vsoil-modules user manual

Cédric Nouguier, Nicolas Moitrier

2024-02-29

Table des matières

Table des figures

Introduction

Ce logiciel fait partie de la plate-forme « Sol Virtuel » . Il permet l'implémentation informatique de processus scientifiques nommés modules. Deux langages informatiques sont supportés : le Fortran (version 2003) et le C++.
Ce logiciel utilise la description scientifique des processus réalisés dans l'atelier vsoil-processes et fournira la description de modules au sens informatique à l'application vsoil-models.
L'atelier permet de gérer (visualiser, créer, modifier) des modules en association avec la partie processus.

Un module est une implémentation possible algorithmique et informatique d'un processus.

Un module sera donc caractérisé par :

- le processus auquel il est lié
- les entrées et sorties du processus qu'il va utiliser et fournir
- les sorties initiales
- les paramètres
- le code source implémentant l'algorithme mathématique

Il sera également agrémenté d'un ensemble d'informations aidant à la caractérisation et l'utilisation de celui-ci.

1 Écran d'accueil

La version de l'application que vous utilisez est indiquée dans le titre de la fenêtre. 11 opérations de manipulation de modules sont accessibles après le lancement de l'application vsoil-modules (cf. Figure 1 Fonctions principales de l'application vsoil-modules).

1. Visualiser un module
2. Créer un nouveau module à partir de la configuration par défaut
3. Créer des modules à partir d'un module existant
4. Modifier un module existant
5. Renommer des modules utilisateurs
6. Supprimer des modules utilisateurs
7. Sauvegarder votre environnement
8. Restaurer la configuration par défaut
9. Importer des modules
10. Exporter des modules
11. Consulter la documentation

FIGURE 1 – Fonctions principales de l'application vsoil-modules

Les modules utilisateurs sont caractérisés par l'icône alors que les modules officiels non modifiables sont représentés par l'icône .

Les opérations d'import ou d'export, au contraire des autres, ne conduisent pas à l'environnement de travail. L'opération est réalisée et lorsqu'elle est terminée, vous revenez sur l'écran d'accueil.

1.1 Visualisation d'un module

Pour visualiser la représentation informatique d'un processus scientifique, vous pouvez sélectionner la fonction « Visualisation ». Ce mode vous permet de voir le module, le tester mais pas de le modifier.

1.2 Création d'un module

Pour implémenter la représentation informatique d'un processus scientifique, vous pouvez soit créer un nouveau module, soit modifier un module existant.

1.2.1 Création d'un nouveau module

Après la sélection de la fonction « New module », l'espace de travail apparaît (cf. 3 Présentation de l'espace de travail).
5 étapes sont nécessaires à la création du module et la première est accessible :

- **Étape 1** , « Process » : sélection du processus.
- **Étape 2** , « Module » : description du module.
- **Étape 3** , « Inputs / Outputs » : choix des entrées / sorties et sorties initiales.
- **Étape 4** , « Parameters » : saisie des paramètres.
- **Étape 5** , « Code editor » : édition du code.

1.2.2 Création d'un nouveau module à partir d'un module déjà existant

Il est possible de copier l'ensemble des caractéristiques et codes d'un module existant en sélectionnant la fonction « Duplicate module ».
Les modules officiels peuvent être copiés pour être modifiés.
La liste des modules disponibles est affichée.

- Sélectionnez le module à copier.
- Saisissez un nouveau nom pour ce module, vous pourrez ensuite soit :
  - copier et éditer ce nouveau module pour le modifier (« Duplicate and edit module »)
  - copier et revenir sur l'Écran d'accueil (« Duplicate module »)
  - annuler l'opération de copie et revenir sur l'Écran d'accueil (« Cancel »)

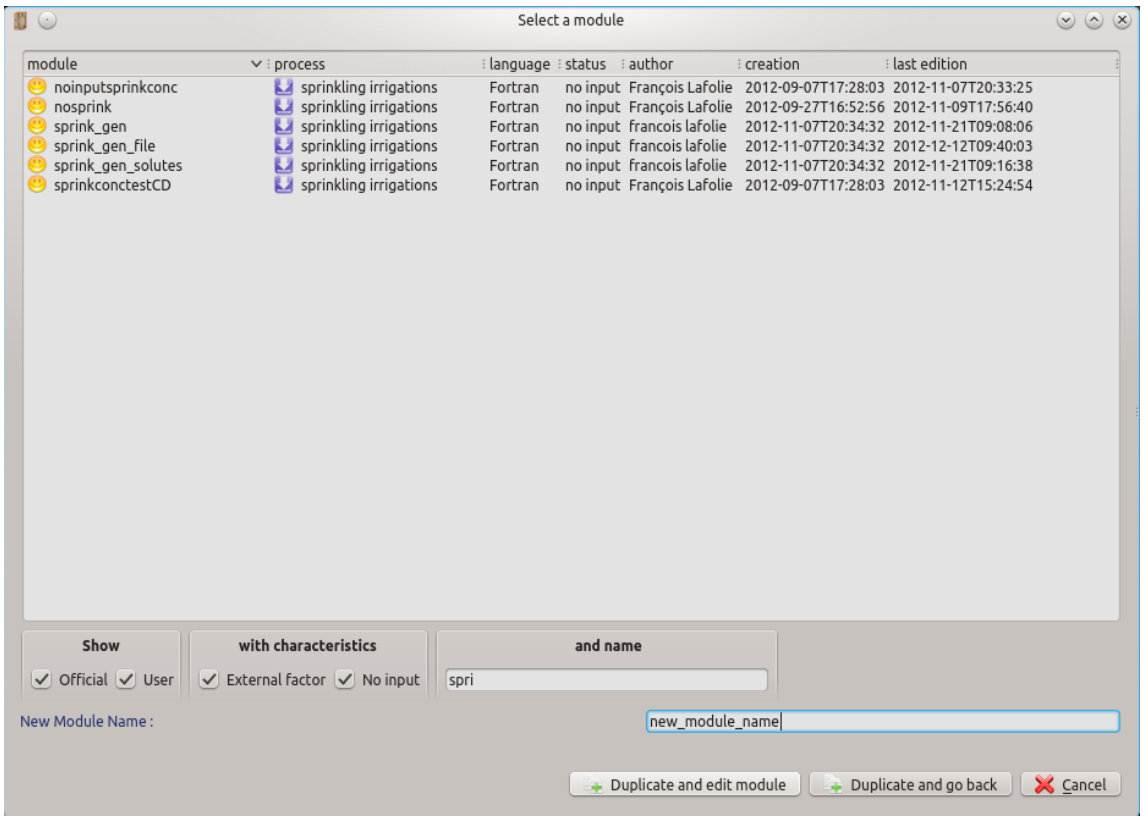


FIGURE 2 – Création d'un module à partir d'un existant

1.3 Modification d'un module

1.3.1 Renommage d'un module

La fonction de renommage est accessible uniquement par l'écran d'accueil.
Les modules officiels ne peuvent pas être renommés.

- Sélectionnez le module à renommer
- Saisissez un nouveau nom valide, vous pourrez ensuite soit :
  - renommer et éditer ce module renommé pour modifier ses caractéristiques et son code informatique (« Rename and edit module »)
  - renommer et revenir sur l'écran d'accueil (« Rename module »)
  - annuler l'opération de renommage et revenir sur l'écran d'accueil (« Cancel »)

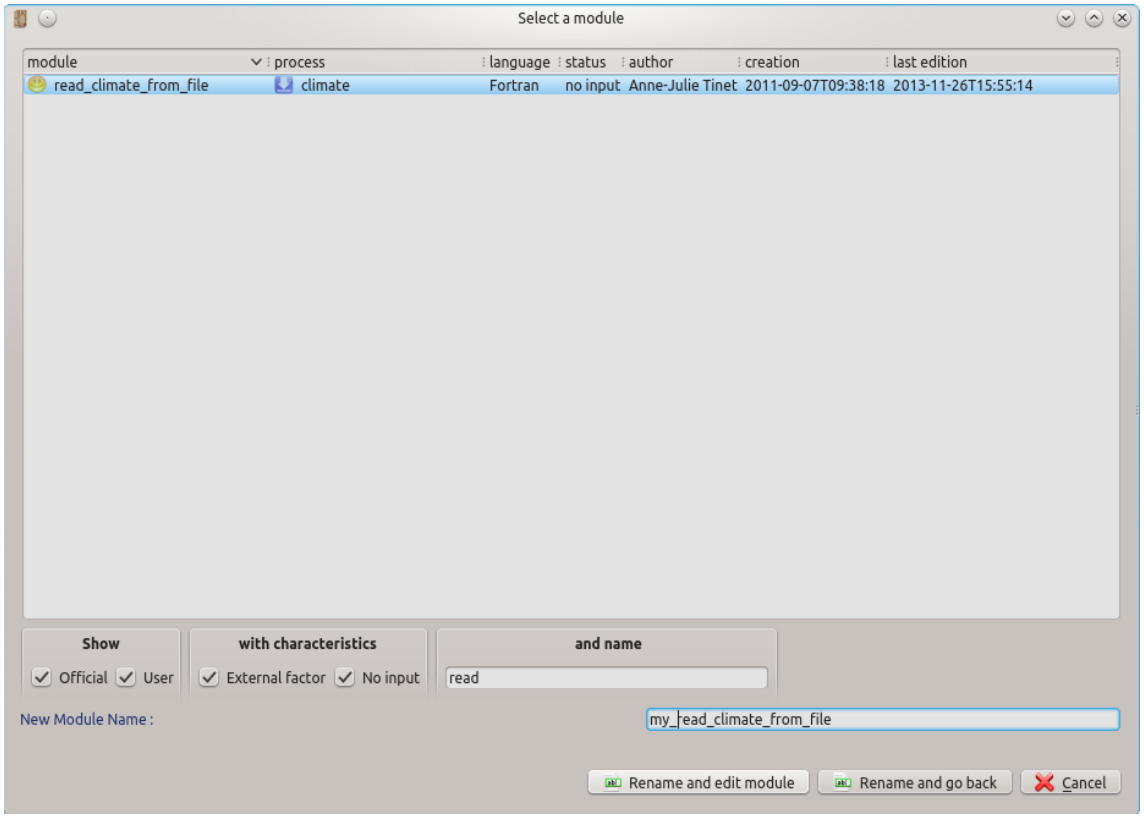


FIGURE 3 – Renommage d'un module

1.3.2 Modification du contenu d'un module

Sélectionnez le module souhaité, des filtres sur la partie basse de l'écran permettent de retrouver rapidement le module souhaité.

Attention les modules officiels ne sont pas modifiables. Leur édition est possible mais en mode « lecture seule », c'est-à-dire que vous pouvez visualiser leur contenu sans modification. Tous les champs de saisies seront inaccessibles.

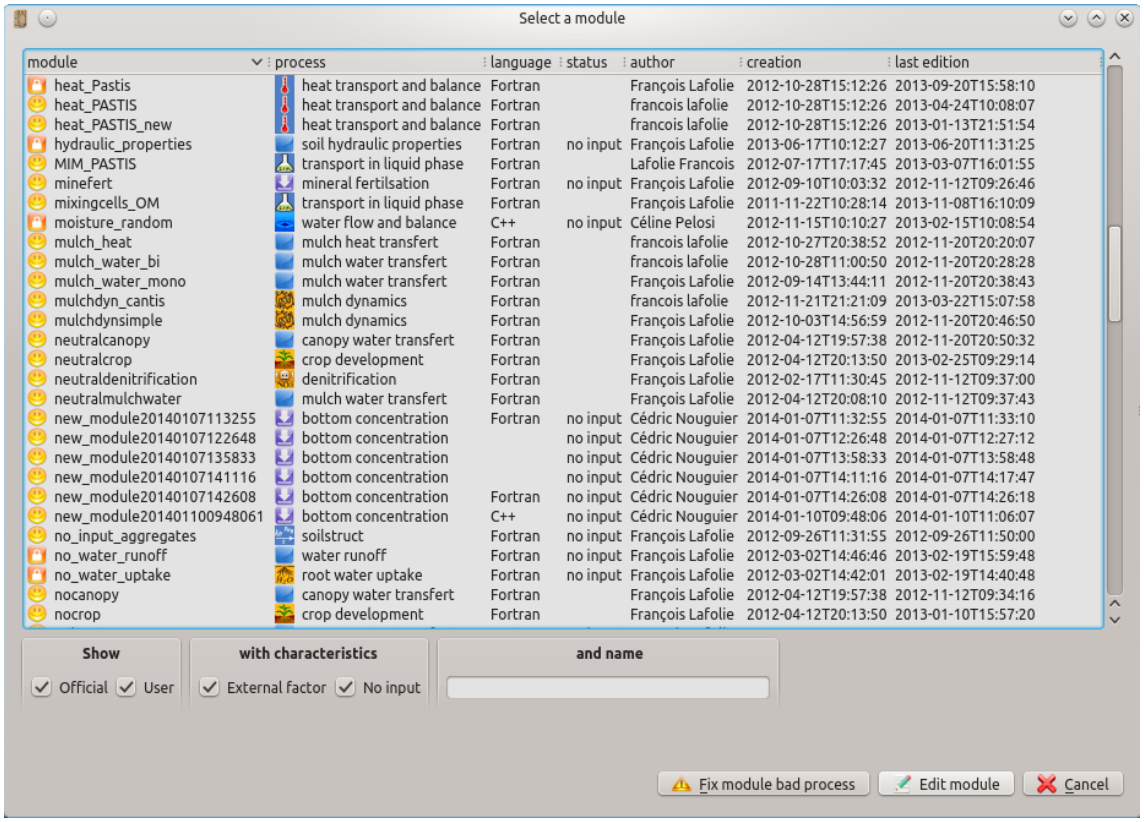


FIGURE 4 – Liste des modules à éditer

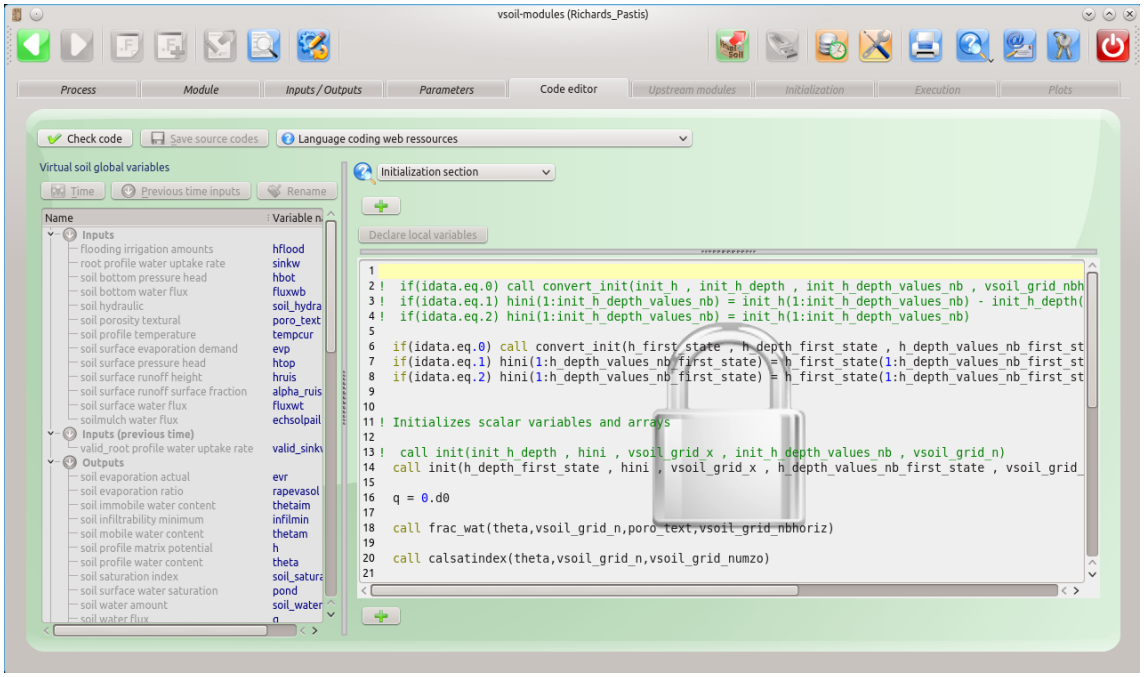


FIGURE 5 – Édition d'un module officiel

L'environnement de travail s'ouvre, initialisé avec les données du module.
L'application se positionnera automatiquement sur l'onglet correspondant à la dernière étape validée.

