Créer des modules avec Vsoil

Tutoriel 1 : root water uptake

Éric Aivayan, Nicolas Moitrier, Cédric Nouguier

2022-05-16

Table des matières

Table des figures

Listings

Licence: CC BY NC SA

Cet ensemble de tutoriels a pour objectif de montrer par des exemples concrets la création de modules. Les différents modules présentés permettent d'aborder les possibles difficultés de réalisation de manière graduelle. Ainsi nous allons partir d'un module indépendant du temps et ne nécessitant pas de données représentées sous forme de tableaux pour arriver à un module dépendant du temps avec tous les types de données possibles.

Pour pouvoir aborder ces tutoriels, vous devez savoir :

— créer un processus,
— modifier un processus,
— créer une entrée/sortie,
— modifier une entrée/sortie.

Le codage des modules étant réalisé en Fortran, vous devez avoir quelques notions de programmation dans ce langage.

Sur toutes les captures d'écran, les textes encadrés en rouge sont présents à titre indicatif. Ils ne s'affichent pas lors de l'utilisation de la plateforme.

1 Présentation scientifique.

Le module que nous allons créer simule l'absorption racinaire en eau.

Par hypothèse le besoin en eau de la plante est répercuté de manière homogène sur la longueur des racines de la plante. C'est à dire que chaque centimètre de racine a une demande d'absorption d'eau identique.

Cependant, les racines ne peuvent pas absorber n'importe quelle quantité d'eau. En effet, l'eau disponible dépend du potentiel matriciel de l'eau du sol. Cela est pris en compte par la fonction alpha qui définit trois régions dans la gamme de potentiels matriciels qui peuvent exister en un point du profil de sol.

$$\alpha(h) = \begin{cases} 0 & , \quad h \geq h_1 \\ \frac{h-h_1}{h_2-h_1} & , \quad h \in]h_2; h_1[\\ 1 & , \quad h \in [h_3; h_2] \\ \frac{h-h_4}{h_3-h_4} & , \quad h \in]h_4, h_3[\\ 0 & , \quad h \leq h_4 \end{cases} \tag{1}$$

Par la suite, la simulation calculera le taux d'absorption d'eau de chaque élément de longueur racinaire et, par intégration de ce taux sur le profil de sol, l'absorption réelle des racines et donc le flux de transpiration réel de la plante.

2 Ce que vous allez découvrir.

Dans ce premier tutoriel nous allons passer en revue toutes les étapes permettant la création d'un module depuis la sélection du processus à la base du module jusqu'à l'écriture du code traduisant le concept scientifique.

Ainsi vous allez, au travers des différents onglet :

— Sélectionner un processus auquel le module va être rattaché.
— Donner les caractéristiques minimales d'un module.
— Sélectionner les entrées et les sorties utilisées dans le module.
— Créer des paramètres.
— Renommer les entrées et les sorties pour faciliter le travail de programmation.

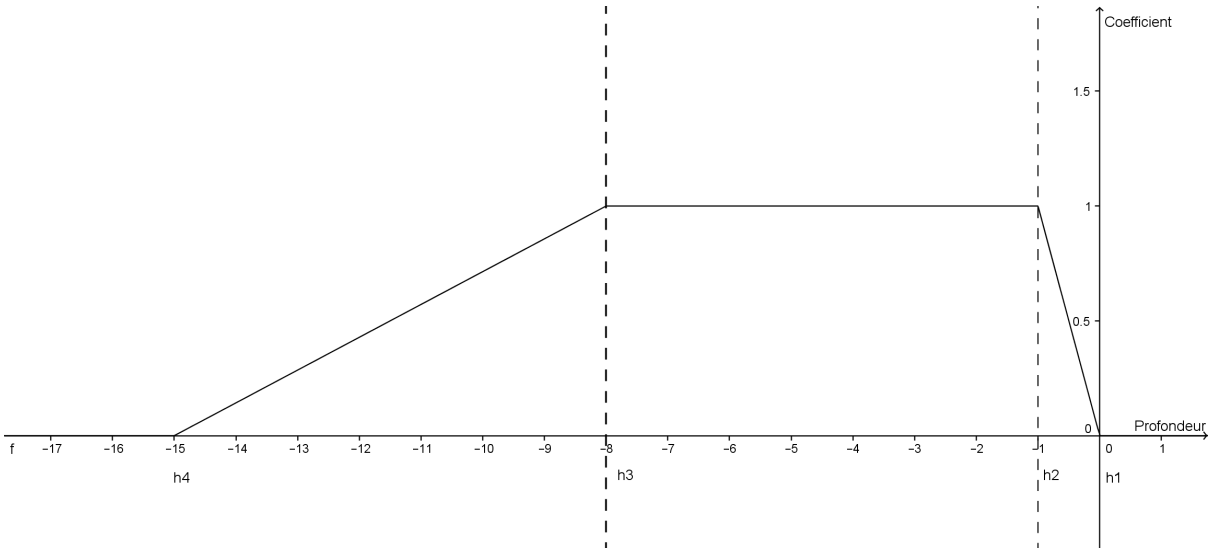


FIGURE 1 – Représentation graphique de la fonction $\alpha(h)$ en fonction des quatre paramètres (potentiels matriciels).

— Initialiser les valeurs des sorties.
— Déclarer des variables locales pour le programme principal de la simulation.
— Écrire le code source du programme principal de la simulation.
— Écrire des fonctions utilitaires pour le programme principal.
— Travailler sur un profil de sol.

Attention !

Dans la suite de ce tutoriel nous présentons des captures d'écran du logiciel. Ces images ont pour but d'illustrer le propos. Elles peuvent être légèrement différentes de ce que vous verrez sur votre écran, en particulier, sur la disposition des informations.

