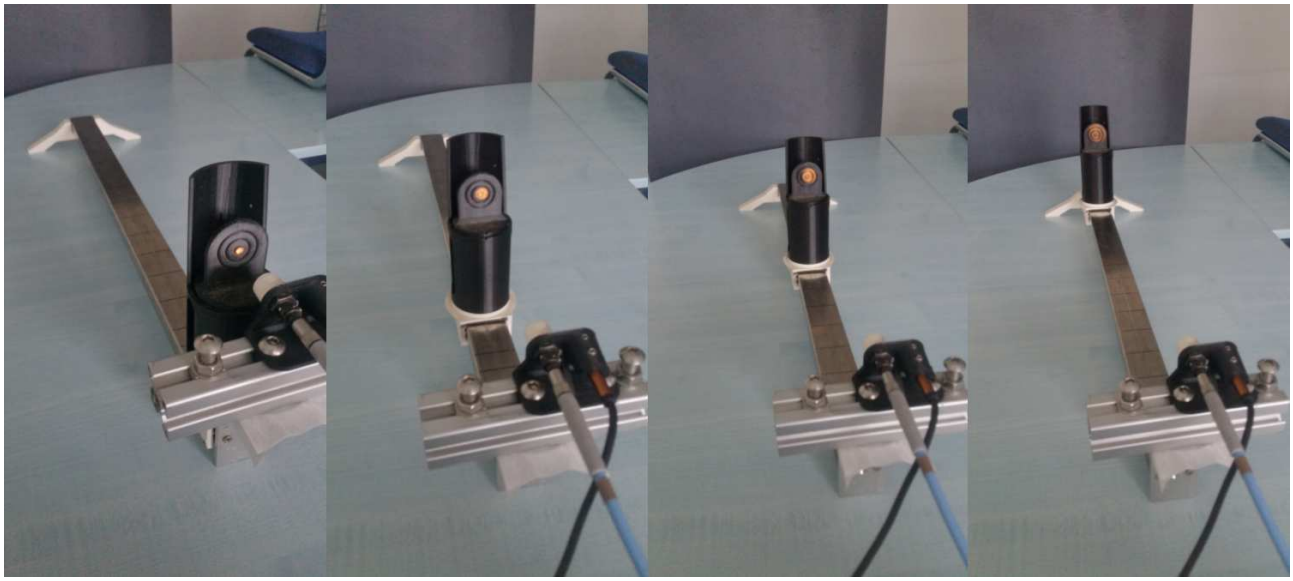


Figure 24: Vérification de l'alignement du capteur avec la plante mesurée pour 4 distances (2 cm, 16 cm, 32 cm et 56 cm).

2.1.2. Étalonnage des dispositifs

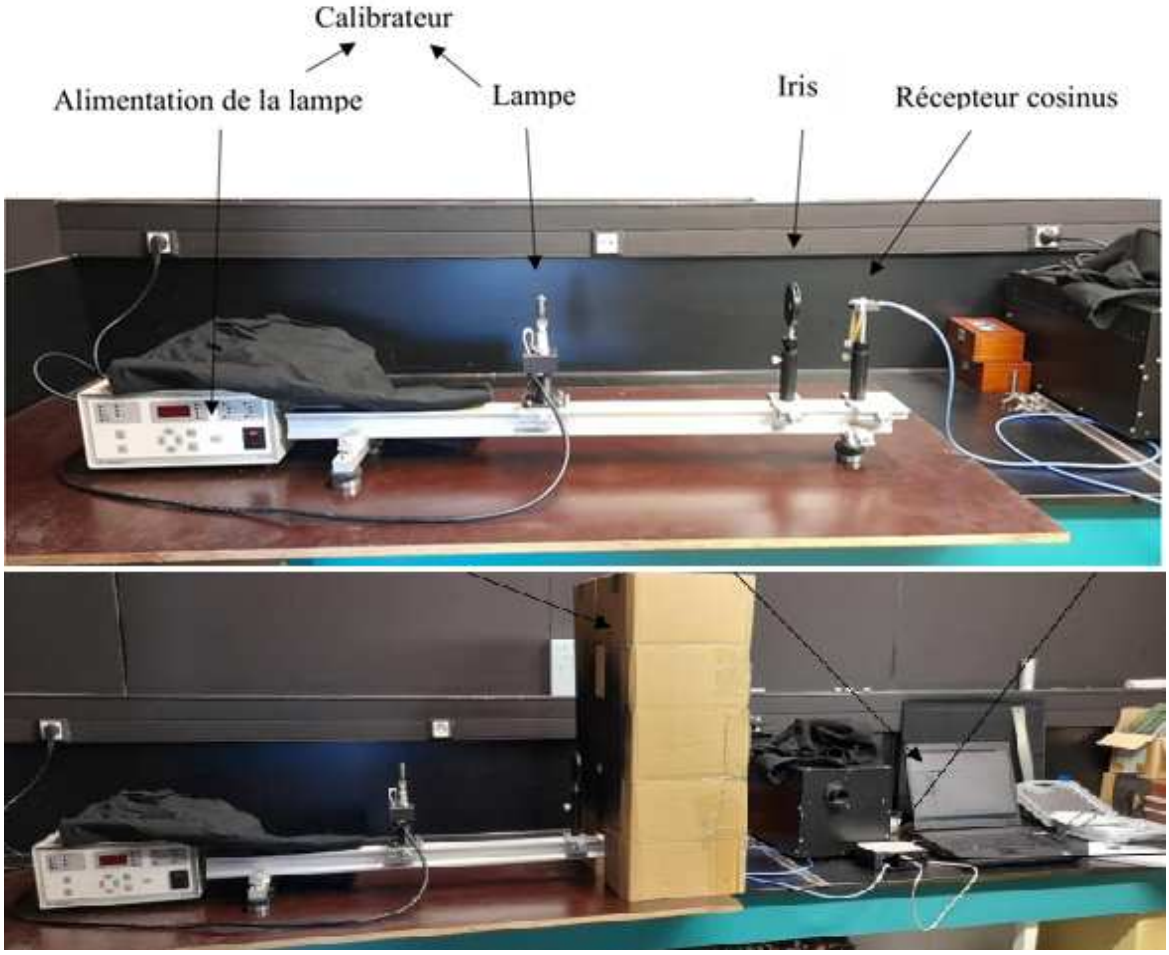


Figure 25: Dispositif utilisé pour réaliser l'étalonnage du capteur directionnel.

Afin de s'assurer de la précision des mesures effectuées avec les différents spectroradiomètres, un étalonnage est réalisé avant leur utilisation.

L'étalonnage du banc spectral est réalisé à plusieurs niveaux comprenant une vérification de l'angle d'ouverture à 3° et un étalonnage du spectroradiomètre.

L'étalonnage du spectroradiomètre est réalisé à l'aide d'un calibrateur (Ampoule de 1000 W, halogène à quartz et filament de tungstène) et d'un récepteur cosinus. Un calibrateur est une source lumineuse dont le spectre d'éclairement est connu. Un récepteur cosinus est un capteur qui permet de mesurer l'éclairement dans un espace donné. La calibration est faite dans une chambre noire afin de réduire les perturbations liées aux appareils de mesures (écran d'ordinateur) (Fig. 25). Des mesures de rayonnement sont réalisées à différentes distances du capteur.

Les distances choisies pour effectuer les mesures sont 30, 40, 50, 60 et 70 cm. Une série de 10 mesures pour chaque distance est faite afin de vérifier la répétabilité des mesures. Ces mesures permettent de calculer les coefficients d'étalonnage en suivant l'équation suivante :

$$Cp = \frac{Tr \times d\,L\,p \times A \times Ip}{(Sp - Dp)}$$

Où :

- Cp correspond au coefficient d'étalonnage (W.s.m-²)
- Ip correspond aux la valeurs étalonnées (W.m-².nm)
- Sp correspond aux valeurs brutes
- Dp correspondent aux valeurs brutes dans le noir complet
- Tr correspond au temps d'intégration (s)
- dLp correspond au nombre de nanomètres pour un pixel donné (nm)
- A correspond à l'air du capteur incident (cm²)

Les coefficients d'étalonnage obtenus pour chaque longueur d'onde sont présentés figure 26.

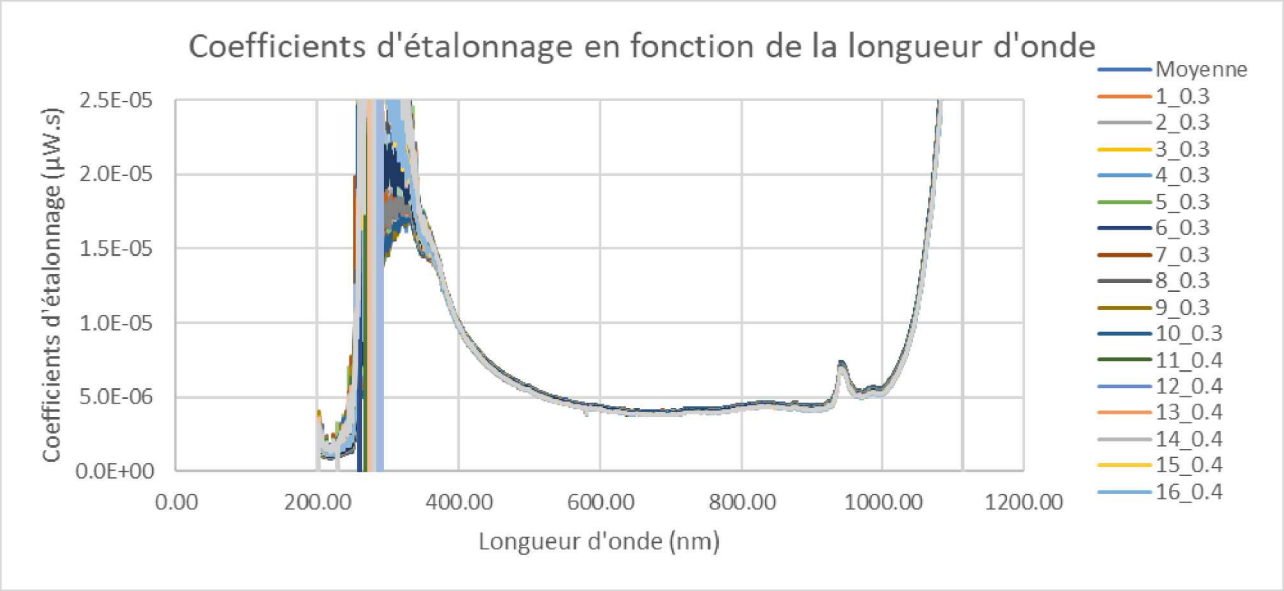


Figure 26: Coefficients d'étalonnage du spectroradiomètre

2.2. Modélisation d'un capteur directionnel

2.2.1. Création d'un capteur basé sur les surfaces végétales présentent dans le champ de vision du capteur

Une première approche simplifiée de capteur directionnel a été réalisée sans utiliser sans les calculs de rayonnement. Ce capteur est construit à l'aide des outils d'OpenAlea, une caméra ayant un champ de vision de 3° est positionnée dans la scène afin de reproduire le capteur utilisé lors des mesures. Un cache est intégré à la scène afin de restreindre la zone capturée par la caméra dans un disque, la vue de la caméra est visualisée dans un écran de 1000 × 1000 pixels. La surface de plante capturée (SVC) par la caméra est comptée en utilisant séparant les pixels par couleur. Dans nos simulations, les organes de plantes sont arbitrairement colorés en vert dans les options d'import d'OpenAlea. La SVC est calculée en pourcentage de pixels attribuées aux plantes par rapport à la surface totale de calcul, en excluant les pixels correspondant au cache.

Cette approche de simulation ne permet cependant pas de simuler le ζ car aucun calcul de rayonnement n'est effectué. Les résultats de simulations sont donc comparés aux mesures afin de vérifier s'il existe une corrélation entre le ζ et la SVC.

2.2.2. Création d'un capteur directionnel dans CARIBU

Paramètres du capteur

Un capteur directionnel est construit dans CARIBU car les outils de CARIBU ne permettent pas de restreindre la lumière reçue par une surface pour un FOV donné. Ce capteur est composé d'une cellule réceptrice recevant la lumière de l'environnement. Afin de restreindre le champ de vision de cette cellule réceptrice à 3°, un collimateur est positionné dans la scène sous la forme d'un cylindre (Fig. 27). Le FOV du capteur ainsi que son diamètre sont des paramètres d'entrées du modèle, le diamètre est fixé à 15 mm afin de reproduire le capteur du spectroradiomètre. Pour que le collimateur ne perturbe pas les interactions lumineuses dans la scène, ses propriétés optiques sont définies comme étant nulles, c'est-à-dire qu'il ne réfléchit et ne transmet aucune lumière.

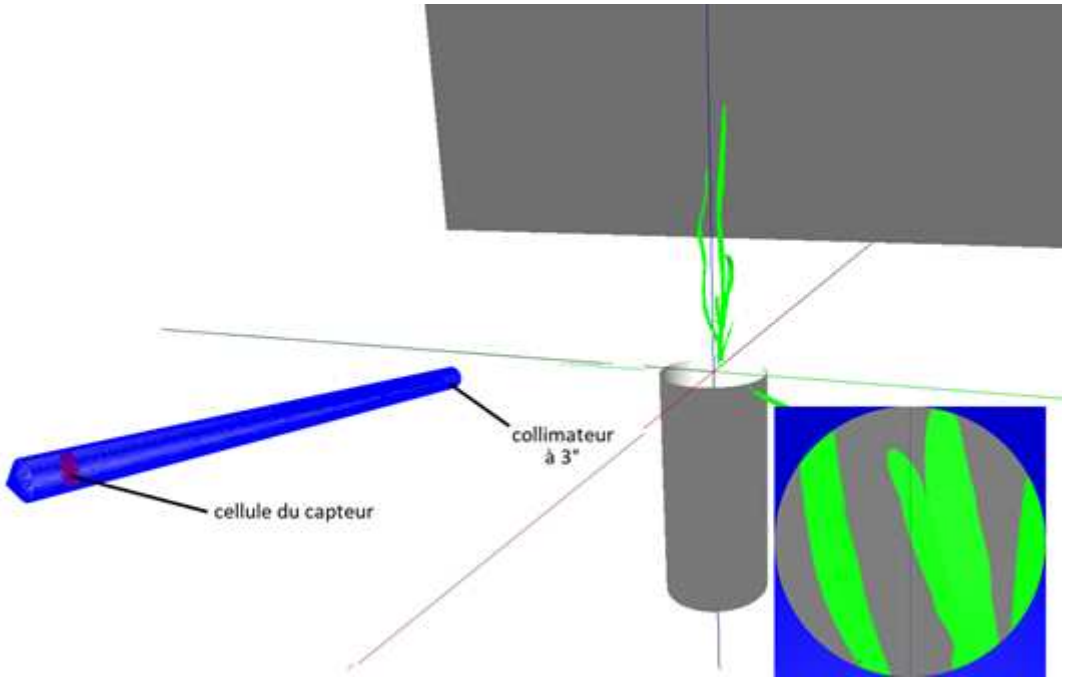


Figure 27: Représentation du capteur directionnelle dans une scène PlantGL

Le calcul du rayonnement est réalisé avec rediffusion, la taille de l'écran est fixée à 12800 × 12800 pixels.

Paramètre des scènes de CARIBU

Le scan d'une plante au format obj est chargé dans la scène avec des propriétés optiques similaires à celles mesurées sur des feuilles de luzernes. La réflectance et la transmittance des organes est sont fixées à 0.15 pour R et à 0.45 pour FR (Combes et al. 2000).

La position du capteur en mm par rapport à la plante fait partie des paramètres d'entrée du modèle de capteur directionnel. Ces paramètres permettent d'avoir pour chaque simulation du ζ une position du capteur reproduisant celle du spectroradiomètre lors de la mesure. Le capteur est positionné en face de la plante (y = 0 mm), à une hauteur z de 25 mm au-dessus du pot de la plante mesurée et à une distance x variable selon la mesure (xmin = 20 mm, xmax = 560 mm).

La nomenclature des noms des scans permet d'obtenir la date et l'identifiant de la plante scannée, à partir de ces informations l'heure exacte de la mesure peut être récupérée dans les acquisitions effectuées avec le banc spectral. La date et l'heure sont ainsi obtenues pour chaque simulation et permet d'obtenir la position du soleil à l'aide des coordonnées géographique de la mesure (Lusignan : latitude 46.4036827, longitude 0.0787160). Les simulations sont effectuées avec les modèles de ciel SOC et UOC.

Afin de reproduire le plus fidèlement les conditions de mesures, le pot ainsi que le support de pot ont été reproduits dans la scène pour les simulations faites avec les scans d'ArchiMono. Le pot et la plaque positionnée à l'arrière du banc spectral ont été reproduits dans la scène pour les simulations avec les scans d'ArchiDuo. Pour chacun de ces éléments, les propriétés optiques définies sont celles obtenues par la mesure faite par un calibrateur, avec 0.1 en réflectance et en transmittance pour R et pour FR pour la plaque et le pot.

Les calculs effectués par CARIBU pour chaque scène sont relativement longs. Afin de réduire le temps global de simulation, une parallélisation des simulations est effectuée à l'aide de la librairie python « multiprocessing ».

Traitement des sorties du capteur directionnel

Les mesures ont été réalisées sous serre. Afin de prendre en compte l'atténuation de rayonnement provoquée par celle-ci, le ζ simulé par CARIBU avec le capteur directionnel est recalculée en post-traitement à partir des données de mesure et des données récoltées par SICLEX.

Le flux de photons de chaque longueur d'onde en direct et en diffus est calculé selon l'équation suivante :

$$R_\lambda n = \tau PARn \times \alpha_{R_\lambda} \times PARn \times R_\lambda n\, caribu$$

Où :

- λ correspond à Rc ou Rs
- n correspond aux conditions de rayonnement direct ou diffus
- τ correspond au pourcentage de rayonnement transmis à l'intérieur de la serre en comparaison avec le rayonnement extérieur. Cette valeur est fixée à 1 pour le direct. Dans le cas du diffus, la valeur de τ a été évaluée selon 2 méthodes : avec une estimation réalisée à partir de photos hémisphériques et en comparant le flux incident mesuré dans la serre au flux incident mesuré à l'extérieur de la serre. Les deux méthodes ont permis d'obtenir une valeur de τ de 0,56.
- α correspond au pourcentage d'énergie correspondant aux longueurs d'ondes choisies par rapport au PAR, repris de précédentes mesures. Dans nos travaux, on postule suppose que la valeur du paramètre α est le même quel que soit le type de ciel.
- PARn correspond aux mesures de PAR en extérieur à la même date et à la même heure que la simulation

Le ζ est ensuite recalculé de la manière suivante :

$$zeta = \frac{Rc_{direct} + Rc_{diffus}}{Rs_{direct} + Rs_{diffus}}$$

2.2.3. Création d'un capteur directionnel dans Malia

La simulation du ζ avec le modèle MALIA suis le pipeline présenté en figure 28.



Figure 28: Étapes de calculs du ζ avec le logiciel MALIA

Paramètres du capteur

MALIA est un modèle basé sur les lancers de rayon qui calcule le spectre d'un ensemble de pixels présents dans le FOV de la caméra. Le capteur directionnel correspond donc au champ de vision de la caméra que l'on restreint à 3°.

Les propriétés de la caméra sont stockées dans un fichier mcf, un format spécialement conçu pour MALIA et codé en XML. La caméra est positionnée à la même hauteur et à la même distance que lors des mesures d'ArchiDuo, le champ de vision est fixé à 3° et la résolution de l'écran est de 512 × 512.

Le spectromètre maya mesure l'environnement lumineux dans un disque tandis que les sorties de MALIA sont prises dans un carré. Une suppression des pixels présents dans les coins de l'image spectrale est effectuée afin de reproduire le capteur du spectromètre. Le positionnement du capteur est basé sur des photos réalisées durant l'expérimentation ArchiDuo.

Paramètres des scènes MALIA

Afin de simuler une mesure directionnelle du ζ, une scène doit être construite. La scène construite dans MALIA est constituée des mêmes éléments que la scène utilisée dans CARIBU : elle comprend le scan d'une plante au format obj ainsi que le pot et la plaque positionnée à l'arrière de la plante. Les scènes sont au format msf, qui correspond à un fichier xml contenant l'emplacement des objets à importer ainsi que leurs propriétés optiques respectives.

Le ciel est simulé grâce au modèle de Preetham (ciel clair). La position du soleil est calculée avec les coordonnées, la date et l'heure des mesures.

MALIA est capable de réaliser une image spectrale pour l'ensemble du spectre entre 300 nm et 1000 nm, cependant chaque longueur d'onde demande un temps de calcul supplémentaire. Pour cette raison, les simulations avec MALIA sont restreintes aux longueurs d'onde 660 nm et 730 nm.

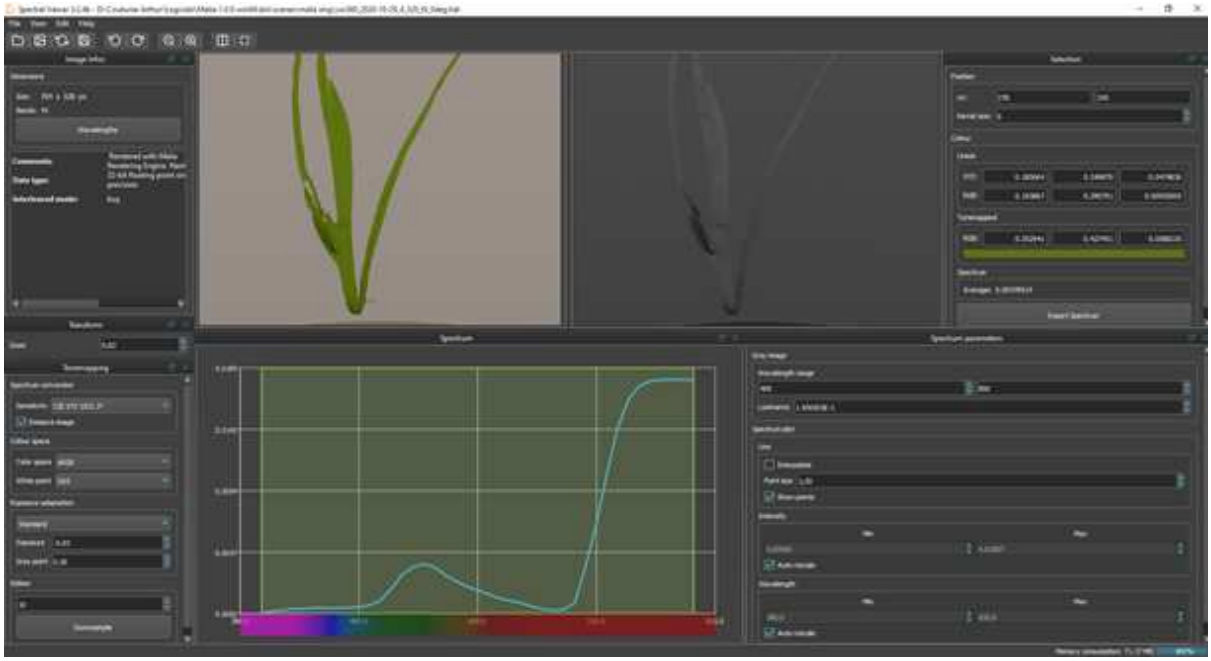


Figure 29: Visualisation de l'image spectrale dans le logiciel SpectralViewer

A partir des images spectrales (Fig. 29), le ζ est calculé à partir du rapport du R total et du FR total de l'image. Aucun recalcul des mesures est nécessaire, car le spectre de l'environnement mesuré par le spectromètre hémisphérique Jaz est utilisé dans les entrées du modèle pour la création du spectre du ciel.

3. Acquisition de l’architecture des plantes

3.1. Scanner laser

Afin d'obtenir l’architecture 3D des plantes mesurées, un scanner laser Konica-minolta Vivid 910 relié à un ordinateur équipé du logiciel Rapidform 2006 est utilisé (Fig.30). Le rayon laser converti en plan laser par l’intermédiaire d’un miroir balaye verticalement la scène et traduit la géométrie de l’objet scanné en un nuage de points en calculant le temps que le laser met pour arriver sur le capteur du scanner (Fig. 31).

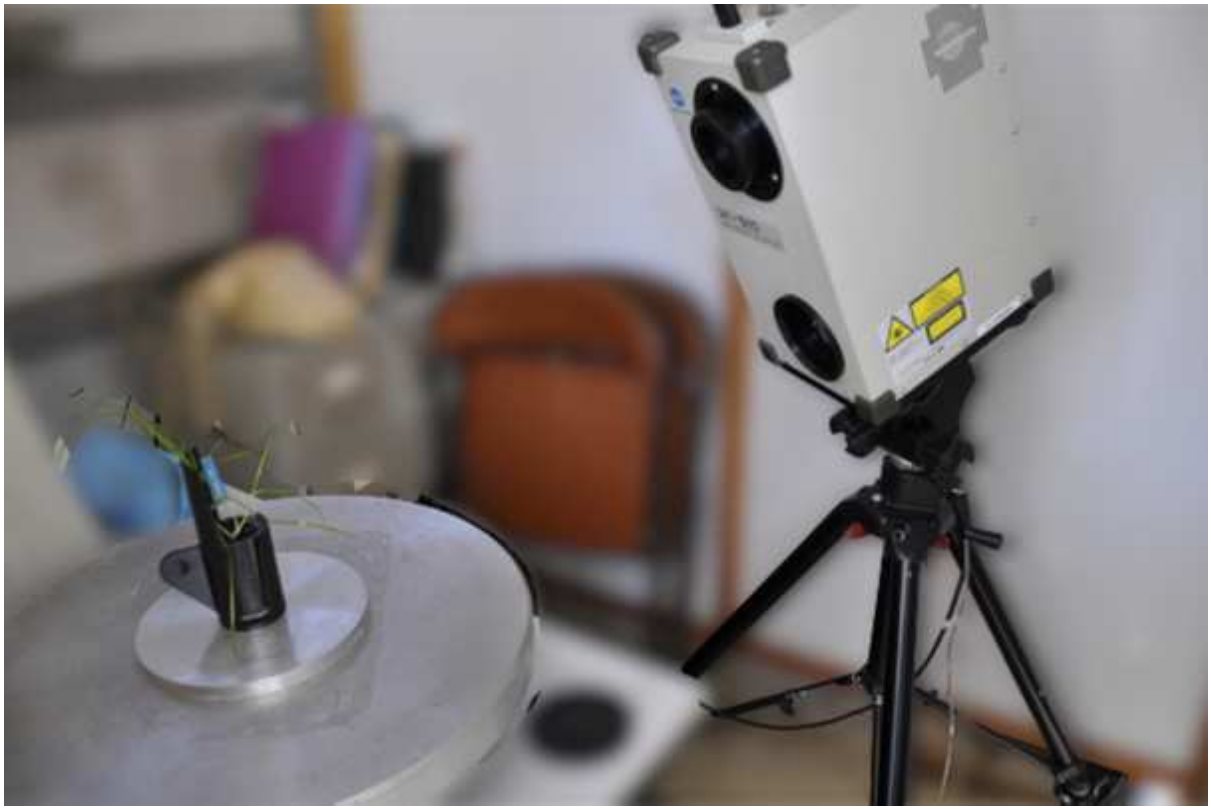


Figure 30: Scan d'une plante à l'aide du scanner laser

Afin d’assurer de bonnes conditions d’éclairement, les acquisitions sont réalisées dans des salles noires sur le site d’expérimentation. Trois objectifs sont adaptables sur le scanner, permettant de régler son FOV. Cependant un agrandissement du FOV réduit la résolution du scan car un pixel couvre une surface réelle plus large. L’objectif 25 mm a été utilisé pour les scans de plantes isolées.

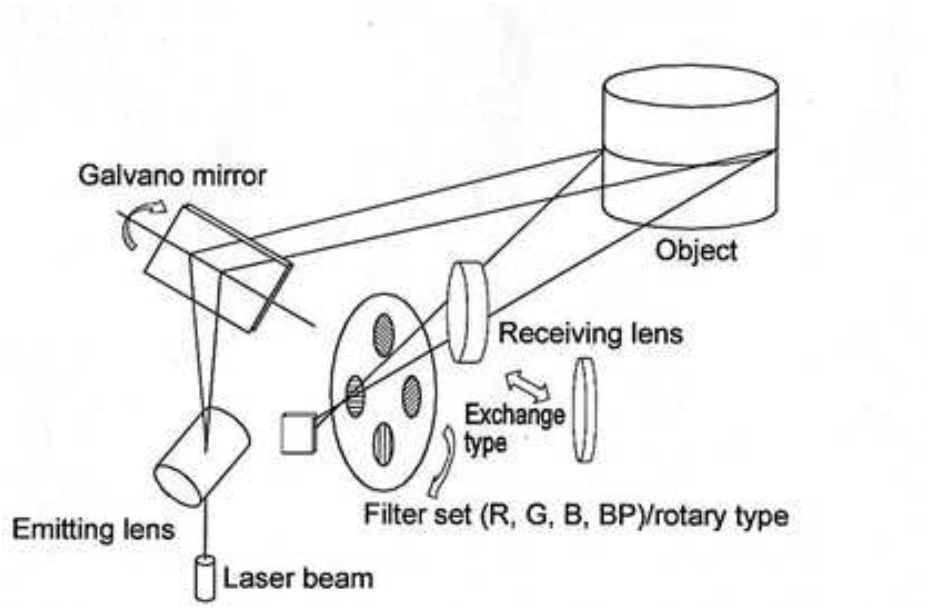


Figure 31: Fonctionnement du scanner laser. Le laser est envoyé sur une lentille pour obtenir un faisceaux redirigé vers l'objet à modéliser à l'aide d'un miroir. Le laser renvoyé par l'objet est reçu par une seconde lentille permettant de calculer la distance entre l'appareil et l'objet.



Figure 32: Table rotative et ordinateur permettant de faire les acquisitions 3D.

Lors des expérimentations, l'orientation des pots est conservée entre la mesure faite par le spectromètre et l'acquisition 3D. L'angle de vue du scanner est différent de celui du spectromètre, cet angle de vue est mesuré et pris en compte lors de l'utilisation des scans. Une reconstruction est opérée sur les nuages de points ainsi obtenus.

3.2. Reconstruction des surfaces

L'acquisition de l'architecture par le scanner laser se fait sous la forme d'un nuage de points. Afin d'exploiter ces données une étape de reconstruction des faces doit être réalisée afin de disposer d'un objet 3D compatible avec les modèles de lumières. Deux méthodes de reconstruction ont été utilisées.

La méthode Ball Pivoting, reconstruit les faces de l'objet 3D en reliant les points situés dans un rayon déterminé. Cette méthode permet de reconstruire des faces pour l'ensemble des points qui composent le nuage de point. Chaque triangle est construit indépendamment des autres, sans lien avec les autres. La reconstruction des surfaces par la méthode Ball Pivoting est réalisée sous python à l'aide de la fonction « surface reconstruction ball pivoting » de la librairie « pymeshlab », le rayon de clustering est fixé à 10 % et le seuil d'angle est fixé à 150°.

La méthode Marching Cubes, se base sur une projection des points de la scène sur un plan et construit les faces en reliant les points de ce plan. Cette méthode permet une cohérence dans la position des faces mais élargit la surface de l'objet reconstruit par rapport au nuage de point d'origine. La reconstruction des surfaces par la méthode Marching Cubes est réalisée sous python à l'aide de la fonction « marching cubes apss » de la librairie « pymeshlab ». L'échelle de filtre est fixée à 3, le nombre d'itération de projection à 1 et la résolution à 500. Afin de corriger les erreurs potentielles de reconstruction, une suppression des surfaces aberrantes est effectuée en comparant la position des points du nuage de point d'origine avec la position des points de la reconstruction surfacique, les points trop éloignés du scan sont ainsi identifiés et supprimés. Une seconde étape de correction est réalisée afin de réduire la largeur excédentaire liée à l'utilisation de la méthode.

Le script de reconstruction des surfaces utilisé pour la méthode Ball Pivoting et la méthode Marching Cubes est sont disponibles en annexe.

4. Expérimentations

4.1. ArchiMono

Une première campagne de mesure de l'architecture des plantes et de l'environnement lumineux, baptisée ArchiMono, a été conduite du 14/10/2019 au 12/11/2019.

Le matériel végétal est composé de 4 groupes de ray-grass âgés de 10 à 50 jours durant les 5 campagnes de mesures, un groupe correspondant à des plants de ray-grass repiqués à la même date. Les 3 premiers groupes sont repiqués à 7 jours d'intervalle et le 4e groupe à 16 jours d'intervalle après le repiquage du groupe 3. Les plantes sont repiquées dans des pots de PVC de 75 mm de hauteur et d'un diamètre de 50 mm (Fig. 33).

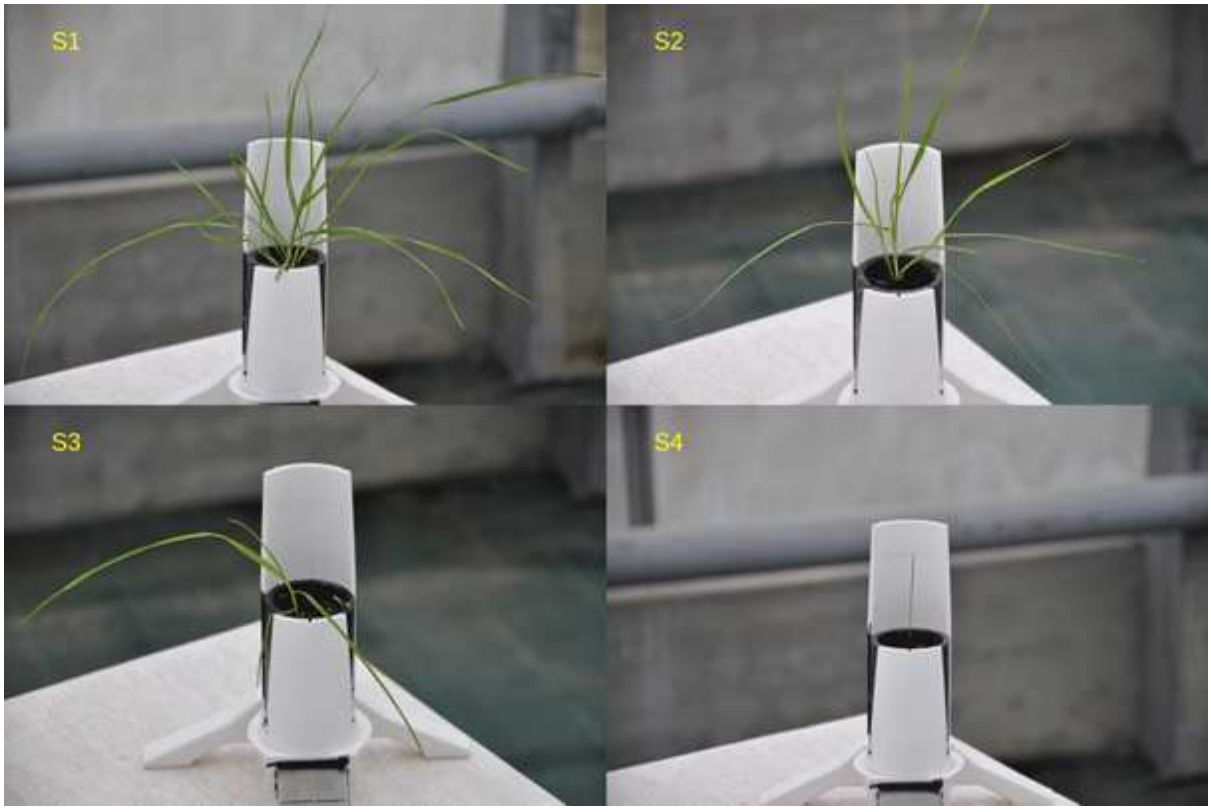


Figure 33: Architecture d'une plantes de chaque groupe de semis S1 à S4 à une même date de mesure.

Une campagne de mesure consiste d'une part à faire des mesures du ζ correspondant au rayonnement renvoyé par la plante et d'autre part à l'acquisition de son architecture 3D. Les mesures de ζ sont effectuées à l'aide du banc spectral pour 25 distances, allant de 20 mm à 560 mm entre le capteur et la base du pot. L'orientation de chaque pot pour les mesures de ζ est conservée entre les différentes campagnes de mesures à l'aide d'un repère mis sur chaque pot. Une mesure

supplémentaire de l'environnement lumineux autour du capteur est réalisée à l'aide d'un capteur hémisphérique.

L'acquisition tridimensionnelle des plantes est réalisée à l'aide du scanner laser. L'orientation du pot pour l'acquisition 3D est la même que celle des mesures de ζ mais pour une seule distance et un angle de vue différent de celui du capteur FOV.

5 campagnes de mesures hebdomadaires ont eu lieu entre le 15/10/2019 et le 13/11/2019. Afin d'avoir les conditions de ciel similaires entre chaque campagne de mesure, l'angle zénithal et l'angle azimutal ont été choisis pour qu'ils soient le plus proches possible d'une campagne de mesure à l'autre. L'angle zénithal a été choisi à partir de l'angle maximal pour le jour de mesure où le soleil est au plus bas (30°) et l'angle azimutal du soleil fixé à 0°. À chaque mesure, l'orientation du banc spectral est vérifiée à l'aide d'une boussole. Les heures de mesures sont sélectionnées pour correspondre à l'angle zénithal et le banc spectral positionné pour que le capteur soit orienté vers le soleil.

4.2. ArchiDuo

Une seconde campagne de mesure sous serre, baptisée ArchiDuo, a été réalisée du 02/10/2020 au 13/11/2020.

Des plants de ray-grass sont isolés par paires pour 15 distances différentes allant de 20 mm à 560 mm, chaque distance ayant 3 réplicas (Fig. 34). Les plantes sont repiquées dans les mêmes pots que pour ArchiMono. Une condition de témoin isolé est réalisée pour chaque distance, où la plante est positionnée face à un pot vide. Le détail des distances entre les plantes est le suivant (en mm): 20, 40, 60, 80, 100, 120, 160, 200, 240, 280, 320, 360, 400, 480, 560.

Les différentes conditions de distance sont séparées les unes des autres à l'aide de plaques de bois enduites de peinture anthracite réduisant la réflexion de lumière et ayant un ζ neutre.

Les mesures nécessitent un déplacement des échantillons. Afin de conserver les conditions de croissances entre chaque campagne de mesure, la distance entre chaque paire de plantes est assurée à l'aide de plaques de styrodur dans lequel sont fixés les pots. Un marquage de la position du pot par rapport au styrodur est réalisé pour que le pot ait la même orientation après chaque mesure du banc spectral.



Figure 34: Dispositif ArchiDuo. Les plantes sont positionnées face à face et irriguées à l'aide de capillaires. Des plaques peintes couleur anthracite séparent les différentes condition.

L'irrigation est réalisée à l'aide de gouttes à goutte allant de part et d'autre des bacs pour atteindre les pots. Le système d'irrigation est amovible pour que les pots puissent être retirés facilement de leur emplacement lors des mesures. Des coupelles sont placées sous les pots afin de récupérer l'eau excédentaire.

4.3. Cinétique journalière du ζ

La cinétique journalière du ζ a été mesurée pour des conditions de ciel clair et de ciel couvert, pour des plantes au stade de développement « 1 talle » et « 10 talles » et pour des plantes isolées ou en peuplement de 6 rangs et 3 lignes pour des distances entre plantes de 5 cm (Fig. 35). Les mesures avec plantes isolées n'ont été réalisées que pour des conditions de ciel clair.

Pour chaque condition, la mesure du ζ directionnel a été effectuée avec le dispositif du banc spectral à une distance de 10 cm entre le capteur et le premier rang à un intervalle de mesure compris entre 5 et 10 minutes.



Figure 35: Disposition des plantes pour la mesure de l'évolution du ζ au cours de la journée

4.4. Mesure de peuplement artificiels

Une mesure du ζ calculé à partir du rayonnement renvoyé par les plantes a été réalisée pour différentes conditions de peuplement. A partir d'une plante cible, le ζ a été mesuré à plusieurs reprises en ajoutant entre chaque mesure une rangée de plantes supplémentaires, allant jusqu'à 6 rangs (Fig. 36). Ce protocole a été reproduit pour une seule ligne ou pour 3 lignes et pour 2 densités correspondant à des distances entre plantes de 5 cm (HD) et de 10 cm (LD).



Figure 36: Mesure de l'effet du voisinage

4.5. Traitement des spectres mesurés

Afin d'avoir accès à l'environnement lumineux global lors des mesures avec les dispositifs basés sur des capteurs directionnels, des mesures hémisphériques de l'environnement lumineux incident ont été effectuées. Ces mesures sont réalisées avec un spectroradiomètre Jaz relié à un capteur cosinus permettant une mesure à 360°.

À partir des spectres mesurés par les différents dispositifs, le ζ est calculé à partir du rapport entre les flux de photon de 655 nm à 665 nm et de 725 nm à 735 nm, qui correspondent aux pics d'absorption des phytochromes.

Pour chaque expérimentation, trois jeux de données ont été acquis : Les données du spectroradiomètre directionnel maya ; Les données du spectroradiomètre hémisphérique jaz ; Les données spectrales extérieures mesurées en continu sur le dispositif « SICLEX » faisant des relevés météorologiques proches des parcelles d'expérimentation en permanence au cours du temps (Perrot et al. 2022).

Ces mesures sont utilisées comme jeu de données de comparaison avec des sorties de modélisation de ζ, afin d'évaluer la capacité de modèles de rayonnement à modéliser le ζ.

5. Évaluation de la sensibilité de la morphogenèse au ζ simulé à partir de deux modèles de rayonnement

5.1. Approche

Différentes méthodes de simulation de l'environnement lumineux existent. Cette variété de modèle pose la question de l'impact du choix de l'un de ces modèles sur la modélisation du ζ et de ses conséquences sur la modélisation de la morphogenèse. Dans ce but, le TrPPFD a été estimé en utilisant deux modèles de transfert radiatif : RIRI, un modèle turbid-medium et CANESTRA, un modèle projectif.

L'estimation dynamique du ζ a été réalisée à l'aide de trois méthodes de calcul différentes avec ou sans relation fonctionnelle en utilisant les valeurs de TrPPFD fournies par ces modèles radiatifs. La sensibilité de la morphogenèse à la modélisation de ζ dans un modèle individu-centré de légumineuses a été évaluée.

Le modèle L-egume, développé par Louarn & Faverjon (2018), est capable de simuler la morphogenèse des plantes et la dynamique des populations chez les légumineuses fourragères. Les variations de morphogenèse ont été évaluées pour les pétioles et les entre-nœuds, deux organes importants pour l'accès à la ressource lumineuse chez les légumineuses et sensible au ζ (Robin et al. 1994; Gautier et al. 1997; Baldissera et al. 2014), à différents stades de développement et de densité végétale. Ces évaluations ont été réalisées pour deux espèces de légumineuses architecturalement contrastées : la luzerne et le trèfle blanc.

5.2. Cadre d'utilisation des modèles de transfert radiatif pour le calcul dynamique du ζ

Le modèle de transfert radiatif CANESTRA appuie ses calculs sur l'architecture végétale. Il est donc considéré comme plus précis pour simuler le ζ que RIRI, un modèle Turbid-medium utilisant uniquement l'orientation et la surface des feuilles comme composantes de l'architecture. En outre, CANESTRA a été évalué précédemment pour un large échantillon de conditions de ciel et de mesures (capteur hémisphérique à orientation horizontale) (Chelle et al. 2007). Pour ces raisons, le modèle CANESTRA est considéré comme modèle de référence.

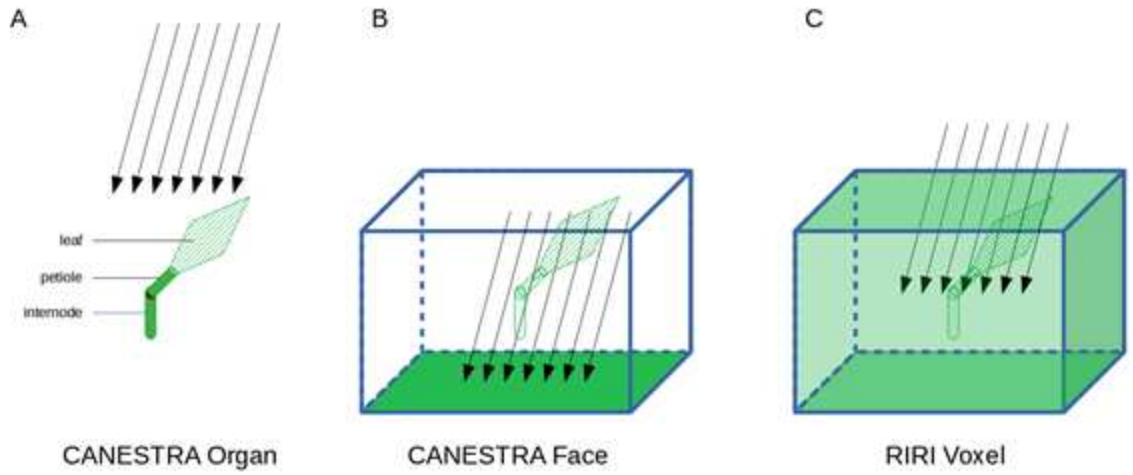


Figure 37: Représentation visuelle des trois modèles de transfert radiatif. Les zones rayées des plantes ne sont pas utilisées pour leur calcul radiatif respectif et les zones unies correspondent à celles utilisées pour le calcul de ζ. Le ζ est calculé à partir : A) du rayonnement intercepté par les triangles des organes, B) du rayonnement intercepté par la face inférieure du voxel où est localisé l'organe, C) du rayonnement intercepté par le voxel où est localisé l'organe et suivant une loi de Beer-Lambert basée sur la distribution uniforme de la surface foliaire dans ce voxel.

Trois méthodes de calcul ont été testées (Fig. 37). Deux ont été construites à partir de CANESTRA et une à partir du modèle RIRI.

Le modèle CANESTRA Organ utilise les sorties du modèle CANESTRA, calculées à partir de l'irradiance de la surface de l'organe considéré.

Le modèle CANESTRA Face utilise les sorties d'une surface horizontale des voxels du modèle turbid-medium. Pour ce modèle, la lumière cumulée interceptée par les organes est estimée par la surface horizontale inférieure du voxel. Dans le cas où l'organe est distribué en deux voxels contigus, les calculs ont été effectués en lien avec les sites de perception en utilisant le voxel positionné à la base de l'organe pour les entre-nœuds et en utilisant le voxel positionné à la pointe de l'organe pour les pétioles (Gautier *et al.* 2000; Faverjon 2018).

Le modèle RIRI Voxel utilise les sorties du modèle RIRI.

Pour ces trois méthodes, dans les simulations sans rediffusion de la lumière, le ζ est calculé à partir de la fraction quotidienne du PPFD transmis à l'aide des relations fonctionnelles. Ainsi, la combinaison des trois méthodes avec la prise en compte ou non des rediffusions pour le modèle CANESTRA aboutit à un total de 5 modalités de simulation (Fig. 38).

Pour les simulations prenant en compte la rediffusion, seules les méthodes basées sur CANESTRA sont utilisées car, dans ce travail , RIRI ne considère pas la diffusion multiple dans la canopée. Dans la deuxième approche, les valeurs de ζ sont calculées en couplant la méthode projective avec la méthode de nested radiosity.

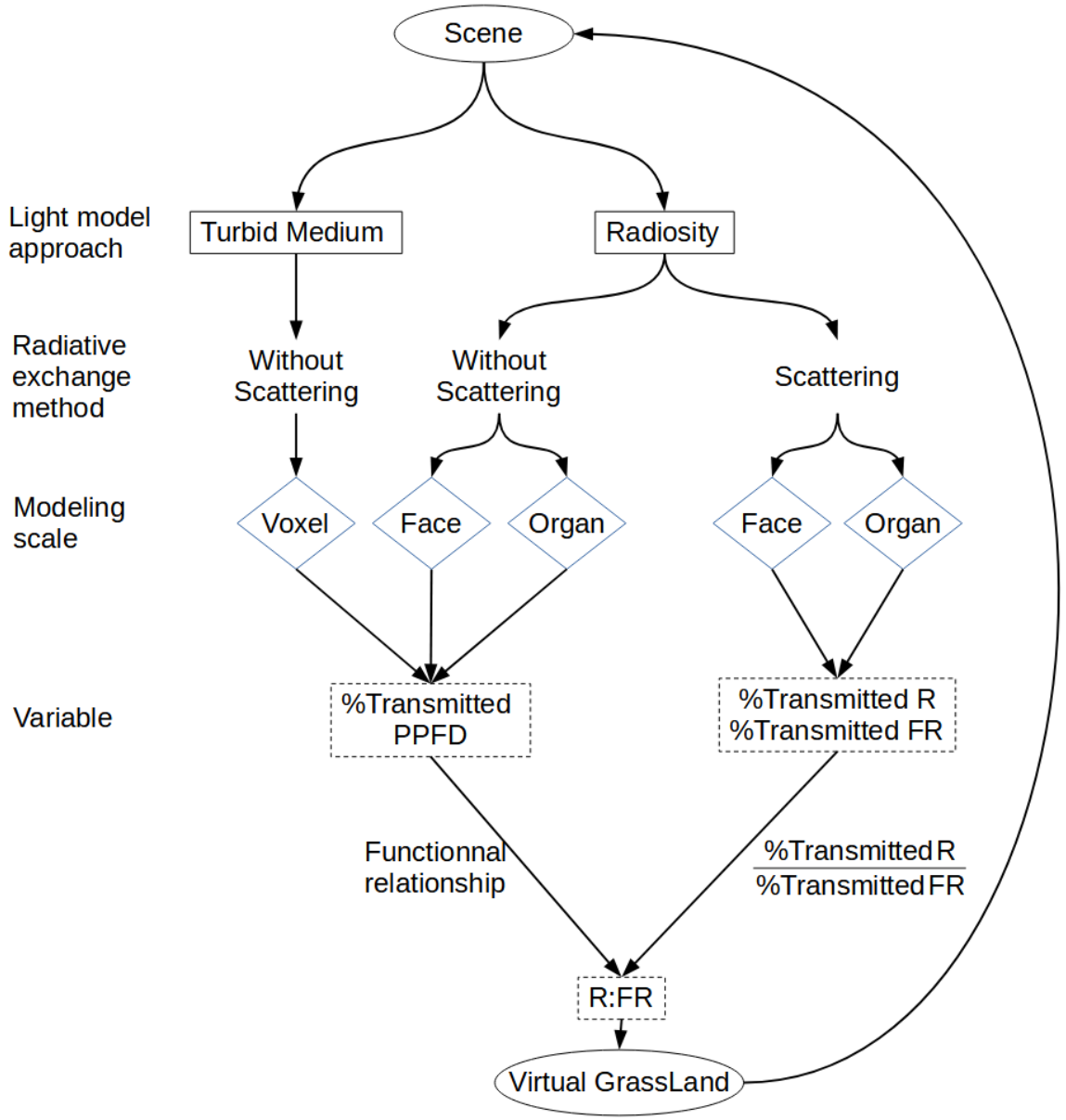


Figure 38: Diagramme du calcul de ζ pour chacun des modèles de transfert radiatif et leur intégration dans Virtual GrassLand.

5.3. Modèle de plante individuel individu-centré: Virtual GrassLand.

L'effet des différents modèles radiatifs pour estimer le ζ a été évalué avec le modèle de légumineuses Virtual GrassLand (VGL). Ce modèle a été décrit en détail dans (Louarn and Faverjon 2018) et son code est disponible en opensource sur https://github.com/OpenAlea-incubator/l-egume.

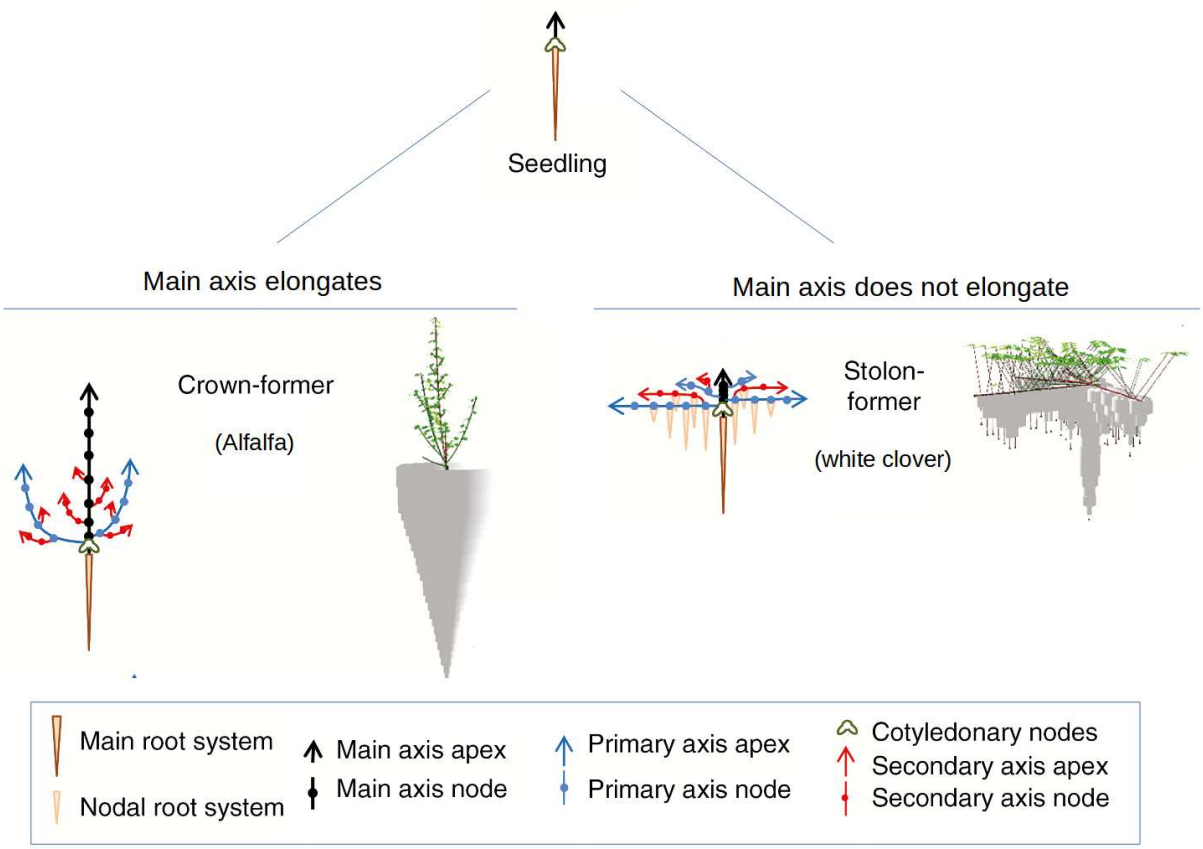


Figure 39: Représentation schématique des groupes morphogénétiques modélisés par Virtual GrassLand.