Cas particulier : modules non affectés à un processus ou à des processus.

Lors de la sélection d'un module à éditer, des modules en grisés peuvent apparaître. Une nouvelle fonction est alors accessible : corriger les modules non affectés à des processus.

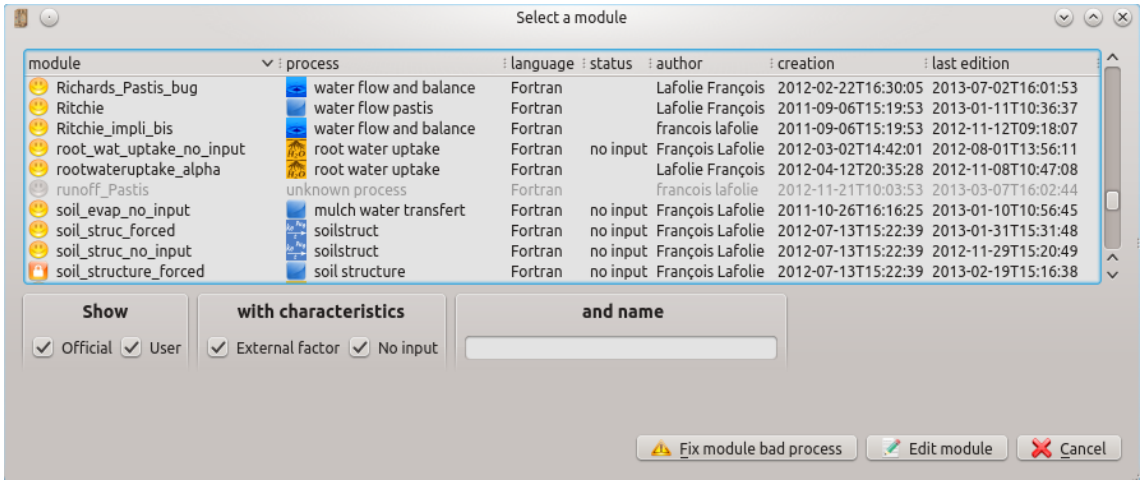


FIGURE 6 – Liste des modules à éditer avec des modules non affectés à un processus

Si vous souhaitez réaffecter les modules, alors cliquez sur « Fix module bad process ».

Une nouvelle fenêtre apparaît, avec une liste de modules qui n'ont plus de processus associés. Sélectionnez le module a réaffecter (partie en haut). Ensuite sélectionnez le processus de réaffectation (liste partie basse) puis cliquez sur « change module process link ». Vous revenez sur la liste des modules à éditer.

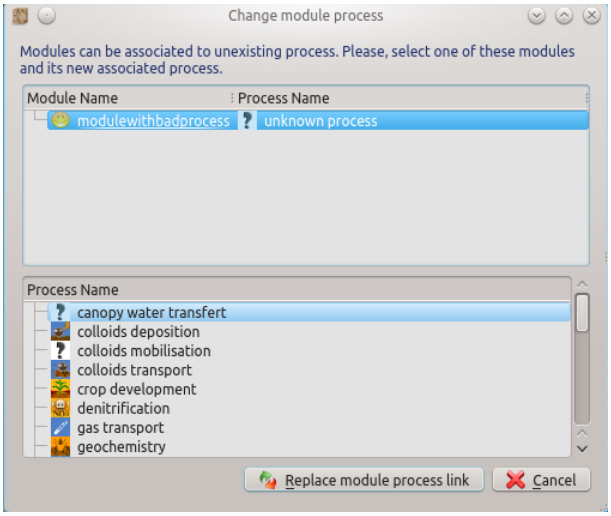


FIGURE 7 – Réaffectation module-processus

1.3.3 Destruction d'un module

La destruction d'un module entraîne la destruction de tous les éléments qui le composent (caractéristiques, codes sources...). Les modules officiels ne peuvent pas être détruits.

- Sélectionnez le module utilisateur à détruire.
- Ensuite vous pouvez soit :
  - le détruire et revenir sur l'Écran d'accueil (« Remove module »)
  - annuler l'opération de destruction et revenir sur l'Écran d'accueil (« Cancel »)

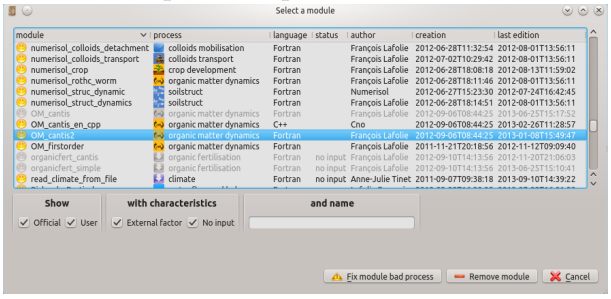


FIGURE 8 – Destruction d'un module

2 Fonction d'importation, exportation de modules et restauration des données

Ces fonctions sont accessibles via l'écran d'accueil et certaines dans l'espace de travail (dans la barre d'outils commune).
Pour les fonctions d'importation et d'exportation, référez-vous à la documentation dédiée vsoil_import_export.pdf.
Pour la fonction de restauration, reportez-vous à la documentation des fonctions générales vsoil_common_toolbar.pdf.

3 Présentation de l'espace de travail

Après avoir sélectionné une des fonctions de création ou d'édition d'un module, l'espace de travail apparaît, configuré en fonction du choix effectué précédemment (écran accueil). La configuration comprend l'initialisation des valeurs des différentes données et l'accessibilité à différentes fonctionnalités et étapes.

L'espace est divisé en deux grandes parties (cf. 9) :

- partie supérieure : barre de fonctions générales
- partie inférieure : espace de saisie/consultation

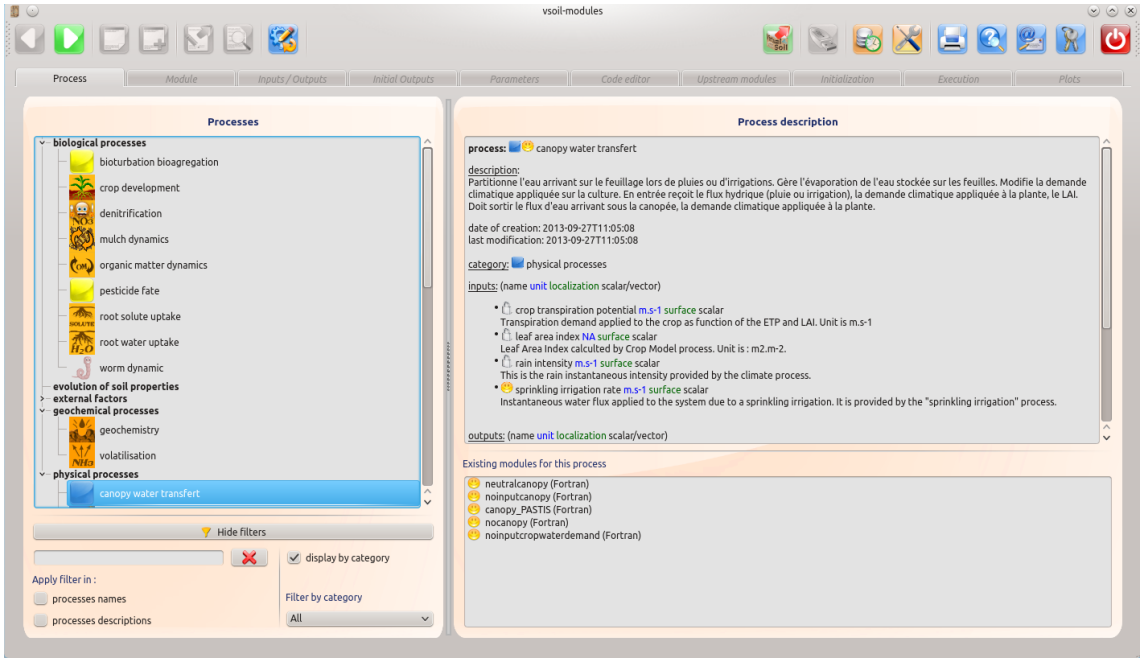


FIGURE 9 – Environnement de travail de l'application vsoil-modules

Barre d'outils générale : commune aux applications de « Sol Virtuel » , se reporter au manuel introductif de la plate-forme.

Espace de travail : informations liées au module sélectionné ou en cours de création. Les informations sont regroupées en thèmes qui peuvent être assimilés à des étapes. Chaque étape est représentée par un onglet dont le contenu doit être validé pour pouvoir accéder aux suivants. L'objectif est de guider l'utilisateur dans la création d'un module « Sol Virtuel » .

6 étapes ont été identifiées pour permettre la création d'un module, 4 autres concernent le test du module. La couleur de fond de chaque écran symbolise le type d'information traité :

- la couleur de fond « saumon »symbolise les processus
- la couleur de fond « verte »symbolise les modules

Une barre d'outils spécifique contient des actions accessibles depuis tous les onglets de l'application. Les icônes grisés seront expliqués plus loin dans ce document (cf. 3.5.7 Autres fonctionnalités).

FIGURE 10 – Aperçu général des icônes spécifiques

 Permet de retourner sur l'onglet précédent.
Lorsque l'on est sur le premier onglet des processus, cette icône est grisée et non active.

 Permet de valider l'onglet courant et d'afficher l'onglet suivant.
Lorsque l'on est sur le dernier onglet ou que l'onglet courant n'est pas validé, cette icône est grisée et non active.

 Permet d'éditer un autre module. Celui en cours est sauvé.

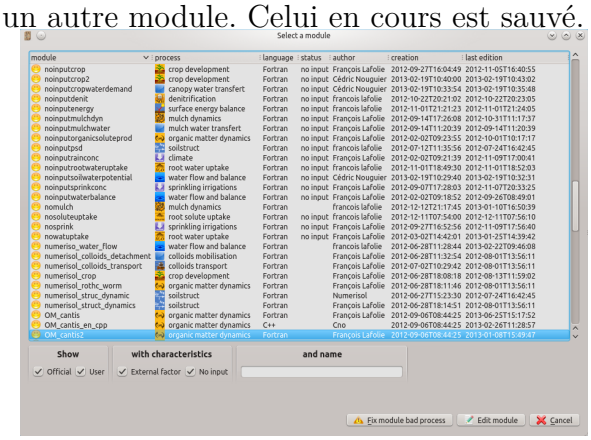


FIGURE 11 – Changement de module, liste pour éditer un autre module

3.1 Étape 1 : lien processus module

Un module est associé à un seul processus. Les processus disponibles sont ceux créés via l'application vsoil-processes. Pour sélectionner le processus dont l'implémentation va être réalisée, cliquez sur le 1er onglet nommé « Process » Process.

Cet onglet est celui par défaut dans le cas de la création d'un nouveau module. Il est alors initialisé à partir de la configuration par défaut.
La liste des processus et des « external factors »(officiels et utilisateurs) disponibles est affichée. Pour obtenir les informations concernant un processus, sélectionnez-le et celles-ci s'afficheront dans la partie droite de l'application. En bas à droite de l'interface une liste de modules déjà créés à partir du processus courant s'affiche.
Si trop de processus sont présents, il est possible de les filtrer grâce aux options présentes en dessous de la liste des processus.

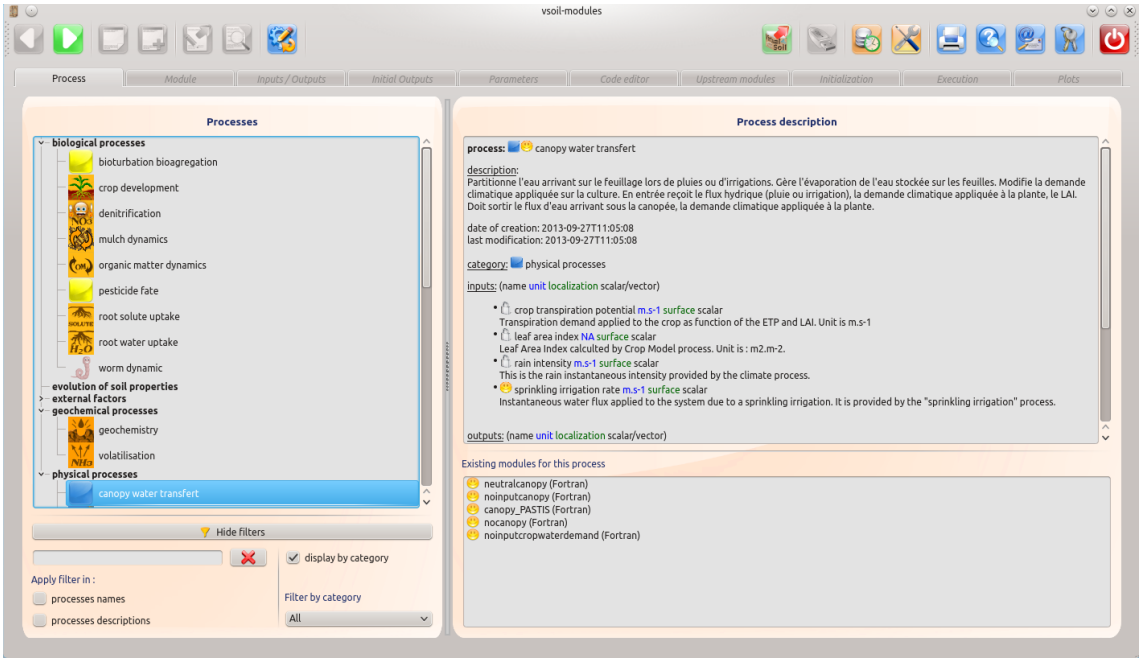


FIGURE 12 – Étape 1, aperçu général de l'onglet « Process »

Pour vous aider dans la recherche du processus souhaité vous avez dans la partie en bas à gauche un espace de filtrage.

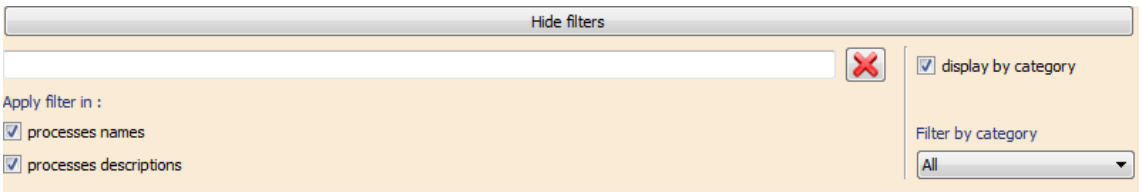


FIGURE 13 – Critères de filtrage des processus

Partie gauche : filtrage est associée au nom des processus et à leur description.

Un espace de saisie vous permet de saisir le nom (ou une partie du nom) du processus ou d'un élément de description de celui-ci.
Vous pouvez lancer la recherche de la chaîne saisie :

- sur les noms de processus
- sur leur description
- sur ces deux critères

Vous recherchez par exemple la chaîne « water ».

Si vous sélectionnez uniquement le critère « processes name », vous obtiendrez la liste des processus dont le nom contient le terme « water ».