D'une manière générale, vous devez retrouver les principales informations et champs même si ils ne sont plus exactement au même emplacement sur l'écran.

3 En avant !

Nous allons dérouler l'élaboration du module en passant d'un onglet à l'autre. Le passage de l'un à l'autre se fait généralement en cliquant sur le bouton « suivant » représenté par une flèche blanche sur fond vert situé en haut à gauche de la fenêtre.

Lancez le composant de plateforme « vsoil_modules » puis sélectionnez « New Module ».

3.1 Choix du processus simulé par le module

L'onglet *process* sert à choisir le processus que le module va simuler.

Sélectionnez le processus que le module va simuler dans la liste de gauche. Dans notre cas nous allons simuler le processus *root water uptake*.

Cliquez sur suivant pour poursuivre.

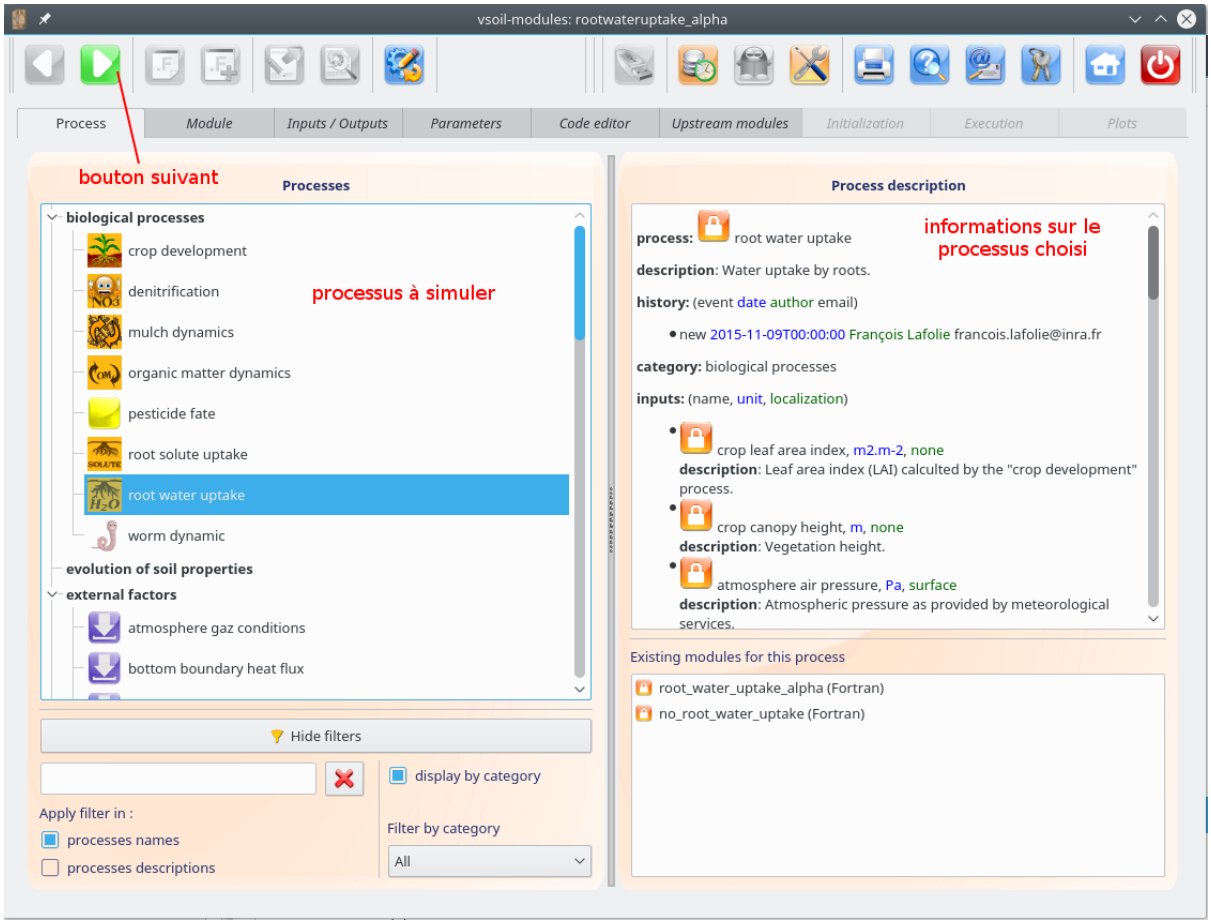


FIGURE 2 – Capture d'écran du composant de plateforme vsoil-module sur l'onglet de choix d'un processus.

3.2 Caractérisation du module

L'onglet *module* sert à renseigner les informations minimales du module. La partie gauche du panneau récapitule l'information concernant le processus sélectionné.

Saisissez dans le panneau de droite les quatre champs obligatoires :

Le nom : le nom du module en caractères minuscules sans espace et sans signe moins. Ici nous inscrirons *rootwateruptake_alpha_basic*.

L'auteur : le nom du rédacteur du module. Par défaut, votre nom devrait apparaître sinon saisissez en un.

La description : ce que réalise le module d'un point de vue scientifique.

La dimension : comment est représenté le profil de sol.

Dans ce panneau, sélectionnez dans le menu déroulant « Spatial dimension » la valeur 1. Ici on crée un module qui simule un profil de sol correspondant à une ligne verticale.

Appuyez de nouveau sur le bouton « suivant » pour aller à l'onglet suivant. Si vous avez oublié de remplir un champs obligatoire, le logiciel vous l'indique alors.

