Rapport de la procédure Simpraise

Effectué par

Thibault RAQUET

CDD/stagiaire INRAE

03/02/2020 – 31/07/2020

Objet de la procédure :
Couplage d'un modèle de morphogénèse avec un modèle de génétique pour simuler les générations successives d'un peuplement de ray-grass anglais

Tuteur professionnel : Didier COMBES
Tuteur enseignant : Benjamin BOUTIN

Sommaire

I) Le modèle lgrass

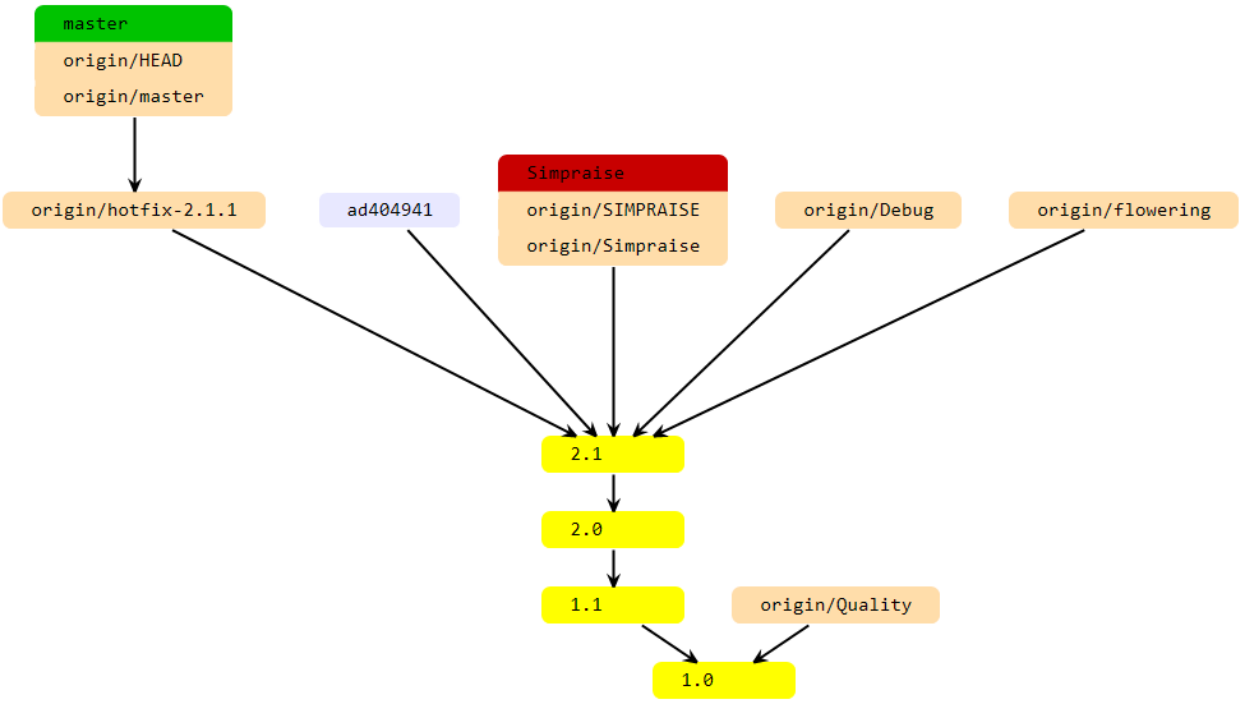
1) Principe du modèle :

Lgrass est un modèle de morphogénèse de plante utilisé au sein de l'unité pluridisciplinaire prairies et plantes fourragères (URP3F) permettant de simuler la croissance de gazon de type ray-grass anglais à partir de l'état de graine.

Une description précise du modèle est présente dans le rapport de stage de Pierre Guinard disponible en introduction.