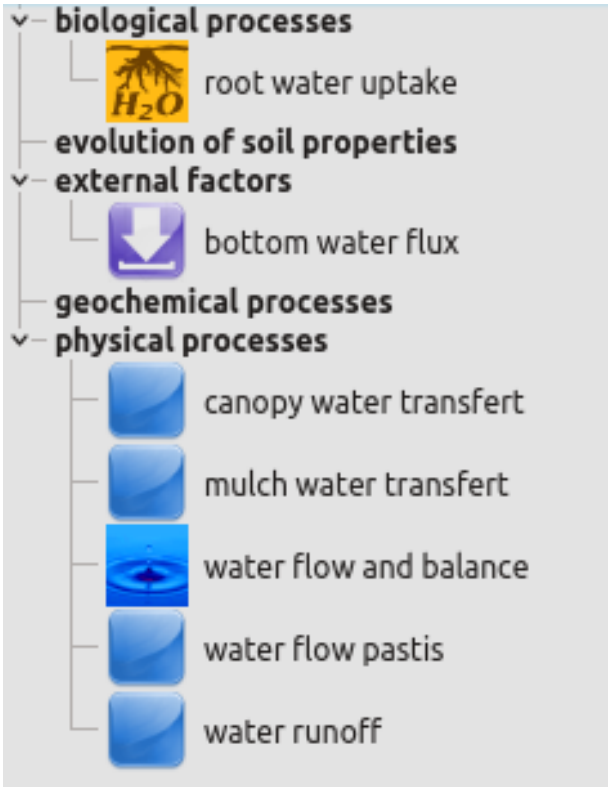


FIGURE 14 – Liste des processus dont le nom contient la chaîne « water »

Si vous sélectionnez uniquement le critère « processes description », vous obtiendrez la liste des processus dont la description contient le terme « water ».

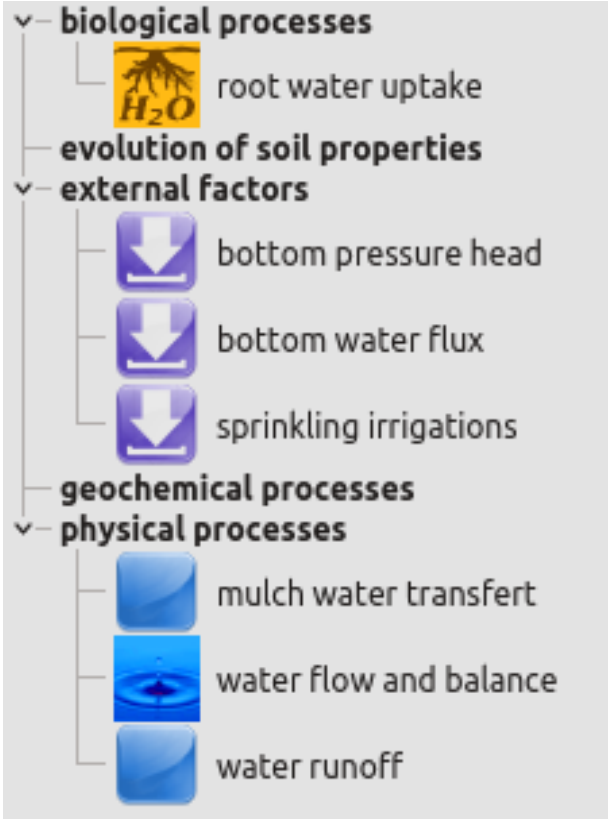


FIGURE 15 – Liste des processus dont la description contient la chaîne « water »

Partie droite : Vous pouvez lister les processus triés par ordre alphabétique ou groupés par leur catégorie d'appartenance.

3.2 Étape 2 : informations sur le module

Lorsqu'un processus a été sélectionné dans le premier onglet, l'utilisateur doit saisir les informations générales du module qu'il souhaite créer.

Sélectionnez le bouton (dans la barre d'outil en haut) ou l'onglet Process.

Cet écran est composé de 2 parties :

- la partie avec la couleur de fond saumon est liée au processus et est en consultation seule
- la partie avec la couleur de fond verte est liée au module et l'ensemble des champs est modifiable, exception faite des dates de création et de modification du module données à titre informatif et gérées par le système

Le nom du module, son auteur et sa description sont obligatoires. Vous pouvez caractériser plus finement le module à l'aide d'un ensemble d'informations optionnelles : une liste de mots clefs, une liste de publications associées et une liste de publications avec lien vers un site internet. Ces informations sont nécessaires pour la diffusion de votre module et obligatoires pour leur « officialisation ».
Le nom du module doit être unique et ne doit pas être un nom de processus ou de variable. Il doit au minimum comporter 3 lettres. Tant que le nom du module est invalide, il sera de couleur rouge et vous ne pourrez pas passer à l'étape suivante.

Lors de l'édition d'un module, l'information déjà saisie est affichée pour modification éventuelle (sauf pour le nom du module qui peut être changé uniquement lors du lancement de

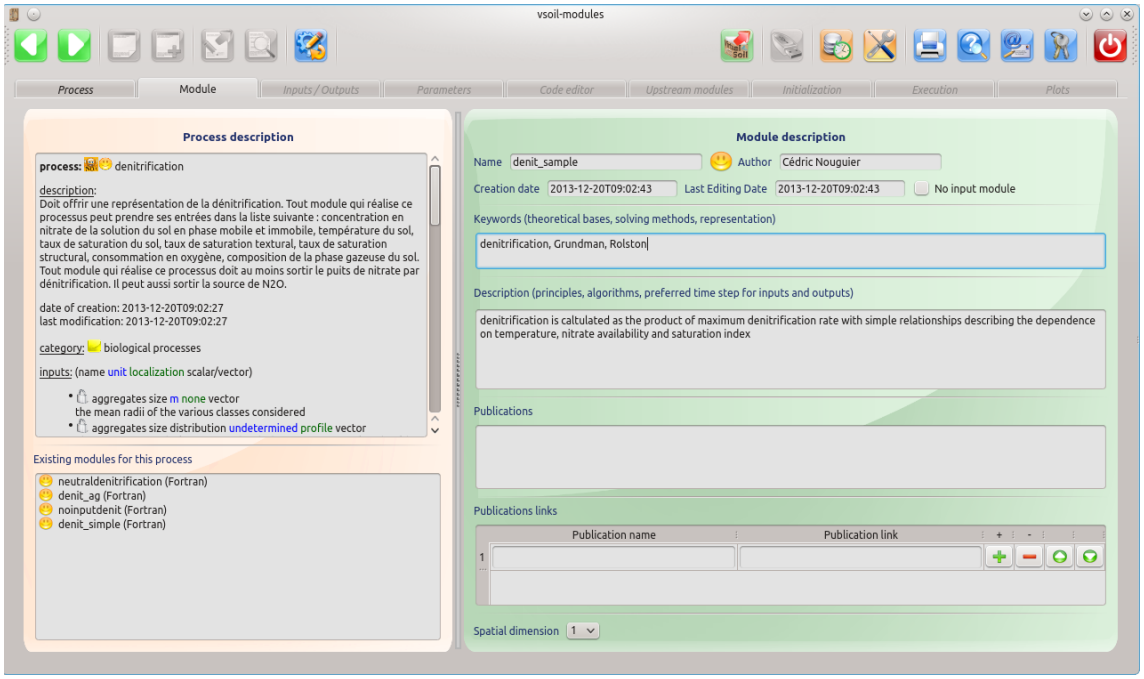


FIGURE 16 – Étape 2, aperçu général de l'onglet « Module »

l'application).

Lorsqu'un module est officiel, il n'est pas possible de le modifier et l'icône suivante apparaît 

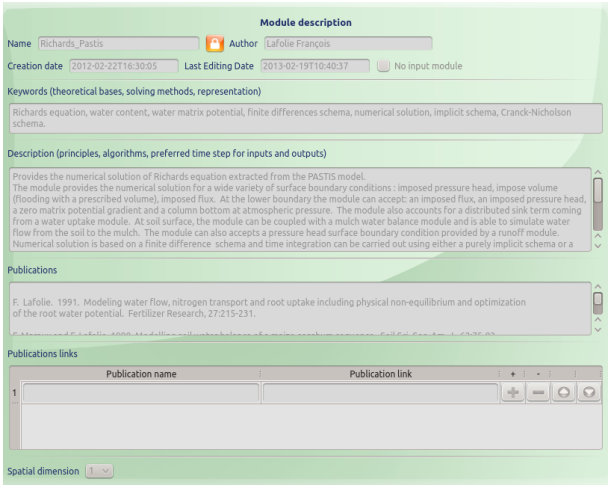


FIGURE 17 – Module officiel, caractéristiques non modifiables

sinon, l'icône permet d'identifier un module utilisateur, donc modifiable.

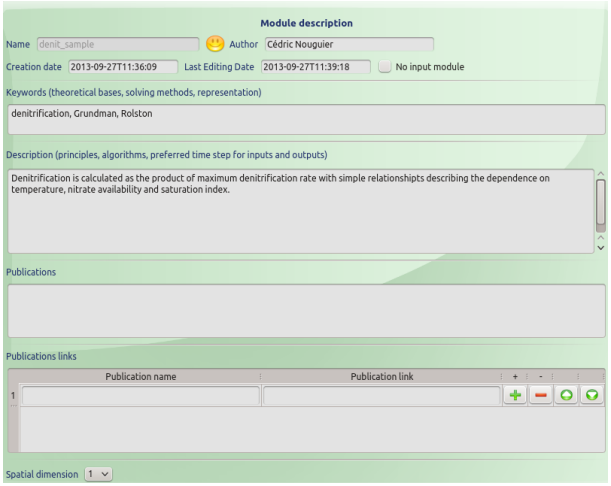
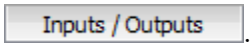


FIGURE 18 – Module utilisateur, caractéristiques modifiables

3.3 Étape 3 : entrées / sorties du module

Le troisième onglet concerne les entrées et sorties du module 

.
Il faut sélectionner au moins une entrée et une sortie parmi celles du processus associé au module. Il y a une exception si le processus est de type “external factor” ou si le module est marqué “no input”. Dans ce cas, il n’y a pas d’entrée. La sélection se fait à l’aide de la case à cocher située à gauche de chaque entrée et sortie.

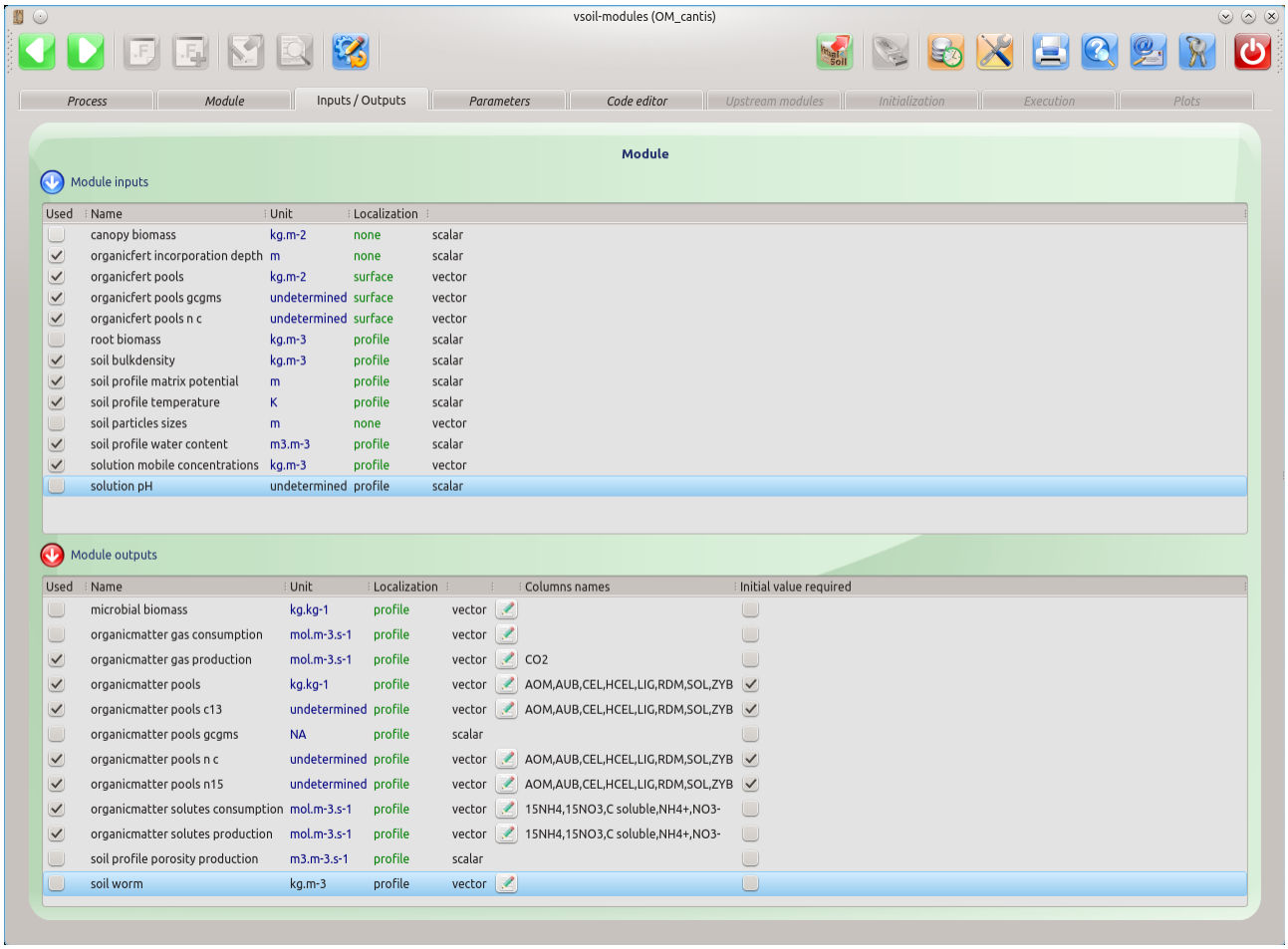


FIGURE 19 – Étape 3, aperçu général de l’onglet « Inputs / Outputs»

3.3.1 Available options for inputs

initial input required In the column “initial input required”, a ticked box means that an initial situation will be provided for the variable at run time by an upstream module. The initial values have the names of their corresponding variables. NB : Variables not selected -without a ticked box- will not be available.

iterate with In the column “iterate with”, a ticked box means that this module may iterate with the module providing this input during the same time step. NB : Variables not selected -without a ticked box- will not be available.

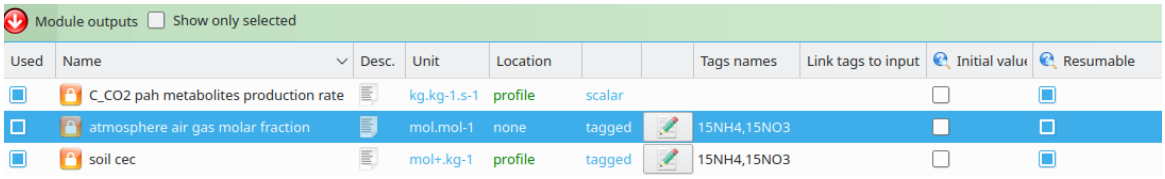


FIGURE 20 – options possibles pour les sorties

3.3.2 Available options for outputs

Pour les sorties, plusieurs options sont possibles. Elles sont accessible à droite du nom de chaque variable de sortie. Voir figure 20.

tagged Lorsqu'une sortie est de type « tagged », il y a alors plusieurs valeurs associées à la sortie. Ils sont appelés des "tags". Il y a 3 façons d'associer des "tags" à une sortie :

1. Choisir des "tags" dès à présent. Pour cela, il faut cliquer sur le bouton de la colonne "tag names". Une boîte de dialogue permet alors de sélectionner les "tags" qui, après validation seront associés à la sortie. Voir la copie d'écran à droite.
2. Si les "tags" ne sont pas connus au moment de la création du module, on peut choisir de ne pas sélectionner de "tags". Ils devront alors être déterminés lors de l'étape de paramétrisation du modèle.
3. Si les "tags" de la sortie sont les mêmes que ceux d'une des entrées du module, il est possible de forcer les mêmes "tags". Il suffit de cocher la case "Link tags to input". La copie se fera immédiatement ou lors de l'étape de paramétrisation du modèle, selon les cas.