3.3 Sélection des entrées/sorties pour le module

L'onglet *inputs/outputs* sert au choix des entrées/sorties pertinentes dans la construction du module. En effet un module ne simule pas forcément un processus avec l'ensemble

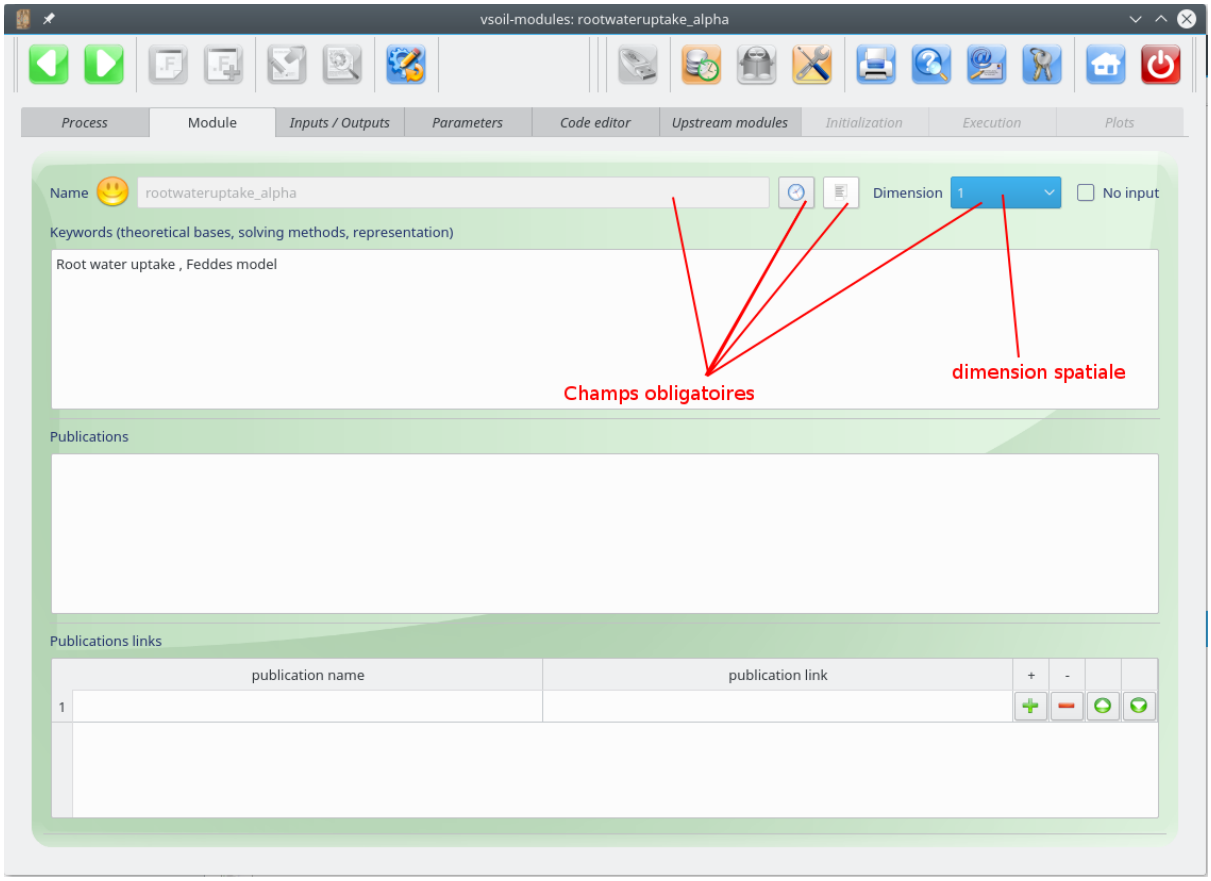


FIGURE 3 – Capture d'écran du composant de plateforme vsoil-module sur l'onglet de renseignement des informations de bases d'un module.

de ses entrées et sorties. Les choix effectués dépendent du modèle théorique que vous allez implanter. Dans notre cas, nous allons simuler la vitesse d'absorption d'eau racinaire et la perte d'eau (transpiration) de la culture en fonction de la demande en eau de la culture (transpiration), de la distribution du système racinaire sur le profil, de la longueur des racines et de la disponibilité en eau du sol (potentiel matriciel).

Par conséquence sur les dix-neuf entrées nous allons en sélectionner quatre :

— crop maximum transpiration volumetric flux density
— crop root length density
— crop root length
— soil water matrix potential

Sur les quatre sorties nous allons en sélectionner deux :

— crop transpiration volumetric flux density
— root water volumetric uptake rate

Cette sélection se fait en cochant la case devant le nom de l'entrée ou sortie.

Dans cette sélection, nous avons choisi des entrées et des sorties qui, pour certaines, sont définies sur le profil de sol. Cela signifie que lorsque le profil de sol aura été maillé, ces entrées et sorties pourront avoir des valeurs différentes pour chacun de ces points. Nous verrons cela au paragraphe 3.5.2.

Cliquez sur suivant pour poursuivre. Une boite de dialogue apparaît alors.

