



**HAL**  
open science

**PESHMELBA. PESTicides et Hydrologie: Modélisation à l'échelle du Bassin versant. Développement d'une modélisation pour évaluer l'efficacité cumulée des zones tampons dans un bassin versant vis-à-vis de l'atténuation des transferts de pesticides en territoire rural**

Nadia Carluer, Emilie Rouzies, C. Baracher, Claire Lauvernet

► **To cite this version:**

Nadia Carluer, Emilie Rouzies, C. Baracher, Claire Lauvernet. PESHMELBA. PESTicides et Hydrologie: Modélisation à l'échelle du Bassin versant. Développement d'une modélisation pour évaluer l'efficacité cumulée des zones tampons dans un bassin versant vis-à-vis de l'atténuation des transferts de pesticides en territoire rural. [Rapport de recherche] irstea. 2019, pp.257. hal-02609861

**HAL Id: hal-02609861**

**<https://hal.inrae.fr/hal-02609861>**

Submitted on 16 May 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**AGENCE FRANÇAISE  
POUR LA BIODIVERSITÉ**

*Établissement public du ministère de l'Environnement*

Programme 2016/2018 – Thème XXXX - Action n° 58

## **PESHMELBA. PESTicides et Hydrologie : Modélisation à l'échelle du Bassin versant**

**Développement d'une modélisation pour  
évaluer l'efficacité cumulée des zones  
tampons dans un bassin versant vis-à-vis  
de l'atténuation des transferts de  
pesticides en territoire rural**

**Rapport final**

**Nadia CARLUER, Emilie ROUZIES, Christine  
BARACHET, Claire LAUVERNET (Irstea)**

**Février 2019**

Document élaboré dans le cadre de la convention cadre Irstea-AFB.:



- **AUTEURS**

**Nadia CARLUER**, ingénieur-chercheur (Irstea), [nadia.carluer@irstea.fr](mailto:nadia.carluer@irstea.fr)

**Emilie ROUZIES**, ingénieur de recherche en modélisation (Irstea), [emilie.rouzies@irstea.fr](mailto:emilie.rouzies@irstea.fr)

**Christine BARACHET**, ingénieur d'étude en géomatique (Irstea), [christine.barachet@irstea.fr](mailto:christine.barachet@irstea.fr)

**Claire LAUVERNET**, chargée de recherche en modélisation (Irstea), [claire.lauvernet@irstea.fr](mailto:claire.lauvernet@irstea.fr)

- **CORRESPONDANTS**

**AFB : Claire BILLY**, chargée de mission « Agriculture et protection des milieux » (AFB), [claire.billy@afbiodiversite.fr](mailto:claire.billy@afbiodiversite.fr)

I

- **AUTRES CONTRIBUTEURS**

**Droits d'usage** : accès libre

**Niveau géographique** : national

**Couverture géographique** : petit bassin versant

**Niveau de lecture** : Professionnels, experts



## • RESUME

Ce rapport présente l'état actuel de développement du modèle PESHMELBA. Ce modèle a pour objectifs de permettre d'évaluer l'efficacité cumulée des zones tampons vis-à-vis de l'atténuation des transferts de pesticides dans un bassin versant rural.

Les transferts de pesticides sont fortement influencés par les discontinuités qui peuvent accélérer ou limiter leurs transferts hydriques, comme les zones tampons enherbées, les fossés, haies, talus, zones tampons humides artificielles ou routes. Les prendre en compte dans une modélisation à l'échelle du petit bassin versant est donc nécessaire, et si leur influence à l'échelle locale de l'élément du paysage a été étudiée et formalisée, les modèles intégrant l'ensemble de ces structures à l'échelle du bassin versant manquent encore.

Dans ce contexte, PESHMELBA est développé pour représenter explicitement l'influence de l'organisation du paysage sur le devenir des pesticides, en prenant en compte les caractéristiques et la position des éléments dans le bassin versant. L'objectif final est de permettre de tester et hiérarchiser différents scénarios d'aménagements (occupation du sol, pratiques agricoles, implantation de zones tampons ou autres éléments du paysage) compte tenu de leur influence sur la qualité de l'eau à l'échelle du petit bassin versant. Un tel outil doit faciliter le dialogue entre les différents acteurs impliqués sur le terrain, et favoriser l'adoption de solutions optimisées à l'échelle du bassin versant.

La structure de modélisation adoptée est une structure modulaire, qui permet de représenter chaque élément du bassin versant (surfacique ou linéaire) par un modèle indépendant, avec le degré de complexité souhaité (susceptible d'évoluer au fil de l'acquisition de connaissances), tout en couplant ces éléments entre eux de façon dynamique. Les processus dominant les transferts d'eau et le devenir des pesticides sont représentés pour chaque type d'élément, soit par un modèle pré-existant, soit par un modèle développé/adapté pour les besoins de PESHMELBA. L'ensemble est couplé pour obtenir un modèle spatialisé à l'échelle du bassin versant grâce au coupleur dynamique Open-PALM (Fouilloux et Piacentini, 1999), développé au CERFACS, qui permet à la fois un couplage dynamique et du calcul parallèle.

Le modèle intègre pour l'instant les écoulements verticaux (variablement saturés) et latéraux (saturés) dans le sol, le ruissellement de surface sur les parcelles, les transferts dans les fossés et les cours d'eau, ainsi que les processus conjoints de transfert (convection-diffusion) et dissipation (adsorption-dégradation) des pesticides. Les zones tampons que sont les bandes enherbées et les haies ou haies sur talus ont également été implémentées. La représentation du devenir d'un métabolite par substance active est également possible. Le modèle est développé en se basant sur les processus et ordres de grandeur représentatifs du bassin de la Fontaine du Theil (35), qui sera le premier bassin versant sur lequel le modèle sera validé, avant celui de la Morcille dans le Beaujolais (69).

En amont de PESHMELBA a été développé GeoMELBA, un outil SIG qui fournit à PESHMELBA un maillage décrivant la configuration du paysage. Le découpage du bassin détermine des éléments homogènes et distingue s'ils sont surfaciques (parcelles, bandes enherbées) ou linéaires (talus, fossés, routes, haies). Il détermine ensuite les connexions hydrologiques entre les éléments, permettant de transférer de façon adéquate les flux simulés par le modèle sur chaque élément. GeoMELBA, outre le pre-processing qu'il donne en entrée à PESHMELBA, permet donc également de réaliser un premier diagnostic des zones à risque de transfert sur le bassin, en amont de la modélisation.

## • MOTS CLES (THEMATIQUE ET GEOGRAPHIQUE)

Modélisation hydrologique spatialisée, pesticides, éléments du paysage, OpenPALM, bassin versant, connexions hydrologiques, modularité



PESHMELBA. PESTicides et  
hydrologie : Modélisation à l'échelle  
du Bassin versant

Rapport final

N. Carlier, E. Rouzies, C. Barachet, C.  
Lauvernet

AGENCE FRANÇAISE  
POUR LA BIODIVERSITÉ

*Établissement public du ministère de l'Environnement*

## **PESH-MELBA: MODELING WATER CIRCULATION AND PESTICIDES FATE AT THE CATCHMENT SCALE BY TAKING INTO ACCOUNT THE INFLUENCE OF LANDSCAPE ELEMENTS**

### **• ABSTRACT**

Pesticide transfers are highly influenced by the presence of discontinuities that can accelerate or slow down and dissipate water and contaminant flows, such as grass strips, ditches, hedgerows or roads. It is thus necessary to take into account landscape features when modeling water and contaminant transfers at the small catchment scale. However, if influence of landscape elements has already been widely explored at field scale, models generally don't reach the catchment scale.

The project PESHMELBA aims at developing a modeling tool of water and contaminants circulation and fate at the scale of small catchments. Furthermore, the model explicitly takes into account spatial organization of landscapes by representing the existing features, their space location and shape. The final aim of the modelling tool is to test different development scenarios in order to assess influence of agricultural/landscape management practices on water quality at the small catchment scale.

Dominating processes ruling water and contaminants circulation for each element type are represented by existing or newly developed models. Those models present different levels of conceptualization and are used as modelling units ensuring a modular structure. In order to implement the spatial and temporal couplings, the different units are gathered and connected in the OpenPALM coupler (Fouilloux and Piacentini, 1999) to get a spatialized model of the whole catchment.

A special attention is paid to design a mesh representative of the landscape configuration. Soil types and land uses are overlaid to deduce surface homogeneous units. Linear elements (ditches, slopes, hedgerows...) are also included in the mesh. In addition to shaping the area of application of each modeling unit, this method also aims to define connectivity between landscape elements. Preliminary results were achieved by considering a hillslope with several plots and ditches. They showed this method is promising to represent pesticides fate in a complex landscape. Currently, the model is considering plots, vegetated zones, ditches and rivers, hedges and hedges on banks. This model will first be applied on La Fontaine du Theil (35) and La Morcille (69) catchments

### **• KEY WORDS (THEMATIC AND GEOGRAPHICAL AREA)**

Spatialized hydrological model, pesticides, landscape elements, OpenPALM, catchment scale, hydrological connections



- [Cette synthèse pourra être utilisée indépendamment du rapport final, elle doit donc être utilisable telle quelle]
- [Cette synthèse est référencée en tant que telle sur le site de l'AFB et sur le portail documentaire national « Les documents techniques sur l'eau ».]
- [Cette synthèse de vulgarisation montre les résultats et leurs utilités (potentialités) pour la gestion ou la décision ; des figures, tableaux peuvent y être insérés. Les objectifs rappelés dans les fiches action de la convention peuvent servir de base pour cette rédaction.]

- **SYNTHESE POUR L'ACTION OPERATIONNELLE** [4 pages maximum]

Le modèle PESHMELBA a pour objectifs de permettre d'évaluer l'efficacité cumulée des zones tampons vis-à-vis de l'atténuation des transferts de pesticides dans un bassin versant rural. Il s'agit de pouvoir tester et hiérarchiser différents scénarios d'aménagements (occupation du sol, pratiques agricoles, implantation de zones tampons ou autres éléments du paysage) compte tenu de leur influence sur la qualité de l'eau, à l'échelle d'un petit bassin versant. Un tel outil doit faciliter le dialogue entre les différents acteurs impliqués sur le terrain, et favoriser l'adoption de solutions optimisées à l'échelle du bassin versant.

PESHMELBA est développé pour représenter explicitement l'influence de l'organisation du paysage sur le devenir des pesticides, en prenant en compte les caractéristiques et la position des éléments dans le bassin versant. En effet, les transferts de pesticides sont fortement influencés par les discontinuités qui peuvent accélérer ou limiter leurs transferts hydriques, comme les zones tampons enherbées, les fossés, haies, talus, zones tampons humides artificielles ou routes. Les prendre en compte dans une modélisation à l'échelle du petit bassin versant est donc nécessaire, et si leur influence à l'échelle locale de l'élément du paysage a été étudiée et formalisée, les modèles intégrant l'ensemble de ces structures à l'échelle du bassin versant manquent encore.

La structure de modélisation adoptée est une structure modulaire, qui permet de représenter chaque élément du bassin versant (surfacique ou linéaire) par un modèle indépendant, avec le degré de complexité souhaité (susceptible d'évoluer au fil de l'acquisition de connaissances), tout en couplant ces éléments entre eux de façon dynamique. Les processus dominant les transferts d'eau et le devenir des pesticides sont représentés pour chaque type d'élément, soit par un modèle pré-existant, soit par un modèle développé/adapté pour les besoins de PESHMELBA. L'ensemble est couplé pour obtenir un modèle spatialisé à l'échelle du bassin versant grâce au coupleur dynamique Open-PALM (Fouilloux et Piacentini, 1999), développé au CERFACS, qui permet à la fois un couplage dynamique et du calcul parallèle.

GeoMELBA, un outil SIG qui fournit à PESHMELBA un maillage décrivant la configuration du paysage a été développé en amont de PESHMELBA. Le découpage du bassin détermine des éléments homogènes et distingue s'ils sont surfaciques (parcelles, bandes enherbées) ou linéaires (talus, fossés, routes, haies). Il détermine ensuite les connexions hydrologiques entre les éléments, permettant de transférer de façon adéquate les flux simulés par le modèle sur chaque élément. Cet outil demande encore à être consolidé, il pourrait alors permettre de réaliser un premier diagnostic des zones à risque de transfert sur le bassin, en amont de la modélisation. Ce pourrait ainsi être un complément utile à PESHMELBA « entier » pour élaborer et pré-trier les scénarios de solutions correctives à tester sur un bassin versant, dans un processus de co-construction avec les acteurs de terrain.

Le modèle représente pour l'instant les parcelles agricoles, les zones tampons enherbées, les haies et haies sur talus, ainsi que les fossés et rivières. Pour chacun de ces types d'éléments du paysage, sont représentés les processus considérés comme dominant le transfert et le devenir des pesticides. Par exemple :

- pour les parcelles les écoulements verticaux (variablement saturés) et latéraux (saturés) dans le sol, le ruissellement de surface, ainsi que les processus conjoints de transfert (Convection-diffusion) et dissipation (adsorption-dégradation) des pesticides.
- pour les fossés et cours d'eau, les transferts par convection (onde cinématique), l'adsorption sur le substrat, la dégradation dans la lame d'eau

- pour les haies, l'évapotranspiration accrue asséchant la zone d'influence de la haie, une infiltration matricielle importante liée à la structuration du sol, l'infiltration préférentielle des pesticides favorisée par le réseau racinaire, l'écoulement de surface possible le long du linéaire de haie
- pour les zones tampons enherbées, une infiltration accrue au sein de la zone, liée à la présence de végétation pérenne favorisant une perméabilité élevée du sol, et une forte teneur en matière organique favorisant l'adsorption des substances.

La représentation de ces éléments permet d'ores et déjà d'envisager l'application de PESHMELBA sur le bassin breton, en polyculture élevage, de la Fontaine du Theil (35). D'autres éléments, comme les zones tampons humides artificielles, ou les parcelles drainées par tuyaux enterrés restent à intégrer pour pouvoir aborder des types de bassins versants plus diversifiés. De par sa conception, le modèle est toutefois essentiellement pertinent pour les bassins sur socle peu perméable, où la ressource en eau est superficielle et vulnérable, et où écoulements de surface et de proche subsurface interagissent fortement, ne permettant pas de découpler la représentation du ruissellement et des transferts advenant en subsurface.

Le modèle est actuellement capable de représenter conjointement le devenir de plusieurs substances actives et d'un ou plusieurs métabolites pour chacune de ces substances. Chacun de ces éléments chimiques est caractérisé par un coefficient d'adsorption ( $K_{oc}$ ), une demie-vie ( $DT_{50}$ ) et, pour les substances actives, la part de chacun de ses métabolites créée quand elle se dégrade (une partie pouvant être totalement minéralisée ou supposer « disparaître »). Substances actives et produits de dégradation sont supposés être soumis aux mêmes processus (adsorption selon l'isotherme de Freundlich et dégradation du premier ordre), si ce n'est que les produits de dégradation, une fois dégradés, disparaissent. Des tests ont été menés sur des versants constitués de quelques parcelles, fossés et tronçons de réseau hydrographique, pour l'isoproturon, un de ses métabolites, et du diflufénicanil (Rouzies et al 2019). Le réalisme des résultats montre la pertinence de l'approche et justifie le soin apporté à la représentation de l'influence des éléments du paysage.

La représentation spatialement explicite des éléments du paysage permet par ailleurs d'envisager de représenter de façon couplée, à terme, la connectivité de ces éléments du point de vue de son influence sur la biodiversité.

## SOMMAIRE

<b>Introduction</b> .....	9
<b>Hypothèses et choix pour le développement du modèle</b> .....	9
<b>GeoMELBA : discrétisation du bassin en éléments homogènes et détermination des connexions hydrologiques</b> .....	10
<b>Développement de PESHMELBA</b> .....	12
Représentations adoptées pour les éléments surfaciques.....	13
Représentations adoptées pour les éléments linéaires.....	15
Représentations adoptées pour les interfaces .....	15
Couplage entre les éléments, gestion du temps .....	16
Applications du modèle. Perspectives immédiates .....	16
<b>Conclusion. Perspectives</b> .....	17
Bibliographie.....	18
Table des illustrations.....	18
<b>Annexe 1 : Rapport technique sur le développement de PESH-MELBA</b> .....	19
Annexe 2 : Article paru dans la revue STOTEN (Science of the Total Environment) .....	19

# **PESHMELBA. PESTICIDES ET HYDROLOGIE : MODELISATION A L'ECHELLE DU BASSIN VERSANT - DEVELOPPEMENT D'UNE MODELISATION POUR EVALUER L'EFFICACITE CUMULEE DES ZONES TAMPONS DANS UN BASSIN VERSANT VIS-A-VIS DE L'ATTENUATION DES TRANSFERTS DE PESTICIDES EN TERRITOIRE RURAL**

## **Introduction**

Ce rapport présente l'état actuel de développement du modèle PESHMELBA. Ce modèle a pour objectifs de permettre d'évaluer l'efficacité cumulée des zones tampons vis-à-vis de l'atténuation des transferts de pesticides dans un bassin versant rural.

Les transferts de pesticides sont fortement influencés par les discontinuités qui peuvent accélérer ou limiter leurs transferts hydriques, comme les zones tampons enherbées, les fossés, haies, talus ou routes. Les prendre en compte dans une modélisation à l'échelle du petit bassin versant est donc nécessaire, et si leur influence à l'échelle locale de l'élément du paysage a été étudiée et formalisée, les modèles intégrant l'ensemble de ces structures à l'échelle du bassin versant manquent encore.

Dans ce contexte, PESHMELBA est développé pour représenter explicitement l'influence de l'organisation du paysage sur le devenir des pesticides, en prenant en compte leurs caractéristiques et position dans le bassin versant. L'objectif final est de permettre de tester et hiérarchiser différents scénarios d'aménagements (occupation du sol, pratiques agricoles, implantation de zones tampons ou autres éléments du paysage) compte tenu de leur influence sur la qualité de l'eau à l'échelle du petit bassin versant. Un tel outil doit faciliter le dialogue entre les différents acteurs impliqués sur le terrain, et favoriser l'adoption de solutions optimisées à l'échelle du bassin versant.

Le rapport présente tout d'abord les hypothèses et choix ayant guidé le développement du modèle (§2), il présente ensuite GeoMELBA, qui assure la discrétisation du bassin en éléments homogènes et permet de déterminer les connexions hydrologiques entre éléments (§3). Le §4 présente très rapidement l'état d'avancement du développement de PESHMELBA, celui-ci étant largement détaillé dans le rapport technique joint en annexe I (peshmelba\_20182019.pdf). Il aborde ensuite les perspectives envisagées pour le développement de ce modèle (§5).

## **Hypothèses et choix pour le développement du modèle**

Une des hypothèses fondamentales sous-tendant le développement de PESHMELBA est que représenter les processus dominant le devenir et le transfert hydriques des pesticides à l'échelle du bassin versant, même de façon simplifiée, permettra de progresser dans la compréhension du fonctionnement du système « bassin versant », et donnera des clés pour agir de façon pertinente pour limiter les pollutions de la ressource en eau de surface par les pesticides (changement de pratiques ou d'occupation du sol, mise en place d'éléments du paysage ou aménagement d'éléments préexistants). On s'attache donc à atteindre l'échelle du bassin versant, éventuellement en adaptant pour certains éléments des représentations simplifiées, quitte à regagner en complexité par la suite.

L'échelle visée est ici celle du petit bassin versant (quelques km<sup>2</sup>) et la ressource en eau concernée est superficielle. PESHMELBA a été développé en particulier pour les bassins sur socle peu perméable et peu profond, où le milieu aquatique est vulnérable, et où ruissellement et écoulements de subsurface proche interagissent fortement, et doivent être représentés de façon conjointe. De surcroit, sur ces bassins aux transferts superficiels ou peu profonds, l'influence des éléments du paysage est particulièrement sensible. Les deux bassins servant de référence pour le développement du modèle sont la Fontaine du Theil, bassin de 1 km<sup>2</sup> en polyculture élevage au nord de Rennes (35), suivi par ARVALIS-Institut du Végétal et l'Union des Industries de Protection de Plantes de 1996 à 2006, et le bassin de la Morcille, bassin très majoritairement viticole dans le Beaujolais de coteaux (69), site-atelier bassin du Rhône suivi par Irstea Lyon-Villeurbanne. PESHMELBA adopte une représentation des processus cohérente avec la connaissance que l'on a des processus dominant le transfert et le devenir des pesticides sur ces deux bassins.

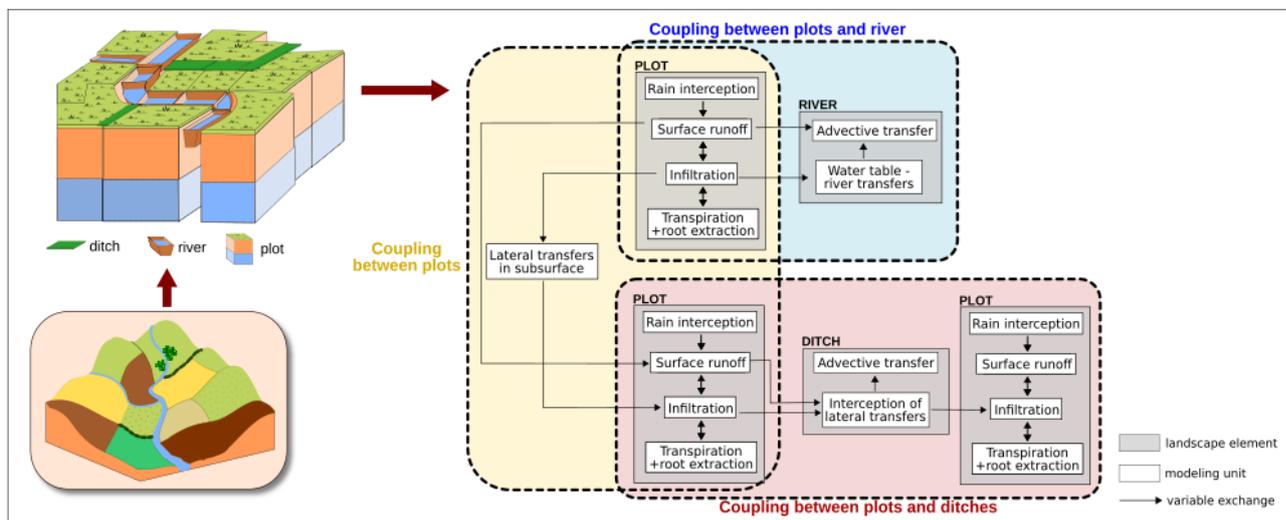


Figure 1 : Principe de la représentation modulaire dans PESHMELBA

La structure de modélisation adoptée est une structure modulaire, qui permet de représenter chaque élément du bassin versant (surfaccique ou linéaire) avec le degré de complexité souhaité (susceptible d'évoluer au fil de l'acquisition de connaissances) tout en couplant ces éléments entre eux de façon dynamique. Les processus dominants les transferts d'eau et le devenir des pesticides pour chaque type d'élément sont représentés par un modèle pré-existant, ou développé/adapté pour les besoins de PESHMELBA. L'ensemble est couplé pour aboutir à un modèle spatialisé à l'échelle du bassin versant avec le coupleur dynamique Open-PALM (Fouilloux et Piacentini, 1999), développé au CERFACS.