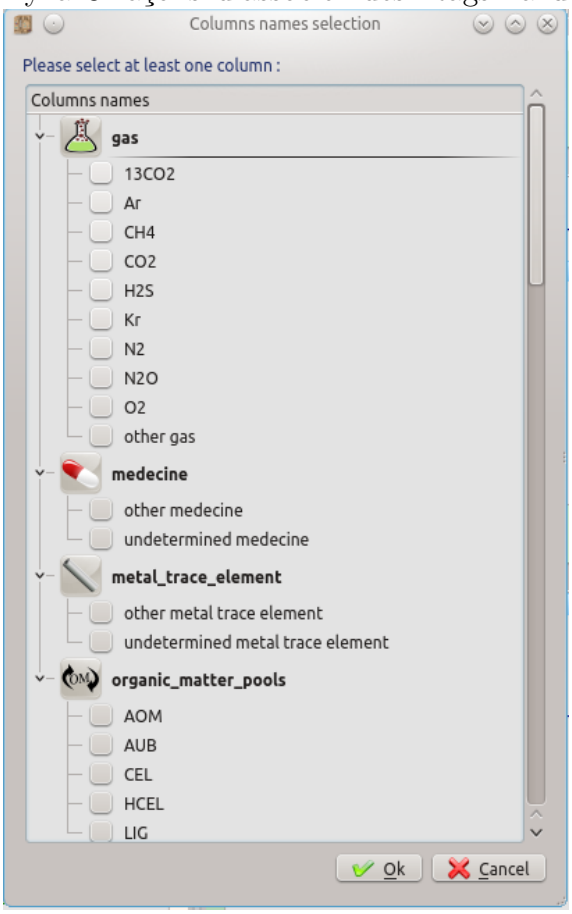
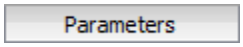


FIGURE 21 – Sélection des "tags" d'une sorties

Initial value required In the column "Initial value required", a ticked box means that an initial situation must be provided for the variable at run time. In this case, this variable must be initialized with explicit values in the "Initialization" tab. Then a new variable, suffixed by the "_first_state" character string, is automatically generated and can be used from the "Initialization section" of the "Code editor" tab. NB : variables not selected without a ticked box must be initialized in the "Initialization section" of the "Code editor" tab in some other way. Generally, it can be done from parameters, constants or other input variables.

Resumable La colonne “Resumable” indique que l’état de la variable peut être sauvegardé, en fin de simulation, dans un fichier d’initialisation généré automatiquement. Si la variable est de type “profile”, son état sera sauvegardé pour tous les points de la grille. Si la variable est “tagged”, ses différentes tags seront sauvegardés. Ainsi, lors de la simulation d’un modèle, à la demande de l’utilisateur, toutes les variables “resumable” sont sauvegardées automatiquement. L’objectif est de pouvoir relancer le modèle à partir d’une précédente simulation. La nouvelle simulation reprend alors au moment de l’arrêt de la précédente, en se servant de la sauvegarde. Les sorties “resumable” sont automatiquement initialisées avec les valeurs de fin précédentes. NB : Une variable ne peut être à la fois “resumable” et “initial output”. Pour savoir comment gérer les variables non résumables dans le code d’initialisation des modules, voir la section dédiée 3.5.6.

3.4 Étape 4 : paramètres du module

Le quatrième onglet 

permet de renseigner une liste de paramètres utiles à l'initialisation du module. Il est également possible d'ajouter des fichiers qui seront accessibles lors de l'étape de codage. Cette étape reste facultative.
Il est possible d'ajouter des paramètres, de modifier leur ordre d'apparition ou de les supprimer.

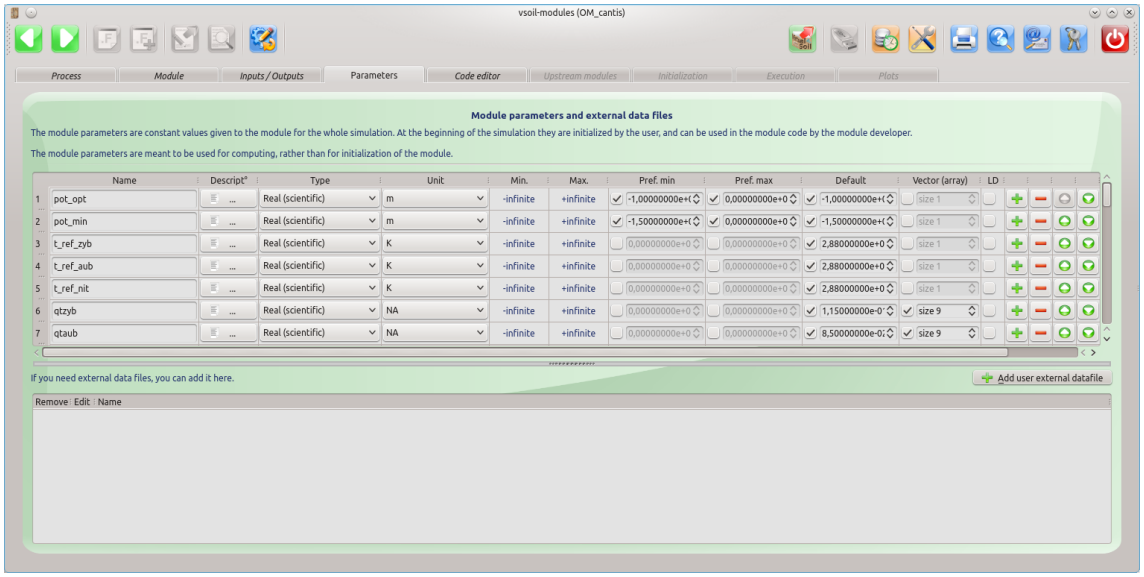


FIGURE 22 – Étape 4, aperçu général de l'onglet « Parameters »

Un ensemble d'informations est nécessaire pour chaque paramètre :

FIGURE 23 – Caractéristiques du paramètre

Le nom doit être unique pour le module, sinon la valeur se met en rouge.
La description du paramètre est obligatoire et accessible en cliquant sur le bouton 
ou  , cette dernière icône indiquant que la description n'a pas encore été renseignée. Le texte de description s'affiche lorsque l'on laisse la souris sur le bouton de description.

Le type est constitué d'une liste est prédéfinie. En fonction du choix le contenu des champs suivants (Unit, Min, Max...) sera modifié.

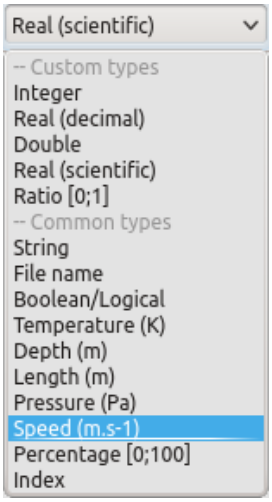


FIGURE 24 – Types de paramètres

L'unité est sélectionnée à partir d'une liste dépendante du type sélectionné.
Pour les types génériques (réel, entier) la liste ci-contre est proposée. Pour les types spécifique l'unité est soit imposée soit non utile (sans unité).

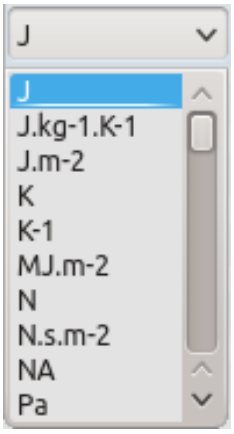


FIGURE 25 – Types de paramètres

Les plages minimale et maximale obligatoires dépendent aussi du type et ne sont pas modifiables. Si vous souhaitez borner plus finement le paramètre, il vous faut modifier la valeur des 2 champs suivants « Pref. min »et « Pref. max ».
Les plages souhaitées minimales et maximales bornent la valeur du paramètre. Une cohérence entre la valeur de ces 2 champs et la valeur par défaut du champ est demandée : la valeur par défaut doit être comprise entre les bornes minimale et maximale, sinon ces 3 champs se colorent en rouge.

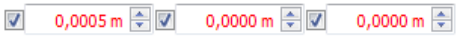


FIGURE 26 – Plages spécifiques du paramètre

Les valeurs de ces plages sont par défaut les mêmes que celles des plages obligatoires. Si vous souhaitez les modifier, cochez la case 

et un champ de saisie apparaît alors 

.
La valeur par défaut : valeur affectée lors de l'initialisation de votre paramètre.

Vector : votre paramètre est de type vecteur (lors de l'implémentation informatique ce sera un tableau) dont le nombre de lignes sera celui saisi.
Exemple : 

tableau de 5 lignes.

Les boutons suivants, associés à chaque ligne de saisie d'un paramètre permettent les actions suivantes :

Ajouter une nouvelle ligne de saisie de paramètre, en dessous de celui-ci.
Supprimer le paramètre sur la ligne active.
Déplacer d'un cran vers le haut le paramètre (non accessible sur le 1er paramètre).
Déplacer d'un cran vers le bas le paramètre (non accessible sur le dernier paramètre)

A la moindre erreur (valeur rouge) il ne sera pas possible d'accéder à l'étape suivante.
Vous pouvez ajouter des fichiers (notamment des jeux de données) à votre module avec le bouton suivant : Add user external datafile.

Ces fichiers pourront être modifiés directement en cliquant sur l'icône qui lancera votre éditeur externe (modifiable dans la barre d'outils transverse).
Pour supprimer un fichier, il faut le sélectionner puis appuyer sur le bouton suivant : .

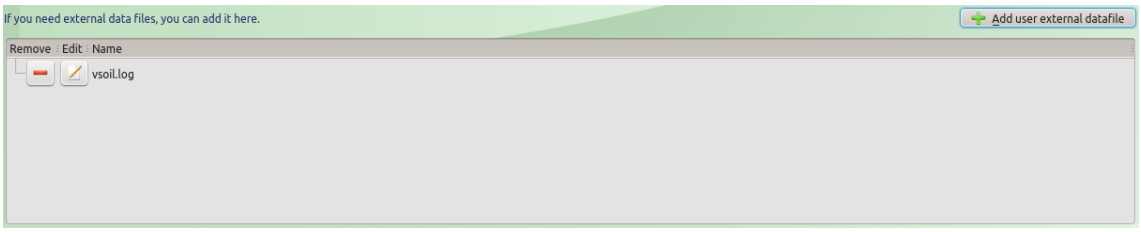


FIGURE 27 – Ajout d'un fichier externe

3.5 Étape 5 : édition du code du module

3.5.1 Présentation de l'onglet

La dernière étape de création d'un module concerne la génération du code associé.
Pour ajouter du code, toutes les étapes précédentes doivent avoir été validées. Cliquez ensuite sur l'onglet « Code editor » Code editor .

L'écran est divisé en trois grandes zones :
- Sur la partie gauche un ensemble de données (organisées en fonction de leur nature), utilisables dans le code. Ces données proviennent soit des caractéristiques des modules, soit des constantes, soit des paramètres de la plate-forme.
- Sur la partie droite, des volets contenant chaque partie du code généré éditable ou non par l'utilisateur sont présentés.
- La partie basse de l'écran contient les icônes des actions possibles ainsi que la fenêtre de vérification du code non visible tant que le code n'a pas été vérifié.

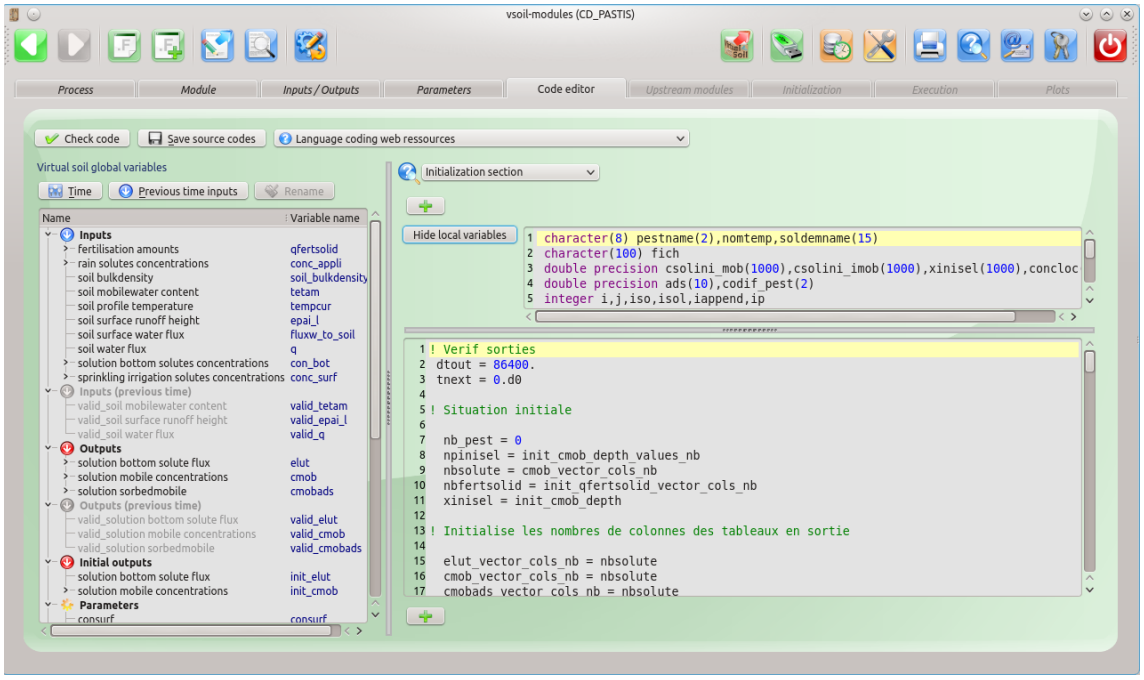


FIGURE 28 – Étape 5, aperçu général de l'onglet « Code editor »

3.5.2 L'arbre des variables

Sur la partie gauche de l'écran, un arbre des variables contient différentes variables et fonctions, facilitant le codage du module. Il s'agit d'un récapitulatif des informations fournies lors des étapes précédentes et d'informations automatiquement fournies par la plate-forme en fonction du langage de programmation choisi.
Pour chaque variable disponible sur la première colonne, correspond un nom de variable compatible avec le langage de programmation sur la deuxième colonne. Si vous souhaitez utiliser, dans une section de code, une variable proposée dans l'arbre, vous devez utiliser la technique du « drag n drop »en sélectionnant la variable avec le bouton gauche de la souris puis en laissant ce bouton appuyé et en lâchant le bouton lorsque la souris se situe sur la zone d'édition de code de la section courante. Certaines variables peuvent être renommées.