Le modèle VGL simule le fonctionnement de plantes individuelles sur la base du formalisme des L-system (Prusinkiewicz and Lindenmayer 1990). Dans VGL, ce formalisme est couplé au modèle de transfert radiatif RIRI traitant l'interception de la lumière au-dessus du sol (Sinoquet et al. 2000) et à un modèle d'eau et d'azote minéral du sol (Louarn et al. 2016). Les interactions plante-plante sont représentées au travers de la compétition pour la lumière, l'eau et l'azote du sol. Ce modèle décrit le développement végétatif des organes aériens et souterrains pour les individus d'une population monospécifique ou de communautés multispécifiques de légumineuses à différentes

densités. Le modèle a été validé non seulement pour une large gamme de formes de croissance de légumineuses mais aussi pour des conditions pédoclimatiques contrastées (Fig. 39).

Dans le modèle VGL, le fonctionnement des méristèmes végétaux et des tissus en croissance est basé sur un pas de temps journalier pour simuler la morphogenèse potentielle des tiges et des racines, adapté respectivement de Faverjon (2018) et de Pagès et al. (2014). La plasticité des plantes et la régulation de leur croissance et de leur morphogenèse par l'environnement ont été prises en compte à travers deux boucles de rétroaction. La première rétroaction est l'effet de la quantité et de la qualité de la lumière sur la morphogenèse et la seconde est celle des ressources du sol disponibles pour les racines des plantes par rapport aux besoins en eau et en azote, afin d'échelonner la croissance réelle des plantes. Afin de tester l'effet de différents modèles de transfert radiatif sur la simulation du signal lumineux, les simulations sont réalisées dans des conditions où les interactions avec l'eau et l'azote du sol ne sont pas limitantes pour la croissance.

Dans ce modèle, la distribution de la qualité de la lumière déclenche des réponses photomorphogénétiques locales qui réduisent la production de phytomères au niveau des axes (Baldissera et al. 2014) et modulent la croissance des organes (Gautier et al. 2000). La réponse adaptative à la qualité de la lumière est considérée pour deux organes d'importance dans les réponses photomorphogénétiques des légumineuses : les entre-nœuds et les pétioles. Le site de perception du signal lumineux pour chacun de ces organes est l'ensemble de l'organe pour les entre-nœuds et la pointe de l'organe pour les pétioles (Robin et al. 1994). La fonction de la longueur des entre-nœuds et des pétioles dans une réponse dynamique à la qualité de la lumière est décrite comme suit dans le modèle :

$$longueur\,de\,croissance = Pot \times WTsurf \times NTsurf \times Photomorph$$

Où Pot est la longueur maximale non contrainte de l'organe (mm), WTsurf la réduction potentielle de la longueur en fonction de la surface des organes dépassant le seuil de stress hydrique (sans unité), NTsurf la réduction potentielle de la longueur en fonction de la surface des organes dépassant le seuil de stress azoté (sans unité) et Photomorph la modification potentielle de la longueur induite par l'effet combiné de la qualité (ζ) et de la quantité (PPFD) de lumière reçue à chaque pas de simulation (sans unité).

Les propriétés optiques des pétioles et des entre-nœuds utilisées sont les mêmes que celles des feuilles, les valeurs typiques des feuilles vertes pour la transmittance et la réflectance (0,15 pour R et 0,45 pour FR pour les 2 paramètres) publiées par Combes et al. (2000) sont utilisées.

5.4. Évaluation de la sensibilité des réponses morphogénétiques à la méthode de simulation du ζ

Les simulations sont réalisées avec une irrigation et une fertilisation en N optimales, les variations de longueur des organes ne peuvent donc être dues qu'aux variations de l'environnement lumineux lié au choix des modèles utilisés pour la simulation du ζ. Ainsi, la sensibilité (S) de la morphogenèse des plantes est évaluée en calculant dans quelle mesure les variations de ζ ont un impact sur la variation de longueur des organes par rapport à sa valeur moyenne, en utilisant l'équation suivante :

$$S = \frac{\Delta l}{\Delta \zeta} \times \frac{\bar{\zeta}}{\bar{L}}$$

Où Δζ est la différence de ζ calculé par deux modèles radiatifs différents, Δl est la différence de longueur d'organe obtenue entre ces deux modèles radiatifs, ces différences sont calculées aux mêmes dates et densité pour chaque organe. $\bar{\zeta}$ et $\bar{L}$ correspondent respectivement la longueur moyenne ζ et la longueur moyenne de l'organe des modèles considérés à partir de différentes plantes pour les mêmes dates et densité. La sensibilité est calculée relativement entre une méthode de modélisation et le modèle de référence CANESTRA Organ avec diffusion.

IV. Résultats

1. Introduction

L'utilisation de modèles pour simuler l'environnement lumineux reçu par les organes des plantes est un levier primordial dans la compréhension interactions plantes-rayonnement. Une évaluation de la sensibilité des modèles à la méthode de modélisation du ζ intercepté par les plantes a été menée.

La mesure des variations de l'environnement lumineux a également été mesurée en fonction de la distance, de l'heure de mesure et du nombre de plantes présentes dans le couvert.

Un modèle de capteur directionnel a été créé dans 2 modèles de lumière, une comparaison entre les mesures effectuées et les résultats obtenus par simulation a été est effectuée.

2. Sensibilité des modèles radiatifs à la simulation du ζ

Le modèle « CANESTRA Organ avec rediffusions » est considéré comme étant le modèle de référence pour l'évaluation de la sensibilité des modèles à la modélisation du ζ.

2.1. Effet des paramètres d'entrée

2.1.1. Effet de la taille des voxels

Deux des trois modèles de transfert radiatif utilisés sont basés sur l'interception radiative des voxels dans lesquels les organes ciblés sont présents. La construction des voxels dans la scène est déterminée par les coordonnées de la pointe de l'organe pour les pétioles et par les coordonnées de la base de l'organe pour les entre-nœuds. L'impact de la taille par défaut attribuée à ces voxels peut donc jouer un rôle sur le calcul du ζ.

Les courbes d'évolution du ζ en fonction du DOY (Day of Year, c'est-à-dire jour de l'année correspondant au numéro attribué) (Fig. 40) montrent que, pour l'ensemble des conditions de simulation, une diminution de la taille des voxels produit des valeurs de ζ simulées plus faibles. De plus, le ζ simulé pour les entre-nœuds présente de plus grandes variations de ζ que le ζ simulé pour les pétioles.

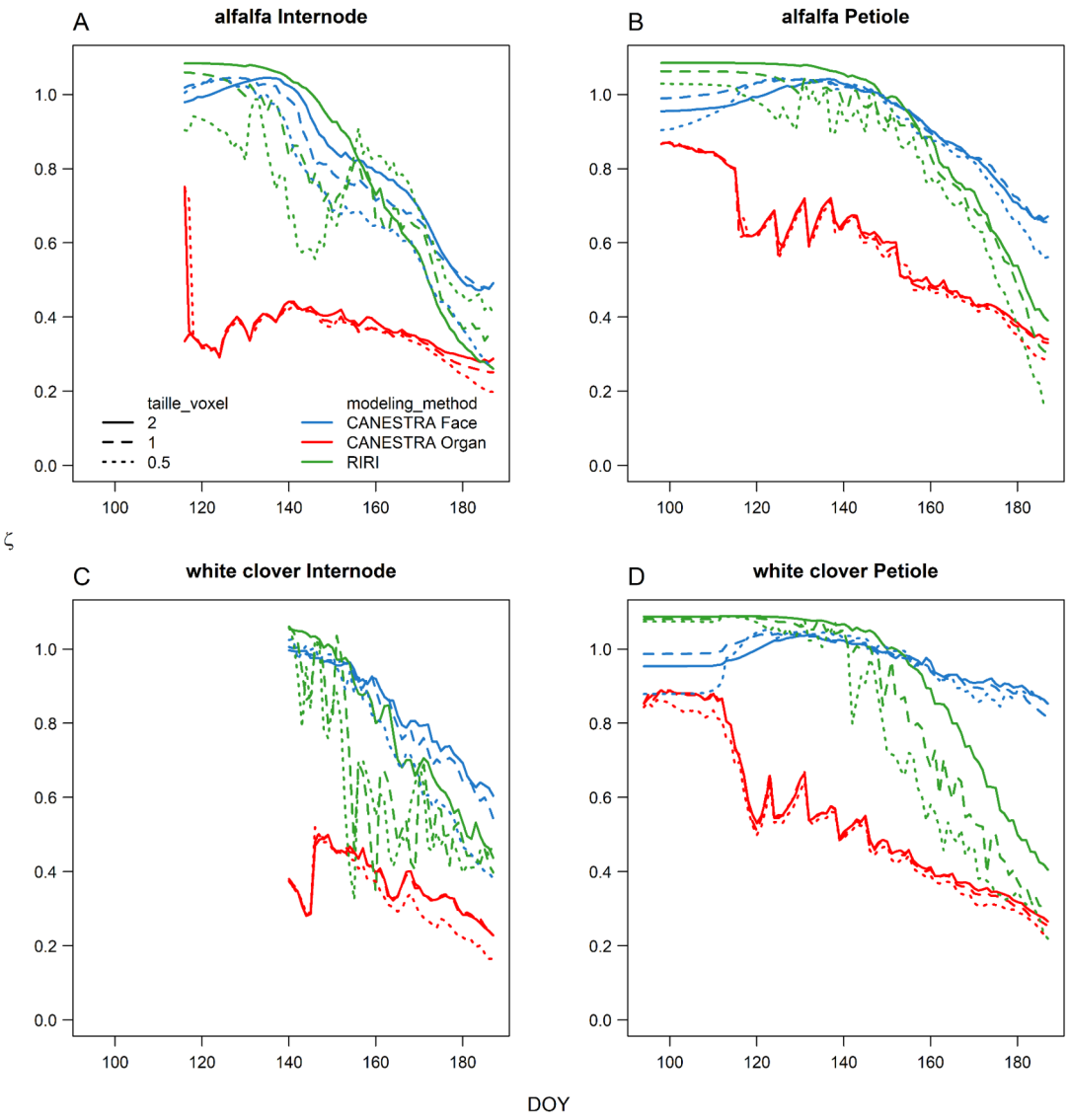


Figure 40: Evolution du ζ en fonction du DOY pour les entre-nœuds et les pétioles de luzerne et de trèfle blanc. Les résultats sont séparés par type de modèle utilisé et par taille de voxel, les simulation avec et sans rediffusion sont mises en commun pour les 2 modèles CANESTRA.

Le modèle CANESTRA Organ est très peu sensible à la paramétrisation de la taille des voxels pour l'ensemble des organes, avec une variation du ζ maximale observée pour les entre-nœuds de trèfle blanc autour de 0.1.

A l'inverse, la taille des voxel a un fort impact sur les résultats de simulation du ζ pour les simulations réalisées avec le modèle RIRI. Les pétioles de luzerne sont les moins impactées par ce paramètre, avec un écart de ζ maximum de 0.2 entre les simulations effectuées pour une taille de voxel de 0.5 et une taille de voxel de 2.

2.1.2. Effet de la densité

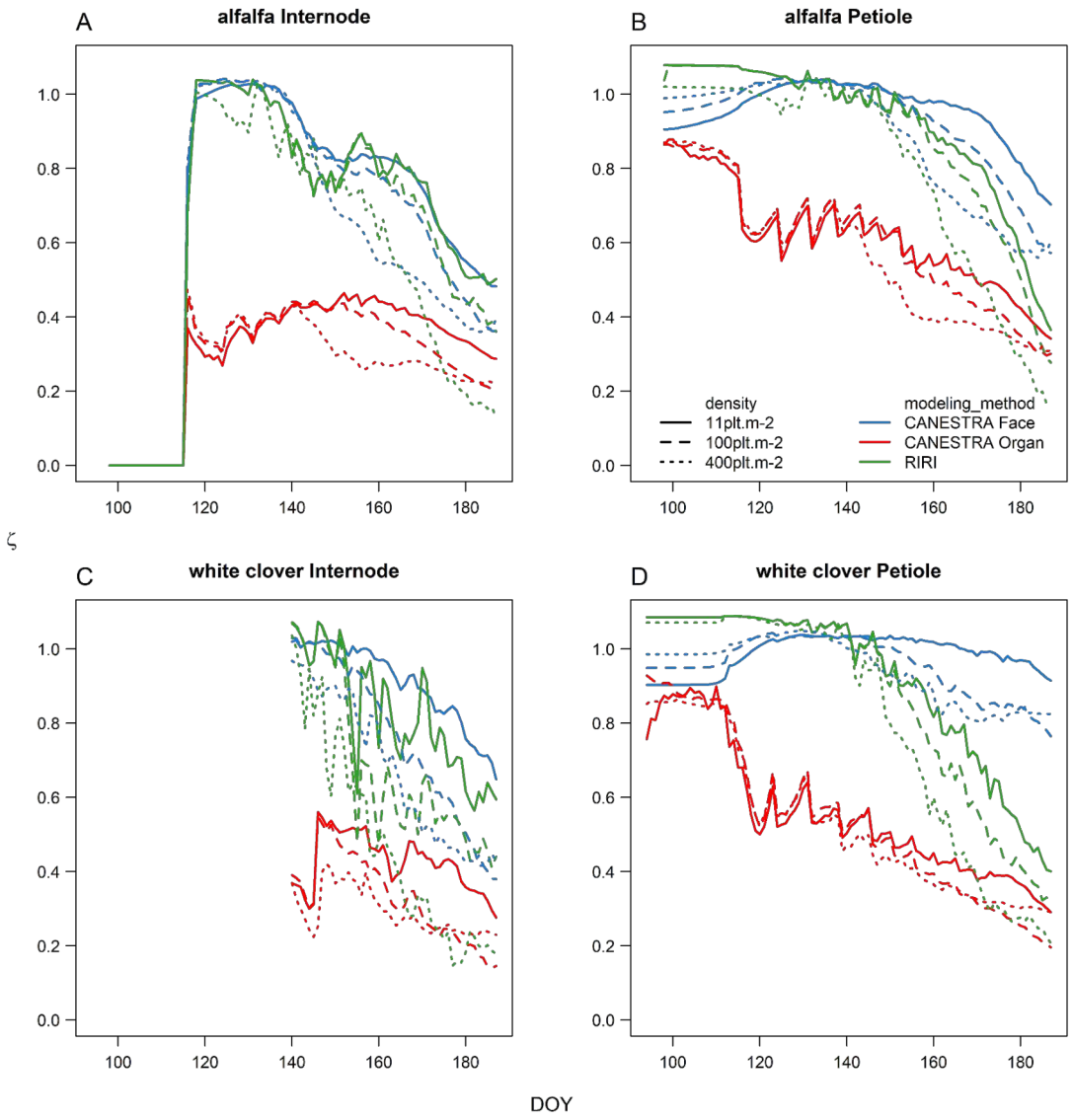


Figure 41: Evolution du ζ en fonction du DOY pour les entre-nœuds et les pétioles de luzerne et de trèfle blanc. Les résultats sont séparés par type de modèle utilisé et par densité, les simulation avec et sans rediffusion sont mises en commun pour les 2 modèles CANESTRA.

Les résultats de simulation pour différentes densités (Fig. 41) montrent pour l'ensemble des modèles un ζ plus faible lorsque la densité augmente. L'écart de ζ simulé est marqué à partir de 140DOY, à partir de laquelle une amplitude des valeurs est observée pour un même modèle en fonction de la densité. Le modèle de référence CANESTRA Organ présente la plus faible amplitude de ζ entre la plus faible et la plus grande densité. À l'inverse, le modèle RIRI est le plus sensible à la densité. L'impact de la densité étant très fort sur la valeur de ζ simulée, notamment à

400 plantes.m^{-2} où l'on observe la plus grande amplitude de variation de ζ, c'est à cette densité que l'effet de la modélisation sur les réponses morphogénétiques est étudié. De plus, cette densité est proche de situations agronomiques pour ces espèces .

2.2. Effet de la méthode de modélisation des rediffusions sur le calcul du ζ

La comparaison entre le modèle de référence et les autres modèles est faite pour l'ensemble des simulations. Cette comparaison combine tous les ζ simulés à chaque pas de temps des modèles pour chacun des phytomères présents dans la scène. Les résultats sont séparés par organes et par densité.

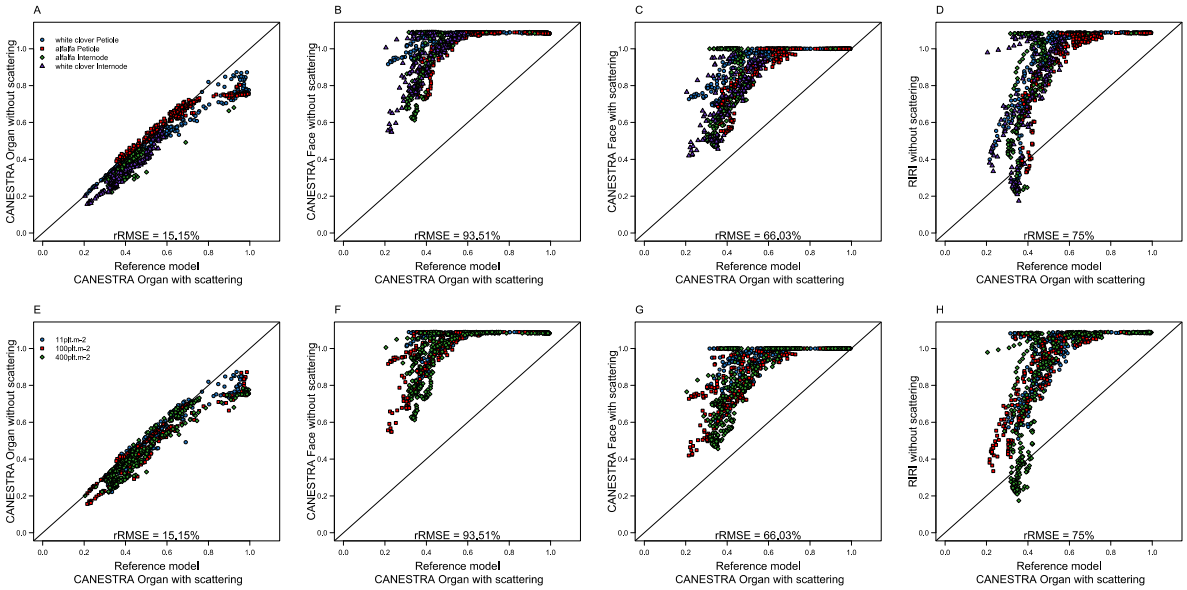


Figure 42: Comparaison entre le ζ calculé par chaque modèle et le modèle de référence CANESTRA Organ avec rediffusion pour les entre-nœuds et les pétioles de la luzerne et du trèfle blanc (A-D) et pour chaque densité (E-H).

CANESTRA Organ sans rediffusion présente une sous-estimation générale du ζ simulé par la relation fonctionnelle à partir du TrPPFD (Fig. 42A, E). Les deux modèles CANESTRA Organ suivent une relation linéaire pour les valeurs de ζ inférieures à 0.7.

La comparaison du ζ simulé entre le modèle de référence et les simulations effectuées avec les modèles CANESTRA Face et RIRI suivent une courbe hyperbolique. CANESTRA Face sans rediffusion et RIRI sont plafonnées à une valeur de 1.09, cette valeur correspond à la valeur maximale de ζ permise par la relation fonctionnelle entre TrPPFD et ζ (Escobar-Gutiérrez et al.

2009). Le ζ simulé par CANESTRA Face avec rediffusion est plafonné à une valeur de 1 correspondant au ζ incident.

La comparaison par densités (Fig. 42E-H) montre peu d'effet de la densité des couverts sur le calcul de ζ, à l'exception du modèle RIRI où un gradient de valeurs calculées peut être observé (Fig. H).

La forme hyperbolique observée sur les comparaisons entre le modèle de référence et les modèles CANESTRA Face et RIRI est due à la distribution de la valeur ζ calculée dans le temps (Fig. 42). Pour les deux organes et les deux espèces végétales, le ζ calculé par le modèle de référence est inférieur à celui des modèles CANESTRA Face (avec ou sans rediffusion) et RIRI. Cependant, à la fin des simulations (DOY 170-180), le ζ simulé par le modèle RIRI est plus faible que le modèle de référence. D'autre part, le ζ calculé par CANESTRA Face sans rediffusion a les valeurs les plus élevées à chaque DOY et pour chaque organe.

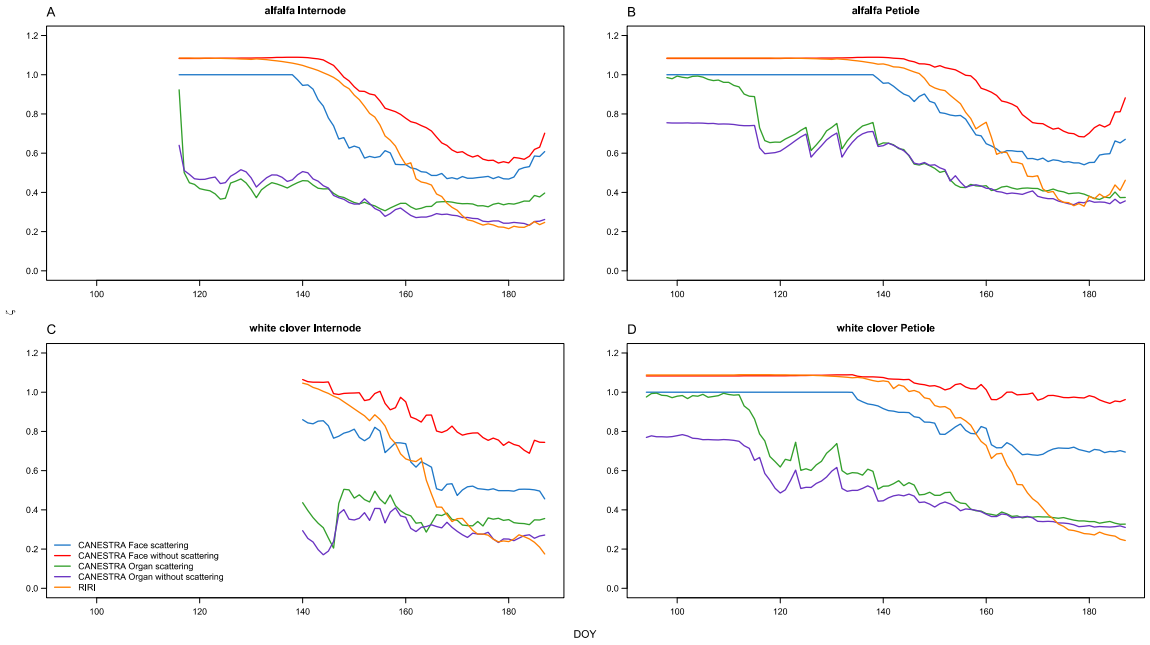


Figure 43: Moyenne du ζ simulé par les différents modèles de transfert radiatif en fonction de la date (DOY), pour les entre-nœuds et les pétioles de la luzerne et du trèfle blanc à une densité de 400 plantes.m^{-2}.

Si l'on regarde l'évolution temporelle de ζ (Fig. 43), la valeur maximale pour la face CANESTRA et RIRI précédemment observée sur la comparaison entre les modèles correspond aux calculs effectués au début de la simulation (DOY 110-140) lorsque les autres organes ne sont pas encore produits. Le ζ simulé par CANESTRA Organ pour les entre-nœuds de la luzerne commence

autour de 0,9 et diminue rapidement sous 0,6 avant le DOY 120. Le ζ calculé pour les entre-nœuds de trèfle blanc présente des valeurs inférieures à 0.7 à leur apparition au DOY 140, lorsque la canopée est déjà développée. Des oscillations sont observées pour les deux modèles d'organes CANESTRA à chaque apparition d'un nouvel organe.

2.3. Stade de développement et architecture des plantes en interaction avec le ζ

Les deux groupes morphogénétiques luzerne et trèfle blanc représentent deux architectures végétales contrastées. La figure 44 présente la moyenne de ζ calculée pour les entre-nœuds et les pétioles de chaque phytomère des plantes lorsqu'elles atteignent respectivement leur maturité.

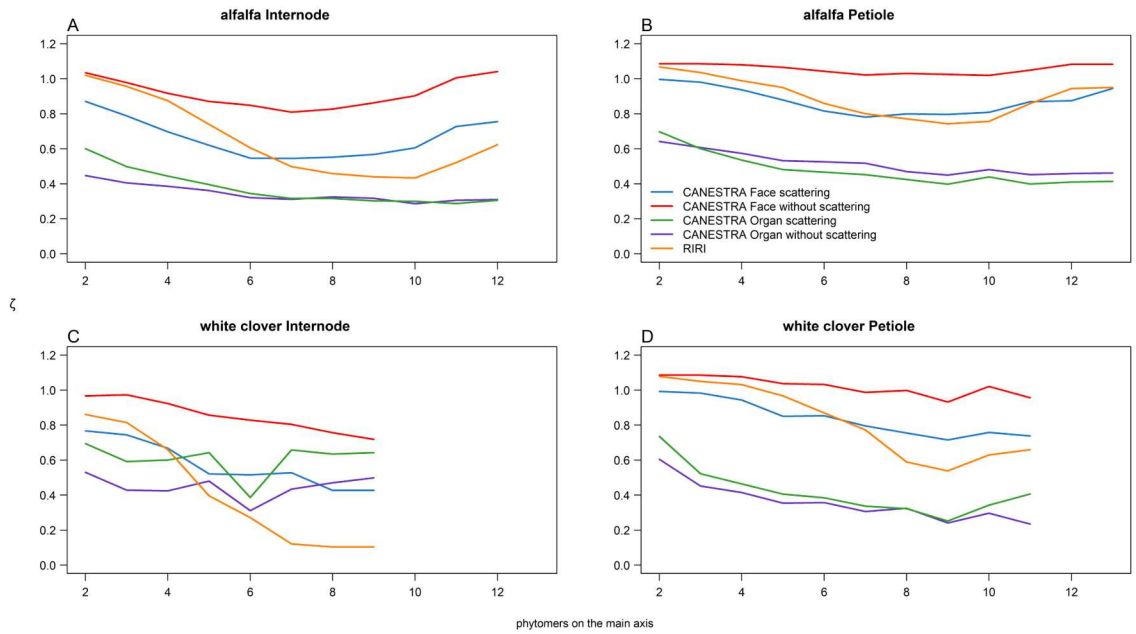


Figure 44: Comparaison de la moyenne du ζ simulé par les différents modèles de transfert radiatif pour chaque phytomère mature des plantes à une densité de 400 plantes.m^{-2}, en fonction des organes et des groupes morphogénétiques des plantes. Les valeurs de ζ utilisées pour chaque phytomère sont prises à différentes date car le ζ est pris à la première date de croissance où chaque phytomère respectif devient mature.

Le ζ moyen calculé par le modèle de référence CANESTRA Organ avec rediffusion pour la luzerne commence autour de 0,6 pour les entre-nœuds et 0,7 pour les pétioles et diminue de façon linéaire jusqu'à 0,3 et 0,4 respectivement à partir du 9^{e} phytomère (Fig. 44A-B). Pour le trèfle blanc, le ζ calculé par le modèle de référence commence autour de 0,75 pour les entre-nœuds et les pétioles (Fig. 44C-D). Pour les entre-nœuds du trèfle blanc, une lente diminution est observée

jusqu'au dernier entre-nœud avec une valeur finale de ζ d'environ 0,5 (Fig. 44C). Une forte diminution de ζ est également observée pour les pétioles du modèle de référence de 0,7 pour le 2e phytomère à 0,5 pour le 3e phytomère suivie d'une diminution plus lente jusqu'au dernier pétiole avec une valeur finale de ζ autour de 0,35 (Fig. 44D).

Le ζ calculé par les modèles CANESTRA Face et RIRI pour les entre-nœuds de la luzerne montre un pattern pour lequel le ζ simulé diminue pour les 10 premiers rangs, avant une augmentation sur les derniers phytomères (Fig. 44A). Le ζ calculé pour les pétioles de la luzerne par les modèles CANESTRA Face et RIRI montre une faible diminution du ζ de 1,0 à environ 0,8 entre le deuxième et le septième phytomère et une augmentation aux valeurs initiales pour le dernier phytomère (Fig. 44B).

Le ζ calculé par RIRI pour les entre-nœuds du trèfle blanc diminue rapidement d'environ 0,9 à 0,1 et a une valeur plus faible par rapport à ζ calculé par tous les modèles CANESTRA et a une valeur plus faible que le modèle de face CANESTRA du 4e phytomère et une valeur plus faible que les modèles d'organes CANESTRA du 5e phytomère (Fig. 44C).

2.4. Effet des valeurs de ζ simulées sur la taille finale des organes simulés

La longueur finale des organes à la fin de chaque simulation est montrée dans la figure 45. Chaque organe calculé à l'aide des modèles CANESTRA Organ avec et sans rediffusion a une longueur proche à chaque rang de phytomère et est plus long que les organes produits par les autres modèles, à l'exception des entre-nœuds du trèfle blanc où chaque modèle produit des organes avec une longueur similaire. CANESTRA Face sans rediffusion produit les organes les plus courts pour la luzerne et pour le pétiole du trèfle blanc. Les deux pétioles calculés à l'aide de RIRI et de CANESTRA Face sans rediffusion ont des organes de longueur similaire, surtout pour les rangs de phytomères 1 à 6.

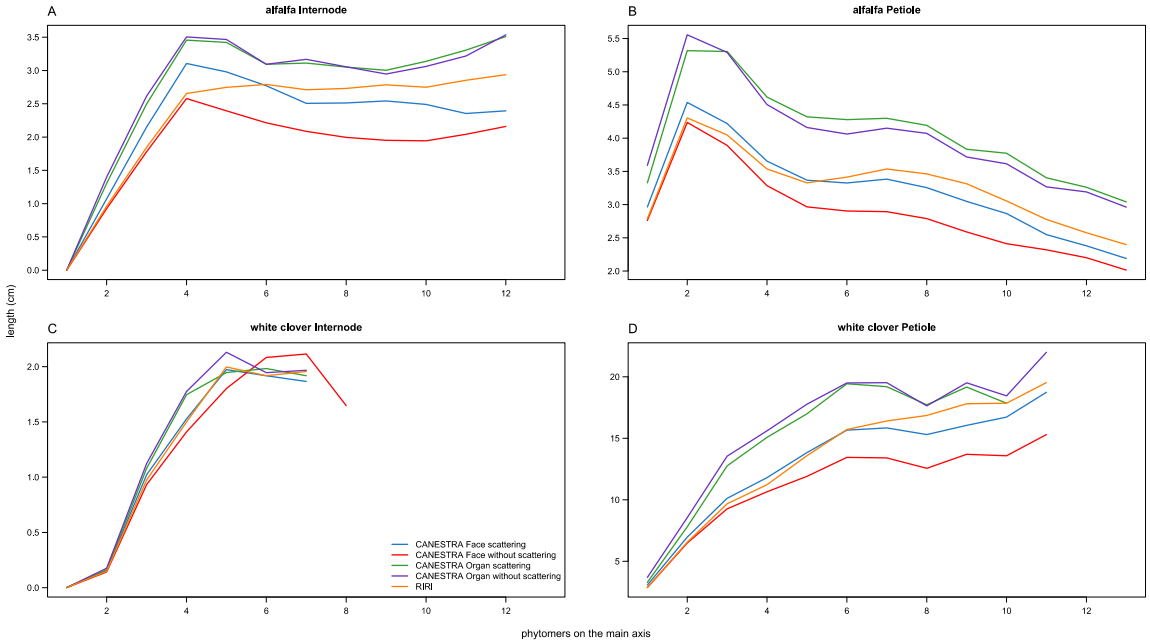


Figure 45: Longueur moyenne des organes finaux des différents modèles de transfert radiatif pour chaque phytomère mature, pour les entre-nœuds et les pétioles de la luzerne et du trèfle blanc à une densité de 400 plantes.m^{-2}.

Nous pouvons observer un pattern similaire entre les cinq méthodes de modélisation ζ pour chaque organe végétal respectif, puisque la même augmentation et la même diminution de la longueur des organes sont observées au même phytomère entre les modèles. Les entre-nœuds de la luzerne atteignent leur longueur maximale au 4e rang phytomère lorsque CANESTRA est utilisé et au 12e rang phytomère lorsque RIRI est utilisé (Fig. 45A). Les entre-nœuds de la luzerne calculés à l'aide de CANESTRA se raccourcissent après avoir atteint leur longueur maximale, ceux calculés à l'aide de RIRI ont une longueur stable entre le 4e et le 12e phytomère (Fig. 45A). Les pétioles de la luzerne atteignent rapidement leur longueur maximale au 2^{e} phytomère avant de se raccourcir sur les phytomères suivants pour chaque méthode de modélisation (Fig. 45B). Les entre-nœuds du trèfle blanc atteignent tous leur longueur maximale à différents rangs de phytomères et les pétioles du trèfle blanc continuent d'augmenter leur longueur jusqu'au dernier phytomère observé (Fig. 45C,D). À l'exception des entre-nœuds du trèfle blanc, les organes sont plus longs pour les plantes modélisées avec les modèles d'organes de CANESTRA.

2.5. Sensibilité de la morphogenèse à la modélisation du ζ

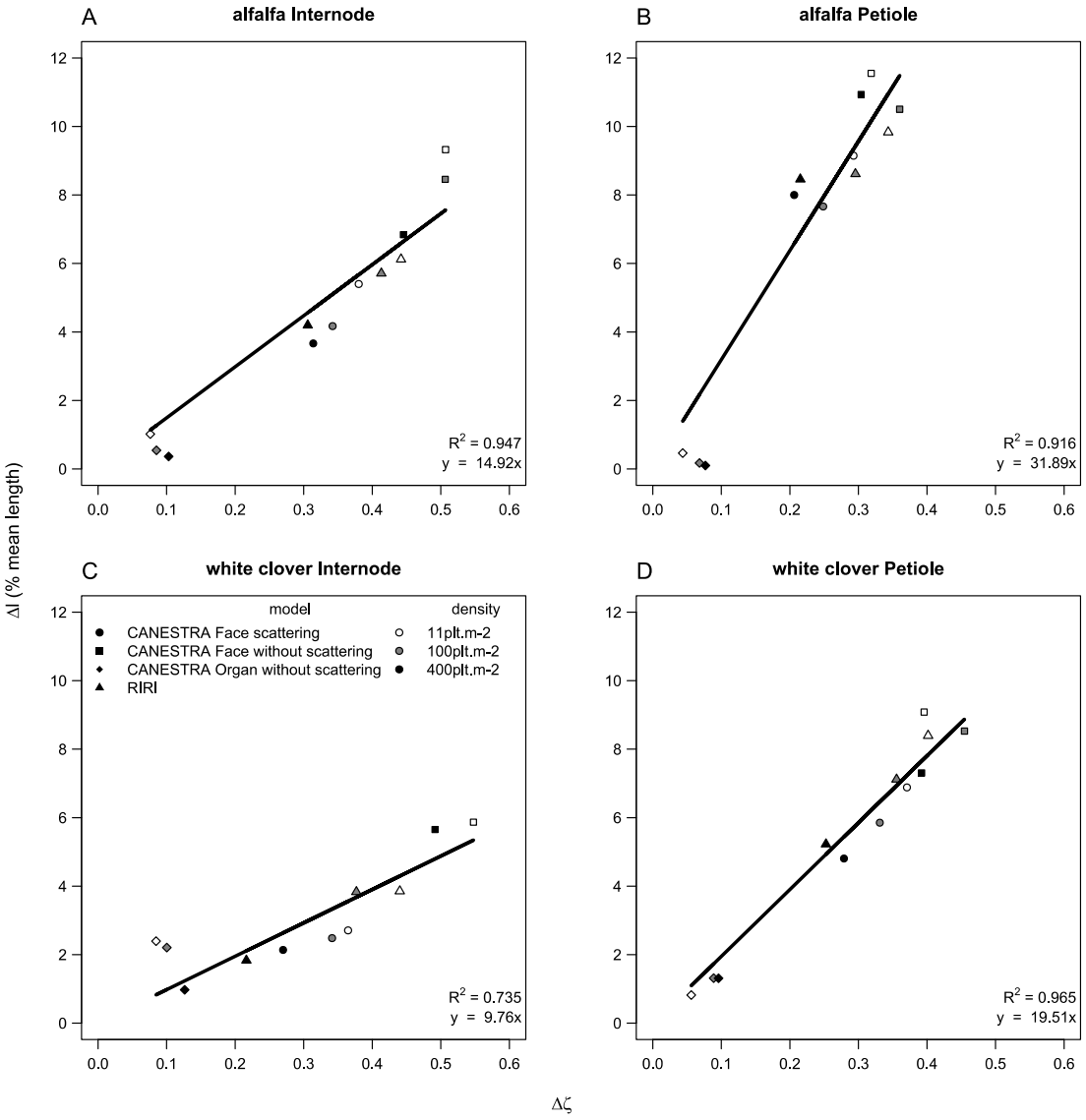


Figure 46: Sensibilité de la morphogenèse aux variations de ζ. Les résultats du modèle de référence sont utilisés pour le calcul du Δ pour les autres méthodes de modélisation. Les variations de la longueur des organes sont basées sur un pourcentage de la longueur moyenne de leurs groupes respectifs (groupe morphogénétique et type d'organe).

La figure 46 présente la sensibilité de la morphogenèse au calcul de ζ. Δζ correspond à la différence de ζ entre RIRI, CANESTRA Face et CANESTRA Organ sans rediffusion avec le modèle de référence. Δl correspond à la différence de la longueur normalisée des organes simulés par VGL sous RIRI, CANESTRA Face et CANESTRA Organ sans rediffusion avec le modèle de référence. Lorsque Δl et Δζ sont comparés par organes, une relation linéaire peut être observée pour

les deux organes de chaque espèce (Fig. 46). La sensibilité de la morphogenèse est liée à la pente de cette relation, puisqu'une pente plus élevée signifie un delta plus important de la longueur de l'organe pour un même delta de ζ.

Pour chaque organe, Δl est faible pour les simulations réalisées avec CANESTRA Organ sans rediffusion, avec une variation maximale inférieure à 3 % pour les entre-nœuds du trèfle blanc. Les simulations réalisées avec le modèle CANESTRA Face sans rediffusion sont celles avec le Δl le plus élevé pour tous les organes. Pour chaque organe, le Δl observé augmente avec la densité pour chaque approche de modélisation ζ.

Les pétioles sont plus sensibles aux variations de ζ que les entre-nœuds, avec une valeur de pente proche du double de la pente des entre-nœuds pour le trèfle blanc, respectivement 31,89 et 14,92, et la luzerne, respectivement 19,51 et 9,76 (Fig. 46).

Ces résultats montrent une plus grande sensibilité de la longueur des organes à la modélisation de ζ pour les pétioles par rapport aux entre-nœuds, et une plus grande sensibilité de la luzerne entre chaque catégorie d'organes.

2.6. Comparaison de la croissance des plantes simulées avec des mesures

Afin d’estimer l’impact du choix de la méthode de modélisation du ζ sur la morphogenèse, une comparaison avec des mesures sur des plantes réelles est effectuée. En effet, si les variations de morphogenèse produisent des organes ayant des propriétés aberrantes par rapport au modèle de plante attendue, ces méthodes de modélisation du ζ seraient écartées comme choix pour de futurs tests in silico.

La taille des organes simulés ainsi que leur taille relative sont comparées à des mesures effectuées par Gautier et al. (2000) et Faverjon (2018). La taille relative correspond à la taille des organes par rapport à leur taille maximale potentielle.

Gautier et al. (2000) ont réalisés des mesures de la taille des pétioles (Fig. 47) et des entre-nœuds (Fig. 48) de trèfle blanc pour 2 densités. Bien que ne suivant pas exactement la courbe de ces mesures, l’ordre de grandeur de la taille des organes ainsi que la forme de la courbe de croissance des simulations effectuées par l’ensemble des modèles que nous avons testés sont proches de leurs mesures.

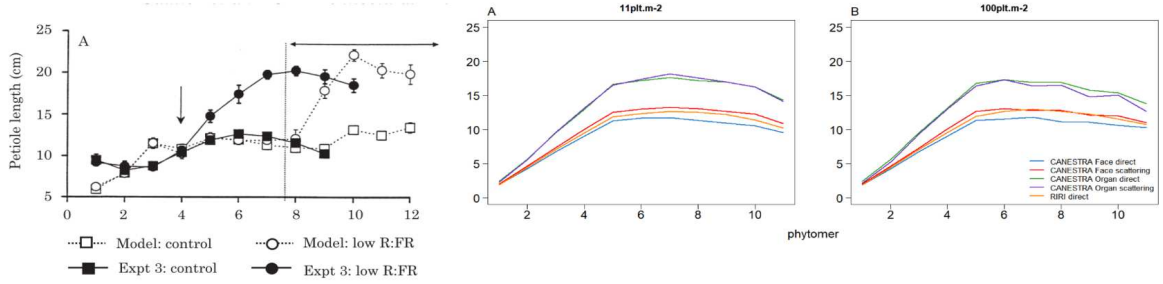


Figure 47: Comparaison entre les données de Gaultier et al. (2000) et les données de simulation de CANESTRA et RIRI pour les pétioles de trèfle blanc. Les données de Gaultier et al. ont été obtenues pour une densité de 100 plantes.m-2.

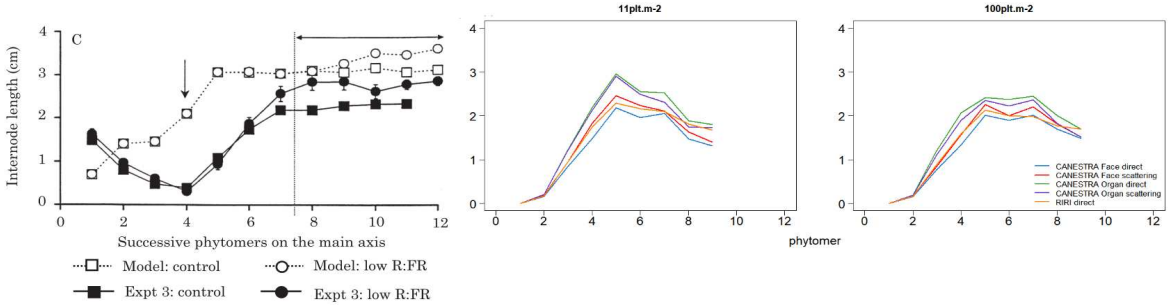


Figure 48: Comparaison entre les données de Gaultier et al. (2000) et les données de simulation de CANESTRA et RIRI pour les entre-nœuds de trèfle blanc. Les données de Gaultier et al. ont été obtenues pour une densité de 100 plantes.m-2.

La taille relative des entre-nœuds de luzerne et des pétioles de trèfle blanc a été mesurée par Faverjon (2018), une comparaison a été faite pour des conditions de densité similaire, soit à 11 plantes.m-2 (Fig. 49). Pour la luzerne, la taille relative simulée est plus faible que celle mesurée pour l'ensemble des modèles, cependant pour les 2 situations la taille relative entre les positions de phytomères est stable autour de 1 pour les mesures et entre 0.5 et 0.8 selon le modèle pour nos simulations. Pour les pétioles de trèfle blanc, une augmentation linéaire de la taille relative peut être observée sur les premiers rangs des phytomères. Contrairement aux mesures, la taille relative simulée par l'ensemble des modèles diminue à partir du 8e phytomère. Le pattern de taille relative étant identique pour l'ensemble des méthodes de modélisation du ζ, le pattern observé sur les simulations est lié à la construction du modèle VGL et non au choix du modèle radiatif. De plus, les simulations effectuées avec le modèle radiatif de référence présentent les résultats de taille relative les plus proches des valeurs mesurées à la fois pour le trèfle et la luzerne.

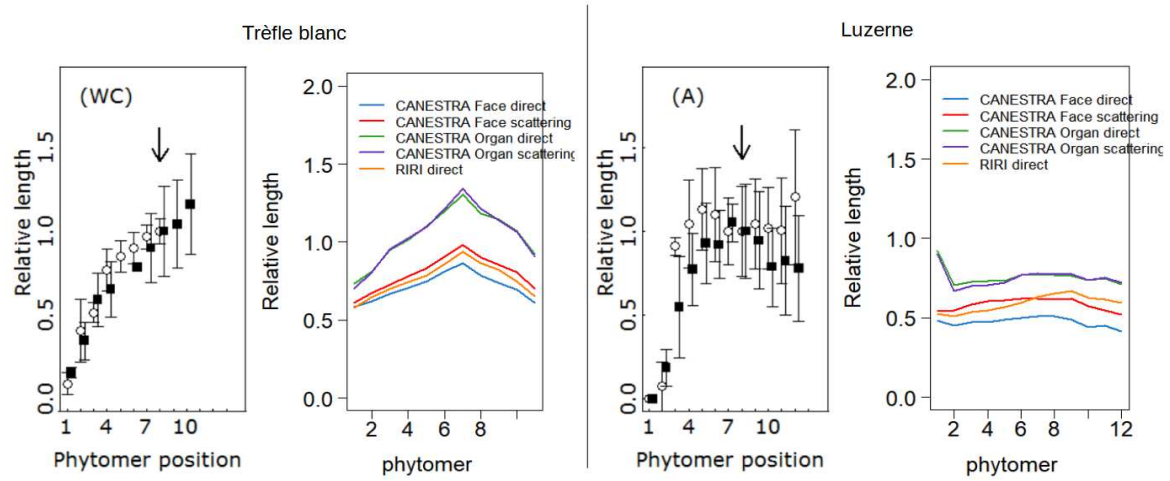


Figure 49: Comparaison de la taille relative des entre-nœuds de luzerne et des pétioles de trèfle blanc mesuré par Faverjon (2018) et les données de simulation de CANESTRA. Les données de Faverjon ont été obtenues pour une densité de 10 plantes.m-2.

La biomasse aérienne moyenne de la luzerne et du trèfle blanc simulée par le modèle VGL a été évaluée par Faverjon (2018) (Fig. 50). Pour chacune des densités, l'ordre de grandeur de la biomasse pour la luzerne et le trèfle blanc simulé par nos modèles est le même que celui mesuré et simulé dans la thèse de Faverjon.

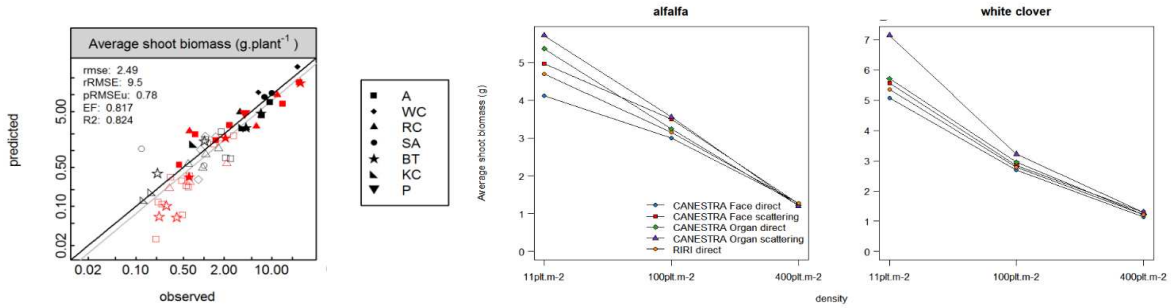


Figure 50: Comparaison de la biomasse moyenne des plantes simulées par CANESTRA et RIRI avec les mesures observées par Faverjon (2018). Les axes de la figure reprise de Faverjon (2018) sont dans une échelle logarithmique.

De la même manière, la hauteur des plantes simulées par les différentes méthodes de modélisation du ζ est proche des valeurs mesurées et simulées par Faverjon (Fig. 51). Pour la luzerne, la hauteur des plantes simulées par le modèle de référence correspond aux valeurs les plus proches de celles mesurées avec des hauteurs proches de 60 cm alors que les autres modèles produisent des plantes avec une hauteur entre 40 cm et 50 cm.

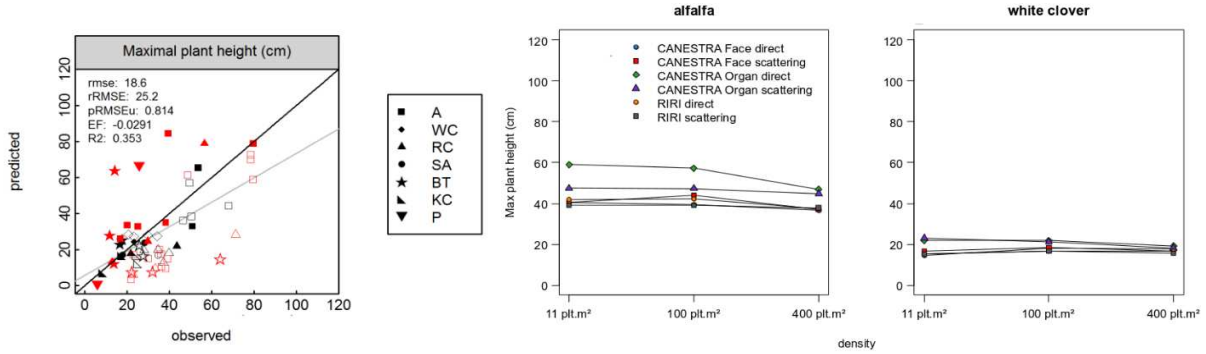


Figure 51: Comparaison de la hauteur maximale des plantes simulées par CANESTRA et RIRI avec les mesures de Faverjon (2018). Les axes de la figure reprise de Faverjon (2018) sont dans une échelle logarithmique.

Enfin, une comparaison entre les données de Faverjon et les simulations a été effectuée pour la surface foliaire (Fig. 52). À l'instar des autres paramètres morphologiques étudiés précédemment, les simulations effectuées par les différents modèles produisent des plantes avec des surfaces foliaires proches des mesures.

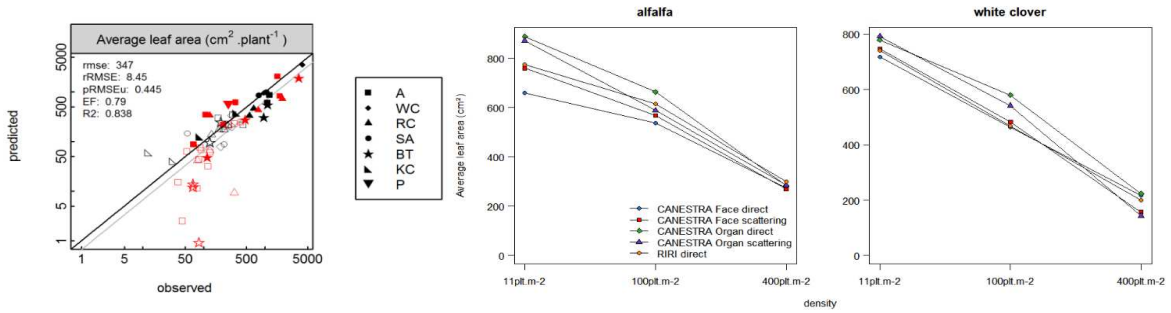


Figure 52: Comparaison de la surface foliaire moyenne des plantes simulées par CANESTRA et RIRI avec les mesures observées par Faverjon (2018). Les axes de la figure reprise de Faverjon (2018) sont dans une échelle logarithmique.

Ainsi, pour l'ensemble des paramètres morphologiques étudiés les plantes simulées avec les différents modèles ne présentent pas de résultat aberrant et le modèle de référence est le plus proche des valeurs attendues sur la majorité des paramètres.

3. Variabilité du signal directionnel

Afin de modéliser au mieux le ζ directionnel, des mesures du signal ont été effectuées pour évaluer l'impacte de la distance, de la présence de plantes voisines et de l'heure de mesure sur la variabilité du signal.

3.1. Evolution du ζ en fonction de la distance

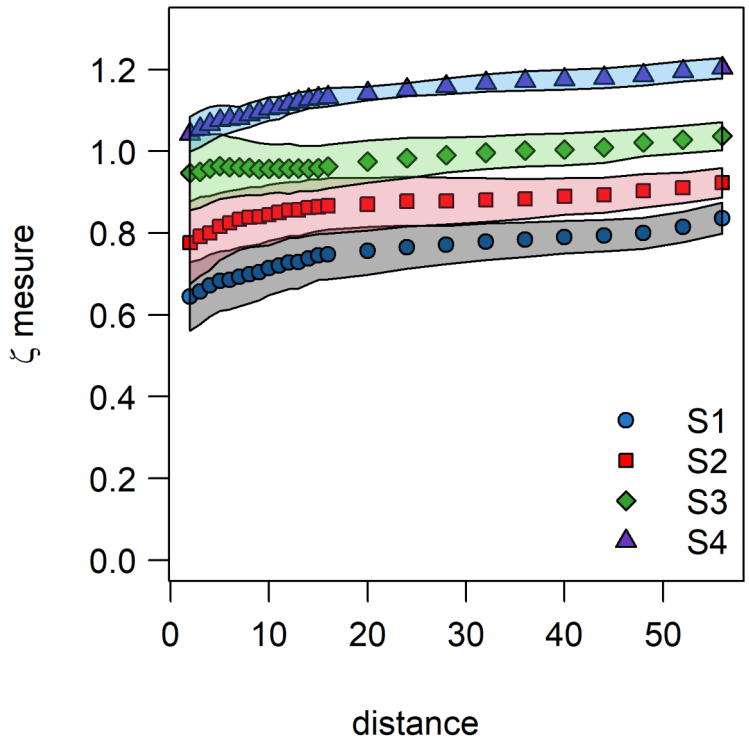


Figure 53: Evolution du ζ moyen en fonction de la distance pour 4 groupes de semis (S1-S4). Les valeurs représentent le ζ moyen mesuré pour l'ensemble des dates et pour l'ensemble des plantes de chaque groupe de semis. Un groupe de semis est constitué de 10 plantes.