## GeoMELBA : discrétisation du bassin en éléments homogènes et détermination des connexions hydrologiques

Le modèle s'appuie sur un maillage permettant de rendre compte de la configuration du paysage, basé sur le découpage du bassin en éléments surfacciques (parcelles, bandes enherbées) et linéaires (talus, fossés, routes, haies) homogènes, et sur la détermination des connexions hydrologiques entre ces éléments. Celles-ci permettant ensuite de transférer de façon adéquate les flux simulés par le modèle sur chaque élément.

La chaîne GeoMELBA a été élaborée pour construire la représentation numérique des éléments du paysage et de leurs connexions que demande le modèle. Elle est inspirée du travail réalisé dans le cadre du projet MIRIPHYQUE (Carluet et al, 2014), qui se basait sur le même type de représentation de l'espace agricole dans un petit bassin versant.

Il s'agit de découper l'espace en éléments surfacciques homogènes du point de vue de la pente, du type de sol et de l'occupation du sol (parcelle ou portion de parcelle) et en tronçons linéaires homogènes du point de vue de leurs caractéristiques relatives au devenir des contaminants : pente, largeur, substrat, végétation éventuellement (Figure 2).

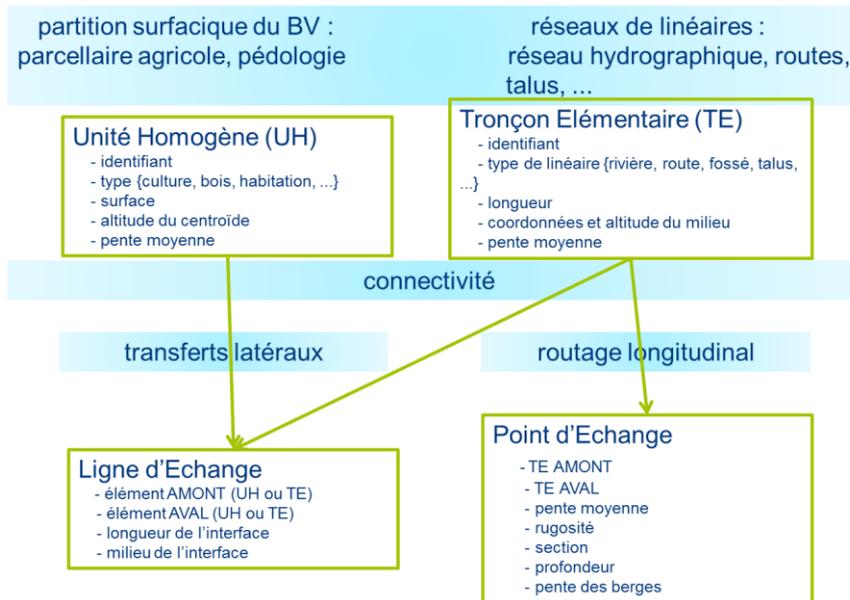
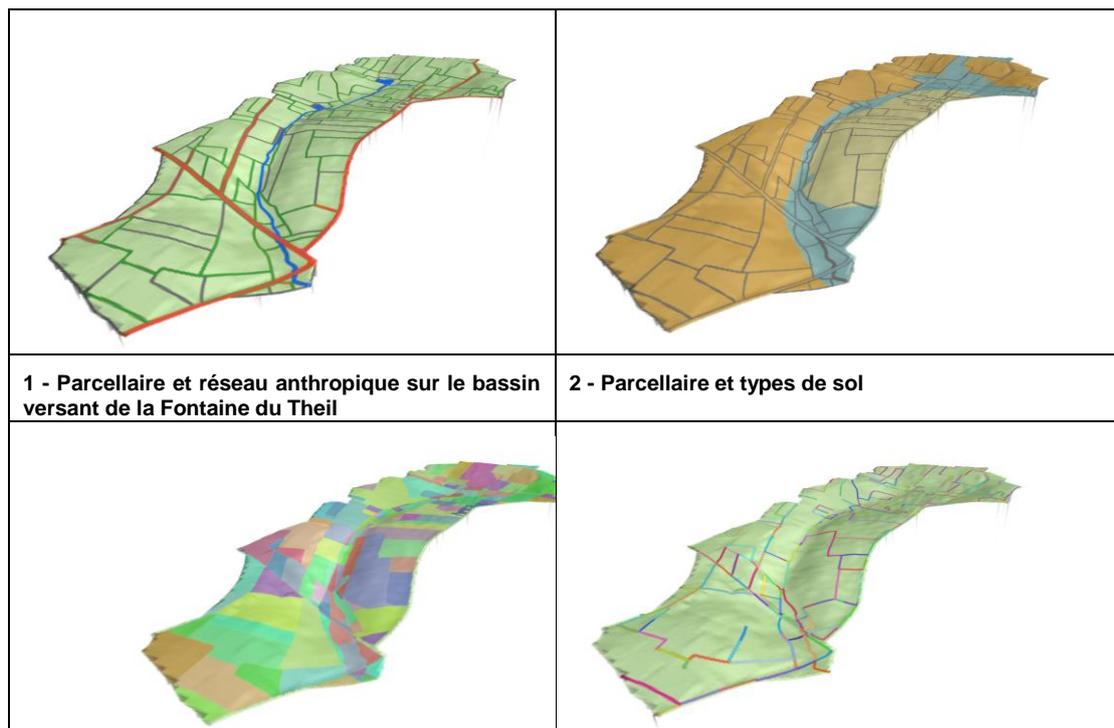


Figure 2 : Schéma de principe des objets et connexions dans GEO-MELBA

La chaîne de traitement commence par la détermination des unités homogènes surfaciques (UH), obtenues par le croisement des cartes de parcelle et de types de sol. Dans des zones de topographie peu marquée, et/ou de parcelle large, un découpage supplémentaire des UHs pourra être envisagé, pour assurer une relative homogénéité de la taille des UHs (notamment du point de vue de la longueur dans le sens de l'écoulement), et pour représenter au mieux les directions d'écoulement (talwegs dans de grandes parcelles par ex, qui ne peuvent être assimilées à des plans). Les zones tampons enherbées sont considérées comme des éléments surfaciques.

Les différents éléments linéaires (fossés, routes, haies, haies sur talus, réseau hydrographique) sont alors découpés en tronçons élémentaires (TE) pour qu'un tronçon ne longe qu'une UH de part et d'autre de son tracé. La représentation des zones tampons humides artificielles, qui ne sont pas encore intégrées dans le modèle, sous forme d'éléments surfaciques ou ponctuels reste à trancher.

Enfin, les connexions entre ces différents éléments sont déterminées, lignes ou points d'échanges, et stockées pour être utilisées ensuite par PESHMELBA.



<b>3 - Création des unités homogènes (surfacique) : UH</b>	<b>4 - Découpage des linéaires suivant les limites des UH ⇒ Tronçons Élémentaires (TE)</b>
<b>5 - Représentation du BV permettant de modéliser les connexions entre les éléments du paysage</b>	<b>6 - Zoom sur les connexions entre éléments :</b> Parcelles agricoles contiguës : transfert latéral ⇒ création d'une Ligne d'Echange : UH1 amont/UH2 aval Entre UH2 et UH3 : transferts latéraux ⇒ Ligne d'Echange UH2 / TE1 ⇒ Ligne d'Echange TE1 :/UH3 Transfert longitudinal entre fossés : ⇒ création d'un Point d'Echange TE1 / TE2

Figure 3 : Succession des étapes de traitement dans GEO-MELBA

La succession des étapes de traitement réalisées par GEO-MELBA est représentée sur la Figure 3.

L'ensemble est développé en PYTHON sous GRASS et SAGA. La géodatabase GRASS permet d'utiliser des MAPSETS pour stocker les résultats des différents scénarios pour un même bassin versant. Les résultats sont transmis à PESHMELBA au format csv, et contiennent l'ensemble des caractéristiques nécessaires au modèle pour construire la paramétrisation relative aux éléments qu'il manipule (cf annexe I décrivant les données et paramètres nécessaires au modèle).

La détermination des connexions hydrologiques entre ces éléments (surface/surface ; linéaire/linéaire, surface/linéaire...) est un point central et qui pose des problèmes non triviaux dans le cas de Systèmes d'Information Géographiques non parfaits (couches d'information non jointives ou au contraire se recouvrant partiellement par exemple), ce qui est le cas le plus courant. Les choix qui ont été effectués pour le développement ont cherché à s'abstraire au maximum des difficultés liées au caractère imparfait des données, de façon à produire une chaîne d'outils qui soit la plus générique possible et limite le temps de manipulation/modification des couches d'information nécessaires à chaque changement de bassin versant d'application. Cette orientation, qui paraît nécessaire pour être le plus efficace possible sur le moyen-long terme, a en revanche exigé des recherches et tests consommateurs en temps sur la durée. Si la chaîne d'outils GeoMELBA est quasi complète, il reste à la consolider pour que sa mise en œuvre soit la plus simple possible et demande le moins de manipulations intermédiaires à l'utilisateur. Notamment, la détermination des lignes d'échange Unité hydrologique surfacique ⇒ Tronçon Élémentaire n'est pas encore totalement automatique quand les couches ne sont pas jointives.

La consolidation de GeoMELBA apparaît comme une priorité pour permettre une application suffisamment souple de PESHMELBA sur de nouveaux bassins, ainsi que pour construire ou modifier aisément des scénarios de solutions correctives (occupation du sol, aménagement d'infrastructures vertes). Elle constituera l'étape suivante la validation de PESHMELBA sur le bassin de la Fontaine du Theil. Outre la résolution des difficultés liées à l'imperfection (inévitable) des données d'entrée citée plus haut, un passage de la chaîne de traitement sous PostGIS est envisagé pour assurer l'évolutivité de l'ensemble.

Notons par ailleurs qu'un outil comme GeoMELBA, qui détermine les connexions hydrologiques entre les différents éléments du paysage, qu'il s'agisse d'éléments surfaciques ou linéaires, en intégrant leur type et leurs caractéristiques, permet de réaliser un premier diagnostic des zones à risque de transfert sur le bassin, en amont de la modélisation, et peut ainsi déjà permettre d'instaurer un dialogue avec les opérationnels autour de la gestion des écoulements et des transferts sur un bassin versant.

## Développement de PESHMELBA

L'annexe I, jointe au rapport (peshmelba\_20182019.pdf), détaille les choix qui ont été effectués (**Partie I : Présentation générale du projet**), ainsi que la façon dont ils ont été implémentés, tant pour la construction des différents modules (**Partie II : Développement des différents modules**) que pour leur articulation au

sein d'OpenPALM (**Partie III : Mise en œuvre du couplage, implémentation dans OpenPALM**). Un article également présenté en annexe II (Rouzies et al, 2019), illustre les tests réalisés sur des scénarios contrastés pour vérifier le réalisme des résultats obtenus.

On donne ici un bref aperçu des modules développés ou adaptés pour PESHMELBA, et implémentés dans PALM. On distingue les modules représentant les processus au sein des éléments du paysage, et les modules représentant les interfaces entre éléments (du même type ou non) et servant à calculer les flux entre les différents éléments.

Les types d'éléments inclus pour l'instant sont :

- Pour les éléments pris en compte par le modèle comme des surfaces : les parcelles, les zones tampons végétalisées, c-a-d bandes enherbées ou boisées,
- Pour les éléments pris en compte par le modèle comme des linéaires : les fossés, haies et haies sur talus, cours d'eau. Les routes sont pour l'instant assimilées à des fossés, car bordées de fossés sur les deux bassins versants sur lesquels l'application du modèle est prévue dans un premier temps. Cette représentation pourra évoluer si besoin.

Un autre type d'élément, virtuel, a été créé pour pouvoir simuler des tests sur des portions de versant tout en assurant des conditions aux limites réalistes, en amont et/ou aval de la zone simulée.

Les types d'interface inclus pour l'instant sont les interfaces :

- Latérales :
  - Parcelle / parcelle, qui servent également pour les interfaces parcelle / bande enherbée
  - Parcelle / fossé / parcelle
  - Parcelle / cours d'eau
  - Parcelle / haie / parcelle
- Longitudinales
  - Fossé / fossé
  - Fossé / cours d'eau
  - Cours d'eau / cours d'eau
  - Haie / Haie

### ***Représentations adoptées pour les éléments surfaciques***

Pour les éléments surfaciques, le module « de base » est celui représentant une **parcelle**. Les processus qui y sont représentés sont :

- Ecoulement vertical variablement saturé, représenté par l'équation de Richards (solution de Ross, 2003)
- Ecoulement vertical saturé, représenté par l'équation de Darcy (Darcy, 1856)
- Transfert vertical par advection-dispersion des pesticides (Ross, 2003)
- Processus d'adsorption (isotherme linéaire ou de Freundlich) et de dégradation (dégradation linéaire)
- Représentation du devenir d'un ou plusieurs produits de dégradation pour chaque substance active. Chacun de ces éléments est supposé soumis aux mêmes processus que les substances actives, caractérisé par un coefficient d'adsorption et une demie vie. La part de chaque métabolite créé est par la dégradation d'une substance active est fixée, indépendamment des conditions du milieu. Cette représentation simple des produits de dégradation pourra évoluer au fur et à mesure du développement des connaissances et de l'acquisition de données de référence sur ces substances.
- Evapotranspiration, calculée à partir d'une évapotranspiration potentielle : interception par la végétation, évaporation du sol, extraction racinaire (équation de Feddes, Varado et al, 2006) ; possibilité de faire évoluer la surface foliaire au cours du temps
- Ruissellement, représenté par l'équation de l'onde cinématique (Li et al, 1975)
- Transfert des pesticides par convection dans le ruissellement, avec adsorption et dégradation en surface, et représentation d'une couche de mélange, permettant la mobilisation des pesticides disponibles.

La représentation des écoulements macroporaux est a priori permise par l'approche de Ross, mais n'a pas

encore été implémentée, bien qu'elle semble nécessaire pour représenter le transfert des pesticides. La représentation des macropores est toutefois essentiellement faite chez Ross par une modification de la conductivité à l'approche de la saturation, dont on peut craindre qu'elle ne suffise pas ici à représenter les processus à l'œuvre. Il faudra donc probablement développer une autre méthode, en se basant sur une méthode existant dans la bibliographie.

De façon originale par rapport aux modèles abordant l'échelle du bassin versant, PESHMELBA permet la représentation de nappes perchées. C'est un point qui paraît essentiel pour rendre compte des observations dans le contexte de bassin sur socle peu perméable sur lequel le modèle sera appliqué. Celles-ci ont un rôle majeur dans le fonctionnement hydrologique de ces bassins, ainsi que sur les transferts de contaminants qui y adviennent (Molénat 1999).

**Les bandes enherbées** sont représentées de façon analogue aux parcelles, leur fonctionnement spécifique est rendu par la prise en compte de la perméabilité accrue du sol et de la rugosité forte liées à la présence de végétation pérenne, favorisant le développement d'un mat racinaire très perméable.

Le schéma d'infiltration au sein de la BE a d'abord été validé sur différents types de sols contrastés provenant de l'article Munoz-Carpena et al., 2018. Cela a permis de vérifier que la bande enherbée seule est bien représentée dans PESHMELBA (Figure 4)

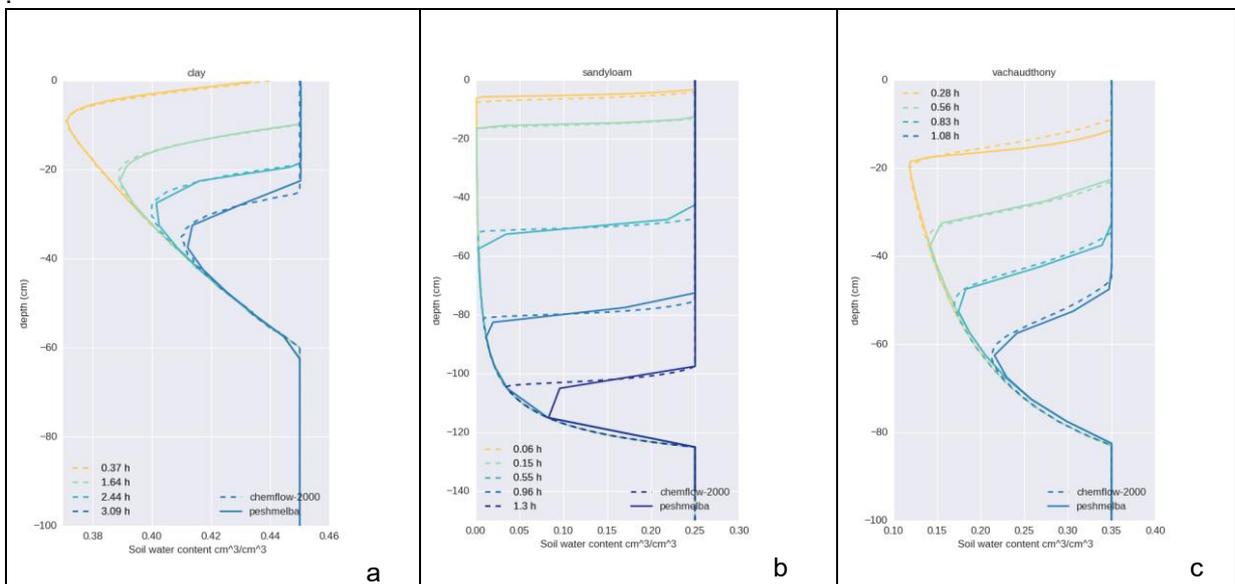
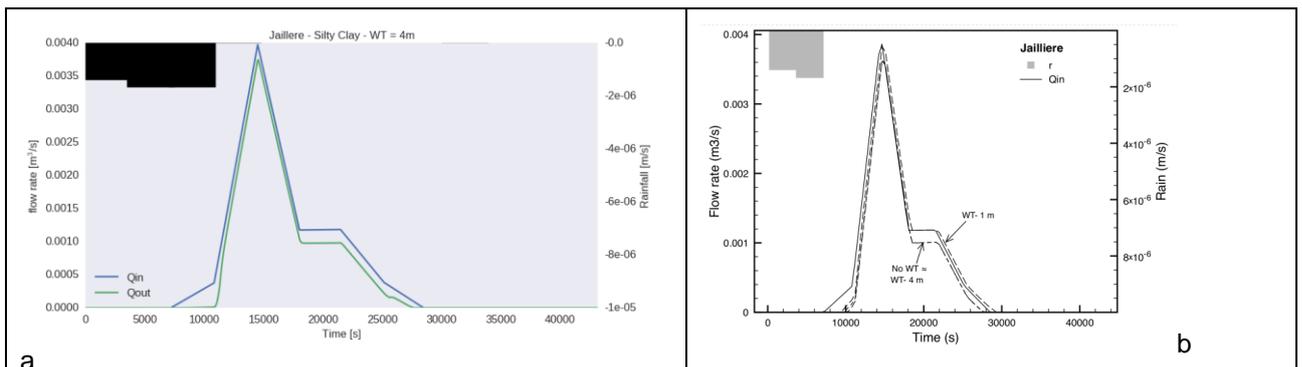


Figure 4 : Comparaison des profils d'infiltration obtenus avec Chemflow (pointillés) et PESHMELBA (trait plein) pour un sol argileux (a), limono-argileux (b). (c) Comparaison du profil d'infiltration observé dans un sol limono-argileux et simulé par PESHMELBA.

Dans un deuxième temps, on a testé le couplage infiltration + ruissellement de surface avec cas sans saturation et avec saturation (nappe peu profonde) en comparant aux résultats de VFSSMOD dans (Lauvernet and Munoz-Carpena 2018). De même on a testé plusieurs types de sols, plus ou moins perméables (Figure 5). La taille restreinte de ces structures (quelques mètres dans le sens de l'écoulement) par rapport à celle des parcelles les alimentant (quelques dizaines à centaines de mètres) peut poser des problèmes de compatibilité des pas de temps utilisés pour y représenter le ruissellement et le partage infiltration / ruissellement. On a pu constater une certaine sensibilité aux pas de temps général et de ruissellement. Des tests sont en cours pour déterminer le meilleur compromis possible pour représenter correctement l'influence de ces structures sans alourdir exagérément les temps de calcul.



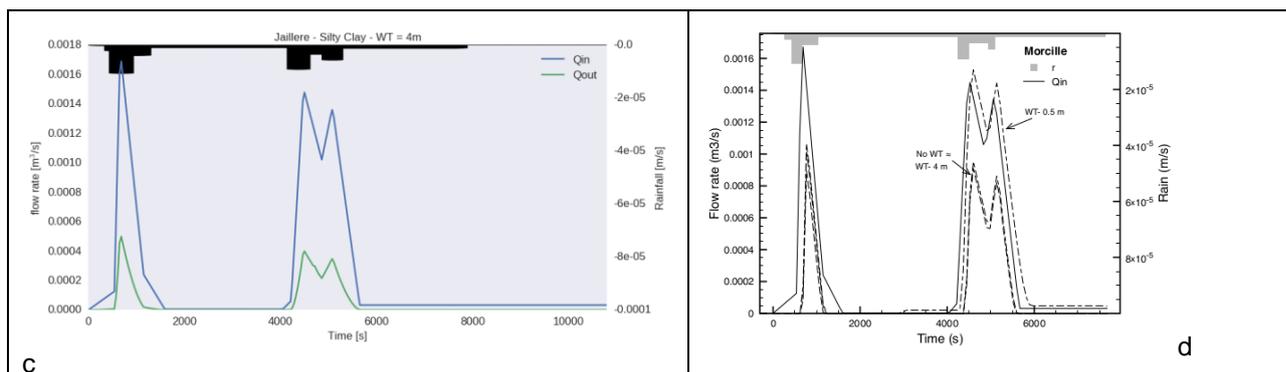


Figure 5 : Comparaison des flux ruisselés en sortie de bande enherbée simulés par PESHMELBA et VFSMOD sur la Jaillière (a et b) et la Morcille (c et d) pour une nappe située à 4 m sous la surface

**Les haies et haies sur talus** sont identifiées dans GeoMELBA comme des linéaires, et représentées au sein de PESMELBA d'une façon mixte : comportement d'un élément surfacique pour l'interception du ruissellement, des écoulements subsurfaciques, et l'infiltration dans le sol de la haie, comportement d'un élément linéaire pour le ruissellement le long du linéaire de la haie. Les haies sont représentées par la juxtaposition de deux colonnes « parcelle » pour permettre de représenter la topographie et les caractéristiques souvent très contrastées que l'on trouve en amont et aval de haies ou haies sur talus anciennes. Le ruissellement parvenant sur la partie amont de la haie est intercepté plus ou moins fortement selon que la haie soit située sur un talus ou non. Une partie de l'eau et des solutés non infiltrés de façon matricielle est infiltrée de façon préférentielle vers la nappe de fond de profil, pour rendre compte des écoulements préférentiels induits par le réseau racinaire et la structuration du sol résultant de la forte teneur en carbone organique sous ces structures. Une partie de cette eau (et des solutés qu'elle contient) est transférée longitudinalement le long de la haie, d'autant plus que le tracé de celle-ci suit la ligne de plus grande pente. Cette représentation a été adoptée en se basant sur les connaissances développées notamment à l'UMR SAS de Rennes sur les haies et leur influence sur les transferts d'azote, et en les transposant, autant que faire se peut, au cas des pesticides. Des recherches complémentaires restent nécessaires pour affiner la représentation adoptée et les ordres de grandeur à utiliser pour les différents paramètres (notamment pour ce qui concerne l'infiltration préférentielle).

