Végéta - Temps et expression du végétal

Rapport de TER mi-parcours

Anaëlle Dambreville, Nabil Ouhi, Idriss Roudmane et Darren Samreth

M1 MIASHS

Encadrement par Sophie Lèbre et François-David Collin

21/12/2017

LAAB
Laboratoire Associatif
d'Art & de Botanique

I - Introduction

L'adaptation des plantes au changement climatique est un des enjeux majeurs de notre société actuelle. C'est pourquoi, de plus en plus d'études s'intéressent à comprendre la réponse des plantes aux variables environnementales. La mesure du flux de sève est reliée directement à l'état hydrique de la plante et permet d'appréhender facilement l'effet de la sécheresse sur la plante. Néanmoins, le flux de sève est une variable complexe à appréhender car elle varie en fonction de plusieurs facteurs environnementaux et propres à la plante (l'espèce de l'arbre par exemple). Au cours d'une même journée, le flux de sève augmente au début du jour puis diminue dans l'après-midi. En fonction des saisons, le flux de sève va aussi être plus ou moins élevé. Ces évolutions qui sont caractéristiques des séries temporelles couplées au nombre important de variables explicatives rendent complexe l'analyse des flux de sève.

Les objectifs de notre étude étaient d'appliquer à ces données une approche de type Big Data permettant de faire ressortir des comportements, des échelles de temps, des rythmes non triviaux, bref, de mettre en place des outils qui pourront servir de base à l'intuition scientifique. Cette étude a été en partie encadrée par le LAAB (Laboratoire Associatif d'Art et de Botanique).

Nous avons choisi une méthode de modélisation par régression linéaire multiple couplée à des outils web permettant au final une visualisation simple et interactive du flux de sève en fonction de plusieurs variables environnementales.

II - Données et méthodes utilisées

A - Les données

Nous avons récupéré notre jeu de données à partir du site : European Fluxes Database Cluster, qui est une grosse base de données qui héberge dans une seule infrastructure des mesures des flux entre les écosystèmes et l'atmosphère. Elle a été créée dans l'objectif de fournir des outils de traitement et de partage de données standard et de bonne qualité.

Les données sur lesquelles nous avons travaillé sont issues d'une expérimentation dans la forêt domaniale de Puéchabon, à proximité des gorges et de la vallée de l'Hérault, à 45 km de Montpellier (43.74139 (latitude), 3.595833 (longitude)). Le site est soumis à un climat méditerranéen typique comprenant une importante sécheresse estivale. La forêt, dominée par le chêne vert, est étudiée par les chercheurs du CNRS (CEFE).

Les données de flux de sève ont été mesurées à l'aide d'un capteur "thermocouple" qui calcule la vitesse du flux de sève dans un arbre entre deux électrodes (Figure 1). Ce capteur est constitué de deux aiguilles séparées de 10 cm environ. L'aiguille du haut est chauffée et la température mesurée par l'aiguille du bas traduit la vitesse du flux de sève. Cette mesure a été réalisée pour un chêne vert toutes les 30 minutes pendant un an (2010). En plus du flux de sève, 27 variables environnementales comme la température (TA) et l'humidité de l'air (RH) ont été mesurées (Tableau 1).

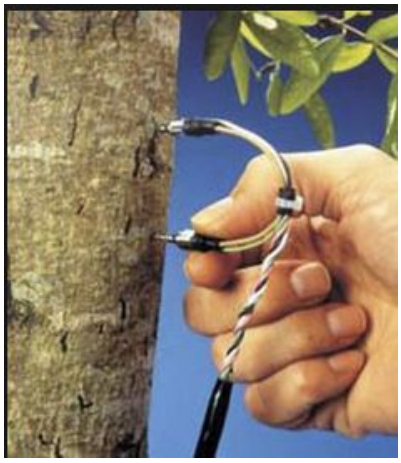


Figure 1 : Thermocouple

Le déficit de pression de vapeur (VPD, kPa) est une variable environnementale qui a un effet important sur la croissance des plantes. Nous avons calculé cette variable à partir de l'humidité relative et de la température de l'air à partir de la formule suivante :