Des mesures de l'environnement lumineux de plantes isolées ont été effectuées sur plusieurs distances avec l'expérimentation ArchiMono pour 4 groupes de semis, correspondant à 4 stades de développement (Fig. 53). Les résultats de ces mesures mettent en évidence un léger effet de la distance sur la mesure du ζ. Pour chaque groupe de semis, le ζ varie entre les mesures prises à 2 cm et à 56 cm, avec entre 0,65 et 0,75 pour les plantes du S1, entre 0,78 et 0,9 pour les plantes du S2, entre 0,95 et 1 pour les plantes du S3 et entre 1,03 et 1,17 pour les plantes du S4. Un effet du groupe de semis est donc également observable sur le stade de développement, les plantes les plus âgées ayant le ζ moyen le plus faible.

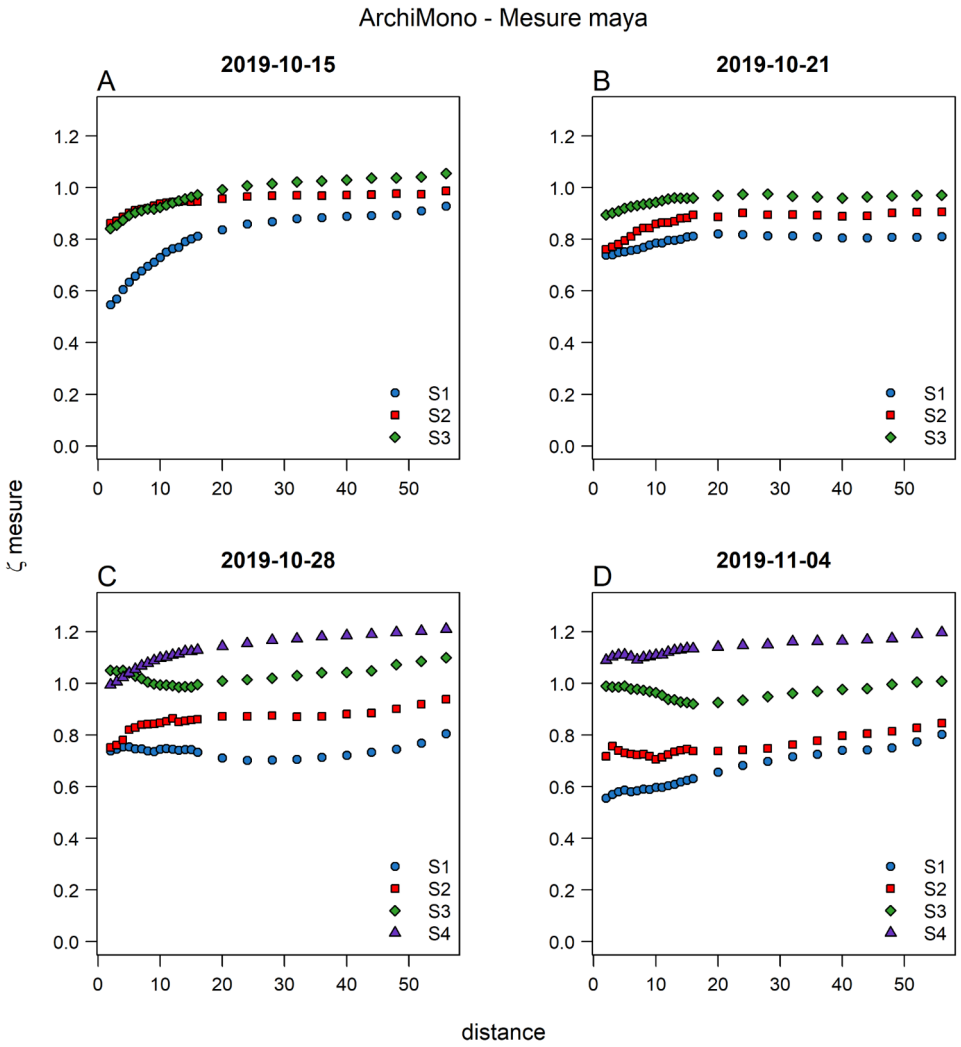


Figure 54: Evolution du ζ en fonction de la distance pour chaque date de mesure et par groupe de semis (S1-S4). Chaque courbe représente la valeur moyenne mesurée sur 10 plantes.

En séparant les résultats par date de mesure (Fig. 54), on observe 2 profils de variation du ζ en fonction de la distance. Le premier profil correspond à une augmentation continue du ζ lorsque la distance augmente jusqu'à ce que le ζ atteigne un plateau. Ce profil de variation peut être observé sur l'ensemble des groupes de semis pour les 2 premières dates de mesure (Fig. 54A,B) et pour les groupes de semis S2 et S4 pour la troisième date de mesure (Fig. 54C). Le second profil de variation observé correspond à une diminution du ζ avec la distance avant d'atteindre une valeur minimum suivie d'une augmentation du ζ mesuré. Ce profil de variation peut être observé pour les groupes de semis S1 et S3 pour la troisième date de mesure et pour les groupes S2, S3 et S4 pour la dernière date de mesure (Fig. 54C,D).

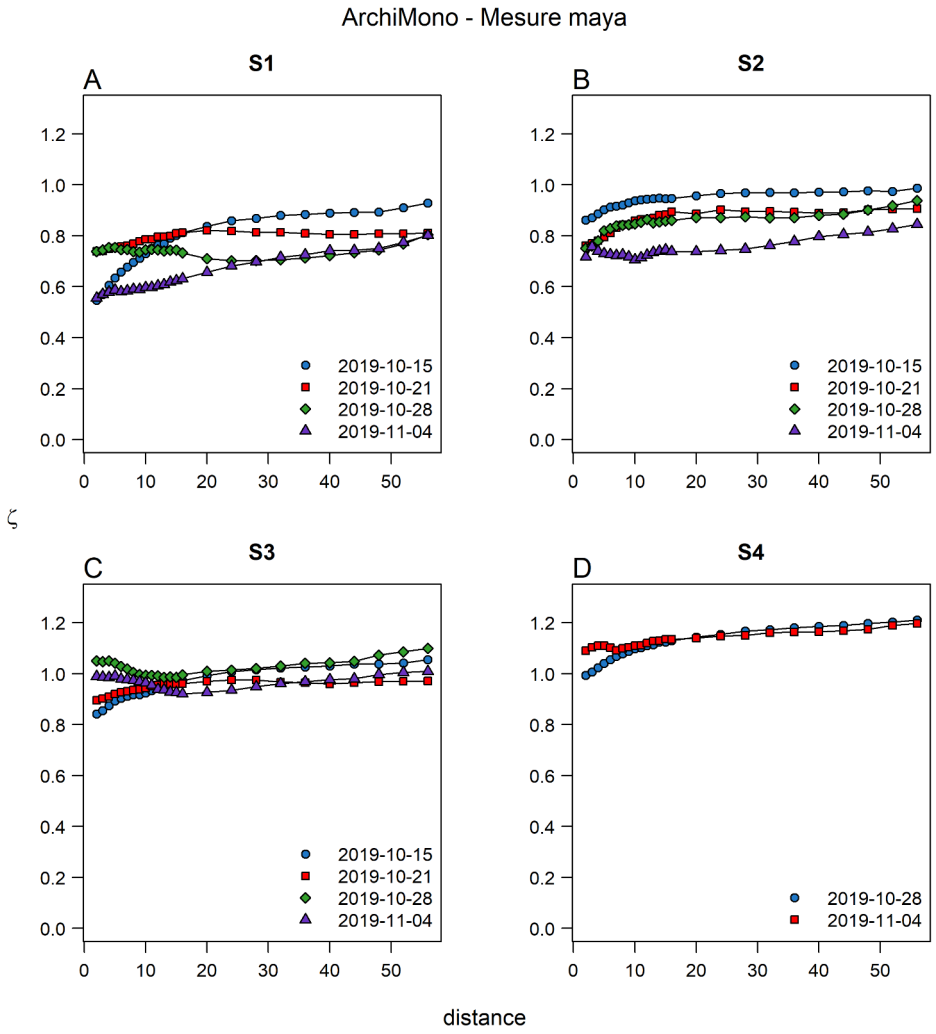


Figure 55: Evolution du ζ en fonction de la distance pour chaque groupe de semis et par date.

La figure 55 montre l'évolution du ζ en fonction de la distance à différentes dates pour un même groupe de semis. En complément des différences observées en fonction de la date (Fig. 54), la variation du ζ est différente en fonction de la date selon les groupes de semis. Le groupe de semis S1 montre une diminution du ζ entre la 2e et la 4e date de mesure pour les distances comprises entre 2 cm et 24 cm, les valeurs de zêta sont identiques pour les mesures de la 3e et de la 4e date à partir de 28 cm. Les valeurs de ζ mesurées pour le groupe de semis S4 sont très proches entre les 2 dates de mesure, elles sont comprises entre 1 et 1,2.

3.2. Evolution du ζ en fonction de l'heure de mesure

La hauteur du soleil varie au cours de la journée, ainsi cette variabilité pose la question du choix de l'heure pour la mesure du ζ. Ainsi, les variations du ζ directionnel sous serre au cours de la journée ont été mesurées pour un couvert au stade précoce (1 talle) et pour un couvert plus développé (10 talles) ainsi que pour des plantes isolées à ces deux stades de développement. Les mesures ont été effectuées en condition de ciel clair et en condition de ciel couvert.

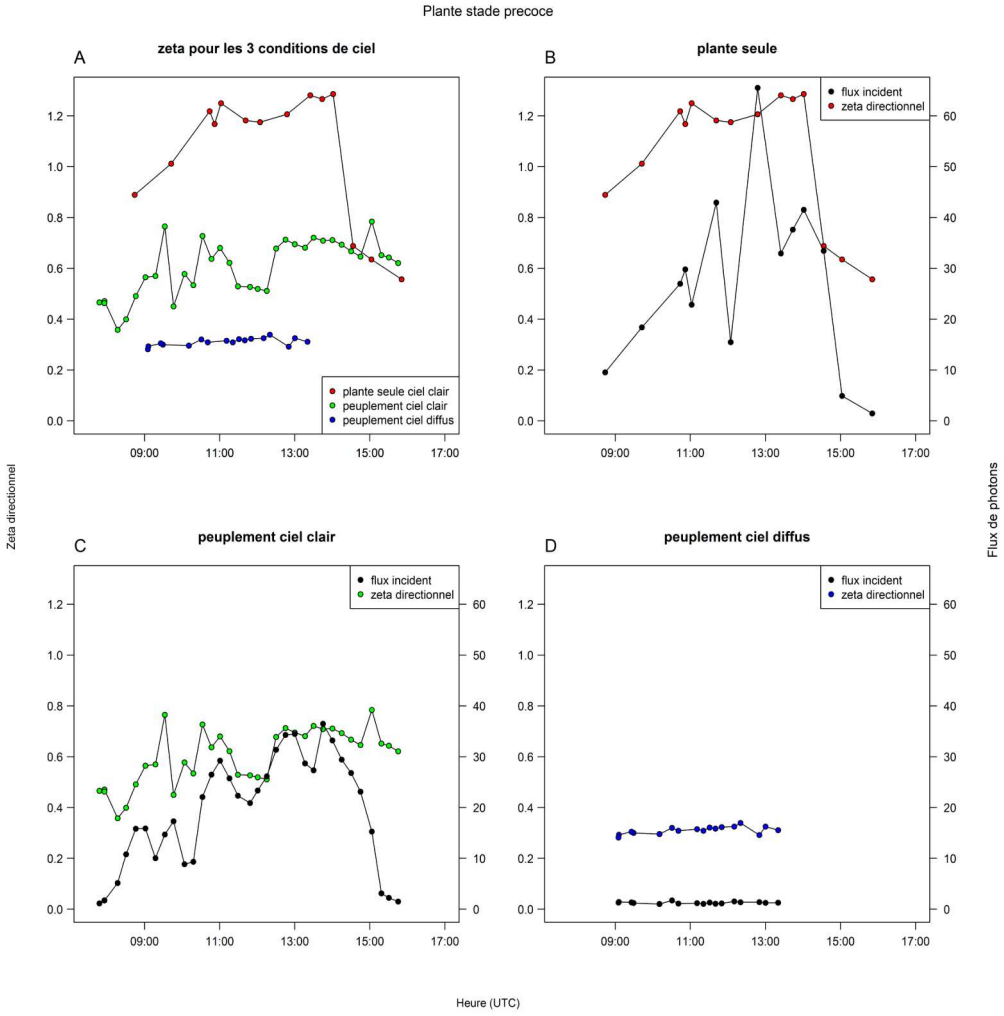


Figure 56: Evolution du ζ au cours d'une même journée (UTC) pour un stade de développement précoce de plants de ray-grass.

Les mesures à un stade précoce présentent des valeurs de ζ très différentes en fonction des conditions (Fig. 62A). Les mesures sur une plante isolée peu développée sous un ciel clair montrent un profil de ζ commençant à une valeur proche de 0.9 à 9 h UTC qui augmente jusqu'à atteindre une valeur supérieure à 1.2 entre 11 h et 15 h UTC avant de chuter pour atteindre un ζ inférieur à

0.6. Pour des conditions de ciel clair en peuplement, le ζ mesuré présente de fortes variations au cours de la journée, oscillant entre 0.38 et 0.8. Enfin, les mesures effectuées en conditions de ciel diffus sont les plus faibles, avec un ζ proche de 0.3 tout au long de la journée.

Les variations de ζ directionnel au cours de la journée sont comparées aux variations du flux incident (Fig. 56B-D). Pour la plante seule (Fig. 56B), une forte diminution du flux incident peut être observée autour de 12 h sur les mesures de flux incident suivie d'une forte augmentation autour de 12h30, ces fortes variations ne provoquent pas de forte chute du ζ directionnel mesuré. Pour le peuplement précoce sous ciel clair (Fig. 56C), le flux incident suit un profil similaire à celui du ζ directionnel, les augmentations de ζ suivent les augmentations du flux directionnel et les diminutions de ζ suivent les diminutions du flux directionnel. Cependant, des pics de valeurs du zêta observées autour de 9h30 et de 15 h ne sont pas corrélés à des pics de flux incident. Pour le peuplement précoce sous ciel couvert (Fig. 56D), le flux incident et le ζ directionnel présentent tout les deux peu de variations.

Pour une plante développée isolée (Fig. 57), les mesures de ζ sous un ciel clair oscillent fortement au cours de la journée et présentent une forte diminution à partir de 14 h UTC. Les profils de ζ mesurées pour les plantes au stade 10 talles en peuplement sont stables au cours du temps pour les 2 conditions météorologiques avec des valeurs de ζ inférieures à 0.2 tout au long de la journée.

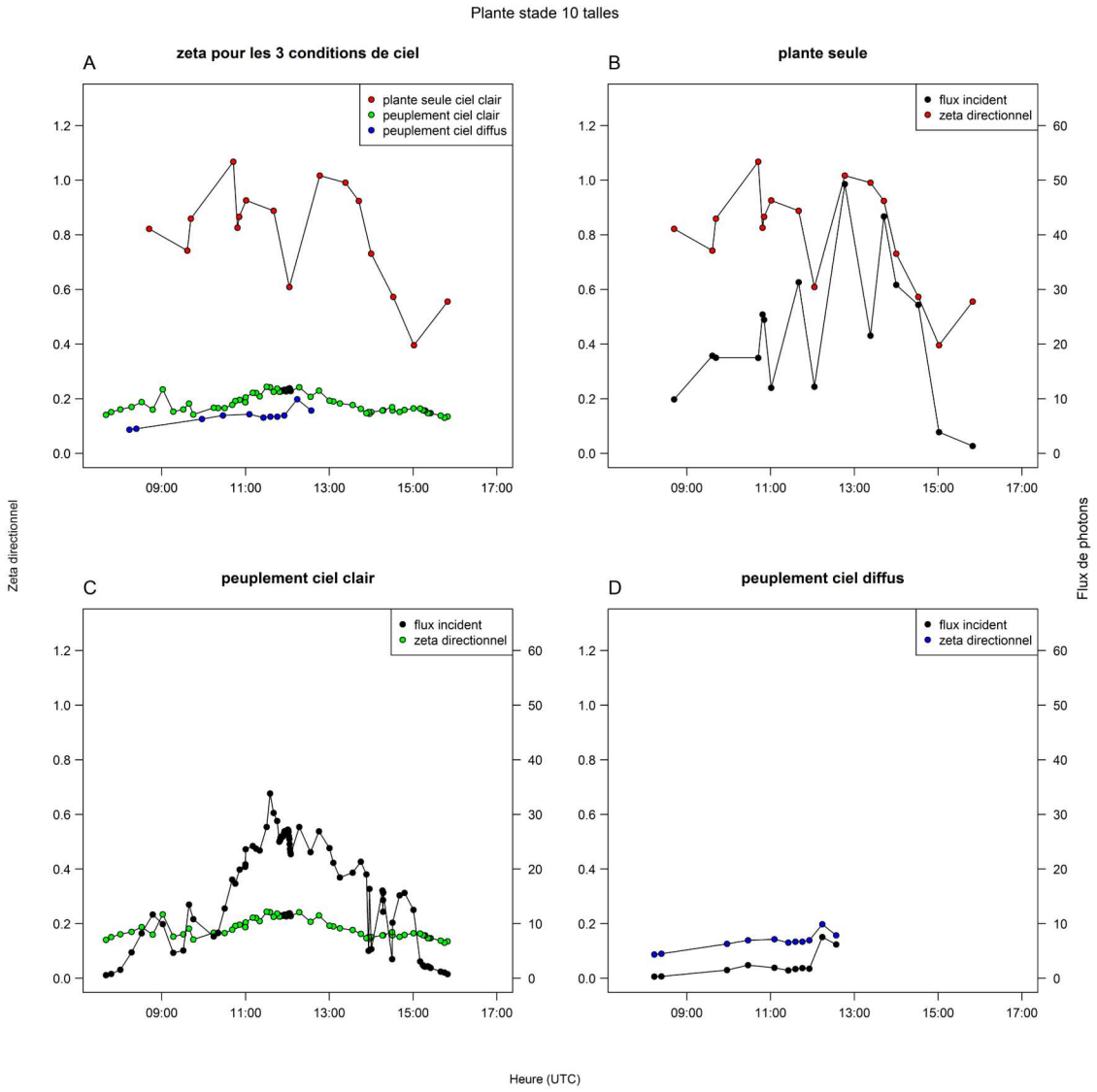


Figure 57: Evolution du ζ au cours d'une même journée (UTC) pour des plants de ray-grass au stade de développement de 10 talles.

3.3. Evolution du ζ en fonction du voisinage

L'effet du nombre de plantes présentes autour d'une plante cible sur le ζ mesuré a été réalisé en modifiant le nombre de plantes latérales (lignes) et le nombre de plantes positionnées derrières la plante cible (rangs), la distance entre les plantes est de 5 cm (HD) ou 10 cm (LD) (Fig. 58). Afin de supprimer les biais liés aux conditions de mesure (heure et conditions de ciel), les résultats sont comparés en normalisant les résultats par rapport au ζ mesuré en condition de plante isolée.

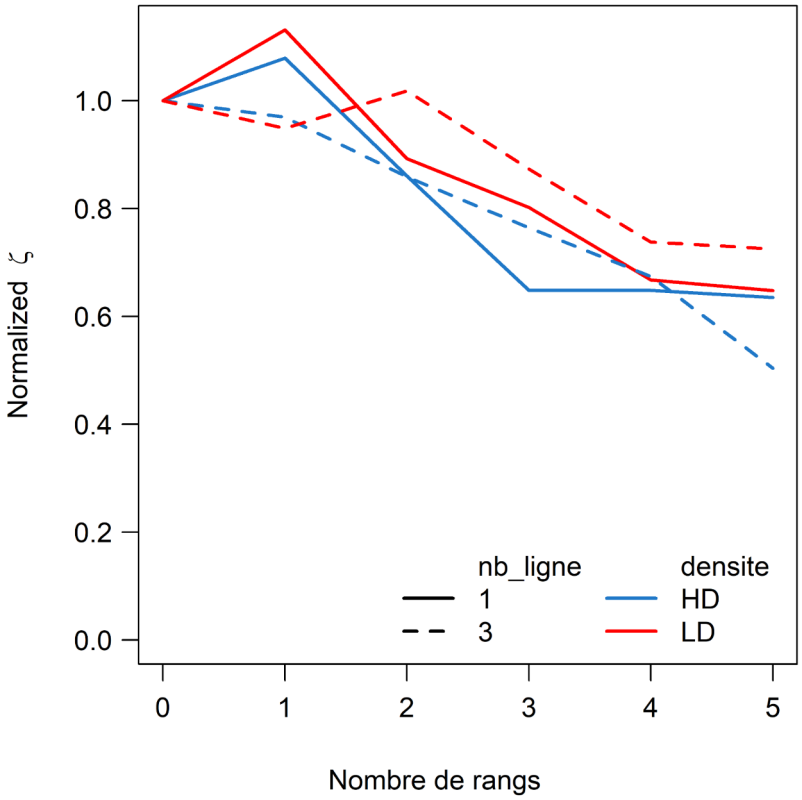


Figure 58: Evolution du ζ en fonction de la présence de plantes voisines. Le nombre de ligne (nb_ligne) indique si le peuplement était composés d'une ligne centrale ou d'une ligne centrale et deux lignes latérales. La condition LD correspond à un espacement de 5 cm entre les plantes et la condition HD a un écart de 5 cm, correspondant à des pots juxtaposés.

Pour les 4 conditions, les résultats montrent une diminution du ζ mesuré en fonction du nombre de rangs. En condition 1 ligne ou 3 lignes, le ζ normalisé est supérieur pour les conditions en faible densité par rapport aux conditions en forte densité.

Pour la condition « 1 ligne en forte densité », un plateau des valeurs du ζ est atteint lorsque 3 plantes sont positionnées derrière la plante cible, avec une valeur de ζ égale à 60 % du ζ initial. Un plateau de ζ à la valeur 0.6 est également observable pour la condition « 1 ligne en faible densité » et est observable lorsque 5 plantes sont positionnées derrière la plante cible.

4. Modélisation de la mesure du rayonnement directionnel

4.1. Modélisation simple du rayonnement directionnel

Une première approche simple de modélisation a été mise en place afin d'évaluer le lien entre la surface verte capturée (SVC) dans le champ de vision d'un capteur directionnel (Fig. 59) et le ζ mesuré. À partir de la reconstruction des plantes mesurées, une capture d'image est prise à l'aide de la caméra du modèle OpenAlea pour laquelle un champ de vision de 3° est appliqué. Le ζ mesuré est comparé au pourcentage de pixel de plante présent dans la région vue par la caméra.

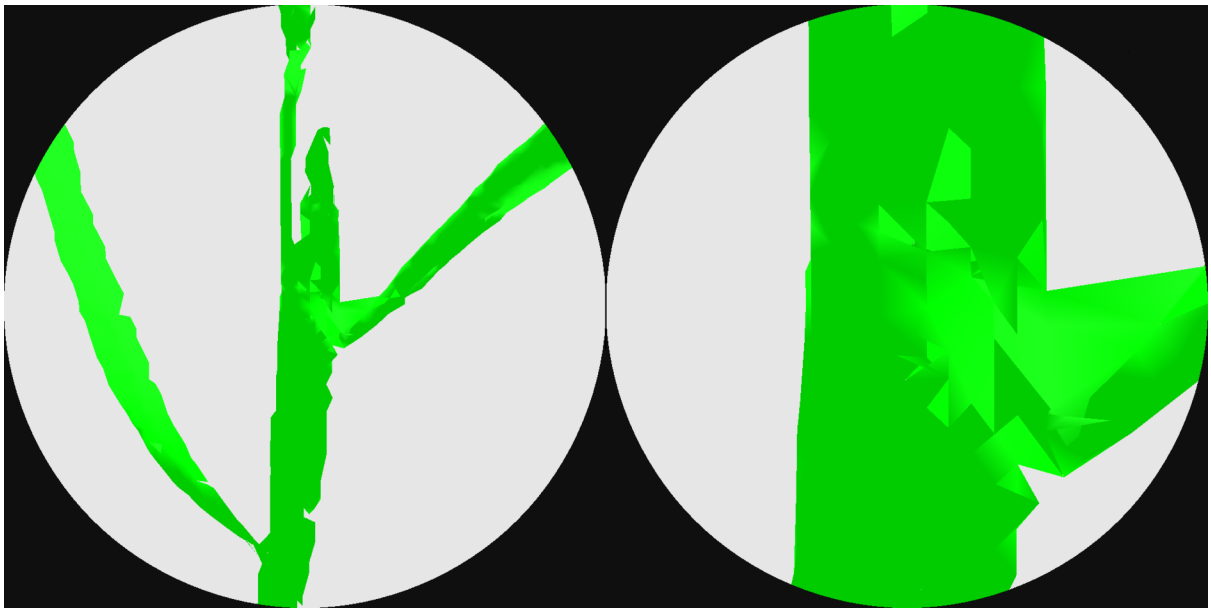


Figure 59: Exemple de mesures effectuées a l'aide du capteur SVC. A gauche le nombre de pixels de plante dans la zone du capteur est de 5 % et de 20 % sur la figure de droite.

La répartition des observations pour le SVC se situe selon une gamme variable pour l'ensemble des dates (Fig. 60). Les valeurs de SVC obtenues sont comprises entre 0 et 0.95. A partir de ces résultats, une relation linéaire négative entre le pourcentage de pixels de plante présent dans le champ de vision du capteur et le ζ mesuré est tracée.

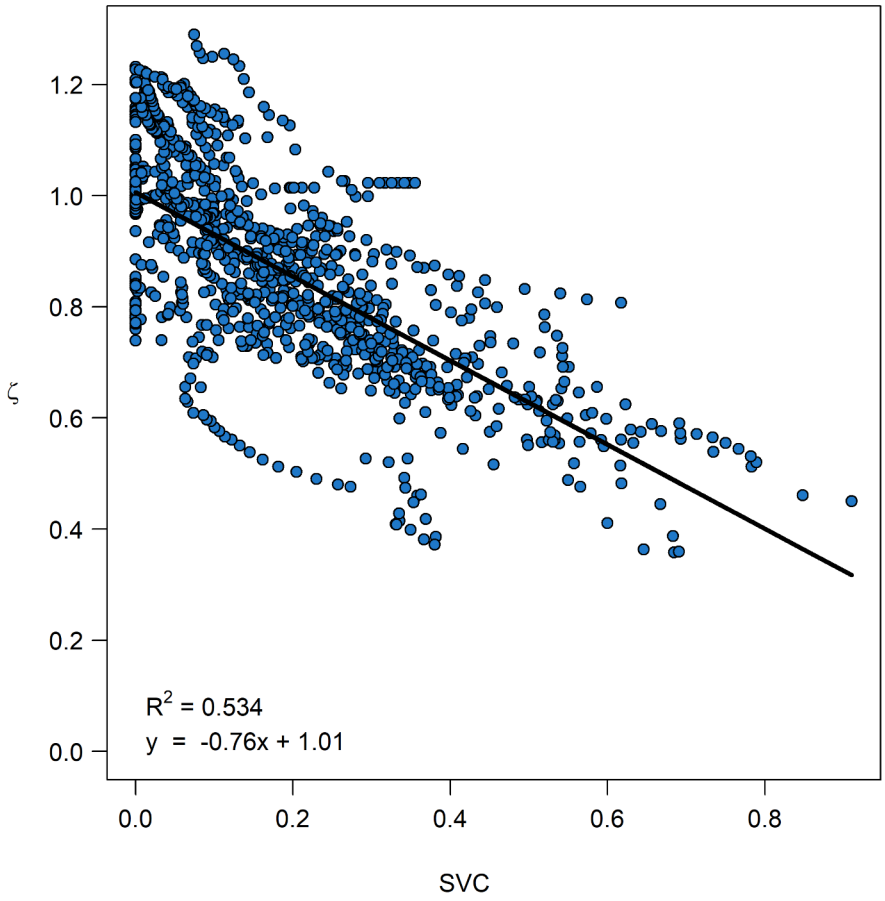


Figure 60: Corrélation entre la SVC obtenu à partir du modèle simple de capteur directionnel et le ζ mesuré. Les données correspondent à l'expérimentation ArchiMono.

Les données sont séparées par date de mesure (Fig. 61). La date du 21/10/2019 a la pente de corrélation entre la SVC et le ζ mesuré la plus faible, avec un coefficient direction de -0,54 ; la date du 28/10/2019 possède la pente la plus faible avec une valeur de -1,19. La corrélation entre la SVC et le ζ directionnel mesuré est la plus forte à la première date de mesure avec un R^2 de 0,801 et est la plus faible à la dernière date de mesure avec un R^2 de 0,439.

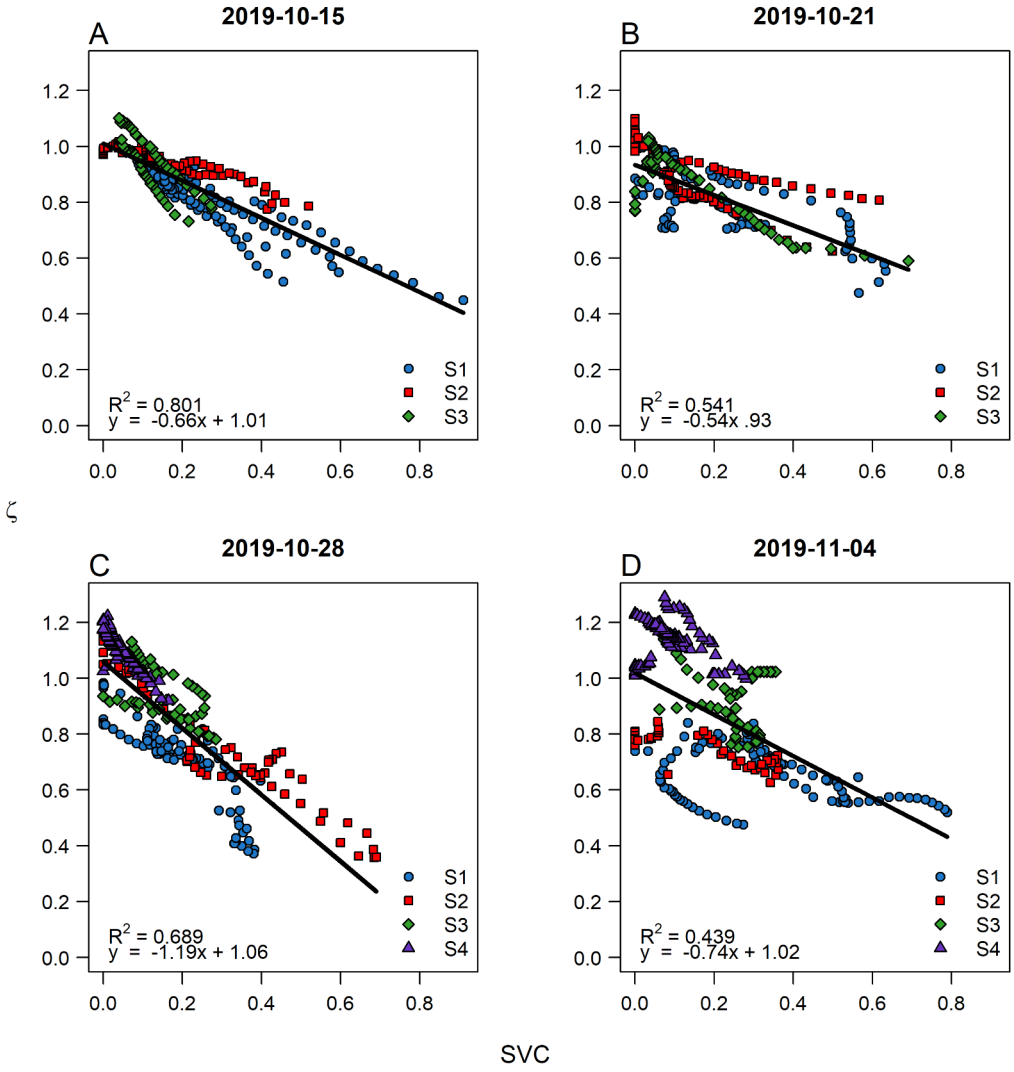


Figure 61: Corrélation entre la SVC et le ζ mesuré en fonction du groupe de semis pour chaque date de mesure.

La gamme de pourcentage de plantes vues par le capteur pour les données du 15/10/2019 est comprise entre 0.05 et 0.95, les valeurs de ζ mesurées sont équitablement dispersées au-dessus et au-dessous de la droite sur l'ensemble de la gamme de SVC. Pour cette date, le groupe de semis S1 possède la plus grande gamme de SVC, allant de 0,05 à 0,95, les valeurs du groupe S2 sont comprises entre 0 et 0,6 et les valeurs du groupe S3 sont comprises entre 0 et 0,3. La gamme de SVC est comprise entre 0.08 et 0.8 pour la date du 21/10/2019 et l'ensemble des valeurs de ζ pour un pourcentage de surface entre 0.08 et 0.6 est surestimé par la droite de régression. Pour la date du 28/10/2019 le SVC est compris entre 0 et 0.7, la majeure partie des points compris entre 0 et 0.2

sont supérieurs à la droite de régression, l'ensemble des points est inférieur à la droite pour des valeurs comprises entre 20 % et 70 %. Pour la date du 04/11/2019 la SVC est comprise entre 0 et 0.95 avec des valeurs de ζ sous estimées de 0.15 pour les conditions où la surface de plante occupe moins de 20 % du champ de vision, le ζ est surestimé par la droite de corrélation entre 20 % et 60 % puis sur-estimé entre 60 % et 100 %. Enfin, pour la date du 13/11/2019 le pourcentage de surface de plante vue par le capteur est compris entre 0.1 et 0.8, les points sont proches de la droite de régression pour des valeurs de SVC supérieures à 0.3 mais sont très dispersés (surestimation comme sous-estimation) pour des SVC inférieures à ce seuil.

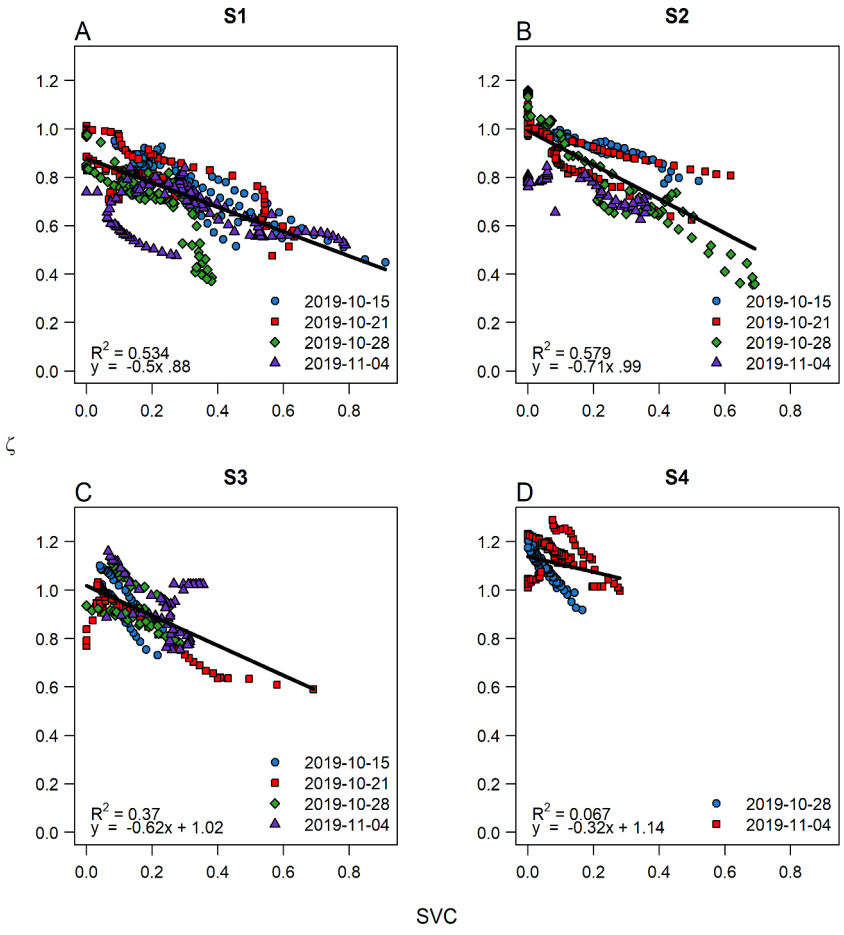


Figure 62: Correlation entre la SVC et le ζ mesuré en fonction de la date pour chaque groupe de semis.

Les données sont séparées par groupe de semis (Fig. 62). Pour le groupe de semis S1, les valeurs de SVC sont comprises entre 0.05 et 0.95, les valeurs de ζ mesurées sont équitablement dispersées au-dessus et au-dessous de la droite sur l'ensemble de la gamme de SVC.

4.2. Comparaisons entre les simulations obtenues avec les modèles directionnels et les mesures

Des simulations de ζ sont effectuées à l'aide des modèles de capteurs directionnel à partir des scans de plante effectuées lors des expérimentations ArchiMono et ArchiDuo. Les scans sont reconstruits à partir de 2 méthodes : la méthode « Ball Pivoting » et la méthode « Marching Cubes ».

4.2.1. Modélisation dans CARIBU avec la méthode de reconstruction Ball Pivoting

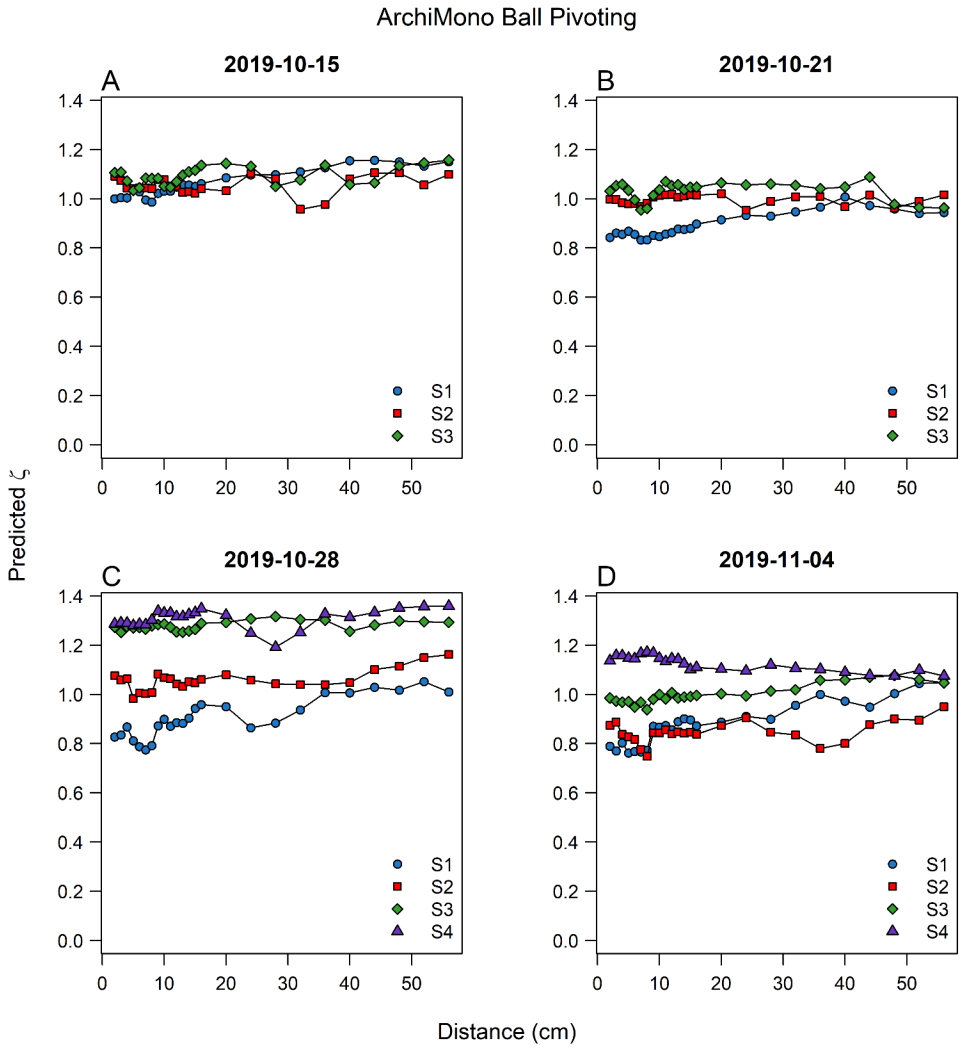


Figure 63: Evolution du ζ simulé en fonction de la distance pour chaque date de mesure. Les simulations sont effectuées avec les scans d'ArchiMono reconstruits par la méthode Ball Pivoting. Les données sont moyennées sur l'ensembles des plantes d'un même groupe de semis.

L'évolution du ζ simulé par CARIBU en fonction de la distance pour chacune des dates met en évidence des variations du ζ sur l'ensemble des distances (Fig. 63). Pour la première date, le ζ simulé est compris entre 1 et 1,2 pour les 3 groupes de semis, à l'exception du groupe S2 qui présente une brève diminution du ζ allant en dessous de 1 autour de 30 cm. Pour la seconde date, les ζ simulé pour les groupes de semis S2 et S3 est stable en fonction de la distance tandis que les valeurs de ζ simulées pour le groupe S1 augmentent en fonction de la distance, de plus une gradation nette de la valeur moyenne de ζ simulée par groupe de semis est observable. Cette séparation par groupe de semis est également observable à la 3e date de simulation, pour laquelle le ζ simulé à partir des scans de S1 est compris entre 0,8 et 1 et le ζ des groupes S3 et S4 est supérieur à 1,2 quelle que soit la distance. Cependant, pour la quatrième date, les simulations présentant les valeurs de ζ les plus faibles sont les simulations du groupe S2, une diminution du ζ en fonction de la distance peut également être observée pour le groupe S4.

La figure 64 montre l'évolution du ζ simulé par le modèle CARIBU en fonction de la distance à différente dates pour un même groupe de semis, à partir des scans d'ArchiMono reconstruits par la méthode Ball Pivoting. Le groupe de semis S1 montre une valeur de ζ plus élevée pour la première date de mesure, comprise entre 1,0 et 1,15. Les valeurs de ζ simulées pour les 3 autres dates du S1 sont proches les unes des autres pour l'ensemble des distances de simulation. Le ζ simulé pour le groupe S1 augmente en fonction de la distance pour l'ensemble des dates de simulation, cependant des diminutions ponctuelles de ζ sont observables pour les dates du 28-10-2019 et du 04-11-2019 avec respectivement une diminution du ζ à 24 cm et à 44 cm. Les valeurs de ζ mesurées pour le groupe de semis S4 sont supérieures à 1 pour les 2 dates de simulation et pour l'ensemble des distances, les valeurs simulées pour la date du 20-10-2019 sont supérieures à celles simulées pour la date du 04-11-2019, avec un écart de ζ proche de 0,3 à une distance de 56 cm.

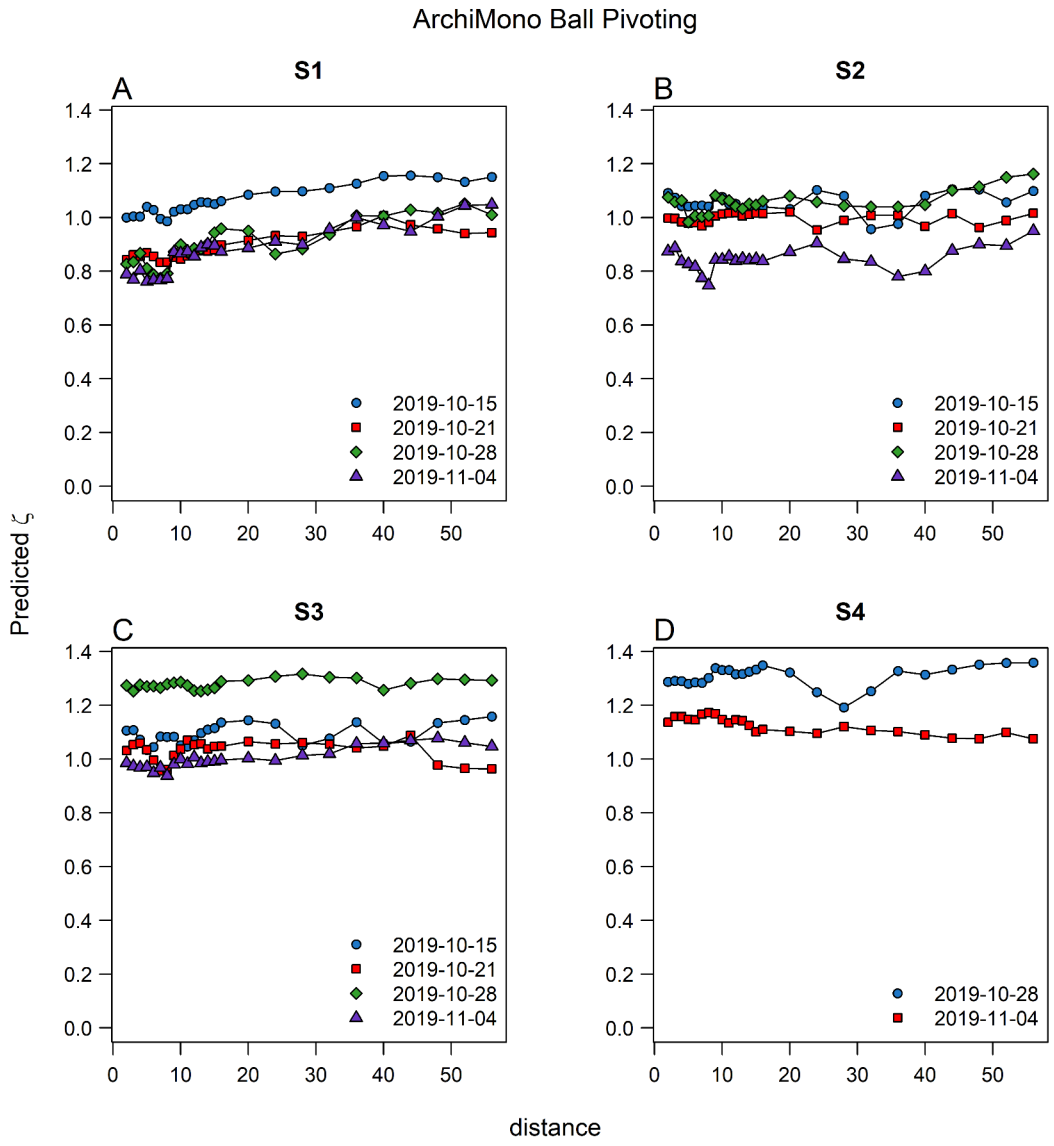


Figure 64: Evolution du ζ simulé en fonction de la date pour chaque groupe de semis.

Les données obtenues lors des expérimentations ArchiMono et ArchiDuo sont comparées aux simulations effectuées à partir des scans correspondant à ces mesures (Fig. 65, 67).

La comparaison entre le ζ moyen simulé par le modèle CARIBU et les mesures d'ArchiMono montrent que les valeurs simulées suivent les valeurs mesurées (Fig. 65). Une surestimation est également observable pour l'ensemble des points simulés. La gamme de ζ simulé est comprise entre 0,76 et 1,37 et la gamme de valeurs mesurées est comprise entre 0,47 et 1.22. L'écart maximal entre les valeurs prédites et les valeurs observées pour chaque mesure est compris entre 0,3 et 0,4.

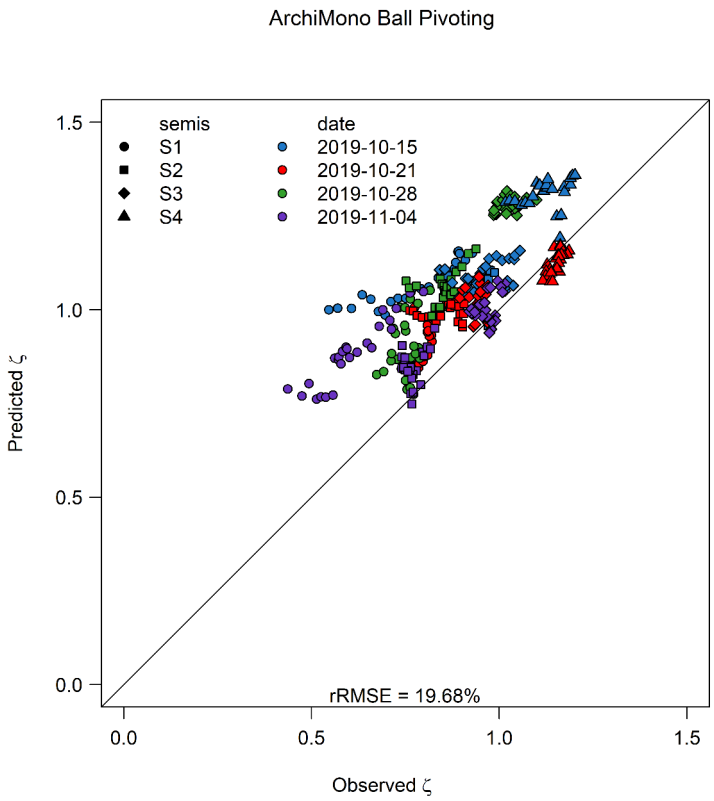


Figure 65: Comparaison entre le ζ mesuré et le ζ simulé pour les données d'ArchiMono. Les résultats sont moyennés par groupe et date de semis, ainsi chaque point correspond à une moyenne de 9 plantes.

En observant les simulations ArchiMono Ball Pivoting par date de mesure et pour chacune des plantes mesurées, un gradient de valeurs mesurées en fonction du stade de développement peut être observé à partir du 28/10/2019 (Fig. 66C). Pour les deux premières dates, le gradient de ζ entre les semis observés pour les mesures n'est pas observée pour les simulations pour lesquels les 3 semis ont des ζ simulés compris entre 0.5 et 1.2. À partir de la 3e date de mesure les plantes du semis S4 (peu développées) sont intégrées à la mesure, ces dernières présentent une valeur de ζ mesurée plus grande que les valeurs mesurées pour les plantes plus développées (S1 et S2). Pour le 28/10/19 un écart maximal de 0.5 peut être observé. Enfin, pour la date du 04/11/2019 la gamme de valeurs simulées est comprise entre 0.6 et 1.2 pour une gamme de valeurs mesurées comprises entre 0.5 et 1.3, les points correspondants aux valeurs de ζ inférieures à 1 sont majoritairement surestimées et les points correspondant aux valeurs de ζ mesurées supérieures à 1 sont sous estimées.

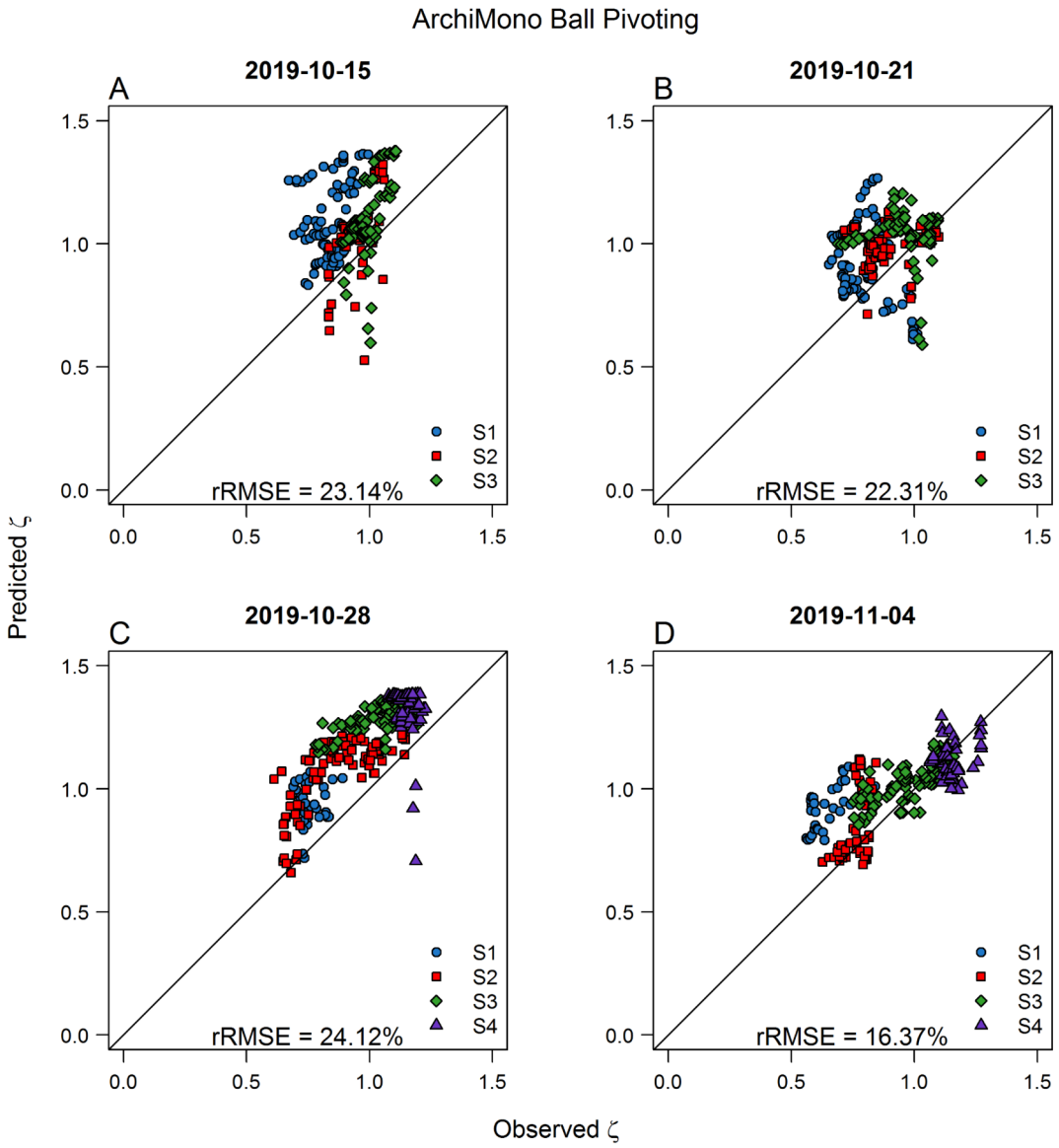


Figure 66: Comparaison entre le ζ mesuré et le ζ simulé pour les données d'ArchiMono en fonction de la date. Chaque point correspond à une mesure unique.