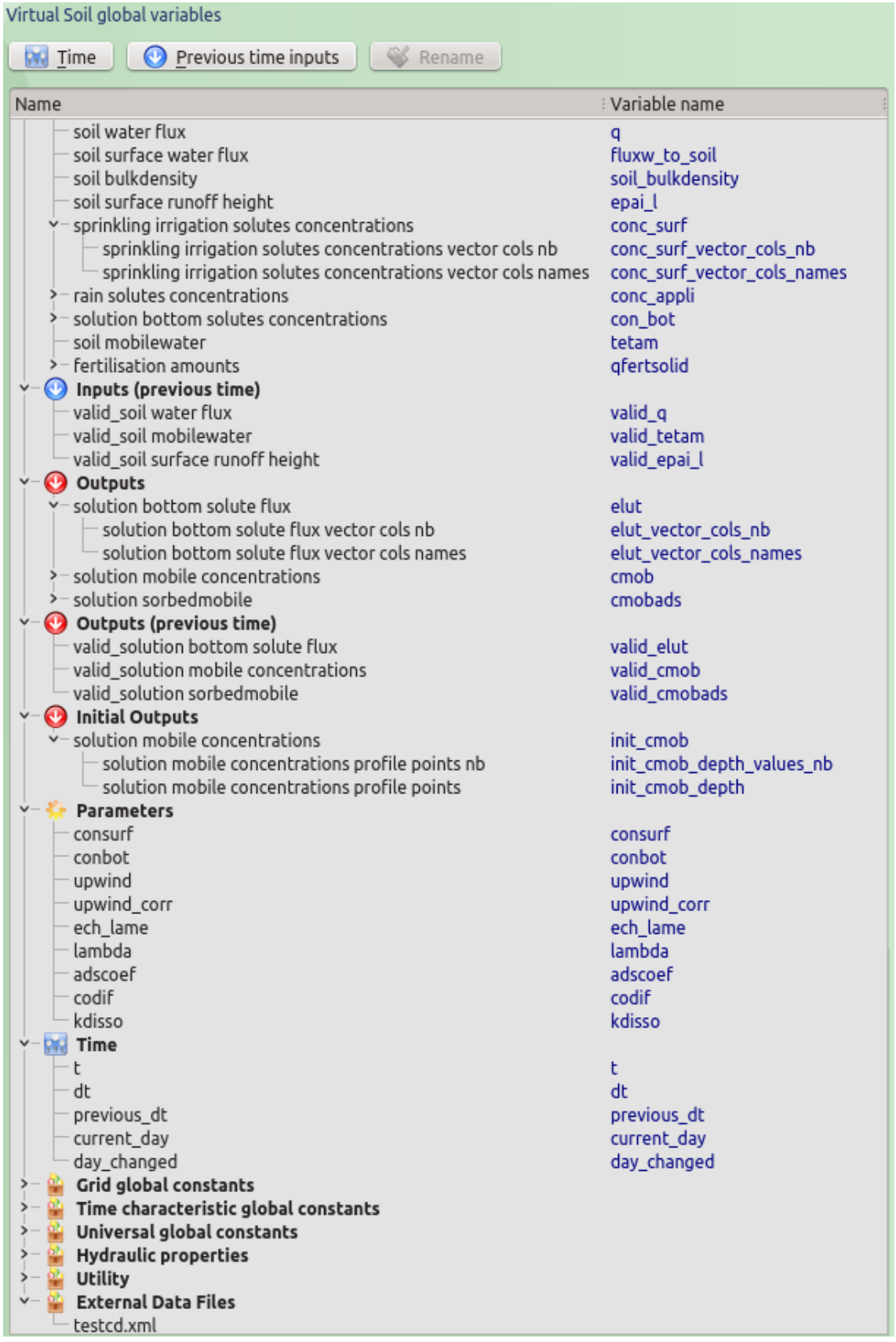


FIGURE 29 – Arbre des variables

L'arbre des variables est constitué de plusieurs sous-arbres.

Inputs

Récapitulatif des entrées sélectionnées dans l'étape 3.
Si l'entrée est de type « is vector », des informations complémentaires sont proposées :

- nombre de colonnes (variable avec suffixe « _vector_cols_nb »)
- Tableau contenant le nom des colonnes (variable avec suffixe « _vector_cols_names »)

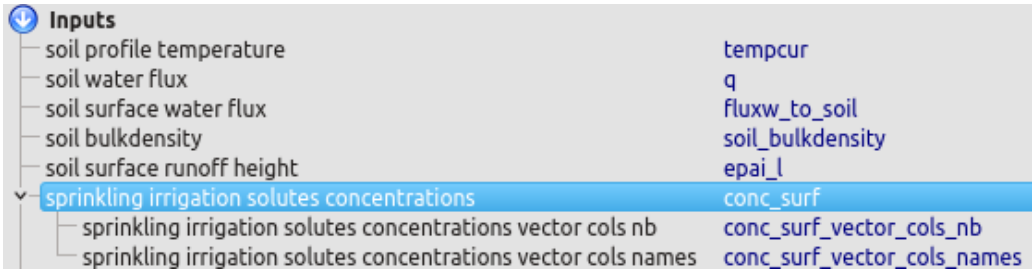


FIGURE 30 – Entrée de type « is vector »et variables associées

Outputs

Récapitulatif des sorties sélectionnées dans l'étape 3. Si la sortie est de type « is vector », des informations complémentaires sont proposées :

- nombre de colonnes (variable avec suffixe « _vector_cols_nb »)
- tableau contenant le nom des colonnes (variable avec suffixe « _vector_cols_names »).

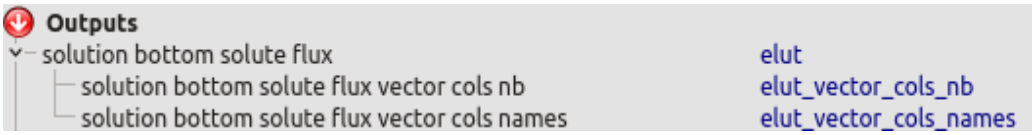


FIGURE 31 – Sortie de type « is vector »et variables associées

Parameters

Récapitulatif des paramètres saisis dans l'étape 5.

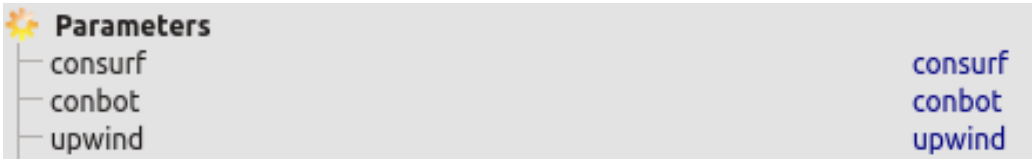


FIGURE 32 – Paramètres du module

Initial outputs

Récapitulatif des sorties initiales sélectionnées dans l'étape 4.
Si la sortie est de type distribuée dans le profil de sol, des informations complémentaires sont proposées :

- nombre de valeurs saisies sur le profil (variable avec suffixe « _depth_values_nb »)
- tableau des valeurs saisies sur le profil (variable avec suffixe «_depth »)

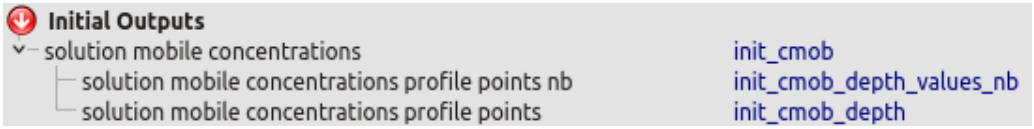


FIGURE 33 – Sortie initiale de type distribué dans le profil

External data files

Récapitulatif des fichiers externes sélectionnés dans l'étape 4.

FIGURE 34 – Fichier de données utilisateur

Les données concernant le temps.
Lorsqu'un module déclare utiliser au moins une variable de temps, il est déclaré « time dependent ». Seules les données de temps déclarées apparaissent et peuvent être sélectionnées pour utilisation dans l'édition de code.

FIGURE 35 – Données de temps

Time

Pour utiliser une variable de temps, il faut utiliser le bouton 

en haut de la zone.

FIGURE 36 – Sélection des variables de temps

Il s'agit des valeurs validées au pas de temps précédent.
Ces valeurs ne sont accessibles que si le module est « time dependent »(cad au moins une variable de temps est utilisée dans la section « Time »).

Outputs previous time

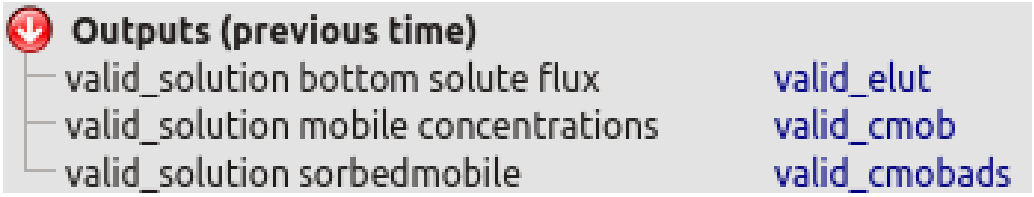
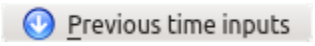


FIGURE 37 – Variables de sorties validées

Il s'agit des entrées validées au pas de temps précédent. Pour les sélectionner, le module doit avoir au moins une variable de temps utilisée, ce qui dégrise le bouton 

qui permet de sélectionner les entrées validées au pas de temps précédent que l'on souhaite.

Inputs previous time

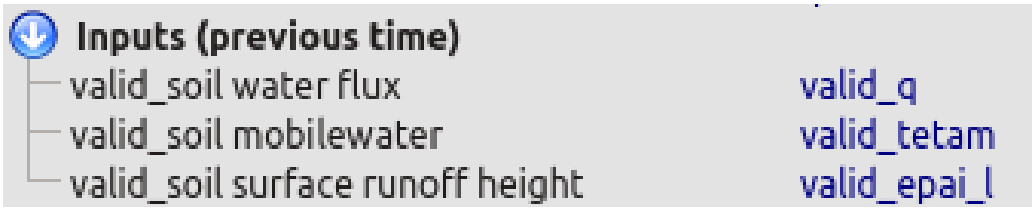


FIGURE 38 – Entrées validées au pas de temps précédent

Données de grille

Grid global constants

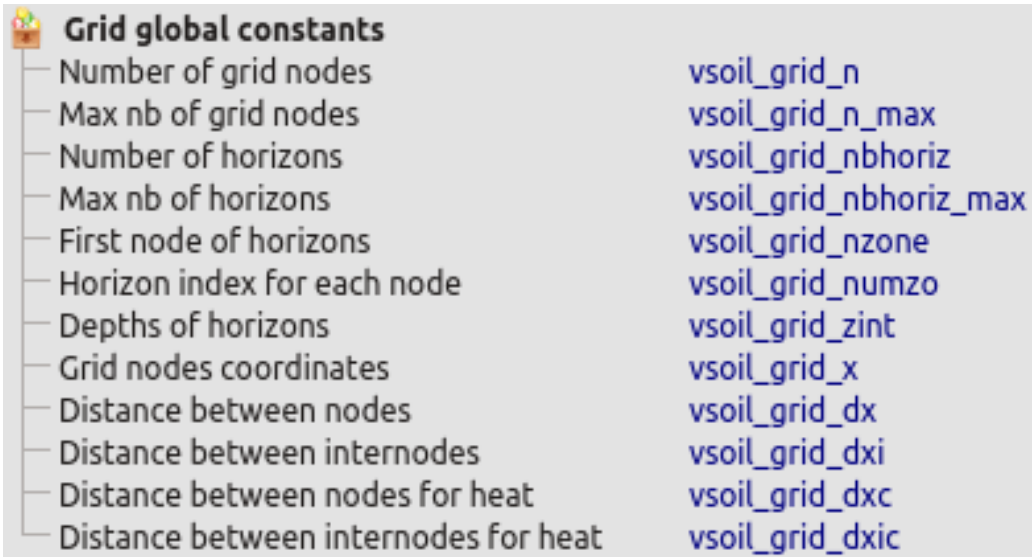


FIGURE 39 – Données de grille

Time characteristic global constants

Données concernant le temps.

FIGURE 40 – Données de temps

Universal global constants

Constantes universelles..

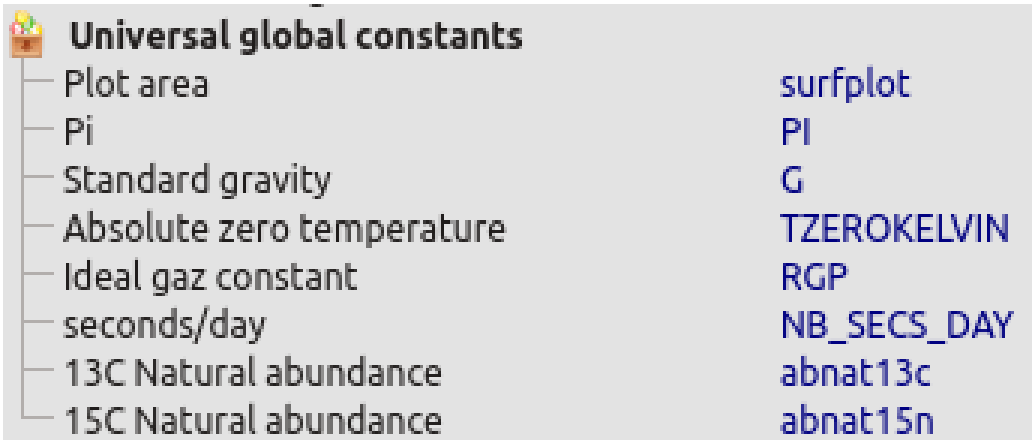


FIGURE 41 – Constantes universelles

Autres fonctions ou variables utilitaires

D'autres variables ou fonctions peuvent être proposées en fonction du langage utilisé.
Des fonctions sur les propriétés hydrauliques (si l'entrée « soil hydraulic »est sélectionnée) ou utilitaires sont par exemple proposées avec le langage Fortran.

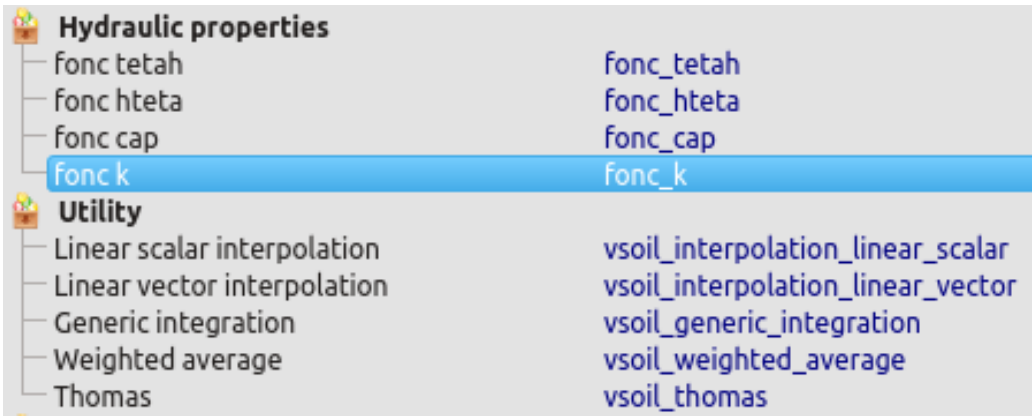


FIGURE 42 – Autres fonctions ou variables

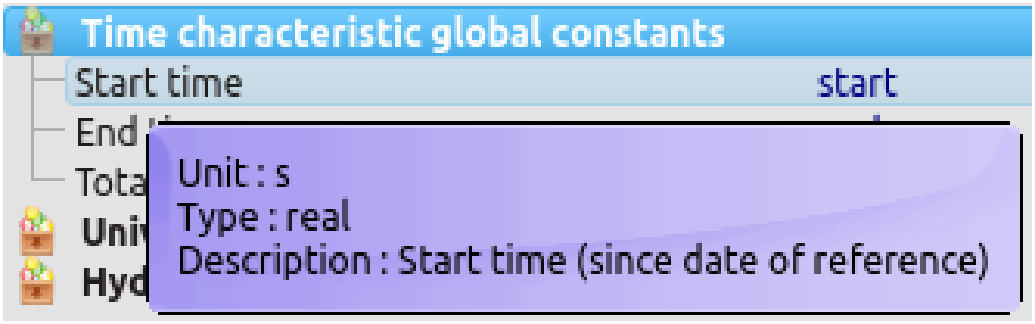


FIGURE 43 – Tooltip de la variable « Start time »

Lorsque vous laissez la souris sur un élément de l'arbre, un tooltip apporte des informations complémentaires sur la variable ou la fonction.