$$VPD = ES - EA$$

avec $ES = 0.6108\, exp(\frac{17.17\, TA}{TA + 237.3})$ et $EA = \frac{RH\_ES}{100}$

où ***ES*** la pression de vapeur saturante, ***EA*** la pression de vapeur de l'air ambiant, ***TA*** la température de l'air (°C) et ***RH*** l'humidité relative de l'air (%).

Les données ont été nettoyées à l'aide de la bibliothèque Pandas de Python, en enlevant les lignes où il y avait uniquement des valeurs manquantes.

Ce jeu de données a été utilisé pour prédire le flux de sève d'une journée en fonction des variables environnementales et de l'heure (heure solaire). D'autre part, elles ont été utilisées pour calculer l'évolution moyenne du flux de sève observée sur une journée pour chaque mois où il y avait assez de mesures.

La prise en main et le nettoyage des données a été réalisé en à peu près 2 semaines. Nous avons eu à notre disposition d'autres jeux de données mais nous n'avons pas pu les exploiter pour le moment car ceux-ci ne comportaient soit pas de mesure du flux de sève, soit pas de variable explicative mesurée en même temps que le flux de sève.

Tableau 1 : Variables mesurées à Puechabon

Variable	Description	Unité
TIMESTAMP_START	Date de début	timestamp
TIMESTAMP_END	Date de fin	timestamp
DTime	Date sur 366 jours	
NETRAD	Radiation nette (incluant radiation solaire et infrarouge)	W m^{-2}
P	Précipitation	mm
PA	Pression atmosphérique	Kpa
PPFD_DIF	PPFD diffuse incidente (400-700 nm)	µmol m^{-2} s^{-1}
PPFD_IN	Densité de Flux Photon Photosynthétique (PPFD)	µmol m^{-2} s^{-1}
PPFD_OUT	PPFD réfléchi (400-700 nm)	µmol m^{-2} s^{-1}
RH	Humidité relative	%
SW_IN	Radiations des ondes courtes incidentes (0.3 to 4.5 micron)	W m^{-2}
SW_OUT	Radiations des ondes courtes sortantes (0.3 to 4.5 micron)	W m^{-2}
TA	Température de l'air	°C
TS	Température du sol	°C
WD	Direction du vent	Decimal degree
WS	Vitesse du vent	m s^{-1}
CO2	Concentration du CO2	ppm
FC	Flux CO2	µmol CO_2 m^{-2} s^{-1}
H	Flux de chaleur sensible	W m^{-2}
H2O	Eau	mmol H_2O mol^{-1}
LE	Flux de chaleur latente	W m^{-2}
SB	Stock de chaleur dans la biomasse	W m^{-2}
SC	Flux de Stockage de CO2	µmol CO_2 m^{-2} s^{-1}
SH	Flux de stockage de chaleur sensible	W m^{-2}
SLE	Flux de stockage de chaleur latente	W m^{-2}
TAU	Momentum flux	Kg m^{-1} s^{-2}

USTAR	Vitesse de frottement	m s^{-1}
ZL	Paramètre de stabilité	Sans dimension
PRI	Indice de réflexion photochimique	Sans dimension
G	Flux de chaleur du sol	W m^{-2}
SAP_FLOW	Vitesse flux de sève	mmol H_2O m^{-2} s^{-1}

B - La modélisation

La modélisation a été réalisée sur toutes les données de flux de sève regroupées sans tenir compte de la date ou de l'heure où la mesure a été faite. Au total, nous avons utilisé 9651 mesures de flux de sève.