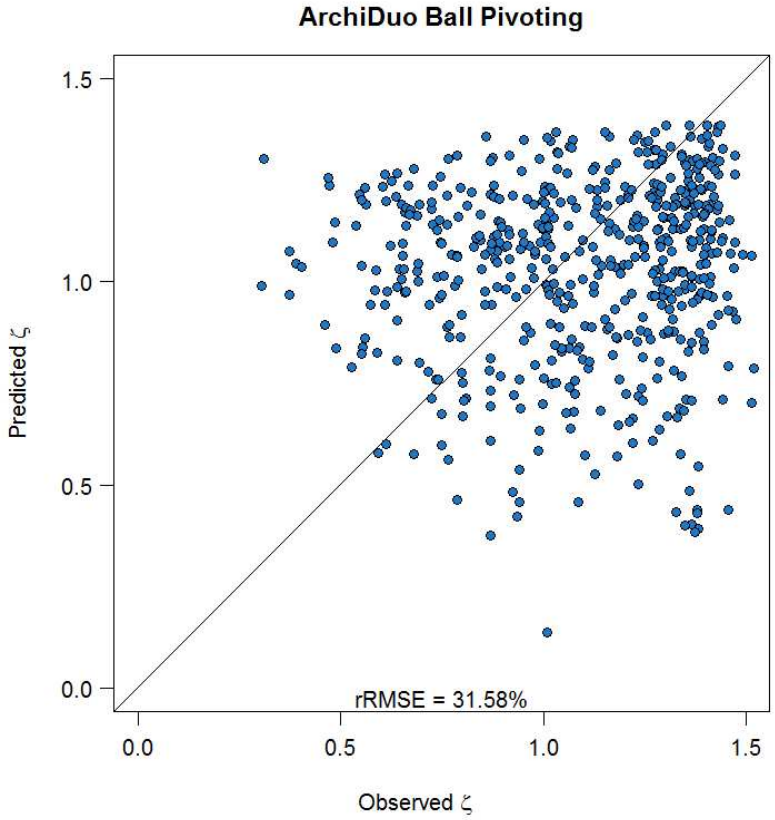


Figure 67: Comparaison entre les simulations et les mesures sur l'ensemble des scans reconstituées avec la méthode Ball Pivoting.

Les valeurs mesurées sont comprises entre 0.3 et 1.5 pour les simulations de CARIBU réalisées à partir des données d'ArchiDuo reconstruites avec la méthode Ball Pivoting (Fig. 67). Ces simulations montrent une large dispersion des points pour l'ensemble des conditions, avec des valeurs de ζ simulées très éloignées des mesures. Une symétrie de dispersion peut être observée avec à la fois des ζ simulés sous estimées pour des fortes valeurs de ζ mesurées, l'écart maximal correspond à une valeur simulée de 1.3 pour un ζ mesuré de 0.3, et des valeurs de ζ surestimées pour de faibles valeurs de ζ mesurées, l'écart maximal correspond à une valeur simulée de 0.4 pour un ζ mesuré proche de 1.5.

Comme précédemment observées, les points correspondants aux simulations de ζ à partir des données d'ArchiDuo sont très dispersés (Fig 68). Cette dispersion est présente sur l'ensemble des dates. Les dates où des relations entre les valeurs mesurées et les valeurs prédites ne se maintiennent pas sur l'ensemble des dates de mesure.

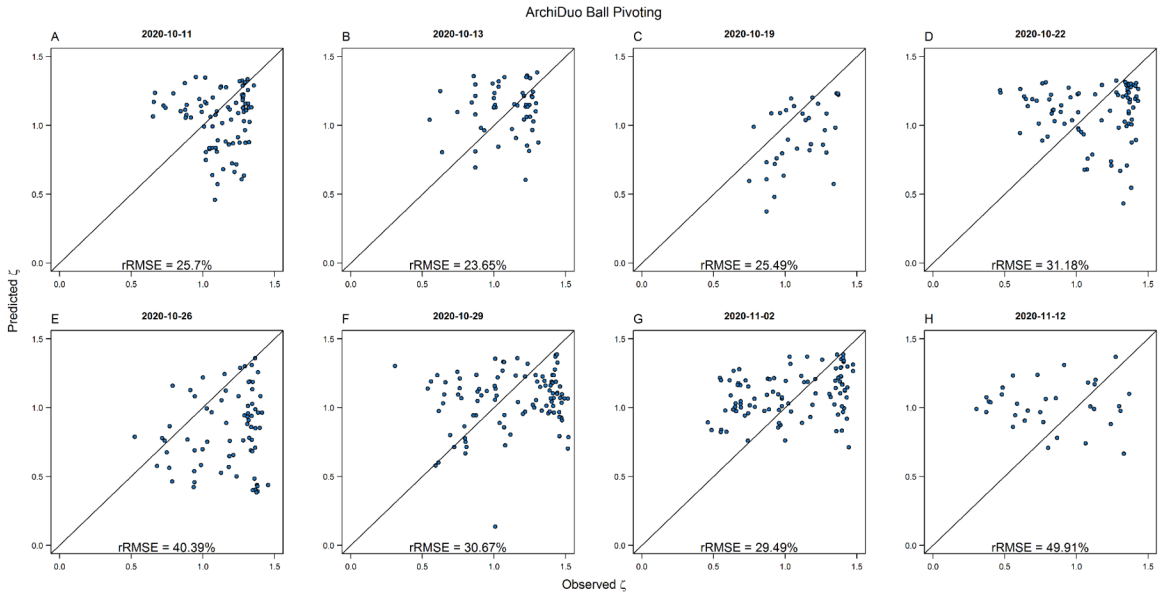


Figure 68: Comparaison des simulations avec les mesures à partir des scans d'ArchiDuo reconstruits avec la méthode Ball Pivoting.

4.2.2. Modélisation dans CARIBU avec la méthode de reconstruction Marching Cubes

L'évolution du ζ simulé par CARIBU pour les scans reconstruits avec la méthode Marching Cubes en fonction de la distance pour chacune des dates met en évidence des variations du ζ sur l'ensemble des distances (Fig. 69). Pour la première date, le ζ simulé diminue en fonction de la distance et est compris entre 0,7 et 1,1 pour les 3 groupes de semis. Pour la seconde date, la diminution du ζ en fonction de la distance est plus importante que pour la 1ere date et les valeurs de ζ sont comprises entre 1 et 0,6 pour les 3 groupes de semis. Une forte séparation par groupe de semis est observable pour la 3e date de mesure (Fig. 69C) et les courbes sont séparées par un gradient correspondant à un stade de développement, le groupe de S1 correspondant aux plantes les plus développées et le groupe S4 correspondant au groupe de plante le plus jeune. Contrairement aux deux dates précédentes, le ζ simulé pour la 3e date augmente en fonction de la distance. Enfin les résultats obtenus pour la 4e date sont également séparés par date de semis, le groupe S1 ayant les valeurs de ζ les plus faibles pour la majorité des distances et le groupe S4 ayant les valeurs les plus élevées. Sur cette dernière date de simulation le ζ diminue lorsque la distance augmente.

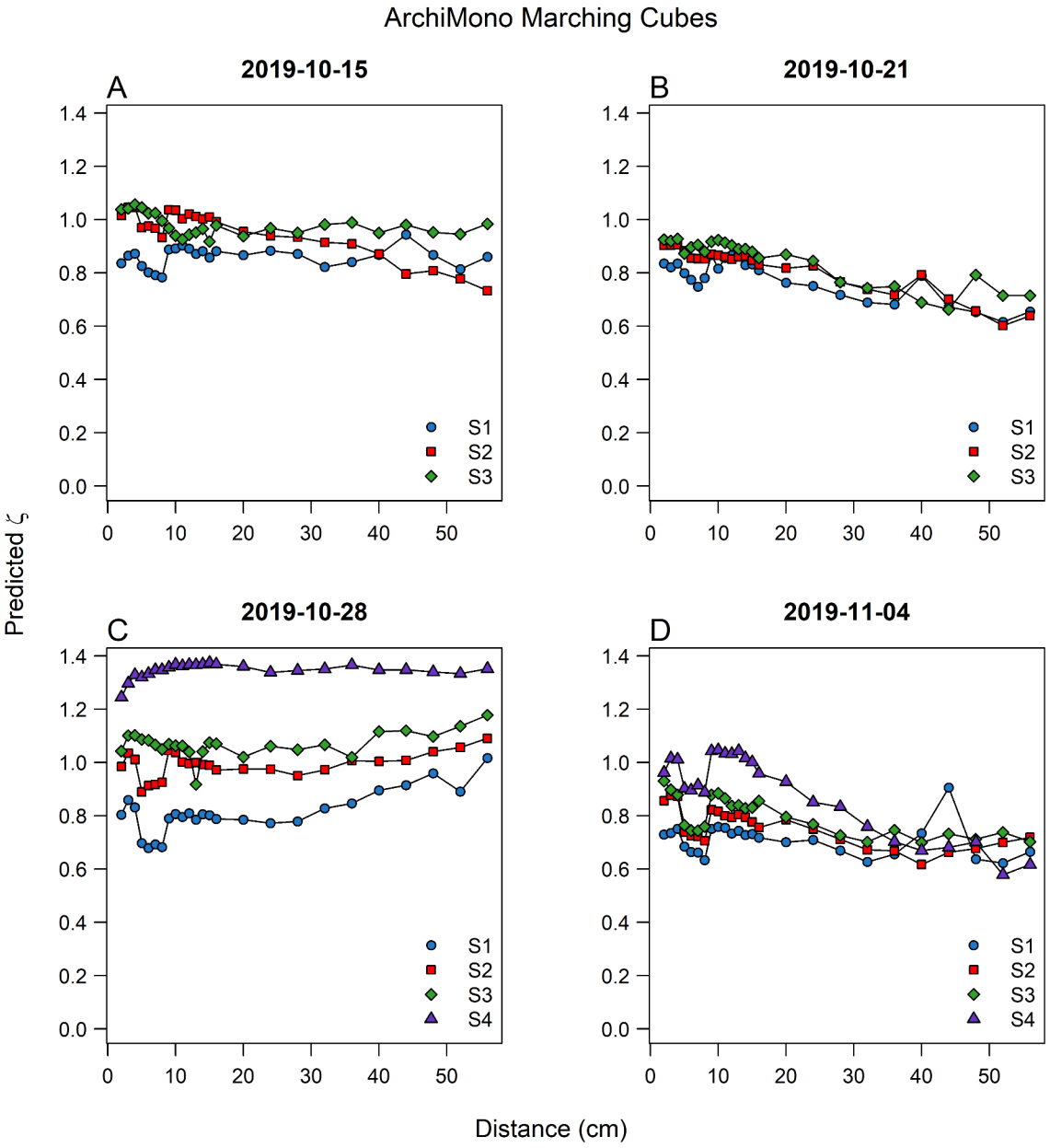


Figure 69: Evolution du ζ simulé en fonction de la distance pour chaque date de mesure. Les simulations sont effectuées avec les scans d'ArchiMono reconstruits par la méthode Marching Cubes. Les données sont moyennées sur l'ensemble des plantes d'un même groupe de semis.

Les simulations de ζ en fonction de la distance pour les simulations effectuées avec la méthode de reconstruction Marching Cubes sont séparées par groupe de semis (Fig. 70). Pour les 4 groupes de semis on peut observer une diminution du ζ en fonction de la date, à l'exception de la date « 2019-10-28 » pour laquelle les valeurs de ζ sont plus élevées que l'ensemble des ζ simulés pour les autres dates pour les groupes de semis S2, S3 et S4 et dépasse le ζ simulé pour la première

date à partir de 30 cm pour le groupe S1. Le ζ simulé est compris entre 0.6 et 1 pour S1 , entre 0,6 et 1,1 pour S2, entre 0,6 et 1,2 pour S3 et entre 0,5 et 1,4 pour S4.

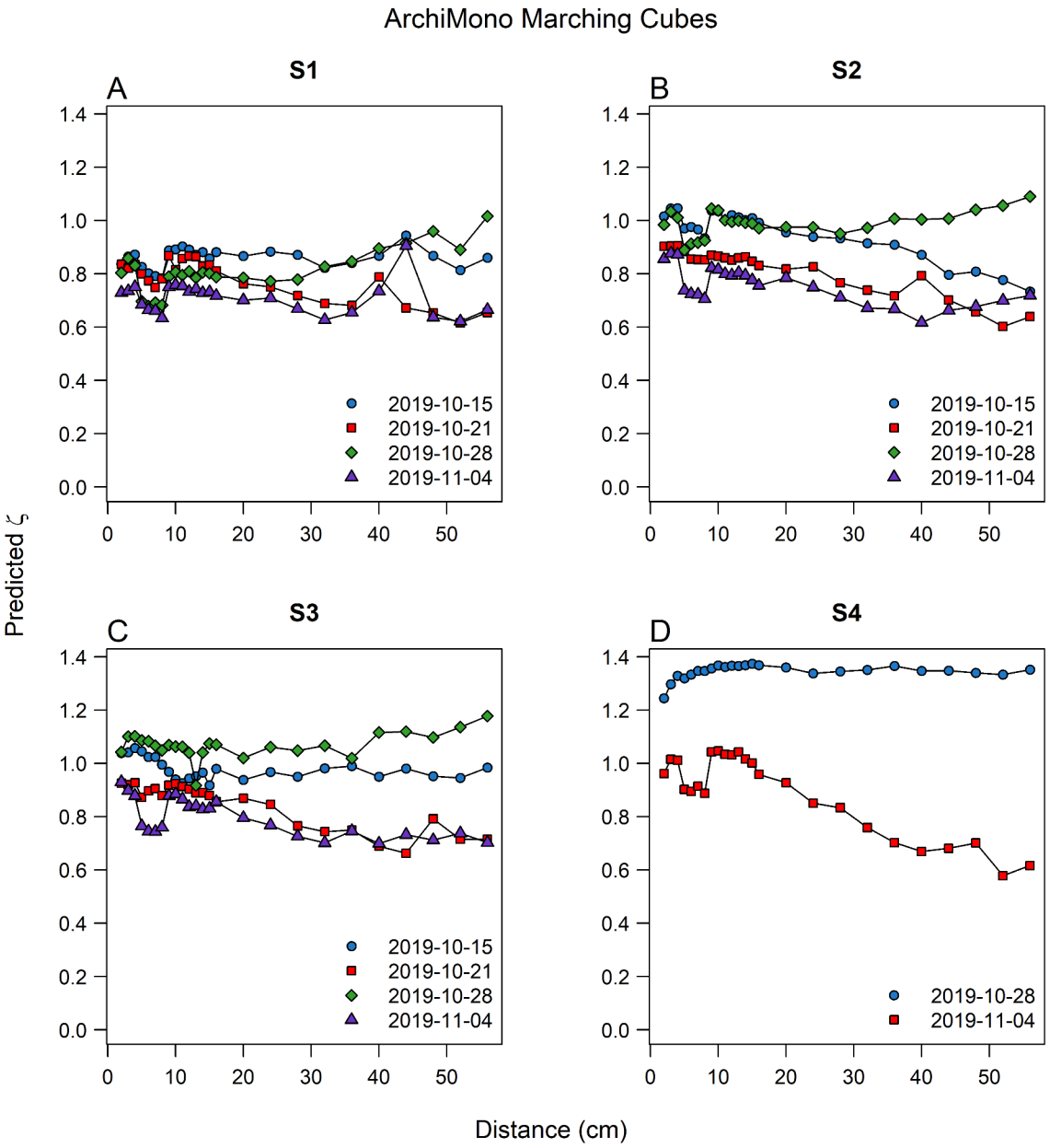


Figure 70: Evolution du ζ simulé pour chaque groupe de semis aux différentes dates de mesure. Les simulations sont effectuées avec les scans d'ArchiMono reconstruits par la méthode Marching Cubes. Les données sont moyennées sur l'ensembles des plantes d'un même groupe de semis.

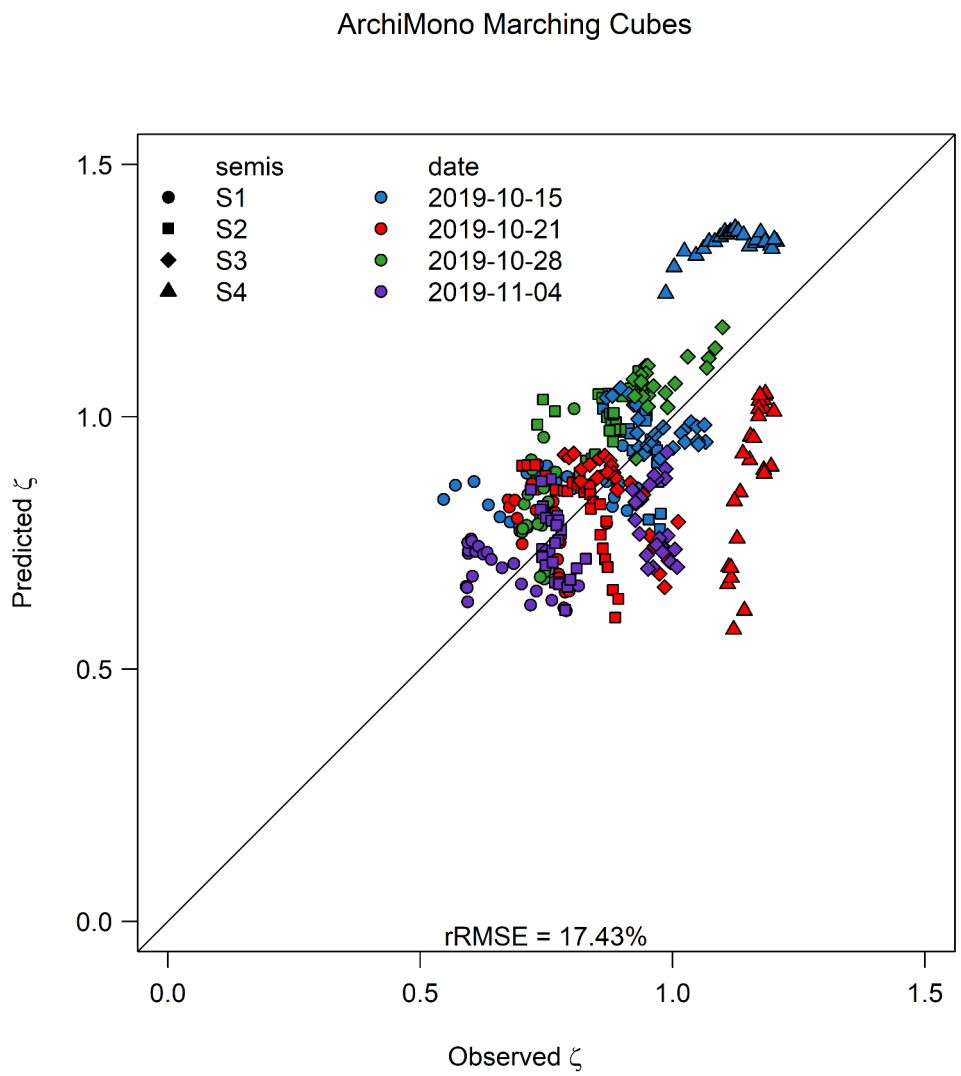


Figure 71: Comparaison entre le ζ mesuré et le ζ simulé par CARIBU pour les données d'ArchiMono. Les scans sont reconstruits avec la méthode Marching Cubes. Les résultats sont moyennés par groupe et date de semis, ainsi chaque point correspond à une moyenne de 9 plantes.

La comparaison entre les simulations effectuées pour ArchiMono avec la méthode de reconstruction Marching Cubes et les mesure (Fig. 71) présente des valeurs de ζ comprises entre 0.3 et 1.4 pour les valeurs prédites. Les points sont équitablement sur-estimées et sous-estimées, la sous-estimation est cependant plus importante que la surestimation : le plus grand écart entre le ζ mesuré et le ζ simulé est de 0.2 pour les valeurs surestimées et de 0.4 pour les valeurs sous-estimées.

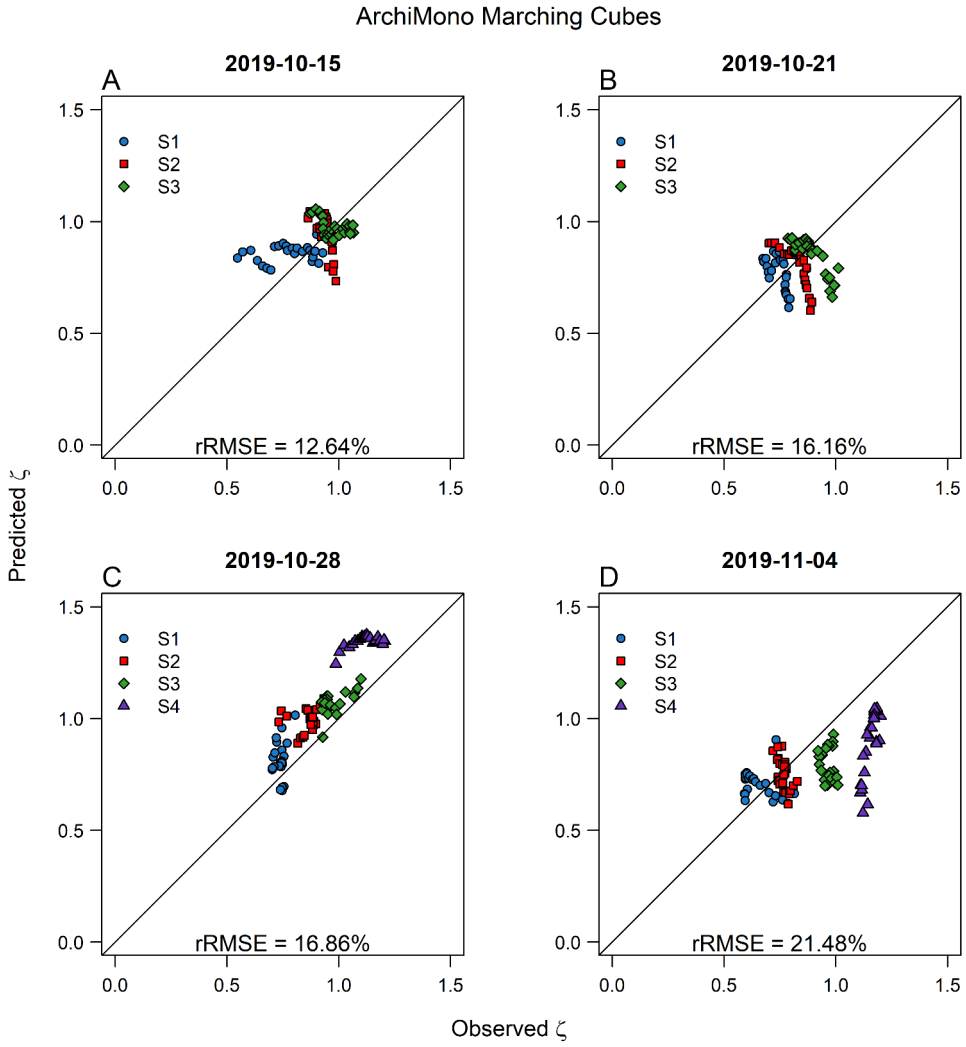


Figure 72: Comparaison entre le ζ mesuré et le ζ simulé pour les données d'ArchiMono à partir des scans reconstruits par la méthode Marching Cubes.

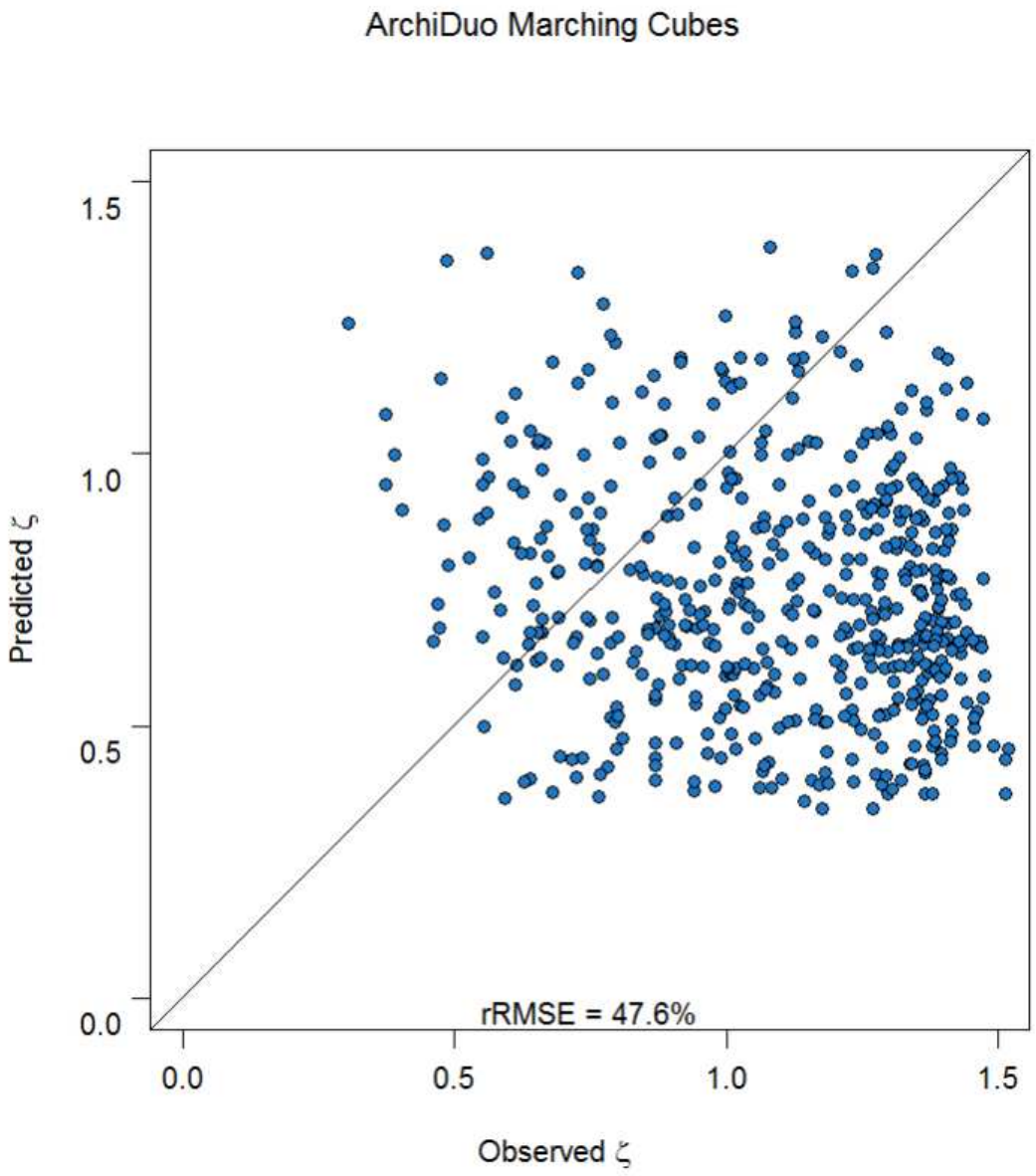


Figure 73: Comparaison entre les simulations et les mesures sur l'ensemble des scans reconstituées avec la méthode Marching Cubes.

Les résultats de simulation effectuées avec la méthode de reconstruction Marching Cubes présentent une forte dispersion des résultats pour les jeux de données d'ArchiDuo (Fig. 73). Les simulations présentent une forte sous-estimation des valeurs de ζ. En particulier pour un ensemble de points dont la mesure est supérieure à 1.3 et dont les valeurs simulées sont distribuées sur une gamme de ζ comprise entre 1.2 et 0.4.

En séparant les résultats d'ArchiMono par date, les simulations effectuées pour les 2 premières dates pour la reconstruction Marching Cubes ont une forte dispersion du ζ simulé par rapport au ζ mesuré, cependant pour la date du 20/10/2019 le ζ simulé suit les résultats des mesures.

Pour les simulations effectuées avec les données d'ArchiDuo, la dispersion des points est semblable entre les différentes dates. Pour la plupart des dates, la dispersion des points est symétrique entre les sur-estimations et les sous-estimations (Fig. 74). Cependant, pour les dates du 2020-10-19 et du 2020-10-26 les résultats de simulation sous-estiment en majorité les mesures.

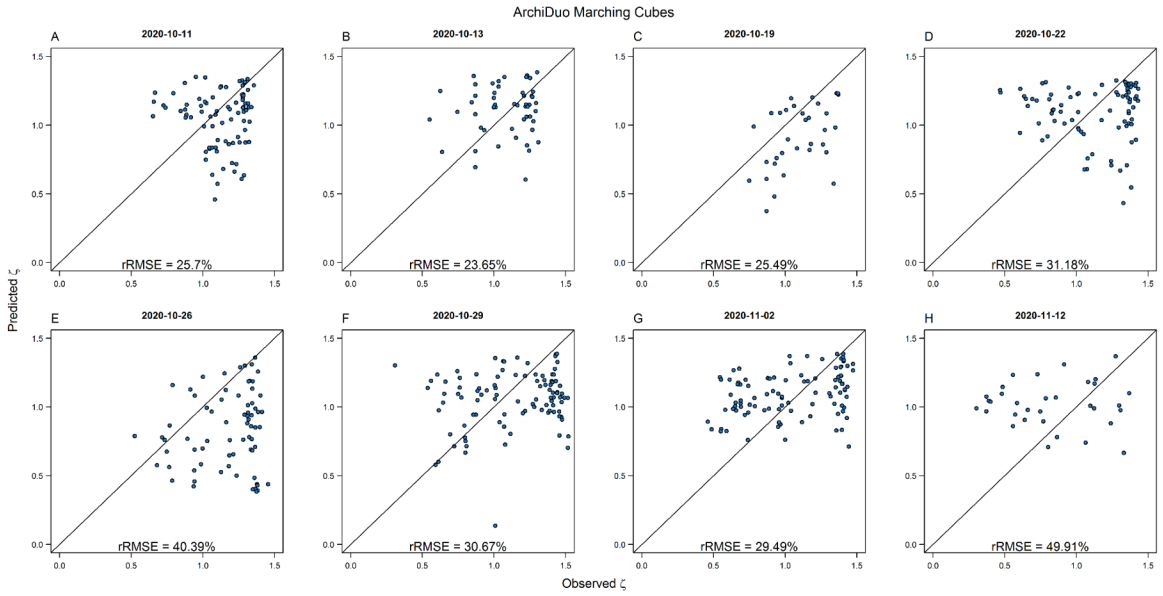


Figure 74: Comparaison entre les simulations et les mesures sur l'ensemble des scans reconstitués avec la méthode Marchin Cubes et pour chacune des sates de simulation.

4.2.3. Modélisation dans MALIA avec la méthode de reconstruction Ball Pivoting

Les résultats de simulation du ζ par MALIA en fonction de la distance pour les scans d'ArchiMono reconstruits avec la méthode Ball Pivoting pour chacune des dates sont présentés en figure 75. Pour l'ensemble des dates, peu de variations de ζ peuvent être observées, les valeurs de ζ augmentent sur les distances comprises entre 0 et 20 cm avant d'atteindre un plateau. Pour la première date, le ζ simulé est compris entre 0,8 et 1,1 pour les 3 groupes de semis. Les valeurs de ζ

simulées sont comprises entre 1 et 1,1 pour la seconde date de simulation, entre 0,9 et 1,1 pour la 3e date et entre 0,8 et 1,1 pour la 4e date de simulation.

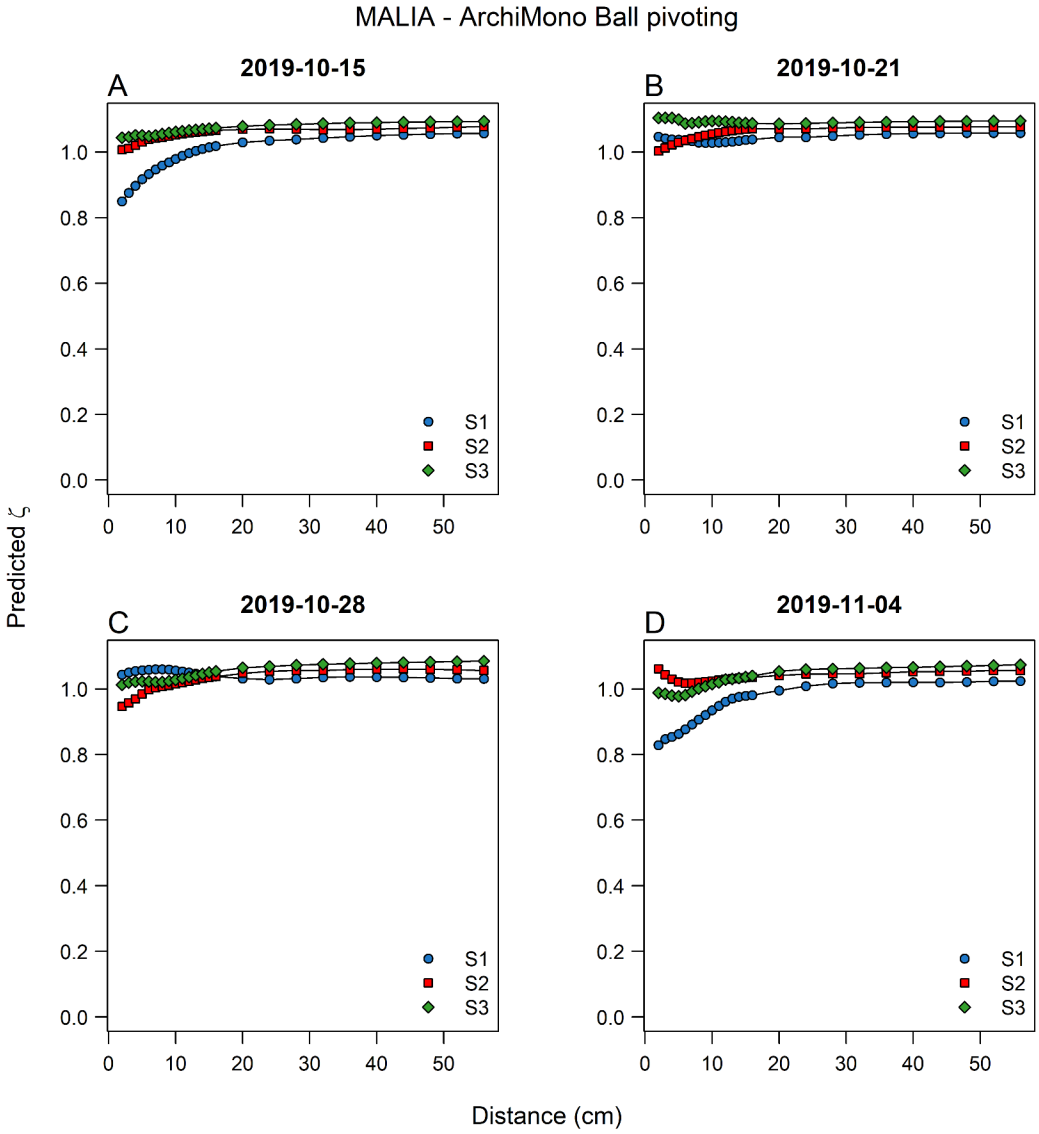


Figure 75: Evolution du ζ simulé par MALIA en fonction de la distance pour chaque date de mesure. Les simulations sont effectuées avec les scans d'ArchiMono reconstruits par la méthode Ball Pivoting. Les données sont moyennées sur l'ensemble des plantes d'un même groupe de semis.

La comparaison entre les simulations pour les données d'ArchiMono Ball Pivoting met en évidence une surestimation des simulations pour l'ensemble des mesures pour lesquelles le ζ est inférieur à 1 (Fig. 76). La gamme des résultats de simulations obtenue à partir des scans d'ArchiMono est comprise entre 0.5 et 1. Les simulations effectuées surestiment majoritairement le ζ, de plus les résultats suivent une pente plus faible que la droite 1:1. Un seuil de valeur maximale de simulation est observé autour de 1,1.

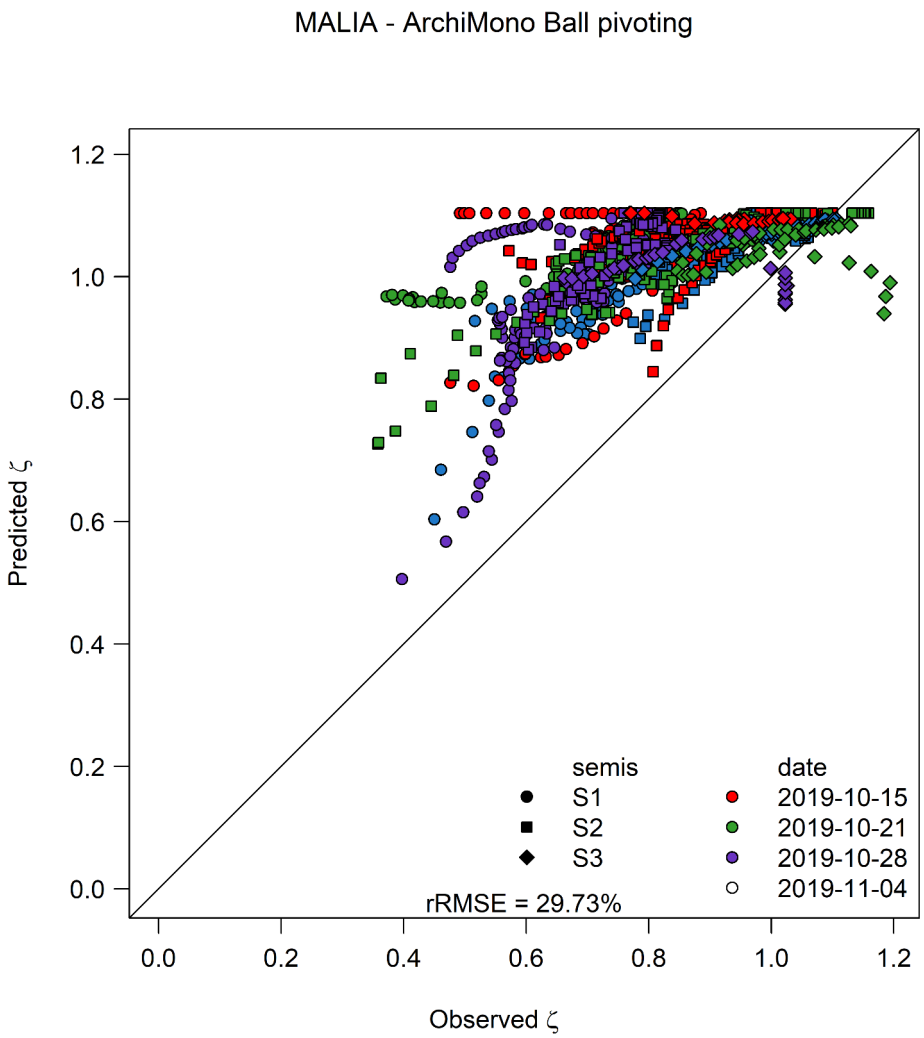


Figure 76: Comparaison entre le ζ mesuré et le ζ simulé par MALIA pour les données d'ArchiMono. Les scans sont reconstruits avec la méthode Ball Pivoting. Les résultats sont moyennés par groupe et date de semis, ainsi chaque point correspond à une moyenne de 9 plantes.

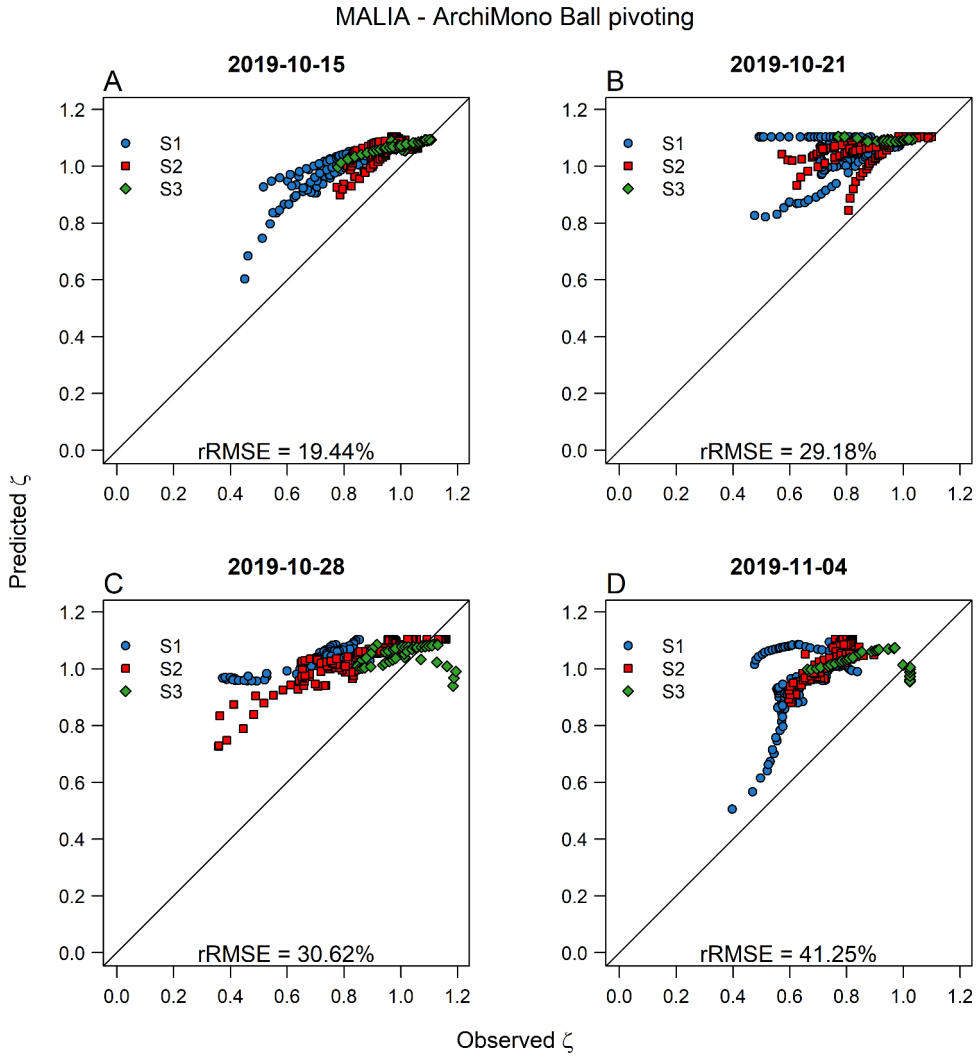


Figure 77: Comparaison entre le ζ mesuré et le ζ simulé par le modèle MALIA pour chaque date de mesure. Les simulations sont effectuées à partir des scans d'ArchiMono reconstruits par la méthode Ball Pivoting. Chaque point correspond à une simulation.

La comparaison entre le ζ mesuré et le ζ simulé par MALIA pour les scans d'ArchiMono reconstruit avec la méthode Ball Pivoting (Fig. 77) montre qu'en fonction des dates de simulations les valeurs de ζ suivent la même dynamique au sein d'une même date. Le seuil de valeur de ζ simulé est uniquement observable pour la 2^{e} date de simulation. Les valeurs de rRMSE montrent que la qualité de simulation par rapport aux mesures diminue en fonction de la date de simulation. Pour les trois premières dates de mesures l'écart entre les valeurs de ζ simulés et les valeurs de ζ mesurés augmente lorsque le ζ mesuré diminue. Une pente est observable pour chacune des conditions. Pour la date du 2019-10-15, le ζ simulé est compris entre 0.6 et 1,1 et les points simulés suivent les valeurs mesurées malgré une surestimation, avec un écart maximal entre la valeur

simulée et la valeur mesurée de 0.35.Pour la date du 2019-10-21, le ζ simulé est compris entre 0.8 et 1,1, avec un écart maximal entre la valeur simulée et la valeur mesurée de 0.55. Pour la date du 2019-10-28 le ζ est surestimé pour les mesures inférieures à 1 et sous-estimé pour les mesures supérieures à 1, le ζ simulé est compris entre 0.7 et 1,1, avec un écart maximal entre le ζ simulé et le ζ mesuré de 0.5. Enfin pour la date du 2019-11-04 une partie des simulations suit de près les mesures, la gamme des simulations est comprise entre 0.5 et 1,1 avec un écart maximal de 0.45.

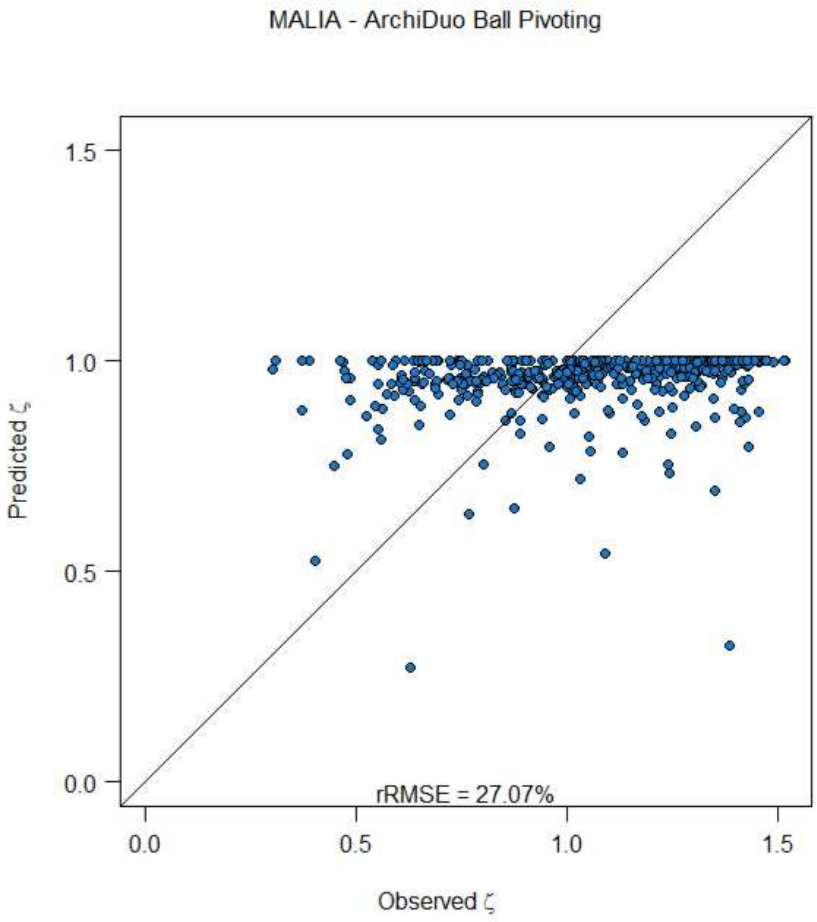


Figure 78: Comparaison entre les simulations effectuées avec le modèle MALIA et les mesures sur l'ensemble des scans d'ArchiDuo reconstruits avec la méthode Ball Pivoting.

Pour les simulations réalisées par MALIA à partir des scans d'ArchiDuo reconstruits avec la méthode Ball Pivoting, les valeurs de ζ sont plafonnées à une valeur de ζ de 1 (Fig. 78). La gamme de ζ simulée est comprise entre 0.3 et 1, cependant la majorité des simulations est comprise entre 0.7 et 1. Pour ces simulations, le ζ simulé est surestimé pour les valeurs mesurées inférieures à 1, avec un écart maximal de 0.7, et surestimées pour les valeurs mesurées supérieures à 1, avec un écart maximal de 1.1.

4.2.4. Modélisation dans MALIA avec la méthode de reconstruction Marching Cubes

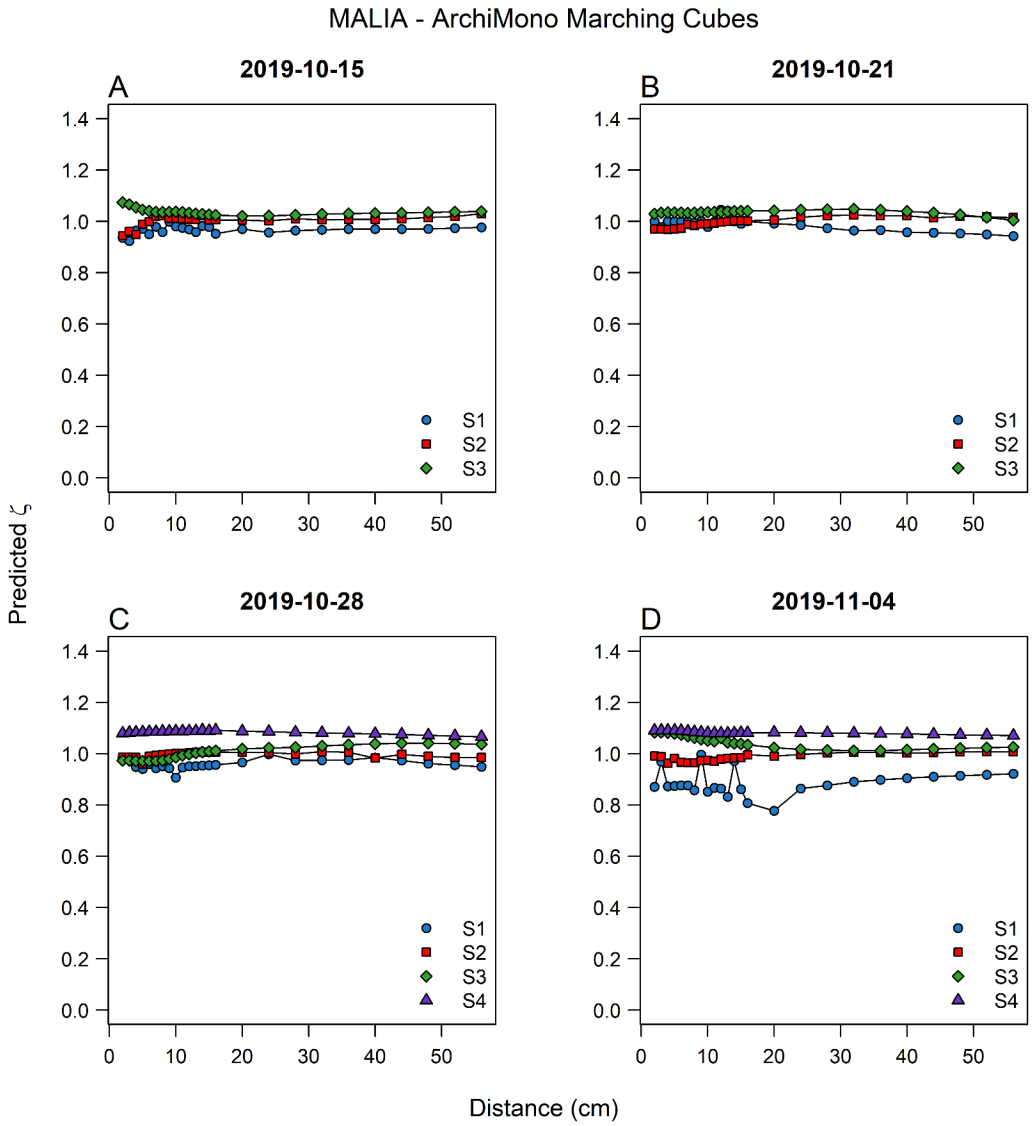


Figure 79: Evolution du ζ simulé par le modèle MALIA en fonction de la distance pour chaque groupe de semis et par date. Les simulations sont effectuées avec les scans d'ArchiMono reconstruit avec la méthode Marching Cubes.

Les résultats de simulation du ζ par MALIA en fonction de la distance pour les scans d'ArchiMono reconstruits avec la méthode Marching Cubes pour chacune des dates sont présentés en figure 79. Comme pour les simulations effectuées avec la méthode de reconstitution Ball Pivoting, peu de variations de ζ peuvent être observées en fonction de la distance pour la majorité des simulations. Les valeurs de ζ sont varient peu quelle que soit la distance pour chaque groupe de semis et pour chaque date de simulation. Pour la première date, le ζ simulé est compris entre 0,9 et

1,1 pour les 3 groupes de semis. Les valeurs de ζ simulées sont comprises entre 0,95 et 1,05 pour la seconde date de simulation, entre 0,9 et 1,1 pour la 3e date et entre 0,8 et 1,1 pour la 4e date de simulation. Des variations de ζ sont à noter pour la dernière date de simulation pour le groupe de semis S1, pour lequel le ζ simulé diminue sur les 20 premiers centimètres avant d'augmenter et d'arriver à un plateau.