2) Branches :

L'URP3F travaille avec l'outil git, se qui dans le cas de lgrass permet à chacun de travailler sur sa propre version ou branche du modèle, et d'en approfondir certains modules sans modifier la version originale. Les maintenances régulières permettent de regrouper les avancements menés dans chaque branche afin de mettre à jour la version principale du modèle. Voici la structure du modèle lgrass sur git en 07/2020 :



La branche Simpraise sur laquelle je me place (en rouge sur le schéma) fait l'objet de ce rapport. Les travaux sur cette branche sont en lien direct avec ceux menés par Simon Rouet dans le cadre de sa thèse *(sujet, dates de la thèse)* et qui est l'auteur de la branche flowering (ci-dessus). En effet, l'un des objectifs de Simon est d'implanter la gestion de la floraison du ray-grass dans lgrass, qui est un point essentiel et dont je présenterai certains phénomènes importants dans le cadre de la procédure Simpraise dans la partie qui lui est consacrée.

3) Simpraise :

L'objectif principal de la branche Simpraise est de réaliser un couplage du modèle lgrass avec un modèle de génétique permettant d'assurer la reproduction d'un couvert végétal de ray-grass en prenant en compte l'évolution génétique des populations successives. *(Présentation du modèle génétique si possible)*

Celle-ci est responsable de la variation du facteur de croissance de chaque plante, appelé le paramètre C dans lgrass. Plus la valeur de ce paramètre est élevée, plus la taille de la plante associée sera grande (feuilles, talles, épis).

Il y a plusieurs étapes dans la réalisation du couplage :

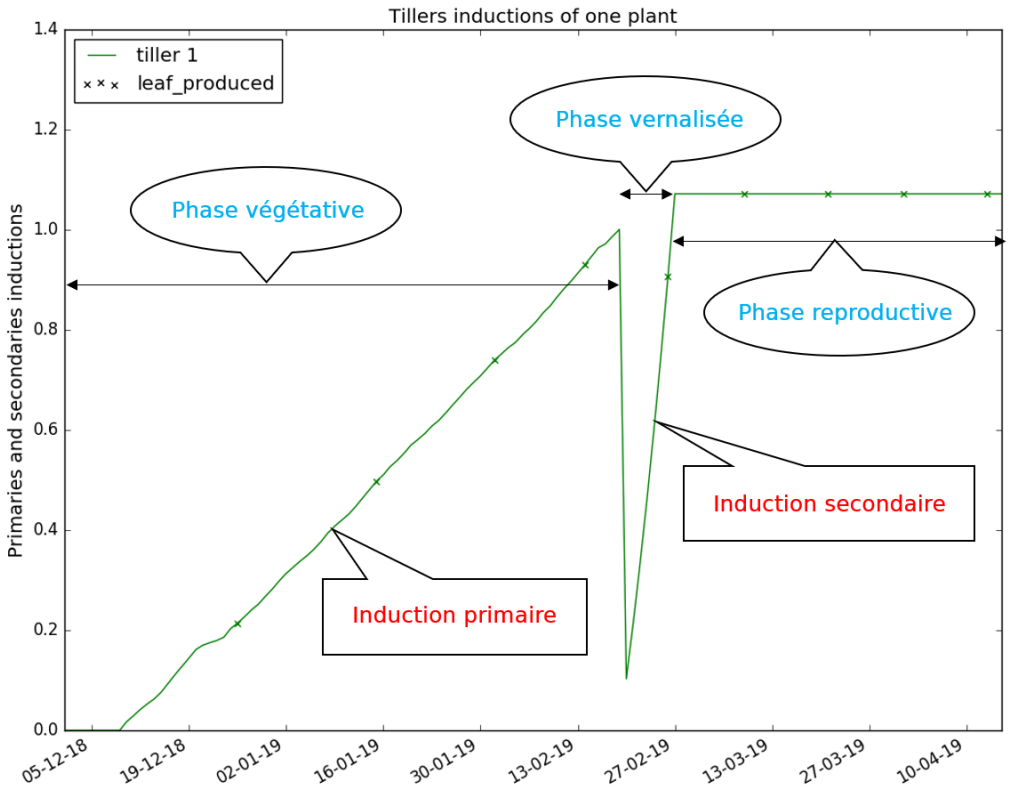
- Récupérer les travaux de Simon sur la floraison implantés dans lgrass
- A partir de cette version, réaliser la production de graines après floraison à partir de plusieurs méthodes détaillées dans la partie couplage
- Etudier les entrées/sorties du modèle génétique, et assurer l'acquisition et la conversion du paramètre de croissance en paramètre C pour lgrass
- Simuler la croissance des plantes jusqu'à leur production de graines via lgrass, et affecter une parenté à celles-ci pour conserver leur empreinte génétique
- Sélectionner aléatoirement les graines parmi celles produites pour la génération suivante et créer une matrice de croisement des graines
- Enchaîner les exécutions des modèles génétique et morphologique jusqu'à obtenir le nombre de générations successives souhaité