1 – Étude des corrélations entre les variables

Nous avons utilisé une régression linéaire multiple afin de modéliser l'effet des différents régresseurs sur le flux de sève. Ce type de modélisation implique que la matrice des régresseurs doit être de plein rang, c'est à dire qu'aucune variable ne doit être une combinaison linéaire d'une autre variable. Dans un premier temps, nous avons donc étudié les corrélations entre les variables. Ces corrélations ont été calculées grâce au logiciel R (fonction cor()).

Nous nous sommes intéressés particulièrement aux fortes corrélations entre les variables ($|r| > 0.80$). Pour chaque groupe de variables fortement corrélées, nous avons gardé uniquement une variable (en se basant sur leur corrélation avec le flux de sève et sur leur pertinence biologique). Douze régresseurs ont été supprimés à cette étape : NETRAD, PPFD_DIF, PPFD_OUT, SW_IN, SW_OUT, H, LE, RH, TA, TAU, USTAR et G.

2 – Temps de latence de l'effet des régresseurs sur le flux de sève

Un intervalle de temps peut être observé entre un stimulus environnemental et la réponse physiologique de la plante. C'est ce qu'on appelle un temps de latence. Pour les 16 régresseurs restants nous avons testé plusieurs temps de latence allant de 0 à 3 heures (par pas de temps de 30 minutes) afin de sélectionner le temps de latence optimal pour chaque régresseur.

Pour chaque temps de latence (6 au total), nous avons sélectionné les régresseurs ayant un effet significatif sur le flux de sève en se basant sur le BIC (Bayesian Information Criteria) (Logiciel R, fonction regsubsets()). Quelque soit le temps de latence, 11 régresseurs ont été sélectionnés : PPFD_IN, TS, WD, CO2, FC, H2O, SB, SH, SLE, ZL et le VPD.

Pour ces variables, nous avons ensuite choisi le temps de latence optimal maximisant l'effet du régresseur sur le flux de sève. Un effet instantané a été observé pour TS, CO2, SB et le VPD (pas de temps de latence). Un délai de 1 heure a été observé pour les variables suivantes : PPFD_IN et FC. Un délai de 2 heures a été observé pour WD. Un délai maximal de 3 heures a été observé pour H2O, SH, SLE et ZL.

3 – Modèle choisi

Le modèle suivant a été calculé à partir des régresseurs sélectionnées précédemment (logiciel R, lm()):

$$\text{Sap flow} = 0.5 + 0.0034\ PPFD + 0.046\ TS - 6e\text{-}04\ WD - 0.0036\ CO2 - 0.046\ FC + 0.026\ H2O - 0.02\ SB + 0.0089\ SH - 4e\text{-}04\ SLE - 0.012\ ZL + 1.51\ VPD$$

4 – Prédictions du flux de sève

Nous n'avons pas utilisé les dates et les heures dans le modèle mais les variables sélectionnées permettent de modéliser intrinsèquement l'effet des saisons (TS, H2O, VPD) et des heures de la journée (grâce au PPFD_IN par exemple) sur le flux de sève.

L'évolution du flux de sève sur une journée « moyenne » a été prédite en calculant la moyenne de chaque régresseur (quelque soit la saison) pour 24 heures avec un pas de temps de 30 min. Une visualisation de l'effet de chaque régresseur est ensuite rendue possible en multipliant la valeur du coefficient associé à chaque variable par différentes valeurs (0 : pas d'effet de la variable, 0.5 : effet divisé par 2, 2 : effet double de la variable).

Les différentes étapes de modélisation ont été réalisées en 3 semaines environ.

C - La visualisation

Notre objectif principal était de proposer une représentation visuelle de la variation du flux de sève. Il a fallu trouver un outil permettant une visualisation efficace, tout en étant interactif. De plus, nous voulions pouvoir utiliser des données stockées sur un fichier externe. En effet, nos données sont stockées dans des fichiers CSV (mesures, coefficients de régression).