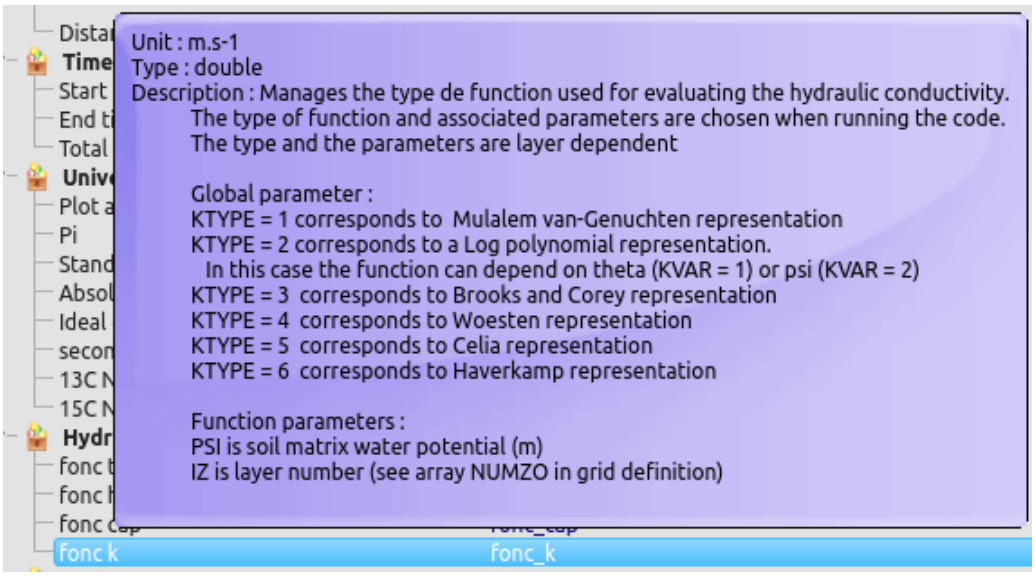


FIGURE 44 – Tooltip de la fonction fonc_k (spécifique au langage Fortran)

3.5.3 Les actions possibles

Il est possible de renommer une variable de type entrée, sortie, paramètre ou argument de temps en la sélectionnant dans l'arbre des variables globales puis en sélectionnant le bouton « Rename » situé au dessus de l'arbre des variables.
Les sorties du dernier pas de temps validé sont, quant à elles, générées automatiquement en préfixant les sorties par « valid_ »et ne sont affichées que lorqu'au moins un argument de temps a été sélectionné.
Dans l'illustration suivante, la variable d'entrée sélectionnée « soil mobilewater »est renommée en « swc ».

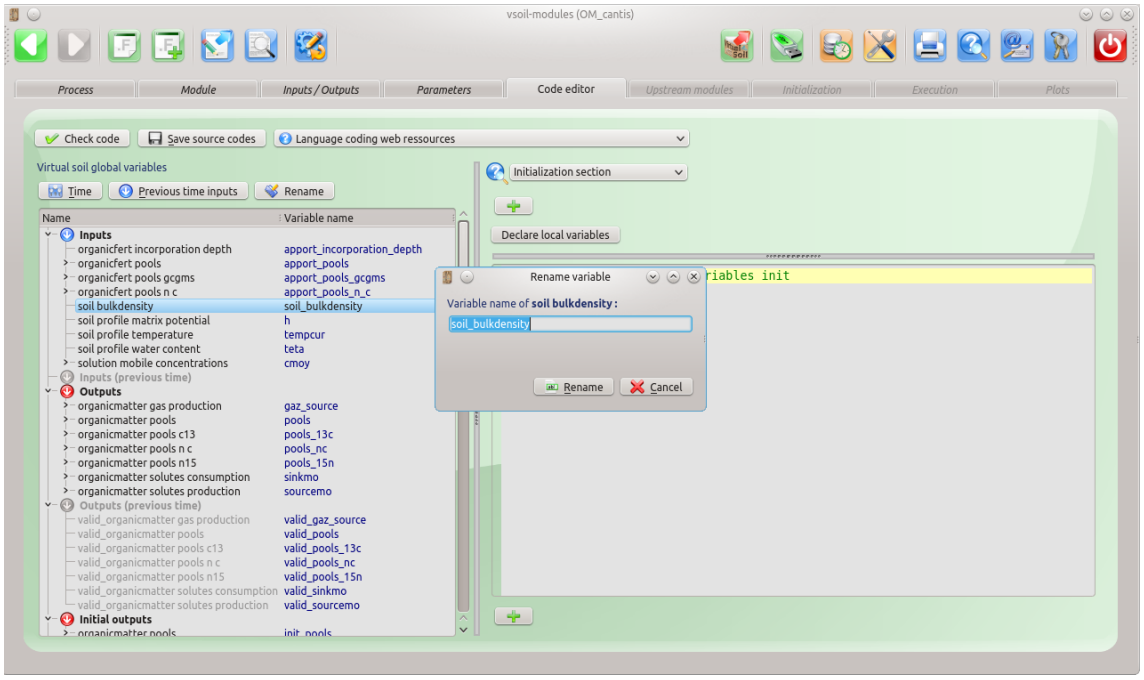


FIGURE 45 – Renommage de variable

3.5.4 Edition du code

Dans la partie droite de l'écran, vous devez sélectionner la section que vous souhaitez coder via la liste déroulante ci-contre.

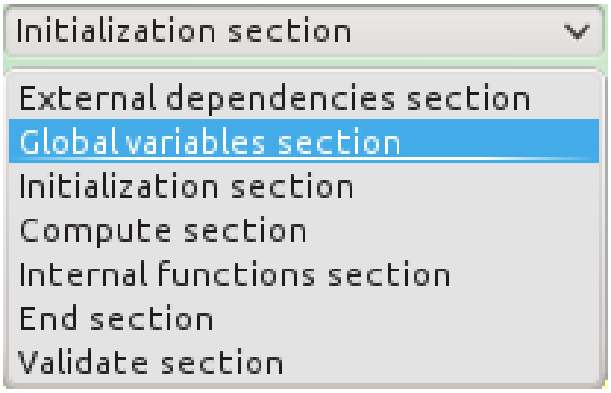
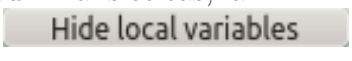


FIGURE 46 – Sélection du bloc de code à éditer

Une explication du contenu de chaque section est proposée en laissant la souris sur le bouton  située à gauche de la liste de sélection des blocs.
Les parties de code générées automatiquement par la plate-forme avant et après le code utilisateur de chaque section sont visibles avec l'icône  et cachées avec l'icône .
En fonction du langage informatique utilisé, une section peut être découpée en deux parties : déclaration des variables locales, propres à la section courante et code utilisateur. Dans ce cas, la déclaration des variables se fera avec les boutons  puis  pour cacher cette partie du code.

Les différentes sections sont décrites ci-dessous :

- **External dependencies :** Ajout des includes (C++) ou modules (Fortran) depuis vos propres fichiers sources ou depuis les fichiers/modules standards.
- **Global variables :** Ajout des variables qui seront connues dans le scope global du module.

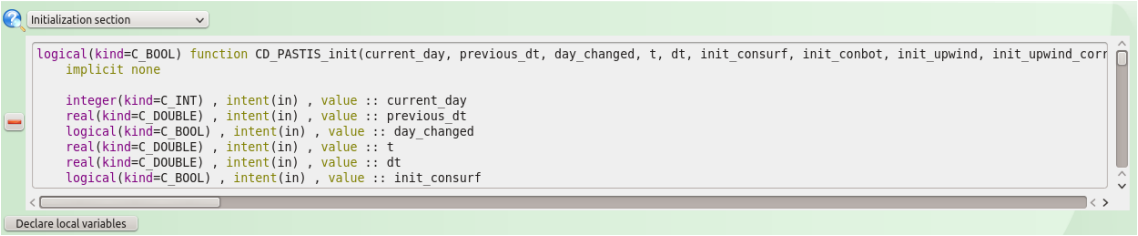


FIGURE 47 – Affichage du code généré par la plate-forme, variables locales cachées

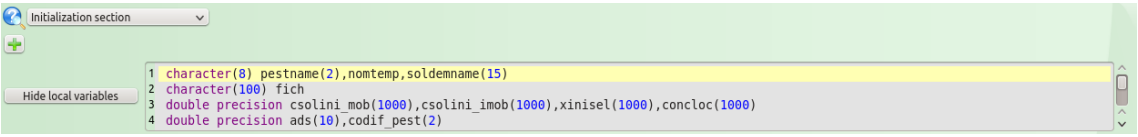


FIGURE 48 – Masquage du code généré par la plateforme, variables locales en saisie utilisateur

- **Initialization :** Partie déclaration des variables (langage Fortran) : Déclaration des variables qui ne seront connues que de la section « Initialization ». Partie code principal : Cette fonction/routine n'est appelée qu'une seule fois au début de la simulation. Son objectif est d'initialiser les variables de sortie du module. Cela signifie que toutes ces variables doivent avoir une valeur à la fin de la section. Si vous déclarez des variables dans la section « Global variables », vous devez également les initialiser. Si l'initialisation n'est pas réalisée correctement, vous aurez probablement un message d'erreur d'exécution. Cette section reçoit différentes variables : - Sorties initiales - Paramètres du module - Ces variables sont énumérées dans la partie gauche de l'écran, respectivement sous les sous-arbres : - Initial Outputs - Parameters Vous pouvez aussi utiliser des fonctions / routines définies dans la section « Internal function ».
- **Compute :** Partie déclaration des variables (langage Fortran) : Déclaration des variables qui ne seront connues que de la section « compute ». Partie code principal : Cette fonction/routine est appelée par le programme principal pour chaque pas de temps. Son objectif est de calculer au temps proposé par le programme principal toutes les variables de sortie du module. Les variables de sorties obtenues au dernier pas de temps validé sont accessibles avec le préfixe « valid_ ». Elles sont accessibles dans la partie gauche de l'écran, dans le sous-arbre « Outputs (previous time) ». Si le module dépend du temps,cette fonction/routine reçoit les variables de temps (sous arbre « Time »). Si la fonction/routine ne peut pas calculer de nouvelles valeurs pour les sorties pour le pas de temps proposé, vous devez retourner la valeur « false ». Par défaut la valeur « true »est retournée. Vous pouvez aussi utiliser des fonctions / routines définies dans la section « Internal function ».
- **Internal functions :** Cette section contient les fonctions/routines dont vous aurez besoin dans les autres section de calcul. Ces fonctions/routines peuvent utiliser les variables définies dans la section « Global variables »et peuvent être utilisées dans les sections « Compute »ou « Initialization ». Les variables d'entrée, de sorties ainsi que les paramètres du module ne sont pas directement accessibles à partir de cette section. Pour y avoir accès, vous devez les déclarer comme paramètres de fonction et les valeurs associées doivent être fournies par la méthode appelante de votre fonction.
- **End :** Cette fonction/méthode est appelée une seule fois à la fin de l'exécution du programme.
- **Validate :** Cette fonction/routine est appelée par le programme principal à la fin de

chaque pas de temps si tous les modules ont terminé correctement leur méthode « Compute ». Dans cette fonction/routine, les variables de sortie validées sont affectées avec les valeurs de sorties des mêmes variables. Cette opération est automatique pour les sorties. Pensez à faire de même pour vos variables globales si nécessaire.

Note : cette zone n'est accessible que si au moins une variable de temps est déclarée pour le module.

3.5.5 Menu contextuel d'édition de code

Dans la zone d'édition du code, un menu contextuel accessible avec le clic droit de la souris permet de faciliter le codage en proposant des fonctions de recherche, d'indentation ou de saisie automatique d'exemples de code. Les éléments proposés sont spécifiques à chaque langage.

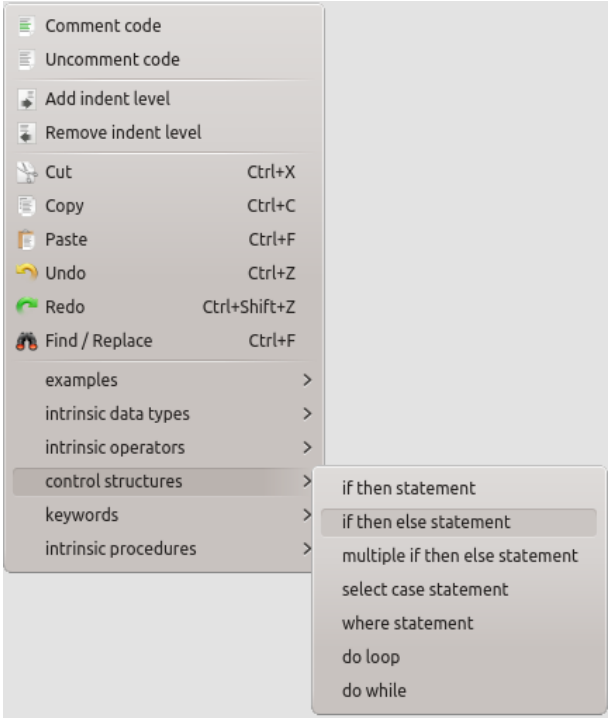


FIGURE 49 – Menu contextuel d'édition de code en Fortran

```
1 if ( logical_expression1 ) then
2 ! commands if logical_expression1 is true
3 else if ( logical_expression2 ) then
4 ! commands if logical_expression1 is false and logical_expression2 is true
5 else
6 ! commands if all logical expressions are false
7 end if
8
9 real,dimension(X) :: myvector
10 integer :: myvar
```

FIGURE 50 – Exemple de code généré en Fortran

3.5.6 gestion des variables "resumable"

Dans la section d'initialisation d'un module, le flag (variable booléenne) "resumeFlag" passé en paramètre de la fonction. Il indique le contexte d'exécution de la simulation. Si le module contient des variables de sorties "resumable" (cf. 3.3.2), il y a 2 cas possibles, selon que la simulation est en mode "resume" ou pas :

1. La simulation est en mode “resume”. Alors les variables de sorties “resumable” sont remplies automatiquement, avec les valeurs sauvegardées de la simulation rechargée. Il n’y a donc rien faire.
2. La simulation n’est pas en mode “resume”. Alors les variables de sorties “resumable” doivent être initialisées, dans la section d’initialisation du module.

Concrêtement, le principe de fonctionnement consiste à tester la variable booléenne “resumeFlag” dans la fonction d’initialisation du module. Si la simulation n’est pas en mode “resume”, alors il faut initialiser les variables de sorties “resumable”. NB : Les variables de sorties qui ne sont pas “resumable” doivent être initialisées dans tous les cas. Ci-dessous, les exemples de codes informatiques, en C++ et en Fortran :

```
if ( ! resumeFlag )
{
    a_resumable_output = a_value ;
}
```

Listing 1 – Exemple de code d’initialisation conditionnelle des variables de sorties “resumable” pour C++

```
if ( resumeFlag.eqv..false. ) then
   a_resumable_output = a_value
endif
```

Listing 2 – Exemple de code d’initialisation conditionnelle des variables de sorties “resumable” pour Fortran

3.5.7 Autres fonctionnalités

Les autres fonctionnalités sont présentées sous forme d’icônes en haut à gauche de l’application. Elles restent grisées lorsque l’utilisateur n’est pas sur l’onglet d’édition de code. Si le module édité est officiel, certaines fonctionnalités ne sont pas accessibles. Les icônes les illustrant deviennent non cliquables.

FIGURE 51 – Aperçu général des icônes

 Permet de retourner sur l’onglet précédent.
Lorsque l’on est sur le premier onglet des processus, cette icône est grisée et non active.
 Permet de valider l’onglet courant et d’afficher l’onglet suivant.
Lorsque l’on est sur l’onglet d’édition de code, cette icône est grisée et non active.
 Permet de sélectionner le langage informatique (Fortran ou C/C++). L’icône est modifiée lorsque le choix a été effectué : “.f” pour le Fortran, “.c” pour le C/C++.
 Permet d’ajouter des codes sources externes.
Les fichiers importés pourront être utilisés pour la réalisation du code du module.

 Cette icône permet de lancer un éditeur externe pour visualiser et modifier directement le code généré par l'outil.
L'édition à l'intérieur de vsoil-modules est alors bloquée, l'icône d'édition externe présente en cadenas indiquant ce changement et il n'est plus possible de modifier les variables de temps et de charger un autre module.