### Représentations adoptées pour les éléments linéaires

**Les fossés** sont les premiers éléments linéaires qui ont été inclus dans le modèle. Dans la représentation choisie, ils interceptent les écoulements de surface et de proche subsurface venant de l'amont (sur une épaisseur de sol correspondant à leur profondeur) alors que les écoulements plus profonds sont transférés à l'aval en subsurface profonde, sans être intercepté par le fossé. Le devenir des pesticides y est décrit de la même façon que pour le ruissellement sur les parcelles : convection (pas de diffusion à ce stade), dégradation linéaire et adsorption sur le fond du fossé (sur une épaisseur de substrat choisie par l'utilisateur, actuellement 2 cm).

**Les tronçons de cours d'eau** ont un fonctionnement analogue à celui des fossés, mais peuvent en plus échanger de l'eau avec la nappe plus profonde, selon une représentation basée sur l'équation de Miles (cf Annexe I).

Les écoulements au sein de ces éléments sont calculés par l'équation de l'onde cinématique. Cette équation est résolue à chaque pas de temps sur chaque réseau indépendant, constitué d'éléments de même type connectés. Les écoulements dans un réseau de fossés se jetant dans un cours d'eau sont ainsi simulés indépendamment des écoulements dans le cours d'eau principal, et l'alimenteront au pas de temps suivant.

### Représentations adoptées pour les interfaces

**Les échanges entre parcelles** sont basés sur l'équation de Darcy pour les écoulements de subsurface. L'originalité de la représentation adoptée est qu'elle tient compte explicitement des nappes perchées, et préserve le gradient de concentration souvent observé depuis la surface vers la profondeur, en réalisant des échanges « couche à couche ».

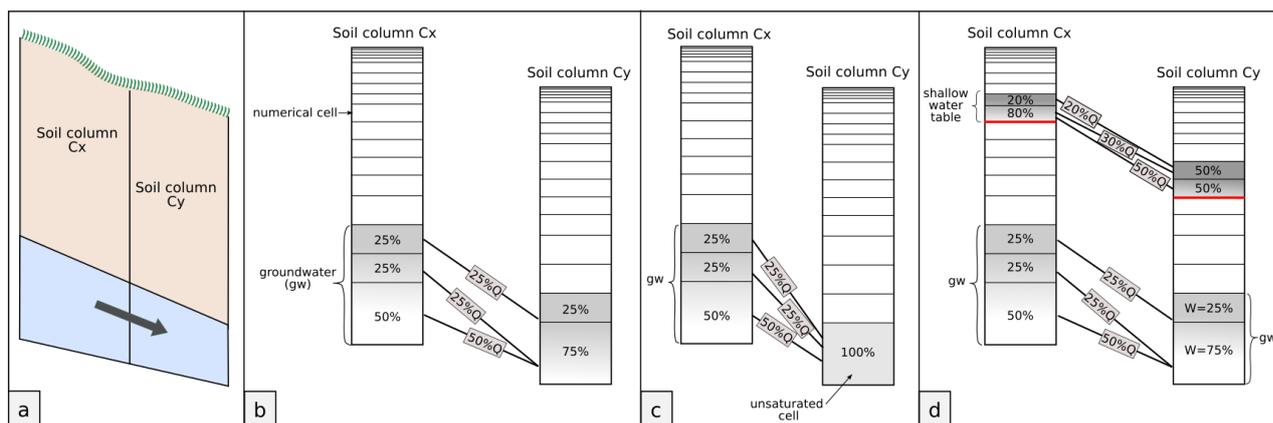


Figure 6 : Gestion des écoulements latéraux subsurface

Les écoulements ne peuvent se faire que de l'amont vers l'aval. Autrement dit, dans les cas où le gradient hydraulique aurait tendance à induire des écoulements remontant le versant, les écoulements sont bloqués. Cette limitation ne devrait a priori pas poser problème sur les bassins à la topographie relativement marquée sur lesquels il est prévu d'appliquer le modèle.

Les flux ruisselés sont calculés en fonction de la lame d'eau présente sur une parcelle (ce qui ne s'est pas infiltré) via l'équation de l'onde cinématique, et transmis aux éléments connectés à l'aval. **Les échanges entre parcelles et haies ou haies sur talus** sont gérés de la même façon que les échanges entre parcelles.

**Les échanges entre parcelles et fossés** sont déterminés en considérant que le fossé intercepte les écoulements de subsurface et le ruissellement qui aurait été transmis à la parcelle aval s'il n'avait pas été là, sur sa profondeur, et qu'il n'influence pas les écoulements plus profonds.

**Les échanges entre parcelles et cours d'eau** sont calculés selon le même principe pour le ruissellement et selon l'équation de Miles pour la nappe de fond de profil. Dans le cas d'une nappe perchée, on considère une surface de suintement pour calculer son écoulement vers la rivière.

Pour l'instant, le débordement des fossés et cours d'eau n'est pas pris en compte par le modèle. Cela ne paraît pas limitant dans la mesure où il ne s'agit pas de représenter des événements extrêmes du point de vue hydrologique, mais relativement courants pour le transfert de pesticides,

**Les échanges entre tronçons de fossés et/ou de cours d'eau** sont calculés de façon analogue au ruissellement entre parcelles : l'écoulement au sein d'un tronçon est calculé en fonction de la hauteur d'eau (liée aux apports depuis l'amont du réseau et aux apports latéraux via les parcelles) selon l'équation de l'onde cinématique et transféré au(x) tronçon(s) aval.

### Couplage entre les éléments, gestion du temps

L'ensemble de ces modules a été couplé dans OpenPALM. Une attention particulière est portée à la gestion des pas de temps, pour que les différents processus soient représentés à un pas de temps correspondant à leur dynamique propre. Par exemple, le pas de temps caractéristique du ruissellement ou des écoulements dans le réseau hydrographique est bien plus faible que celui caractéristique de l'infiltration dans les parcelles ou des échanges en subsurface. Certains modules sont donc appelés à un pas de temps moindre que le pas de temps « de base » utilisé par PALM pour les processus les plus lents. Par ailleurs, ce pas de temps PALM varie lui-même en fonction des variables de forçage : il est raffiné en période de pluie ou de ressuyage suivant les principaux événements pluvieux, toujours dans un souci d'efficacité au niveau numérique. Le développement et l'intégration « au fil de l'eau » des différents modules avait conduit à un modèle représentant correctement les différents processus, aux pas de temps souhaités, mais avec des temps de calcul trop élevés pour un modèle à vocation opérationnelle, par suite des appels répétés que OpenPALM devait faire aux scripts Python représentant les différents modules. Une réflexion approfondie a été menée pour optimiser le temps de calcul en rationalisant les appels de PALM aux scripts Python tout en maintenant la cohérence du déroulé des simulations. Elle a permis des gains significatifs au niveau du temps de calcul, et permet d'envisager sereinement l'application à l'échelle du bassin versant.

### Applications du modèle. Perspectives immédiates

Le modèle a pour l'instant été testé sur des scénarios contrastés représentant des versants avec ou sans fossés, pour des pesticides aux caractéristiques très différentes : isoproturon (et un de ses métabolites, le desmethylisoproturon), et le diflufenicanil (Rouzies, et al. 2019). Les résultats sont encourageants : les ordres de

grandeur obtenus (transferts d'eau et de solutés) sont conformes aux attendus, et les différents scénarios conduisent effectivement à des résultats très contrastés selon les cas : présence ou non de fossés, comportement dans l'environnement contrasté pour les différentes molécules représentées. Ils confirment l'intérêt de représenter explicitement et de façon dynamique l'influence des éléments du paysage sur les transferts et le devenir des pesticides.

PESHMELBA comporte maintenant l'ensemble des éléments nécessaires à son application sur le bassin de la Fontaine du Theil. Toutefois, certains éléments (haies et haies sur talus par exemple) ont été récemment intégrés et demandent à être plus amplement testés pour bien cerner leur fonctionnement et définir correctement les paramètres à utiliser pour la partie relativement conceptuelle qu'ils incluent, le cas échéant. L'étape suivante concerne l'application sur le bassin versant de la Fontaine du Theil, pour quelques herbicides pour lesquels des données d'application et d'exportation sont disponibles entre 1996 et 2006, période de suivi du bassin par ARVALIS et l'UIPP (atrazine, dimethenamid, ioxynil, isoproturon, sulcotrione, diflufenicanil). Cette validation « en vraie grandeur » constituera une étape essentielle pour le développement du modèle. Elle permettra notamment d'évaluer si le modèle est capable de rendre compte des évolutions (usages des pesticides –doses et dates d'application-, implantation de zones tampons enherbées ou de talus) intervenues pendant la période et ayant conduit à une baisse des exportations. Un point qui apparaît comme essentiel à tester est la pertinence du maillage choisi, basé sur les objets du paysage, et qui conduit à des mailles de taille et de forme très variables. Selon les résultats obtenus, il pourra être nécessaire de faire évoluer GeoMELBA pour qu'il fournisse des mailles plus homogènes. Quelques scénarios contrastés d'aménagements et d'occupation du sol et pratiques associés seront explorés, pour évaluer l'ampleur des solutions à mettre en œuvre sur ce bassin pour y diminuer significativement les transferts de pesticides.

L'application du modèle à d'autres bassins versants pourra alors être envisagée, notamment sur le bassin de la Morcille (69) pour lequel il sera nécessaire d'inclure la représentation de parcelles comportant des rases (petits fossés sub-horizontaux destinés à limiter l'érosion au sein des parcelles viticoles). L'intégration de parcelles drainées et de zones tampons humides artificielles est également à réaliser dans un proche avenir pour pouvoir couvrir une plus grande diversité de bassins versants. Les connaissances et modélisations déjà développées sur le sujet (Branger, Tournebize et al. 2009, Tournebize, Vincent et al. 2011) permettent d'ores et déjà d'envisager une première intégration de ces éléments, qui pourra être affinée par la suite, toujours dans la même logique que celle qui a guidé le développement de PESHMELBA.

## Conclusion. Perspectives

Le modèle PESHMELBA est parvenu à un stade de développement qui permet son application à l'échelle du petit bassin versant, en l'occurrence le bassin versant de la Fontaine du Theil, sur lequel le modèle sera testé et validé dans les mois prochains.

L'intérêt suscité par ce modèle, à l'international comme au niveau plus opérationnel : 1 communication par poster en 2017 (Rouzies, Barachet et al. 2017), deux communications orales en 2018 dans des congrès internationaux (Rouzies, Barachet et al. 2018, Rouzies, Barachet et al. 2018), une communication orale acceptée au congrès 2019 de la SETAC ; une communication orale au congrès des SAGE en 2018 (Rouzies, Barachet et al. 2018) montre, si besoin en était, le potentiel d'un système de modélisation permettant de représenter et comparer différents scénarios d'occupation du sol, de pratiques agricoles et d'aménagements, sur un bassin versant agricole, du point de vue du transfert des pesticides.

La poursuite du développement de PESHMELBA dépend de l'acceptation de projets permettant de le financer. Trois lettres d'intention ont été acceptées et doivent donner lieu à dépôt de projet complet : deux (Région AURA et Ecophyto-Leviers territoriaux) portent sur l'utilisation de PESHMELBA pour co-construire avec les acteurs de terrain des scénarios de solutions correctives à l'échelle du bassin versant, et prévoient une application sur le bassin de la Morcille, du Rû d'Ancoeur (Grandes cultures, Brie) et de la Gimond (Polyculture élevage, Monts du Lyonnais) ; le troisième (ANR Jeune Chercheur) a pour objectif de développer l'assimilation de données au sein du modèle, afin d'intégrer au mieux les informations disponibles sur le bassin d'application (Morcille) et de mieux contraindre le modèle et diminuer les incertitudes de modélisation. Le principe d'une thèse sur ce sujet est déjà acquis, indépendamment de l'acceptation ou non du projet ANR. L'utilisation de PESHMELBA de façon « fluide » pour construire et tester différents scénarios d'aménagements suppose de consolider la chaîne de traitement GeoMELBA, notamment pour ce qui concerne la gestion des données imparfaites.

La structure modulaire et évolutive du modèle permet d'envisager de représenter d'autres contaminants (métabolites, substances vétérinaires). La représentation du devenir et des transferts des métabolites, nécessaire pour représenter l'exposition globale à laquelle sont soumis les organismes aquatiques, suppose de représenter correctement les zones et temps de résidence, et la réactivité des différentes zones du bassin. Cela passe entre autres par la représentation d'un compartiment profond dans PESHMELBA, qui pour l'instant ne représente que des nappes peu profondes. L'intégration de l'érosion et du transfert de matières en suspension est également envisagée, de façon simplifiée dans un premier temps, pour pouvoir rendre compte

du transfert des contaminants adsorbés et de l'influence des éléments du paysage sur ces transferts.

Enfin, la représentation spatialement explicite des éléments du paysage que fait le modèle permet par ailleurs d'envisager de représenter de façon couplée, à terme, la connectivité de ces éléments du point de vue de son influence sur la biodiversité : biodiversité « ordinaire », et auxiliaires de culture.

## Bibliographie

Branger, F., J. Tournebize, N. Carluier, C. Kao and M. Vauclin (2009). "A conceptual modelling approach for pesticides transport in a subsurface drained field: the PESTDRAIN model." Agricultural Water Management **96**: 415-428.

Carluier, N., P. Benoit, Y. Coquet, B. Cheviron, C. Gascuel Odoux, V. Gouy, C. Guyot, C. Lauvernet, G. Le Hénaff, J. Maillet-Mezeray, A. Miralles, G. Molla, F. Pinet and B. Réal (2014). Mise au point de descripteurs du risque de contamination des eaux de surface par les phytosanitaires à l'échelle du bassin versant. Prise en compte des dimensions spatiales et temporelles. Appui à l'évaluation et à la gestion du risque (MIRIPHYQUE). Appel d'offres Pesticides du MEEDD.

Fouilloux, A., Piacentini, A. (1999) : The PALM project: MPMD paradigm for an oceanic data assimilation software. In: Euro-Par 1999 - Parallel Processing: 5th International Conference, Proceedings. (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1999, 1685 LNCS:1423-1430)

Lauvernet, C. and R. Munoz-Carpena (2018). "Shallow water table effects on water, sediment, and pesticide transport in vegetative filter strips -- Part 2: model coupling, application, factor importance, and uncertainty " Hydrology and Earth System Sciences **22**: 71-87.

Li, R.-M., B. Simons, D., & A. Stevens, M. (1975). Nonlinear kinematic wave approximation for water routing. Water Resour. Res., 245–252.

Molénat, J. (1999). Rôle de la nappe sur les transferts d'eau et de nitrates dans un bassin versant agricole. Etude expérimentale et modélisation., Université Rennes I.

Muñoz-Carpena, R., J. E. Parsons and J. W. Gilliam (1999). "Modeling hydrology and sediment transport in vegetative filter strips." Journal of Hydrology **214**: 111-129.

Ross, P. (2003). Modeling soil water and solute transport - fast, simplified numerical solutions. Agronomy Journal, 1352–1361.

Rouzies, E., C. Barachet, C. Lauvernet and N. Carluier (2017). Understanding water circulation and pesticides fate at the catchment scale by modeling the influence of landscape elements. Pesticide Behaviour in Soils, Water and Air, University of York, York, UK. 30 august-1<sup>st</sup> septembre 2017

Rouzies, E., C. Barachet, T. Morel, C. Lauvernet and N. Carluier (2018). Peshmelba : a spatialized model of water circulation and pesticide fate at the catchment scale coupling the landscape elements with the openpalm coupler. EGU General Assembly 2018, Vienne.

Rouzies, E., C. Barachet, T. Morel, C. Lauvernet and N. Carluier (2018). PESHMELBA : pesticides et hydrologie, modélisation à l'échelle du bassin versant. Séminaire SAGE, Orléans. France.

Rouzies, E., C. Barachet, T. Morel, C. Lauvernet and N. Carluier (2018). PESHMELBA: modelling water transfer and pesticide fate at the catchment scale considering landscape elements. CMWR. Computational Methods in Water Resources, Saint Malo (France).

Rouzies, E., C. Lauvernet, C. Barachet, T. Morel, F. Branger, I. Braud and N. Carluier (2019). "From agricultural catchment to management scenarios: a modular tool to assess effects of landscape features on water and pesticide behavior " Science of the Total Environment. 671, 1144-1160. Doi : /10.1016/j.scitotenv.2019.03.060

Tournebize, J., B. Vincent, C. Chaumont, C. Gramaglia, C. Margoum, P. Molle, N. Carluier and J. J. Gril (2011). "Ecological services of artificial wetland for pesticide mitigation Socio-technical adaptation for watershed management through TRUSTEA project feedback." Procedia Environmental Sciences **9**: 183-190.

Varado, N., Braud, I., Ross, P., 2006. Development and assessment of an efficient vadose zone module solving the 1d Richards' equation and including root extraction by plants. J. Hydrol. 258–275.

## Table des illustrations

Figure 1 : Principe de la représentation modulaire dans PESHMELBA .....	10
Figure 2 : Schéma de principe des objets et connexions dans GEO-MELBA .....	11
Figure 3 : Succession des étapes de traitement dans GEO-MELBA.....	12
Figure 4 : Comparaison des profils d'infiltration obtenus avec Chemflow (pointillés) et PESHMELBA (trait plein) pour un sol argileux (a), limono-argileux (b). (c) Comparaison du profil d'infiltration observé dans un sol	

limono-argileux et simulé par PESHMELBA. ....	14
Figure 5 : Comparaison des flux ruisselés en sortie de bande enherbée simulés par PESHMELBA et VFSMOD sur la Jaillière (a et b) et la Morcille (c et d) pour une nappe située à 4 m sous la surface .....	15
Figure 6 : Gestion des écoulements latéraux subsurfaciques .....	16

## **Annexe 1 : Rapport technique sur le développement de PESH-MELBA**

Fichier : PESHMELBA\_ManuelTechnique\_2019\_02\_28.pdf

## **Annexe 2 : Article paru dans la revue STOTEN (Science of the Total Environment)**

Fichier : PESHMELBA\_Stoten\_2019\_02\_28.doc ; version auteur





**Irstea**

1, rue Pierre-Gilles de Gennes  
CS 10030  
92761 Antony Cedex

**01 40 96 61 21**

[www.irstea.fr](http://www.irstea.fr)

**Agence Française pour la Biodiversité**

Hall C – Le Nadar  
5, square Félix Nadar  
94300 Vincennes

**01 45 14 36 00**

[www.afbiodiversite.fr](http://www.afbiodiversite.fr)



---

# PESHMELBA

---

**PESticides et Hydrologie : Modelisation à l'echELLE  
du BAssin versant**

Le 12 avril 2019



# Table des matières

<b>I</b>	<b>Introduction</b>	<b>1</b>
<b>1</b>	<b>Présentation générale du projet</b>	<b>3</b>
1.1	Objectifs . . . . .	3
1.2	Réflexion sur l’outil de modélisation, choix d’OpenPALM . . . . .	3
1.3	Démarche envisagée . . . . .	5
<b>2</b>	<b>Devenir des produits phytosanitaires : aperçu des processus à représenter</b>	<b>7</b>
2.1	Transport hydrique des pesticides . . . . .	7
2.2	Adsorption/Désorption . . . . .	8
2.3	Dégradation . . . . .	9
<b>3</b>	<b>Éléments et connexions à représenter</b>	<b>11</b>
3.1	Types d’éléments à modéliser . . . . .	11
3.1.1	Parcelles et Bandes enherbées . . . . .	11
3.1.2	Fossés et rivières . . . . .	12
3.1.3	Haies, talus . . . . .	12
3.1.4	Boundary Condition Storage (BCS) . . . . .	13
3.2	Topologie . . . . .	13
<b>4</b>	<b>Problématiques relatives au couplage</b>	<b>15</b>
4.1	Gestion de l’aspect spatial du modèle . . . . .	15
4.2	Gestion du pas de temps . . . . .	16
4.3	Fonctionnement des lignes d’échange . . . . .	19
<b>II</b>	<b>Développement des différents modules</b>	<b>21</b>
<b>5</b>	<b>Propos généraux sur l’implémentation des modules dans PESHMELBA</b>	<b>23</b>
<b>6</b>	<b>Pre-processing</b>	<b>29</b>
6.1	Importation des données d’entrée et initialisation des variables PALM . . . . .	29
6.2	Initialisation des pas de temps et des séries temporelles . . . . .	29
6.3	Analyse du paysage . . . . .	31
<b>7</b>	<b>Transferts d’eau (1D) dans le sol</b>	<b>35</b>
7.1	Introduction . . . . .	35
7.2	Méthode de résolution de Ross . . . . .	36

7.2.1	Propriétés hydrauliques du sol utilisées . . . . .	36
7.2.2	Potentiel de Kirchhoff . . . . .	36
7.2.3	Discrétisation . . . . .	37
7.2.4	Calcul des flux . . . . .	38
7.2.5	Transport des solutés . . . . .	39
7.2.6	Transfert réactif de solutés . . . . .	40
7.2.7	Prise en compte du flux latéral dans le module FRER1D . . . . .	41
7.2.8	Calcul de l'équilibre d'adsorption en surface . . . . .	42
7.2.9	Couche de mélange . . . . .	43
7.3	Extension de la méthode de résolution de Ross à des propriétés hydrauliques du sol plus générales . . . . .	44
7.3.1	Modification de la courbe de rétention . . . . .	44
7.3.2	Modification de la courbe de conductivité . . . . .	45
7.4	Implémentation . . . . .	48
7.4.1	module FRER1D . . . . .	48
7.4.2	module FRER1D_HEDGE . . . . .	48
7.4.3	module FLOW et FLOW_HEDGE . . . . .	48
7.4.4	modules HYPBC et HYPVGVG . . . . .	49
<b>8</b>	<b>Extraction racinaire</b>	<b>51</b>
8.1	Introduction . . . . .	51
8.2	Représentation des processus . . . . .	52
8.2.1	Interception (précipitations et radiations) . . . . .	52
8.2.2	Extraction racinaire : le modèle de Li and Zhang (2001) . . . . .	53
8.2.3	Utilisation d'un coefficient cultural . . . . .	55
8.3	Implémentation . . . . .	55
8.3.1	PARTITION dans les modules FRER1D et FRER1D_HEDGE . . . . .	55
8.3.2	ROOTS . . . . .	55
8.3.3	getrex . . . . .	56
<b>9</b>	<b>Écoulements dans un réseau</b>	<b>57</b>
9.1	Introduction . . . . .	57
9.2	Représentation des écoulements dans un réseau . . . . .	58
9.2.1	Devenir des pesticides dans un fossé ou une rivière . . . . .	58
9.2.2	Domaine de modélisation, représentation d'un réseau ramifié . . . . .	59
9.3	Schéma de résolution . . . . .	60
9.4	Implémentation . . . . .	61
9.4.1	Analyse de réseau . . . . .	61
9.4.2	Classe Reach . . . . .	62
9.4.3	Classe Network . . . . .	63
9.4.4	Module use_launch_RIVER1D . . . . .	65
<b>10</b>	<b>Transferts latéraux de subsurface</b>	<b>67</b>
10.1	Considérations générales sur l'implémentation des lignes d'échanges . . . . .	67
10.2	Transferts entre deux parcelles . . . . .	68
10.2.1	Rôle de la ligne d'échanges ColumnColumnExchangeLine . . . . .	68
10.2.2	Cahier des charges . . . . .	68
10.2.3	Implémentation de la ligne d'échanges Colonne-Colonne . . . . .	74