Nous avons décidé de présenter notre travail sous la forme d'un service web. Chaque onglet permet de présenter un type de visualisation différent. Les graphiques doivent être interactifs, car l'objectif est de pouvoir faire varier les variables et de visualiser le résultat de ces variations en temps réel sur les graphiques.

1 – Premier outil testé : Python

Python est un langage de programmation flexible qui offre de nombreuses possibilités. L'avantage est que son utilisation aurait permis de développer plusieurs outils dans le même environnement : traitement et analyses des données avec Pandas, modélisation de la régression grâce à Scikit-learn ou Statsmodels, mis en place d'un serveur WEB avec à Flask et visualisation grâce à Dash.

Ces outils auraient facilement pu être connectés entre eux. Ainsi, les données traitées grâce à Pandas dans un premier temps auraient pu générer un modèle sur Scikit-learn qui lui-même aurait été utilisé par Dash pour la visualisation.

Toutefois, nous avons finalement penché en faveur d'une deuxième méthode de visualisation qui s'inscrivait directement dans notre programme du semestre, et qui nécessitait moins de prérequis en programmation Python. Celle d'une application Web. Ce type d'application a pour avantage qu'aucune installation ne soit nécessaire, si ce n'est celle d'un navigateur Web. Il a donc été nécessaire de créer une interface web facile d'utilisation.

2 – L'application WEB

a - Bootstrap , le framework

Nous avons utilisé Bootstrap comme framework (qui est une librairie pour le front-end), qui nous permet d'avoir une application à la fois simple d'utilisation et à la fois organisée sur nos pages html. Bootstrap inclus des fichiers ".css" ce qui permet d'avoir un style plus épuré. Bootstrap permet de créer entre autres une barre de navigation fixe qui prend souvent trop de place sur un petit écran. Il faut donc parfois la cacher et la montrer uniquement lorsqu'elle est nécessaire. Un menu déroulant s'affiche lorsque l'utilisateur clique sur le bouton de la barre de navigation (option pour les petits écrans ex: smartphone). La fonctionnalité principale pour laquelle nous avons retenu bootstrap comme framework est la fonctionnalité de grille et d'organisation des éléments d'une page permettant de disposer plus aisément les textes, les graphiques et les sliders.

b - Les fonctions Javascript

Différents éléments composent les outils de visualisations :

- Les sliders
- Les courbes (associées aux modèles et aux fonctions de calculs)
- Les axes de coordonnées et leur echelles

Pour construire et assembler ces différentes composantes, nous avons utilisé des fonctions javascript et la librairie D3.js. Pour le javascript c'est un ensemble de boucles et fonctions qui parcourt le jeu de données et qui les croise dans la fonction du modèle choisi. Nous récupérons ensuite ces données qui sont des coordonnées, puis nous les affichons avec D3 et des balises svg sur la page web.

c - La librairie D3.js

D3.js est une bibliothèque Javascript de représentation graphique interactive. C'est aujourd'hui un des outils les plus populaires de visualisation. De plus, il est relativement accessible : étant codé en Javascript, son utilisation nécessite juste de savoir comment utiliser la librairie et des connaissances en programmation Web (SVG-Javascript-CSS -HTML).

3 – Visualisations choisies pour représenter les données

Nous avons décidé de mettre en place deux types de représentation :

- Une représentation du modèle général, qui permet à l'aide de curseurs de faire varier des l'effet des régresseurs sur le flux de sève
- Une représentation du modèle pour chaque mois, qui présente l'évolution moyenne sur 24 heures observée pour plusieurs mois (de mai à novembre)

Ces deux représentations n'ont pas été faites par les mêmes personnes. Ainsi, même si les outils utilisés restent les mêmes, le déroulement des deux scripts ne sont pas identiques. Cette différence est effective dès l'importation du CSV.