II) Cycle de reproduction d'une plante

1) Inductions :

Pour qu'une talle de ray-grass puisse fleurir, celle-ci a tout d'abord besoin d'une période de jours courts à basse température (induction primaire) à l'issue de laquelle la talle est dite vernalisée. Puis d'une période de jours longs à température printanière (induction secondaire). Suite à celle-ci, la talle entre en phase de reproduction.

Ci-dessous la courbe d'inductions primaire et secondaire d'une talle simulée avec lgrass avec les conditions météo du 12/2018 au 05/2019. Les nouvelles feuilles produites sur la talle sont symbolisées par des croix sur la courbe.



2) Floraison :

Pour qu'une talle puisse fleurir, elle doit se trouver en phase de reproduction. Elle débute alors la production de son épi, lui-même constitué d'épillets qui contiendront les graines, et d'entrenœuds. Les talles qui fleurissent sont celles qui finalisent la production de cet épi et assurent donc la pérennité de la plante, tandis que les autres talles en assurent la survie, attendant la prochaine période de jours courts à températures basses pour vernaliser.

Le nombre d'épillets produit par une talle est directement lié au nombre de feuilles qu'elle produit, dépendant entre autres du paramètre de croissance de la plante.

Il y a deux méthodes de production de graines implémentées dans lgrass : la première compte le nombre total d'épillets dans une plante auquel on ajoute un nombre de graines dans un épillet (jusqu'ici on lui en donne 3).

La seconde se base sur les résultats des expérimentations présentés dans le rapport de Pierre Guinard, reliant via une régression linéaire le nombre de tiges (talles fleuries) avec le nombre de graines au sein d'une population. Cette seconde méthode ne permet pas d'identifier clairement la mère de la graine.

Les deux méthodes présentent jusqu'ici une différence significative pour le nombre de graines (jusqu'à 100 pour une plante dans la première méthode, et 50 dans la seconde). Après conseil, la méthode de calcul via une régression semble moins précise du fait d'une corrélation imparfaite et d'un calcul ne se faisant qu'à l'échelle du couvert et non de la plante. On préfèrera donc la méthode prenant en compte le nombre d'épillets, bien que celle-ci nécessite encore quelques ajustements et vérifications.

3) Reproduction :

La reproduction des plantes dans Simpraise, telle qu'elle est construite actuellement, est la transition entre l'exécution du modèle morphologique et du modèle génétique. Elle débute en fin de simulation de lgrass en recueillant les épillets produits par talle pour chaque plante, et modélise une graine à partir du couple mère-père dont elle provient. Sa mère est la plante de laquelle l'épillet concerné est issu, et son père est l'une des autres plantes du couvert sélectionnée aléatoirement. Actuellement un épillet arrivé à maturation contient 3 graines dans Simpraise, on a ainsi créé l'ensemble des graines issues du modèle morphologique. On applique alors sur celle-ci une nouvelle sélection aléatoire pour retenir celles qui seront utilisées pour simuler la nouvelle génération.

On dispose les élues dans une matrice de croisement : c'est une matrice carrée dans laquelle les lignes comme les colonnes sont indicées par l'identifiant des plantes du couvert. Les lignes correspondent aux plantes qui ont été mère, tandis que les colonnes représentent celles qui ont été père. Il ne reste plus qu'à placer dans cette matrice les graines sélectionnées précédemment. Au départ la matrice est nulle, puis pour chaque graine sélectionnée on incrémente le coefficient correspondant au couple père-mère qui l'a formée.

