

Étude documentaire et méthodologique

Groupe SBT-11

2 décembre 2015

1 Livrable

1.1 Implémentation du modèle LNAS sous Julia

On va travailler sur le Modèle LNAS, adapté à la modélisation du blé. Le client a déjà implémenté ce modèle en C++. Nous allons pour notre part l'implémenter sous Julia.

Ce modèle était d'abord utilisé pour modéliser la betterave, plus simple à modéliser que le blé. En effet, pour la betterave il suffit de modéliser ses racines (partie "utile") et ses feuilles. On va modéliser le blé, on rajoute la tige et les grains (partie "utile" du blé) pour maximiser le rendement des cultures en minimisant l'apport d'intrans. Nous allons donc utiliser la Plateforme Pygmalion pour implémenter ce modèle du blé, en nous basant sur un document fourni par le client.

Il y a environ 30 paramètres dans le modèle. Une fois le modèle implémenté, nous allons donc procéder à une analyse de paramètres et de leur influence. En effet, on ne peut pas mesurer tous les paramètres tout le temps (cela ayant un coût) Le but sera d'obtenir une liste *des paramètres pertinents*.

Le modèle LNAS est un modèle de Markov caché : n : temps (discret) X_n : variable au temps n Q_{grain} , Q_{leaf} ... : biomasse des grains, feuilles... f : fonction qui permet de passer du temps n au temps $n+1$ P : paramètres E_n : précipitations, température... au temps n

$$X_{n+1} = f(X_n, E_n, P, n) \quad (1)$$

$Q_{\text{prod}}(n)$ proportionnelle à $(1 - \exp(-\lambda \cdot \text{LAI}(n)))$ avec LAI la quantité lumineuse reçue par la plante

C'est un modèle de Markov caché car $P(X_{n+1}/X_0, \dots, X_n) = P(X_{n+1}/X_n)$: pas de mémoire du passé.

On déduit $Q_{\text{grain}}(n)$... grâce à X_n et la fonction d'allocation de biomasse.

2 Contexte du projet

2.1 Client du Projet

Pierre Carmier, le client du projet, étant un chercheur à Digiplante, une équipe à labo du MAS. Cette équipe travaillait notamment sur le modèle mathématique de croissance de plantes depuis 2002. Ils travaillent notamment sur le modèle LNAS qui a bien adapté à un mono type de plante.

2.2 Contexte

Le problème de ressources nous attire plus en plus d'attention depuis le 21ème siècle surtout en agriculture. Donc il nous pousse à optimiser la gestion pour l'agriculture à réduire le volume des ressources en intrants (eau, fertilisants, etc.) et récolter aussi bien de produits enfin d'améliorer l'efficacité avec des critères économiques et écologiques.

D'autre part, le développement de technologie (l'imagerie satellitaire, drones etc.) nous permet d'obtenir de plus en plus précises et nombreuses données. Avec des nombreuses données, nous pouvons le résoudre avec l'informatique en développer des modèles décisionnel, précis et pratiques. Tous nous rendent des résultats plus précis qu'avant.

Donc il sort le développement de 'smart agriculture' dont but est de préparer l'agriculture de demain qui nous permet de nourrir la plante écologiquement et économiquement. Et nous concentrons sur la modélisation de la croissance des plantes pour l'agriculture de précision comme une partie et nous attendons les résultats d'un programme capable de s'adapter à des nouveaux modèles de croissance et aussi la pertinence des paramètres pour des utilisations pratiques.

La plante apparaît comme un exemple de système complexe typique concernant multiple échelles. Il est non-linéaire et possède beaucoup de paramètres d'environnement et génétique et (regulation and retroaction loops ?en français ?). De plus, le processus biologique est complexe et a l'intersection entre eux (<http://digi plante.mas.ecp.fr>). Tous sont des contraintes que nous devons respecter. Mais heureusement, il existe déjà plusieurs modèles possibles pour certaines parts par exemple le LNAS et la plateforme de Julia.

Nous espérons de trouver des modèles plus flexibles qui adaptent mieux à la diversité de variables et conditions que l'on observe mais aussi avec moins d'incertitude par rapport aux données possibles. Avec des travaux nous pouvons bien les appliquer dans l'agriculture mais aussi de prédire la croissance de plante.