Le logiciel avertit que, lors de l'exécution, des valeurs initiales ne seront pas demandées

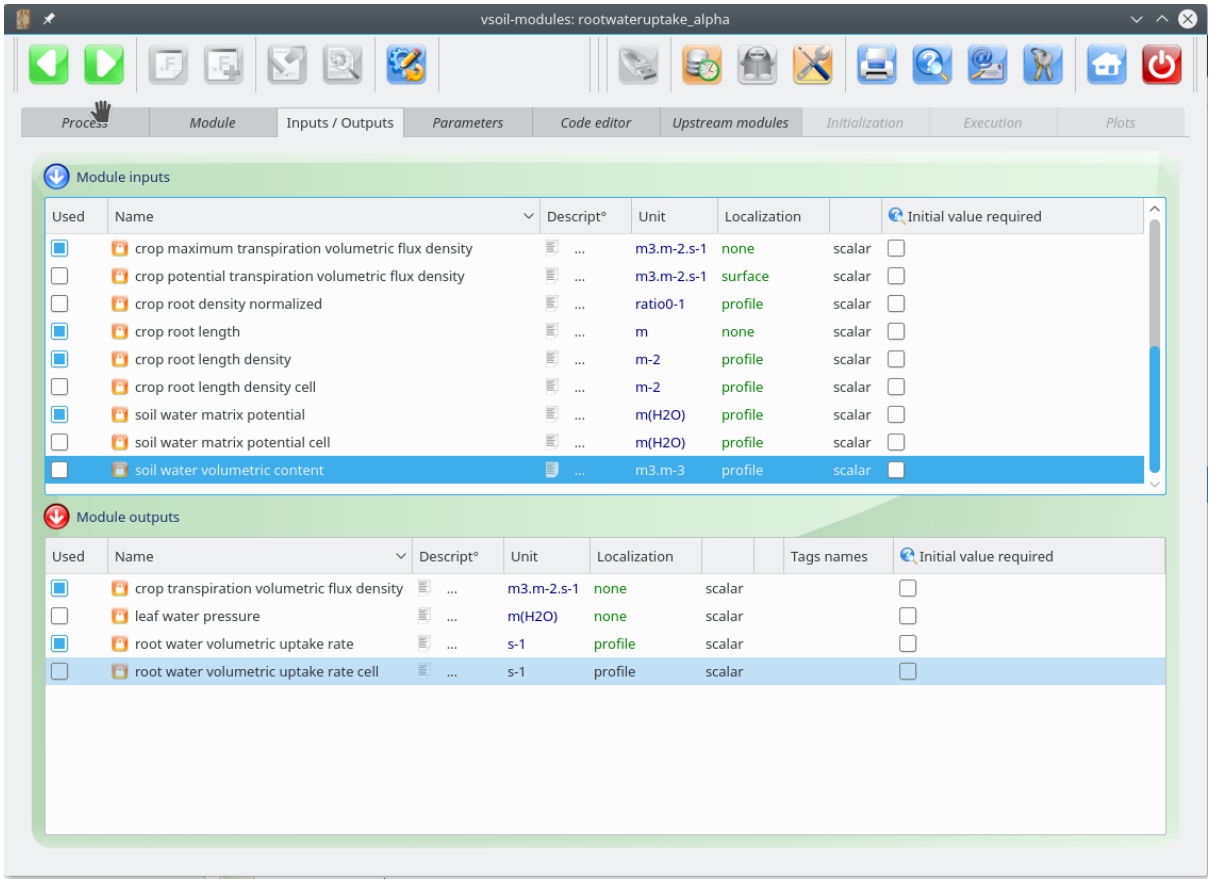


FIGURE 4 – Capture d'écran du composant de plateforme vsoil-module sur l'onglet de sélection des entrées et des sorties utilisées dans la mécanique du module décrit.

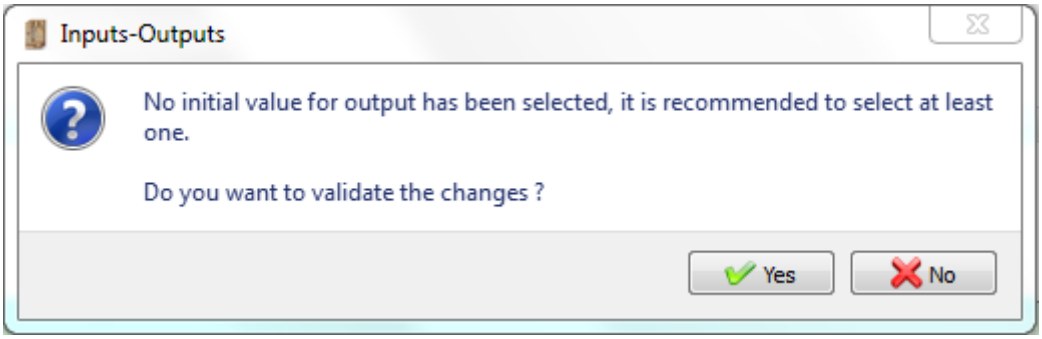


FIGURE 5 – Capture d'écran du composant de plateforme vsoil-module affichant un message d'avertissement concernant l'initialisation des valeurs des sorties.

et donc pas fournies pour les variables de sortie du module. Dans le cas que nous traitons, nous n'avons effectivement pas besoin de donner des valeurs initiales aux deux variables de sortie.

Nous verrons l'initialisation au paragraphe Initialization section page 13.

Répondons « yes » à la boite de dialogue et passons à l'onglet suivant.

3.4 Définir des paramètres pour le module

Dans cet onglet nous allons définir les principaux paramètres de notre module. Ces paramètres se retrouveront dans l'onglet « Code editor ». La plateforme les initialisera et vérifiera automatiquement les valeurs de ces paramètres selon les préférences que vous avez entrées. Dans notre cas, les paramètres seront de type simple (scalaire) et ne dépendront pas des couches de sol (donc la case « LD » pour layer dependent ne sera pas cochée).

Dans cet exemple nous avons besoin de quatre paramètres h_1, h_2, h_3 et h_4 qui correspondront aux valeurs des potentiels matriciels pour lesquelles la fonction alpha change de comportement. Ces paramètres seront des nombres réels décimaux dont l'unité est le mètre. On définit la valeur maximale et la valeur par défaut de chacun de ces paramètres. La description doit être rentrée dans la boite de dialogue qui apparaît en appuyant sur le bouton après le nom.