10.3	Transferts entre deux parcelles séparées par un fossés . . . . .	81
10.3.1	Introduction : fonctionnement hydrologique d'un fossé . . . . .	81
10.3.2	Représentation . . . . .	81
10.3.3	Implémentation de la ligne d'échanges entre 2 colonnes séparées par un fossé . . . . .	83
10.4	Transferts entre une parcelle et la rivière . . . . .	87
10.4.1	Fonctionnement hydrologique de la ligne d'échanges . . . . .	87
10.4.2	Implémentation . . . . .	88
10.5	Transferts entre une parcelle et une haie . . . . .	90
10.5.1	Introduction : fonctionnement hydrologique d'une haie . . . . .	90
10.5.2	Représentation adaptée aux processus de subsurface . . . . .	90
10.5.3	Implémentation . . . . .	92
10.6	Transferts vers/depuis un élément de condition limite (BCS) . . . . .	94
10.6.1	Fonctionnement d'un élément de type BCS . . . . .	94
10.6.2	Implémentation . . . . .	95
10.7	Module EXCHANGES_SUBSURFACE . . . . .	98
10.7.1	Initialisation d'instances d'éléments du paysage et de lignes d'échanges entre éléments . . . . .	98
10.7.2	Calcul des flux latéraux entre éléments . . . . .	98
10.7.3	Envoi des variables PALM . . . . .	99
10.7.4	Fonction de service : initialize_interface_subsurface . . . . .	99
<b>11</b>	<b>Écoulements de surface</b>	<b>101</b>
11.1	Introduction . . . . .	101
11.2	Représentation du ruissellement . . . . .	101
11.2.1	Couplage surface-subsurface . . . . .	101
11.2.2	Interception du ruissellement par les éléments linéaires . . . . .	102
11.3	Représentation du ruissellement dans PESHMELBA . . . . .	103
11.3.1	Contraintes et problématiques . . . . .	103
11.3.2	Choix adoptés . . . . .	104
11.3.3	Transfert de solutés en surface, couplage surface-subsurface . . . . .	108
11.3.4	Fonctionnement d'une haie/talus . . . . .	108
11.3.5	Connexion avec un élément de condition limite (BCS) . . . . .	109
11.4	Implémentation dans PESHMELBA . . . . .	110
11.4.1	classe HomogeneousSurface . . . . .	111
11.4.2	Classe SurfaceTank . . . . .	113
11.4.3	classe HedgeSurface . . . . .	113
11.4.4	classe SurfaceInterface . . . . .	115
11.4.5	classe HomogeneousSurfaceExchangeLine . . . . .	115
11.4.6	classe TankExchangeLine . . . . .	117
11.5	Module EXCHANGES_SURFACE . . . . .	118
11.5.1	Initialisation des instances représentant les éléments du paysage et des interfaces entre eux . . . . .	118
11.5.2	Calcul du ruissellement sortant et transfert vers les éléments voisins	119
11.5.3	Calcul de l'adsorption et envoi des variables vers PALM . . . . .	121

<b>III Mise en oeuvre du couplage : Implémentation dans OpenPALM</b>	<b>123</b>
<b>12 Introduction</b>	<b>125</b>
<b>13 Intégration du module FRER1D dans OpenPALM</b>	<b>127</b>
13.1 Implémentation basique . . . . .	127
13.2 Gestion du pas de temps externalisée dans PALM . . . . .	128
<b>14 Couplage de modules FRER1D par l'interface Plot-Plot-Interface</b>	<b>131</b>
14.1 Modules utilisés . . . . .	131
14.2 Assemblage dans une branche PALM . . . . .	132
14.3 Mise en place du couplage . . . . .	133
14.4 Optimisation . . . . .	136
<b>15 Gestion des linéaires dans le schéma PALM</b>	<b>139</b>
<b>16 Insertion du ruissellement dans le schéma PALM</b>	<b>141</b>
<b>17 Intégration de River1D, gestion des modules Python et boucles infinies</b>	<b>145</b>
<b>18 Schéma PALM définitif de l'application</b>	<b>149</b>
 <b>Bibliographie</b>	 <b>156</b>
 <b>Annexes</b>	 <b>156</b>
<b>A Description des variables transmises dans OpenPALM</b>	<b>157</b>
<b>B Algorithme de Runge-Kutta au second et quatrième ordre</b>	<b>169</b>
B.1 Introduction . . . . .	169
B.2 Runge-Kutta d'ordre 2 . . . . .	169
B.3 Runge-Kutta d'ordre 4 . . . . .	170
B.4 Pas de temps adaptatif pour la méthode de Runge-Kutta, formules emboîtées de Runge-Kutta . . . . .	171
<b>C Données d'entrée du modèle PESHMELBA</b>	<b>173</b>
C.1 LANDSCAPE : organisation du paysage . . . . .	173
C.2 INITIAL_CONDITIONS : conditions initiales dans les éléments . . . . .	178
C.3 VEGETAL : paramètre décrivant le couvert végétal . . . . .	180
C.4 SOIL_PROPERTIES : types de sol . . . . .	182
C.5 PRACTICES : pratiques agricoles . . . . .	185
C.6 Autres . . . . .	186

# Table des figures

1	Exemple de configuration à l'échelle du bassin versant . . . . .	4
2	Interface du coupleur PALM . . . . .	5
3	Exemple de couplage entre une parcelle, une bande enherbée et une rivière	5
4	Représentation d'une parcelle ou d'une bande enherbée dans le modèle PESHMELBA. . . . .	12
5	Représentation d'un fossé dans le modèle PESHMELBA. . . . .	12
6	Représentation d'une haie, d'un talus ou d'une haie sur talus dans le modèle PESHMELBA. . . . .	13
7	Illustration de différents types de connexions possibles à l'échelle du BV. En haut à gauche : connexions latérales linéaire/linéaire, en haut à droite : connexion latérale surfacique/surfacique, en bas à gauche : connexion latérale surfacique/linéaire/surfacique, en bas à droite : connexion longitudinale linéaire vers plusieurs linéaires. . . . .	14
8	Extrait du réseau de linéaires obtenu sur le bassin de la Fontaine du Theil avec 2 exemples de réseaux discontinus. . . . .	14
9	Module FRER1D : entrées et sorties . . . . .	16
10	Articulation de 2 parcelles (module FRER1D) et de l'interface ColumnColumnExchangeLine . . . . .	17
11	Déroulement du couplage entre FRER1D et ColumnColumnExchangeLine. S est le taux de saturation (compris entre 0 et 1). . . . .	17
12	Couplage entre 2 parcelles incluant le ruissellement en surface. . . . .	18
13	Déroulement du couplage en prenant en compte le ruissellement. S est le taux de saturation (compris entre 0 et 1) et H le niveau de la flaue (m). EXCSURF désigne le module ExchangeSurface. . . . .	18
14	Exemple de relations amont-aval . . . . .	19
15	Objets élémentaires actuellement représentés dans PESHMELBA. . . . .	24
16	Classes représentant les différents éléments du paysage actuellement inclus dans PESHMELBA. . . . .	24
17	Types de lignes d'échange actuellement représentées dans PESHMELBA.	25
18	Vecteur dx par rapport à la colonne. . . . .	27
19	Exemple de chronique de pluie et pas de temps associés . . . . .	30
20	Exemple de configuration entre surfaciques et linéaires. BCS désigne les éléments BoundaryConditionStorage, F désigne les fossés, P désigne les parcelles, R désigne les rivières et H désigne les haies. . . . .	32
21	Changement de CL en surface pour FRER1D (début de simulation). . . . .	43
22	Illustration du concept de la couche d'adsorption $h_{ads}$ pour l'objet surface et de la couche de mélange pour l'objet colonne. . . . .	44

23	Exemple de courbe de conductivité basée sur les paramètres de Van Genuchten utilisée dans FRER1D. . . . .	47
24	Fonction de stress hydrique $\alpha_2$ en fonction de la pression capillaire. . .	55
25	Bande de sol disponible pour l'adsorption . . . . .	59
26	Caractéristiques géométriques d'une section trapézoïdale de brin. . . . .	59
27	Exemple de réseaux et ordre de parcours associés. . . . .	60
28	Variables WaterContent, WaterLevel_TOTAL, WaterLevel_RO . . . . .	63
29	Gradient hydraulique vs gradient topographique. Gauche : gradient hydraulique compatible avec le sens de la pente. Droite : gradient hydraulique non compatible avec le sens de la pente. . . . .	69
30	Exemple d'une nappe perchée et de la répartition du flux sortant. . . . .	70
31	Exemple de la répartition du flux sortant en cas de nappe perchée plus haute à l'aval. . . . .	70
32	Distances utilisées pour calculer la conductivité hydraulique à l'interface ( $d_x$ et $d_y$ ) et le gradient de charge ( $d_{xy}$ ). . . . .	71
33	Exemple de répartition du flux de Darcy Q entre 2 colonnes avec le système de poids. . . . .	73
34	Exemple de répartition d'un flux d'eau Q scalaire entre une colonne amont et une colonne aval. Les coefficients W sont les poids de répartition <i>weight</i> . 80	
35	Exemple de calcul d'échanges latéraux de solutés entre une colonne amont et une colonne aval. Les coefficients W sont les poids de répartition <i>weight</i> , les indices (1), (2) et (3) repèrent les cellules saturées de la colonne amont et les étoiles rouges représentent schématiquement la masse de solutés disponible dans chaque cellule numérique. . . . .	81
36	Fonctionnement hydrologique d'un fossé. a : fossé qui infiltre, b : fossé qui intercepte, c : fossé qui draine, d : fossé qui intercepte et qui transfère. Tiré de Adamiade (2004) . . . . .	82
37	Répartition des flux latéraux sortant de la colonne amont avec et sans fossé. . . . .	85
38	Répartition des nappes dans les dictionnaires water_table_dic et wt_2linear en fonction de leur position par rapport au fossé. . . . .	85
39	Décomposition du calcul des vecteurs Q_UH1, Q_UH2 et du scalaire Q_lin. . . . .	87
40	Variables utilisées dans les équations de Miles. . . . .	88
41	Configurations possibles pour le positionnement de la nappe par rapport au niveau de la rivière. . . . .	88
42	Fonctionnement complet d'une haie. . . . .	91
43	Représentation des échanges latéraux de subsurface dans une haie entre 2 parcelles. . . . .	92
44	Types de conditions limites en subsurface qu'il est possible de simuler avec un élément du type BCS. . . . .	94
45	Dictionnaire ExchangeLine_to_class_map dans la fonction initialize_interface_subsurface100	
46	Devenir du ruissellement intercepté par un fossé . . . . .	103
47	Devenir du ruissellement intercepté par une haie/talus . . . . .	103
48	Couplage surface-subsurface dans PESHMELBA. $dt_{PALM}$ désigne le pas de temps de base du modèle et $dt_{RO}$ désigne le sous pas de temps de ruissellement. . . . .	105

49	Exemple de partage du stock disponible pour le ruissellement . . . . .	105
50	Estimation d'une longueur caractéristique de ruissellement . . . . .	106
51	Exemple de configuration et calcul des débits associés . . . . .	107
52	Représentation des processus de transfert hydrique en surface d'une haie.	109
53	Bande de sol disponible pour l'adsorption sur l'objet Homogeneous Surface	112
54	Déroulement des étapes de calcul du script EXCHANGES_SURFACE	119
55	Exemple d'assemblage des modules développés dans un modèle complet à l'échelle du bassin versant. . . . .	125
56	Branche OpenPALM correspondant à l'implémentation d'un module FRER1D . . . . .	127
57	Branche OpenPALM correspondant à l'adaptation du module FRER1D	129
58	Branche OpenPALM correspondant au fonctionnement d'une UH de type parcelle . . . . .	134
59	Branche OpenPALM correspondant au fonctionnement de 2 UH de type parcelle . . . . .	135
60	Boucles FOR de la branche b1 . . . . .	135
61	Implémentation PALM gérant l'intégralité des UH de type parcelle sur le bassin versant . . . . .	137
62	Schéma PALM intégrant les optimisations. . . . .	138
63	Schéma PALM incluant les linéaires. . . . .	140
64	Chronogramme de couplage entre ruissellement sur les parcelles, transferts de surface dans les linéaires, infiltration et transferts latéraux saturés en subsurface dans les parcelles. . . . .	141
65	Schéma PALM incluant le module de ruissellement . . . . .	143
66	Différence de fonctionnement entre l'ancienne syntaxe et l'optimisation incluant l'utilisation d'une boucle infinie. . . . .	146
67	Schéma PALM après optimisation avec les boucles infinies . . . . .	147
68	Schéma PALM final . . . . .	151
69	Chronogramme du lancement des différents modules de PESHMELBA avec les flags de synchronisation. . . . .	152
70	Méthode du trapèze, tirée de <a href="#">ref runge-kutta-methode</a> . . . . .	170
71	Exemple de fichier runoff_bcsXXX.csv . . . . .	177
72	Exemple de paramètres contenus dans le fichier vegetal1.in. . . . .	181
73	Exemple de paramètres contenus dans le fichier LAI_vegetal1.in. . . . .	181
74	Exemple de paramètres contenus dans le fichier roots_vegetal1.in. . . . .	181
75	Exemple de paramètres contenus dans le fichier crop_itinerary.csv . . . . .	182
76	Fichier de paramètres soil_propertiesBC pour une colonne de sol à 3 horizons contenant et avec 2 solutés. . . . .	183
77	Fichier de paramètres soil_propertiesVG pour une colonne de sol à 2 horizons contenant 2 solutés. . . . .	185
78	Exemple de fichier practice_plot1.csv . . . . .	185
79	Exemple d'un fichier degradation.csv. . . . .	186



# Liste des tableaux

6.1	Exemple de table lateral_connection transmise à PALM . . . . .	33
6.2	Exemple de table lateral_connection_complete transmise à PALM . . . . .	33
11.1	Structure de la table surface_in_out associée à la configuration de la fig. 51 . . . . .	107
11.2	Structure de la table linear_in_out associée à la configuration de la fig. 51. Cette table étant transmise à chaque sous pas de temps, le volume indiqué correspond bien au sous pas de temps $dt_{RO}$ . . . . .	108



# Première partie

## Introduction



# Chapitre 1

## Présentation générale du projet

### 1.1 Objectifs

Dans les modèles de transfert d'eau, pesticides, et sédiments à l'échelle du bassin versant, la prise en compte des éléments du paysage, tels que les haies, les bandes enherbées, les fossés, etc... est indispensable, en particulier pour modéliser le devenir des pesticides. En effet, leur transfert est très sensible à la présence de discontinuités, qui peuvent soit les accélérer (c'est le cas des fossés dans certains cas ou des routes) soit les ralentir et les atténuer (cas des haies et des bandes enherbées). Il existe déjà des modèles de devenir des pesticides à l'échelle de la parcelle, à base empirique ou physique, mais lorsque l'on passe à l'échelle du bassin versant, les modèles ne permettent en général pas de représenter l'influence des éléments du paysage. Pourtant, un tel outil est nécessaire pour la mise en place de plans d'action pertinents et notamment l'implantation de zones tampons à l'échelle du bassin versant, pour atténuer les flux de pesticides. Ce besoin a été exprimé par le ministère de l'Agriculture et l'AFB (Agence Française pour la Biodiversité) et c'est dans ce cadre qu'a débuté le projet PESHMELBA (PESticides et Hydrologie : Modélisation à l'EcheLle du BAassin versant).

Son objectif est de développer un modèle distribué de devenir des pesticides qui prend en compte explicitement l'organisation spatiale du paysage. Celui-ci doit permettre, à terme, de tester différents scénarios d'aménagement en intégrant ou en déplaçant des éléments du paysage afin de quantifier l'influence de ces discontinuités et de leur emplacement sur le versant. Même si certaines représentations et processus sont grandement simplifiés dans un premier temps, on s'attachera d'abord à arriver à l'échelle du bassin versant, quitte à regagner en complexité par la suite. Le modèle développé sera appliqué dans un premier temps à 2 "petits" bassins versants aux caractéristiques contrastées et sur lesquels on dispose de données expérimentales (Fontaine du Theil, Ile-et-Vilaine et la Morcille, Rhône).

### 1.2 Réflexion sur l'outil de modélisation, choix d'Open-PALM

L'objectif n'est pas de développer de nouveaux modèles pour chaque élément du paysage alors que beaucoup existent déjà et sont largement validés à l'échelle locale. On cherchera plutôt à ré-utiliser des modèles existants lorsque cela est pertinent. De plus,

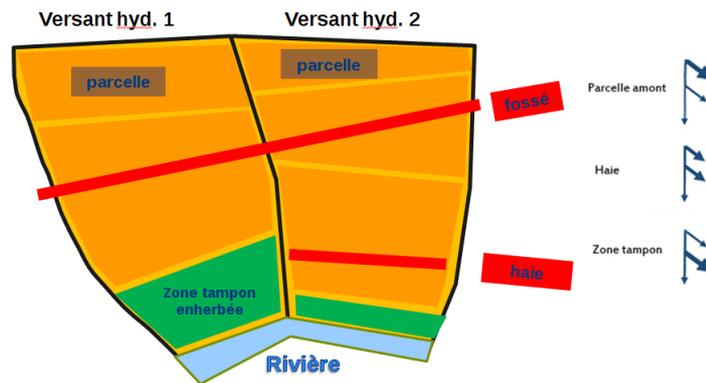


FIGURE 1 – Exemple de configuration à l'échelle du bassin versant

dans un souci de simplification, le parti est pris d'adopter des conceptualisations de complexités différentes (du modèle de réservoirs au modèle physique) pour représenter différents processus. Dans cette optique, l'utilisation d'un outil de modélisation à la structure *modulaire* pour coupler ces différentes représentations à l'échelle du bassin versant semble le choix le plus approprié. Un tel outil permet en effet la représentation du paysage et de ses éléments mais aussi la représentation des processus selon différents degrés de complexité. Cela signifie aussi qu'il doit laisser la possibilité de développer de nouveaux modules ou de modifier les existants. Ceci suppose d'utiliser un code open source et une prise en main raisonnable de l'outil pour des "non-informaticiens". Plusieurs plateformes de modélisation, sorte de "boîte à outils" pour la modélisation hydrologique et répondant en partie aux critères ci-dessus ont déjà été utilisées dans des travaux précédents. On cite entre autre les travaux de Djabelkhir (2015) et de Branger (2007b) utilisant les plateformes CMF et LIQUID respectivement. Celles-ci ont l'avantage de déjà posséder plusieurs fonctions dédiées à la modélisation hydrologique et de représenter un certain nombre de processus. Cependant, bien que les résultats aient été encourageants d'un point de vue de la démarche adoptée, ces mêmes travaux relatent aussi de difficultés majeures liées à l'utilisation de ces outils. La principale difficulté réside dans le fait que ces plateformes sont la plupart du temps difficiles à prendre en main et que l'ajout de nouveaux modules ne peut pas se faire sans un soutien significatif des développeurs. Il faut donc compter sur la réactivité et la volonté de maintenir les outils de ces derniers ce qui n'est pas toujours évident comme l'ont démontré certains projets précédents.

C'est dans cette optique que nous avons choisi de nous tourner vers le coupleur OpenPALM (Fouilloux and Piacentini, 1999) dont la flexibilité et la facilité de prise en main nous paraissent être des avantages non négligeables. Il s'agit d'un coupleur open-source développé par le CERFACS<sup>1</sup> qui se veut être un outil général pour intégrer des composants complexes de manière flexible et évolutive (Buis et al., 2006). Contrairement à CMF ou LIQUID, il ne s'agit pas d'une plateforme purement hydrologique qui dispose déjà d'un certain nombre de modules prêts à utiliser mais sa facilité de prise en main (notamment grâce à une interface simple d'utilisation, fig. 2) et les nombreuses possibilités en terme de calcul parallèle, interpolation et autres optimisations qu'il propose nous ont poussés à nous tourner vers cet outil. De plus, OpenPALM permet le couplage de plusieurs codes écrits en Fortran, Python, R ou C, offrant ainsi la possibilité d'explorer

1. Centre Européen de Recherche et Formation Avancée en Calcul Scientifique, Toulouse

les forces de chaque langage selon l'objectif à atteindre.

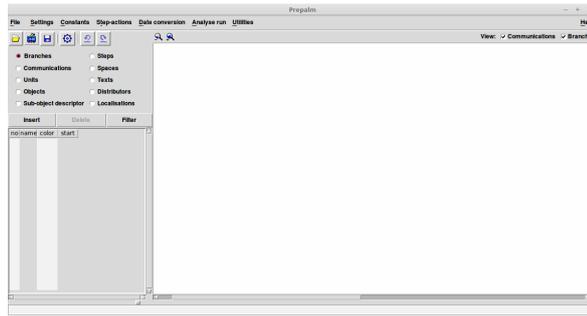


FIGURE 2 – Interface du coupleur PALM

### 1.3 Démarche envisagée

Pour arriver à un modèle distribué à l'échelle du bassin versant, la démarche adoptée consiste à modéliser les flux d'eau et de phytosanitaires :

- dans chaque élément du paysage (éventuellement avec des modèles différents) ;
- entre ces éléments avec le coupleur OpenPALM.

Un élément du paysage sera représenté dans PALM par une unité qui recevra en entrée des forçages climatiques, ainsi que des flux entrants (ou sortants) provenant d'autres unités en amont (ou en direction d'autres unités en aval) (exemple fig. 3). Ces flux et forçages sont transmis par le coupleur sous la forme de variables ou d'objets PALM. L'interface entre les unités peut se faire soit par une nouvelle unité, soit par une région de code en accord avec des règles de connexion déterminées en fonction des éléments.