3 Organisation du groupe

Nous présenterons dans cette section l'organisation générale du groupe. On abordera d'abord les outils que nous avons utilisés pour partager notre travail le plus

efficacement possible. On présentera ensuite notre gestion de la bibliographie. On détaillera finalement l'organisation interne du groupe, plus précisément le système mis en place pour optimiser la communication ainsi que la manière dont les tâches ont été réparties pendant ce premier semestre.

Outils de travail collaboratif

Pour des raisons pratiques et esthétiques, nous avons décidé d'écrire nos rapports en L^AT_EX. Il s'agissait donc de trouver la meilleure façon de partager le code source et de pouvoir contrôler les changements apportés au document. Une première idée pourrait être d'utiliser ShareLaTeX qui propose une plate-forme de compilation en ligne ainsi qu'un système de gestion de versions assez simple à utiliser. Nous n'avons pas choisi cette solution notamment pour les raisons suivantes. L'utilisateur doit être connecté dès qu'il veut travailler sur le projet, le système de compilation est assez lent et l'utilisateur n'est pas libre d'utiliser son éditeur de texte ou son visualisateur de PDF favori.

Pour palier aux problèmes décrits ci-dessus, le logiciel `git` est une très bonne alternative. Il permet en effet à chaque membre du groupe de travailler localement et de ne faire un *push* sur le serveur que lorsqu'on juge que ce qu'on a fait est utile pour les autres. L'algorithme de fusion, *merge*, permet également de fusionner intelligemment les lignes d'un fichier qui ont été modifiées par plusieurs membres. Le dernier point à souligner est qu'il permet une gestion des branches, particulièrement pratique lorsqu'on veut développer une partie du projet sans risquer de créer des erreurs dans le programme principal.

Nous l'utilisons pour implémenter un programme en JULIA pour le client dont le code source est sur la plate-forme GitLab. Nous avons également un dossier sur GitHub¹ qui contient nos fichiers pour rédiger l'étude documentaire.

Gestion de la bibliographie

Pour la gestion de la bibliographie au sein du document, nous utilisons le package `biblatex`. Celui-ci permet d'écrire l'ensemble de nos références dans un document `.bib` sous la forme suivante.

```
@online{histoire_mod_plantes,
  title = {Une histoire de la modélisation des plantes},
  author = {Philippe de Reffye and Marc Jaeger
    and Paul-Henry Cournède},
  url = {https://interstices.info/jcms/c_38032/une-histoire-de-
    la-modelisation-des-plantes},
  year = {2009},
  month = "04",
}
```

1. <https://github.com/jdewasseige/projet-sbt11>

La mise en page est alors automatique en fonction des informations fournies et le rendu de l'exemple est présenté ci dessous.

- [3] Philippe de REFFYE, Marc JAEGER et Paul-Henry COURNÈDE. *Une histoire de la modélisation des plantes*. Avr. 2009. URL : https://interstices.info/jcms/c_38032/une-histoire-de-la-modelisation-des-plantes.

Cela parait à priori assez lourd d'écrire soi-même toutes les informations en suivant cette syntaxe mais il existe des logiciels comme Zotero qui font le travail à notre place. Les sources trouvées sur Google Scholar peuvent également être exportées aisément au format BibTex.

Organisation de l'équipe et partage de tâches

Il nous reste un dernier point à décrire, celui de la *communication* au sein du groupe. Nous utilisons Slack² qui est un logiciel de plus en plus utilisé pour les travaux de groupe ainsi que dans les start-ups. Il permet d'éviter de devoir alterner entre plusieurs applications comme les mails, DropBox et Twitter, puisqu'il permet d'être connecté à celles-ci au sein de l'application. On peut également créer plusieurs *channels* pour séparer la communication entre les différentes tâches. Par exemple dans ce projet nous avons les *channels* suivantes : **general**, **etude-documentaire**, **planning** et **dev_plate-forme**. On trouve aussi un système d'historique et de gestion de fichiers efficace.

2. <https://slack.com/>