Exemple pour un couvert de 4 individus sur lequel on sélectionne 4 graines parmi celles produites pour établir une nouvelle génération :
Les graines sont exprimées de la manière suivante : graine = (id_mere, id_pere)
Les graines sélectionnées sont (1,2), (4,1), (2,3) et (1,2).
Voici la matrice que l'on obtient :

Id_pere

$$\begin{pmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Id_mere

Cette matrice est à renseigner en entrée du modèle génétique, ainsi que le nombre d'individus à créer (ici 4), le modèle possédant les données génétiques des parents issues de l'exécution précédente, il ne reste plus qu'à l'exécuter de nouveau. Il génère alors les données génétiques pour chaque graine, ainsi que leur paramètre de croissance, qui via une conversion sera leur paramètre C pour la nouvelle simulation du modèle morphologique.

III) Autres ajouts spécifiques de la branche Simpraise

1) Tontes :

Il y a eu plusieurs autres modifications distinguant la branche Simpraise de la version actuelle de lgrass. L'une d'entre elles est l'ajout d'une routine permettant de fixer les dates de tontes en début de simulation en fonction de la fréquence de coupe. Le modèle lgrass itère sur des degrés jour et non des jours, le principe pour fixer les dates de tontes est donc de parcourir le fichier météo pour compter le nombre de degrés jour dans chaque jour, afin de compter le nombre de jours et d'y placer (régulièrement pour le moment) les dates de tontes. Les tontes s'effectuent sur 1 degré jour, mais décomposent celui-ci en 3 itérations, ce qui implique d'augmenter la longueur de dérivation de 3 itérations lors de chaque tonte pour conserver la durée de simulation initiale. Par exemple, si l'on place une tonte à 200°Cj sur une longueur de dérivation de 1000, la tonte va se décomposer en 200 -> 200.25, 200.25 -> 200.5, 200.5 -> 200.75 puis 200.75 -> 201. Il y a donc 3 itérations supplémentaires utilisées, on rajoute donc 3 itérations par tonte pour conserver les 1000 degrés jour à simuler.

2) Sauvegarde d'une simulation :

Il est désormais possible de sauvegarder l'intégralité des objets servant à construire le couvert et les plantes à l'issue d'une simulation. On appelle lstring l'objet stockant tous ces objets, constituant un lsystem. Un lsystem n'étant pas un objet python, la procédure pour récupérer les objets est légèrement plus complexe que l'emploi des fonctions dump et load du module pickle. Python ne pouvant lire directement la lstring, le choix a été de stocker tous les objets (exclusivement python) ainsi que leurs paramètres contenus dans la lstring dans des fichiers distincts en utilisant le dump. Tous les fichiers de paramètre sont numérotés dans leur ordre croissant d'apparition dans l'objet, et classés dans des répertoires par objet, eux-mêmes classés par ordre d'apparition dans la lstring. Il est alors possible de load chacun de ces objets avec l'ensemble de leurs paramètres dans le bon ordre afin de reconstituer telle quelle la lstring lors de la création de l'axiom (état initial du lsystem). Certaines variables locales de lgrass sont également essentielles à stocker, puis charger. Le fonctionnement du module est détaillé dans la documentation de Simpraise.

L'objectif principal de cet outil est de réduire le temps de simulation du modèle, dans les cas où l'on est amené à étudier le comportement des plantes adultes en faisant varier les paramètres d'entrée uniquement à partir d'un point de temps donné. Dans le cadre de la reproduction, cela pourrait également être utile pour modifier rapidement la matrice de croisement ou les techniques de production de graines. L'outil développé ici peut s'appliquer pour tous les lsystem dont l'ensemble des objets se place dans une lstring. D'autres opportunités suite à un avancement plus complet de la reproduction via lgrass restent à envisager.

IV) Pistes d'améliorations, points techniques :

1) Partie reproduction :

La partie d'implémentation de la reproduction se trouve uniquement dans le module param_reproduction_functions.py.

Tout d'abord, concernant les méthodes de calcul du nombre de graines dans le couvert :

- Pour la méthode de calcul des graines à partir du nombre d'épillets par talle, il est possible d'améliorer la calibration du modèle dans la production d'épillets. Celle-ci repose sur deux paramètres d'entrée qui sont des coefficients de production d'épillets en phase végétative et vernalisée/reproductive, l'ajustement de ceux-ci sera peut-être prochainement rectifié par Simon Rouet.
- En ce qui concerne la production d'un nombre de graines par épillet, celle-ci est actuellement fixée de manière empirique à 3. Une expérimentation pourrait consister à compter le nombre de graines produites par épillet sur du ray-grass au sein d'une population d'individus identifiés génotypiquement.