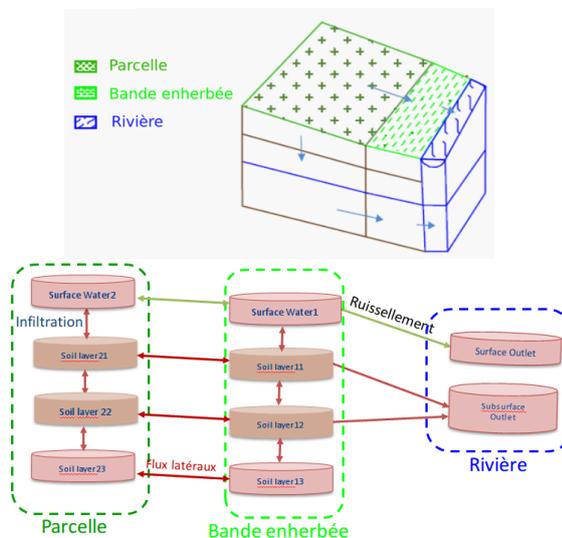


FIGURE 3 – Exemple de couplage entre une parcelle, une bande enherbée et une rivière



# Chapitre 2

## Devenir des produits phytosanitaires : aperçu des processus à représenter

Le devenir des pesticides à l'échelle du bassin versant, depuis leur application sur les parcelles jusqu'à leur transfert vers les cours d'eaux, est essentiellement régi par les processus de transfert suivants (Carluer, 1998) :

- la diffusion gazeuse dans les pores du sol et dans l'atmosphère ;
- l'adsorption, l'exsudation et la rétention par les cultures et les résidus de culture ;
- le transport hydrodynamique dans la phase aqueuse du sol ;
- le transport à la surface du sol, par l'intermédiaire du ruissellement ;
- la sorption et la désorption.

À ces 5 processus de transferts viennent s'ajouter des processus de transformation des pesticides réunis sous le terme de dégradation et regroupant des processus abiotiques et biotiques.

Dans le cadre du projet PESHMELBA, nous nous limitons aux transferts en phase solide ou dissoute et ne traiterons donc pas des phénomènes de diffusion gazeuse.

### 2.1 Transport hydrique des pesticides

Le transfert hydrique des pesticides fait intervenir des processus de convection, dus au mouvement d'ensemble de la phase liquide, de dispersion mécanique, résultant de l'hétérogénéité des champs de vitesse et de diffusion moléculaire (que l'on suppose négligeable devant la dispersion mécanique au vu des ordres de grandeur des vitesses d'écoulement) (**hypothèse à confirmer!**). On fait l'hypothèse que le flux d'eau peut être découplé du transport chimique, que ce soit pour les zones saturées ou non saturées, hypothèse justifiée par les concentrations des produits, supposées assez faibles pour pouvoir assimiler le comportement d'un fluide à celui de l'eau (Carluer, 1998). Une fois les flux d'eau déterminés, on peut donc facilement déterminer les flux de solutés.

On se base sur l'équation de convection-dispersion pour représenter le transport de

substances chimiques dans les milieux poreux :

$$\frac{\partial \theta c}{\partial t} + \vec{V} \vec{\nabla} \theta c = \vec{\nabla} \cdot (D \vec{\nabla} \theta c) \quad (2.1)$$

Avec :

- $\theta$  : teneur volumique en eau [ $L^3.L^{-3}$ ]
- $c$  : concentration du soluté dans la phase liquide [ $M.L^{-3}$ ]
- $\vec{V}$  : vitesse du fluide (flux volumique) [ $L.T^{-1}$ ]
- $D$  : tenseur de dispersion avec une composante liée à la dispersivité mécanique du milieu et une seconde liée à la diffusion moléculaire (négligée dans notre cas) [ $L^2.T^{-1}$ ]

Des termes puits/source peuvent être ajoutés à cette équation, représentant par exemple des échanges de matière liés aux phénomènes de rétention et de dégradation (Peyrard, 2016).

Il faut donc commencer par représenter correctement les flux d'eau pour représenter les solutés mis en mouvement. Pour cela, on modélisera dans PESHMELBA les processus de transferts d'eau suivants :

- les transferts d'eau verticaux dans le sol (saturés et non saturés) ;
- les transferts latéraux saturés ;
- les écoulements à surface libre : ruissellement de surface et écoulements en cours d'eau (ou fossés) ;
- l'extraction racinaire ;
- les transferts préférentiels.

Chacun de ces processus sera décrit plus précisément dans la partie II, lors de la présentation du module associé.

## 2.2 Adsorption/Désorption

L'adsorption désigne l'immobilisation des molécules de pesticides ou de leurs produits de transformation, due à des phénomènes physicochimiques. La désorption fait référence au processus inverse. Dans le cas des pesticides, on s'intéresse particulièrement à l'adsorption à l'interface liquide/solide du sol qui est la mieux connue et probablement la plus importante dans les sols cultivés (Calvet, 2005). L'adsorption peut être décrite par un isotherme d'équilibre qui, dans le cas d'adsorption à l'interface liquide/solide, représente, à une température donnée, la variation de la quantité d'une substance adsorbée sur le solide en fonction de la concentration à l'équilibre dans la phase liquide. Plusieurs formulations ont été proposées pour cet isotherme d'adsorption et les plus répandues en terme de modélisation sont l'isotherme linéaire et l'isotherme de Freundlich. Ces formulations s'appuient sur les hypothèses suivantes :

- l'équilibre d'adsorption s'établit instantanément ;
- le mécanisme est totalement réversible.

L'équilibre entre les solutés dissous et adsorbés s'exprime par la relation :

$$s = \theta c_w + \rho_b f_x(c_w) \quad (2.2)$$

Avec :

- $s$  : masse de soluté/volume de sol [M.L<sup>-3</sup>]
- $\theta$  : teneur volumique en eau (volume d'eau / volume de sol) [L<sup>-3</sup>.L<sup>-3</sup>]
- $\rho_b$  : masse volumique du sol [M.L<sup>-3</sup>]
- $f_x(c_w)$  : masse de soluté / masse de sol [M.M<sup>-1</sup>]

Et on a la relation suivante pour l'isotherme de Freundlich (Freundlich, 1909) :

$$f_x(c_w) = K_d c_w^n \quad (2.3)$$

Avec :

- $K_d$  [L<sup>3n</sup>.M<sup>-n</sup>] le coefficient d'adsorption
- $n$  le coefficient de l'isotherme de Freundlich. Si  $n = 1$ , on a la relation de l'isotherme d'adsorption linéaire.

Le coefficient d'adsorption  $K_d$  dépendant fortement de la teneur en carbone organique du substrat, on utilise souvent une expression normalisée de ce dernier :

$$K_{oc} = \frac{100K_d}{\%OC} \quad (2.4)$$

avec %OC le pourcentage en carbone organique du matériau.

## 2.3 Dégradation

La dégradation est un des principaux processus intervenant dans la dissipation et l'élimination des pesticides dans les milieux naturels. De nombreuses transformations chimiques modifient la composition et la structure des molécules aboutissant à leur dégradation. Cependant, ces processus de transformation peuvent aussi aboutir à la formation de toute une série de produits intermédiaires de transformation aux propriétés de rétention et de toxicité différentes. Le processus de dégradation peut d'ailleurs ne pas conduire à la dissipation totale de la substance active mais s'arrêter en produisant une de ces molécules intermédiaires, appelée métabolite qui lui ne sera pas dégradé et dont les effets sont parfois plus délétères que ceux de la molécule mère.

Les principales réactions conduisant à la dégradation sont (Calvet, 2005) :

- la photolyse ;
- l'hydrolyse ;
- les mécanismes biotiques (dominant dans la zone biologiquement réactive du sol).

C'est souvent une succession complexe de réactions abiotiques et biotiques qui aboutit à la dégradation d'une molécule. Pour représenter simplement l'ensemble de ces mécanismes complexes, on s'en tient généralement à décrire la vitesse de dégradation via une

cinétique de premier ordre. Ainsi, on peut comparer les molécules entre elles grâce au temps de demi-vie, durée nécessaire pour qu'une quantité initiale soit divisée par deux :

$$C(t_{1/2}) = \frac{C_0}{2} \quad (2.5)$$

où  $C_0$  [ML<sup>-3</sup>] est la concentration en pesticides dans le sol à l'instant initial et  $C(t_{1/2})$  [ML<sup>-3</sup>] la concentration au temps de demi-vie. La cinétique d'ordre 1 s'exprime alors :

$$C(t) = C_0 e^{-tk} \quad (2.6)$$

Avec :

- $C_0$  : concentration en pesticide à l'instant initial [M.L<sup>-3</sup>];
- $k$  : constante de vitesse du premier ordre [T<sup>-1</sup>] reliée au temps de demi-vie par la relation :  $t_{1/2} = \frac{\ln 2}{k}$

# Chapitre 3

## Éléments et connexions à représenter

Pour atteindre l'objectif de modélisation du devenir des pesticides, on insistait plus haut sur l'importance de représenter la diversité d'éléments constituant un paysage agricole et les connexions entre eux. La description de l'organisation du bassin à modéliser est donnée par l'outil GeoMELBA au terme de la première étape de pre-processing. Cet outil permet de générer le maillage sur lequel est appliqué PESHMELBA à partir du croisement de couches SIG décrivant le parcellaire, les types de sol et les différentes structures paysagères. L'ensemble de ces données est réuni sous forme de tables au format .csv décrites en détails dans l'Annexe C.

### 3.1 Types d'éléments à modéliser

GeoMELBA fournit un maillage composé d'éléments **surficiels** et **linéaires**. Dans la présente version de ce manuel, on détaille l'implémentation des éléments surficiels de type parcelles et bandes enherbées et des linéaires de type fossés, rivières, haie, talus et haies sur talus. Un élément BoundaryConditionStorage (BCS) a aussi été implémenté afin de disposer de conditions limites réalistes lors des tests indépendants de certains éléments ou assemblages d'éléments.

Des représentations différentes sont choisies pour chaque type d'élément en fonction des processus que l'on souhaite modéliser. Elles sont parfois basiques mais pourront évoluer si l'on veut représenter de nouveaux processus ou représenter de manière plus fine un processus donné. Ces représentations ont été choisies à partir de la connaissance des processus que l'on a sur les bassins de la Fontaine du Theil et de la Morcille aux fonctionnements très contrastés.

#### 3.1.1 Parcelles et Bandes enherbées

Les parcelles et les bandes enherbées sont représentées de la même manière, par une colonne de sol divisée en cellules numériques (fig. 4). Cette colonne est caractérisée par une position, une hauteur, une surface et un développement de profil de sol.

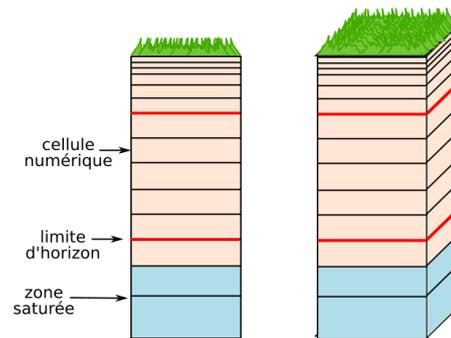


FIGURE 4 – Représentation d’une parcelle ou d’une bande enherbée dans le modèle PESHMELBA.

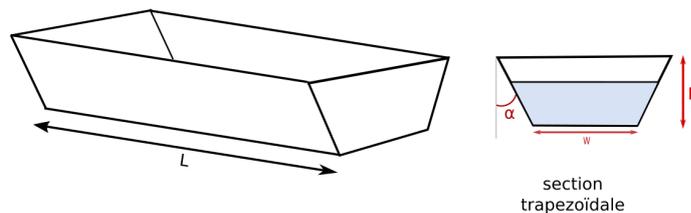


FIGURE 5 – Représentation d’un fossé dans le modèle PESHMELBA.

### 3.1.2 Fossés et rivières

Un fossé ou une rivière est représenté par un réservoir unique, à section trapézoïdale (fig. 5) et qui possède les attributs suivants :

- Longueur ;
- Largeur de fond ;
- Profondeur ;
- Fruit des berges ;
- Pente moyenne longitudinale.

Les fossés et rivières sont également caractérisés par un substrat (teneur en eau à saturation, densité volumique, ...) qui permet la description de l’adsorption des solutés.

### 3.1.3 Haies, talus

Une haie, un talus ou une haie sur talus est représentée en 2 parties, une haie "amont" et une haie "aval" chacune représentée par un réservoir de surface et une colonne de sol divisée en cellules numériques. Les parties amont et aval de l’élément sont séparées par un mur dans les cas où la haie est implantée sur un talus (voir fig. 6).

Cette représentation en 2 éléments séparés permet de faciliter la représentation des processus d’infiltration (voir chapitre 7) et de distinguer de jeunes haies de haies plus anciennes sur lesquelles apparaissent de vrais décrochages topographiques entre les parties amont et aval. L’élément complet possède les attributs suivants :

- Longueur ;
- Largeurs amont et aval ;
- Pente moyenne longitudinale ;

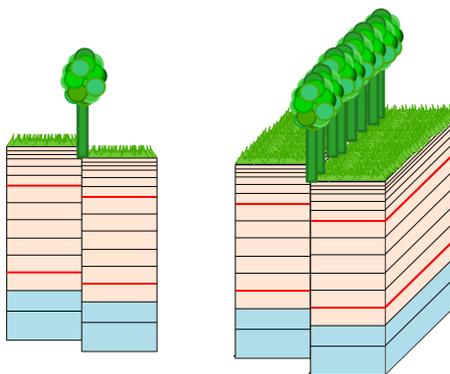


FIGURE 6 – Représentation d’une haie, d’un talus ou d’une haie sur talus dans le modèle PESHMELBA.

- Pentes moyennes transversales amont et aval ;
- Inclinaison par rapport à la ligne de niveau locale ;
- Décrochage entre la partie amont et la partie aval ;
- Hauteur de talus ;
- Type de végétation ;
- Type de sol.

### 3.1.4 Boundary Condition Storage (BCS)

Les éléments de type BCS se connectent en amont ou en aval d’éléments surfaciques (parcelle, bande enherbée). Ils peuvent fournir, selon leur position, différentes conditions limites pour l’eau et les solutés, pour les compartiments de subsurface et de surface et servent à tester la représentation d’autres éléments de façon suffisamment réaliste. Ils sont constitués d’un réservoir de surface et d’un réservoir de subsurface.

## 3.2 Topologie

En plus d’une description des différents éléments qui composent le bassin, GeoMELBA fournit aussi une description des connexions entre eux. Ce sont ces connexions qui permettront de rendre compte du fonctionnement hydrologique du bassin. Les linéaires agissent comme des discontinuités vis à vis du fonctionnement hydrologique du bassin versant et il faut bien rendre compte de ce rôle de redirection des flux. Les connexions entre éléments peuvent être de différents types. Il peut s’agir de connexions latérales (connexion entre 2 éléments surfaciques, entre un élément surfacique et un linéaire ou entre 2 linéaires) ou de connexions longitudinales (entre 2 linéaires). Un élément peut-être connecté en latéral à un ou plusieurs élément(s) surfacique(s) (parcelle, bande enherbée,...) ou linéaire(s). Un linéaire peut également être connecté à un ou plusieurs autre(s) linéaire(s) dans le sens longitudinal. Un exemple simple de configuration entre parcelles, bandes enherbées, fossés et talus est présenté fig. 7 pour représenter les différents types de connexions que GeoMELBA peut représenter et que PESHMELBA sera amené à gérer.

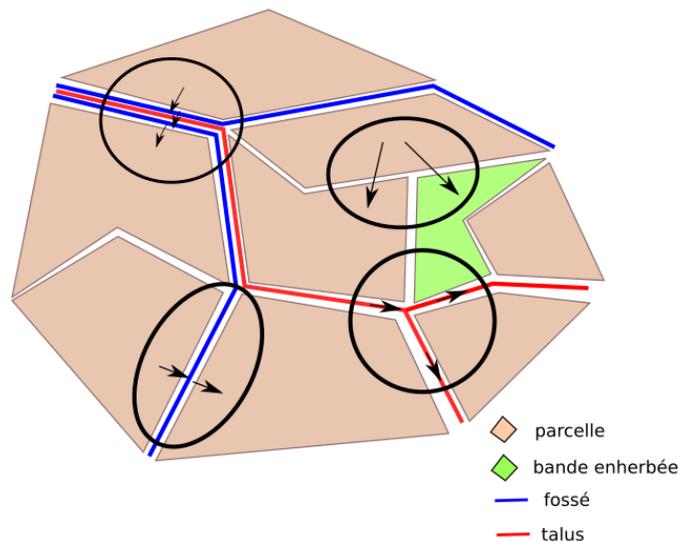


FIGURE 7 – Illustration de différents types de connexions possibles à l'échelle du BV. En haut à gauche : connexions latérales linéaire/linéaire, en haut à droite : connexion latérale surfacique/surfacique, en bas à gauche : connexion latérale surfacique/linéaire/surfacique, en bas à droite : connexion longitudinale linéaire vers plusieurs linéaires.

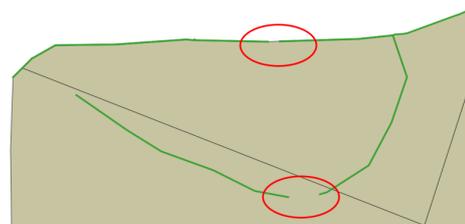


FIGURE 8 – Extrait du réseau de linéaires obtenu sur le bassin de la Fontaine du Theil avec 2 exemples de réseaux discontinus.

On attire également l'attention sur le fait qu'en plus des connexions présentées fig. 7 et qui balaient les cas rencontrés sur un bassin à la configuration idéale, PESHMELBA sera également amené à gérer des cas plus dysfonctionnels. Ces cas rendent par exemple compte de trous dans les réseaux de linéaires, réellement existants ou dus à des données de mauvaise qualité fournies à GeoMELBA. Un exemple est présenté fig. 8. Dans ces cas là, PESHMELBA doit être capable de repérer ces trous et de modéliser les écoulements auxquels ils aboutiront (déversement dans l'unité homogène la plus proche).

# Chapitre 4

## Problématiques relatives au couplage

La modélisation spatialisée du devenir des pesticides suppose d’aborder avec attention les questions de gestion de l’aspect spatial et de gestion du pas de temps. Avant de commencer le couplage à proprement parler, il faut définir quelles sont les possibilités offertes par OpenPALM et la stratégie à adopter.

### 4.1 Gestion de l’aspect spatial du modèle

L’outil de modélisation doit avant tout être capable de faire le lien entre les éléments linéaires (route, fossés,...) et surfaciques (parcelle, prairie, bande enherbée,...) qui définissent des domaines de modélisation et les unités PALM qui les représenteront. Ces éléments proviendront de traitement géomatiques et seront contenus dans des shapefiles ou des fichiers .csv. Un ensemble de caractéristiques sera attaché à chacun de ces éléments et sera utilisé comme entrée pour le modèle.

D’autre part, le couplage spatial entre ces éléments doit permettre de représenter à chaque pas de temps, en simultané :

- Les forçages qui adviennent à l’échelle globale du bassin et à une échelle plus locale ;
- Les processus physiques advenant pour chaque élément du versant ;
- Les connexions et les transferts entre les différents éléments.

Le premier point est fondamental et arriver à conjuguer couplage local et global est un aspect auquel il faudra prêter une attention toute particulière. Par exemple, un forçage climatique (pluie) sera appliqué sur l’ensemble du bassin. Une telle pluie peut entraîner très rapidement un flux de ruissellement sur une parcelle P1 si les caractéristiques topographiques et physiques de la parcelle le permettent. Ce ruissellement peut-être rapidement transféré vers une parcelle P2 et deviendra un forçage local pour cette dernière dès le pas de temps suivant. Il faut donc être capable de gérer pour cette seconde parcelle, 2 forçages advenant à des échelles différentes et avec des pas de temps différents également.

La représentation du ruissellement sur une unité surfacique illustre les difficultés relatives à l'aspect spatial du modèle : le ruissellement, modélisé dans notre cas par l'équation de l'onde cinématique est classiquement représenté avec un maillage horizontal assez fin et doit être couplé aux transferts verticaux dans une unité dont le maillage horizontal n'est composé que d'une maille. Dans ce cas, une stratégie doit alors être trouvée pour gérer le couplage entre ces 2 processus modélisés sur des maillages différents. Une autre approche peut aussi consister à modifier la représentation du ruissellement pour que le maillage horizontal coïncide avec le découpage du paysage et les éléments verticaux sur lesquels est représentée l'infiltration. Nous verrons par la suite que la seconde solution est plus envisageable dans le cas de PESHMELBA.

## 4.2 Gestion du pas de temps

Chaque processus physique ou module représenté possède un pas de temps différent pour tenir compte des dynamiques propres de chaque processus. Certains modules possèdent même un pas de temps "adaptatif" (i.e. le pas de temps suivant est réévalué à chaque instant pour être optimisé). Dans ce cas-là, le coupleur doit assurer la synchronisation des différentes unités.

L'exemple suivant permet d'illustrer une telle situation avec un cas de couplage simple entre deux parcelles. Il permet d'illustrer simplement les problématiques liées au couplage et les solutions imaginées. Une parcelle est modélisée par le module FRER1D permettant de simuler les transferts d'eau verticaux dans des sols hétérogènes (voir Chapitre 7). Le module calcule à chaque pas de temps, en fonction des forçages (pluie, ruissellement en surface ou forçage latéral) reçus en entrée et des propriétés du sol, ses variables d'état (notamment le taux de saturation  $S$ ) et détermine son prochain pas de temps.

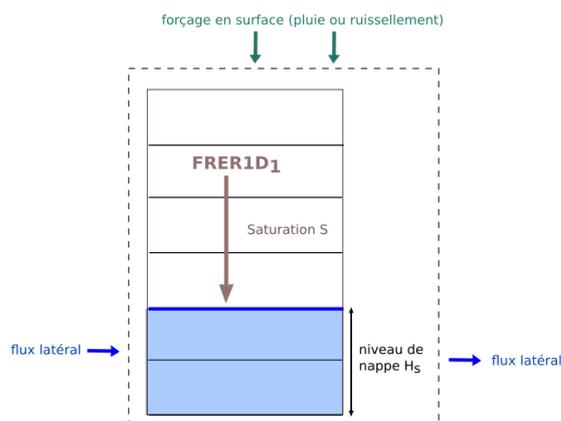


FIGURE 9 – Module FRER1D : entrées et sorties

Les transferts latéraux entre 2 parcelles se font uniquement au niveau des zones saturées de la colonne de sol (nappe ou nappe perchée). S'il existe un gradient de hauteurs de nappes entre 2 colonnes modélisées par FRER1D, une ligne d'échange ColumnColumnExchangeLine (désignée CCEX dans ce qui suit) (voir chapitre 10.2) calcule le flux d'eau qui en résultera suivant la loi de Darcy.