On ajoute des lignes en appuyant sur le bouton « plus » en fin de ligne.

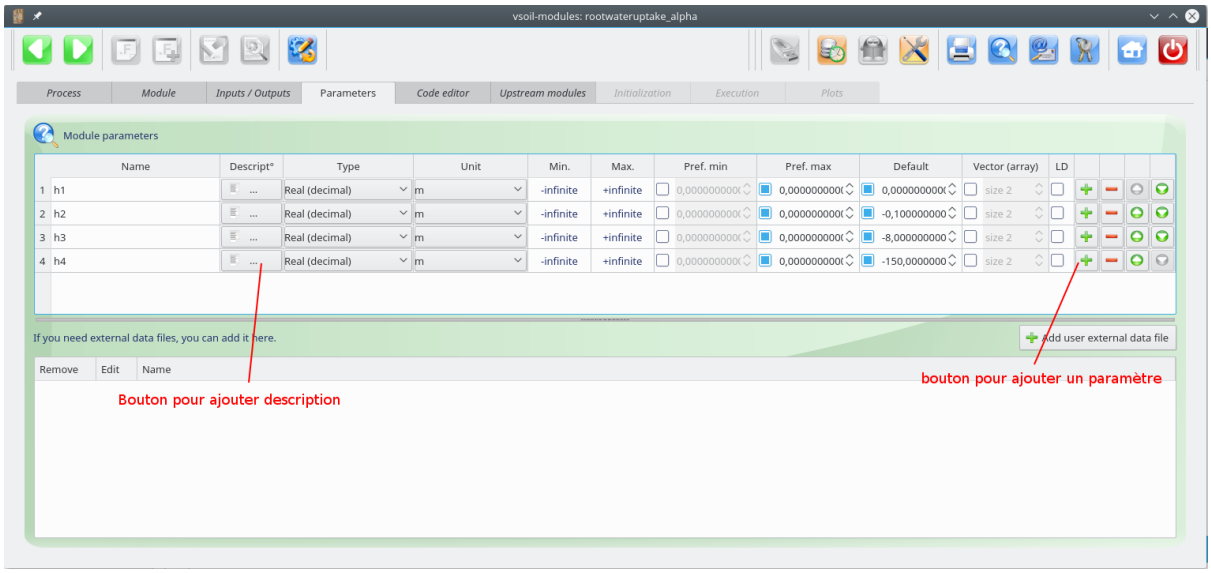


FIGURE 6 – Capture d'écran du composant de plateforme vsoil-module sur l'onglet de définition des paramètres du module.

Le tableau ci-dessous rassemble toutes les informations sur les paramètres à saisir. Les cases vides correspondent à un champ non modifié.

Nom	Description	Type	Unit	Pref.min	Pref.max	default	Vector(array)	LD
h_1	First matrix potential threshold in the definition of the alpha function. Above this threshold, the alpha function equals zero	Real(decimal)	m(H2O)		0	0		
h_2	Second matrix potential threshold for the definition of the alpha function. At this value of the soil water potential the alpha function value is 1. Alpha varies linearly between thresholds h1 and h2.	Real(decimal)	m(H2O)		0	-0.1		
h_3	Third matrix potential threshold of the alpha function. From h2 to h3, alpha value is 1.	Real(decimal)	m(H2O)		0	-8.0		
h_4	Below this matrix potential threshold, alpha value is 0. Between h3 and h4, alpha varies linearly.	Real(decimal)	m(H2O)		0	-150.0		

Passez à l'onglet suivant.

3.5 Entrer le code informatique du module

Nous arrivons au cœur de la création du code informatique du module.

Avant de pouvoir accéder à l'onglet, le logiciel nous demande le langage de programmation que nous allons utiliser pour créer notre module. Dans le cas présent le programme sera en *fortran*. Il est aussi possible de choisir le *C++*.

Cliquez sur « OK » pour poursuivre.

Pour faciliter l'écriture du code, les entrées/sorties vont être renommées. Notez que le fortran limite la longueur des variables : il peut être judicieux de raccourcir les noms pour éviter de futures erreurs.

Pour cela sélectionnez l'entrée/sortie à renommer et cliquez sur le bouton « rename ». Une boite de dialogue vous invite alors à entrer le nouveau nom.

Renommez les entrées/sorties comme illustré sur la capture d'écran. Le bouton de renommage n'est accessible que dans certaines section d'édition de code. Sélectionner la section « compute section » de l'éditeur de code pour renommer à la fois les entrées et les sorties.

La dernière étape est la rédaction du code du programme de simulation. Cette rédac-

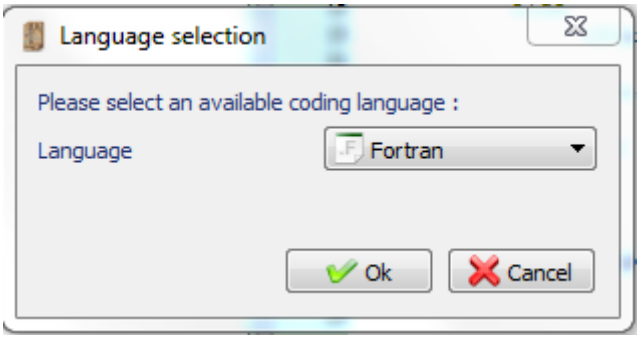


FIGURE 7 – Capture d'écran du composant de plateforme vsoil-module sur la boite de dialogue de sélection du langage de programmation du module.