En effet, dans la première représentation, les fichiers CSV qui contiennent les coefficients du modèle de régression et les valeurs des régresseurs sont importés par le biais d'objet ***XMLHttpRequest***. Le texte récupéré est ensuite mis en forme pour qu'il soit utilisable. Les valeurs sont ensuite stockées dans deux dictionnaires. Ces deux dictionnaires sont ensuite multipliés entre eux, et les résultats des produits sont additionnés pour chaque heure, afin d'avoir une valeur de flux de sève ponctuelle prédite. C'est comme cela que notre courbe initiale est générée. Ensuite, à chaque variation du curseur, la courbe est recalculée avec la modification du coefficient du régresseur correspondant.

Dans la deuxième représentation, le fichier CSV (qui contient les valeurs de flux de sève moyens par heure et mois) est importé directement par la fonction ***d3.csv()***. Les différentes lignes du CSV correspondent aux différents mois. Les valeurs correspondantes au flux de sève et à l'heure sont stockées dans deux tableaux. La courbe est ensuite tracée en utilisant les deux tableaux de données avec d3.js. Les tableaux de données sont mis à jour à chaque changement de mois, soit à chaque variation du curseur.

Il reste des points qui sont encore à améliorer si nous voulons rendre notre outil adaptable à tous les fichiers CSV en respectant notre méthode d'importation. Par exemple, il manque une méthode qui permettrait de calculer la valeur maximale de flux de sève dans la première visualisation. En effet, actuellement la valeur maximale de

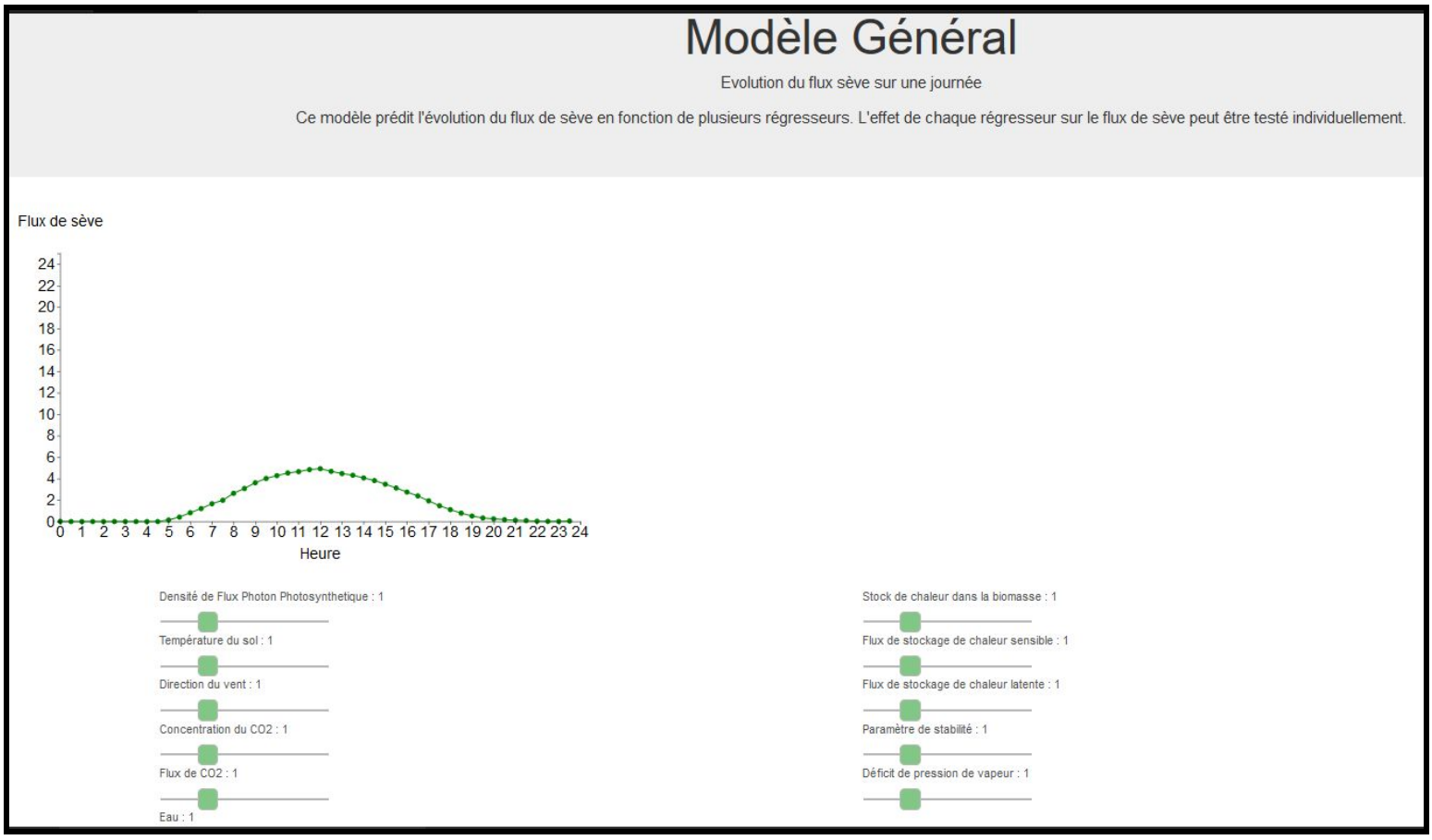
l'ordonnée est fixée dans le script, car il faut envisager les cas les plus extrêmes en fonction des coefficients des régresseurs. La deuxième représentation est aussi ancrée sur le jeu de données actuel qui comprend certains mois inexploitables (car trop de données manquantes).