Un pas de temps global doit être géré par PALM et à la fin de chacun de ces pas de temps, l'unité de ligne d'échange est activée pour inspecter le profil de saturation, en

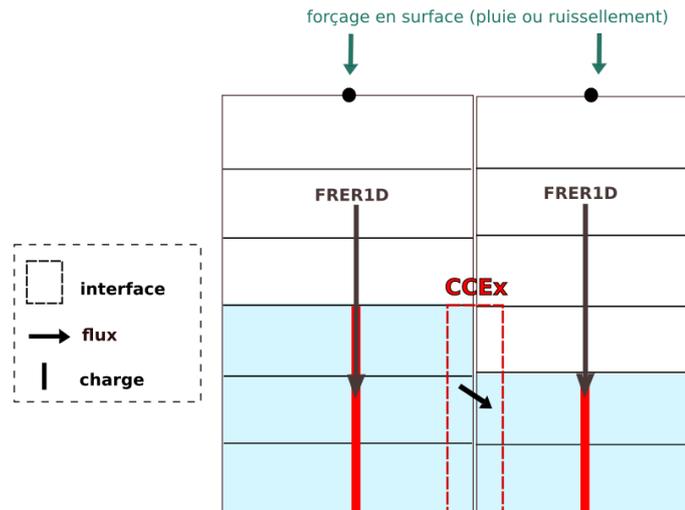


FIGURE 10 – Articulation de 2 parcelles (module FRER1D) et de l'interface ColumnColumnExchangeLine .

déduire les hauteurs de nappe de chaque élément et les gradients entre eux avant de calculer éventuellement les flux latéraux résultants. Au pas de temps suivant, chaque module FRER1D doit intégrer ces flux latéraux dans ses forçages en plus des éventuels forçages de surface comme détaillé sur la fig. 11.

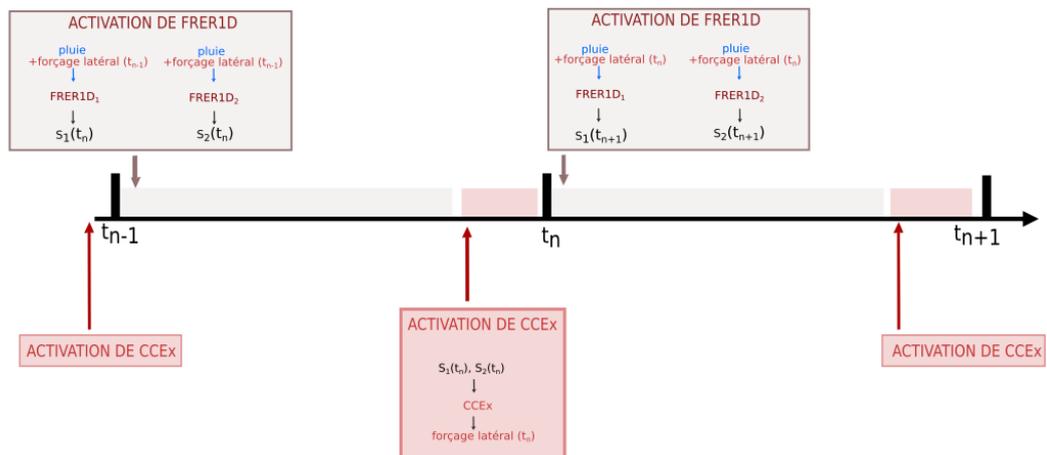


FIGURE 11 – Déroulement du couplage entre FRER1D et ColumnColumnExchangeLine. S est le taux de saturation (compris entre 0 et 1).

Pour compléter la représentation des transferts entre deux parcelles, rajoutons la composante de ruissellement en surface, caractérisé par une dynamique plus rapide. On prévoit que celui-ci soit représenté par le module ExchangeSurface basé sur l'onde cinématique et qui ne dépendra donc que des caractéristiques de la parcelle courante, sans tenir compte de la parcelle en aval. Ce module servira pour calculer la composante de ruissellement sortant d'un élément et ensuite pour répartir ce ruissellement vers les éléments connectés à l'aval. Dans le déroulement de la simulation, au cours d'un pas de temps PALM, les forçages climatiques seront d'abord calculés et envoyés à FRER1D. Celui-ci prendra en entrée ces flux verticaux ainsi que d'éventuels flux latéraux calculés à la fin du pas de temps précédent. Le rééquilibrage vertical de FRER1D pouvant conduire

à modifier le niveau des flaques en surface, le module ExchangeSurface interviendra ensuite pour calculer et répartir le ruissellement correspondant, comme représenté sur le chronogramme 13. A la fin du pas de temps, la ligne d'échange CCEX interviendra pour calculer les flux latéraux qui pourraient apparaître entre les différentes unités. Alors que la ligne d'échange CCEX se déclenchera à la fin de chaque pas de temps PALM, les unités FRER1D et ExchangeSurface posséderont leur propres pas de temps et le coupleur devra permettre de les synchroniser.

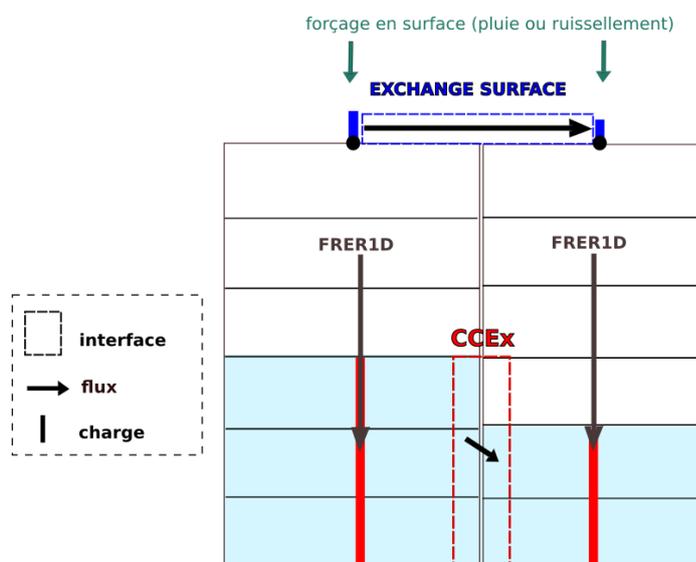


FIGURE 12 – Couplage entre 2 parcelles incluant le ruissellement en surface.

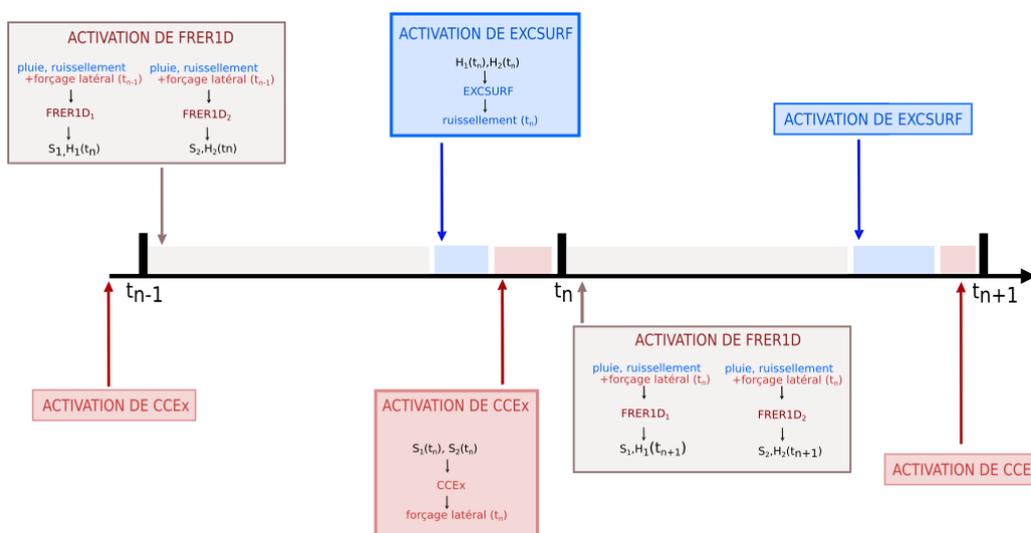


FIGURE 13 – Déroulement du couplage en prenant en compte le ruissellement.  $S$  est le taux de saturation (compris entre 0 et 1) et  $H$  le niveau de la flaque (m). EXCSURF désigne le module ExchangeSurface.

### 4.3 Fonctionnement des lignes d'échange

Que ça soit ColumnColumnExchangeLine pour la subsurface ou ExchangeSurface pour la surface, ces deux unités ont le rôle d'interface et doivent permettre le transfert de flux d'une parcelle à l'autre. Pour représenter fidèlement le bassin versant, on peut imaginer rencontrer des configurations où un élément transfère son flux à plusieurs ou au contraire, qu'une parcelle reçoit le flux provenant de plusieurs éléments. Il sera donc nécessaire de définir des relations amont/aval pour les parcelles. Pour chacune de ces relations, un test est fait sur la hauteur de nappe et la ligne d'échange CCEX calcule le flux résultant si besoin en prenant en compte la longueur de contact entre les deux éléments. Chacun de ces flux calculés contribue au flux total entrant/sortant de la parcelle et qui sera donné en entrée de FRER1D comme forçage latéral comme illustré sur la fig.14.

CCEX peut donc être conçu de 2 manières :

- Soit un module basique qui ne calcule que le flux entre deux éléments donnés en entrée. L'identification de toutes les entrées/sorties relatives à une parcelle doit être faite par ailleurs (par PALM par exemple qui fournirait les longueurs, les hauteurs de nappe, etc...).
- Soit un module un peu plus complexe qui est lui-même capable d'identifier tous les éléments qui sont en aval d'une parcelle. Dans ce cas-là, un seul appel à CCEX doit être capable de gérer le calcul de tous les flux dans le cas de gauche de la fig. 14 (avec plusieurs appels encapsulés pour gérer chacun des couples) mais il faudra quand même 2 appels à CCEX pour le cas de droite (un seul élément en aval de a et un seul élément en aval de b)

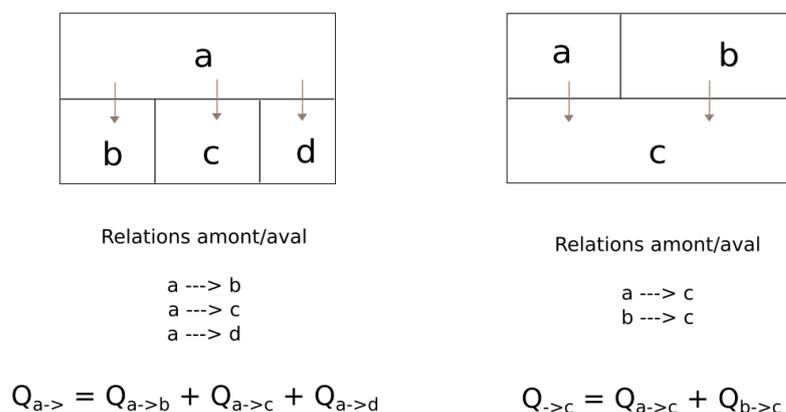


FIGURE 14 – Exemple de relations amont-aval

La première option a finalement été choisie pour tous les types de lignes d'échange : un premier module d'analyse du paysage est utilisé pour identifier toutes les connexions entre éléments puis le module de ligne d'échange est appelé pour chacune de ces connexions. L'implémentation de ces différents modules est décrite dans la partie II.



Deuxième partie

Développement des différents  
modules



# Chapitre 5

## Propos généraux sur l'implémentation des modules dans PESHMELBA

Pour modéliser les différents processus décrivant le devenir des pesticides, PESHMELBA se base sur un ensemble de modules présentés dans cette partie. Certains, existants auparavant (module d'infiltration verticale ou extraction racinaire) ont été récupérés et seulement adaptés pour les besoins du modèle, notamment en ce qui concerne les échanges de variables et la gestion du pas de temps. L'utilisation du coupleur OpenPALM permet d'établir des communications entre ces modules indépendants, éventuellement écrits dans des langages différents. Cela permet de conserver le formalisme et la structure des codes pré-existants et d'en établir éventuellement de nouveaux pour chaque module développé. Même si tous les codes ne sont pas homogènes en terme de niveau de complexité, de formalisme et de syntaxe, OpenPALM permet de construire un ensemble cohérent.

Ainsi, chaque module nouvellement développé aurait pu être codé indépendamment mais on a choisi de mettre en place une structure commune de classes afin de pouvoir réutiliser facilement certaines fonctions d'un module à l'autre. Un diagramme UML a aussi été construit pour décrire chaque élément du paysage et chaque type de ligne d'échange. Il exploite notamment les notions d'héritage et de composition et se base sur un ensemble de briques élémentaires qui une fois assemblées, décriront une parcelle, un fossé, etc... Par exemple, une parcelle est composée d'une colonne de sol, d'une surface, d'un type de végétation et de la description des applications de pesticides qui sont faites sur cette parcelle. En conséquence, les variables PALM qui sont transmises entre modules sont séparées selon la classe à laquelle elles font référence : dans la mesure du possible, tous les attributs de la classe Element seront transmis dans un objet Element\_o, ceux de la classe HomogeneousSurface dans un objet Surface\_o, etc... Une telle description peut paraître laborieuse au premier abord mais elle permet d'optimiser les échanges de variables entre modules. Par exemple, dans le cas de la description d'une parcelle, les mêmes variables d'état, paramètres et forçages ne seront pas nécessaires lorsqu'on voudra décrire le ruissellement de surface ou les échanges latéraux saturés en subsurface. Ce découpage permet de compartimenter les variables à récupérer ou à envoyer pour coder les modules relatifs à chacun de ces processus. Il se base sur 3 types d'objets de base qui héritent de la classe Element : les éléments

ponctuels, les éléments surfaciques et les éléments linéaires. Chacun de ces éléments se décline ensuite pour décrire soit un objet de surface (homogeneous surface, surface tank) soit un objet de subsurface (colonne) soit un objet à cheval sur les deux (bcs, reach). L'ensemble des objets actuellement représentés dans PESHMELBA est résumé fig. 15. Cette classification est bien sûr amenée à évoluer si besoin.

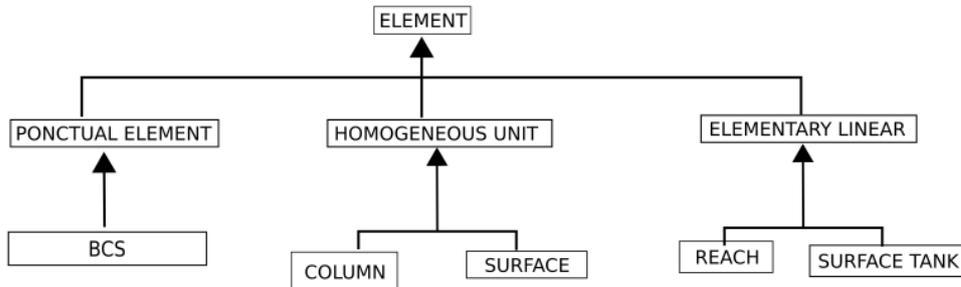


FIGURE 15 – Objets élémentaires actuellement représentés dans PESHMELBA.

Ensuite, chacun des éléments du paysage est décrit comme la composition de ces éléments de base (voir fig. 16). On note que l'objet Network a été créé pour disposer d'un élément générique permettant de traiter indifféremment des réseaux de rivières ou de fossés. C'est pour cela qu'il est composé directement d'instances de la classe Reach et non pas d'instances de Ditch ou River.

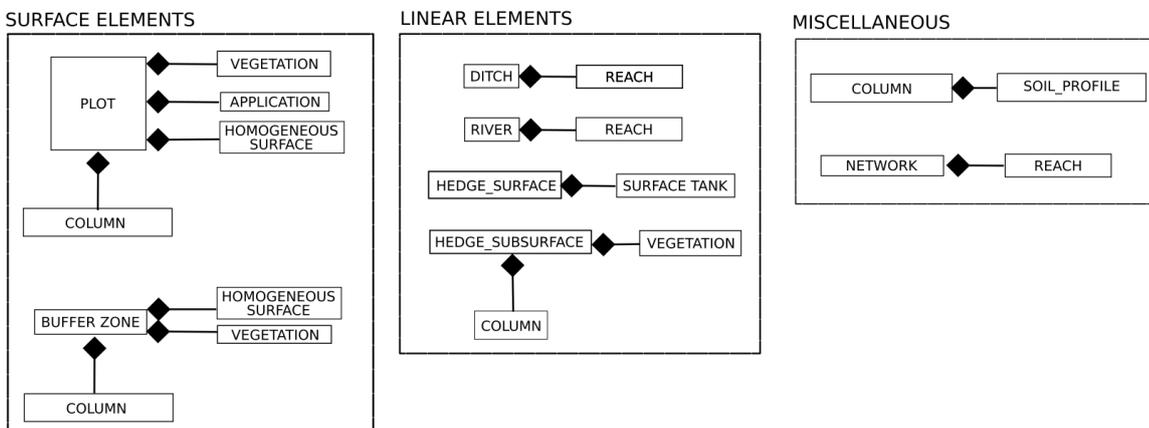


FIGURE 16 – Classes représentant les différents éléments du paysage actuellement inclus dans PESHMELBA.

Enfin, chaque type de ligne d'échange hérite des classes `Interface_surface` ou `Interface_subsurface` (fig. 17). Ces niveaux d'héritage permettront d'introduire la notion de polymorphisme dans les codes et ainsi d'augmenter la généricité de l'application finale. Ces notions seront détaillées dans les parties relatives à chaque processus.

On présente ici quelques classes de base dont héritent ou dont sont composées l'ensemble des sous-classes présentées dans ce rapport.

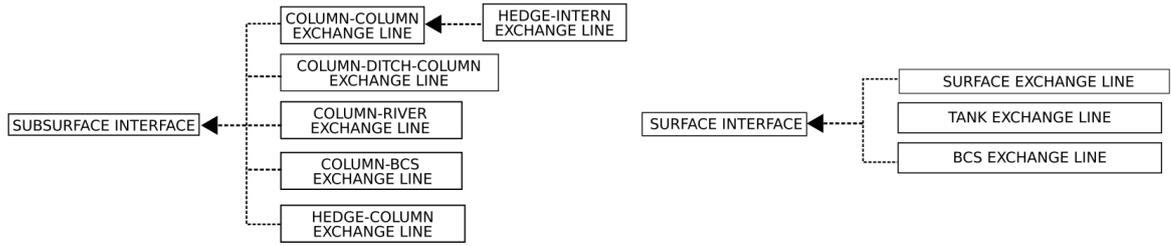


FIGURE 17 – Types de lignes d'échange actuellement représentées dans PESHMELBA.

## Classe Element

La classe Element est la classe dont héritent tous les éléments du paysage représentés dans PESHMELBA. C'est la description la plus basique que l'on peut faire d'un objet. Elle est initialisée à partir de :

- idx (int) : un index d'identification ;
- lon (double) : une longitude [L] ;
- lat (double) : une latitude [L] ;
- alt (double) : une altitude [L].

## Classe HomogeneousUnit

La classe HomogeneousUnit hérite de la classe Element et permet de représenter tout objet élémentaire de type surfacique. Elle s'initialise à partir du constructeur de la classe Element et un attribut area décrivant la surface de l'élément [L<sup>2</sup>].

## Classe PonctualElement

La classe PonctualElement hérite de la classe Element et permet de représenter tout objet élémentaire ponctuel. Elle sert par exemple pour la représentation d'objets de type Condition Limite (BCS) ou à terme pour la représentation d'ouvrages ponctuels sur un réseau. Pour le moment, elle n'a pas grande utilité puisqu'elle s'initialise exactement comme la classe Element dont elle hérite. Si elle ne prouve pas son utilité, à terme elle pourra être supprimée.

## Classe ElementaryLinear

La classe ElementaryLinear hérite de la classe Element. Son unique méthode est un constructeur qui se base entre autre sur le constructeur de la classe Element. Une instance de la classe ElementaryLinear est initialisée à partir des attributs suivants :

- x (double) : longitude du centre du linéaire [L] ;
- y (double) : latitude du centre du linéaire [L] ;
- z (double) : altitude moyenne du linéaire [L] ;

- length (double) : longueur du linéaire **corde ou longueur réelle du coup** ? [L] ;
- width (double) : largeur du linéaire [L]. Dans le cas d'un fossé ou d'une rivière, il s'agit de la largeur du fond du linéaire.

## Classe SoilProfile

La classe SoilProfile donne les caractéristiques d'un développement de profil associé à une objet de type colonne de sol. Elle s'initialise comme suit :

- n (integer) : nombre de cellules numériques dans la colonne de sol ;
- dx (double(1 :n)) : vecteur contenant les profondeurs de chaque cellule numérique [L] ;
- interface (integer(1 :n)) : vecteur contenant le type de sol associé à chaque cellule numérique ;
- he (double(1 :n)) : vecteur contenant la pression d'entrée d'air dans chaque cellule numérique (si description des propriétés hydrodynamiques du sol utilisant les relations de Brooks & Corey. Le cas échéant, tous les éléments du vecteur sont fixés à zéro) [L] ;
- Ksat (double(1 :n)) : vecteur contenant la conductivité hydraulique à saturation dans chaque cellule numérique [ $L.T^{-1}$ ] ;
- ratio (integer) : coefficient de proportionnalité entre Ksat horizontal et vertical ( $Ksat_{xy} \setminus Ksat_z$ ) ;
- nb\_st (integer) : nombre d'horizons de sol dans la colonne ;
- thick (double(1 :n)) : vecteur contenant l'épaisseur de chaque couche numérique [L].

Le vecteur dx fournit la profondeur de chaque cellule numérique par rapport à la **surface** (attention aux confusions!). Un exemple est présenté Figure 18.