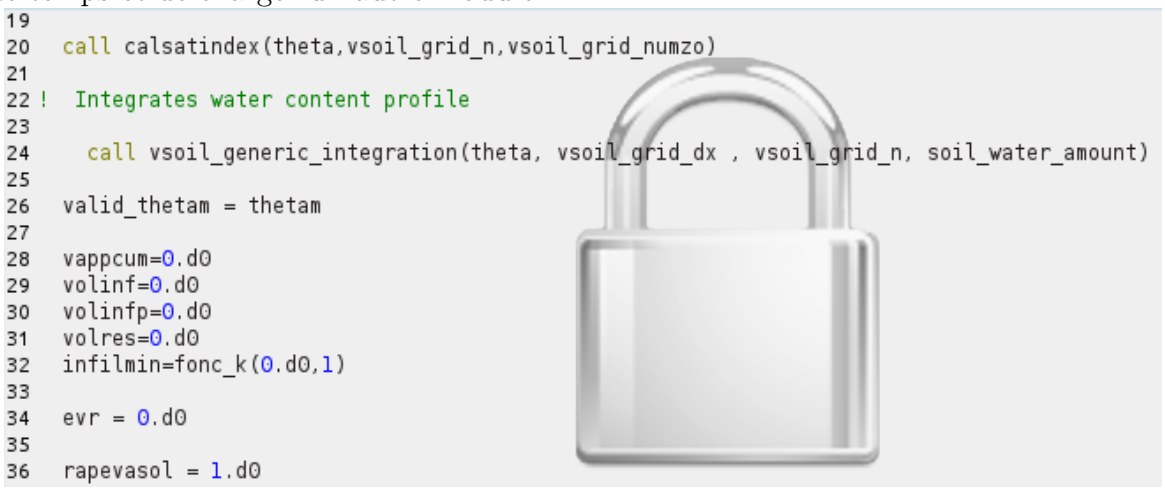


FIGURE 52 – Editeur de code verrouillé

Si l'éditeur de code par défaut du système d'exploitation n'est pas configuré, un message d'erreur apparaitra. Vous devrez alors sélectionner votre éditeur de code dans les préférences en cliquant sur l'icône  et en renseignant le chemin d'accès à l'éditeur de code pour la ligne « External editor program ».
Lorsque l'utilisateur reclique sur l'icône d'édition externe  , l'utilisateur doit avoir fermé son éditeur et, s'il confirme la fin de l'édition externe, son code sera automatiquement intégré dans vsoil-modules.

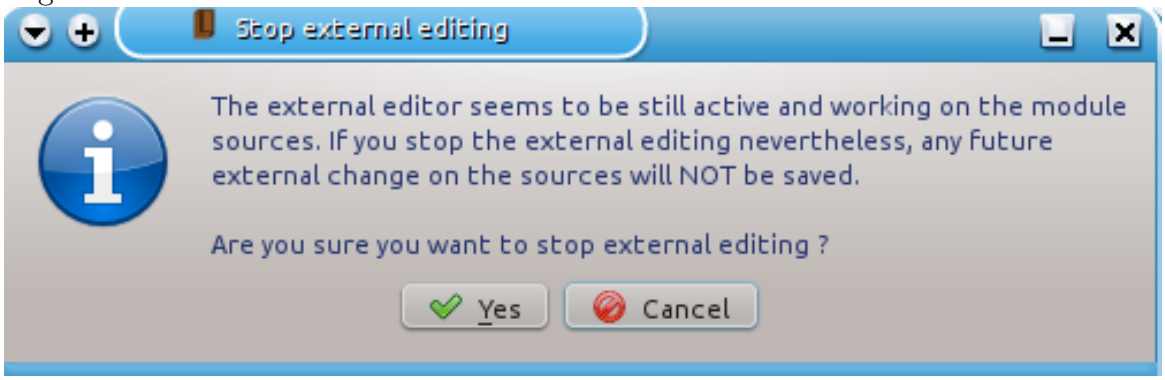


FIGURE 53 – Popup de fin d'édition de code externe

Le cadenas disparaît des volets de droite et l'édition classique peut reprendre.
 Cette icône permet d'afficher la totalité du code généré (fichier header et source) sous forme de deux onglets.

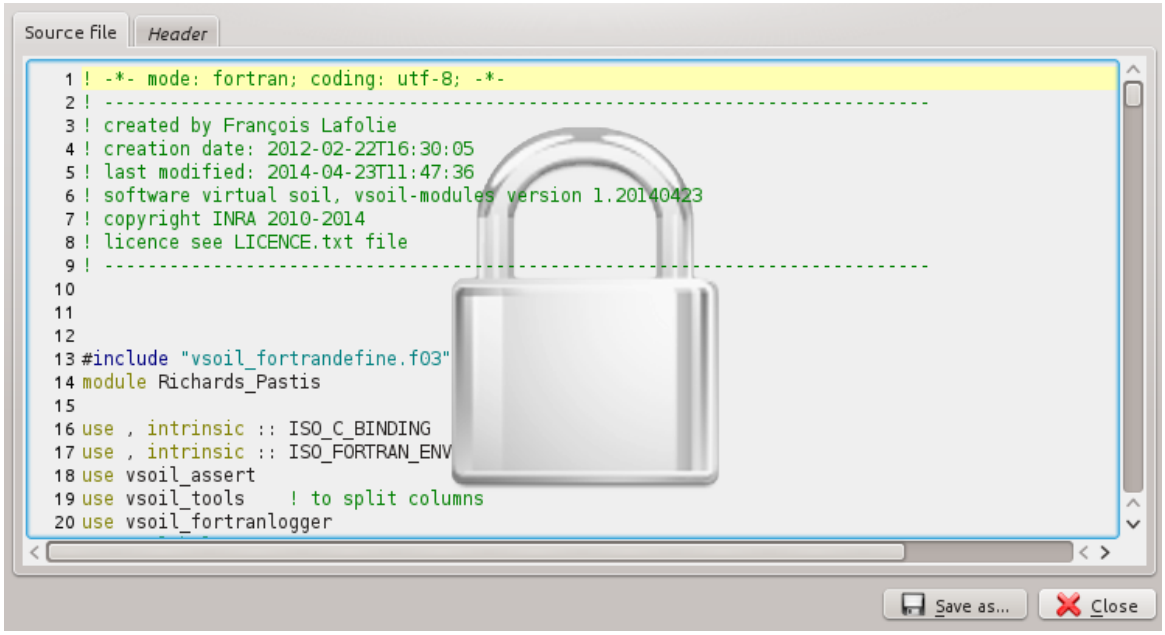


FIGURE 54 – Visualisation du code généré

La fenêtre d'affichage du code généré permet de sauvegarder chaque onglet sous forme de fichier à l'aide du bouton « Save as ».

 Permet d'éditer un autre module. Celui en cours est sauvé.

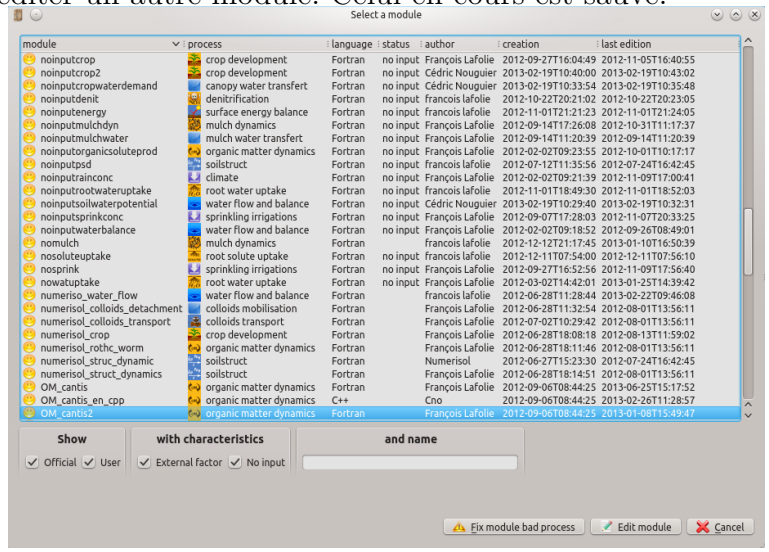


FIGURE 55 – Changement de module

3.5.8 Vérification du code

Le bouton  permet de vérifier la syntaxe du code généré par la plate-forme à partir des informations du module et des parties de code ajoutées par l'utilisateur. Un message d'erreur peut indiquer que le compilateur n'a pas pu être lancé lors de la première exécution. Dans ce cas, modifiez les préférences de compilateur Fortran et C++ en cliquant sur le bouton  puis en sélectionnant le compilateur souhaité.
Des messages d'erreur ou d'avertissement peuvent s'afficher lorsque des problèmes sont détectés par la plate-forme. Si l'erreur se trouve dans votre code, vous pouvez double cliquer sur l'erreur pour que le curseur de la souris se positionne sur celle-ci.

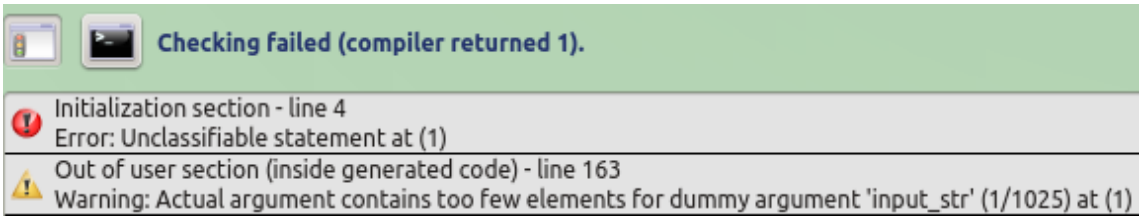


FIGURE 56 – Sortie de vérification du code

Les icônes et permettent de basculer entre un affichage simple des erreurs et un affichage de type sortie brute utile sur certains débogages.

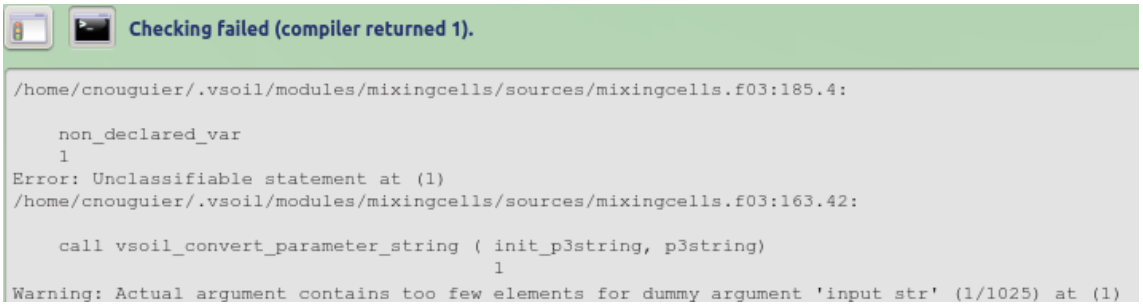


FIGURE 57 – Sortie brute de vérification du code

3.6 Étape 6 : construction du modèle

L'application module permet de tester son module seul ou avec des modules amont n'ayant pas d'entrée (modules « no input »ou modules basés sur des processus de type « external factor »).

Lorsque vous avez vérifié votre code avec succès avec le bouton Check code de l'onglet Code editor, votre module est à présent opérationnel, vous entrez dans la phase de test. Pour cela, vous accédez à l'onglet Upstream modules qui permet de compiler la solution. La sélection du bouton de compilation se fera en fonction de votre besoin :

- test unitaire sans module amont avec le bouton Compile module with upstreams
- test avec modules amont simples avec le bouton Compile standalone module

Si vous souhaitez tester votre module avec des modules amonts, sélectionnez un module qui fournira une valeur pour chaque entrée de votre module. Si aucun n'existe, le bouton Create no input module vous permet de créer un module amont vide que vous devrez compléter (en ajoutant le code nécessaire au module pour fonctionner).

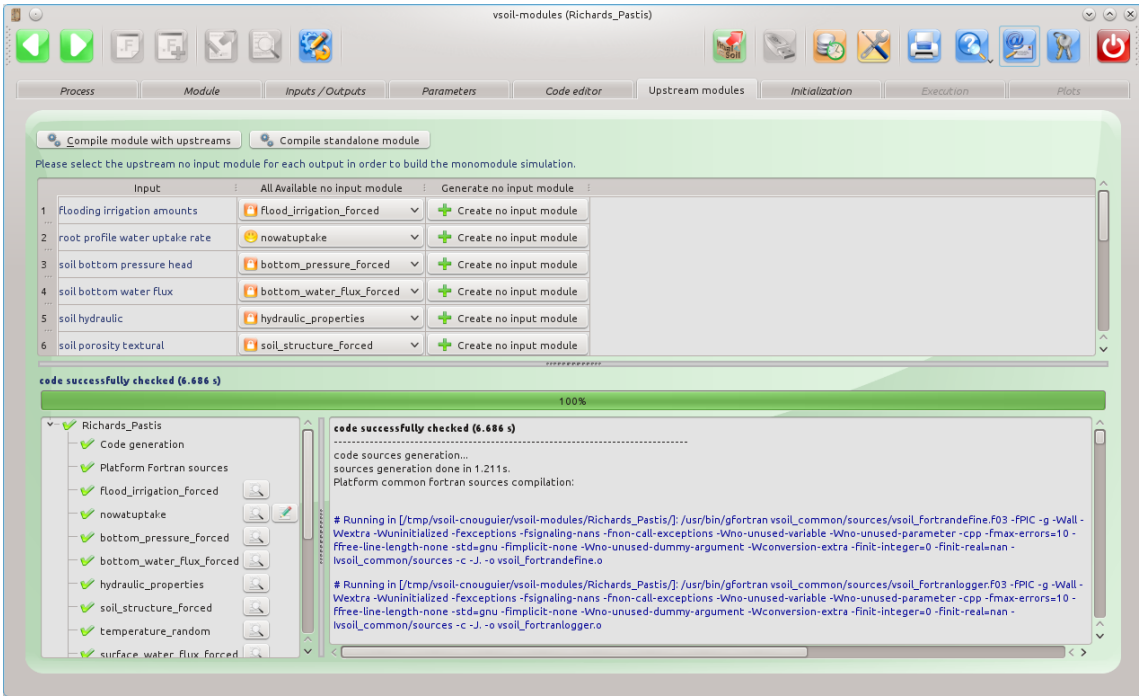


FIGURE 58 – Étape 6, aperçu général de l'onglet « Upstream modules »

Une barre de progression indique la progression de la compilation et la sortie du compilateur s'affiche en bas de l'écran. Si une erreur est détectée, la barre de progression devient rouge et vous devez corriger le code du module en erreur (étape 5), le vérifier avant de relancer la compilation avec les modules amonts.

Il est possible d'afficher le code source d'un des modules constituant le modèle avec le bouton  .

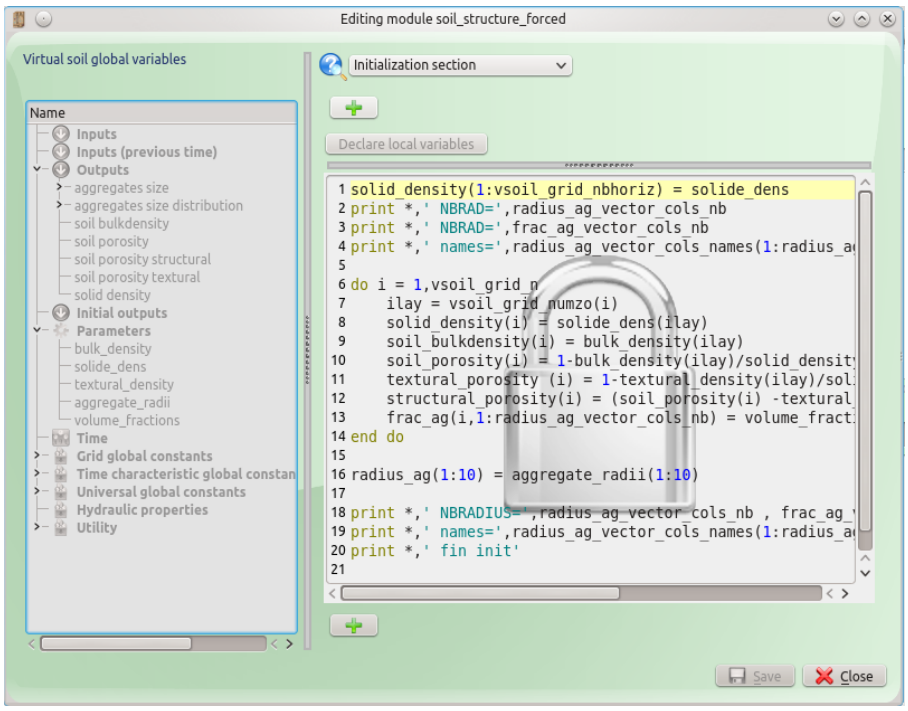


FIGURE 59 – Fenêtre de visualisation du code d'un module

Lorsqu'un module du modèle n'est pas officiel, il est possible de modifier son code en cliquant sur le bouton  .