Ainsi, la visualisation est une partie importante de notre travail, qui nous a pris environ trois semaines de travail, et qui est encore perfectible.

III - Résultats obtenus jusqu'à présent

Notre résultat principal est une application web. Le projet global y est présenté sur le premier onglet de la page d'accueil (Figure 2). Un onglet "modèle général" permet de tester l'effet de 11 régresseurs sur la courbe de flux de sève au cours d'une journée (Figure 3). Cette visualisation est basée sur une prédiction dynamique du flux de sève grâce au modèle décrit précédemment. Un onglet "modèle mensuel" présente la courbe moyenne du flux de sève observée en fonction des mois choisis par l'utilisateur (Figure 3). Enfin, un onglet "info" détaille les différentes variables utilisées et les délais de latence observés pour certains régresseurs.

Figure 2 : Page d'accueil



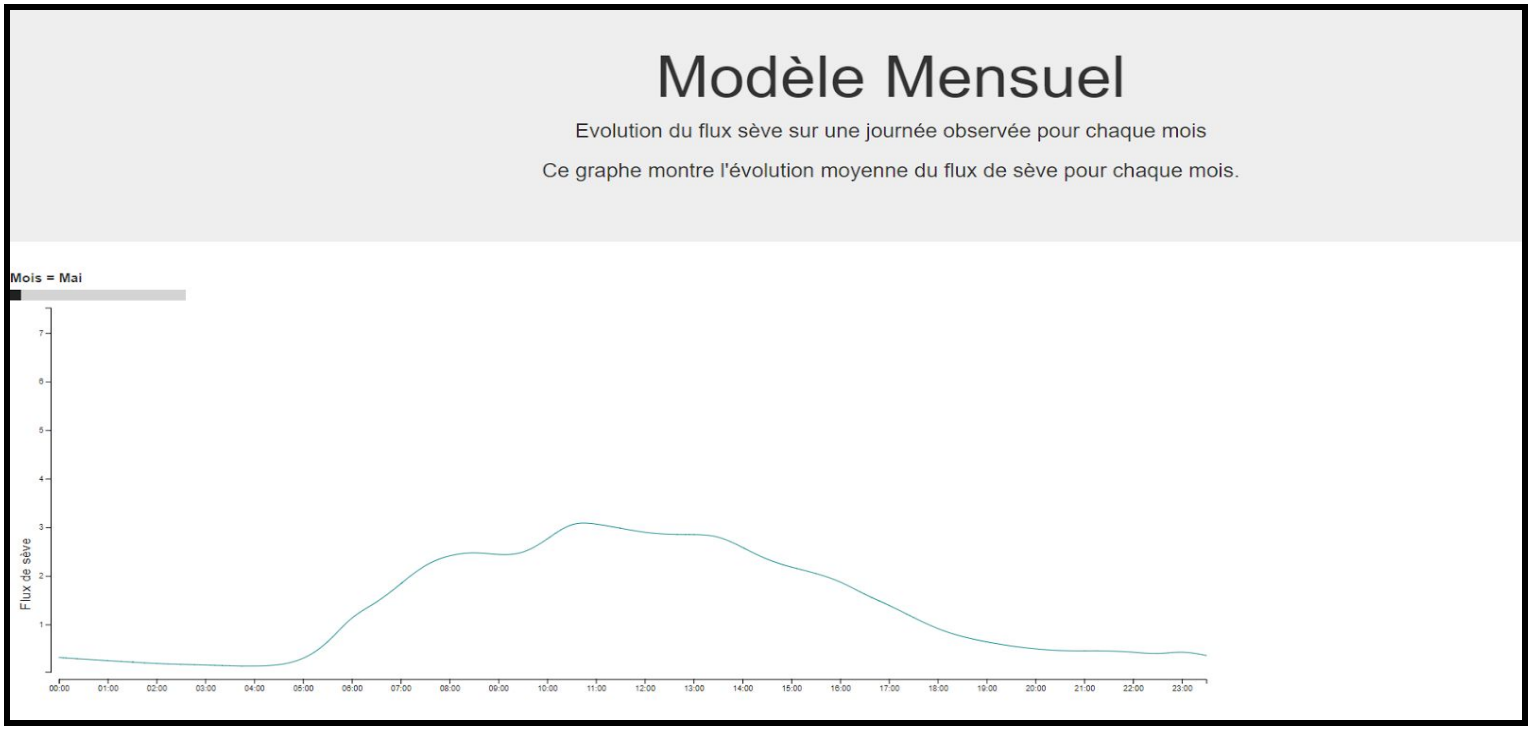


Figure 3 : Pages du modèle général (haut) et de la visualisation mensuelle (bas)

IV - Conclusion et perspectives

Au long de ce projet nous avons dans premier temps été amenés à explorer et à analyser des données brutes afin de concevoir un modèle permettant de représenter et d'analyser au mieux le flux de sève. Dans un second temps, nous avons conçu un réel outil d'analyse exploratoire en implémentant le modèle calculé pour que cette application soit consultable par toute personne voulant étudier le comportement du flux de sève selon un ensemble de variables configurables.

Ce TER a déjà été l'occasion pour nous d'appliquer dans un cadre plus réaliste et avec un jeu de données issu du monde réel, les connaissances acquises depuis le début de notre formation. En effet, il nous a permis d'aborder plusieurs disciplines (telles que la régression et sémiologie graphique) et d'explorer des outils de visualisation comme D3js plus en profondeur.

En perspective, il y a un travail d'homogénéisation des deux représentations graphiques à effectuer, que ce soit pour la maintenance du code ou pour l'esthétique des graphiques.

Si nous ne trouvons pas plus de données pour enrichir notre modèle ou pour proposer de nouveaux modèles basés sur différentes espèces d'arbres ou d'autres variables environnementales, un ajout en terme de représentation pourrait être une visualisation de la variation des variables environnementales mesurées à Puéchabon en fonction du temps.

Outre ceci, une nouvelle fonctionnalité pourrait être de permettre aux utilisateurs d'implémenter leurs propres données, et ainsi de recalculer un nouveau modèle ou de prédire le flux de sève en fonction de leurs régresseurs et de notre modèle.