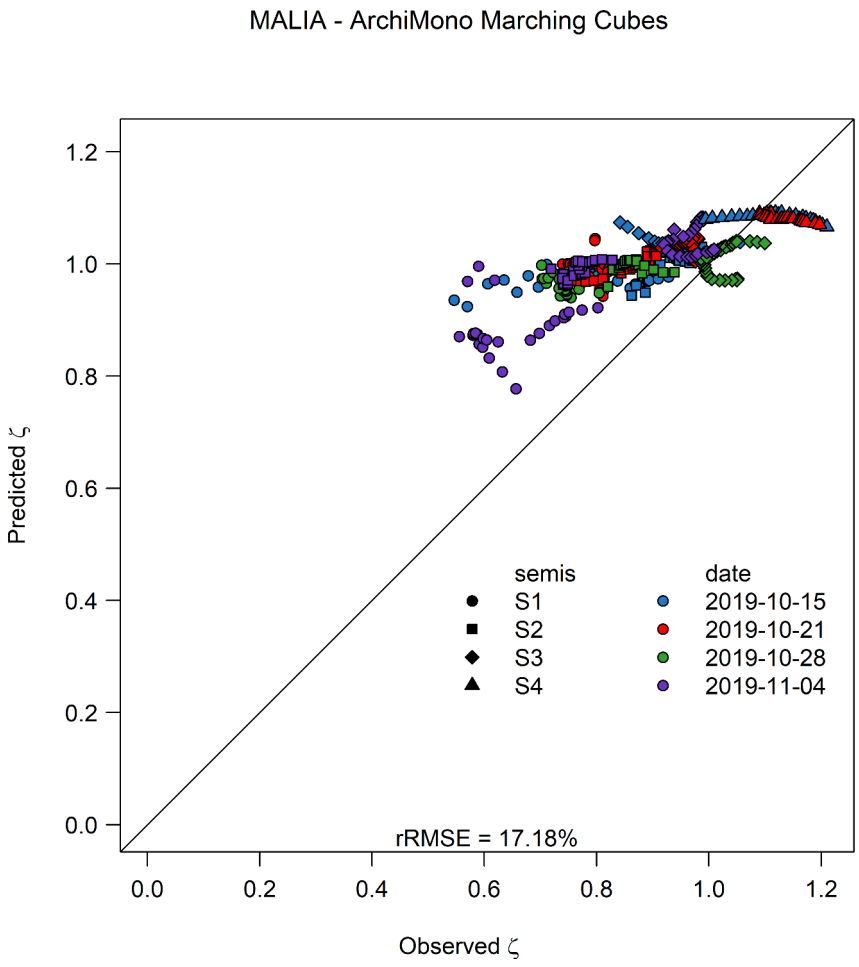


Figure 80: Comparaison entre le ζ mesuré et le ζ simulé par MALIA pour les données d'ArchiMono. Les scans sont reconstruits avec la méthode Marching Cubes. Les résultats sont moyennés par groupe et date de semis, ainsi chaque point correspond à une moyenne de 9 plantes.

La comparaison entre les simulations pour les données d'ArchiMono Marching Cubes met en évidence une surestimation des simulations pour l'ensemble des mesures pour lesquelles le ζ est inférieur à 1 (Fig. 80). La gamme des résultats de simulations obtenue à partir des scans d'ArchiMono est comprise entre 0.5 et 1. Les simulations effectuées surestiment majoritairement le

ζ, de plus les résultats suivent une pente plus faible que la droite 1:1. Un seuil de valeur maximale de simulation est observé autour de 1,1.

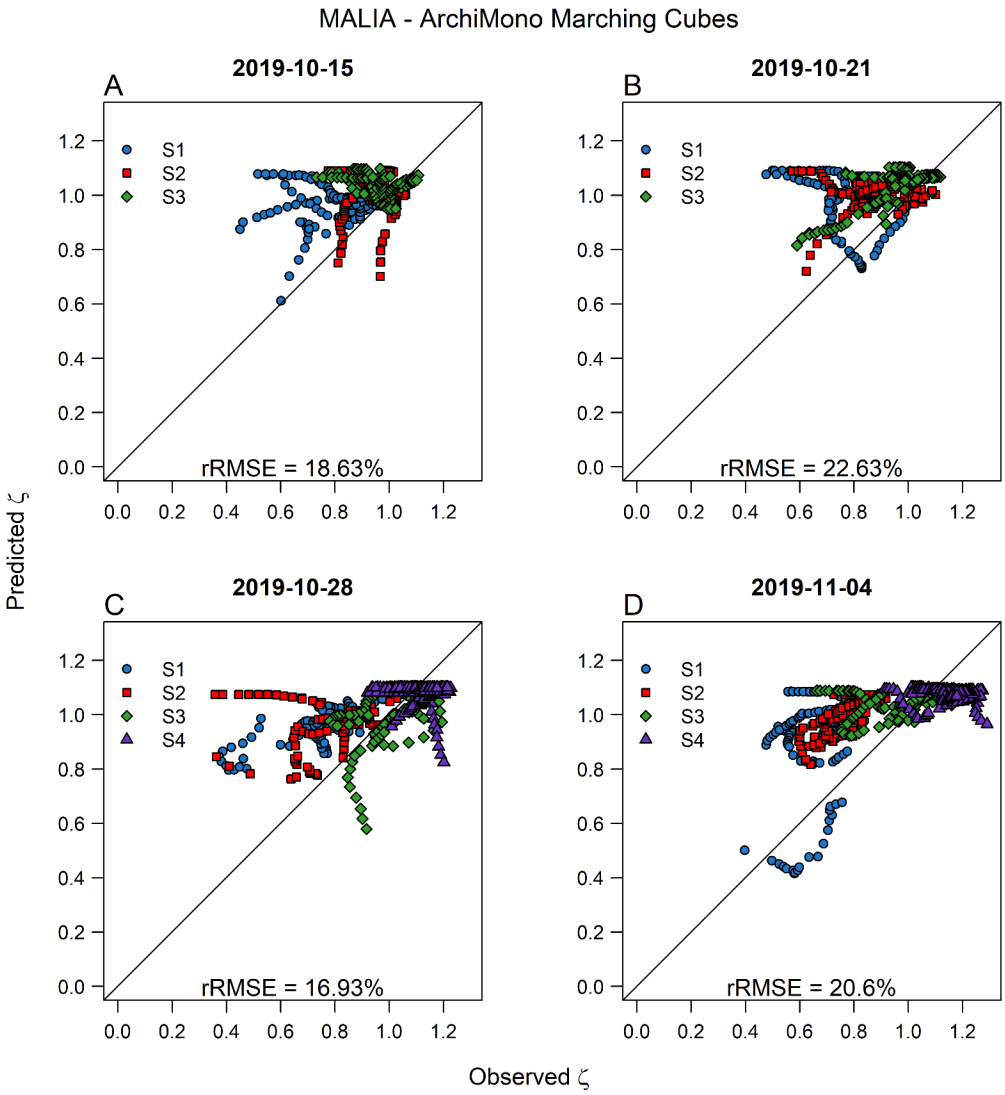


Figure 81: Comparaison entre le ζ mesuré et le ζ simulé pour les données d'ArchiMono en fonction de la date. Chaque point correspond à une mesure unique.

Séparation par date des comparaisons entre le ζ mesuré et le ζ simulé par MALIA pour les scans d'ArchiMono reconstruits avec la méthode Ball Pivoting (Fig. 81) montre une forte dispersion des mesures pour chacune des dates. Un seuil de valeur de ζ est observable sur les 4 dates de simulations. La dispersion des points entre les simulations et les mesures est très variable en fonction des dates. Pour la date du 2019-10-15 (Fig. 81A), le ζ simulé est compris entre 0,6 et 1,1 et les points simulés suivent les valeurs mesurées malgré une surestimation, avec un écart maximal entre la valeur simulée et la valeur mesurée de 0.6. Pour la date du 2019-10-21 (Fig. 81B), le ζ

simulé est compris entre 0,65 et 1,1, avec un écart maximal entre la valeur simulée et la valeur mesurée de 0,8. Pour la date du 2019-10-28 (Fig. 81C) le ζ est surestimé pour les mesures inférieures à 1 et sous-estimé pour les mesures supérieures à 1, le ζ simulé est compris entre 0,55 et 1,1, avec un écart maximal entre le ζ simulé et le ζ mesuré de 0,7. Enfin pour la date du 2019-11-04 (Fig. 81D) une partie des simulations suit de près les mesures, la gamme des simulations est comprise entre 0,4 et 1,1 avec un écart maximal de 0,45.

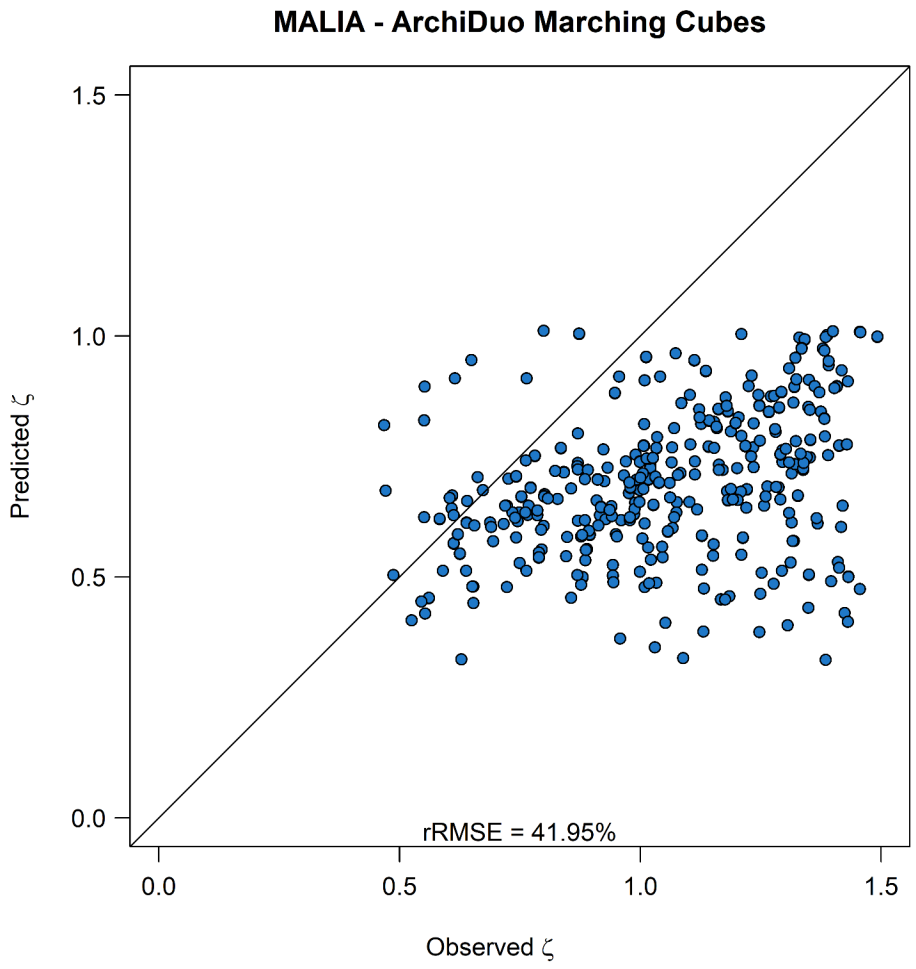


Figure 82: Comparaison entre les simulations effectuées avec le modèle MALIA et les mesures sur l'ensemble des scans d'ArchiDuo reconstruits avec la méthode Marching Cubes.

Les simulations effectuées avec le modèle MALIA à partir des scans reconstruits avec la méthode Marching Cubes sont comparées aux valeurs mesurées (Fig. 82). Comme précédemment observé avec la reconstruction Ball Pivoting, les simulations effectuées par le modèle Malia sont plafonnées à 1.0.

Les simulations réalisées à partir des scans d'ArchiMono ont des ζ simulés compris entre 0.37 et 1, avec une majorité de points compris entre 0.6 et 1. Les simulations réalisées à partir des scans d'ArchiDuo ont des ζ simulés compris entre 0.33 et 1. La quasi-totalité des points présente une sous-estimation de la valeur de ζ simulée. En particulier, une forte sous-estimation peut être observée pour les valeurs de ζ mesurées supérieures à 1, avec un écart de valeur de ζ allant jusqu'à 1.1.

5. Application de diverses conditions à la modélisation de la mesure du rayonnement directionnel

Des simulations de l'évolution du ζ journalier en condition de ciel clair et de ciel couvert ont été réalisées avec le modèle CARIBU pour des plantes isolées.

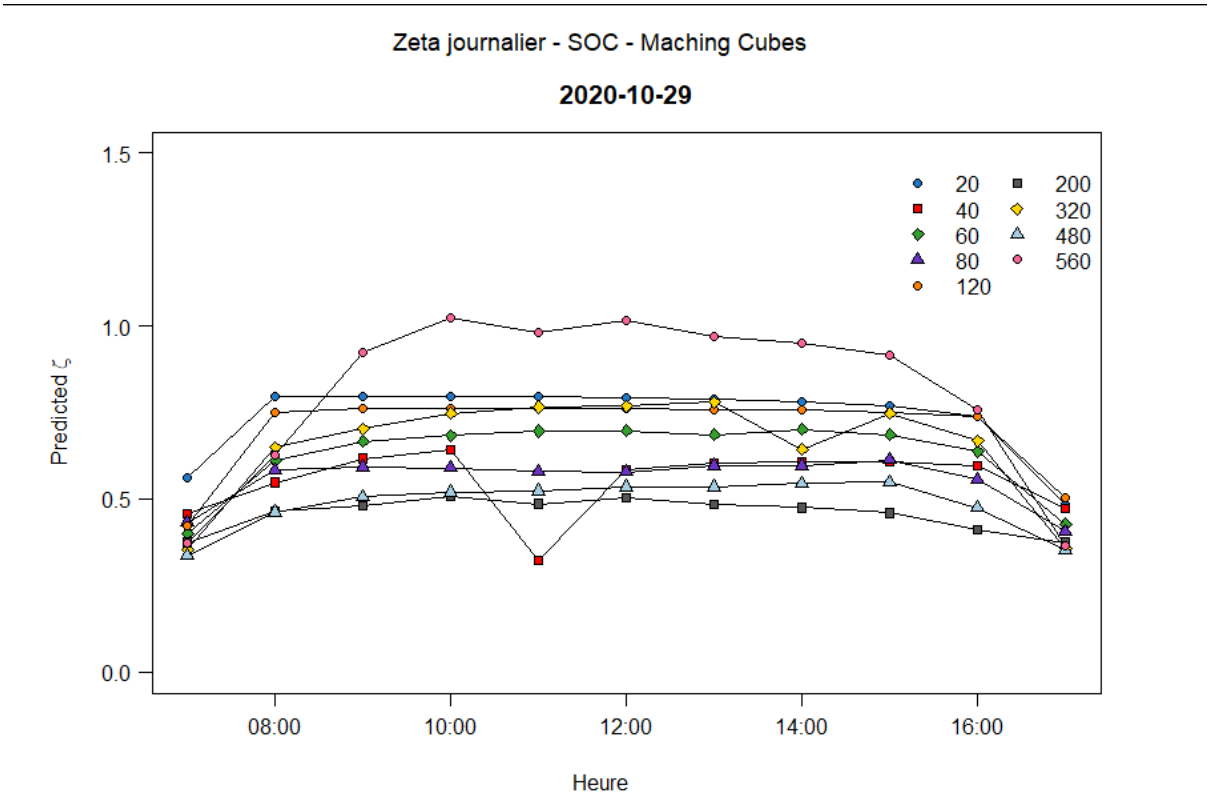


Figure 83: Modélisation de l'évolution du ζ au cours de la journée par le modèle CARIBU pour des conditions de ciel clair.

Les simulations sont effectuées avec les scans de plantes d'ArchiDuo, où chaque distance correspond à une plante différente. Les simulations faites avec le modèle de ciel clair SOC présentent pour les différentes distances de mesure différentes gammes de variations au cours du temps. Pour chacune d'entre elles, le ζ est faible à 7 h UTC, puis le ζ augmente jusqu'à 12 h UTC avant de redescendre pour atteindre une valeur basse à 17 h UTC proche de celle obtenue à 7 h UTC. Les plus grandes variations sont observées pour la distance 560 mm, pour laquelle le ζ est compris entre 0.37 et 1.05.

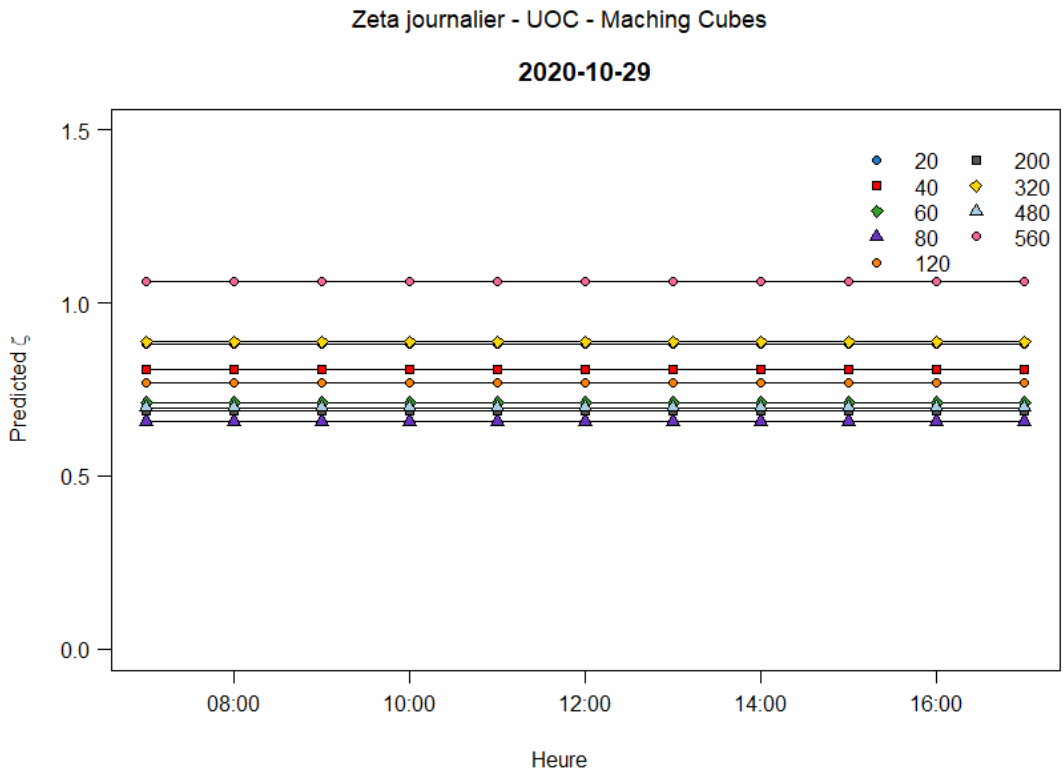


Figure 84: Modélisation de l'évolution du ζ au cours de la journée par le modèle CARIBU pour des conditions de ciel couvert.

A l'inverse des simulations effectuées avec le modèle de ciel SOC, les simulations réalisées avec le modèle de ciel UOC présentent un ζ ne variant pas au cours du temps pour l'ensemble des plantes et des distances (Fig. 84).

Des simulations avec CARIBU ont également été réalisées à l'aide d'un peuplement précoce virtuel reproduisant les peuplements utilisés pour les mesures journalières (Fig. 85). Ces simulations ont été faites avec les modèles de ciel SOC et UOC. Les valeurs de ζ obtenues sont constantes au cours du temps pour les 2 modèles de ciel utilisés et seules 2 diminutions sont observables à 11 h et 13 h.

Aucune simulation avec des peuplements virtuels développés n'a été effectuée car le temps de simulation était trop important (plusieurs semaines).

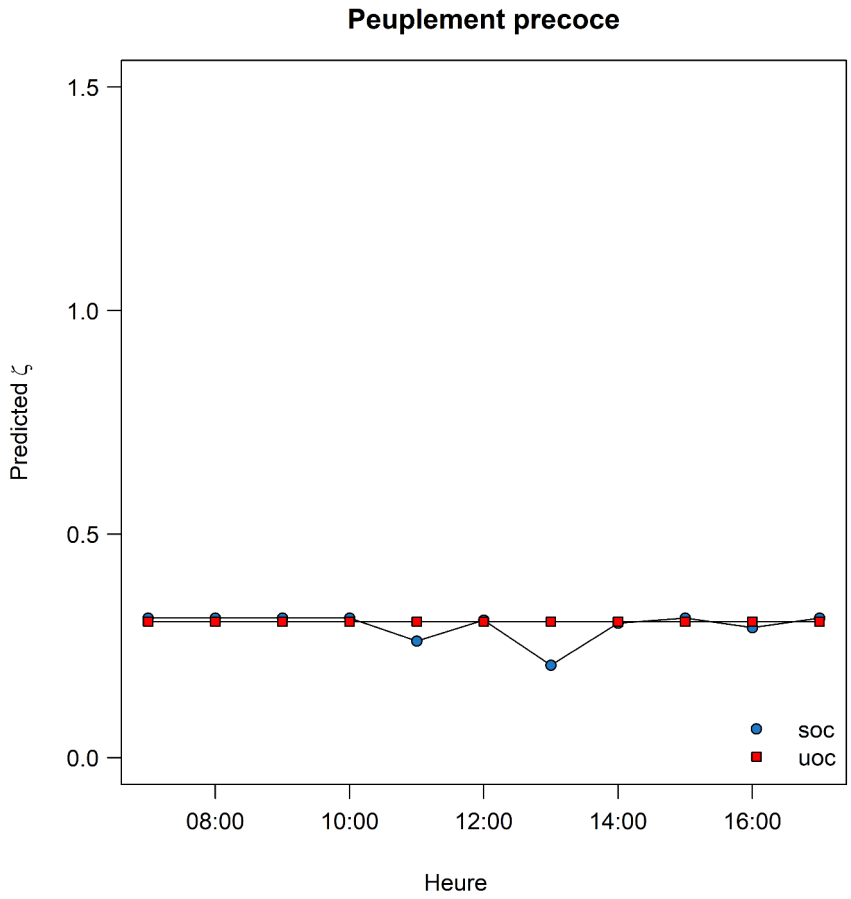


Figure 85: Modélisation de l'évolution du ζ au cours de la journée par le modèle CARIBU pour des conditions de ciel couvert pour un peuplement au stade précoce (1 talle).

V. Discussion

1. Introduction

Jusqu'à présent, les lois de réponses établies entre des niveaux de ζ mesurés et des réponses morphogénétiques observées ont été établies à partir de mesures hémisphériques de l'environnement lumineux. Cette mesure hémisphérique a pour effet de moyenner l'ensemble des interactions entre une plante et son environnement lumineux en ignorant l'aspect architectural et directionnel dans les interactions plante-environnement. Or différentes études ont clairement mis en évidence un effet de la direction du signal lumineux sur l'orientation des organes (Maddonni *et al.* 2002; Smith *et al.* 2020). De plus, des hypothèses ont été formulées sur la capacité des plantes à distinguer les plantes voisines apparentées des plantes non-apparentées en fonction du profil de lumière résultant de l'architecture des plantes (Crepy and Casal 2015). L'information directionnelle est donc cruciale dans la compréhension de la photomorphogenèse.

En partant de ce constat, il est alors nécessaire d'avoir à disposition des outils capables de rendre compte des variations de l'environnement lumineux en fonction de la croissance des plantes, quel que soit leur stade de développement. Dans le cas des graminées, les organes sont individuellement très fins, c'est pour cette raison qu'une restriction de l'angle de mesure à 3° a été choisie afin de pouvoir détecter toute variation du ζ. Un dispositif basé sur cet angle de vue a donc été utilisé pour caractériser les variations du ζ directionnel en fonction de la distance, des stades de développement et de l'heure de mesure.

Dans cette thèse nous avons cherché à comprendre quelles étaient les sources de variabilité du signal directionnel et à reproduire ces variations par la modélisation d'un capteur virtuel directionnel. La sensibilité des modèles radiatifs CARIBU et RIRI à la méthode de modélisation du ζ a également été évaluée. Enfin un modèle de capteur directionnel a été créé dans 2 modèles de ciel et une simulation de calcul du ζ à partir de scans 3D de plantes a été comparée avec des mesures.

2. Mesure et variabilité du signal ζ pour une plante dans un couvert

2.1. Directionnalité du signal

L'ensemble des mesures de rayonnement ont été réalisées en utilisant un capteur mesurant le rayonnement compris dans un faible champ de vision. Ce capteur directionnel a été conçu en positionnant une lentille à l'extrémité d'une fibre optique. Une vérification de l'angle d'ouverture du capteur a été effectuée afin de s'assurer que la mesure du rayonnement soit restreinte dans un champ de vision équivalent à un angle solide de 3° (Fig. 24). Un déportement d'environ 1 cm est observable sur la dernière distance de mesure (56 cm). Avec cette vérification, on s'aperçoit que la netteté de la tache lumineuse projetée est réduite avec la distance. On peut supposer que cette dégradation provoque une perte d'information lors de la mesure et qu'elle a donc un léger impact sur la mesure du rayonnement, cet impact n'a pas pu être quantifié.

Des premières approches de mesures directionnelles ont été effectuées avec des capteurs hémisphériques (Sattin *et al.* 1994). Le ζ ainsi mesuré par un capteur orienté verticalement et mesurant un flux horizontal est plus faible que le ζ mesuré par un capteur hémisphérique positionné horizontalement et mesurant le flux vertical. De plus la diminution du ζ est mesurée dès les premiers stades de développement du couvert (LAI = 0,1) lorsque le capteur est positionné verticalement tandis que le capteur orienté horizontalement ne détecte une diminution du ζ qu'à partir d'un certain stade de développement (LAI = 1) pour 3 espèces végétales.

Cette différence de ζ illustre la nécessité de prendre en compte la composante directionnelle dans la mesure des signaux lumineux. Ainsi, une mesure de l'environnement lumineux de manière directionnelle a premièrement été effectuée en utilisant un collimateur sur un capteur hémisphérique. Dans leurs travaux précurseurs, Ballaré et al. (1987) ont construit un capteur directionnel en positionnant un cache permettant de restreindre à un angle solide de 45° l'angle de vue du capteur hémisphérique orienté verticalement et intégrant le rayonnement à 360°. Ce premier capteur directionnel a permis de mettre en évidence une diminution du ζ dès les stades précoces d'un couvert de *Datura ferox*. Une observation réalisée par Chelle et al. (2007) avec un capteur directionnel ayant un angle de vue de 80° met en évidence une différence de profil de diminution du ζ en fonction du LAI, les capteurs directionnels horizontaux mesurent une diminution linéaire du ζ

en fonction du LAI et les capteurs verticaux orientés horizontalement mesurent une très rapide diminution du ζ avant d'atteindre un plateau.

Ces précédents travaux sur la directionnalité du ζ ont été réalisées du sur des dicotylédones, pour lesquelles le couvert se développe rapidement en hauteur. Dans le cas des graminées, l'utilisation de capteurs directionnels ayant des angles de vue de 45° ou plus n'est ~~e~~ sont pas adaptées~~s~~. En Les graminées se développent en produisant de fines feuilles dont l'inclinaison avec le sol diminue lorsque les feuilles atteignent leur maturité. Ces éléments architecturaux rendent difficile la détection d'un signal directionnel provenant d'une graminée seule, d'autant plus dans les premiers stades de développement du couvert où les individus ne sont composés que d'une talle portant une feuille. L'expérience préliminaire présentée dans la figure 15 montre clairement qu'un très faible angle de vue est nécessaire afin de mesurer les variations de l'environnement lumineux liées à la présence d'un plant de graminée. L'utilisation de capteurs directionnels mesurant le rayonnement dans un angle solide de 3° est donc nécessaire pour établir des lois de réponses entre le signal lumineux et les réponses morphogénétiques. Un capteur ayant un champ de vision trop important ne permet pas d'observer ces changements de niveau alors que des réponses sont observables.

Pour Ballaré (2017) les plantes réalisent la photomorphogenèse à une échelle locale, pour chaque organe. Pour accéder à cette échelle de perception, l'utilisation de capteurs directionnelles est indispensable. Malgré des premières tentatives de mesures directionnelles réalisées dès la fin des années 1980, peu d'études se sont intéressées aux signaux directionnels. L'utilisation de capteurs directionnels intégrant une lentille permettant de restreindre le champ de vision à de faibles angles de vue était et reste marginale. Cela peut s'expliquer par la difficulté de ce genre de mesure due aux limitations technologiques. En effet, une mesure du rayonnement dans un angle solide faible la réduit la quantité de rayonnement parvenant au spectroradiomètre. Pour obtenir des mesures de rayonnement exploitables, les cellules de photodiode du spectroradiomètre doivent être suffisamment sensibles pour détecter ces faibles quantités de photons et en particulier les photons correspondant aux longueurs d'onde de Rc et de Rs. Les premiers appareils capables de réaliser ces mesures ont commencés à être commercialisés à partir des années 2000. Cependant, les études utilisant cette technologie pour des mesures directionnelles sont malgré tout peu nombreuses, en partie due à la place prépondérante à la même période des approches moléculaires dans les études sur la photomorphogenèse. Ce travail de thèse est une contribution forte qui a permis de développer

des dispositifs innovant dans la compréhension des interactions plante-environnement et sur ses aspects directionnels.

Un tel dispositif aurait permis d'approfondir l'analyse d'études antérieures telle que l'expérimentation menée par Smith et al. (2020) dans laquelle une plante parasite était placée entre deux plantes artificielles de forme basique ayant la même surface mais présentant une architecture différente (en forme de T ou en forme de I). Leurs résultats mettent en évidence un favoritisme de la plante parasite vers l'architecture en forme de T, montrant ainsi une réception différente des signaux lumineux en fonction de l'architecture. Ce papier montre clairement l'importance des signaux directionnels qui sont directement liés à la forme géométrique. Par extension, on peut émettre l'hypothèse que les plantes traitent l'information lumineuse de manière directionnelle et que l'architecture des plantes voisines est déterminante dans l'étude de la photomorphogenèse. Malgré la mise en évidence de l'importance de l'architecture des plantes, et donc de la composante directionnelle des interactions lumineuses, dans la morphogenèse d'une plante parasite la mesure de l'environnement lumineux perçu par les plantes n'a pas été mesurée réalisée. Ce type d'expérimentation combiné à des mesures directionnelles sur des faibles angles d'ouverture permettrait une meilleure compréhension des mécanismes de photomorphogenèse. Les surfaces étant identiques, un angle d'ouverture trop grand (par exemple 45° ou hémisphérique) mesurerait le ζ correspondant au rayonnement renvoyé par l'ensemble du dispositif et le ζ mesuré pourrait ainsi être identique entre les mesures sur les formes T et les formes I, alors qu'une réponse à une des deux architectures est clairement mise en évidence.

Comme les flux mesurés dans un champ de vision restreint étant faibles, ces outils directionnels sont extrêmement sensibles aux variations environnementales. Pour cette raison l'étude directionnelle de la photomorphogenèse doit être accompagnée de l'étude des variations de l'environnement en fonction des conditions de mesure. Afin de mieux comprendre les interactions entre les plantes et les signaux lumineux, dans ce travail de thèse la variabilité du signal ζ a été mesurée de manière directionnelle selon quatre paramètres sur des plantes isolées : la distance de mesure, le stade de développement, l'heure de mesure ainsi que la présence de voisinage. L'utilisation de capteur directionnel permet d'obtenir l'information lumineuse reçue par une plante à un point donné, le signal ainsi mesuré correspond au signal que les plantes échangent entre elles.

2.2. Distance et stade de développement

Afin de mieux comprendre les interactions entre les plantes, une caractérisation des variations des signaux lumineux renvoyés pas une plante en fonction de la distance a été effectuée. Cette variation a été mesurée pour 4 différents stades de développement.

Lorsque l'on observe les ζ en fonction de la distance pour l'ensemble des dates (Fig. 53) le même profil de variation est observé pour les 4 groupes de semis. Ces résultats montrent que, pour une plante isolée, la valeur de ζ augmente en fonction par de la distance sur les 20 premiers centimètres, distance à partir de laquelle la valeur du ζ mesuré atteint un plateau. Entre 0 et 20 cm, plus le capteur est proche de la plante plus la valeur de ζ mesurée est faible. Ces résultats sont proches des mesures effectuées par Smith et al. (1990) et réalisées à l'aide d'un capteur hémisphérique orienté vers un couvert végétal à plusieurs distances. Comme dans notre expérimentation mesurant la variabilité du signal en fonction de la distance, le capteur est orienté face au soleil dans leur étude. Smith et al. (1990) ont mis en évidence qu'une variation du ζ était observable jusqu'à 30 cm, et que même à cette distance la valeur de ζ mesurée était inférieure au ζ en absence de canopée (Fig. 86). Il existe donc une distance seuil à partir de laquelle les plantes perçoivent peu les variations de l'environnement lumineux, cependant il est possible que cette distance seuil diffère selon l'espèce considérée et qu'elle dépende des dimensions de la plante étudiée.

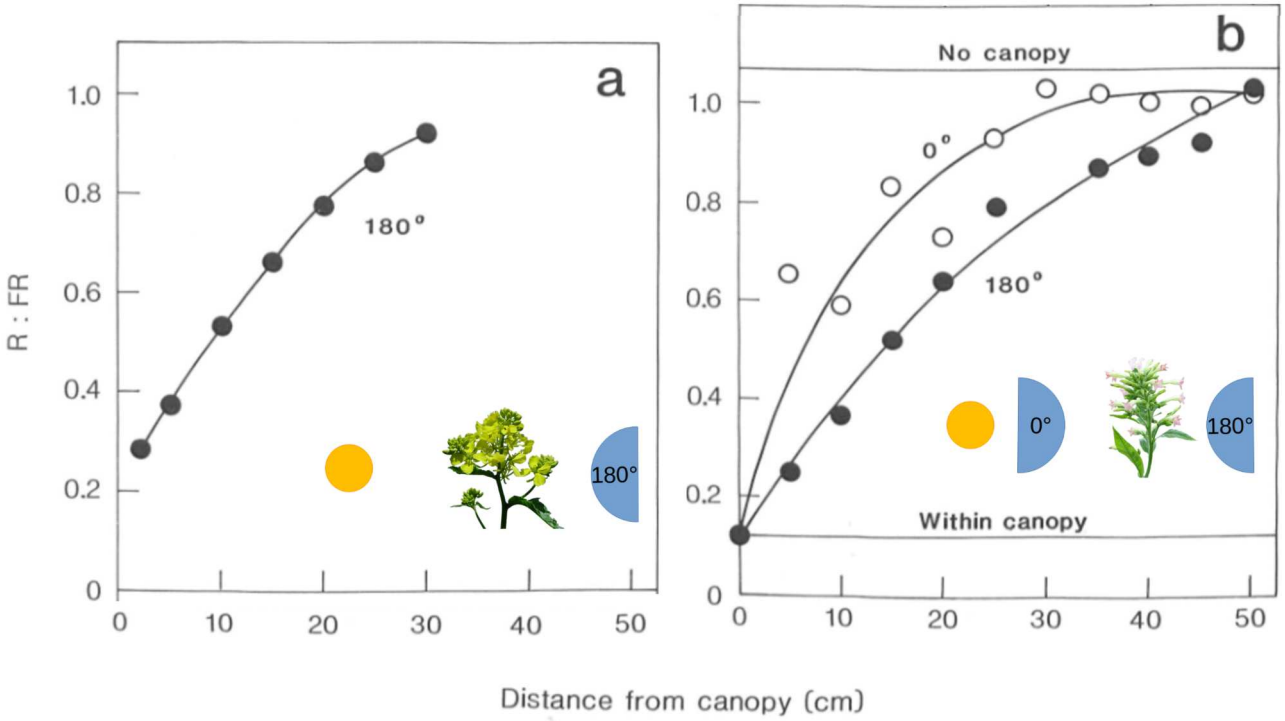


Figure 86: ζ en fonction de la distance des canopées de moutarde (a) et de tabac (b). Les valeurs de ζ ont été mesurés avec le capteur face à la canopée et à un angle par rapport au soleil de 180° (moutarde et tabac) ou de 0° (tabac uniquement). Repris de .

Lorsque l'on sépare les résultats par dates de mesure, le pattern de variation du ζ en fonction de la distance pour les 2 premières dates (Fig. 54A,B) présente le même pattern que celui observé pour la moyenne des données (Fig. 53). En particulier, l'évolution hyperbolique du ζ très marquée sur la première date de mesure (Fig. 54A).

L'augmentation du ζ en fonction de la distance s'explique par la surface végétale présente dans le champ de vision du capteur lié aux propriétés optiques des organes des plantes. Lorsque le capteur est proche des plantes les organes végétaux occupent une grande partie du champ de vision du capteur, le FR renvoyé par la plante participe donc aux faibles valeurs de ζ observées sur les premières distances de mesures et l'occupation du champ de vision du capteur par la plante réduit également le R provenant de l'environnement lumineux ambiant reçu par le capteur. L'augmentation de la quantité de FR combinée à la réduction du R produit les faibles valeurs de ζ mesurées et l'augmentation de la distance provoque une augmentation proportionnelle du ζ à mesure que la surface végétale réduit en proportion dans le champ de vision du capteur directionnel.

Le profil très marqué de l'évolution du ζ en fonction de la distance observé à la première date s'explique par le fait que, lors de la 1ere série de mesure, l'orientation des plantes a été choisie pour

qu'elles soient positionnées dans le champ de vision du capteur, afin de s'assurer de mesurer le flux renvoyé par celles-ci, et à cette date les plantes n'étaient constituées que d'une talle avec une feuille. Cette même orientation a été conservée tout au long de l'expérimentation, ainsi avec l'apparition de nouveau organes et un changement d'orientation des feuilles il n'y avait pas de garantie d'avoir les organes des feuilles présents dans le champ de vision du capteur sur les distances les plus proches, expliquant le second pattern observé à partir de la 3e date de mesure: à partir d'une certaine distance de nouveaux organes entrent dans le champ de vision du capteur ce qui a pour effet de diminuer le zêta mesuré par rapport à la distance précédente.

Un second pattern d'évolution du ζ en fonction de la distance est observable sur les 3e et 4e dates de mesures (Fig. 54C, D). Ce pattern correspond à une diminution du ζ sur les premiers centimètres lorsque la distance augmente avant d'observer une augmentation du ζ entre 20 cm et 30 cm.

À ces dates, les semis S1 à S3 commencent à être composés de plusieurs talles. Dans la croissance des graminées l'angle d'insertion des feuilles se réduit et les organes ne sont plus dressés verticalement mais se rapprochent du sol. Dans cette configuration, les mesures à de faibles distances mesurent peu d'organes végétaux, lorsque la plante est éloignée du capteur sur les premiers centimètres, les organes apparaissent alors dans le champ de vision du capteur et provoquent la réduction du ζ (Fig. 87). À partir d'une certaine distance l'ensemble des organes est vu par le capteur et le ζ mesuré augmente alors à nouveau en fonction de la distance. La diminution du ζ observée s'explique par le faible angle de vue et la proportion de feuille présente dans le champ de vision du capteur, ainsi le ζ mesuré dépend directement de l'architecture des plantes mesurées et dans le cas des graminées la modification de l'angle moyen des organes en fonction du stade de développement joue un rôle sur cette mesure.

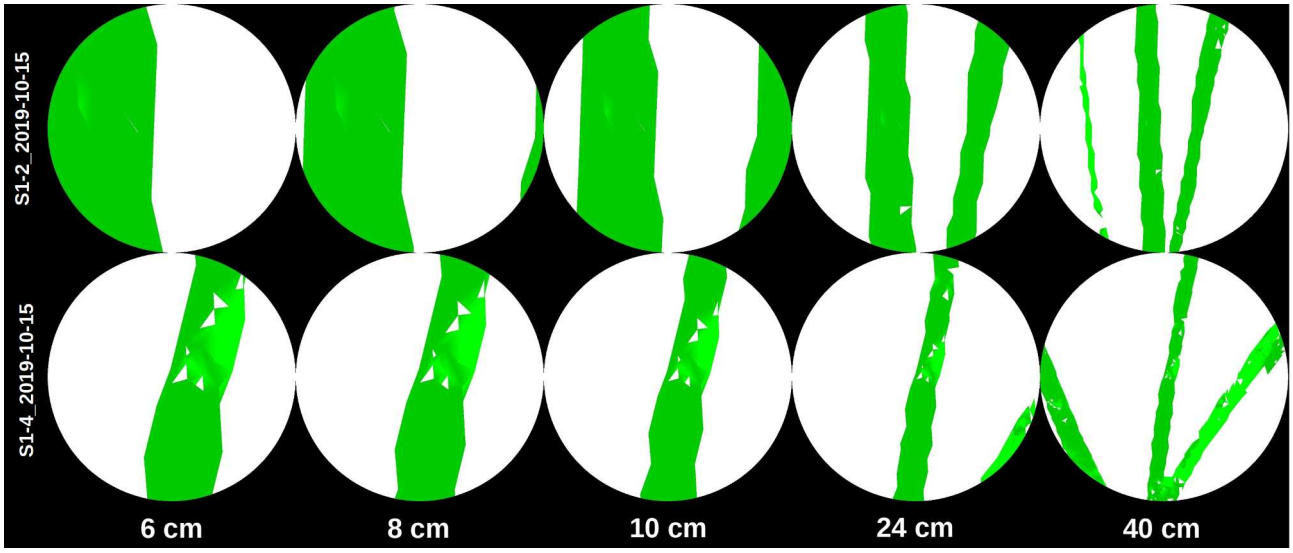


Figure 87: Surface de scan de plante présente dans le champ de vision du capteur directionnel à 5 distances. Cette figure illustre la différence d'architecture de plante au sein d'un même groupe de semis de l'expérimentation ArchiMono. Les scans sont reconstruits avec la méthode Ball Pivoting.

Lorsque les mesures de ζ en fonction de la distance sont moyennées sur l'ensemble des dates, seul le premier profil de variation du ζ est observé les profils de variations (Fig. 53). Ainsi, la prise en compte moyenne des résultats de mesure fait disparaître une partie des informations des variations du ζ. Dans le cas d'une mesure hémisphérique, le très large champ de vision du capteur participe à moyenner les valeurs de ζ et invisibilise une partie des interactions lumineuses entre les plantes. Or les résultats obtenus mettent en évidence l'importance de la prise en compte de l'intégralité des résultats pour établir des lois de réponses. Dans les modèles prenant en compte la photomorphogenèse, les lois de réponses utilisées restent les mêmes quelles que soient les conditions de mesures, il est donc possible que celles-ci ne rendent pas entièrement compte des interactions lumineuses ayant lieu.

Dans le cas de plantes isolées, l'architecture des plantes a un fort impact sur le ζ mesuré de manière directionnelle et 2 patterns d'évolution du ζ en fonction de la distance ont pu être décrits. Le dispositif utilisé est composé d'un seul capteur directionnel, si plusieurs capteurs orientés vers la plante et placées à différentes positions (hauteur, angle horizontal) quelles auraient été les valeurs de ζ ? Bien que l'on puisse supposer que la gamme de valeur puisse être différente en fonction de la position du capteur, on peut également formuler l'hypothèse que les profils de variation du ζ en fonction de la distance soient observables quelle que soit la position du capteur directionnel. De plus, on peut supposer que des processus interviennent également dans la mesure du ζ vertical (Ballaré et al. 1987) et s'ajoutent aux effets de l'ombrage de la canopée.

Les variations observées sont valables dans le cas de plantes isolées ou pour des couverts peu développés. Dans le cas d'un couvert fortement développé les organes des différentes plantes occupent l'ensemble de l'espace et les variations de ζ sont alors corrélés au LAI comme décrit par Ballaré et al. (1987). Ces résultats renforcent l'hypothèse de l'importance de l'utilisation de mesures directionnelles dans la compréhension de la photomorphogenèse à des stades de développement précoce du couvert.

Dans l'expérimentation de Smith et al. (2020), les plantes sont positionnées à une distance d'environ 10 cm des plantes artificielles. Cette distance correspond à la gamme dans laquelle les variations du ζ en fonction de la distance sont observables, ce dispositif est donc bien compatible avec une mesure directionnelle complémentaire. Le dispositif de capteur directionnel permettrait ainsi de mieux comprendre les interactions entre l'architecture des plantes, les signaux lumineux et la morphogenèse.

2.3. Évolution du ζ en fonction de l'heure de mesure

Les mesures de variations de ζ en fonction de l'heure de mesures ont été réalisées à 10 cm afin de s'assurer d'être dans la gamme de distance identifiée dans la partie V.2.2 pour laquelle une variation de ζ en fonction de l'architecture était détectable par un capteur directionnel ayant un champ de vision de 3°.

Dans la littérature, la mesure du ζ au sein des couverts végétaux dans la caractérisation des réponses de morphogenèse aux signaux lumineux a été principalement réalisée de manière dynamique au cours du temps mais avec de forts intervalles de temps entre les mesures (Deregibus et al. 1983; Casal et al. 1986; Wan and Sosebee 1998; Gautier et al. 1999; Xie et al. 2016; Adjesiwor and Kniss 2020). La majorité de ces études portant sur ce sujet se focalisent sur une mesure de l'environnement lumineux, réalisée de manière hémisphérique et souvent mesurant le flux vertical dans le couvert (Casal et al. 1986; Skinner and Simmons 1993; Davis and Simmons 1994; Wan and Sosebee 1998; Gautier et al. 1999; Wherley et al. 2005; Sparkes and King 2008; Zhang et al. 2021). Ainsi, ces études ignorent la composante directionnelle de la qualité de la lumière et l'évolution du ζ au cours du temps au sein d'un couvert reste peu pris en compte. La valeur de ζ obtenue est ensuite considérée comme constante pour l'intervalle de temps avant la prochaine mesure. Or, la plante modifie son propre environnement et dans un couvert développé ce phénomène est d'autant plus important (Smith 1982). Bien que des lois de réponses ont pu être

obtenues à partir de mesures hémisphériques réalisées avec de forts intervalles de temps, ces relations ne permettent pas de comprendre les phénomènes de photomorphogenèse. Pour cela il est nécessaire de descendre à une échelle de mesure plus fine et plus proche de la variable physique.

Des mesures caractérisant les variations du ζ au cours d'une même journée avec un intervalle compris entre 1 et 15 minutes a précédemment été effectué en fonction de l'heure de mesure (Turnbull and Yates 1993; Chiang et al. 2019) ou de l'élévation solaire (Holmes and Smith 1977; Kotilainen et al. 2020). Cependant ces mesures ne correspondent à une évaluation du rayonnement renvoyé par un couvert végétal, ces mesures sont effectuées en plein air ou dans des sous-bois à l'aide de capteurs hémisphériques orientés vers le ciel. Ces mesures caractérisent donc le ζ arrivant sur un couvert végétal et non une évolution du ζ correspondant au rayonnement renvoyé par un couvert végétal au cours de la journée. De plus, des simulations réalisées par Varlet-Grancher et al. (1993) ont été réalisées dans un couvert végétal, cependant les mesures ont été effectuées avec un capteur hémisphérique orienté vers le ciel. Nos travaux apportent donc une première caractérisation de l'évolution du ζ directionnel en fonction de l'heure de mesure.

Les résultats de variation du ζ correspondant au rayonnement renvoyé par un peuplement végétal en fonction de l'heure de mesure est sont synthétisés dans le tableau 1 pour les différentes conditions de mesure. Les résultats mettent en évidence que les conditions de ciel clair augmentent la variabilité des mesures de ζ au cours de la journée tandis que les conditions de ciel couvert réduisent la variabilité du signal en fonction de l'heure de mesure. Les résultats de la variabilité du ζ journalier présentent de fortes variations dans le cas de peuplements précoces sous un ciel clair (Fig. 56). Ces variations posent la question de l'importance du moment de la mesure afin d'analyser les facteurs de déclenchement des réponses des jeunes pousses aux signaux lumineux. On peut également noter la présence de pics de valeurs pour les conditions de mesures en ciel clair, ces pics sont liés à des variations de lumière provoquées par la structure de la serre, il serait donc pertinent de reproduire ces mesures en extérieur.

Dans le cas de la mesure d'un peuplement développé pour des conditions de ciel couvert on remarque également que les valeurs de ζ mesurés correspondent aux seuils de ζ théoriques les plus bas, en lien avec les propriétés optiques des plantes.

		Météo	
		clair	couvert
Stade	précoce	variable	constante a
	développé	constante b	constante c

Tableau 1: Variabilité du ζ en fonction de l'heure de mesure selon les conditions de ciel et le stade de développement des plantes.

L'évolution du ζ en fonction de la présence de plantes voisines montre une diminution du ζ lorsque le nombre de plantes présentes augmente. Ces résultats étaient attendus puisqu'une augmentation du LAI provoque une diminution du ζ. Cependant un seuil de ζ est observable pour les mesures effectuées avec une seule ligne de plante, ce seuil peut être expliqué par le fait que l'ajout de plantes pour cette condition ne rajoute pas de surface végétale dans le champ de vision du capteur, alors que le ζ continue de descendre jusqu'au 6e rang lorsque 3 lignes de plantes en forte densité sont présentes. Cette diminution peut s'expliquer par la présence des feuilles des plantes latérales qui rentrent dans le champ de vision du capteur, produisant un couvert végétal plus dense.

Ainsi, le signal ζ est extrêmement variable au cours du temps et il est à la fois lié aux conditions de ciel et à l'architecture des plantes. L'étude de la photomorphogenèse reste très compliquée au sein des couverts végétaux, cette complexité explique en partie les lacunes en termes de données sur l'intégration du signal lumineux par les plantes. Afin de pallier cette complexité, une approche de modélisation est nécessaire pour comprendre les interactions plante-environnement ayant lieu au sein du couvert.

Les résultats de mesures obtenues correspondent à l'étude d'une seule direction, pour une seule espèce, à une seule localisation, pour une seule saison et pour des conditions de plantes isolées. Des mesures supplémentaires reprenant le même dispositif pourraient donc être réalisées pour de multiples conditions de mesure, notamment en termes de différences architecturales, afin d'alimenter les connaissances sur la variation du ζ directionnel.

3. Capacité des modèles à simuler le ζ de manière directionnelle

Un capteur directionnel a été modélisé à l'aide de deux modèles de lumière: le modèle CARIBU et le modèle MALIA. Dans ces deux modèles, un capteur de lumière directionnel a été représenté afin de reproduire des mesures de rayonnement directionnel effectuées sous serre.

3.1. Création du capteur

Ce travail se veut comme une première tentative de mesure virtuelle de l'environnement directionnel perçu par les plantes. Ce travail a nécessité de nombreuses étapes de conception car les outils utilisés ne sont pas adaptés à cette méthode de mesure de l'environnement lumineux.

Le principal problème du capteur virtuel créé sous CARIBU est le besoin d'un collimateur. En effet, le modèle d'origine est conçu pour calculer l'ensemble des rayons atteignant des surfaces d'intérêt, ainsi la mesure est par défaut réalisée dans toutes les directions. Afin de restreindre la zone de calcul pour correspondre au champ de vision du capteur, un collimateur a dû être ajouté à la scène. La longueur du collimateur est égale au rapport entre le rayon de la cellule du capteur et la tangente de la moitié de l'angle de vue. Le diamètre du capteur a également dû être évalué. En effet, le champ de vision du capteur est très faible et puisque la longueur du collimateur est directement proportionnelle au diamètre du capteur et à l'angle de vue, le capteur virtuel nécessite un long cylindre pour que les rayons arrivant sur la cellule du capteur soient collimatés dans un angle solide de 3°. Un premier choix de taille du diamètre du capteur a été effectué en utilisant la taille réelle du capteur en cm (soit 0,8 cm), puis une série de tests de modélisation sur un échantillon de 5 scans d'ArchiMono a été réalisée avec des tailles de capteurs comprises entre 0,5 cm et 2,5 cm afin de déterminer quel diamètre produisait les valeurs de ζ les plus proches des mesures. La taille finale utilisée pour le diamètre du capteur est de 1,5 cm. Le diamètre finale du capteur virtuel est donc plus grand que celui du capteur réel, le capteur directionnel virtuel reçoit donc théoriquement plus de flux que le capteur utilisé sur le banc spectral et est ainsi une source potentielle de différence entre les mesures et les simulations. Afin de s'assurer qu'il n'y avait pas de superposition entre le collimateur et la cellule du capteur virtuelle, le diamètre de la cellule du capteur est 0,5 % plus petit que celui du collimateur.

Bien que la cellule du capteur directionnelle soit virtuelle et n'intervient donc pas dans les interactions lumineuses de la scène, le collimateur utilisé pour restreindre le champ de vision à 3° est présent dans la scène. Les propriétés optiques ont donc dû être choisies afin de réduire au

maximum les perturbations liées à la présence de ce collimateur. Les propriétés optiques de chaque triangle sont sous la forme d'un tuple contenant 4 paramètres dans CARIBU : la réflectance des faces supérieures, la transmittance des faces supérieures, la réflectance des faces inférieures, la transmittance des faces inférieures. Dans la construction du collimateur, on s'est assuré que les faces supérieures correspondaient aux faces extérieures du collimateur et que les faces inférieures correspondaient aux faces intérieures. Le capteur ne doit recevoir que du rayonnement provenant de l'ouverture du collimateur, la réflectance et la transmittance des faces supérieures sont donc de 0 pour que les rayons arrivant sur l'extérieur du cylindre ne pénètrent pas dans le capteur et les faces intérieures ont une transmittance de 1 et une réflectance de 0 pour que les rayons entrant dans le collimateur avec un angle supérieur à 3° traversent le collimateur.

Une approche de restriction de l'angle de vue des capteurs a été proposée par Gautier et al. (2000). Dans leurs travaux la cellule du capteur utilisée filtre les rayons en fonction de leur angle de vue à l'aide d'une fonction de sensibilité. Le champ de vision de leur capteur virtuel directionnel peut ainsi être estimé à 60°. De plus dans leur modèle utilisé une approche de lancer de rayons pour lesquels les rayons partent depuis le capteur vers les objets de la scène et seulement 20 rayons sont ainsi émis.

Dans leurs travaux, Chelle et al. (2007) présentent des simulations de ζ réalisées avec des capteurs directionnels mesurant l'environnement lumineux dans un angle solide de 40° et de 80°. La méthodologie utilisée pour la réalisation de ces capteurs directionnels n'est pas détaillée. Compte tenu du fait que les résultats obtenus pour les 2 angles de vue ne sont pas identiques, une restriction de l'angle de vue du capteur modélisé a donc dû être réalisée avec succès. On peut supposer qu'un collimateur a été créé pour ces simulations ou que le modèle a été modifié afin d'avoir accès à l'information directionnelle des rayons arrivant sur le capteur, à l'instar des travaux de Gautier et al. (2000). Ces deux approches de construction de capteurs directionnelles ont été réalisées pour des angles de vue supérieurs à ceux que nous avons étudiés. De plus ces capteurs n'ont pas été clairement présentés par leurs auteurices, limitant leur utilisation dans d'autres travaux.