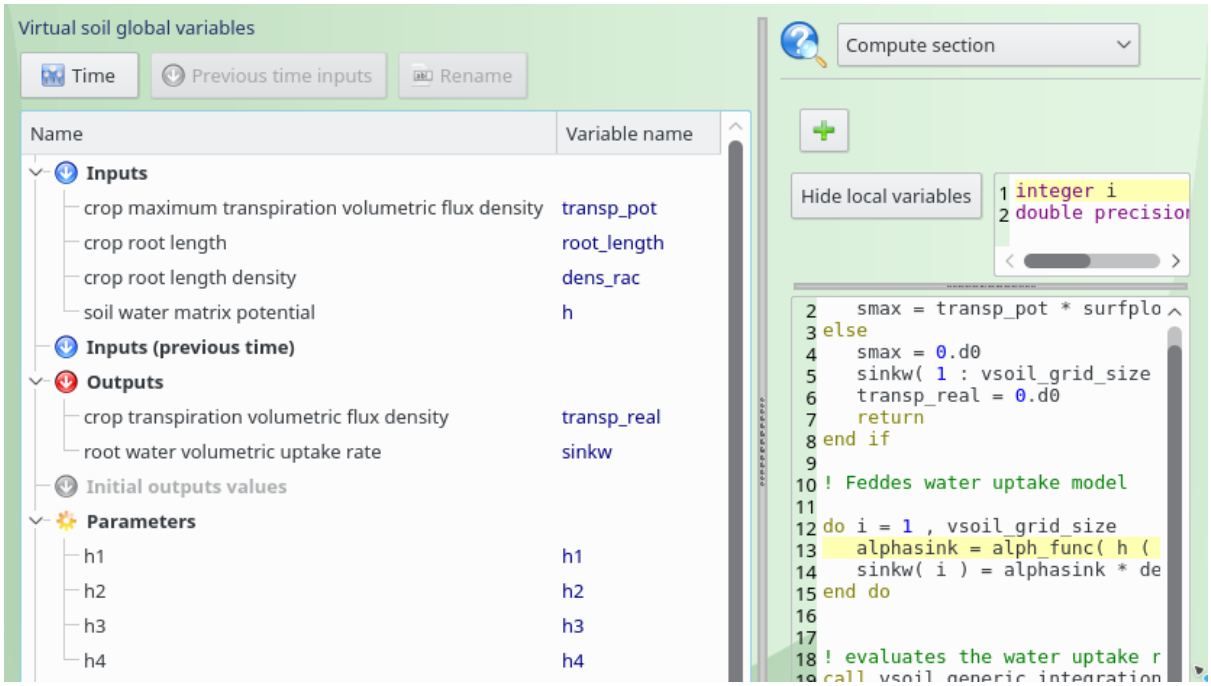
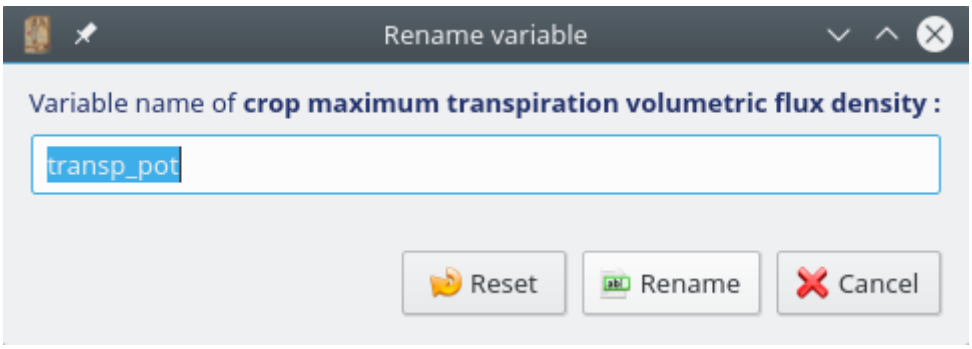


FIGURE 8 – Capture d'écran du composant de plateforme vsoil-module sur l'onglet de l'éditeur de code. Il n'est représenté que la partie haute de la colonne de gauche concernant les entrées, les sorties et les paramètres du module.



tion se fait dans le panneau de droite de cet onglet. Pour nous accompagner dans cette rédaction, plusieurs parties ont été identifiées. Nous allons nous concentrer principalement sur trois d'entre elles, les autres seront vues ultérieurement.

3.5.1 Initialization section

Dans cette section nous allons donner une valeur initiale à toutes les variables de sortie du module. Nous nous rappelons que le logiciel nous avait averti qu'il n'y avait pas de valeur initiale pour les sorties (voir le paragraphe 3.3). Nous allons donc les initialiser ici.

```
transp_real = 0.d0
sinkw( 1 : vsoil_grid_size ) = 0.d0
```

Listing 1 – Initialisation des valeurs des sorties.

Cette initialisation est nécessaire car le module construit pourra être couplé à un autre module. Notre module pourra fournir en entrées ses variables de sorties. Or, par construction, les entrées sont considérées comme fournissant des données valables.

Nous remarquons aussi, que la variable sinkw qui est un tableau, est initialisée simplement en « fortran ».

FIGURE 9 – Capture d'écran du composant de plateforme vsoil-module sur l'onglet de l'éditeur de code. Il est représenté la partie de droite lorsque la section d'initialisation est sélectionnée.

3.5.2 Compute section

Dans cette partie nous allons écrire le programme de simulation. Dans un premier temps il faut calculer la quantité maximum d'eau qui est demandée par unité de longueur de racine, puis calculer la quantité réellement absorbée et enfin intégrer cette quantité sur l'ensemble du profil de sol pour obtenir la quantité réelle absorbé par les racines et donc transpirée.