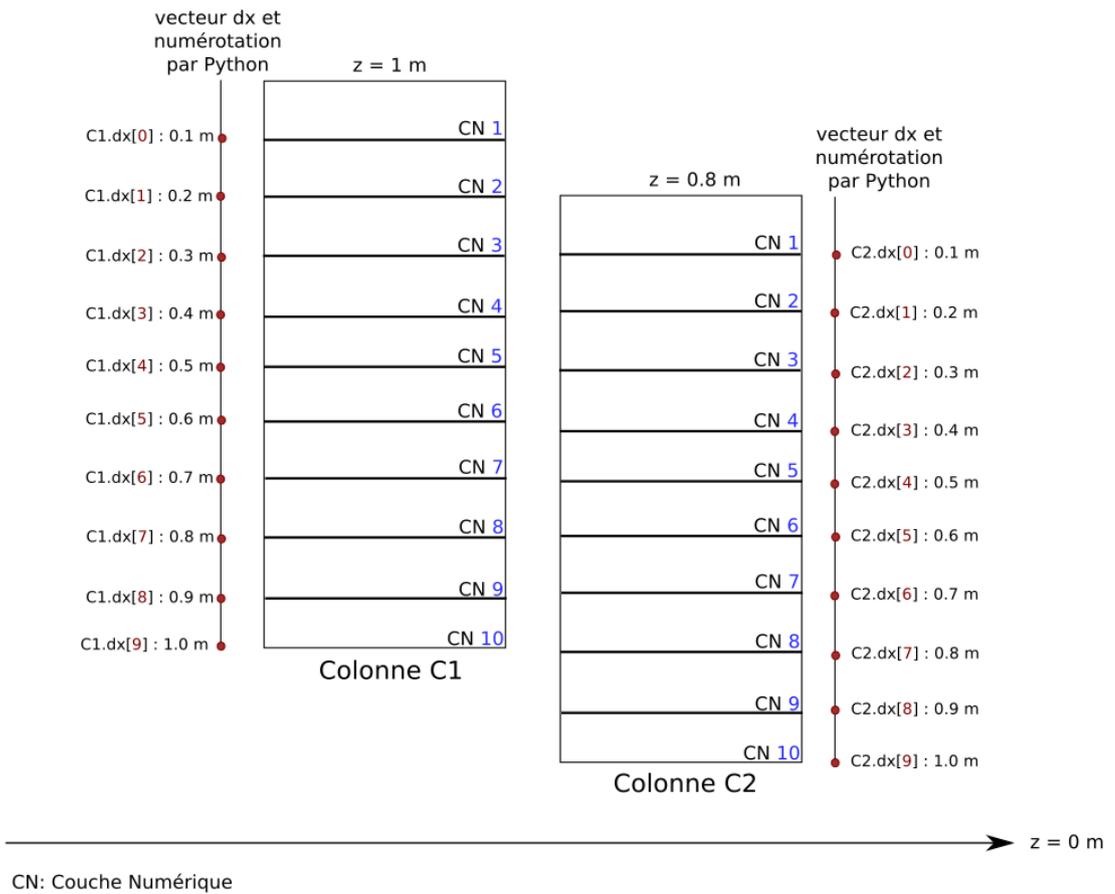


FIGURE 18 – Vecteur dx par rapport à la colonne.



# Chapitre 6

## Pre-processing

Un premier module est nécessaire pour initialiser les différentes variables qui seront envoyées à PALM et pour importer les paramètres de la simulation depuis les fichiers d'entrée. C'est le module `init_CONF` qui gère cette initialisation.

### 6.1 Importation des données d'entrée et initialisation des variables PALM

Dans un premier temps, les paramètres de la simulation, les forçages et les propriétés de chaque élément du paysage sont importés depuis les tables fournies par GeoMELBA. Ces tables permettent d'initialiser les différents objets manipulés par PESHMELBA. Les paramètres et les variables d'état sont stockés dans des objets ou des variables PALM qui seront échangés entre modules par l'intermédiaire du coupleur OpenPALM. Les objets suivants sont ainsi initialisés :

- L'organisation spatiale du paysage : liste des parcelles, des linéaires et des éléments fictifs (éléments de conditions limites type BCS) composant le maillage, table des connectivités entre éléments ;
- Les caractéristiques des différents types de sol : nombre d'horizons et propriétés hydrauliques du sol ;
- La géométrie et les propriétés de chaque Unité Homogène (UH) et de chaque linéaire ;
- Les propriétés des solutés appliqués (paramètres d'adsorption et de dégradation) ;
- Les propriétés des autres éléments (BCS,...).

Une table présentant l'ensemble des variables PALM initialisées dans ce module puis échangées dans PESHMELBA est présentée en Annexe [A](#).

### 6.2 Initialisation des pas de temps et des séries temporelles

Dans PESHMELBA, le pas de temps n'est pas imposé par l'utilisateur pour toute la durée de la simulation. Il est modulable afin de pouvoir disposer d'un pas de temps variable selon les forçages climatiques. Alors qu'un pas de temps assez fin est souhaitable

pendant les périodes de pluie ou de ressuyage de nappe, il peut être choisi plus grossier pendant les périodes sèches (voir fig. 19). Les valeurs des pas de temps de base et raffinés sont choisies par l'utilisateur en fonction du bassin versant et de la durée totale de la modélisation. L'utilisation de l'un ou de l'autre se décide en analysant la chronique de pluie à chaque instant. Cette analyse est faite en une seule fois dans le module d'initialisation. Le vecteur contenant tous les temps de simulation est alors stocké dans la variable PALM *alltime\_vect* puis transmis à chaque module qui met à jour son temps de simulation en parcourant ce vecteur.

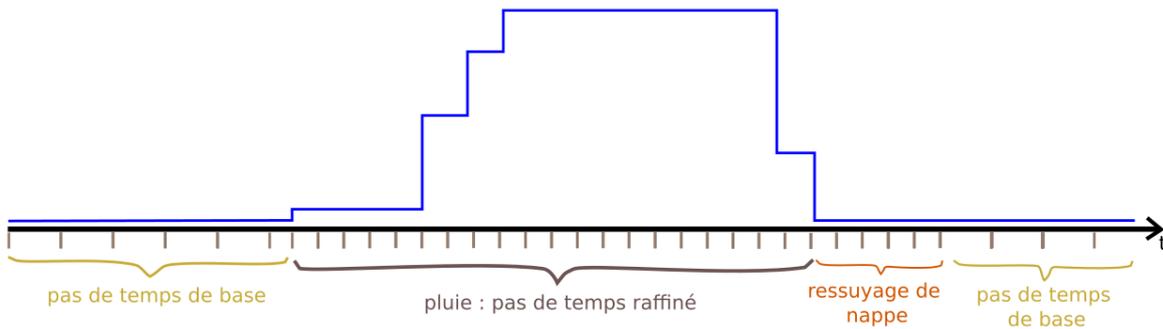


FIGURE 19 – Exemple de chronique de pluie et pas de temps associés

Dans un second temps, on peut interpoler les paramètres végétaux pour lesquels on dispose de séries temporelles (LAI, profondeur racinaire  $Z_r$ , fraction de densité racinaire présente dans les 10 % supérieurs de la zone racinaire  $F_{10}$ , voir détails chapitre 8) aux échéances de temps stockées dans le vecteur *alltime\_vect*. On dispose en effet de séries temporelles renseignant l'évolution du couvert végétal mais sans forcément disposer de mesures à chacun des temps de simulation du vecteur *alltime\_vect*. Une étape supplémentaire consiste donc à récupérer les paramètres végétaux existants et à les interpoler sur les valeurs de temps de *alltime\_vect*. Pour cela, on considère que chaque mesure d'entrée correspond à un changement d'état du couvert végétal. Cette hypothèse utilisée dans le script devra peut-être être revue en fonction des données d'entrée disponibles. Enfin, on détermine également les applications de pesticides associées à chaque instant de simulation.

C'est le bloc "temporal\_pre\_processing" qui gère la partie de détermination des pas de temps et des séries temporelles relatives aux végétaux dans le module d'initialisation. Il se présente sous la forme d'une boucle TANT QUE  $t < \text{fin}$  de la simulation. A chaque passage dans la boucle, les actions suivantes sont réalisées :

- Au temps  $t$ , détermination du type de végétation *vegetype* présent sur chaque parcelle et sur chaque haie (car on considère que le type de couvert végétal peut changer au cours de la simulation) ;
- Lecture des paramètres ne variant pas temporellement pour le type de végétation *vegetype* : coefficient cultural, coefficient d'extinction, paramètre  $\lambda$  pour le calcul de la fonction de compensation, pressions capillaires et transpirations potentielles associées aux pressions capillaires ;
- Interpolation des paramètres variant temporellement à l'instant  $t$  (LAI,  $Z_r$ ,  $F_{10}$ ) ;
- Calcul à l'instant  $t$  des forçages climatiques *qprec* et *qevap* ;

- Détermination des applications de pesticides (masse de solutés et volume d'eau associé) à la date  $t$  ;
- Détermination du pas de temps suivant en fonction des événements climatiques à venir. Pour cela, la fonction explore la chronique de pluie pour déterminer si le temps  $t$  correspond à un épisode sec, de pluie, de ressuyage de nappe ou de transition (passage d'une période de pluie à une période sèche ou d'une période sèche à une période de pluie). Dans ce dernier cas, on autorise l'adaptation du pas de temps pour se caler sur le temps où le changement est enregistré dans la chronique. Le pas de temps  $dt$  renvoyé en sortie est donc adapté en fonction de la prochaine valeur de précipitation enregistrée ;
- Stockage du temps  $t$  dans le vecteur *alltime\_vect*, et incrémentation des vecteurs contenant les séries temporelles de paramètres relatifs à la végétation et de forçages climatiques avec les valeurs déterminées juste avant ;
- Passage au temps de simulation suivant.

⚠ Concernant le format des données de pluie, on suppose qu'elles ont subi un pré-traitement de sorte que PESHMELBA dispose de chroniques de pluie à un pas de temps régulier et assez fin (qu'il faut peut-être redéfinir en fonction de la durée de l'événement à modéliser et du bassin versant).

### 6.3 Analyse du paysage

Le module *init\_CONF* contient également une partie d'analyse du paysage. Pour chaque élément composant le maillage de PESHMELBA, on cherche à connaître quels sont les éléments connectés en aval (et éventuellement en amont), la longueur totale d'interface avec les éléments aval  $L_{int}$ , les coefficients de répartition du ruissellement *weight\_runoff* et une longueur moyenne de ruissellement *Loff* (la signification de ces 2 derniers paramètres est détaillée dans le chapitre 11). Ces informations étant réutilisées à chaque pas de temps dans plusieurs modules (voir chapitres 10.2 et 11), il est plus judicieux de les déterminer une fois pour toute en début de simulation. Cette partie d'analyse du paysage est faite dans le bloc "landscape\_analyse" qui fait appel aux fonctions *calculate\_fromto\_elements\_surface()*, *calculate\_distribution\_downstream()* et *calculate\_runoff\_length()*. Cette étape permet de déterminer pour chaque élément  $i$  les tables suivantes :

- *weight\_runoff* contenant pour chaque exutoire identifié de l'élément  $i$ , la proportion du volume de ruissellement sortant qui lui sera attribué. Pour un objet  $i$  connectée à  $n$  éléments, la table *weight\_runoff* se présente comme suit :

element connecté à $i$	proportion de ruissellement sortant de $i$ qu'il recevra
1	0.1
2	0.4
...	...
n	0.1

- *connect\_to\_HU* contenant les indices des UH situés à l'aval de l'élément  $i$ .

- *connect\_to\_reach\_HU* contenant les indices des fossés situés à l’aval de l’élément *i* (première colonne) puis l’indice de l’UH potentiellement situé à l’aval du fossé de la première colonne (seconde colonne). En cas d’absence de parcelle à l’aval du fossé, l’indice 9999 est stocké.
- *connect\_to\_bcs* contenant les indices des éléments de type BCS (BoundaryConditionStorage) situés à l’aval de l’élément *i*.
- *connect\_from\_bcs* contenant les indices des éléments de type BCS (BoundaryConditionStorage) situés à l’amont de l’élément *i*.
- *connect\_to\_tanklat* contenant les indices des haies/talus situés à l’aval de l’élément *i*.

Pour chaque linéaire, on identifie les éléments du même type, connectés en longitudinal, à l’aval de cet élément. Ceci permettra d’identifier le réseau orienté duquel il fait partie. Pour cela, on détermine la table *table\_connect\_ditch\_ditch* (respectivement *connect\_river\_river* ou *connect\_hedge\_hedge*) pour chaque fossé (resp. rivière ou haie/talus). La table regroupe pour un tronçons de fossé (resp. rivière ou haie/talus) donné tous les tronçons de fossés (resp. rivière ou haie/talus) connectés à son aval.

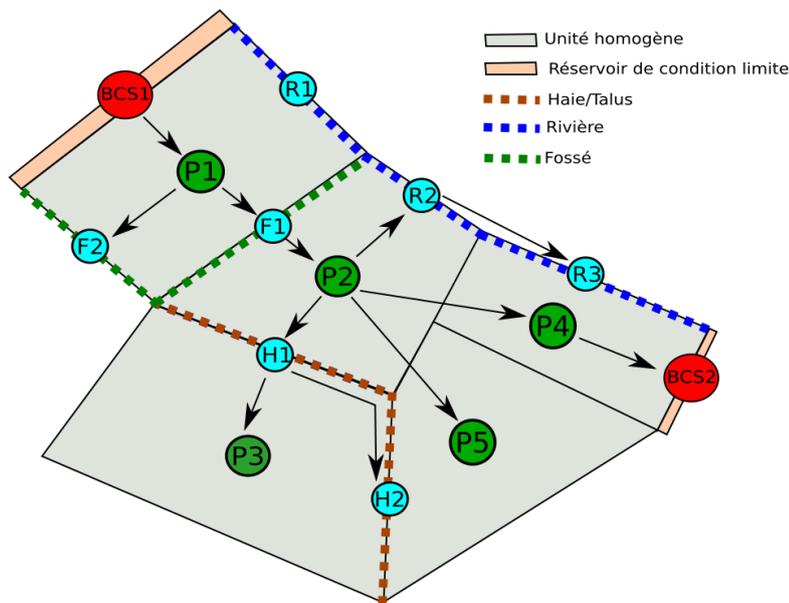


FIGURE 20 – Exemple de configuration entre surfaciques et linéaires. BCS désigne les éléments BoundaryConditionStorage, F désigne les fossés, P désigne les parcelles, R désigne les rivières et H désigne les haies.

Dans le cas de la fig. 20, voici quelques exemples de tables obtenues après l’étape d’analyse du paysage :

- Pour la parcelle **P1**, la table *connect\_to\_reach\_HU* contient : (F1,P2); (F2,9999)
- Pour la parcelle **P1**, la table *connect\_from\_bcs* contient : BCS1
- Pour la parcelle **P2**, la table *connect\_to\_HU* contient : P4; P5
- Pour la parcelle **P2**, la table *connect\_to\_tanklat* contient : H1

— Pour la parcelle **P4**, la table *connect\_to\_bcs* contient : BCS2

En plus de ces éléments, la table contenant la description de toutes les connexions latérales est stockée dans un objet PALM *lateral\_connection* et envoyée au coupleur. Cette table contient les indices des éléments amont et aval de chaque connexion, les coordonnées, l'altitude et la longueur de l'interface concernée (exemple tab. 6.1). De plus, pour les connexions entre parcelle amont-fossé puis fossé-parcelle aval, on considère plutôt ces 2 connexions comme une connexion parcelle amont-parcelle aval qui serait en partie interceptée par le tronçon de fossé (voir chapitre 10.3). Pour faciliter le traitement de ces interfaces, la table *lateral\_connection\_complete* est également élaborée et envoyée vers le coupleur. Elle contient exactement les mêmes informations que la table *lateral\_connection* mais regroupe cette fois-ci les connexions parcelle-parcelle interceptées par un fossé sur une même ligne. Ainsi, à la table *lateral\_connection* suivante :

Element amont	Element aval	Length	Lon	Lat	Alt
Plot1	Ditch1	1800	2800	5420	1000
Ditch1	Plot2	1800	2800	5420	1000
Plot2	River3	1400	2900	5462	900

TABLE 6.1 – Exemple de table *lateral\_connection* transmise à PALM

correspond la table *lateral\_connection\_complete* suivante (on note que pour les connexions autre que parcelle-fossé-parcelle, l'absence d'élément interceptant est signalé par "9999" dans la troisième colonne :

Element amont	Element aval	Element interceptant	Longueur d'interface	Lon	Lat	Alt
Plot1	Plot2	Ditch1	1800	2800	5420	1000
Plot2	River3	9999	1400	2900	5462	900

TABLE 6.2 – Exemple de table *lateral\_connection\_complete* transmise à PALM



# Chapitre 7

## Transferts d'eau (1D) dans le sol

### 7.1 Introduction

Le premier processus que PESHMELBA est capable de représenter est le transfert d'eau dans le sol. Comme les éléments sont au mieux représentés en subsurface par une colonne de sol avec un maillage seulement vertical, on parle ici de transferts unidimensionnels dans la direction verticale. Ces derniers peuvent être modélisés dans tous les éléments surfaciques : parcelles et bandes enherbées ainsi que dans le compartiment de subsurface des haies, lui aussi composé de 2 colonnes de sol.

Pour modéliser ces transferts, on se place dans le formalisme le plus communément utilisé pour décrire les transferts en milieux poreux ; celui de l'équation de Richards. Selon ce formalisme, les écoulements dans la zone saturée sont régis par l'équation de Darcy qui suppose qu'il existe une relation linéaire entre la vitesse d'écoulement et le gradient hydraulique. Le coefficient de proportionnalité est appelé conductivité hydraulique et décrit la capacité du milieu à transmettre de l'eau :

$$q = -K \frac{dh}{dz} + K \quad (7.1)$$

avec  $q$  [L.T<sup>-1</sup>], le flux d'eau,  $K$  [L.T<sup>-1</sup>] la conductivité hydraulique,  $h$  [L], la pression capillaire et  $z$  la profondeur [L].

En combinant l'équation de Darcy à une équation de conservation de la masse, on obtient la loi de Richards (1931) ou loi de Darcy généralisée qui décrit les écoulements en milieu poreux non saturé :

$$\frac{\partial \theta}{\partial t} = \frac{K \frac{\partial h}{\partial z}}{\partial z} - \frac{\partial K}{\partial z} + Q(z, t) \quad (7.2)$$

avec  $\theta$  la teneur en eau volumique [-] et  $Q$  terme puits/source [T<sup>-1</sup>] (terme représentant infiltration d'eau de pluie, l'évapotranspiration, la présence d'un drain par exemple. Il sera considéré nul dans la suite).

## 7.2 Méthode de résolution de Ross

Le module FRER1D (**F**ast **1D** **R**ichards **E**quation **R**esolution) (Ross, 2003) permet la représentation des transferts d'eau dans des sols hétérogènes en se basant sur la résolution de l'équation de Richards 1D. Le module initial permet de reproduire l'infiltration verticale de l'eau dans le sol, l'accumulation d'eau à la surface du sol (ponding) et l'évaporation en réponse à des forçages climatiques (pluie et évapotranspiration potentielle). Dans le cadre de PESHMELBA, il est nécessaire de pouvoir également représenter les flux latéraux.

L'ensemble du schéma de résolution est largement décrit dans l'article de Ross (2003) et dans les travaux de Master de Martinez (2007) et la description qui suit s'appuie en grande partie sur ces travaux.

### 7.2.1 Propriétés hydrauliques du sol utilisées

Dans le schéma de résolution initial, les relations de Brooks and Corey sont utilisées pour décrire les propriétés hydrauliques du sol (conductivité hydraulique et courbe de rétention) :

$$S = \begin{cases} (\frac{h}{h_e})^{-\lambda}, & \text{si } h < h_e \\ 1, & \text{si } h \geq h_e \end{cases} \quad (7.3)$$

avec  $h_e$  la pression d'entrée de l'air [L] ( $h_e < 0$ ),  $\lambda$  et  $\mu$  des paramètres de forme strictement positifs.

$$\frac{K}{K_s} = \begin{cases} (\frac{\theta}{\theta_s})^{-\eta}, & \text{si } h < h_e \\ 1, & \text{si } h \geq h_e \end{cases} \quad (7.4)$$

D'autre part, dans une extension de son travail publiée en 2006 (Ross, 2006), Ross propose une extension de sa solution à des propriétés de sol plus générales (utilisation des relations de Van Genuchten) :

$$S = \begin{cases} \frac{1}{(1+(\frac{h}{h_g})^n)^m}, & \text{si } h < h_e \\ 1, & \text{si } h \geq h_e \end{cases} \quad (7.5)$$

avec  $h_g$ ,  $m$ ,  $n$  des paramètres de formes de la courbe de rétention d'eau.

$$\frac{K}{K_s} = \begin{cases} \left( \frac{(1-(h/h_g)^m)^{mn} (1+(h/h_g)^n)^{-m}}{(1+(\frac{h}{h_g})^n)^{m/2}} \right)^2, & \text{si } h < h_e \\ 1, & \text{si } h \geq h_e \end{cases} \quad (7.6)$$

### 7.2.2 Potentiel de Kirchhoff

La méthode de résolution proposée par Ross se base sur l'utilisation du potentiel de Kirchhoff  $\Phi$  [ $L^2T^{-1}$ ] :

$$\Phi(h) = \int_{-\infty}^h K(\tilde{h}) d\tilde{h} \quad (7.7)$$

avec  $h$  la pression capillaire [L] et  $K$  la conductivité hydraulique [LT<sup>-1</sup>].

La relation de Brooks and Corey 7.4 donne :

$$\Phi(h) = \frac{-h_e K}{\lambda\eta - 1} \left( \frac{h}{h_e} \right)^{1-\lambda\eta} \quad (7.8)$$

et la relation de Van Genuchten aboutit à :

$$\Phi(h) = \frac{-h_g K}{n} B_x(m\eta - 1/n, 1/n) \quad (7.9)$$

avec  $B_x$  la fonction beta incomplète définie comme :

$$B_x(a, b) = \int_0^x t^{a-1} (1-t)^{b-1} dt \quad (7.10)$$

et  $x = \frac{1}{1+h-r^n}$ .

Dans ce cas-là, on utilisera une approximation de la fonction beta et donc du potentiel de Kirchhoff.

### 7.2.3 Discrétisation

L'équation de Richards est une EDP hautement non linéaire qui doit être résolue de manière numérique.

Dans la méthode de résolution de Ross, pour résoudre numériquement cette équation, le sol est divisé en  $n$  couches horizontales d'épaisseurs  $\Delta x_i$ . Le centre de chaque couche est un noeud de calcul et la distance entre deux noeuds de calcul est noté  $\Delta z_i$ .  $q_i$  est le flux à l'interface entre les cellules  $i$  et  $i+1$ . L'équation du bilan de masse pour la couche  $i$  donne à chaque instant :

$$\frac{dV_i}{dt} = q_{i-1} - q_i \quad (7.11)$$

où  $V_i$  est le volume d'eau dans la cellule  $i$ , lequel est égal à  $V_i = \Delta x_i \theta_{si} S_i$  avec  $\theta_{si}$  la teneur en eau à saturation de la cellule  $i$  et  $S_i$  le degré de saturation de la couche  $i$ . En discrétisant l'équation du bilan de masse entre  $t$  et  $t + \Delta t$  et en évaluant les flux au temps  $\sigma\Delta t$ , avec  $\sigma \in [0,1]$ , on a :

$$\frac{\Delta V_i}{\Delta t} = q_{i-1}^\sigma - q_i^\sigma \quad (7.12)$$

En effectuant un développement limité au premier ordre de  $q_i$  en supposant qu'il dépend uniquement des degrés de saturation  $S_i$  et  $S_{i+1}$ , on peut écrire :

$$q_i^\sigma = q_i^0 + \sigma \left( \left. \frac{\partial q_i}{\partial S_i} \right|_0 \Delta S_i + \left. \frac{\partial q_i}{\partial S_{i+1}} \right|_0 \Delta S_{i+1} \right) \quad (7.13)$$

En substituant 7.13 dans 7.12, on obtient un système de  $n$  équations du type :

$$a_i \Delta S_{i-1} + b_i \Delta S_i + c_i \Delta S_{i+1} = d_i, i = 1 \dots n \quad (7.14)$$

avec les coefficients définis comme :

$$\begin{cases} a_i = \left. \frac{\partial q_{i-1}}{\partial S_{i-1}} \right|^0 \\ b_i = \left. \frac{\partial q_{i-1}}{\partial S_i} \right|^0 - \left. \frac{\partial q_i}{\partial S_i} \right|^0 - \frac{\partial x_i(\theta_{si} - \theta_{ri})}{\sigma \Delta t} \\ c_i = - \left. \frac{\partial q_i}{\partial S_{i+1}} \right|^0 \\ d_i = - \left. \frac{(q_{i-1}^0 - q_i^0)}{\sigma \Delta t} \right|^0 \end{cases} \quad (7.15)$$

avec  $a_1 = c_n = 0$  car  $S_0$  et  $S_{n+1}$  n'existent pas.