De nombreux problème de simulation sont liés à notre méthodologie. En effet, la simulation de la photomorphogenèse est réalisée à l'aide de modèles FSP où les problèmes de reconstruction et de positionnement des scans ne sont pas à prendre en compte. Les limitations liées à la reconstruction 3D de plantes réelles mises en évidence par nos travaux restent donc marginales.

La principale limitation de ce capteur virtuel créé avec CARIBU est la présence du collimateur dans la scène. Bien que les propriétés optiques ont été choisies pour perturber le moins possible les interactions lumineuses entre les différents éléments de la scène, sa construction actuelle reproduit une « zone noire » dans laquelle les rayons arrivant sur l'extérieur du collimateur sont complètement absorbées. De plus des éléments de la scène, en particulier des organes végétaux, peuvent passer au travers du collimateur et peuvent obstruer le champ de vision de la cellule du capteur, faussant les valeurs de ζ simulées. Dans le cadre de nos simulations le capteur était situé à distance d'une plante isolée, cette limitation n'a pas posé de problème dans la reproduction de nos conditions de mesure. Cependant cette contrainte pourra avoir un fort impact pour de futures utilisations, en particulier si l'on cherche à positionner ce capteur directionnel au sein d'un couvert végétal dans lequel des organes pourraient traverser le collimateur.

La seconde limitation principale des simulations effectuées avec CARIBU est le temps de calcul. En effet, CARIBU a été principalement été utilisé pour le calcul de rayonnement dans des modèles FSPM, dans lesquels la représentation des organes végétaux est simple et est faite avec un faible nombre de triangle. Dans le cadre de nos simulations, le nombre de triangle des différents scans était très supérieur à ceux d'une scène OpenAlea classique que peuvent représenter les modèles VGL ou L-grass par exemple. La taille des fichiers importés à un fort impact sur le temps de calcul.

La création d'un capteur directionnelle dans MALIA a été beaucoup plus simple à réaliser. Dans ce modèle la caméra réalise une capture de la scène avec les informations spectrales de chaque pixel. Ainsi la création du capteur a consisté à modifier le champ de vision de la caméra à 3° et une restriction de la zone de calcul a été réalisée en supprimant les données des pixels qui n'étaient pas présents dans le cercle de mesure assurant la mesure à 3°. Dans ce modèle, 128 rayons sont envoyés depuis la caméra vers la scène pour chaque pixel et chaque longueur d'onde pour lesquelles on réalise la simulation. Dans sa construction, ce capteur virtuel directionnel se rapproche de celui construit par Gautier et al. (2000), pour lequel seulement 20 rayons étaient lancés à partir du capteur. Cette différence d'échelle entre leurs travaux et les simulations effectuées dans ce travail de thèse pose la question du nombre de rayons minimum nécessaire afin d'obtenir une simulation du ζ proche des résultats attendus. Une analyse approfondie de MALIA permettrait d'estimer la sensibilité de ce capteur au nombre de rayon lancé pour chaque pixel.

La première méthode, basée sur le modèle de transfert radiatif CARIBU, a permis d'obtenir des résultats de simulation du ζ proche des mesures à partir des données de l'expérimentation ArchiMono et en utilisant la méthode de reconstruction « Ball Pivoting ». Cependant, les résultats obtenus avec le même modèle de capteur directionnel se sont avérés très éloignés des mesures pour les données de l'expérimentation ArchiDuo. Cette différence de résultat s'explique principalement par les différences de dispositif entre les 2 expérimentations, les erreurs de simulations sont particulièrement attribuables à la non reproduction de la serre dans le modèle et à l'orientation de la mesure. Le modèle de capteur directionnel construit dans CARIBU semble donc capable de reproduire les valeurs de ζ mesurées dans des conditions où le capteur est orienté vers le soleil et où aucune ombre portée ne perturbe la mesure.

Les modèles de capteurs directionnels construits à partir des modèles CARIBU et MALIA se basent sur des calculs de rayonnement. Ces calculs sont cependant coûteux en temps, bien que les simulations réalisées avec MALIA soient plus rapides que celles effectuées avec CARIBU. Une approche alternative à ces modèles a été réalisée en se basant uniquement sur la surface végétale capturée (SVC) par une caméra représentant un capteur directionnel. Cette approche a permis d'obtenir une corrélation entre le ζ mesuré et la SVC et d'établir une équation permettant de calculer un ζ à partir de n'importe quelle valeur de SVC. La limite de cette approche se situe dans l'impossibilité de prendre en compte les conditions de ciel et peut dès maintenant être intégré à des modèles prenant en compte la photomorphogenèse. Cette méthode se rapproche de modèles déjà existants basant leurs calculs de ζ à partir de LAI ou des surfaces foliaires (Kahlen and Stützel 2011), et permettraient une première prise en compte de la composante directionnelle des interactions entre les plantes et la lumière dans la modélisation.

3.1.1. Simulation avec le modèle CARIBU

Les résultats de simulation réalisées par CARIBU sont très différents entre les simulations basées sur les données d'ArchiMono et d'ArchiDuo, la rRMSE étant systématiquement plus faible pour ArchiMono que pour ArchiDuo. Cette plus grande rRMSE pour les données d'ArchiDuo s'explique par une très forte dispersion des points entre les valeurs mesurées et les valeurs simulées.

Les simulations réalisées simulées avec les scans d'ArchiMono et d'ArchiDuo reconstruits avec la méthode Ball Pivoting (Fig. 65,67) ont des valeurs de ζ simulées plus élevées que leurs simulations respectives effectuées avec les scans reconstruits par la méthode Marching Cubes

(Fig. 71,73). Les valeurs plus faibles pour les simulations d'ArchiMono rapprochent les résultats de simulation des mesures mais renforcement la sous-estimation des simulations d'ArchiDuo.

Une comparaison de la variation du ζ en fonction de la distance est possible entre les mesures et les simulations d'ArchiMono. Deux profils de variation du ζ en fonction de la distance ont été mis en évidence dans la partie IV.3.1 : un profil pour lequel le ζ augmente sur les 20 premiers centimètres avant d'atteindre un plateau et un profil pour lequel le ζ diminue sur les premiers centimètres avant d'augmenter. Les simulations effectuées avec la méthode de reconstruction Ball Pivoting reproduisent ces profils de variation, ainsi on observe des courbes pour lequel le ζ simulé augmente au cours du temps et se stabilise à partir d'une certaine distance, et des courbes où une diminution temporaire du ζ peut être observée. Cependant les variations ne se stabilisent pas à partir de 20 cm comme pour les mesures. Cette variabilité présente quelle que soit la distance de simulation peut s'expliquer par le fait que dans les simulations, la netteté des informations reste la même quelle que soit la distance de mesure, tandis qu'une perte de netteté a pu être observée sur les bords de la zone de mesure (Fig. 24). Pour les simulations effectuées avec la méthode de reconstruction Marching Cubes, quelques courbes présentent un profil similaire à ceux observés lors des mesures. Cependant pour la plupart des simulations le profil de variation du ζ en fonction de la distance correspond à une diminution constante du ζ lorsque la distance de simulation augmente. La méthode de reconstruction Marching Cubes produit donc des résultats de simulation allant à l'inverse des résultats attendus et semble inadaptée pour la simulation de ζ directionnelle avec CARIBU.

Les résultats obtenus à l'aide des simulations d'ArchiMono montrent que le modèle CARIBU est capable de produire des simulations de ζ directionnel similaire à des mesures effectuées. Ce modèle de capteur directionnel reste cependant améliorable. La dispersion observée sur les données d'ArchiDuo quelle que soit la méthode de reconstruction met en évidence une limite de simulation liée aux conditions de simulations.

Cependant, le modèle CARIBU est un modèle gourmand en temps de calcul et en nombre de processeurs. Ce temps de calcul est notamment non négligeable pour des simulations utilisant plusieurs scans de plantes visant à reproduire un peuplement. Le temps de calcul observé pour une seule condition de ciel et de scène a été de plusieurs jours.

3.1.2. Simulation avec le modèle MALIA

La représentation du ζ en fonction de la distance à partir des simulations d'ArchiMono réalisées par MALIA met en évidence de très faibles variations du ζ. Les simulations réalisées avec les scans reconstruits par la méthode Ball Pivoting correspondent au profil principal de variation identifié dans la partie IV.3.1 pour lequel le ζ augmente sur les premiers centimètres avant d'atteindre un plateau. Les valeurs de ζ sont cependant fortement sous-estimées par rapport aux mesures. De plus, le second profil de variations du ζ pour lequel le ζ diminue avant d'augmenter de nouveau n'est pas observable à partir de ces simulations. Pour les simulations réalisées à partir des scans reconstruits avec la méthode Marching Cubes, le ζ simulé est pour la quasi-totalité des simulations constant en fonction de la distance.

Les simulations de ζ directionnel réalisées par le modèle MALIA sont pour le moment moins précises que les simulations réalisées par le modèle CARIBU. Cependant, ces simulations représentent une première tentative de simulation directionnelle avec un modèle de lancer de rayon dont l'utilisation est nouvelle pour ce domaine de recherche. Contrairement à CARIBU qui est utilisé depuis longtemps par les équipes de recherches se focalisant sur la photomorphogenèse, le modèle MALIA est un nouvel outil à notre disposition et de nombreuses modifications peuvent être apportées afin d'améliorer cette première approche de capteur directionnel.

Le modèle MALIA a pour avantage de ne pas nécessiter une représentation du capteur directionnel dans la scène et de réaliser un calcul rapide du rayonnement. Malgré cet écart entre les mesures et les simulations, ses avantages sont précieux pour de futurs travaux basés sur la modélisation de la photomorphogenèse. Pour cette raison, des pistes d'amélioration telle que l'intégration de la serre dans le calcul et l'utilisation de différents ciels sont à explorer afin de le rendre pertinent dans la prédiction du ζ directionnel de ce modèle de capteurs directionnelles.

3.1.3. Impact de la méthode de reconstruction des nuages de points sur la modélisation du ζ directionnelle

Deux méthodes de reconstruction ont été utilisées : la méthode « Ball Pivoting » qui reconstruit le modèle 3D des plantes en reliant les points du scan sans règle d'organisation, et la méthode « Marching Cubes » qui projette les points du scan sur une surface plane avant de créer les triangles afin de reproduire la cohérence des structures végétales. Les deux méthodes présentent chacune leurs limitations. La méthode « Ball Pivoting » présente pour de nombreux scans des trous

dans la reconstruction des feuilles. La méthode « Marching Cubes » produit à la fois des surfaces excédentaires et supprime des organes que l'algorithme n'arrive pas à regrouper sur un même plan.

Les résultats de simulation montrent que la méthode de reconstruction « Ball Pivoting » présente des résultats de simulation plus proche des mesures que les simulations effectuées avec les scans reconstruits avec la méthode « Marching Cubes ».

3.2. Difficulté de représentation de la mesure dans les modèles

Les résultats de simulations directionnelles donnent des prédictions de ζ fortement éloignées des mesures. Il est donc nécessaire d'identifier l'ensemble des éléments provoquant ces erreurs de modélisation afin de les corriger ou, a minima, d'en tenir compte pour l'amélioration de ces modèles directionnels. Les sources d'incertitude provoquant ces écarts de modélisation peuvent être séparées en 2 catégories : les sources de variations liées à la méthode de modélisation, et donc communes aux 2 expérimentations, et les sources de variations liées aux données des deux différentes expérimentations.

De fortes différences de résultats sont observables entre les simulations effectuées à partir des scans des deux expérimentations ArchiMono et ArchiDuo, pour les simulations effectuées avec les modèles CARIBU et MALIA. Ces différences peuvent être liées aux conditions de mesures (météo, matériel, protocole) ou à la représentation de ces conditions de mesure dans les modèles (construction de la scène virtuelle).

La reconstitution des conditions de mesure est un élément ayant pu introduire de nombreuses différences entre la modélisation et la mesure. La visualisation de la zone de mesure du capteur à l'aide d'un illuminateur (Fig. 24) a mis en évidence un déportement de la zone de mesure en fonction de la distance, ce déportement n'a pas été pris en compte pour les simulations. La perte d'information sur les bords de la zone de mesure qui a été observée n'a également pas été prise en compte pour la modélisation : quelque-soit la distance, les simulations calculent le rayonnement perçu nettement dans une zone de 3° sans perdition d'information. Ces deux éléments ont ainsi pu induire une sur-estimation du ζ, en particulier la non-perdition d'information sur les bords de la zone de mesure car lesorganes présents à la limite de la zone de mesure voient le rayonnement qu'ils renvoient entièrement comptabilisé alors que cette information a été possiblement altérée lors de la mesure avec le banc spectral. Cette hypothèse pourrait expliquer l'atteinte d'un plateau à partir

d’une certaine distance (Fig. 54) sur les mesures alors que des variations de ζ peuvent être observées quelque-soit la distance pour les simulations (Fig. 64).

Un autre élément de variation lié à la représentation de la scène concerne la position des scans. Les scans d’ArchiDuo ont été corrigés à partir des photos prises lors des mesures. Les données d’ArchiMono n’ont pas bénéficié de cette correction, car la caméra n’était pas encore intégrée au dispositif du banc spectral. Avec ou sans recalage, l’acquisition 3D des plantes a nécessité un déplacement des plantes après la mesure avec le banc spectral. Ce déplacement et le temps écoulé entre la mesure du ζ a pu modifier la position et l’inclinaison des organes. La mesure de l’architecture ne garantit donc pas que la position de l’ensemble des organes soit identique entre ces 2 mesures. Cette différence est donc également une source de variation potentielle de la simulation du ζ.

Des sources de variations peuvent également être liées aux différences d’expérimentations. Les mesures d’ArchiMono et d’ArchiDuo ont été effectuées en octobre-novembre, sur les années 2019 et 2020 respectivement. Cependant, les conditions de mesures de l'environnement lumineux sont très différentes entre les expérimentations ArchiMono et ArchiDuo. Pour ArchiMono le capteur était constamment orienté vers le soleil afin de réduire l’effet de l’heure de mesure sur les résultats et le capteur était ainsi déplacé au cours de la journée. Inversement, le dispositif d’ArchiDuo était fixe entre chaque mesure et chaque campagne de mesure, avec le capteur constamment orienté vers le Nord. Cette différence de positionnement du capteur implique également le fait que pour les mesures d’ArchiMono, les plantes n'étaient jamais ombragées par la structure de la serre alors que les plantes d’ArchiDuo subissaient les variations de l’environnement lumineux liées à la structure de la serre. En complément des différences de position lors des mesures, les dispositifs se différencient par la présence d’un cache à l’arrière du pot pour ArchiMono et d’une plaque anthracite à l’arrière du banc pour ArchiDuo. Sur les résultats, on constate que les différences entre les mesures et les simulations sont beaucoup plus dispersées pour les données d’ArchiDuo. Cette forte dispersion des résultats montre que le modèle CARIBU n’est pas capable de simuler le ζ en prenant en compte la serre.

3.3. Limites des modèles de ciel utilisés

L’une des principales limitations de nos expérimentations a été de réaliser les mesures sous serre. Bien que la mesure en conditions contrôlées apportait de nombreux avantages, l’impact de la serre sur nos résultats a été sous-estimé, et en particulier sur les simulations qui n’étaient pas

capables de reproduire correctement ces conditions. Des mesures de variation du ζ selon les conditions testées dans cette thèse pourraient être réitérées dans des conditions extérieures, dans le but de vérifier si les variations observées restent similaires en condition de plein champ et afin d'améliorer la modélisation.

Dans les simulations effectuées, le ζ est recalculé à partir des sorties de simulations et des données de mesures. Cette approche a comme limite de ne pas pouvoir prendre en compte l'ombre portée provoquée par la serre, où les valeurs de flux de direct devraient être fixés à 0 au lieu de 1 comme c'est actuellement le cas. Cette méthode considère donc que pour chaque simulation, le soleil parvient directement sur la scène. Bien que la plupart des mesures aient été effectuées entre 10 h et 14 h UTC, limitant les problèmes d'ombre portée plutôt observée en début et en fin de journée, il n'est pas impossible que des mesures aient été réalisées dans des conditions où les plantes ne recevaient pas de direct. Ce recalcul est pour le moment nécessaire, car le choix a été fait de ne pas représenter la serre dans les modèles. Ce choix s'explique par la résolution d'écran, en particulier pour CARIBU, pour laquelle la présence d'une serre de taille importante par rapport au modèle de capteur directionnel risquait de perturber la simulation. Cette approche simplifiée sans reproduction de serre est également possible du fait de l'accès aux mesures de flux à l'extérieur et à l'intérieur de la serre.

Le modèle CARIBU utilise les modèles de ciel SOC et UOC. Ces modèles ont pour avantage leur simplicité, cependant ils sont décrits comme étant de mauvaise qualité depuis plus de 20 ans (Frazer et al. 1997). Les simulations de peuplements se sont vues limitées par le temps de calcul, en effet les scènes recréées contiennent un nombre de triangle important et le calcul des rediffusions en est impacté, CARIBU est donc préférablement utilisable pour des scènes avec peu de plantes reconstruites ou en recréant des peuplements avec une simplification des surfaces afin de réduire le nombre de triangle. Cette limitation est restreinte dans les cadres d'utilisation classique ds FSPM où les plantes sont simulées par des formes géométriques simples.

L'utilisation de modèles de ciel plus performant est donc une piste d'amélioration du modèle de capteur directionnel, mais également une piste d'amélioration du modèle CARIBU quelles que soient les simulations effectuées nécessitant un calcul de rayonnement.

4. Sensibilité des modèles de transfert radiatifs à la simulation du ζ

Des fortes incertitudes de prédiction du ζ directionnelle ont été identifiées sur les simulations effectuées avec les deux modèles de capteur directionnel. Il est donc nécessaire de savoir si ces incertitudes ont un impact sur la morphogenèse dans les modèles FSP utilisés et s'il est possible d'établir des lois de réponses malgré ces incertitudes. La sensibilité de la morphogenèse à la méthode de modélisation du ζ dans des modèles FSP a donc été évaluée pour deux espèces de légumineuses présentant des architectures contrastées.

Nos résultats montrent des différences considérables de calcul de ζ liées au modèle de transfert radiatif. Le modèle de référence CANESTRA a montré une diminution précoce de ζ aux premiers stades du développement, ces diminutions précoces de ζ dans la canopée ne peuvent être observées in vivo que par un capteur à champ de vision réduit (Ballaré et al. 1987) alors que les valeurs de ζ observées pour CANESTRA Face et RIRI étaient plus proches des valeurs de ζ mesurées par des capteurs hémisphériques. L'utilisation d'un capteur avec un FOV restreint par Ballaré et al. (1987) montre une détection précoce de la diminution de ζ par rapport aux capteurs hémisphériques. Cette détection précoce est observable dans les simulations effectuées avec l'organe CANESTRA, qui montrent des profils proches des mesures effectuées avec un capteur orienté verticalement (Sattin et al. 1994; Sparkes et al. 2006) et des simulations effectuées par Chelle et al. (2007). Alors que les simulations effectuées par CANESTRA Face et RIRI sont proches des mesures effectuées avec un capteur orienté horizontalement (Chelle et al. 2007).

Pour chaque organe, la diminution du ζ moyen reçu par les phytomères observée est cohérente avec les valeurs attendues car la fermeture de la canopée fait apparaître de nouveaux organes dans une canopée déjà développée et reçoit donc plus de rediffusion. Cependant, pour les entre-nœuds de trèfle blanc, les stolons émis sont éloignés des autres organes ce qui explique la stabilité du ζ observé pour l'organe CANESTRA, qui prend en compte les structures voisines. La diminution de ζ observée pour RIRI s'explique par l'augmentation du LAI.

Bien que notre travail soit purement in silico, aucune des méthodes de modélisation ζ utilisées dans cet article n'a produit des plantes avec des propriétés aberrantes : des variables telles que la surface foliaire moyenne, la hauteur maximale de la plante, la biomasse moyenne des tiges et la croissance relative des organes se situaient dans la même fourchette de valeurs que celles mesurées par Faverjon (2018) et Gautier et al. En outre, les simulations de la longueur des organes réalisées

avec le modèle de référence ont donné des résultats plus proches de la mesure effectuée par Faverjon (2018) qu'avec les autres modèles, ce qui prouve que l'utilisation de ce modèle rend la scène virtuelle plus réaliste. De plus, les simulations réalisées par Gautier et al. (2000) sur le trèfle blanc avec la même densité présentaient une taille d'organe similaire à celle de nos simulations, assurant leur exactitude.

Pour évaluer la sensibilité de la morphogenèse à la simulation du ζ, la taille finale des organes de la plante (pétiole, entre-nœud) a été calculée en utilisant 5 approches différentes de modélisation du ζ ; le calcul a été effectué uniquement sur quelques zones restreintes qui sont connues pour être des sites de perception au début de la croissance des organes. Les simulations de la taille finale des organes en réponse aux valeurs calculées de ζ et PPFD étaient en accord avec leurs données expérimentales.

Une comparaison entre l'effet de la distance et du stade de développement observé dans la partie V.2.2 et les simulations effectuées par RIRI et CARIBU dans le but de tester la sensibilité des modèles à la méthode de modélisation du ζ peut être réalisée à partir de la figure 41 qui présente les différentes valeurs de ζ pour 3 densités. La distance entre les plantes pour une densité de 11, 100 et 400 plantes.m-2 correspond respectivement à une distance de 30 cm, 10 cm et 5 cm. Le calcul du rayonnement est fait à partir des rayons provenant de l'ensemble des directions et sont ainsi proche du fonctionnement des capteurs hémisphériques, on s'attend donc à observer des résultats similaires à ceux observés par Smith et al. (1990) (Fig. 86). Quelle que soit la méthode de modélisation, à partir de 140 DOY le ζ simulé par une même méthode diminue lorsque la distance entre les plantes augmente. Les 3 méthodes de modélisation présentent donc une variabilité de la mesure en fonction de la distance, mais cette différence de ζ est principalement due au LAI des différentes conditions. En particulier pour la méthode RIRI qui prend en compte la surface foliaire dans le voxel de calcul pour simuler le ζ.

5. Intégration du signal

La photomorphogenèse est liée aux variations du photoéquilibre des phytochromes, c'est-à-dire à la proportion de phytochrome sous forme active dans les organes végétaux. Or pour qu'un phytochrome passe de la forme active à la forme passive, il passe par une forme intermédiaire qui doit être exposé à l'obscurité pour atteindre la forme finale (Mazliak 1982). Pour un couvert végétal en plein champ cette obscurité est apportée par la nuit, cependant l'intégration du signal au cours de la journée reste encore inconnue. On peut ainsi émettre plusieurs hypothèses sur la manière dont ce signal est intégré. Le photoéquilibre des phytochromes peut par exemple être influencé par le ζ moyen reçu au cours de la journée, par le ζ de la journée où la quantité d'énergie est la plus importante ou par les dernières heures d'exposition à la lumière avant le passage à l'obscurité.

Selon Varlet-Grancher et al. (1993) la photomorphogenèse au niveau d'un site de perception doit être évaluée en prenant en compte 3 éléments : le temps de résolution (c'est-à-dire la durée de la mesure), l'intervalle de mesure et le temps d'intégration. Le temps entre une nouvelle valeur de ζ perçue par la plante et son expression est divisé en 3 temps:

1. l'obtention d'un nouvel équilibre des phytochromes
2. la période de réversibilité de l'action de la lumière
3. le délai des réponses morphogénétiques

Le temps de mesure doit être rapide, car il doit permettre de mesurer les courtes variations des phytochromes, compris entre 1 s dans des conditions naturelles idéales et 30 s selon certaines conditions météorologiques (vent). L'intervalle de temps entre chaque mesure est déterminé par la période de réversibilité de l'équilibre des phytochromes, qui dure entre 5 min et 15 min, les mesures de ζ devraient donc idéalement être effectuées à cet intervalle. À partir de leurs travaux, Varlet-Grancher et al. (1993) ont mis en évidence que le ζ intercepté par les plantes en début de journée a une faible influence sur la morphogenèse mais que la dernière valeur de ζ de la journée (supérieur à 1µmol.m-2.s-1 et pour une valeur de bleu supérieure à un seuil) doit être mesurée avec précision.

Le ζ est un moyen standardisé de mesurer la photomorphogenèse Varlet Grancher & Gautier (1995), mais la mesure réalisée avec les capteurs hémisphériques ne permet pas d'obtenir une description de la lumière directionnelle reçue par les organes. Le temps d'intégration pour mesurer

le MAR doit être fait sur 24 h afin de prendre en compte l'ensemble des réactions des phytochromes à la lumière et à l'obscurité.

Ainsi dans ces deux travaux, Varlet-Grancher et Gautier ont dictés une ligne directrice à suivre pour la mise en place de protocoles de mesure de la photomorphogenèse. Malgré cela, peu de recherche ont fait le choix de travailler sur un intervalle de mesure aussi fréquent. Différentes raisons peuvent expliquer cette différence de méthode de mesure entre la théorie proposée par Varlet-Grancher et Gautier et les travaux sur la photomorphogenèse. La principale raison à cela est la difficulté technique, un suivi au long terme avec des mesures aussi régulières nécessite un fort investissement en temps et en matériel. Cette difficulté technique est d'autant plus importante lorsque l'on essaye de réaliser des mesures directionnelles, pour lesquelles le nombre de mesure est multiplié par le nombre de site de perception. Enfin toutes les espèces ne s'adaptent pas à cette méthode, notamment pour le cas des graminées où la mesure de l'impact du signal sur la morphogenèse est différé par le temps de croissance hors des gaines. La difficulté d'interprétation des résultats est également à prendre en compte : l'échelle de temps entre la mesure du signal lumineux et les réponses morphologiques ne se font pas sur la même échelle de temps, quelles mesures réalisées au cours de la journée j doivent être mises en lien avec les réponses observées à j+1 ? Les mesures en fin de journée doivent-elles être considérées comme ayant plus de poids comme proposé par Varlet-Grancher et al. (1993) ou une autre méthode doit-elle être appliquée ?

La mesure de l'environnement lumineux à de cours intervalles est contraignante. On peut cependant s'interroger sur la pertinence de cette fréquence de mesure selon les conditions expérimentales. Les résultats de mesure de variation du signal en fonction de différents paramètres peuvent ainsi servir de grille d'analyse de l'importance de la fréquence de mesure de l'environnement lumineux. La mesure de l'évolution du ζ directionnel en fonction de la distance a clairement mis en évidence une variabilité du signal entre 0 cm et 20 cm, les mesures réalisées à plus de 20 cm présentent moins de variation du ζ. Ainsi, une mesure du ζ correspondant au rayonnement renvoyé par une plante située à plus de 20 cm du capteur ne semble pas nécessiter des mesures à des intervalles de temps compris entre 5 et 15 minutes. Cependant les mesures réalisées à moins de 20 cm des plantes se trouvent dans la zone de variation du ζ et le choix d'une mesure à intervalles réguliers doit alors se faire en considérant d'autre paramètres. Les mesures du ζ directionnel en fonction de l'heure de mesure peuvent également aider au choix de l'intervalle de mesure. Les mesures de ζ en fonction de l'heure pour des plantes isolées montrent clairement une variation du ζ en fonction de l'heure de mesure. Pour les mesures sur un peuplement, les conditions

météo et le stade de développement du couvert jouent un rôle sur la variabilité du signal au cours de la journée. La synthèse des résultats (Tableau 1) met en évidence que le ζ directionnel varie au cours de la journée dans le cas d'un peuplement peu développé et pour des conditions de ciel clair. Pour d'autres stades de développement et sous ciel diffus le ζ directionnel est constant en fonction de l'heure de mesure. En conclusion, la mesure du ζ directionnel doit idéalement être effectuée à des intervalles réguliers (5-15 minutes) lorsque l'on mesure le ζ correspondant au rayonnement renvoyé par des plantes isolées à une distance inférieure à 20 cm, quelques-soient les conditions météo, et lorsque l'on mesure le ζ correspondant au rayonnement renvoyé par des peuplements au stade précoce sous condition de ciel clair.

En se basant sur la méthode de mesure proposée par Varlet-Grancher et al. (1993) et sur les variations du ζ observées, il apparaît que lorsque l'on cherche à mesurer le ζ pour des peuplements de graminées peu développées de nombreuses mesures au cours du temps sont nécessaires afin de capturer l'ensemble des valeurs de ζ perçu par les plantes au cours d'une journée. L'utilisation de ce protocole de mesure permettrait d'avoir l'ensemble des éléments nécessaire à l'établissement de lois de réponses.

Réaliser de nombreuses mesures du ζ dans un couvert végétal nécessite un dispositif difficile à mettre en place. De plus, dans le cadre de mesure directionnelle, la mesure du ζ peut nécessiter d'être réalisée pour plusieurs directions. Le dispositif MARscope permet de répondre en partie à ces besoins de caractérisation de l'environnement lumineux mais il reste limité, notamment par son incapacité à effectuer plusieurs mesures simultanées dans différentes directions. La modélisation est alors un outil pertinent dans la compréhension de la photomorphogenèse, car un nombre de mesure de rayonnement peut théoriquement être réalisé sur un nombre infini de site de perception au même instant. Bien que les modèles directionnels créés dans cette thèse nécessitent des améliorations, ils sont une étape supplémentaire dans cette démarche de compréhension.

VI. Conclusion et Perspectives

Deux principaux objectifs étaient au cœur de cette thèse : la mesure et la modélisation des variations du ζ directionnelle selon différentes conditions.

Si on pouvait s'attendre à ces variations du ζ directionnel, l'amplitude des variations n'était jusqu'alors pas connue. Cette thèse a permis de caractériser pour la première fois la gamme des variations de ζ directionnel pour un faible angle de vue. En effet, les résultats mettent clairement en évidence l'importance de l'architecture dans la variabilité du ζ directionnel en fonction de la distance. Ces variations sont plus importantes pour une plante peu développée que pour une plante développée. Pour les plantes étudiées il existe une distance seuil à partir de laquelle on observe peu de variation du ζ directionnel. Dans le cadre de la caractérisation de la variabilité du ζ directionnel, il faut donc prendre en compte ces différents éléments en interaction avec les variations identifiées pour les conditions météorologiques.

Du fait des simplifications liées à l'utilisation des modèles, on pouvait s'attendre à des différences entre les mesures et les simulations. Ces différences sont plus ou moins importantes selon les conditions de modélisations. En l'état, les simulations effectuées avec le modèle CARIBU se rapprochent le plus des mesures. Les résultats obtenus à partir de ces modèles restent cependant encourageants et de plus amples tests d'amélioration doivent être réalisés afin d'aboutir à un outil permettant de mieux comprendre les mécanismes de photomorphogenèse.

Pour réaliser ces tests d'amélioration, un travail est nécessaire sur les degrés de précision nécessaire sur les variables. Notamment sur l'architecture des plantes, les propriétés optiques le rayonnement incident. À l'issue de ces tests d'amélioration des modèles de capteurs directionnels, l'étape suivante consiste à les utiliser pour établir des lois de réponses au ζ. Ces lois de réponses pourront par la suite être intégrées dans des modèles FSP.

VII. Bibliographie

Adjesiwor AT, Kniss AR. **2020**. Light Reflected from Different Plant Canopies Affected Beta vulgaris L. Growth and Development. *Agronomy* **10**: 1771.

Ahmad M, Cashmore AR. **1993**. HY4 gene of A. thaliana encodes a protein with characteristics of a blue-light photoreceptor. *Nature* **366**: 162–166.

Baldissera TC, Frak E, Carvalho PC de F, Louarn G. **2014**. Plant development controls leaf area expansion in alfalfa plants competing for light. *Annals of Botany* **113**: 145–157.

Ballaré CL. **2017**. Phytochrome Responses: Think Globally, Act Locally. *Trends in Plant Science* **22**: 909–911.

Ballaré CL, Sánchez RA, Scopel AL, Casal JJ, Ghersa CM. **1987**. Early detection of neighbour plants by phytochrome perception of spectral changes in reflected sunlight. *Plant, Cell & Environment* **10**: 551–557.

Ballaré CL, Scopel AL, Sánchez RA. **1989**. Photomodulation of Axis Extension in Sparse Canopies: Role of the Stem in the Perception of Light-Quality Signals of Stand Density. *Plant Physiology* **89**: 1324–1330.

Ballaré CL, Scopel AL, Sanchez RA. **1990**. Far-Red Radiation Reflected from Adjacent Leaves: An Early Signal of Competition in Plant Canopies. *Science* **247**: 329–332.

Barillot R, Escobar-Gutiérrez AJ, Fournier C, Huynh P, Combes D. **2014**. Assessing the effects of architectural variations on light partitioning within virtual wheat–pea mixtures. *Annals of Botany* **114**: 725–737.

Boudon F, Pradal C, Cokelaer T, Prusinkiewicz P, Godin C. **2012**. L-Py: An L-System Simulation Framework for Modeling Plant Architecture Development Based on a Dynamic Language. *Frontiers in Plant Science* **3**.

Casal JJ. **2012**. Shade Avoidance. *The Arabidopsis Book* **2012**.

Casal JJ, Deregibus VA, Sánchez RA. **1985**. Variations in Tiller Dynamics and Morphology in Lolium multiflorum Lam.Vegetative and Reproductive Plants as affected by Differences in Red/Far-Red Irradiation. *Annals of Botany* **56**: 553–559.

Casal JJ, Sáchez RA, Deregibus VA. **1987**. Tillering Responses of Lolium multiflorum Plants to Changes of Red/Far-Red Ratio Typical of Sparse Canopies. *Journal of Experimental Botany* **38**: 1432–1439.

Casal JJ, Sanchez RA, Deregibus VA. **1986**. The effect of plant density on tillering: The involvement of R/FR ratio and the proportion of radiation intercepted per plant. *Environmental and Experimental Botany* **26**: 365–371.

Cerny TA, Faust JE, Layne DR, Rajapakse NC. **2003**. Influence of Photoselective Films and Growing Season on Stem Growth and Flowering of Six Plant Species. *Journal of the American Society for Horticultural Science* **128**: 486–491.

Chamblee DS, Collins M. **1988**. Relationships with Other Species in a Mixture In: *Alfalfa and Alfalfa Improvement*. John Wiley & Sons, Ltd, 439–461.

Chelle M. **1996**. Developpement d’un modele de radiosite mixte pour simuler la distribution du rayonnement dans les couverts vegetaux.

Chelle M, Andrieu B. **1998**. The nested radiosity model for the distribution of light within plant canopies. *Ecological Modelling* **111**: 75–91.

Chelle M, Andrieu B, Bouatouch K. **1998**. Nested radiosity for plant canopies. *The Visual Computer* **14**: 109–125.

Chelle M, Evers JB, Combes D, Varlet-Grancher C, Vos J, Andrieu B. **2007**. Simulation of the three-dimensional distribution of the red:far-red ratio within crop canopies. *New Phytologist* **176**: 223–234.

Chelle M, Hanan J, Autret H. **2004**. *FSPM04 Lighting virtual crops: the CARIBU solution for open L-systems*.

Chiang C, Olsen JE, Basler D, Bånkestad D, Hoch G. **2019**. Latitude and Weather Influences on Sun Light Quality and the Relationship to Tree Growth. *Forests* **10**: 610.

Combes D, Chelle M, Sinoquet H, Varlet-Grancher C. **2008**. Evaluation of a turbid medium model to simulate light interception by walnut trees (hybrid NG38 × RA and Juglans regia) and sorghum canopies (Sorghum bicolor) at three spatial scales. *Functional Plant Biology* **35**: 823–836.

Combes D, Sinoquet H, Varlet-Grancher C. **2000**. Preliminary measurement and simulation of the spatial distribution of the Morphogenetically Active Radiation (MAR) within an isolated tree canopy. *Annals of Forest Science* **57**: 497–511.

Cordukes WE, Fisher JE. **1974**. Effects of Shading of the Leaf Sheath on the Growth and Development of the Tiller Stems of Kentucky Bluegrass. *Canadian Journal of Plant Science* **54**: 47–53.

Crepy MA, Casal JJ. **2015**. Photoreceptor-mediated kin recognition in plants. *New Phytologist* **205**: 329–338.

Davis MH, Simmons SR. **1994**. Far-red light reflected from neighbouring vegetation promotes shoot elongation and accelerates flowering in spring barley plants. *Plant, Cell & Environment* **17**: 829–836.

Demotes-Mainard S, Péron T, Corot A, *et al.* 2016. Plant responses to red and far-red lights, applications in horticulture. *Environmental and Experimental Botany* **121**: 4–21.

Deregibus VA, Sanchez RA, Casal JJ. **1983**. Effects of Light Quality on Tiller Production in Lolium spp. *Plant Physiology* **72**: 900–902.

Dulk JA den. **1989**. The interpretation of remote sensi

Evers JB, Vos J, Andrieu B, Struik PC. **2006**. Cessation of Tillering in Spring Wheat in Relation to Light Interception and Red : Far-red Ratio. *Annals of Botany* **97**: 649–658.

Evers JB, Vos J, Chelle M, Andrieu B, Fournier C, Struik PC. **2007**. Simulating the effects of localized red:far-red ratio on tillering in spring wheat (Triticum aestivum) using a three-dimensional virtual plant model. *New Phytologist* **176**: 325–336.

Faverjon L. **2018**. Calibration et évaluation d'un modèle individu-centré générique de morphogenèse des légumineuses fourragères. Application à la prédiction des équilibres inter-spécifiques dans des communautés prairiales expérimentales.

Finlayson SA, Krishnareddy SR, Kebrom TH, Casal JJ. **2010**. Phytochrome Regulation of Branching in Arabidopsis. *Plant Physiology* **152**: 1914–1927.

Franssen JM, Bruinsma J. **1981**. Relationships between xanthoxin, phototropism, and elongation growth in the sunflower seedling Helianthus annuus L. *Planta* **151**: 365–370.

Fraser DP, Hayes S, Franklin KA. **2016**. Photoreceptor crosstalk in shade avoidance. *Current Opinion in Plant Biology* **33**: 1–7.

Frazer GW, Trofymow JA, Lertzman KP. **1997**. *A method for estimating canopy openness, effective leaf area index, and photosynthetically active photon flux density using hemispherical photography and computerized image analysis techniques*.

Gastal F, Julier B, Surault F, *et al.* 2012. Intérêt des prairies cultivées multiespèces dans le contexte des systèmes de polyculture-élevage. *Innovations Agronomiques* **22**: 169.

Gautier H, Měch R, Prusinkiewicz P, Varlet-Grancher C. **2000**. 3D Architectural Modelling of Aerial Photomorphogenesis in White Clover (Trifolium repens L.) using L-systems. *Annals of Botany* **85**: 359–370.

Gautier H, Varlet-grancher C, Baudry N. **1997**. Effects of Blue Light on the Vertical Colonization of Space by White Clover and their Consequences for Dry Matter Distribution. *Annals of Botany* **80**: 665–671.

Gautier H, Varlet-Grancher C, Hazard L. **1999**. Tillering Responses to the Light Environment and to Defoliation in Populations of Perennial Ryegrass (Lolium perenne L.) Selected for Contrasting Leaf Length. *Annals of Botany* **83**: 423–429.

Gillet M. **1980**. *Les graminées fourragères description, fonctionnement, applications à la culture de l'herbe.* Paris: Gauthier-Villars.

Gruntman M, Groß D, Májeková M, Tielbörger K. **2017**. Decision-making in plants under competition. *Nature Communications* **8**: 2235.

Harper JL. **1977**. *Population biology of plants*. London: Acad. Pr.

Hay MJM, Newton PCD, Robin C, Cresswell A. **2001**. Branching responses of a plagiotropic clonal herb to localised incidence of light simulating that reflected from vegetation. *Oecologia* **127**: 185–190.

Hectors K, Prinsen E, De Coen W, Jansen MAK, Guisez Y. **2007**. Arabidopsis thaliana plants acclimated to low dose rates of ultraviolet B radiation show specific changes in morphology and gene expression in the absence of stress symptoms. *New Phytologist* **175**: 255–270.

Heo J, Lee C, Chakrabarty D, Paek K. **2002**. Growth responses of marigold and salvia bedding plants as affected by monochromic or mixture radiation provided by a Light-Emitting Diode (LED). *Plant Growth Regulation* **38**: 225–230.

Heraut-Bron V, Robin C, Varlet-Grancher C, Afif D, Guckert A. **2000**. Light quality (red:far-red ratio): does it affect photosynthetic activity, net CO2 assimilation, and morphology of young white clover leaves? *Canadian Journal of Botany* **77**: 1425–1431.

Héraut-Bron V, Robin C, Varlet-Grancher C, Guckert A. **2001**. Phytochrome Mediated Effects on Leaves of White Clover: Consequences for Light Interception by the Plant under Competition for Light. *Annals of Botany* **88**: 737–743.

Hewitson J, Dixon T, Price R. **2000**. A demonstration of communication between plants. *Journal of Biological Education* **34**: 160–162.

Holmes MG, Smith H. **1977**. The Function of Phytochrome in the Natural Environment—Iv. Light Quality and Plant Development. *Photochemistry and Photobiology* **25**: 551–557.

Huyghe C, De Vliegher A, van Gils B. **2014**. *Grasslands and herbivore production in Europe and effects of common policies*.

Huyghe C, Peeters A, De Vliegher A. **2015**. La prairie en France et en Europe In: Adaptation des prairies semées au changement climatique. *Colloque présentant les méthodes et résultats du projet Climagie (métaprogramme ACCAF)*. Poitiers, France: INRA, 223 p.

Jenkins GI. **2014**. The UV-B Photoreceptor UVR8: From Structure to Physiology. *The Plant Cell* **26**: 21–37.

Justes E, Bedoussac L, Dordas C, *et al.* 2021. The 4C approach as a way to understand species interactions determining intercropping productivity. *Frontiers of Agricultural Science and Engineering* **8**: 387–399.

Kahlen K, Stützel H. **2011**. Modelling photo-modulated internode elongation in growing glasshouse cucumber canopies. *New Phytologist* **190**: 697–708.

Kahlen K, Wiechers D, Stützel H. **2008**. Modelling leaf phototropism in a cucumber canopy. *Functional Plant Biology* **35**: 876–884.

Kasperbauer MJ. **1971**. Spectral Distribution of Light in a Tobacco Canopy and Effects of End-of-Day Light Quality on Growth and Development. *Plant Physiology* **47**: 775–778.

Kasperbauer MJ, Karlen DL. **1994**. Plant Spacing and Reflected Far-Red Light Effects on Phytochrome-Regulated Photosynthate Allocation in Corn Seedlings. *Crop Science* **34**: 1564–1569.

Kendrick RE, Kronenberg GHM (Eds.). **1994**. *Photomorphogenesis in Plants*. Springer Netherlands.

Kotilainen T, Aphalo PJ, Brelsford CC, *et al.* 2020. Patterns in the spectral composition of sunlight and biologically meaningful spectral photon ratios as affected by atmospheric factors. *Agricultural and Forest Meteorology* **291**: 108041.

Kozuka T, Horiguchi G, Kim G-T, Ohgishi M, Sakai T, Tsukaya H. **2005**. The Different Growth Responses of the Arabidopsis thaliana Leaf Blade and the Petiole during Shade Avoidance are Regulated by Photoreceptors and Sugar. *Plant and Cell Physiology* **46**: 213–223.

Kruk B, Insausti P, Razul A, Benech-Arnold R. **2006**. Light and thermal environments as modified by a wheat crop: effects on weed seed germination. *Journal of Applied Ecology* **43**: 227–236.

Kwesiga F, Grace J. **1986**. The Role of the Red/Far-Red Ratio in the Response of Tropical Tree Seedlings to Shade. *Annals of Botany* **57**: 283–290.

Lafarge M, Durand J-L. **2011**. *Comment l'herbe pousse: développement végétatif, structures clonales et spatiales des graminées*. Versailles: Éd. Quae.

Lecharny A. **1979**. Phytochrome and internode elongation in Chenopodium polyspermum L. sites of photoreception. *Planta* **145**: 405–409.

Lee DW, Baskaran K, Mansor M, Mohamad H, Yap SK. **1996**. Irradiance and Spectral Quality Affect Asian Tropical Rain Forest Tree Seedling Development. *Ecology* **77**: 568–580.

Lemaire G, Avice J-C, Kim T-H, Ourry A. **2005**. Developmental changes in shoot N dynamics of lucerne (Medicago sativa L.) in relation to leaf growth dynamics as a function of plant density and hierarchical position within the canopy. *Journal of Experimental Botany* **56**: 935–943.

Li B, Shibuya T, Yogo Y, Hara T, Matsuo K. **2001**. Effects of light quantity and quality on growth and reproduction of a clonal sedge, Cyperus esculentus. *Plant Species Biology* **16**: 69–81.

Libenson S, Rodriguez V, Pereira ML, Sánchez RA, Casal JJ. **2002**. Low Red to Far-Red Ratios Reaching the Stem Reduce Grain Yield in Sunflower. *Crop Science* **42**: 1180–1185.

Lin C, Yang H, Guo H, Mockler T, Chen J, Cashmore AR. **1998**. Enhancement of blue-light sensitivity of Arabidopsis seedlings by a blue light receptor cryptochrome 2. *Proceedings of the National Academy of Sciences* **95**: 2686–2690.

Loiseau P, Carrère P, Lafarge M, Delpy R, Dublanchet J. **2001**. Effect of soil-N and urine-N on nitrate leaching under pure grass, pure clover and mixed grass/clover swards. *European Journal of Agronomy* **14**: 113–121.

Louarn G, Corre-Hellou G, Fustec J, *et al.* 2010. Déterminants écologiques et physiologiques de la productivité et de la stabilité des associations graminées-légumineuses. *Innovations Agronomiques* **11**: 79–99.

Louarn G, Faverjon L. **2018**. A generic individual-based model to simulate morphogenesis, C–N acquisition and population dynamics in contrasting forage legumes. *Annals of Botany* **121**: 875–896.

Louarn G, Faverjon L, Bijelić Z, Julier B. **2016**. Dynamique de l'azote dans les associations graminées -légumineuses : quels leviers pour valoriser l'azote fixé ? *Fourrages* **226**: 135–142.

Maddonni GA, Otegui ME, Andrieu B, Chelle M, Casal JJ. **2002**. Maize Leaves Turn Away from Neighbors. *Plant Physiology* **130**: 1181–1189.

Mauchamp L, Gillet F, Mouly A, Badot P-M. **2020**. IV. Les services écosystémiques des prairies In: Pratiques & techniques. *Les prairies : biodiversité et services systémiques*. Besançon: Presses universitaires de Franche-Comté, 61–82.

Mazliak P. **1982**. Le Phytochrome In: Methodes. *Physiologie végétale II - Croissance et développement*. Hermann, 91–127.

Mohr H. **1972**. *Lectures on photomorphogenesis*. Berlin [West], Heidelberg, New York: Springer.

Monteith JL. **1976**. Spectral Distribution of Light in Leaves and Foliage In: *Light and Plant Development*. London ; Boston: Butterworths, 447–460.

Morgan DC, Smith H. **1976**. Linear relationship between phytochrome photoequilibrium and growth in plants under simulated natural radiation. *Nature* **262**: 210.

Morgan DC, Smith H. **1978**. The relationship between phytochrome-photoequilibrium and Development in light grown Chenopodium album L. *Planta* **142**: 187–193.

Novoplansky A. **2009**. Picking battles wisely: plant behaviour under competition. *Plant, Cell & Environment* **32**: 726–741.

Pagès L, Bécel C, Boukcim H, Moreau D, Nguyen C, Voisin A-S. **2014**. Calibration and evaluation of ArchiSimple, a simple model of root system architecture. *Ecological Modelling* **290**: 76–84.

Pantazopoulou CK, Bongers FJ, Kupers JJ, *et al.* 2017. Neighbor detection at the leaf tip adaptively regulates upward leaf movement through spatial auxin dynamics. *Proceedings of the National Academy of Sciences of the United States of America* **114**: 7450–7455.

Pedmale UV, Huang SC, Zander M, *et al.* 2016. Cryptochromes Interact Directly with PIFs to Control Plant Growth in Limiting Blue Light. *Cell* **164**: 233–245.

Perrot C, Durand J-L, Ghesquière M. **2022**. *SICLEX simulateur de climat extrême - Un outil pour étudier l'adaptation des prairies au changement climatique*. https://www.inrae.fr/sites/default/files/pdf/SICLEX%202022%20VF_web.pdf.

Pradal C, Dufour-Kowalski S, Boudon F, Fournier C, Godin C. **2008**. OpenAlea: a visual programming and component-based software platform for plant modelling. *Functional Plant Biology* **35**: 751–760.

Prieto I, Violle C, Barre P, Durand J-L, Ghesquiere M, Litrico I. **2015**. Complementary effects of species and genetic diversity on productivity and stability of sown grasslands. *Nature Plants* **1**: 1–5.

Procko C, Crenshaw CM, Ljung K, Noel JP, Chory J. **2014**. Cotyledon-Generated Auxin Is Required for Shade-Induced Hypocotyl Growth in Brassica rapa. *Plant Physiology* **165**: 1285–1301.

Prusinkiewicz P, Lindenmayer A. **1990**. *The Algorithmic Beauty of Plants*.

Rajapakse NC, Kelly JW. **1994**. Influence of spectral filters on growth and postharvest quality of potted miniature roses. *Scientia Horticulturae* **56**: 245–255.

Reed JW, Nagpal P, Poole DS, Furuya M, Chory J. **1993**. Mutations in the gene for the red/far-red light receptor phytochrome B alter cell elongation and physiological responses throughout Arabidopsis development. *The Plant Cell* **5**: 147–157.

Robin Ch, Hay MJM, Newton PCD. **1994**. Effect of light quality (red: far-red ratio) and defoliation treatments applied at a single phytomer on axillary bud outgrowth in Trifolium repens L. *Oecologia* **100**: 236–242.

Rosen MA, Hooper FC. **1989**. A comparison of two models for the angular distribution of diffuse sky radiance for overcast skies. *Solar Energy* **42**: 477–482.

Sasidharan R, Chinnappa CC, Staal M, *et al.* 2010. Light Quality-Mediated Petiole Elongation in Arabidopsis during Shade Avoidance Involves Cell Wall Modification by Xyloglucan Endotransglucosylase/Hydrolases. *Plant Physiology* **154**: 978–990.

Sattin M, Zuin MC, Sartorato I. **1994**. Light quality beneath field-grown maize, soybean and wheat canopies – red:far red variations. *Physiologia Plantarum* **91**: 322–328.

Seavers GP, Smith H. **1997**. The reflectance properties of plant internodes modify elongation responses to lateral far-red radiation. *Plant, Cell & Environment* **20**: 1372–1380.

Sinoquet H, Bonhomme R. **1992**. Modeling radiative transfer in mixed and row intercropping systems. *Agricultural and Forest Meteorology* **62**: 219–240.

Sinoquet H, Rakocevic M, Varlet-Grancher C. **2000**. Comparison of models for daily light partitioning in multispecies canopies. *Agricultural and Forest Meteorology* **101**: 251–263.

Skinner RH, Nelson CJ. **1995**. Elongation of the Grass Leaf and its Relationship to the Phyllochron. *Crop Science* **35**: cropsci1995.0011183X003500010002x.

Skinner RH, Simmons SR. **1993**. Modulation of leaf elongation, tiller appearance and tiller senescence in spring barley by far-red light. *Plant, Cell & Environment* **16**: 555–562.

Smith H. **1982**. Light Quality, Photoperception, and Plant Strategy. *Annual Review of Plant Physiology* **33**: 481–518.

Smith H, Casal JJ, Jackson GM. **1990**. Reflection signals and the perception by phytochrome of the proximity of neighbouring vegetation. *Plant, Cell & Environment* **13**: 73–78.

Smith H, Holmes MG. **1977**. The Function of Phytochrome in the Natural Environment—III. Measurement and Calculation of Phytochrome Photoequilibria. *Photochemistry and Photobiology* **25**: 547–550.

Smith JD, Johnson BI, Mescher MC, Moraes CMD. **2020**. A plant parasite uses light cues to detect differences in host-plant proximity and architecture. *Plant, Cell & Environment* **44**: 1142–1150.

Sparkes DL, Holme SJ, Gaju O. **2006**. Does light quality initiate tiller death in wheat? *European Journal of Agronomy* **24**: 212–217.

Sparkes DL, King M. **2008**. Disentangling the effects of PAR and R:FR on lodging-associated characters of wheat (Triticum aestivum). *Annals of Applied Biology* **152**: 1–9.

Tao Y, Ferrer J-L, Ljung K, *et al.* 2008. Rapid Synthesis of Auxin via a New Tryptophan-Dependent Pathway Is Required for Shade Avoidance in Plants. *Cell* **133**: 164–176.

Thomas RG. **2003**. Comparative growth forms of dryland forage legumes. *NZGA: Research and Practice Series* **11**: 19–25.

Thompson L. **1995**. Sites of photoperception in white clover. *Grass and Forage Science* **50**: 259–262.

Thornley JHM, Bergelson J, Parsons AJ. **1995**. Complex dynamics in a carbon-nitrogen model of a grass-legume pasture. *Annals of Botany* **75**: 79–94.

Turkington R. **1989**. The Growth, Distribution and Neighbour Relationships of Trifolium Repens in a Permanent Pasture. VI. Conditioning Effects by Neighbours. *Journal of Ecology* **77**: 734–746.

Turnbull MH, Yates DJ. **1993**. Seasonal variation in the red/far-red ratio and photon flux density in an Australian sub-tropical rainforest. *Agricultural and Forest Meteorology* **64**: 111–127.

Ugarte CC, Trupkin SA, Ghiglione H, Slafer G, Casal JJ. **2010**. Low red/far-red ratios delay spike and stem growth in wheat. *Journal of Experimental Botany* **61**: 3151–3162.

Varlet Grancher C, Gautier H. **1995**. Plant morphogenetic responses to light quality and consequences for intercropping In: Guadeloupe: Institut National de la Recherche Agronomique, 231–256.