Nous avons besoin de déclarer des variables locales que nous utiliserons dans cette section. On clique sur le bouton « Declare local variables » et entrons le type et le nom des variables que l'on souhaite utiliser. Ici nous écrirons :

```
integer( kind = C_SIZE_T ) :: i
double precision alphasink,smax
```

Listing 2 – Déclaration des variables locales nécessaires à la « Compute section »

En dessous nous entrons la mécanique du module. La variable *surfplot* est une constante implantée dans la plateforme qui correspond à une surface de référence et que l'on trouve dans le panneau de gauche dans la section *Universal global constants*. De même *vsoil_grid_size* correspond au nombre de points de calcul dans le profil de sol.

Remarque : Une condition est ajoutée au début du code. Elle détecte le cas où il n'y a pas de racines, pour éviter une erreur dûe à une division par zéro. Dans ce cas, la variable *sinkw* est mise à 0 à tous les points du profil, ainsi que la variable *transp_real*. Les calculs dans cette section se terminent immédiatement après.

La première partie est le calcul de la quantité maximum que doit absorber une unité de longueur de racine :

```
if ( root_length <= 0.d0 ) then
    sinkw( 1 : vsoil_grid_size ) = 0.d0
    transp_real = 0.d0
    return
end if
```

Listing 3 – Initialisation de la quantité maximum que doit absorber les racines.

La seconde partie est le calcul des quantités réelles absorbées par les racines. Elle fait appel à une fonction annexe que nous définirons dans la section suivante.

Comme nous travaillons avec des entrées et sorties définies sur le profil, il est nécessaire de faire une boucle sur tous les points de calcul du profil de sol. Le nombre de ces points n'est pas connu à la création du module. Il ne sera défini que lors de l'exécution quand vous choisirez le maillage de votre profil de sol. C'est pour cela que la plateforme met à votre disposition une variable correspondant au nombre de points de calculs dans le profil de sol. De plus, nous utilisons une autre constante *vsoil_grid_nodes_delta* correspondant à la distance entre deux points contigus de la grille.

```
! Feddes water uptake model
do i = 1 , vsoil_grid_size
    alphasink = alph_func( h ( i ) )
    sinkw( i ) = alphasink * dens_rac( i ) * smax
end do
```

Listing 4 – Absorption racinaire réelle

La troisième et dernière partie est l'intégration du prélèvement d'eau par les racines calculé dans la deuxième partie sur le profil pour obtenir la transpiration réelle de la plante. Pour cela on utilise une fonction d'intégration générique de la plateforme qui se trouve dans le panneau de gauche dans la section *Utility*.

Listing 5 – Intégration sur le profil

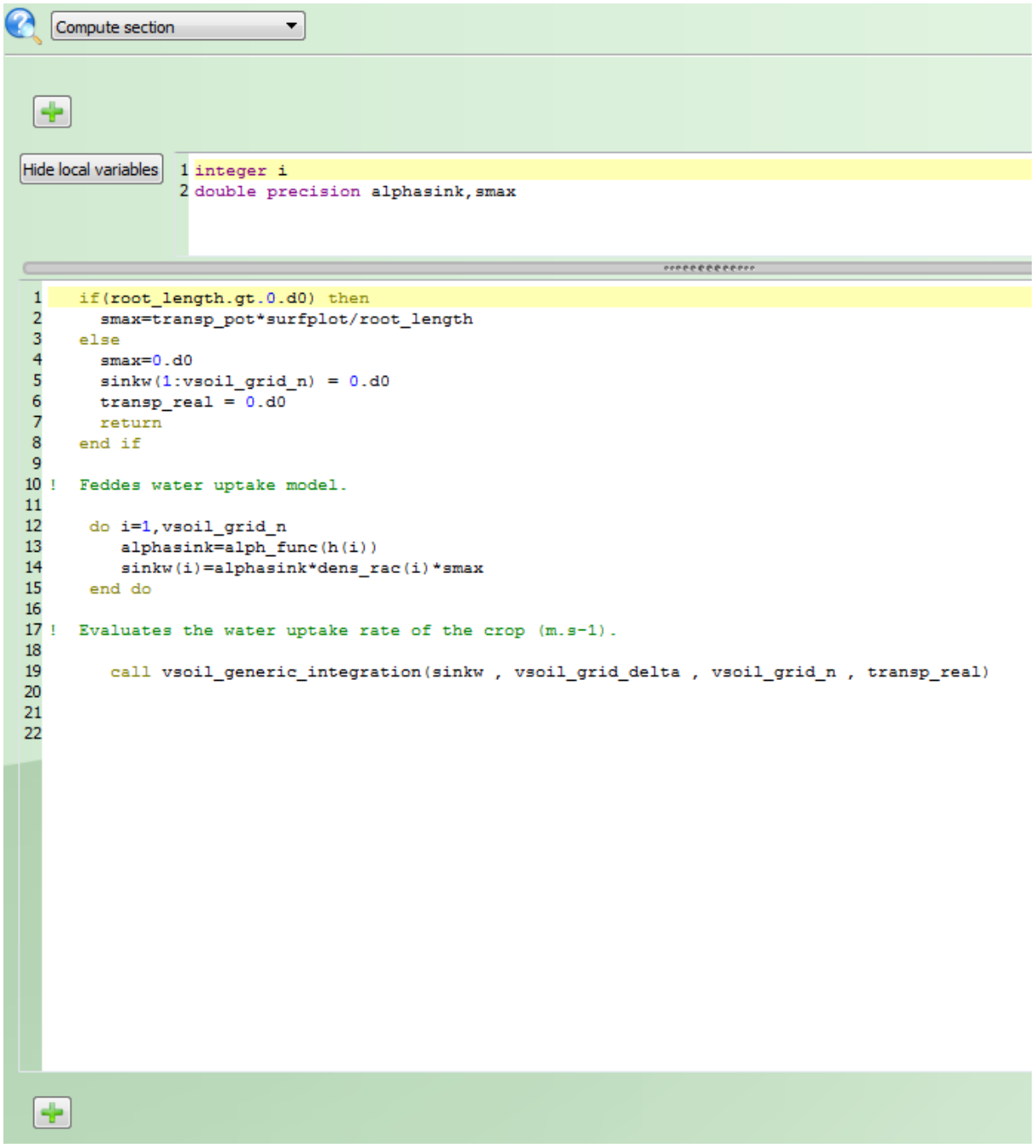


FIGURE 10 – Capture d'écran du composant de plateforme vsoil-module sur l'onglet de l'éditeur de code. Il est représenté la partie de droite lorsque la section de calcul est sélectionnée et que les variables locales sont affichées.