Dans l'implémentation, le programme renvoie une erreur lorsque le nombre de graines produites est inférieur à celui qui doit être produit au minimum (le nombre de candidats du modèle génétique). De même lorsqu'il n'y a qu'une seule plante dans le couvert, ce qui en principe ne se produit pas car dans ce cas le programme créé des copies de cette plante jusqu'à en avoir autant que le num_candidats (par choix, peut être modifié), et avertit l'utilisateur de cette action.

La sélection des individus mâles se fait totalement aléatoirement sur le couvert, hormis sur l'individu femelle correspondant. On pourrait rajouter un facteur de distance entre les plantes dans l'aléatoire, en attribuant un poids plus ou moins fort en fonction de la proximité des plantes par rapport à l'individu femelle. De même on pourrait prendre en compte le stade de développement de chaque plante, pour n'attribuer la parenté qu'aux plantes qui ont atteint leur de floraison.

Actuellement les paramètres morphologiques (excepté le paramètre de croissance) et le management génétique (num_candidats surtout) sont constants durant l'enchaînement des générations, ce qui pourrait être amélioré. Il faut également assurer la correspondance entre le num_candidats génétiques et le nombre de graines sélectionné à la fin du modèle morphologique, qui lui est fonction de nb_plantes morpho.

Enfin, ce qui serait sans doute l'avancée la plus intéressante à réaliser sur le court terme serait l'intégration des nouvelles générations au sein du même couvert que les précédentes, en produisant les graines dès l'épiaison. Pour cela, une première piste serait d'augmenter le maillage du couvert, puis d'augmenter la densité du couvert lors de chaque implantation d'un nouvel individu.

2) Implémentations diverses :

a) La sélection des plantes à simuler dans le modèle morphologique est contenue dans le fichier param_plantes.csv ; actuellement le programme simule l'intégralité des plantes présentes dans ce fichier. Il pourrait être intéressant d'ajouter un paramètre dans la fonction runlsystem du batch permettant de sélectionner uniquement les plantes souhaitées dans la liste complète.

b) La configuration des chemins d'accès aux fichiers/modules est fonctionnelle mais pourrait sans doute être mieux regroupée. Par exemple, les chemins d'accès nécessaires au fonctionnement du module de sauvegarde de lstring sont en dur au début du module.

c) Le script lgrass.lpy ne peut plus s'exécuter directement dans lpy car la gestion de certains paramètres et options d'entrée ont été déplacés dans le batch ou dans le plan de simulation.

d) Contrairement aux autres branches, la gestion du modèle de calcul de rayonnement caribu n'est plus directement présente dans lgrass, mais couplée extérieurement au modèle dans un script run_caribu_lgrass.py. Le fonctionnement reste identique aux versions précédentes. Cependant, il reste à pouvoir l'appeler depuis lgrass.lpy si besoin.

e) Le modèle génétique ne sort actuellement que la valeur de C, néanmoins la variable lgrass Premiercroiss semble également pouvoir être issue d'indicateurs génétiques, puisqu'elle influe également sur la croissance des plantes. A priori sa signification biologique la rend indépendante de la valeur de C mais c'est à vérifier. Selon sa valeur elle peut également faire remonter un bug lié à la variable NextAccroiss intervenant dans le module Feuille de lgrass (voir Issue sur la branche Simpraise de github), une calibration de ce paramètre pourrait donc être envisagée.

f) Les paramètres d'entrée des plantes ou de caribu se trouvent dans des fichiers csv. Il pourrait être pratique de les transformer en classes python afin de regrouper tout ce qui peut l'être dans les scripts et de limiter le nombre de fichiers d'entrée du modèle.

g) Il sera utile d'unifier les deux méthodes de coupe actuelles : la première étant d'ajuster les paramètres de coupe directement dans le fichier d'entrée des plantes liste_plantes.csv, et la deuxième étant l'emploi du script cuts.py générant les dates de tontes. L'unification permettrait d'éviter totalement les conflits d'utilisation.