Varlet-Grancher C, Moulia B, Sinoquet H, Russell G. **1993**. Spectral modification of light within plant canopies : how to quantify its effects on the architecture of the plant stand In: Varletgrancher C, Bonhomme R, Sinoquet H, eds. *Crop Structure and Light Microclimate*

Waddington DV, Carrow RN, Shearman RC, Dudeck AE, Peacock CH. **1992**. Shade and Turfgrass Culture In: *Agronomy Monograph*. American Society of Agronomy, Crop Science Society of America, Soil Science Society of America, .

Wan C, Sosebee RE. **1998**. Tillering responses to red:far-red light ratio during different phenological stages in Eragrostis curvula. *Environmental and Experimental Botany* **40**: 247–254.

Wang Q, Lin C. **2020**. Mechanisms of Cryptochrome-Mediated Photoresponses in Plants. *Annual Review of Plant Biology* **71**: 103–129.

Wargent JJ, Gegas VC, Jenkins GI, Doonan JH, Paul ND. **2009**. UVR8 in Arabidopsis thaliana regulates multiple aspects of cellular differentiation during leaf development in response to ultraviolet B radiation. *New Phytologist* **183**: 315–326.

Wedin D, Tilman D. **1993**. Competition Among Grasses Along a Nitrogen Gradient: Initial Conditions and Mechanisms of Competition. *Ecological Monographs* **63**: 199–229.

Wherley BG, Gardner DS, Metzger JD. **2005**. Tall Fescue Photomorphogenesis as Influenced by Changes in the Spectral Composition and Light Intensity. *Crop Science* **45**: 562–568.

Whitelam GC, Johnson CB. **1982**. Photomorphogenesis in Impatiens Parviflora and Other Plant Species Under Simulated Natural Canopy Radiations. *New Phytologist* **90**: 611–618.

Wyszecki G, Stiles WS. **2000**. *Color science: concepts and methods, quantitative data, and formulae*. New York: John Wiley & Sons.

Xie Q, Mayes S, Sparkes DL. **2016**. Optimizing tiller production and survival for grain yield improvement in a bread wheat × spelt mapping population. *Annals of Botany* **117**: 51–66.

Zhang L, Liu X, Zhou S, Shipley B. **2021**. Explaining variation in productivity requires intraspecific variability in plant height among communities. *Journal of Plant Ecology*.

Zhen S, Bugbee B. **2020**. Far-red photons have equivalent efficiency to traditional photosynthetic photons: Implications for redefining photosynthetically active radiation. *Plant, Cell & Environment* **43**: 1259–1272.

Zhu J, Vos J, van der Werf W, van der Putten PEL, Evers JB. **2014**. Early competition shapes maize whole-plant development in mixed stands. *Journal of Experimental Botany* **65**: 641–653.

VIII. Annexes

Les scripts ont été développés à l'aide du logiciel PyCharm. Il se peut que certaines fonctions puissent ne fonctionner que dans cet environnement.

zeta_pixel_multisim_py37.py

Script utilisé pour le modèle directionnel simple basé sur la SVC.

```
import os
import sys
import multiprocessing
from glob import glob
from numpy import sign, tan, radians
from pandas import read_csv
from time import strftime, time, gmtime
import openalea.plantgl.all as pgl
import create_directional_sensor as ds


def zeta_pixel(obj_path, x=100, y=0, z=25, fov=3, area_name="area.csv", timing=time(), i=None, imax=100,
z_rot=180, frame_size=1000):
    i = imax - 1 if i is None else i
    x = float(x) if isinstance(x, float) is False else x
    y = float(y) if isinstance(y, float) is False else y
    z = float(z) if isinstance(z, float) is False else z
    perspective = True

    # Recuperation des donnees de mesure
    if "ArchiMono" in obj_path:
        manip = ds.data_archimono(obj_path, x)
    elif "ArchiDuo" in obj_path:
        manip = ds.data_archiduo(obj_path)
    else:
        sys.exit("Script pas encore prevu pour cette experimentation")

    date_mesure  = manip["date"]
    heure_mesure = manip["heure"]
    semis        = manip["semis"]
    rep          = manip["rep"]
    zeta_manip   = manip["zeta_maya"]

    # Creation scene
    sensor_pos = (int(sign(90 - z_rot)) * int(x), y, z)
    s_archi = pgl.Scene()
    s_archi += ds.shapepgl(pgl.Scene(r" + obj_path)[0].geometry, rgb=(0, 255, 0, 0), z_rot=z_rot, id=1312)

    # Circle radius mimicking the area seen by the real sensor
    cache_path = r"C:\Users\Arthur\Documents\Couturier Arthur\These\Experimentation\ArchiMono\Scan3D\
Cache.obj"
    cache_trans = (sensor_pos[0] + int(sign(-90 + z_rot)) * (0.5 / tan(radians(fov / 2.))), sensor_pos[1], sensor_pos[2])  #
rotation 180 degrees
    s_archi += ds.shapepgl(pgl.Scene(cache_path)[0].geometry, trans=cache_trans, id=2)

    area_plt = ds.zeta_area(s_archi, sensor_pos, fov, perspective=perspective, frame_size=frame_size)

    # Output
    screenshot_folder = os.path.dirname(obj_path) + 'Scene/screenshot/' + 'fov' + str(fov) + '/' +
```

```
os.path.basename(obj_path)
    if os.path.exists(screenshot_folder) is False:
        os.makedirs(screenshot_folder)
    pgl.Viewer.frameGL.saveImage(screenshot_folder + '/' + str(x / 10) + 'cm.bmp')

    if not os.path.exists(os.path.dirname(area_name)):
        os.makedirs(os.path.dirname(area_name))

    with open(area_name, 'a') as output_area:
        if os.path.exists(area_name) is False:
            output_area.write('date;heure;semis;rep;distance;fov;'
                              'camera;zeta_manip;'
                              'percent_plant_area;px_plt;zeta_px\n')
        output_list = (date_mesure, heure_mesure, semis, rep, x / 10, fov,
                       "perspective" if perspective else "orthographic", zeta_manip,
                       area_plt['percent_plant'], area_plt['px_plt'], area_plt['zeta_px'])
        output_area.write((("%s;" * len(output_list))[:-1] + "\n") % output_list)

    i.value += 1
    runtime = time() - timing
    print_runtime = strftime('done in %Hh %Mmin %Ss', gmtime(runtime))
    ds.print_progress_bar(i.value, imax, suffix=print_runtime)


if __name__ == '__main__':
    debug = False if sys.gettrace() is None else True  # Test if pycharm is in debug mode
    timing = time()
    starttime = strftime("%Y-%m-%d_%HH%M")

    fov = 3
    experimentation = 'ArchiMono'  # 'ArchiDuo'  #
    reconstruction = 'Ball Pivoting'  # 'Marching Cubes'
    recalage = False
    path_these = os.path.join(os.path.expanduser('~'), 'Documents', 'Couturier Arthur', 'These')
    path_scans = glob(os.path.join(path_these, 'Experimentation', experimentation, 'Scan*3D'))[0]

    if experimentation == "ArchiMono":
        list_obj = glob(path_scans + "/**/" + reconstruction + "/**/*.obj", recursive=True)
    elif experimentation == "ArchiDuo":
        list_obj = glob(path_scans + "/**/2020-10-29/**/obj apss/*.obj", recursive=True)

    if len(list_obj) == 0:
        sys.exit("Probleme de chemin de fichier dans " + experimentation)

    area_name = os.path.join(path_scans, 'output', 'area', reconstruction, starttime + '_scan_archi_' + reconstruction +
'.csv')

    # Test avec recalage
    if recalage:
        recalage_path = glob(path_these + '/Bordeaux/**/recalage_brute_force.csv', recursive=True)[0]
        df_recalage = read_csv(recalage_path, sep=";")
        list_obj = [obj_name for obj_name in list_obj if any(obj_decal in obj_name for obj_decal in
list(df_recalage.scan))]  # filtre les scans
        area_name = area_name.replace(".csv", "_recalage.csv")

    # Multiprocessing
    CPUnb = 10  # nombre de processeurs utilises
    multiprocessing.freeze_support()  # aucune idee de l'utilite, repris de l-egume
```

```
    pool = multiprocessing.Pool(processes=CPUnb)
    it_nb = multiprocessing.Manager().Value('i', 0)  # equilavent de global pour multiprocessing

    print(starttime + " " + experimentation + " " + reconstruction + "\n" + area_name)
    print("Simulation sur " + str(CPUnb) + " processeurs")
    for obj_path in list_obj:
        if experimentation == "ArchiMono":
            distances = list(range(20, 160, 10)) + list(range(160, 600, 40))
            rotx, roty, rotz = 0, 0, 180
            sensor_diam = [1.5]
            z = 25

        elif experimentation == "ArchiDuo":
            distances = [int(os.path.basename(obj_path).split('_')[3])]
            rotx, roty, rotz = 90, 0, 0
            sensor_diam = [1.5]
            z = 33

        it_max = len(list_obj) * len(distances) * len(sensor_diam)
        for sd in sensor_diam:
            for x in distances:
                if recalage:
                    ech = os.path.basename(obj_path[:-4].replace('cor360_', ''))
                    y = float(df_recalage[df_recalage.scan == ech].translation_y)
                    rotz = 180 + float(df_recalage[df_recalage.scan == ech].rotation_z)
                    scale = 1  # float(df_recalage[df_recalage.scan == ech].scale) * 10
                else:
                    y = 0
                    scale = 1

                if debug:
                    print("Debug")
                    zeta_pixel(obj_path, x=x, y=y, z=z, fov=fov, area_name=area_name, timing=timing, i=it_nb,
imax=it_max, z_rot=rotz, frame_size=800)
                else:
                    if CPUnb == 1:
                        result = pool.apply_async(zeta_pixel, args=(obj_path, x, y, z, fov, area_name, timing, it_nb, it_max,
rotz))
                        result.get()  # permet de recuperer les messages d'erreurs du multisim lors d'un bug
                    else:
                        pool.apply_async(zeta_pixel, args=(obj_path, x, y, z, fov, area_name, timing, it_nb, it_max, rotz))
    pool.close()
    pool.join()
    print("\n\nSimulation terminee a", strftime("%HH%M"), "en", int((time() - timing) / 60), "minutes")
```

Calcul_Zeta_scanner_multisim_py37.py

Script permettant de simuler le ζ à partir de CARIBU.

```
import os
import sys
import pandas
import multiprocessing
from shutil import copy
from glob import glob
from numpy import sign
from time import strftime, time
from alinea.caribu.CaribuScene import CaribuScene
import openalea.plantgl.all as pgl
import create_directional_sensor as ds


def zeta_caribu(obj_path, expe, x=100.0, y=0.0, z=25.0, fov=3.0, sensor_diam=1.5,
                caribu_name="caribu.csv", timing=time(), i=None, imax=100,
                x_rot=0.0, y_rot=0.0, z_rot=180.0, scale=None, journalier=False):
    """
    obj_path - chemin du scan
    x - position x du capteur
    y - position y du capteur
    z - position z du capteur
    fov - champ de vision, en degrees
    sensor_diam - diametre du capteur
    caribu_name - nom du fichier de sortie
    timing - début de la simulation
    i - numero de la simulation
    imax - nombre total de simulations a effectuer
    x_rot - rotation x du scan
    y_rot - rotation y du scan
    z_rot - rotation z du scan
    scale - modification de la taille du scan
    """
    i = imax - 1 if i is None else i

    # Recuperation des donnees de mesure
    if "ArchiMono" in obj_path:
        param = ds.data_archimono(obj_path, x)
        ra = 180  # angle de correction de l'azimut
    elif "ArchiDuo" in obj_path or "Test" in obj_path:
        param = ds.data_archiduo(obj_path)
        ra = None
    else:
        try:
            infos_obj = os.path.basename(obj_path.replace(".obj", "")).split('_')
            param = {"date": infos_obj[1], "heure": strftime("%H:%M:%OS"), "semis": "_".join((infos_obj[2:5])), "rep":
infos_obj[2], "zeta_maya": "NA"}
        except:
            param = {"date": "2020-11-26", "heure": strftime("%H:%M:%OS"), "semis": "NA", "rep": "NA", "zeta_maya":
"NA"}

        ra = None

    side = int(sign(90 - z_rot))
```

```
    x, y, z = float(x), float(y), float(z)
    sensor_pos = (side * x, y, z)

    # Creation scene
    s_archi = pgl.Scene()
    x_rot, y_rot, z_rot = float(x_rot), float(y_rot), float(z_rot)
    s_archi += ds.shapepgl(pgl.Scene(r" + obj_path)[0].geometry, rgb=(0, 255, 0, 0), x_rot=x_rot, y_rot=y_rot,
z_rot=z_rot, scale=scale, id=1312)
    if len(s_archi) == 0:
        print("\n\n\nWARNING: Pas de plante dans la scene\n\n\n")

    # Wavelength red (rc) and far-red (rs)
    opt = {'rc': {}, 'rs': {}}
    count_sh = 0
    rc = (0.15, 0.15)  # (reflectance, transmitance)
    rs = (0.45, 0.45)
    tess = pgl.Tesselator()

    # Plant
    s_archi_sensor = pgl.Scene()
    for sh in s_archi:
        opt['rc'][count_sh] = rc
        opt['rs'][count_sh] = rs
        sh.geometry.apply(tess)
        s_archi_sensor.add(pgl.Shape(sh.geometry, sh.appearance, id=count_sh))
        count_sh += 1
    if len(s_archi_sensor) == 0:
        param["semis"], param["rep"] = "Blanc", "1"

    if expe == "ArchiMono":
        # Ajout d'un sol neutre
        s_archi_sensor += ds.shapepgl(pgl.Disc(100), rgb=(139, 69, 19, 0), id=count_sh)
        opt['rc'][count_sh] = (10 ** -15, 10 ** -15)
        opt['rs'][count_sh] = (10 ** -15, 10 ** -15)
        count_sh += 1

        # Arriere du pot, la rotation met la face superieure vers la plante
        pot = []
        for trgl in pgl.Scene('pot_archimono.vgx'):
            zoom = (1000, 1200, 900)
            pot.append(ds.shapepgl(trgl.geometry, id=count_sh, scale=zoom, z_rot=z_rot, trans=(side * -45, side * -
zoom[1] / 40, 0), rgb=(230, 230, 230, 0)))
            opt['rc'][count_sh] = (0.90, 0.30)
            opt['rs'][count_sh] = (0.90, 0.30)
            count_sh += 1
        s_archi_sensor += pgl.Scene(pot)

    if expe == "ArchiDuo" or expe == "cj":
        # Arriere du pot, la rotation met la face superieure vers la plante
        s_archi_sensor += ds.shapepgl(ds.quad_cyl(120, 27.5), y_rot=90, rgb=(150, 150, 150, 0), id=count_sh)
        po_pot = (0.1, 0.0)
        opt['rc'][count_sh] = po_pot
        opt['rs'][count_sh] = po_pot
        count_sh += 1

        # Ajout plaque
        plaque_pos = [-side * (600 - x), 0, 450 / 2]
        s_archi_sensor += ds.shapepgl(ds.pgl_square(1250, 450), trans=plaque_pos, y_rot=-90, rgb=(50, 50, 50, 0),
```

```
id=count_sh)
        po_plaque = (0.1, 0.0)
        opt['rc'][count_sh] = po_plaque
        opt['rs'][count_sh] = po_plaque
        count_sh += 1

    # po_collim = (10 ** -25, 0, 0, 0)
    po_collim = (1, 0, 0, 1)
    po_sensor = (0.01, 1, 1, 0.01)
    vs = True

    s_archi_sensor, opt, sensor_dico, sensor_id, sensor_pos = ds.create_sensor(scene=s_archi_sensor, opt=opt,
id=count_sh,
                                                               po_collim=po_collim, po_sensor=po_sensor,
                                                               position=sensor_pos, fov=fov, diameter=sensor_diam,
                                                               virtual_sensor=vs)

    debug = False if sys.gettrace() is None else True  # Test if pycharm is in debug mode
    if debug:
        print("debug")
        pgl.Viewer.frameGL.setSize(800, 800)
        pgl.Viewer.display(s_archi_sensor)  # scene vide: pgl.Scene()
        # pgl.Viewer.camera.setViewAngle(fov / 2.)  # Plantgl viewer use radius
        # pgl.Viewer.camera.set(sensor_pos, 0, 0)
        # pgl.Viewer.camera.lookAt((0, 0, z))

    # Caribu computation
    sky = 'soc'  # 'uoc'  #
    rayonnement = ["direct", "diffus"] if sky == 'soc' else ['diffus']
    liste_heures = [str(h) + ":00:00" for h in range(17, 18)] if journalier else [param["heure"]]
    if journalier:
        caribu_name = os.path.splitext(caribu_name)[0] + "_zeta_journalier" + os.path.splitext(caribu_name)[1]
    imax = imax * len(rayonnement) * len(liste_heures)
    for heure in liste_heures:
        param["heure"] = heure
        for ray in rayonnement:
            ds.print_progress_bar(i.value, imax, start_time=timing, print_end="|" + param["semis"] + " " + ray)
            nb_az, nb_ze = 12, 6  # 4, 5  #
            pat = (-600, -1250 / 2, 600, 1250 / 2)  # defini par la position et la taille de la plaque

            light_dir, azimut, elevation = ds.caribu_light(param["date"], param["heure"], ray, energy=1000.0,
azimuths_nb=nb_az, zeniths_nb=nb_ze, diffuse_model=sky, rectify_azimuth=ra)

            by_direction = False
            nlight = 0
            for light in light_dir:
                caribu_time = time()
                if by_direction is False:
                    light = light_dir[0] if len(light_dir) == 1 else ("NA", ("NA", "NA", "NA"))
                c_scene = CaribuScene(scene=s_archi_sensor, light=[light] if by_direction else light_dir, opt=opt,
scene_unit="m", pattern=pat)
                raw, aggregated = c_scene.run(direct=False, split_face=True, screen_size=12800, sensors=sensor_dico)  #
avec rediff
                # raw, aggregated = c_scene.run(direct=True, split_face=True, screen_size=12800, sensors=sensor_dico)  #
sans rediff

                if not os.path.exists(os.path.dirname(caribu_name)):
                    os.makedirs(os.path.dirname(caribu_name))
```

```
            with open(caribu_name, 'a') as output_caribu:
                if i.value == 0 and nlight == 0:
                    output_caribu.write('obj_name;date;heure;semis;rep;distance;fov;sensor_diameter;rotation_z;'
                                        'ray;sky;azimut;elevation;azimuths_nb;zeniths_nb;'
                                        'energy;light_pos_x;light_pos_y;light_pos_z;'
                                        'virtual_sensor;'
                                        'rc_plant;rs_plant;po_sensor;po_collim;zeta_manip;'
                                        'rc_ei;rs_ei;'
                                        'rc_ei0;rs_ei0;rc_ei_sup;rs_ei_sup;rc_ei_inf;rs_ei_inf;rc_eabs;rs_eabs;'
                                        'runtime\n')

            with open(caribu_name, 'a') as output_caribu:
                aggreg_rc = aggregated["rc"]["sensors"] if "sensors" in aggregated["rc"] else aggregated["rc"]
                aggreg_rs = aggregated["rs"]["sensors"] if "sensors" in aggregated["rs"] else aggregated["rs"]
                output_list = (obj_path, param["date"], param["heure"], param["semis"], param["rep"], x, fov,
sensor_diam, z_rot,
                               ray, sky, azimut, elevation, nb_az, nb_ze, light[0], light[1][0], light[1][2], light[1][2],
                               vs, rc, rs, "NA" if vs else po_sensor, po_collim, param["zeta_maya"],
                               aggreg_rc["Ei"][sensor_id] if "Ei" in aggreg_rc else "NA",  # rc_ei'
                               aggreg_rs["Ei"][sensor_id] if "Ei" in aggreg_rs else "NA",  # rs_ei
                               aggreg_rc["Ei0"][sensor_id] if "Ei0" in aggreg_rc else "NA",  # rc_ei0
                               aggreg_rs["Ei0"][sensor_id] if "Ei0" in aggreg_rs else "NA",  # rs_ei0
                               aggreg_rc["Ei_sup"][sensor_id] if "Ei_sup" in aggreg_rc else "NA",  # rc_ei_sup
                               aggreg_rs["Ei_sup"][sensor_id] if "Ei_sup" in aggreg_rs else "NA",  # rs_ei_sup
                               aggreg_rc["Ei_inf"][sensor_id] if "Ei_inf" in aggreg_rc else "NA",  # rc_ei_inf
                               aggreg_rs["Ei_inf"][sensor_id] if "Ei_inf" in aggreg_rs else "NA",  # rs_ei_inf
                               aggreg_rc["Eabs"][sensor_id] if "Eabs" in aggreg_rc else "NA",  # rc_eabs
                               aggreg_rs["Eabs"][sensor_id] if "Eabs" in aggreg_rs else "NA",  # rs_eabs
                               time() - caribu_time)
                output_caribu.write((("%s;" * len(output_list))[:-1] + "\n") % output_list)

            nlight += 1
            if by_direction is False:
                break

        i.value += 1
    ds.print_progress_bar(i.value, imax, start_time=timing, print_end="|" + param["semis"] + " " + ray)


if __name__ == '__main__':
    debug = False if sys.gettrace() is None else True  # Test if pycharm is in debug mode
    timing = time()
    starttime = strftime("%Y-%m-%d_%HH%M")

    fov = 3  # 3.15
    experimentation = 'ArchiMono'  # 'peuplement'  # 'ArchiDuo'  #"cj"  #
    reconstruction = 'Marching Cubes'  # 'Ball Pivoting'  #
    journalier = True
    recalage = False
    continue_run = False  # permet de reprendre une simulation la ou elle s'est arretee
    path_these = os.path.join(os.path.realpath(__file__).split("These")[0], "These")

    if experimentation == "ArchiMono":
        path_scans = glob(os.path.join(path_these, 'Experimentation', experimentation, 'Scan*3D'))[0]
        list_obj = glob(path_scans + "/**/" + reconstruction + "/**/*.obj", recursive=True)
        recalage = False

    elif experimentation == "ArchiDuo":
```

```
        path_scans = glob(os.path.join(path_these, 'Experimentation', experimentation, 'Scan*3D'))[0]
        obj_dir = "obj apss" if reconstruction == 'Marching Cubes' else "obj"
        list_obj =  glob(path_scans + "/**/" + obj_dir + "/*.obj", recursive=False)

        # Test avec recalage
        if recalage:
            recalage_path = glob(path_these + '/Bordeaux/**/recalage_brute_force_manuel.csv', recursive=True)[0]
            df_recalage = pandas.read_csv(recalage_path, sep=";")
            list_obj = [obj_name for obj_name in list_obj if any(obj_decal in obj_name for obj_decal in
list(df_recalage.scan))]  # conserve uniquement les scans qui ont eu un recalage

    if experimentation == "cj":
        path_scans = glob(os.path.join(path_these, 'Experimentation', "Cinetique journaliere", 'scans'))[0]
        obj_dir = "obj apss" if reconstruction == 'Marching Cubes' else "obj bp"
        list_obj = glob(os.path.join(path_scans, "**", obj_dir, "cor360_2020-11-25*.obj"), recursive=True)
        recalage = False

    elif experimentation == "peuplement":
        path_scans = glob(os.path.join(path_these, 'Experimentation', "Cinetique journaliere", 'scans'))[0]
        # list_obj = glob(os.path.join(path_scans, '**', 'peuplement dense.obj'))
        list_obj = glob(os.path.join(path_scans, '**', 'peuplement precoce.obj'))
        recalage = False

    if len(list_obj) == 0:
        sys.exit("Probleme de chemin de fichier dans " + experimentation)

    # Multiprocessing
    CPUnb = min(len(list_obj), int(multiprocessing.cpu_count() - multiprocessing.cpu_count() / 2))  # nombre de
processeurs utilises
    multiprocessing.freeze_support()  # aucune idee de l'utilite, repris de l-egume
    pool = multiprocessing.Pool(processes=CPUnb)
    it_nb = multiprocessing.Manager().Value('i', 0)  # equilavent de global pour multiprocessing
    print("Simulation sur " + str(CPUnb) + " processeurs")

    if continue_run:
        caribu_name = glob(os.path.join(path_scans, 'output', 'caribu', reconstruction, '*_caribu_*' + reconstruction +
'*.csv'), recursive=True)[-1]
        list_obj = ds.run_check(list_obj, caribu_name, column="obj_name")
        print("\nReprise de la simulation précédente\n")
    else:
        caribu_name = os.path.join(path_scans, 'output', 'caribu', reconstruction, starttime + '_caribu_' + reconstruction +
'.csv')
    if recalage:
        caribu_name.replace(".csv", "_recalage.csv")

    print(os.path.basename(caribu_name)[:-4].replace('_', ' ') + " " + experimentation + "\n" + caribu_name)
    for obj_path in list_obj:
        if os.path.getsize(obj_path) / 1024 < 2500:# * 10**15:
            if experimentation == "ArchiMono":
                distances = list(range(20, 160, 10)) + list(range(160, 600, 40))
                rotx, roty, rotz = 0, 0, 0

            elif experimentation == "ArchiDuo":
                distances = [int(os.path.basename(obj_path).split('_')[3])]
                rotx, roty, rotz = 90 if reconstruction == 'Marching Cubes' else 0, 0, 180  # rotation a 90° pour marching
cubes car rotation provoquee par la reconstruction
```

```
        elif experimentation == "peuplement":
            distances = [0]
            rotx, roty, rotz = 0, 0, 0

        else:
            distances = [int(os.path.basename(obj_path).split('_')[3])]
            rotx, roty, rotz = 0, 0, 180  # rotation a 90° pour marching cubes car rotation provoquee par la reconstruction

        sensor_diam = [2]
        z = 33

        it_max = len(list_obj) * len(distances) * len(sensor_diam)
        for sd in sensor_diam:
            for x in distances:
                if recalage and experimentation == "ArchiDuo":
                    info_name = os.path.basename(obj_path[:-4]).split("_")
                    date_pos = [i for i, x in enumerate(info_name) if "2020" in x][0]
                    ech = "_".join(info_name[date_pos:(date_pos + 4)])
                    y = float(df_recalage[df_recalage.scan == ech].translation_y)
                    rotz = 180 + float(df_recalage[df_recalage.scan == ech].rotation_z)
                    scale = 1
                else:
                    y = 0
                    scale = 1

                if debug:
                    print("Debug")
                    zeta_caribu(obj_path, expe=experimentation, x=x, y=y, z=z, fov=fov, sensor_diam=sd,
                                caribu_name=caribu_name, timing=timing, i=it_nb, imax=it_max,
                                x_rot=rotx, y_rot=roty, z_rot=rotz, scale=scale, journalier=journalier)
                else:
                    # save this script
                    src = os.path.realpath(__file__)
                    dst = caribu_name.split("_caribu_")[0] + '_' + os.path.basename(__file__)
                    if not os.path.exists(os.path.dirname(dst)):
                        os.makedirs(os.path.dirname(dst))
                    if os.path.isfile(dst) is False:
                        copy(src, dst)

                    zeta_caribu_args = (obj_path, experimentation, x, y, z, fov, sd, caribu_name, timing, it_nb, it_max, rotx,
roty, rotz, scale, journalier)
                    if CPUnb == 1:
                        zeta_caribu(obj_path, expe=experimentation, x=x, y=y, z=z, fov=fov, sensor_diam=sd,
caribu_name=caribu_name,
                                    timing=timing, i=it_nb, imax=it_max, x_rot=rotx, y_rot=roty, z_rot=rotz,
                                    scale=scale, journalier=journalier)
                        result = pool.apply_async(zeta_caribu, args=zeta_caribu_args)
                        result.get()  # permet de recuperer les messages d'erreurs du multisim lors d'un bug
                    else:
                        pool.apply_async(zeta_caribu, args=zeta_caribu_args)
        else:
            print(obj_path + " trop volumineux, ignore")
    pool.close()
    pool.join()
    print("\n\nSimulation terminee a", strftime("%HH%M"), "en", int((time() - timing) / 60), "minutes")
```

create_directional_sensor.py

Script contenant différentes fonctions utilisées pour la simulation du ζ avec CARIBU, notamment les fonctions de construction de capteur directionnel.

```
from numpy import *
from time import strftime, time, gmtime
from glob import glob
import os
import sys
import pandas
import datetime as dt
import openalea.plantgl.all as pgl
from alinea.caribu.sky_tools import GenSky, GetLight, Gensun


# Divers
def print_progress_bar(iteration, total, prefix=None, suffix='', decimals=2, length=50, fill='', print_end='',
                       start_time=None):
    """
    Call in a loop to create terminal progress bar
    @params:
        iteration   - Required  : int. current iteration
        total       - Required  : int. total iterations
        prefix      - Optional  : str. prefix string
        suffix      - Optional  : str. suffix string
        decimals    - Optional  : int. positive number of decimals in percent complete
        length      - Optional  : int. character length of bar
        fill        - Optional  : str. bar fill character
        printEnd    - Optional  : str. end character (e.g. "\r", "\r\n")
        start_time  - Optional  : time of the first iteration
    """
    prefix = strftime("%Y-%m-%d %HH%M") if prefix is None else prefix
    percent = ("{0:." + str(decimals) + "f}").format(100 * (iteration / float(total)))
    filled_length = int(length * iteration // total)
    bar = fill * filled_length + '-' * (length - filled_length)

    if start_time is not None:
        progress = iteration / total if iteration > 0 else 1 / total
        run_time = int(time()) - start_time

        remaining_time = run_time / progress - run_time if progress < 1 else run_time
        d = int(remaining_time / (3600 * 24))
        h = int((remaining_time - d * 3600 * 24) / 3600)
        m = int((remaining_time - (d * 3600 * 24 + h * 3600)) / 60)
        s = int(remaining_time - (d * 3600 * 24 + h * 3600 + m * 60))

        label = "- " if progress < 1 else "- Ended in "
        label = label + str(d) + "d " if d > 0 else label
        label = label + str(h) + "h " if h > 0 else label
        label = label + str(m) + "min " if m > 0 else label
        label = label + str(s) + "sec" if s > 0 else label
        label = label + " remaining" if progress < 1 else label
    else:
        label = ''

    sys.stdout.write('\r%s|%s| %s%% %s %s%s' % (prefix, bar, percent, suffix, label, print_end))
```

```
def grep(pattern, x, invert=False):
    if isinstance(pattern, str):
        if invert:
            grep_list = list(filter(lambda y: pattern not in y, x))
        else:
            grep_list = list(filter(lambda y: pattern in y, x))

        grep_list = grep_list[0] if len(grep_list) == 1 else grep_list

    else:
        grep_list = list()
        for p in pattern:
            if invert:
                x = list(filter(lambda y: p not in y, x))
                grep_list = x
                # grep_list.append(list(filter(lambda y: not p in y, x))[0])
            else:
                list_grep = list(filter(lambda y: p in y, x))
                for g in list_grep:
                    grep_list.append(g)

    return grep_list


def run_check(list_run, output_run, column):
    output_df = pandas.read_csv(output_run, delimiter=";")
    # already_done = list(pandas.unique(output_df.date + '_' + output_df.semis))
    already_done = list(pandas.unique(output_df[column]))
    list_not_run = grep(already_done, list_run, invert=True)

    return list_not_run


def data_archimono(obj_path, distance=None, path_these=None):
    """
    Importe les donnees d'archimono
    @params:
        obj_path - Required  : str. path to the obj file
        distance - Optional :
        path_these - Required  : str. path to the experiment folder
    """
    if path_these is None:
        path_these = os.path.join(os.path.expanduser('~'), 'Documents', 'Couturier Arthur', 'These')
    obj = os.path.basename(obj_path)[:-4].replace("sans_pot_", "")

    date_mesure = obj.split("_")[1]
    semis, rep = obj.split("_")[0].split("-")
    csv_path = glob(path_these + "/Experimentation/**/ArchiMono/**/*" + date_mesure + '*.csv', recursive=True)[0]
    csv_manip = pandas.read_csv(csv_path, delimiter=";")
    csv_manip = csv_manip[csv_manip[csv_manip.columns[0]].str.contains(semis + '-' + rep + '_')]

    try:
        mesure = csv_manip[csv_manip[csv_manip.columns[0]].str.contains(str(int(distance / 10)) + 'cm')].iloc[0]
        heure_mesure = mesure.Date.split(" ")[1]
        zeta_manip = mesure.ZETA_655_665_725_735_nm
    except:
```

```
        print("\rDonnees manquantes pour", date_mesure, semis, rep, ", estimation de l'heure de mesure")
        h_prev = csv_manip.iloc[0].Date.split(" ")[1].split(':')  # premiere mesure pour cet echantillon
        h_next = csv_manip.iloc[csv_manip.shape[0] - 1].Date.split(" ")[1].split(
            ':')  # derniere mesure pour cet echantillon
        h_mean = (int(h_prev[0]) * 3600 + int(h_prev[1]) * 60 + int(h_prev[2]) + int(h_next[0]) * 3600 + int(
            h_next[1]) * 60 + int(h_next[2])) / 2
        heure_mesure = strftime('%H:%M:%S', gmtime(h_mean))  # heure moyenne entre la 1ere et la derniere mesure
        zeta_manip = "NA"

    return {"date": date_mesure, "heure": heure_mesure, "semis": semis, "rep": rep, "zeta_maya": zeta_manip}


def data_archiduo(obj_path, path_these=None):
    """
    Importe les donnees d'archiduo
    @params:
        obj_path - Required  : str. path to the obj file
        path_these - Required  : str. path to the experiment folder
    """
    if path_these is None:
        path_these = os.path.join(os.path.expanduser('~'), 'Documents', 'Couturier Arthur', 'These')
    obj = os.path.basename(obj_path)[:-4].replace("sans_pot_", "")
    date_mesure, rep, x, orientation = obj.split("_")[1:5]
    semis = "_".join((rep, x, orientation))
    maya_path = glob(path_these + "/**/ArchiDuo/**/*" + date_mesure + '.csv', recursive=True)[0]
    df_manip = pandas.read_csv(maya_path, delimiter=";")
    df_manip = df_manip[df_manip[df_manip.columns[0]].str.contains(semis)]
    heure_mesure = df_manip.Date.iloc[0].split(" ")[1]
    zeta_manip = float(df_manip['ZETA_655_665_725_735_nm'])

    return {"date": date_mesure, "heure": heure_mesure, "semis": semis, "rep": rep, "zeta_maya": zeta_manip}


# Fonction plantGL
def quad_cyl(height, radius, slices=100):
    """
    Makes a textured pgl.Cylinder with quads as primitives.
    @params:
        height - Required  : float. cylinder height (scene unit)
        radius - Required  : float. cylinder radius (scene unit)
        slice  - Optional  : int. slice number of the cylinder
    """
    points = []
    ind = []
    tex_coords = []
    for slice_nb in range(slices):
        x = radius * cos(2 * slice_nb * pi / slices)
        y = radius * sin(2 * slice_nb * pi / slices)
        # points += [(x, y, 0), (x, y, height)]
        # tex_coords += [(slice_nb * 1. / slices, 0), (slice_nb * 1. / slices, 1)]
        points += [(x, y, height), (x, y, 0)]
        tex_coords += [(slice_nb * 1. / slices, 1), (slice_nb * 1. / slices, 0)]
        if slice_nb is not (slices - 1):
            ind += [(2 * slice_nb, 2 * slice_nb + 1, 2 * slice_nb + 3, 2 * slice_nb + 2)]
        else:
            ind += [(2 * slice_nb, 2 * slice_nb + 1, 1, 0)]
    tex_coords += [(1, 0), (0, 0)]
    cyl = pgl.QuadSet(points, ind)
```

```
    cyl.texCoordList = tex_coords
    cyl.texCoordIndexList = ind
    cyl = pgl.AxisRotated((0, 1, 0), pi / 2., cyl)
    return cyl


def pgl_square(height, width=None):
    if width is None:
        width = height

    height = height / 2
    width = width / 2

    points = [(-width, -height, 0),
              (width, -height, 0),
              (width, height, 0),
              (-width, height, 0)]
    carre = pgl.QuadSet(points, [(0, 1, 2, 3)])
    return carre


def triangulate_shape(obj, id):
    tess = pgl.Tesselator()
    obj.geometry.apply(tess)
    # sensor_shape.geometry.apply(tess)
    tri = tess.triangulation
    tri_shp = []
    for index in range(tri.indexListSize()):
        tple = [(tri.pointAt(index, 0).x, tri.pointAt(index, 0).y, tri.pointAt(index, 0).z),
                (tri.pointAt(index, 1).x, tri.pointAt(index, 1).y, tri.pointAt(index, 1).z),
                (tri.pointAt(index, 2).x, tri.pointAt(index, 2).y, tri.pointAt(index, 2).z)]
        tri_shp.append(tple)

    return {id: tri_shp}


def shapepgl(obj, id, scale=None, trans=None, x_rot=0, y_rot=0, z_rot=0, rgb=None):
    """
    Simplified version of plantgl shape functions
    @params:
        obj    - Required  : int. object to tranform into plantgl shape
        id_obj - Required  : int. shape id
        scale  - Optional  : float or tuple. increase the object size, can be x or (x, y, z)
        trans  - Optional  : tuple. translation coordonate (x, y, z)
        x_rot  - Optional  : degree. rotate the object around the x axis
        y_rot  - Optional  : degree. rotate the object around the y axis
        z_rot  - Optional  : degree. rotate the object around the z axis
        rgb    - Optional  : tuple. shape color (red, green, blue, alpha), int between 0 and 255 for rgb, float between 0
(solid) and 1 (transparent) for alpha
    """
    if isinstance(scale, int) or isinstance(scale, float):
        obj = pgl.Scaled(scale, scale, scale, obj)
    elif scale is not None:
        obj = pgl.Scaled(scale[0], scale[1], scale[2], obj)

    obj = pgl.AxisRotated((1 if x_rot == 0 else float(sign(x_rot)), 0, 0), radians(abs(x_rot)), obj)
    obj = pgl.AxisRotated((0, 1 if y_rot == 0 else float(sign(y_rot)), 0), radians(abs(y_rot)), obj)
    obj = pgl.AxisRotated((0, 0, 1 if z_rot == 0 else float(sign(z_rot))), radians(abs(z_rot)), obj)
```

```
    if trans is not None:
        obj = pgl.Translated(trans[0], trans[1], trans[2], obj)

    if rgb is None:
        rgb = (0, 0, 0, 0)
    shape_color = pgl.Material(pgl.Color3(rgb[0], rgb[1], rgb[2]))
    shape_obj = pgl.Shape(obj, shape_color, id=id)
    if len(rgb) == 4:
        shape_obj.appearance.transparency = rgb[3]

    return shape_obj


# Caribu
def caribu_light(day, hour, ray="vertical", energy=1.0, azimuths_nb=4, zeniths_nb=5, diffuse_model='soc',
                 rectify_azimuth=None, rectify_elevation=None):
    """
    Create the light environment for Caribu for Lusignan experimentation site.
    @params:
        day  - Required  : str. Date of the measure ("YYYY-MM-DD")
        hour - Required  : str. Hour of the measure ("hh:mm:ss")
        ray              - Optional : str. slice number of the cylinder. options are "direct", "diffus" or "vertical"
        energy           - Optional : float. energy of the light environment
        azimuths_nb      - Optional : int. azimuthal sectors for diffuse ray
        zeniths_nb       - Optional : int. zenital sectors for diffuse ray
        diffuse_model    - Optional : str. sky distribution type ('soc' or 'uoc'). soc = ciel clair. uoc = ciel couvert
        rectify_azimuth  - Optional : float. force the azimuth value for direct ray (degree)
        rectify_elevation - Optional : float. force the elevation value for direct ray (degree)
    """
    if ray == "direct":
        coord_lusignan = {"lat": 46.4036827, "lon": 0.0787160}
        date = list(map(int, day.split("-")))
        heure = int(hour.split(":")[0])
        doy = (dt.date(date[0], date[1], date[2]) - dt.date(date[0], 1, 1)).days
        sun = Gensun.Gensun()(energy, doy, heure, coord_lusignan["lat"])
        sun.azim = radians(rectify_azimuth) if rectify_azimuth is not None else sun.azim
        sun.elev = -radians(rectify_elevation) if rectify_elevation is not None else sun.elev
        light = list(map(float, sun.toLight().split(' ')))
        light = [tuple((float(light[0]), tuple((round(light[1], 10), round(light[2], 10), round(light[3], 10)))))]

        azim = degrees(sun.azim)
        elev = degrees(sun.elev)

    elif ray == "diffus":
        sky_string = GetLight.GetLight(GenSky.GenSky()(energy, diffuse_model, azimuths_nb, zeniths_nb))
        # Convert string to list in order to be compatible with CaribuScene input format
        light = []
        for string in sky_string.split('\n'):
            if len(string) != 0:
                string_split = string.split(' ')
                sky_tup = tuple((float(string_split[0]),
                                 tuple((float(string_split[1]), float(string_split[2]), float(string_split[3])))))
                light.append(sky_tup)
        azim, elev = "NA", "NA"

    else:  # Default light in caribu is a vertical light
        light = None
```

```
        azim = 0
        elev = 0
        print("vertical sun")

    return light, azim, elev


# Capteur directionnel
def sensor_radius(height, fov):
    """
    Calculate the radius of a cone from the height and the top angle.
    @params:
        height - Required  : float. cone height (degree)
        fov    - Required  : float. top angle (degree)
    """
    slant_height = height / cos(radians(fov / 2.))
    radius = cos(radians(90 - fov / 2.)) * slant_height
    return radius


def sensor_cylinder_height(diameter, fov):
    """
    @params:
        diameter  - Required  : float.
        fov       - Required  : float. top angle (degree)
    """
    cylinder_height = (diameter / 2) / tan(radians(fov / 2))

    return cylinder_height


def create_sensor(scene, opt, position, id=None,
                  fov=3., diameter=1.5,
                  po_collim=(0, 0, 10 ** -11, 0), po_sensor=(10 ** -11, 10 ** -11),
                  virtual_sensor=True):
    """
    Add a directionnal sensor in the scene.
    @params:
        scene             - required : openalea scene
        opt               - required : dict. optical properties of the scene objects
        position          - required : tuple. (x, y, z) position of the sensor

        id                - optional : integer. Id of the last object in the scene. Default is scene length
        fov               - Optional : float. Field of view the the sensor. Default is 3 degrees.
        diameter          - Optional : float.
        trunc_cone_height - Optional : float.
        po_collim         - Optional : tuple. Collimator optical properties. Default is set to have no impact on the measure.
        po_sensor         - Optional : tuple. Sensor optical properties (reflectance, transmittance). Default is set to have no
impact on the measure.
            2-tuple = (reflectance, transmittance)
            4-tuple = (reflectance upper side, transmittance upper side, reflectance lower side, transmittance lower side)
        virtual_sensor    - Optional : logical. If True the sensor cell is virtual in the scene.
    """
    # tess = pgl.Tesselator()
    id = len(scene) if id is None else id
    side = int(sign(position[0]))
    side = 1 if side == 0 else side
```

```
    # Collimateur
    cylinder_height = sensor_cylinder_height(diameter, fov)
    x1 = position[0]
    x2 = x1 + side * cylinder_height
    # collim_pos = (x1, position[1], position[2])  # V1
    position = (x2, position[1], position[2])  # V2

    add_collim = True
    if add_collim:
        collim_cyl = quad_cyl(cylinder_height, diameter)
        scene += shapepgl(collim_cyl, trans=position, y_rot=180 if side == 1 else 0, rgb=(0, 0, 255), id=id)
        opt['rc'][id] = po_collim
        opt['rs'][id] = po_collim
        id += 1

        collim_cone = pgl.Cone(diameter, 1, slices=100, solid=False)
        scene += shapepgl(collim_cone, trans=position, y_rot=side * 90, rgb=(0, 0, 255), id=id)  # rotation 180 degrees
        opt['rc'][id] = po_collim
        opt['rs'][id] = po_collim
        id += 1

        # test d'obturation
        obturation = False
        if obturation:
            print("Warning: obturateur ON")
            bouchon = pgl.Disc(diameter * 1.2, slices=100)  # reduction legere de la taille du capteur pour s'assurer qu'il ne
ressorte pas
            scene += shapepgl(bouchon, y_rot=side * 90, trans=position, rgb=(255, 0, 255, 0), id=id)  # derriere le capteur
            opt['rc'][id] = (10 ** -25, 0)
            opt['rs'][id] = (10 ** -25, 0)
            id += 1
    else:
        print("\n\n\nWarning: pas de collimateur\n\n\n")

    sensor = pgl.Disc(diameter * 0.995, slices=100)  # reduction legere de la taille du capteur pour s'assurer qu'il ne
ressorte pas
    sensor_shape = shapepgl(sensor, y_rot=-side * 90, trans=position, rgb=(255, 0, 0, 0), id=id)

    if virtual_sensor:
        # Virtual sensor
        sensor_dico = triangulate_shape(sensor_shape, id=id)
        sensor_id = id
    else:
        # Scene sensor
        scene += sensor_shape
        opt['rc'][id] = po_sensor
        opt['rs'][id] = po_sensor
        sensor_dico = None
        sensor_id = id

    return scene, opt, sensor_dico, sensor_id, position


# Capteur directionnel surfacique
def zeta_area(scene, camera_pos, fov, perspective=True, frame_size=1000):
    """
    The plant material color of the scene must be coded as green (rgb = (0, >50, 0)) in order to run with this function.
    :param scene: plantgl scene
```

```
    :param camera_pos: (x,y,z) for the camera
    :param fov: float
    :param perspective: logical # Depth can only be evaluated with a perspective view
    :param frame_size: int
    :return: raw green pixel number and percentage of green pixel in the frame
    """
    pgl.Viewer.frameGL.setSize(frame_size, frame_size)
    pgl.Viewer.camera.setViewAngle(fov / 2.)  # Plantgl viewer use radius
    pgl.Viewer.display(scene)  # Must be before camera position
    pgl.Viewer.grids.set(0, 0, 0, 3)  # Quadrillage XY
    pgl.Viewer.frameGL.setBgColor(255, 255, 255)  # white background
    pgl.Viewer.camera.setPerspective() if perspective else pgl.Viewer.camera.setOrthographic()
    pgl.Viewer.camera.set(camera_pos, 0, 0)
    pgl.Viewer.camera.lookAt((0, 0, camera_pos[2]))

    # Plant area
    if perspective:
        buff = pgl.Viewer.frameGL.grabZBuffer()
        if len(buff) == frame_size:
            px_plt = sum((1 for line in buff for pix in line if
                          pix is not None and pix[1].green > 50 and pix[1].red < 200 and pix[1].blue < 200))
            percent_plant = px_plt / (pi * (frame_size / 2.) ** 2)
        else:
            pgl.Viewer.frameGL.setSize(frame_size, frame_size)
            pgl.Viewer.camera.setViewAngle(fov / 2.)  # Plantgl viewer use radius
            pgl.Viewer.display(scene)  # Must be before camera position
            pgl.Viewer.grids.set(0, 0, 0, 3)  # Quadrillage XY
            pgl.Viewer.frameGL.setBgColor(255, 255, 255)  # white background
            pgl.Viewer.camera.setPerspective() if perspective else pgl.Viewer.camera.setOrthographic()
            pgl.Viewer.camera.set(camera_pos, 0, 0)
            pgl.Viewer.camera.lookAt((0, 0, camera_pos[2]))
            print("Problem with buff size", len(pgl.Viewer.frameGL.grabZBuffer()))

            px_plt = sum((1 for line in buff for pix in line if
                          pix is not None and pix[1].green > 50 and pix[1].red < 200 and pix[1].blue < 200))
            percent_plant = px_plt / (pi * (frame_size / 2.) ** 2)
    else:
        px_plt = pgl.Viewer.frameGL.getProjectionSize()[1]
        percent_plant = px_plt / frame_size ** 2

    pgl.Viewer.hide()
    return {'px_plt': px_plt, 'percent_plant': percent_plant, 'zeta_px': -0.84 * percent_plant + 1.07}
```

malia_multisim.py

Script à partir duquel les simulations de MALIA sont effectuées.

```
import os
import pandas
import sys
from glob import glob
from shutil import copy
from time import strftime, time
import malia_tools
import multiprocessing


def grep(pattern, x):
    if isinstance(pattern, str):
        grep_list = list(filter(lambda y: pattern in y, x))[0]

    else:
        grep_list = list()
        for p in pattern:
            grep_list.append(list(filter(lambda y: p in y, x))[0])

    return grep_list


def prepare_malia_multisim(expe, reconstruction, archive_name=None, recalage=False,
                           fov_camera=15, fov_final=3.15, lw="400:800:10", journalier=False, starttime=None):
    debug = False if sys.gettrace() is None else True  # Test if pycharm is in debug mode
    timing = time()
    if starttime is None:
        starttime = strftime("%Y-%m-%d_%HH%M")

    if archive_name is None:
        archive_name = '_'.join(('malia_archive', starttime, expe, reconstruction, 'FOV' + str(fov_final)))
        archive_name = archive_name + "_recalage" if recalage else archive_name
        archive_name = archive_name + "_zeta_journalier" if journalier else archive_name

    # Methode de reconstruction
    if reconstruction == "mc":
        recons = 'Marching cubes'
    elif reconstruction == "bp":
        recons = 'Ball Pivoting'
    else:
        sys.exit('pb choix reconstruction')

    # Experimentation
    if expe.lower() == "archimono":
        if reconstruction == "bp":
            list_obj_path = os.path.join(os.path.dirname(malia_path), '**', expe, recons, 'obj', '*.obj')
        else:
            list_obj_path = os. path.join(os.path.dirname(malia_path), '**', expe, recons, 'obj apss', '*.obj')
    elif expe.lower() == "archiduo":
        list_obj_path = os.path.join(os.path.dirname(malia_path), '**', expe, recons, 'obj', '*.obj')
    elif expe.lower() == "peuplement":
        list_obj_path = os.path.join(os.path.dirname(malia_path), '**', expe, recons, 'obj', '*.obj')
    else:
```

```
        sys.exit('pb choix expe')

    list_obj = glob(list_obj_path, recursive=True) if debug is False else glob(list_obj_path, recursive=True)[:1]
    if len(list_obj) == 0:
        sys.exit("Probleme d'import de fichier")
    else:
        work_dir = os.path.dirname(os.path.dirname(list_obj[0]))

    # Test avec recalage
    if recalage:
        recalage_path = glob(path_these + '/Bordeaux/**/recalage_brute_force.csv', recursive=True)[0]
        df_recalage = pandas.read_csv(recalage_path, sep=';')
        list_obj = [obj_name for obj_name in list_obj if any(obj_decal in obj_name for obj_decal in
list(df_recalage.scan))]  # filtre les scans

    print(starttime, expe, recons, "\n", work_dir)
    malia_args = []
    i = 1
    for obj in list_obj:
        obj = os.path.abspath(obj)
        obj_root = os.path.dirname(os.path.dirname(obj))
        obj_name = os.path.basename(obj[:-4])

        default_malia_args = malia_tools.get_default_args(malia_tools.run_malia_sensor)

        if expe == "ArchiMono":
            x_cam_list, y_cam, z_cam = list(range(2, 16)) + list(range(16, 60, 4)), 0, 2.5
            info_manip = malia_tools.data_archimono(obj, distance=None, path_these=path_these)
            date_heure = info_manip['date_heure']
            obj_scale, rot = 0.1, [-90, (1, 0, 0)]
            extra_obj = expe

        if expe == "ArchiDuo":
            info_manip = malia_tools.info_archiduo(obj, path_these)
            date_heure = info_manip['date_heure']
            dist = info_manip['dist']
            extra_obj = expe
            obj_scale, rot = 0.1, None if recons == 'Marching cubes' else [-90, (1, 0, 0)]
            x_cam_list, y_cam, z_cam = [int(dist) * obj_scale], 0, 33 * obj_scale

        if expe == "Peuplement":
            info_manip = {"date_heure": "26-10-2020" + " " + "12:00:00", "date": "26-10-2020", "heure": "12:00:00",
                          "semis": "NA", "bloc": "NA", "dist": 100, "orientation": "NA", "zeta": "NA"}
            date_heure = info_manip['date_heure']
            obj_scale, rot = 1, [-90, (1, 0, 0)]
            extra_obj = None
            x_cam_list, y_cam, z_cam = [100], 0, 33

        if recalage:
            param_decal = df_recalage[df_recalage.scan == obj_name.replace('cor360_', '')]
            rot = [float(param_decal.rotation_z), [0., 0., 1.]]
            obj_scale = float(param_decal.scale)
            y_cam = y_cam + float(param_decal.translation_y)

        for x_cam in x_cam_list:
            if recalage:
                file_name = "_".join((obj_name, "recalage", "FOV" + str(fov_camera), str(int(x_cam)) + "cm"))
            else:
```

```
            file_name = "_".join((obj_name, "FOV" + str(fov_camera), str(int(x_cam)) + "cm"))

        if journalier is False:
            malia_args.append((obj, malia_path,
                               file_name, path_these, info_manip,
                               date_heure, obj_scale, rot,
                               x_cam, y_cam, z_cam,
                               lw, fov_camera, fov_final,
                               recalage, reconstruction, expe,
                               archive_name, starttime,
                               extra_obj))
            nb_tot = len(list_obj) * len(x_cam_list)
            malia_tools.print_progress_bar(i, nb_tot, start_time=timing, prefix="Preparation malia_args")
            i += 1
        else:
            liste_heures = [str(h) + ":00:00" for h in range(7, 18)]
            for h in liste_heures:
                malia_args.append((obj, malia_path,
                                   file_name, path_these, info_manip,
                                   info_manip["date"] + " " + h, obj_scale, rot,
                                   x_cam, y_cam, z_cam,
                                   lw, fov_camera, fov_final,
                                   recalage, reconstruction, expe,
                                   archive_name, starttime,
                                   extra_obj))
                nb_tot = len(list_obj) * len(x_cam_list) * len(liste_heures)
                malia_tools.print_progress_bar(i, nb_tot, start_time=timing, prefix="Preparation malia_args")
                i += 1

    # save this script
    if debug is False:
        src = os.path.realpath(__file__)
        dst = os.path.join(os.path.dirname(os.path.dirname(list_obj[0])), starttime + '_' + os.path.basename(__file__))
        if os.path.isfile(dst) is False:
            copy(src, dst)
    else:
        print("\nDebug ON")
        malia_tools.run_malia_sensor(obj, malia_path,
                                     output_name=file_name, path_these=path_these, info_manip=info_manip,
                                     date_heure=date_heure, obj_scale=obj_scale, rot=rot,
                                     x_cam=x_cam, y_cam=y_cam, z_cam=z_cam,
                                     lw=lw, fov_camera=fov_camera, fov_final=fov_final,
                                     recalage=recalage, reconstruction=reconstruction, expe=expe,
                                     archive_name=archive_name, starttime=starttime,
                                     extra_obj=extra_obj)

    return malia_args


if __name__ == '__main__':
    debug = False if sys.gettrace() is None else True  # Test if pycharm is in debug mode
    starttime = strftime("%Y-%m-%d_%HH%M")

    path_these = os.path.join(os.path.realpath(__file__).split("These")[0], "These")  # Repertoire principal de travail
    malia_path = glob(path_these + '/**/bin/malia.exe', recursive=True)[0]  # Chemin vers le logiciel malia

    expe = "ArchiDuo"  # "ArchiMono"  # "Peuplement"
    reconstruction = "bp"
```

```
lw = "400:800:10" if debug else "660:730:70"  # "660:800:10"

malia_args = prepare_malia_multisim(expe, reconstruction, lw=lw, journalier=False)

i = multiprocessing.Manager().Value('i', 0)  # equilavent de global pour multiprocessing
imax = len(malia_args)

timing = time()
CPUnb = 30
pool = multiprocessing.Pool(processes=CPUnb)
for args in malia_args:
    args = args + (i, imax, timing)
    result = pool.apply_async(malia_tools.run_malia_sensor, args)
    result.get()
pool.close()
pool.join()

print("\n")
malia_tools.print_progress_bar(1, 1, fill="", border="", start_time=timing, prefix="\n", suffix="\r")
```