3.5.3 Internal functions section

Dans cette partie nous allons écrire toutes les fonctions annexes que nous utilisons dans le programme principal. Ici nous n'en retrouvons qu'une seule.

```
!
```

```
!       FUNCTION ALPH_FUNC
!
    ------------------------------------------------------------------

!       This is the weighting function for the water uptake black box model
!       Here, a piece-wise linear function is used. A smooth function can also
!       be used.
!
    ------------------------------------------------------------------


double precision function alph_func(x)

double precision,intent(in) :: x

if(x.ge.h1.or.x.le.h4) then
    alph_func = 0.d0
else
    if(x.lt.h3) then
        alph_func = (x-h4)/(h3-h4)
    else
        if(x.gt.h2) then
            alph_func = (x-h1)/(h2-h1)
        else
            alph_func = 1.d0
        end if
    end if
end if

end function alph_func
```

Listing 6 – Écriture de la fonction $\alpha(h)$

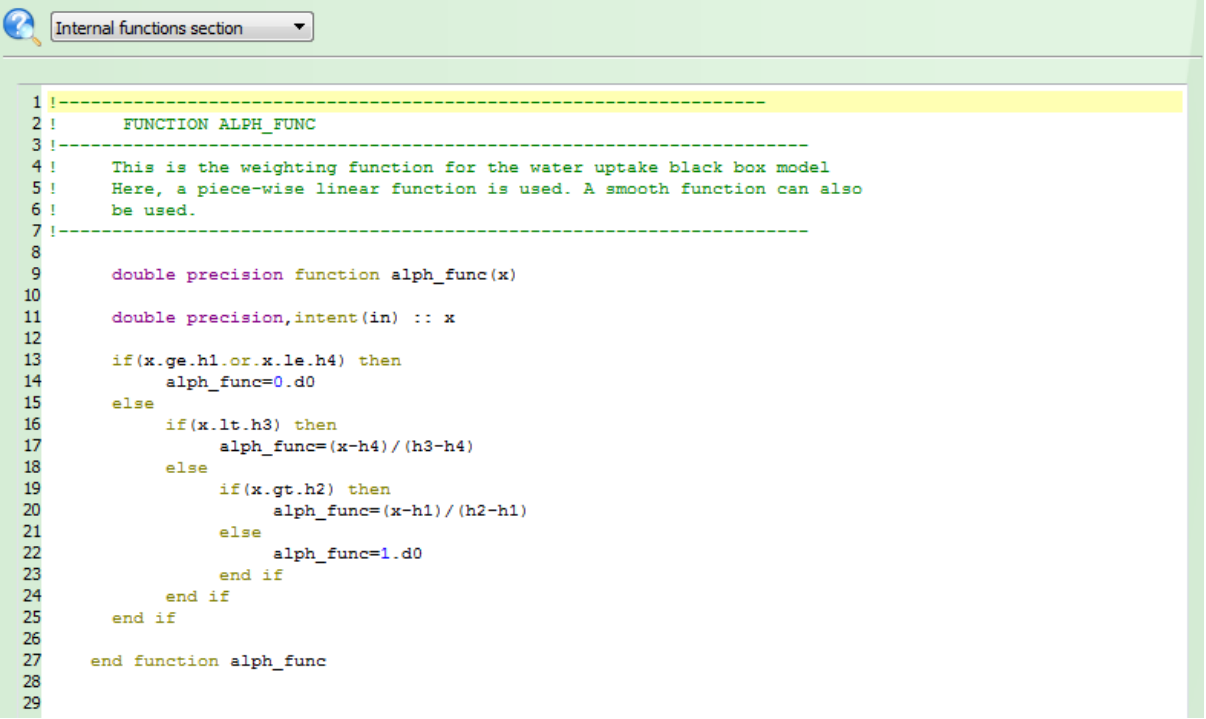


FIGURE 11 – Capture d'écran du composant de plateforme vsoil-module sur l'onglet de l'éditeur de code. Il est représenté la partie de droite lorsque la section des fonctions internes est sélectionnée.

Une fois toutes ces parties de code remplies, nous faisons vérifier à la plateforme que le programme ne comporte pas d'erreur de syntaxe. Pour cela on clique sur le bouton en haut à gauche « Check code ». Lorsqu'il n'y a pas d'erreur, nous pouvons passer à l'onglet suivant.

4 Ce que vous avez appris.

A l'issue de ce tutoriel vous avez vu comment on lie un module à un processus. Vous avez pu suivre la sélection des entrées et sorties utilisées dans un module pour la simulation ainsi que la création de paramètres simples nécessaires au déroulement de la simulation.

Enfin, vous avez découvert l'utilité de trois des parties de code :

— « Initialization section » qui permet d'affecter des valeurs initiales à toutes les variables.
— « Compute section » qui correspond au fonctionnement principal de la simulation.
— « Internal functions section » qui permet de déclarer des fonctions utilitaires à exploiter dans la section « Compute section ».