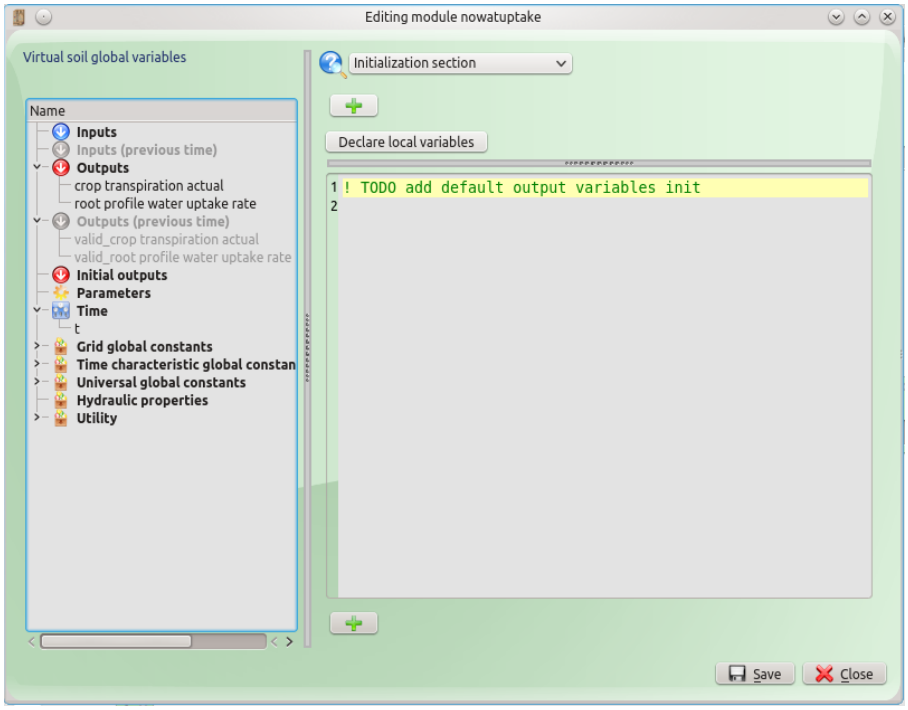


FIGURE 60 – Fenêtre d'édition du code d'un module

3.7 Étape 7 : paramétrisation du modèle

Lorsque la solution est compilée avec succès, vous accédez à l'onglet de paramétrisation.

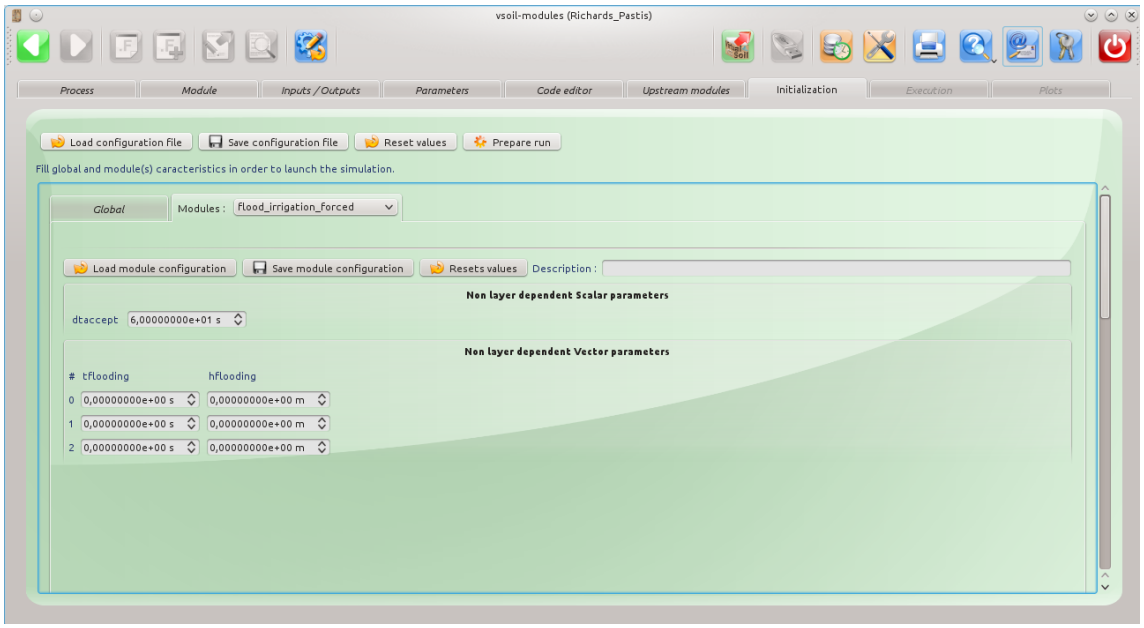


FIGURE 61 – Étape 7, aperçu général de l'onglet « Initialization »

Lorsque l'onglet général est correctement paramétré, vous devez renseigner les valeurs des paramètres et des sorties initiales de chaque module utilisé dans la solution (module courant + modules amonts si vous avez sélectionné dans l'étape précédente et que vous avez compilé la solution avec le bouton « Compile module with upstreams »).

Vous devez générer le fichier de simulation avec le bouton Prepare run. Si une erreur est détectée, une fenêtre vous alertera des raisons de l'échec de génération du fichier de simulation. Dans le cas ci-dessous, tous les noms des colonnes des sorties de type « is vector »ne sont pas renseignés (sortie « soil mobile solution concentration »). L'onglet représentant le module en erreur voit son titre mis en rouge.

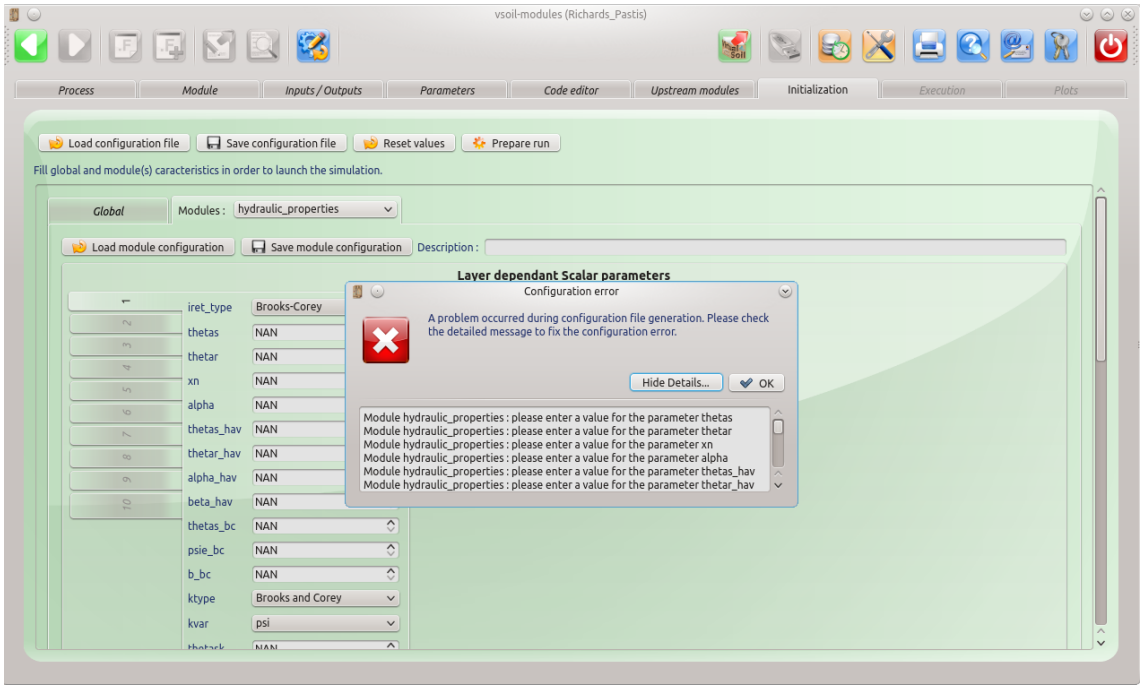


FIGURE 62 – Exemple d'erreur de saisie de données

Il est possible d'enregistrer le fichier de paramétrage de simulation en cliquant sur le bouton Save configuration file puis de recharger l'interface graphique avec les valeurs contenues dans le fichier sauvegardé avec le bouton Load configuration file. Vous pouvez à tout moment restaurer l'état initial des valeurs des différents onglets de paramétrage en appuyant sur le bouton Reset values.
Après un rechargement des données de simulation depuis un fichier, vous devez regénérer le fichier de simulation avec le bouton Prepare run pour que les modifications soient prises en compte.

3.8 Étape 8 : lancement de la simulation

Après génération du fichier de paramétrisation de la solution, vous accédez à l'avant-dernière étape, l'onglet Execution.

Appuyez sur le bouton Prepare run pour lancer la simulation. Un affichage de la sortie standard sera mis à jour en temps réel dans la fenêtre du haut « Execution output ».
Dans la partie basse, lorsque la simulation sera terminée, un tableau affichera les résultats de la sortie du module sélectionné. Vous pourrez consulter la sortie du module souhaité en le sélectionnant dans la liste de choix des modules puis dans celle des sorties de ce module. Il est possible d'exporter les données avec le séparateur de son choix ou d'accéder directement au répertoire contenant les fichiers de sortie des modules avec le bouton Output directory.

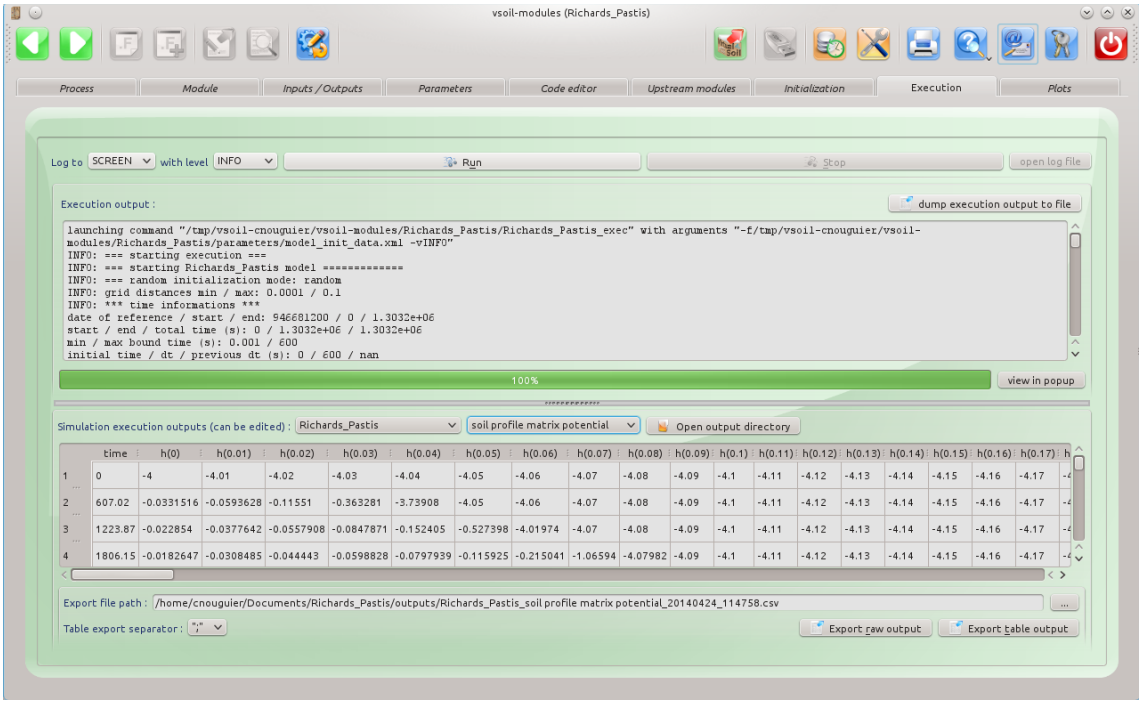


FIGURE 63 – Étape 8, aperçu général de l'onglet « Exécution »

3.9 Fonctionnalités générales

3.9.1 La sauvegarde des données

Si le système détecte que vous n'avez pas sauvé vos modifications et que vous souhaitez quitter l'application alors le message suivant apparaît avec la liste des onglets à valider :

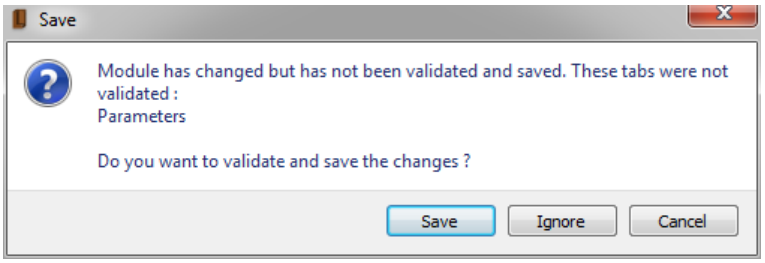


FIGURE 64 – Message alertant que les données de certains onglets ne sont pas sauvegardés

3.9.2 Le panneau de configuration

Dans la barre d'outil (partie 1), l'icône permet de configurer certains outils comme les éditeurs de code externe ou les compilateurs.

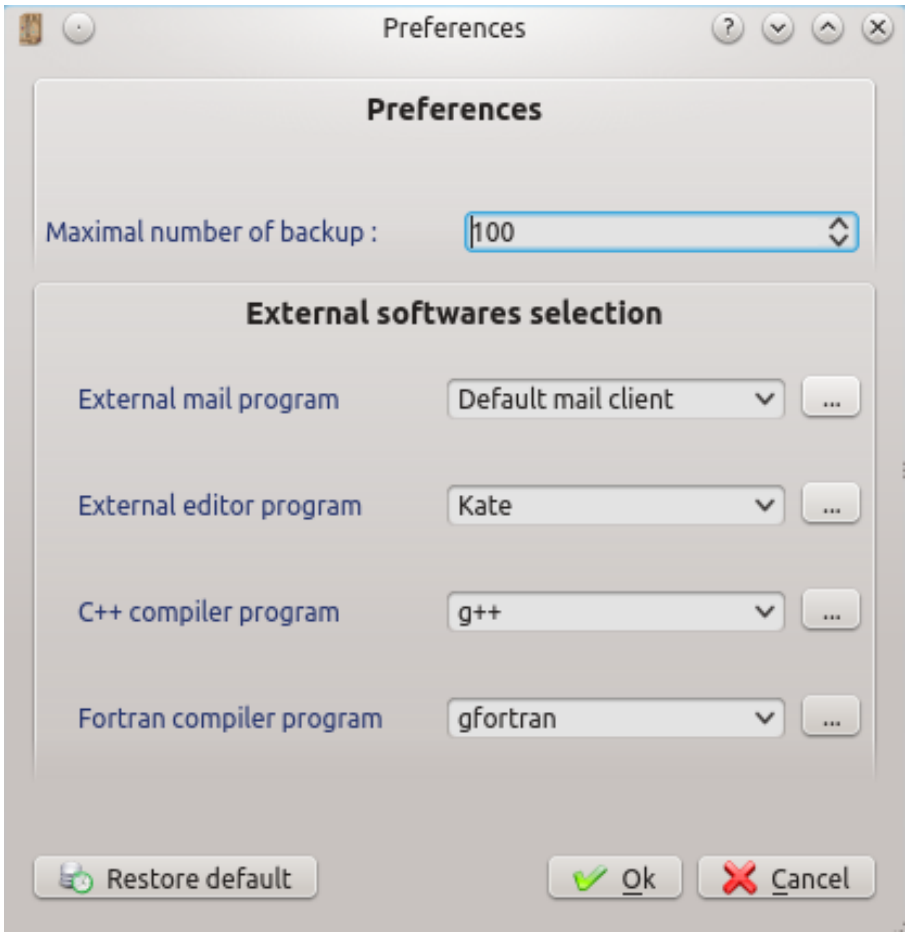


FIGURE 65 – Panneau de configuration