malia_tools.py

Ce fichier répertorie différentes fonctions appelées dans les scripts *asc2obj.py* et *malia_multisim.py*

```
import os
import sys
import pandas
import pymeshlab
import numpy as np
import spectral
import xml.etree.ElementTree as ETree
from glob import glob
from shutil import copy
from subprocess import call, DEVNULL
from time import strftime, time, gmtime, sleep
from py7zr import SevenZipFile
from pvlib.solarposition import get_solarposition


# Divers
def get_default_args(func):
    import inspect
    signature = inspect.signature(func)
    return {
        k: v.default
        for k, v in signature.parameters.items()
        if v.default is not inspect.Parameter.empty
    }

def print_progress_bar(iteration, total, prefix=None, suffix='', decimals=2, length=50, fill='', border='|', print_end='',
start_time=None):
    """
    Call in a loop to create terminal progress bar
    @params:
        iteration   - Required  : int. current iteration
        total       - Required  : int. total iterations
        prefix      - Optional  : str. prefix string
        suffix      - Optional  : str. suffix string
        decimals    - Optional  : int. positive number of decimals in percent complete
        length      - Optional  : int. character length of bar
        fill        - Optional  : str. bar fill character
        printEnd    - Optional  : str. end character (e.g. "\r", "\r\n")
        start_time  - Optional  : time of the first iteration
    """
    prefix = strftime("%Y-%m-%d %HH%M") if prefix is None else prefix
    percent = ("{0:." + str(decimals) + "f}").format(100 * (iteration / float(total)))
    filled_length = int(length * iteration // total)
    bar = border + fill * filled_length + '-' * (length - filled_length) + border

    if start_time is not None:
        progress = iteration / total if iteration > 0 else 1 / total
        run_time = int(time()) - start_time

        remaining_time = run_time / progress - run_time if progress < 1 else run_time
        d = int(remaining_time / (3600 * 24))
        h = int((remaining_time - d * 3600 * 24) / 3600)
        m = int((remaining_time - (d * 3600 * 24 + h * 3600)) / 60)
```

```
        s = int(remaining_time - (d * 3600 * 24 + h * 3600 + m * 60))

        label = "- " if progress < 1 else " ".join(("Ended the", strftime('%Y-%m-%d'), "at", strftime("%H:%M:%OS"), "in
"))
        label = label + str(d) + "d " if d > 0 else label
        label = label + str(h) + "h " if h > 0 else label
        label = label + str(m) + "min " if m > 0 else label
        label = label + str(s) + "sec" if s > 0 else label
        label = label + " remaining" if progress < 1 else label
    else:
        label = ''

    sys.stdout.write('\r%s%s %s%% %s %s%s' % (prefix, bar, percent, suffix, label, print_end))


def grep(pattern, x, invert=False):
    """
    Imitation de la fonction grep() de R.
    :param pattern:
    :param x:
    :param invert:
    :return:
    """
    if isinstance(pattern, str):
        if invert:
            grep_list = list(filter(lambda y: pattern not in y, x))
        else:
            grep_list = list(filter(lambda y: pattern in y, x))

        grep_list = grep_list[0] if len(grep_list) == 1 else grep_list

    else:
        grep_list = list()
        for p in pattern:
            if invert:
                x = list(filter(lambda y: p not in y, x))
                grep_list = x
                # grep_list.append(list(filter(lambda y: not p in y, x))[0])
            else:
                list_grep = list(filter(lambda y: p in y, x))
                for g in list_grep:
                    grep_list.append(g)

    return grep_list


def run_check(to_run, output_run, column):
    """
    Verifie les simulations qui ont deja ete effectuees par rapport a la liste des conditions de simulation.
    :param to_run:
    :param output_run:
    :param column:
    :return:
    """
    if os.path.splitext(output_run)[1] == ".7z":
        list_archive_files = SevenZipFile(output_run, mode='r').files.files_list
        list_archive_files = [list_archive_files[x]['filename'] for x in range(len(list_archive_files))]
```

```
        already_done = grep(column, list_archive_files)
        length_name = len(os.path.basename(to_run[0]).split("_"))
        already_done = ['_'.join(os.path.splitext(x)[0].split("_")[0:length_name]) for x in already_done]

        dir_run = os.path.dirname(to_run[0])
        to_run = [os.path.basename(to_run[x]) for x in range(len(to_run))]
        list_not_run = grep(already_done, to_run, invert=True)
        list_not_run = [os.path.join(dir_run, x) for x in list_not_run]
    else:
        output_df = pandas.read_csv(output_run, delimiter=";")
        already_done = list(pandas.unique(output_df[column]))
        already_done = [x.split("_id")[0] for x in already_done]
        list_not_run = grep(already_done, to_run, invert=True)

    return list_not_run


def info_archiduo(obj_path, path_these):
    """
    Recupere les infos de mesures pour un scan de plante a partir des infos contenues dans le nom de l'obj.
    :param obj_path: chemin vers l'obj
    :param path_these: chemin vers le dossier contenant les mesures
    :return:
    """
    obj = os.path.basename(obj_path)[:-4].replace("sans_pot_", "")
    date_pos = [i for i, x in enumerate(obj.split("_")) if "2020" in x][0]  # s'assure de la position de la date dans la
nomenclature du nom
    date_mesure, rep, x, orientation = obj.split("_")[date_pos: date_pos + 4]
    semis = "_".join((rep, x, orientation))

    csv_path = glob(path_these + "/**/ArchiDuo/**/*" + date_mesure + '.csv', recursive=True)[0]
    df_manip = pandas.read_csv(csv_path, delimiter=";")
    df_semis = df_manip[df_manip[df_manip.columns[0]].str.contains(semis)]  # recupere la ligne de mesure
correspondant au semis

    if len(df_semis) == 0:
        heure_mesure = "12:00:00"
        zeta = "NA"
    else:
        heure_mesure = df_semis.Date.iloc[0].split(" ")[1]
        zeta = df_semis.ZETA_655_665_725_735_nm.iloc[0]

    return {"date_heure": date_mesure + " " + heure_mesure, "date": date_mesure, "heure": heure_mesure,
            "semis": semis, "bloc": rep, "dist": x, "orientation": orientation,
            "zeta": zeta}


def data_archimono(obj_path, distance=None, path_these=None):
    """
    Importe les donnees d'archimono
    """
    if path_these is None:
        path_these = os.path.join(os.path.expanduser('~'), 'Documents', 'Couturier Arthur', 'These')
    obj = os.path.basename(obj_path)[:-4].replace("sans_pot_", "")

    date_mesure = obj.split("_")[1]
```

```
    semis, rep = obj.split("_")[0].split("-")
    csv_path = glob(path_these + "/Experimentation/**/ArchiMono/**/*" + date_mesure + '*.csv', recursive=True)[0]
    csv_manip = pandas.read_csv(csv_path, delimiter=";")
    csv_manip = csv_manip[csv_manip[csv_manip.columns[0]].str.contains(semis + '-' + rep + '_')]

    try:
        mesure = csv_manip[csv_manip[csv_manip.columns[0]].str.contains(str(int(distance / 10)) + 'cm')].iloc[0]
        heure_mesure = mesure.Date.split(" ")[1]
        zeta_manip = mesure.ZETA_655_665_725_735_nm
    except TypeError:
        h_prev = csv_manip.iloc[0].Date.split(" ")[1].split(':')  # premiere mesure pour cet echantillon
        h_next = csv_manip.iloc[csv_manip.shape[0] - 1].Date.split(" ")[1].split(':')  # derniere mesure pour cet echantillon
        h_mean = (int(h_prev[0]) * 3600 + int(h_prev[1]) * 60 + int(h_prev[2]) + int(h_next[0]) * 3600 + int(h_next[1]) *
60 + int(h_next[2])) / 2
        heure_mesure = strftime('%H:%M:%S', gmtime(h_mean))  # heure moyenne entre la 1ere et la derniere mesure
        zeta_manip = "NA"

    return {"date_heure": date_mesure + " " + heure_mesure, "date": date_mesure, "heure": heure_mesure,
            "semis": semis, "rep": rep, "zeta": zeta_manip}


def archive(file, zip_name):
    """
    :param file: chemin du fichier a archiver
    :param zip_name: nom de l'archive dans lequel mettre le fichier
    """
    if zip_name[-2:] != "7z":
        zip_name = zip_name + ".7z"  # SevenZipFile fonction avec les extensions 7z
    zip_path = os.path.join(os.path.dirname(file), zip_name)  # place l'archive dans le meme dossier que le fichier
    with SevenZipFile(zip_path, 'a' if os.path.isfile(zip_path) else 'w') as archive:
        archive.write(file, arcname=os.path.basename(file))  # creer l'archive ou enregistre le fichier dans une archive
existante
    try:
        os.remove(file)  # supprime le fichier
    except PermissionError:
        sleep(30)
        os.remove(file)


# Reconstruction a partir des nuages de point
def asc2obj_apss(asc_path, out_path=None, rotation=None):
    """
    Effectue une reconstruction en utilisant la methode Marching Cubes de meshlab
    :param asc_path:
    :param out_path:
    :param rotation:
    :return:
    """
    asc_path = os.path.abspath(asc_path)
    obj_name = os.path.basename(asc_path)[:-3] + "obj"
    if out_path is None:
        dir_path = os.path.dirname(os.path.dirname(asc_path))
        out_path = dir_path + "/obj apss/" + obj_name
    else:
        dir_path = out_path
        out_path = dir_path + "/" + obj_name
    if not os.path.exists(os.path.dirname(out_path)):
```

```
        os.makedirs(os.path.dirname(out_path))

    # Reconstruction
    ms = pymeshlab.MeshSet()
    ms.load_new_mesh(asc_path)
    ms.apply_filter("compute_normals_for_point_sets", k=16)
    ms.apply_filter("marching_cubes_apss", filterscale=3, maxprojectioniters=1, resolution=500)
    ms.save_current_mesh(out_path, save_vertex_normal=False)

    # Supression des faces eloignees du nuage de point
    print("rpy2 not working")
    # robjects.r.source("function_correction_apss.R")
    # remove_vertice = robjects.globalenv['remove_vertice']  # supprime les faces trop eloignees du nuage de points
    # remove_vertice(asc_path, out_path, out_path, 0.5)

    # Supression des points sans face
    ms_obj = pymeshlab.MeshSet()
    ms_obj.load_new_mesh(out_path)
    ms_obj.apply_filter("select_faces_with_edges_longer_than", threshold=0)  # selectionne toute les faces
    ms_obj.apply_filter("select_vertices_from_faces")  # selectionne les vertices reliees aux faces
    ms_obj.apply_filter("invert_selection", invfaces=True, invverts=True)  # selectionne les vertices sans face
    ms_obj.apply_filter("delete_selected_vertices")

    # Suppression des petits elements isoles (attention: peut enlever des feuilles)
    ms_obj.apply_filter("select_small_disconnected_component")
    ms_obj.apply_filter("delete_selected_faces_and_vertices")

    # Lissage des bordures des feuilles
    ms_obj.apply_filter("select_border")
    ms_obj.apply_filter("laplacian_smooth_surface_preserving", selection=False, angledeg=20, iterations=100)
    if rotation is not None:
        ms_obj.apply_filter("transform_rotate", rotaxis='X axis', angle=-90)

    # Export
    ms_obj.save_current_mesh(out_path, save_vertex_normal=False)


def asc2obj_ball_pivoting(asc_path):
    """
    Effectue une reconstruction en utilisant la methode Ball Pivoting de meshlab
    :param asc_path:
    :return:
    """
    out_path = "./obj bp/" + os.path.basename(asc_path)[:-3] + "obj"
    ms = pymeshlab.MeshSet()
    ms.load_new_mesh(asc_path)
    ms.apply_filter("compute_normals_for_point_sets", k=16)
    ms.apply_filter("surface_reconstruction_ball_pivoting", clustering=10, creasethr=150)
    ms.save_current_mesh(out_path)


# Creation d'une scene malia
def prettify(elem, encoding="utf-8"):
    """Return a pretty-printed XML string for the Element."""
    from xml.dom.minidom import parseString
    rough_string = ETree.tostring(elem, encoding)
    reparsed = parseString(rough_string)
```

```
    pretty_elem = reparsed.toprettyxml(indent="  ")
    return pretty_elem


def sph2cart(azimuth, elevation, r):
    x = r * np.cos(np.radians(elevation)) * np.cos(np.radians(azimuth))
    y = r * np.cos(np.radians(elevation)) * np.sin(np.radians(azimuth))
    z = r * np.sin(np.radians(elevation))
    return [x, y, z]


def cart2sph(x, y, z):
    azimuth = np.arctan2(y, x)
    elevation = np.arctan2(z, np.sqrt(x**2 + y**2))
    r = np.sqrt(x**2 + y**2 + z**2)
    return azimuth, elevation, r


def msf_background(element, spectrum="0:0.05,1000:0.05", rgb=None):
    bg = ETree.SubElement(element, "background")
    bg.set("type", "uniform")

    spec = ETree.SubElement(bg, "spectrum")
    spec.set("value", spectrum)

    if rgb is not None:
        rgb_color = ETree.SubElement(bg, "rgb_color")
        rgb_color.set("r", str(rgb[0]))
        rgb_color.set("g", str(rgb[1]))
        rgb_color.set("b", str(rgb[2]))

    return element


def msf_light(element, date=None, light_direct=True, radius=0.1, radiance=1000.0, spectrum="0:1.0,1000:1.0",
rgb=None, sun_pos=None, transform=False):
    """
    :param element: element scene
    :param date: date
    :param light_direct: True = direct, False = diffuse
    :param radius: radius of the sphere light
    :param radiance:
    :param spectrum: 'min:value,max:value'
    :param rgb:
    :param sun_pos:
    :param transform:
    :return: scene with a light
    """

    if rgb is None:
        rgb = [1.0, 1.0, 1.0]
    if sun_pos is None:
        if date is None:
            sys.exit("Veuillez preciser une date ou des coordonnees pour la position du soleil")
        else:
            sun = get_solarposition(date, 46.4036827, 0.0787160)  # lusignan coordinate
            elev = float(sun["elevation"])
```

```
        azim = float(sun["azimuth"])
        sun_pos = sph2cart(azim, elev, 100)
        sun_pos = [sun_pos[0], sun_pos[2], sun_pos[1]]  # inverse y et z par rapport a la position des axe dans malia

lights = ETree.SubElement(element, "lights")

if light_direct:
    sphere_light = ETree.SubElement(lights, "sphere_light")
    sphere_light.set("radius", str(radius))

    emittance = ETree.SubElement(sphere_light, "emittance")
    emittance.set("name", "Light")
    emittance.set("type", "uniform")

    rad = ETree.SubElement(emittance, "radiance")
    rad.set("value", str(radiance))

    rgb_color = ETree.SubElement(emittance, "rgb_color")
    rgb_color.set("r", str(rgb[0]))
    rgb_color.set("g", str(rgb[1]))
    rgb_color.set("b", str(rgb[2]))

    spec = ETree.SubElement(emittance, "spectrum")
    if "spd" in spectrum:
        spec.set("file", spectrum)
    else:
        spec.set("value", spectrum)

    position = ETree.SubElement(sphere_light, "position")
    position.set("x", str(sun_pos[0]))
    position.set("y", str(sun_pos[1]))
    position.set("z", str(sun_pos[2]))

else:
    area_light = ETree.SubElement(element, "area_light")

    emittance = ETree.SubElement(area_light, "emittance")
    emittance.set("name", "Area")
    emittance.set("type", "diffuse")

    rad = ETree.SubElement(emittance, "radiance")
    rad.set("value", str(radiance))
    spec = ETree.SubElement(emittance, "spectrum")
    if "spd" in spectrum:
        spec.set("file", spectrum)
    else:
        spec.set("value", spectrum)

if transform:
    print("transform pas encore operationnel")
    # transformations = ETree.SubElement(area_light, "transformations")
    # transform = ETree.SubElement(transformations, "transform")
    #
    # scale = ETree.SubElement(transform, "scale")
    # scale.set("x", "130.0")
    # scale.set("y", "115.0")
    # scale.set("z", "1.0")
    #
```

```
        # translation = ETree.SubElement(l_transform, "translation")
        # translation.set("x", "278")
        # translation.set("y", "548.79")
        # translation.set("z", "284.5")
        #
        # rotation = ETree.SubElement(l_transform, "rotation")
        # rot_angle = ETree.SubElement(l_rotation, "angle")
        # rot_angle.set("value", "0")
        # rot_axis = ETree.SubElement(l_rotation, "axis")
        # rot_axis.set("x", "0.0")
        # rot_axis.set("y", "0.0")
        # rot_axis.set("z", "0.0")

    return element


def msf_materials(element, malia_path=None, name=None, rgb=None, albedo=1.0):
    """
    :param element: xml scene
    :param malia_path:
    :param name: name or list of names of preset materials.
    Options are "all", "white", "red", "green", "Default OBJ". Default is "all"
    :param rgb:
    :param albedo:
    :return: scene with the materials needed
    """
    if name is None:
        name = "all"
    if rgb is None:
        rgb = [0.0, 0.6, 0.0]
    if malia_path is None:
        malia_path = glob('\\'.join(os.path.abspath(__file__).split("\\")[0:4]) + '/**/Malia-1.0.0-win64/bin/malia.exe',
recursive=True)[0].replace("\\", "/")

    materials = ETree.SubElement(element, "materials")

    if "Default OBJ" in name or name == "all":
        material_obj = ETree.SubElement(materials, "material")
        material_obj.set("name", "Default OBJ")
        material_obj.set("type", "principled")

        base_color = ETree.SubElement(material_obj, "base_color")
        material_rgb_color = ETree.SubElement(base_color, "rgb_color")
        material_rgb_color.set("r", str(rgb[0]))
        material_rgb_color.set("g", str(rgb[1]))
        material_rgb_color.set("b", str(rgb[2]))
        roughness = ETree.SubElement(material_obj, "roughness")
        roughness.set("value", "0.5")
        subsurface = ETree.SubElement(material_obj, "subsurface")
        subsurface.set("value", "0.0")
        sheen = ETree.SubElement(material_obj, "sheen")
        sheen.set("value", "0.0")
        sheentint = ETree.SubElement(material_obj, "sheenTint")
        sheentint.set("value", "0.5")
        metallic = ETree.SubElement(material_obj, "metallic")
        metallic.set("value", "0.0")
        specular = ETree.SubElement(material_obj, "specular")
```

```
    specular.set("value", "0.5")
    specularTint = ETree.SubElement(material_obj, "specularTint")
    specularTint.set("value", "0.0")
    anisotropic = ETree.SubElement(material_obj, "anisotropic")
    anisotropic.set("value", "0.0")
    clearcoat = ETree.SubElement(material_obj, "clearcoat")
    clearcoat.set("value", "0.0")
    clearcoat_gloss = ETree.SubElement(material_obj, "clearcoatGloss")
    clearcoat_gloss.set("value", "0.03")

if "white" in name or name == "all":
    material_white = ETree.SubElement(materials, "material")
    material_white.set("name", "white")
    material_white.set("type", "Lambert")

    mat_albedo = ETree.SubElement(material_white, "albedo")
    mat_albedo.set("value", str(albedo))

    spectrum = ETree.SubElement(material_white, "spectrum")
    # spectrum.set("file", glob(os.path.dirname(malia_path) + '/**/white.spd', recursive=True)[0].replace("\\", "/"))
    spectrum.set("file", "../_assets/white.spd")

if "green" in name or name == "all":
    material_green = ETree.SubElement(materials, "material")
    material_green.set("name", "green")
    material_green.set("type", "Lambert")

    mat_albedo = ETree.SubElement(material_green, "albedo")
    mat_albedo.set("value", str(albedo))

    spectrum = ETree.SubElement(material_green, "spectrum")
    # spectrum.set("file", glob(os.path.dirname(malia_path) + '/**/green.spd', recursive=True)[0].replace("\\", "/"))
    spectrum.set("file", "../_assets/green.spd")

if "red" in name or name == "all":
    material_red = ETree.SubElement(materials, "material")
    material_red.set("name", "red")
    material_red.set("type", "Lambert")

    mat_albedo = ETree.SubElement(material_red, "albedo")
    mat_albedo.set("value", str(albedo))

    spectrum = ETree.SubElement(material_red, "spectrum")
    # spectrum.set("file", glob(os.path.dirname(malia_path) + '/**/red.spd', recursive=True)[0].replace("\\", "/"))
    spectrum.set("file", "../_assets/red.spd")

if "light" in name or name == "all":
    material_light = ETree.SubElement(materials, "material")
    material_light.set("name", "light")
    material_light.set("type", "Lambert")

    mat_albedo = ETree.SubElement(material_light, "albedo")
    mat_albedo.set("value", str(albedo))

    spectrum = ETree.SubElement(material_light, "spectrum")
    spectrum.set("file", "../_assets/light.spd")

if "pot" in name or name == "all":
```

```
        material_pot = ETree.SubElement(materials, "material")
        material_pot.set("name", "pot")
        material_pot.set("type", "Lambert")

        mat_albedo = ETree.SubElement(material_pot, "albedo")
        mat_albedo.set("value", str(albedo))

        spectrum = ETree.SubElement(material_pot, "spectrum")
        spectrum.set("file", "../_assets/pot.spd")

    if "plaque" in name or name == "all":
        material_plaque = ETree.SubElement(materials, "material")
        material_plaque.set("name", "plaque")
        material_plaque.set("type", "Lambert")

        mat_albedo = ETree.SubElement(material_plaque, "albedo")
        mat_albedo.set("value", str(albedo))

        spectrum = ETree.SubElement(material_plaque, "spectrum")
        spectrum.set("file", "../_assets/plaque.spd")

    if "feuille" in name or name == "all":
        material_feuille = ETree.SubElement(materials, "material")
        material_feuille.set("name", "feuille")
        material_feuille.set("type", "Lambert")

        mat_albedo = ETree.SubElement(material_feuille, "albedo")
        mat_albedo.set("value", str(albedo))

        spectrum = ETree.SubElement(material_feuille, "spectrum")
        spectrum.set("file", "../_assets/spectre_trans_luzerne_corrige.spd")

    if "caribu" in name or name == "all":
        material_caribu = ETree.SubElement(materials, "material")
        material_caribu.set("name", "caribu")
        material_caribu.set("type", "Lambert")

        mat_albedo = ETree.SubElement(material_caribu, "albedo")
        mat_albedo.set("value", str(albedo))

        spectrum = ETree.SubElement(material_caribu, "spectrum")
        spectrum.set("file", "../_assets/caribu_light.spd")

    return element


def obj2msf(obj_file, malia_path=None, date=None, material_name="Default OBJ", background=True,
extra_obj=None,
            bg_rgb=None, mat_rgb=None, albedo=1.0, translation=None, scale=None, rotation=None,
            light_direct=True, sphere_light_radius=0.1, radiance=1000.0, sun_pos=None, out_path=None):
    """
    :param obj_file: path to an obj file
    :param malia_path: path to malia.exe
    :param date:
    :param material_name:
    :param background: logical
    :param extra_obj: logical. If True add 'pot', 'table' and 'plaque' in the scene.
```

```
    :param bg_rgb:
    :param mat_rgb:
    :param albedo:
    :param translation:
    :param scale: int or [x, y, z]. increase size if < 1 and decrease size if > 1.
    :param rotation: obj transformation, either angle (default rotation around y axis) or [angle (degree), [x, y, z]]
    :param light_direct:
    :param sphere_light_radius:
    :param radiance:
    :param sun_pos:
    :param out_path: path to save the msf
    :return:
    """
    geom_transform = False
    if translation is not None:
        geom_transform = True
    if scale is not None:
        geom_transform = True
        if isinstance(scale, int) or isinstance(scale, float):
            scale = [1 / scale, 1 / scale, 1 / scale]
    if rotation is not None:
        geom_transform = True
        if isinstance(rotation, int) or isinstance(scale, float):
            rotation = [rotation, [0.0, 1.0, 0.0]]

    if os.path.isfile(obj_file) is False:
        raise TypeError("File not found")

    if malia_path is None:
        malia_path = glob('\\'.join(os.path.abspath(__file__).split("\\")[0:4]) + '/**/Malia-1.0.0-win64/bin/malia.exe',
recursive=True)[0]

    filename = ".".join(os.path.basename(obj_file).split(".")[:-1]) + ".msf"
    scene = ETree.Element("scene")
    if out_path is None:
        out_dir = os.path.join(os.path.dirname(malia_path), "scenes")
        out_path = os.path.join(out_dir, filename)
        print(out_path)
    else:
        out_path = os.path.dirname(out_path) + '/' + os.path.basename(out_path) + ".msf"
    # obj_geom_file = obj_file.replace(out_dir, ".").replace("\\", "/")
    obj_geom_file = "../obj/" + os.path.basename(obj_file)

    # Light sources
    if background:
        scene = msf_background(scene, rgb=bg_rgb)

    scene = msf_light(scene, date=date,
                      spectrum="../_assets/archiduo_light.spd",  # glob(os.path.dirname(malia_path) + '/**/archiduo_light.spd',
recursive=True)[0].replace("\\", "/"),
                      light_direct=light_direct, radius=sphere_light_radius, radiance=radiance, sun_pos=sun_pos)

    # Materials
    scene = msf_materials(scene, malia_path, "all", mat_rgb, albedo)

    # Obj geometrie
```

```
shapes = ETree.SubElement(scene, "shapes")
shape_obj = ETree.SubElement(shapes, "shape")
shape_obj.set("ref_material", material_name)
mesh = ETree.SubElement(shape_obj, "mesh")
obj_geom = ETree.SubElement(mesh, obj_file[-3:])
obj_geom.set("file", obj_geom_file)
# obj_geom.set("file", obj_file)

try:
    is_date = [len(i) == 10 for i in filename.split('_')]  # identifie le format date a 10 caracteres
    is_dist = [i for i, val in enumerate(is_date) if val][0] + 2  # la distance est 2 positions apres la date
    distance = int(filename.split('_')[is_dist])
except Exception:
    distance = 0

if geom_transform:
    transformations = ETree.SubElement(shape_obj, "transformations")
    transform = ETree.SubElement(transformations, "transform")

    if scale is not None:
        geom_scale = ETree.SubElement(transform, "scale")
        geom_scale.set("x", str(scale[0]))
        geom_scale.set("y", str(scale[1]))
        geom_scale.set("z", str(scale[2]))

    if translation is not None:
        geom_translation = ETree.SubElement(transform, "translation")
        geom_translation.set("x", str(translation[0] + distance))
        geom_translation.set("y", str(translation[1]))
        geom_translation.set("z", str(translation[2]))

    if rotation is not None:
        geom_rot = ETree.SubElement(transform, "rotation")
        rot_angle = ETree.SubElement(geom_rot, "angle")
        rot_angle.set("value", str(rotation[0]))
        rot_axis = ETree.SubElement(geom_rot, "axis")
        rot_axis.set("x", str(rotation[1][0]))
        rot_axis.set("y", str(rotation[1][1]))
        rot_axis.set("z", str(rotation[1][2]))

if extra_obj is not None and extra_obj.lower() == "archiduo":
    # Pot geometrie
    pot_geom_file = "../_assets/pot_archiduo.obj"
    shape_pot = ETree.SubElement(shapes, "shape")
    shape_pot.set("ref_material", "pot")
    mesh_pot = ETree.SubElement(shape_pot, "mesh")
    pot_geom = ETree.SubElement(mesh_pot, "obj")
    pot_geom.set("file", pot_geom_file)

    # Plaque geometrie
    plaque_geom_file = "../_assets/plaque_archiduo.obj"
    shape_plaque = ETree.SubElement(shapes, "shape")
    shape_plaque.set("ref_material", "plaque")
    mesh_plaque = ETree.SubElement(shape_plaque, "mesh")
    plaque_geom = ETree.SubElement(mesh_plaque, "obj")
    plaque_geom.set("file", plaque_geom_file)
    plaque_transformation = ETree.SubElement(shape_plaque, "transformations")
    plaque_transform = ETree.SubElement(plaque_transformation, "transform")
```

```
        plaque_translation = ETree.SubElement(plaque_transform, "translation")
        plaque_translation.set("x", "100")
        plaque_translation.set("y", "0")
        plaque_translation.set("z", "0")

        # Table geometrie
        table_geom_file = "../_assets/table_archiduo.obj"
        shape_table = ETree.SubElement(shapes, "shape")
        shape_table.set("ref_material", "white")
        mesh_table = ETree.SubElement(shape_table, "mesh")
        table_geom = ETree.SubElement(mesh_table, "obj")
        table_geom.set("file", table_geom_file)

    if extra_obj is not None and extra_obj.lower() == "archimono":
        pot_geom_file = "../_assets/pot_archimono_mock.obj"
        shape_pot = ETree.SubElement(shapes, "shape")
        shape_pot.set("ref_material", "white")
        mesh_pot = ETree.SubElement(shape_pot, "mesh")
        pot_geom = ETree.SubElement(mesh_pot, "obj")
        pot_geom.set("file", pot_geom_file)

        pot_transformations = ETree.SubElement(shape_pot, "transformations")
        pot_transform = ETree.SubElement(pot_transformations, "transform")
        pot_scale = ETree.SubElement(pot_transform, "scale")
        pot_scale.set("x", str(10))
        pot_scale.set("y", str(10))
        pot_scale.set("z", str(10))

        # pot_translation = ETree.SubElement(pot_transform, "translation")
        # pot_translation.set("x", str(translation[0] + distance))
        # pot_translation.set("y", str(translation[1]))
        # pot_translation.set("z", str(translation[2]))

        pot_rot = ETree.SubElement(pot_transform, "rotation")
        pot_angle = ETree.SubElement(pot_rot, "angle")
        pot_angle.set("value", str(rotation[0]))
        pot_axis = ETree.SubElement(pot_rot, "axis")
        pot_axis.set("x", str(rotation[1][0]))
        pot_axis.set("y", str(rotation[1][1]))
        pot_axis.set("z", str(rotation[1][2]))

    # Sortie
    if not os.path.exists(os.path.dirname(out_path)):
        os.makedirs(os.path.dirname(out_path))
    with open(out_path, 'w') as f:
        original_stdout = sys.stdout
        sys.stdout = f  # Change the standard output to the file we created.
        print(prettify(scene))
        sys.stdout = original_stdout  # Reset the standard output to its original value

    return out_path


def obj2mcf(filename, width=640.0, height=480.0, cam_position=None, cam_lookat=None, fov=60, out_path=None):
    """
    :param filename:
    :param width:
```

```
    :param height:
    :param cam_position: [x, y, z]. Attention: x correspond a la profondeur, y correspond a la hauteur et z au
deplacement lateraux
    :param cam_lookat:
    :param fov:
    :param out_path:
    :return:
    """
    if cam_position is None:
        try:
            is_date = [len(i) == 10 for i in os.path.basename(filename).split('_')]  # identifie le format date a 10 caracteres
            is_dist = [i for i, val in enumerate(is_date) if val][0] + 2  # la distance est 2 positions apres la date
            distance = int(os.path.basename(filename).split('_')[is_dist])
            cam_pos_x = distance / 10  # distance a rajouter constatee apres tests
        except:
            cam_pos_x = 10
        cam_position = [cam_pos_x, 3.3, 0]

    if cam_lookat is None:
        cam_lookat = [-cam_position[0] if cam_position[0] != 0 else -10, cam_position[1], cam_position[2]]

    cameras = ETree.Element("cameras")
    camera = ETree.SubElement(cameras, "camera")
    camera.set("id", "Camera")
    camera.set("type", "pinhole")

    position = ETree.SubElement(camera, "position")
    position.set("x", str(cam_position[0]))
    position.set("y", str(cam_position[1]))
    position.set("z", str(cam_position[2]))

    lookat = ETree.SubElement(camera, "lookat")
    lookat.set("x", str(cam_lookat[0]))
    lookat.set("y", str(cam_lookat[1]))
    lookat.set("z", str(cam_lookat[2]))

    up = ETree.SubElement(camera, "up")
    up.set("x", "0")
    up.set("y", "1")
    up.set("z", "0")

    fovy = ETree.SubElement(camera, "fovy")
    fovy.set("value", str(fov))

    lens = ETree.SubElement(camera, "lens")
    lens.set("focal_length", "0.03117")

    sensor = ETree.SubElement(camera, "sensor")
    sensor.set("spp", "128")  # sensor.set("spp", "1024")

    res = ETree.SubElement(sensor, "resolution")
    res.set("height", str(height))
    res.set("width", str(width))

    size = ETree.SubElement(sensor, "size")
    size.set("height", "24.0")
    size.set("width", "36.0")
```

```
    dist = ETree.SubElement(sensor, "distance")
    dist.set("value", "0.1590")

    if filename is None:
        print(prettify(cameras))
    else:
        if out_path is None:
            out_path = os.path.join(os.path.abspath(filename).split("scenes")[0], "scenes", os.path.basename(filename[:-3])
+ "mcf")  # attention va bugguer si pas de dossier "scenes"
        else:
            out_path = os.path.dirname(out_path) + '/' + os.path.basename(out_path) + ".mcf"
        # if os.path.exists(out_path) is False:
        with open(out_path, 'w') as f:
            original_stdout = sys.stdout
            sys.stdout = f  # Change the standard output to the file we created.
            print(prettify(cameras))
            sys.stdout = original_stdout  # Reset the standard output to its original value

    return out_path


# Run malia
def run_malia_sensor(obj, malia_path, output_name, path_these, info_manip,
                     date_heure, obj_scale=0.1, rot=0,
                     x_cam=0, y_cam=0, z_cam=0,
                     lw="400:800:10", fov_camera=60, fov_final=3,
                     recalage=False, reconstruction=None, expe=None,
                     archive_name=None, starttime=None, extra_obj=None,
                     i=None, imax=None, timing=None):
    t = time()
    output_name = output_name + "_id" + str(t)  # ajoute un id unique pour le multiprocessing
    dir_path = os.path.dirname(os.path.dirname(obj))
    scene_path = os.path.join(dir_path, "1. msf scene", output_name)
    debug = False if sys.gettrace() is None else True  # Test if pycharm is in debug mode

    # save this script
    if debug is False:
        src = os.path.realpath(__file__)
        dst = os.path.join(dir_path, starttime + '_' + os.path.basename(__file__))
        if os.path.isfile(dst) is False:
            copy(src, dst)

    # preparation de la scene
    material = "caribu"  # "feuille"
    msf_path = obj2msf(obj, malia_path, date=date_heure, material_name=material, scale=obj_scale, rotation=rot,
extra_obj=extra_obj, out_path=scene_path)
    mcf_path = obj2mcf(obj, 800, 800, fov=fov_camera, cam_position=[x_cam, z_cam, y_cam], out_path=scene_path)

    # malia
    malia_out = os.path.join(dir_path, "2. malia img", output_name + ".hdr")
    if not os.path.exists(os.path.dirname(malia_out)):
        os.makedirs(os.path.dirname(malia_out))
    if debug:
        print(malia_path, "-scene", msf_path, "-i")
        # malia_rgb_path = os.path.join(os.path.dirname(malia_path), "malia_rgb.exe")
        if os.path.exists(malia_out):
            call(args=[malia_path, "-scene", msf_path, "-i"])  # visualise la scene
```

```
        else:
            call(args=[malia_path, "-scene", msf_path, "-o", malia_out, "-wr", lw], stdout=DEVNULL)  # enregistre la
scene
    else:
        # if os.path.exists(malia_out) is False:
        call(args=[malia_path, "-scene", msf_path, "-o", malia_out, "-wr", lw], stdout=DEVNULL)  # DEVNULL
supprime la console de malia

    # traitement hdr
    hdr_out = os.path.join(dir_path, "3. zeta hdr", output_name + ".csv")
    csv_path = read_hdr(malia_out, out_path=hdr_out, wl_range=[600, 800], fov_voulu=fov_final,
fov_mcf=fov_camera, distance=x_cam, rotation=rot)

    # calcul du zeta
    strftime("%Y-%m-%d_%HH%M")
    zeta_file = '_'.join((starttime, 'zeta simule par malia', expe, reconstruction, 'FOV' + str(fov_final)))
    zeta_file = zeta_file + "_recalage" if recalage else zeta_file
    zeta_path = os.path.join(dir_path, zeta_file + ".csv")
    zeta_from_csv(csv_path, expe_path=path_these, date_heure=date_heure, info_manip=info_manip,
outpath=zeta_path, expe=expe, reconstruction=reconstruction, runtime=t)

    # Compression et suppression des fichiers en fin d'analyse pour reduire l'espace pris par les fichier
    if archive_name is not None:
        archive(msf_path, zip_name=archive_name)
        archive(mcf_path, zip_name=archive_name)
        archive(malia_out, zip_name=archive_name)
        archive(malia_out[:-3] + 'raw', zip_name=archive_name)
        archive(csv_path, zip_name=archive_name)

    if i is not None:
        i.value += 1
        print_progress_bar(i.value, imax, prefix=str(i.value) + "/" + str(imax), start_time=timing)
    else:
        print(i)


# Calcul du zeta ####
def cache_photo(img):
    hauteur, largeur = img.shape  # taille de image
    xc, yc = hauteur / 2, largeur / 2  # centre en x et y
    r = min(xc, yc)  # rayon du cercle

    x, y = np.meshgrid(np.arange(hauteur), np.arange(largeur))  # coordonnes de chaque pixel de l'image
    d2 = (x - xc) ** 2 + (y - yc) ** 2  # distance au carre depuis le centre du cercle
    mask = d2 > r ** 2  # mask est True en dehors du cercle

    img[mask] = 0  # suppression des donnees

    return img


def zeta_from_csv(path, expe_path, info_manip=None, date_heure=None, outpath=None, rc=660, rs=730, step=0,
expe="archimono", reconstruction=None, reshape=True, runtime=None):
    """
    MALIA - Calcul le zeta a partir du csv produit par la fonction read_hdr()
    :param path: path to csv
    :param expe_path:
```

```
    :param outpath: path to save the results
    :param rc: red lengthwave
    :param rs: far-red lengthwave
    :param step: lengthwave area around red and far-red to consider
    :param expe: "archimono" or "archiduo"
    :param reshape:
    :return:
    """
    spec_serre = pandas.read_csv(path, sep=";")

    # Recuperation des infos de MALIA
    malia_fov = list(set(spec_serre.fov_cor))[0]
    rotation = list(set(spec_serre.rotation))[0]
    distance = list(set(spec_serre.distance))[0]  # transform to give each band a column

    # Calcul du zeta de MALIA
    rc_list = list(spec_serre[(spec_serre.band >= rc - step) & (spec_serre.band <= rc + step)].value)
    rs_list = list(spec_serre[(spec_serre.band >= rs - step) & (spec_serre.band <= rs + step)].value)
    # rs_list = [rs if rs != 0 else rc for rc, rs in zip(rc_list, rs_list)]  # empeche les bugs si rs == 0
    is_empty = len([rc / rs for rc, rs in zip(rc_list, rs_list) if rs != 0 and rc / rs < 1]) == 0
    if is_empty:
        # Si aucun pixel de plante est present la mesure de zeta est mise en NA
        rc_moyen = np.mean(rc_list)
        rs_moyen = np.mean(rs_list)
        rc_rs_moyen = "NA"
    else:
        rc_moyen = np.mean(rc_list)
        rs_moyen = np.mean(rs_list)
        rc_rs_moyen = round(rc_moyen / rs_moyen, 3)

    # Import de la mesure de zeta
    if expe is not None and expe.lower() == "archimono":
        date_mesure = date_heure if date_heure is not None else info_manip["date_heure"]
        id_plante = info_manip["semis"]
        zeta_maya = info_manip["zeta"]

    elif expe is not None and expe.lower() == "archiduo":
        date_mesure = date_heure if date_heure is not None else info_manip["date_heure"]
        id_plante = info_manip["semis"]
        zeta_maya = info_manip["zeta"]

    else:
        date_mesure = strftime("%Y-%m-%d %HH%M")
        id_plante = "NA"
        zeta_maya = "NA"

    if runtime is not None:
        runtime = time() - runtime

    # Save the computed zeta
    zeta_path = os.path.join(os.path.dirname(path), "zeta simule par malia.csv") if outpath is None else outpath
    if os.path.exists(zeta_path) is False:
        open(zeta_path,
'w').write('nom;expe;date_manip;date_simul;id;fov;distance;reconstruction;rotation;zeta_maya;rc_malia;rs_malia;zeta_
malia;runtime\n')

    with open(zeta_path, 'a') as output:
```

```
        text = (os.path.basename(path[:-4]), expe, date_mesure, date_heure, id_plante,
                str(malia_fov), str(distance), reconstruction, str(rotation),
                str(zeta_maya), str(rc_moyen), str(rs_moyen), str(rc_rs_moyen),
                runtime)
        output.write((("%s;" * len(text))[:-1] + "\n") % text)
        output.close()

    # Modification du tableau d'entree
    if reshape:
        ids = [n for n in list(spec_serre.columns) if n not in ('band', 'value')]
        new_csv = spec_serre.pivot_table(index=ids, columns='band', values='value').reset_index()
        new_csv.to_csv(path, sep=";", index=False)

    return path, zeta_path


def read_hdr(path, out_path=None, wl_range=None, fov_voulu=None, fov_mcf=None, distance=None, rotation=0):
    """
    :param path: path to hdr file
    :param out_path: path to save the csv file
    :param wl_range: min and max wavelength
    :param fov_voulu: fov to get from the image
    :param fov_mcf: fov of the image
    :param distance: distance between camera and plant
    :param rotation: only if msf had rotation
    :return: path to the saved csv
    """
    if wl_range is None:
        wl_range = [400, 800]
    spectral.settings.envi_support_nonlowercase_params = True  # bug if False
    img = spectral.open_image(path)
    nb_bands = img.nbands
    wavelengths = img.metadata["Wavelength"][0:nb_bands]
    r_fr_id_bands = [x for x in range(len(wavelengths)) if wl_range[0] <= int(wavelengths[x]) <= wl_range[1]]

    if fov_mcf is None:
        mcf_path = glob(os.path.join(r'..\Logiciels\Malia-1.0.0-win64\bin\scenes', os.path.basename(path)[:-3] + "mcf"),
recursive=True)[0]
        elem_list = ETree.parse(mcf_path).getroot()[0]
        index_fov = [i for i in range(len(elem_list)) if elem_list[i].tag == "fovy"][0]
        fov_mcf = int(ETree.parse(mcf_path).getroot()[0][index_fov].attrib["value"])

    if fov_voulu is None:
        fov_voulu = fov_mcf

    if distance is None:
        mcf_path = glob(os.path.join(r'..\Logiciels\Malia-1.0.0-win64\bin\scenes', os.path.basename(path)[:-3] + "mcf"),
recursive=True)[0]
        elem_list = ETree.parse(mcf_path).getroot()[0]
        index_position = [i for i in range(len(elem_list)) if elem_list[i].tag == "position"][0]
        distance = int(ETree.parse(mcf_path).getroot()[0][index_position].attrib["x"])

    if fov_voulu == fov_mcf:
        list_x = range(img.nrows - 1)
        list_y = range(img.ncols - 1)
    else:
        largeur = (img.ncols / fov_mcf) * fov_voulu
```

```
    hauteur = largeur
    list_x = range(int(img.nrows / 2 - hauteur / 2), int(img.nrows / 2 + hauteur / 2))  # range(img.nrows - 1)
    list_y = range(int(img.ncols / 2 - largeur / 2), int(img.ncols / 2 + largeur / 2))

if out_path is None:
    dir_path = os.path.dirname(os.path.abspath(__file__))
    out_path = os.path.join(dir_path, 'malia img', os.path.basename(path)[:-3] + 'csv')

if not os.path.exists(os.path.dirname(out_path)):
    os.makedirs(os.path.dirname(out_path))
elif os.path.exists(out_path):
    os.remove(out_path)

open(out_path, 'w').write('x;y;band;fov_malia;fov_cor;distance;rotation;value\n')
for band in r_fr_id_bands:
    img_band = np.array(np.memmap.tolist(img._memmap[band]))
    img_band = cache_photo(img_band)
    for x in list_x:
        with open(out_path, 'a') as output:
            col_values = img_band[x][list_y]
            text = "".join(str(x) + ";" +
                           str(list_y[y]) + ";" +
                           str(wavelengths[band]) + ";" +
                           str(fov_mcf) + ";" +
                           str(fov_voulu) + ";" +
                           str(distance) + ";" +
                           str(rotation) + ";" +
                           str(col_values[y]) + "\n"
                           for y in range(len(col_values)))
            output.write(text)
            output.close()

return out_path
```

asc2obj.py

Permet de convertir un nuage de point en objet 3D avec les méthodes « Ball Pivoting » et « Marching cubes ».

```
import os
import sys
import pandas
import pymeshlab
import rpy2.robjects as robjects
import multiprocessing
import subprocess
from glob import glob
from time import time
from malia_tools import print_progress_bar, grep


def create_obj(list_asc, thr=1.0, CPUnb=10, method="bp", out_dir=None):
    if len(list_asc) == 0:
        sys.exit("Erreur dans l'import des fichiers")

    start_time = int(time())
    asc2obj_apss_args = []
    for path in list_asc:
        rotation = None
        remove_borders = 1
        nb_neighbour = 3
        asc2obj_apss_args.append((path, out_dir, rotation, remove_borders, nb_neighbour, thr))

    pool = multiprocessing.Pool(processes=CPUnb)
    i = multiprocessing.Manager().Value('i', 0)
    for args in asc2obj_apss_args:
        args = args + (i, len(asc2obj_apss_args), time())
        if method == "bp":
            pool.apply_async(asc2obj_ball_pivoting, [args[0]])
        if method == "mc":
            pool.apply_async(asc2obj_apss, args)
    pool.close()
    pool.join()

    print("")
    print_progress_bar(1, 1, fill="", border="", start_time=start_time, prefix="\n", suffix="\r")


def asc2obj_apss(asc_path, out_dir=None, rotation=None, remove_borders=0, nb_neighbour=16, thr=1, i=None,
imax=None, timing=None):
    """
    asc_path
    out_dir
    rotation
    remove_borders
    nb_neighbour
    """
    asc_path = os.path.abspath(asc_path)
    if os.path.isfile(os.path.abspath(asc_path)) is False:
        sys.exit("Fichier " + os.path.basename(asc_path) + " introuvable")
```

```
    obj_name = os.path.basename(asc_path)  # [:-3] + "obj"
    filetype = os.path.splitext(asc_path)[1]
    if out_dir is None:
        out_dir = os.path.dirname(os.path.dirname(asc_path))
    out_path = out_dir + "/obj apss/" + obj_name

    if not os.path.exists(os.path.dirname(out_path)):
        os.makedirs(os.path.dirname(out_path))

    # Reconstruction
    ms = pymeshlab.MeshSet()
    ms.load_new_mesh(asc_path)
    ms.apply_filter("compute_normals_for_point_sets", k=nb_neighbour)
    ms.apply_filter("marching_cubes_apss", filterscale=3, maxprojectioniters=1, resolution=500)
    out_path = out_path[:-4] + ".obj"
    ms.save_current_mesh(out_path, save_vertex_normal=False)

    # Supression des faces eloignees du nuage de point
    robjects.r.source("function_correction_apss.R")
    remove_vertice = robjects.globalenv['remove_vertice']
    remove_vertice(asc_path, obj_path=out_path, out_path=out_path, close_threshold=thr)  # plus close_threshold est
faible plus les points sont supprimes

    # Supression des points sans face
    ms_obj = pymeshlab.MeshSet()
    ms_obj.load_new_mesh(out_path)
    ms_obj.apply_filter("select_faces_with_edges_longer_than", threshold=0)  # selectionne toute les faces
    ms_obj.apply_filter("select_vertices_from_faces")  # selectionne les vertices reliees aux faces
    ms_obj.apply_filter("invert_selection", invfaces=True, invverts=True)  # selectionne les vertices sans face
    ms_obj.apply_filter("delete_selected_vertices")

    # Suppression des petits elements isoles (attention: peut enlever des feuilles)
    ms_obj.apply_filter("select_small_disconnected_component")
    ms_obj.apply_filter("delete_selected_faces_and_vertices")

    if rotation is not None:
        # rotation recommandee : -90
        ms_obj.apply_filter("transform_rotate", rotaxis='X axis', angle=rotation)

    # Reduction de la largeur des reconstructions
    if remove_borders == 0:
        # Lissage des bordures des feuilles
        ms_obj.apply_filter("select_border")
        ms_obj.apply_filter("laplacian_smooth_surface_preserving", selection=False, angledeg=20, iterations=100)

    for nb_bd in range(remove_borders):
        ms_obj.apply_filter("select_border")
        ms_obj.apply_filter("delete_selected_faces_and_vertices")

        # Lissage des bordures des feuilles
        ms_obj.apply_filter("select_border")
        ms_obj.apply_filter("laplacian_smooth_surface_preserving", selection=False, angledeg=20, iterations=100)

        # Export pour chaque largeur
        # rb_out_path = out_path[:-4] + "_" + str(nb_bd) + "rb.obj"
        # ms_obj.save_current_mesh(rb_out_path, save_vertex_normal=False)

    ms_obj.save_current_mesh(out_path, save_vertex_normal=False)
```

```
    if i is not None:
        i.value += 1
        print_progress_bar(i.value, imax, prefix=str(i.value) + "/" + str(imax), start_time=timing)


def asc2obj_ball_pivoting(asc_path):
    """
    asc_path - str. path to the asc file
    """
    out_path = os.path.dirname(os.path.dirname(asc_path)) + "/obj bp/" + os.path.basename(asc_path)[:-3] + "obj"
    if not os.path.exists(os.path.dirname(out_path)):
        os.makedirs(os.path.dirname(out_path))

    ms = pymeshlab.MeshSet()
    ms.load_new_mesh(asc_path)
    ms.apply_filter("compute_normals_for_point_sets", k=16)
    ms.apply_filter("surface_reconstruction_ball_pivoting", clustering=10, creasethr=150)
    ms.save_current_mesh(out_path)


def fuse_obj(list_obj, list_coord, to=None):
    i = 0
    ms = pymeshlab.MeshSet()

    for obj in list_obj:
        ms.load_new_mesh(obj)
        ms.apply_filter("transform_translate_center_set_origin",
                        traslmethod=0,
                        axisx=float(list_coord['x'][i]) * 10,
                        axisy=float(list_coord['y'][i]))
        i += 1

    ms.flatten_visible_layers()  # Fusionne tout les meshs avant l'export
    output_path = to if to is not None else os.path.dirname(list_obj[0]) + "/fuse_obj.obj"
    ms.save_current_mesh(output_path, save_vertex_normal=False)
    print("'obj' saved at " + output_path)


if __name__ == '__main__':
    expe_path = r"C:\Users\Arthur\Documents\Couturier Arthur\These\Experimentation"
    expe = "NA"  #"peuplement"  # "cj"  # "ArchiMono"  # "ArchiDuo"  #

    if expe.lower() == "archimono":
        list_asc = glob(os.path.join(expe_path, "ArchiMono", "**", "sans_pot*.asc"), recursive=True)  # Expe ArchiMono
        create_obj(list_asc, thr=0.5)

    elif expe.lower() == "archiduo":
        list_asc = glob(os.path.join(expe_path, "ArchiDuo", "**", "cor360*.asc"), recursive=True)  # Expe ArchiDuo
        create_obj(list_asc, thr=0.5)

    elif expe.lower() == "archiprintemps":
        list_asc = glob(os.path.join(expe_path, "ArchiPrintemps", "**", "Reconstruction 3D", "*.ply"), recursive=True)
[1:2]
        create_obj(list_asc, thr=0.5)

    elif expe.lower() == "cj":
        list_asc = glob(os.path.join(expe_path, "Cinetique journaliere", "**", "cor360_*.asc"), recursive=True)
```

```
    create_obj(list_asc, thr=1)

elif expe.lower() == "peuplement":
    plmt_path = os.path.join(expe_path, "Cinetique journaliere", "scans", "2020-11-26")
    pos_plt = glob(os.path.join(plmt_path, "disposition peuplement 2020-11-26.csv"), recursive=True)[0]

    disposition = pandas.read_csv(pos_plt, delimiter=";")
    list_obj = [glob(os.path.join(plmt_path, "obj bp", "*" + id + "*"))[0] for id in disposition['id']]
    list_coord = disposition[['x', 'y']]

    fuse_obj(list_obj, list_coord, to=os.path.join(plmt_path, "peuplement dense.obj"))
```