Dans le cas où les cellules sont saturées, on ne peut plus utiliser les  $\Delta S_i$  comme variables de calcul car les  $S_i$  sont égaux à 1 et donc  $\Delta S_i = 0$ . On utilise donc à la place les variations du potentiel de Kirchoff  $\Delta \Phi_i$ . Dans le cas saturé, la variation du volume d'eau de la cellule  $\Delta V_i$  est nulle. En conduisant un développement limité identique à 7.13 mais e,  $\Delta \Phi_i$  et en remplaçant les termes dans le bilan de masse, on obtient pour tout  $i$  :

$$\frac{\partial q_{i-1}}{\partial \Phi_{i-1}} \Delta \Phi_{i-1} + \left( \frac{\partial q_{i-1}}{\partial \Phi_i} - \frac{\partial q_i}{\partial \Phi_i} \right) \Delta \Phi_i - \frac{\partial q_i}{\partial \Phi_{i+1}} \Delta \Phi_{i+1} = \frac{q_{i-1}^0 - q_i^0}{\sigma} \quad (7.16)$$

Il s'agit de la même équation que 7.14 avec des définitions différentes de  $a_i$  à  $d_i$ . Pour une colonne de sol saturée/non saturée, on obtient donc un système d'équations linéaires mixte en  $\Delta S_i$  et  $\Delta \Phi_i$ .

## 7.2.4 Calcul des flux

Pour déterminer les flux  $q_i$  et leur dérivés intervenant dans l'expression des coefficients  $a_i, b_i, c_i$  et  $d_i$ , on fait appel à la définition du potentiel de Kirchhoff ( $\frac{d\Phi}{dh} = K(h)$ ) et à la loi de Darcy :

$$q = K(h) \left( \frac{\partial h}{\partial z} - 1 \right) \quad (7.17)$$

pour linéariser le flux sous la forme suivante :

$$q = - \frac{\partial \Phi}{\partial z} + K \quad (7.18)$$

Ainsi, le flux  $q_i$  entre les cellules  $i$  et  $i + 1$  peut-être estimé par :

$$q_i = \frac{\Phi_i - \Phi_{i+1}}{\Delta z_i} + K_{i+1/2} \quad (7.19)$$

$K_{i+1/2}$  est la conductivité calculée à l'interface entre les cellules  $i$  et  $i + 1$ . Son calcul est délicat et une formulation utilisant la moyenne pondérée par l'épaisseur des cellules n'est pas satisfaisante. On écrit plutôt  $K_{i+1/2}$  sous la forme :

$$K_{i+1/2} = \omega K_i + (1 - \omega) K_{i+1} \quad (7.20)$$

où  $\omega$  est un facteur de pondération, calculé pour chaque interface et à chaque pas de temps ajusté de telle sorte que le flux serait nul dans le cas d'un profil hydrostatique.

On obtient donc les expressions suivantes pour les dérivés du flux  $q_i$  :

— Cellule non saturée :

$$\begin{aligned}\frac{\partial q_i}{\partial S_i} &= \frac{d\Phi_i}{\Delta z_i dS_i} + \omega \frac{dK_i}{dS_i} \\ \frac{\partial q_i}{\partial S_{i+1}} &= -\frac{d\Phi_{i+1}}{\Delta z_i dS_{i+1}} + (1 - \omega) \frac{dK_{i+1}}{dS_{i+1}}\end{aligned}\quad (7.21)$$

— Cellule saturée :

$$\begin{aligned}\frac{\partial q_i}{\partial \Phi_i} &= \frac{1}{\Delta z_i} \\ \frac{\partial q_i}{\partial \Phi_{i+1}} &= -\frac{1}{\Delta z_i}\end{aligned}\quad (7.22)$$

## 7.2.5 Transport des solutés

Le module FRER1D prévoit également la résolution du transfert vertical par advection-dispersion des solutés. La méthode de résolution de l'équation d'advection-dispersion est détaillée dans l'article de 2003, on en donne ici seulement les principaux éléments.

Pour déterminer les flux de solutés, le même principe que pour l'eau est appliqué mais cette fois-ci, ces flux dépendent non seulement des variations de teneur en eau  $\theta$  mais aussi des variations de concentration en solutés  $s$  [M.L<sup>-3</sup>]. Le développement limité du flux  $q_i$  donne donc :

$$q_i^\sigma = q_i^0 + \sigma \left( \left. \frac{\partial q_i}{\partial s_i} \right|^0 \Delta s_i + \left. \frac{\partial q_i}{\partial s_{i+1}} \right|^0 \Delta s_{i+1} + \left. \frac{\partial q_i}{\partial \theta_i} \right|^0 \Delta \theta_i + \left. \frac{\partial q_i}{\partial \theta_{i+1}} \right|^0 \Delta \theta_{i+1} \right) \quad (7.23)$$

Les variations  $\Delta \theta$  sont connues à partir de la résolution de l'équation de Richards ce qui permet d'arriver à un système tridiagonal d'équations linéaires comme dans l'équation 7.14 mais pour lequel le coefficient  $d_i$  inclut les termes en  $\Delta \theta$ .

Les flux sont donnés par la formule :

$$q_i = 0.5(\epsilon_i + \epsilon_{i+1}) | q_i | \frac{c_{wi} + c_{wi+1}}{\Delta z_i} + 0.5q_{wi}(c_{wi} + c_{wi+1}) \quad (7.24)$$

où  $\epsilon$  est la dispersivité (cm),  $q_w$  le flux d'eau moyen et  $c_w$  la concentration de solutés dans l'eau du sol (g/cm<sup>3</sup>). La contribution de la diffusion à la dispersion est négligée.

Dans la version originale de FRER1D, les solutés étaient résolus régulièrement (toutes les  $n$  itérations pour l'eau, la valeur de  $n$  étant fixée par l'utilisateur). Pour alléger la résolution et se débarrasser de problèmes de calculs (la même boucle était appelée plusieurs fois lorsqu'on laissait PALM gérer le pas de temps), la partie de résolution de l'ADE (**A**dvection-**D**ispersion **E**quation) a été déplacée à la fin du script, hors de la boucle temporelle. Ainsi, l'équation ne sera résolue qu'une seule fois, à chaque fin de simulation FRER1D.

Le script initial prévoyait la mise à jour du vecteur  $s$  (g de solutés/volume de sol) au cours de la résolution de l'ADE et la concentration dissoute  $c_w$  était ensuite immédiatement mise à jour avec le calcul de l'équilibre d'adsorption. Dans le cadre de

PESH-MELBA, on passera d'abord par une étape de calcul de la dégradation comme détaillé dans la partie suivante. Pour cela, il est nécessaire de rajouter une étape de mise à jour de  $c_w$  avant le calcul de la dégradation.

Pour chaque cellule numérique  $j$ , la concentration finale  $c_{wj}$  s'exprime comme suit :

$$c_{wj,final} = \frac{m_{j,final}}{w_{j,final}} = \frac{cellvol * c_{wj,ini} * \theta_{j,ini} + cellvol * \Delta m}{cellvol * \theta_{j,final}} \quad (7.25)$$

Avec :

- $m_{jf}$  : masse de soluté dissous contenu dans la cellule  $j$  à la fin du pas de temps (g)
- $w_{jf}$  : volume d'eau contenu dans la cellule  $j$  à la fin du pas de temps ( $\text{cm}^3$ )
- $m_{ij}$  : masse de soluté dissous contenu dans la cellule  $j$  au début du pas de temps (g)
- $\Delta m$  : Variation de masse dans dans la cellule  $j$  au cours du pas de temps (g)
- $\theta_f$  : teneur en eau dans la cellule  $j$  à la fin du pas de temps ( $\text{cm}^3 \cdot \text{cm}^{-3}$ )
- $cellvol$  : volume de la cellule  $j$  ( $\text{cm}^3$ )

### 7.2.6 Transfert réactif de solutés

En plus du transport des solutés, on représente les processus de dégradation et d'adsorption/désorption. L'adsorption était déjà implémentée dans la version de FRER1D fournie par Ross alors que la dégradation a du être rajoutée.

Dans un premier temps, on fait l'hypothèse que seuls les solutés **dissous** sont dégradés. Après la résolution de l'ADE, la quantité dégradée est calculée puis retranchée au stock de solutés disponibles :

$$sm_{deg} = c_w \theta (1 - e^{-(t_f - t_i)k}) \quad (7.26)$$

Avec :

- $sm_{deg}$  : masse de solutés dégradés entre  $t_i$  et  $t_f$  [M]
- $c_w$  : concentration de solutés dissous [ $\text{M} \cdot \text{L}^{-3}$ ]
- $\theta$  : teneur en eau [ $\text{L}^3 \cdot \text{L}^{-3}$ ]
- $k$  constante de vitesse [ $\text{T}^{-1}$ ]

Si besoin, la masse de soluté dégradée ne disparaît pas totalement et une partie est incrémentée au stock de ses différents métabolites. Pour cela, on donne pour chaque soluté, l'indice de son (ou ses) métabolites et un coefficient de transformation. La somme des coefficients de transformation des différents métabolites associés à un soluté donné ne doit pas dépasser 1.

Puis, l'équilibre d'adsorption est calculé grâce à l'utilisation d'isothermes d'adsorption comme détaillé dans la partie [ref](#). On peut utiliser au choix un isotherme d'adsorption linéaire permettant une résolution directe ou un isotherme d'adsorption de Freundlich pour lequel une méthode itérative (méthode de Newton) est nécessaire pour la résolution.

### 7.2.7 Prise en compte du flux latéral dans le module FRER1D

Pour pouvoir coupler plusieurs colonnes de sol comme prévu dans le modèle PESHMELBA, il est nécessaire de pouvoir représenter les flux latéraux. Comme on ne prétend pas résoudre l'équation de Richards en 3D mais plutôt inclure une simple représentation de ces flux comme forçage latéral (uniquement les flux en zone saturée, calculés via l'équation de Darcy), ceux-ci interviendront comme terme source/puit dans la résolution de l'équation de Richards. Pour cela, on se base en partie sur des travaux déjà réalisés par Braud (2006) et Varado et al. (2006b), où un tel ajout avait déjà été implémenté (extraction racinaire dans le second cas).

Le flux net  $Q$  [ $L^3T^{-1}$ ] à utiliser comme forçage sera calculé par le module PPI et fourni en entrée de FRER1D. Le module PPI permet de gérer la répartition du flux entre les différentes couches de calcul et fournira donc un vecteur  $\tilde{Q}$  indiquant le forçage latéral correspondant à chaque couche numérique.

#### Prise en compte des surfaces et des volumes

La première adaptation du code concerne la prise en compte des surfaces et des volumes. Alors que la version initiale de FRER1D considérait une colonne à la surface unitaire, le code doit être adapté pour PESHMELBA de manière à prendre en compte la surface réelle des UH mais aussi pour être homogène avec le forçage latéral fourni en [ $L^3T^{-1}$ ].

Le module doit être capable de récupérer des informations sur la géométrie de la colonne afin de calculer les surfaces d'échange et le volume de chaque cellule. Les flux  $q_i$  doivent donc être exprimés en [ $L^3T^{-1}$ ] et non plus en [ $LT^{-1}$ ] en multipliant par l'aire réelle les coefficients  $a_i$ ,  $b_i$ ,  $c_i$ ,  $d_i$  :

$$a_i = \frac{\partial q_{i-1/2}}{\partial S_{i-1}} Area \quad (7.27)$$

$$b_i = \frac{\partial q_{i-1/2}}{\partial S_{i-1}} Area - \frac{\partial q_{i+1/2}}{\partial S_i} Area - \frac{|C_i| \theta_{S_i}}{\sigma \partial t} Cellvol_i \quad (7.28)$$

$$c_i = \frac{\partial q_{i+1/2}}{\partial S_{i+1}} Area \quad (7.29)$$

$$d_i = \frac{q_{i-1/2} Area - q_{i+1/2} Area}{\sigma} \quad (7.30)$$

avec  $Area$ , la surface d'échange entre les cellules  $C_{i-1}$  et  $C_i$  [ $L^2$ ],  $q_{i+1/2}$  le flux entre la cellule  $C_i$  et la cellule  $C_{i+1}$  [ $LT^{-1}$ ],  $Cellvol$  le volume de la cellule  $i$  [ $L^3$ ].

Les flux de forçage (précipitation, évaporation, etc....) doivent eux aussi être exprimés en [ $L^3T^{-1}$ ].

#### Implémentation du flux latéral pour l'eau et les solutés

Dans le schéma de résolution de Ross, la prise en compte du forçage latéral intervient seulement dans le calcul du coefficient  $d_i$  de la matrice tridiagonale :

$$d_i = \left( \frac{q_{i-1/2}Area - q_{i+1/2}Area}{\sigma} + \frac{Lat_i}{\sigma} \right) \quad (7.31)$$

avec  $Lat_i$ , le flux latéral à inclure [ $L^3T^{-1}$ ] (voir chapitre 10.2).

De la même manière, pour les solutés, le coefficient  $d_i$  est modifié dans le système tridiagonal d'équations relatives à la résolution de l'ADE. On y ajoute un terme  $\frac{SLAT_i}{\sigma}$  pour le flux latéral de solutés [ $M.T^{-1}$ ].

### 7.2.8 Calcul de l'équilibre d'adsorption en surface

Le module FRER1D prend initialement en compte 2 types de conditions limites en surface :

- Une condition de charge correspondant à une hauteur de ponding. Cette hauteur d'eau  $h_0$  est associée au ruissellement provenant des parcelles amont (voir chapitre 11) ou à un refus d'infiltration sur la parcelle au pas de temps précédent. Des solutés peuvent être associés à ce volume d'eau à la concentration  $c_0$ .
- Une condition de flux  $qprec$  correspondant aux précipitations ou à un apport ponctuel en eau comme cela peut être le cas lors de l'application de pesticides sur la parcelle. A ce flux est associé une concentration en solutés  $c_{in}$ .

Dans ces deux cas, il faut rendre compte du fait qu'une partie des solutés disponibles en surface peut être adsorbée et ainsi n'être plus disponible pour s'infiltrer dans la colonne de sol. FRER1D a été modifié pour calculer l'équilibre d'adsorption en surface avant de déclencher une éventuelle infiltration des solutés dissous restants. Pour cela le schéma initial a été modifié pour pouvoir gérer correctement différents apports en surface (charge et flux). La condition initiale limite en charge a été transformée en condition en flux et intégrée à l'apport des précipitations et des applications  $qprec$ . Attention, cela n'empêche pas que du ponding peut apparaître au cours du pas de temps et la condition limite correspondante sera bien une CL de charge. C'est uniquement en début de pas de temps qu'une éventuelle condition limite-initiale en charge sera transformée en flux comme illustré fig. 21. Ainsi, la parcelle ne reçoit qu'un flux surfacique  $qprec$ , somme de la pluie, du ruissellement et des applications de pesticides (Branger et al., 2008).

On raisonne à partir de ce flux unique  $qprec$  pour calculer l'équilibre d'adsorption. Pour cela, on considère que l'adsorption des solutés se fait sur une bande de sol définie par sa hauteur ( $h_{ads}$ ), sa masse volumique ( $\rho_b$ ) et sa teneur en eau à saturation ( $\theta_{sat}$ ). On considère un mélange complet avec le volume d'eau disponible sur la hauteur  $h_{ads}$  c'est-à-dire que sur l'épaisseur de sol  $h_{ads}$  le sol/substrat peut être considéré comme en suspension dans l'eau de la parcelle, autorisant une adsorption complète et instantanée. Les différentes étapes du calcul sont définies comme suit :

1. Détermination du volume équivalent au flux  $qprec$  pendant le pas de temps en cours.
2. Détermination de la concentration initiale de solutés dans ce volume : on considère initialement que tout le stock de solutés en surface est dissous (vrai pour les solutés déjà présents dans le ponding et ceux appliqués pendant le pas de temps).
3. Détermination du volume d'eau en contact avec la bande de terre sur laquelle se fait l'adsorption et de la masse totale de solutés contenue dans ce volume.

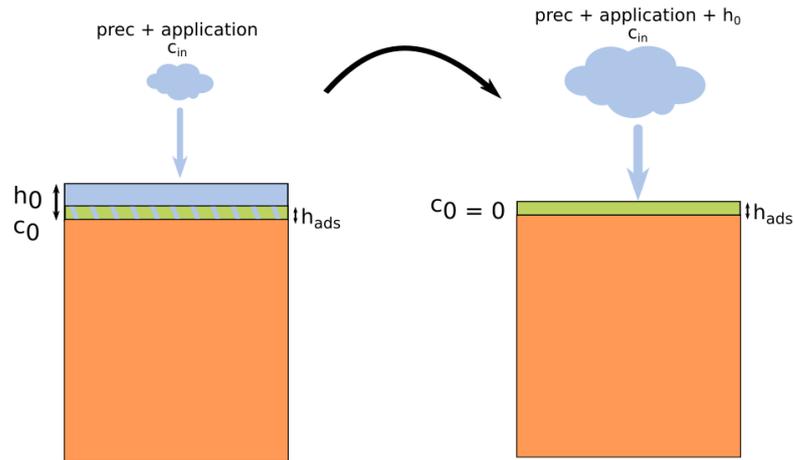


FIGURE 21 – Changement de CL en surface pour FRER1D (début de simulation).

4. Détermination des quantités adsorbées et dissoutes sur ce stock là par un isotherme linéaire ou de Freundlich.
5. Mise à jour de la quantité de solutés adsorbés en surface et de réévaluation de la concentration  $c_{in}$  sur tout le volume d'eau  $q_{prec} \cdot dt$  en intégrant les solutés dissous.

On insiste sur le fait que la couche d'adsorption est virtuelle puisqu'elle est rattachée à la surface qui ne possède pas d'épaisseur propre. Les solutés adsorbés et dissous dans l'épaisseur  $h_{ads}$  ne sont jamais intégrés dans le stock de solutés des premiers cm de la colonne de sol. Ils restent associés au stock de surface.

### 7.2.9 Couche de mélange

La version de FRER1D de Ross (2006) permet de simuler l'infiltration de solutés mais interdit leur exfiltration (ceci reste néanmoins possible pour l'eau). Rajouter cette composante directement dans FRER1D s'est avéré difficile mais une solution a été trouvée en considérant une couche de mélange entre les solutés du sol et les solutés de la lame ruisselante. L'usage d'une telle couche de mélange permet entre autre de mieux modéliser la remobilisation des solutés par le ruissellement. Cette méthode est assez communément utilisée comme c'est le cas dans Gatel et al. (2018). Dans PESHMELBA, le mélange est uniforme et complet avec la première cellule numérique de la colonne de sol et a lieu en fin de pas de temps de FRER1D s'il reste une hauteur de ponding non nulle. Il n'y a pas de correspondance entre la couche d'adsorption (section 7.2.8) et la couche de mélange. Conceptuellement, la couche d'adsorption et la couche de mélange se chevauchent toutefois. Les 2 concepts sont illustrés fig. 22.

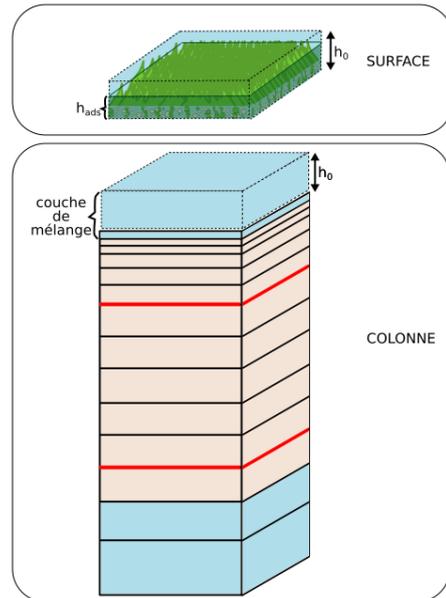


FIGURE 22 – Illustration du concept de la couche d'adsorption  $h_{ads}$  pour l'objet surface et de la couche de mélange pour l'objet colonne.

### 7.3 Extension de la méthode de résolution de Ross à des propriétés hydrauliques du sol plus générales

La méthode de Ross (2003) se base exclusivement sur les relations de Brooks & Corey pour décrire les propriétés hydrodynamiques du sol. Cette formulation permet une expression analytique du potentiel de Kirchhoff mais limite également les applications de la méthode. En effet, de nombreuses fonctions de pédotransfert ont été établies pour d'autres formulations que celle de Brooks and Corey, comme celle de Van Genuchten par exemple (?). Dans sa version du code datant de 2006, Ross propose donc une extension de la méthode de résolution aux relations de Van Genuchten pour décrire les propriétés hydrodynamiques du sol. L'utilisation de ces fonctionnelles pose cependant plusieurs problèmes. Tout d'abord, la courbe de conductivité de Van Genuchten décroît trop brutalement lorsque  $h$  approche de zéro et ne décrit pas correctement le comportement du sol près de la saturation (Ross, 2006). D'autre part, il n'existe pas d'expression analytique simple du potentiel de Kirchhoff basée sur la relation de Van Genuchten. Enfin, la pente importante de la courbe de rétention près de la saturation peut conduire à des problèmes numériques. Ross propose donc une modification de l'expression de la courbe de rétention et de la courbe de conductivité pour pallier à ces difficultés.

#### 7.3.1 Modification de la courbe de rétention

L'une des approches classiques pour contourner les difficultés numériques causées par la forme de la courbe de rétention près de la saturation consiste à déclarer que la saturation n'a pas lieu pour une pression nulle mais pour une pression légèrement inférieure (-1 ou -2 cm) (Vogel and Císlerová, 1998; Vogel et al., 2001; Schaap and

van Genuchten, 2006). Ici, une autre approche est utilisée permettant de maintenir un passage à la saturation pour  $h=0$  cm mais aussi d'intégrer par la suite une composante de "macropores" à la courbe de conductivité. Pour cela, au delà d'une saturation  $S_1$  fixée très proche de 1 (e.g. 0.99),  $h$  est définie comme une fonction quadratique de  $S$  avec la même valeur et la même pente que la fonction classique de VG en  $S_1$  et une valeur de 0 en  $S=1$ . On a donc :

$$h(S) = \begin{cases} (S^{-1/m} - 1)^{1/n}, & \text{si } S \leq S_1 \\ a_1 \left(\frac{S-1}{S_1-1}\right) + a_2 \left(\frac{S-1}{S_1-1}\right)^2, & \text{si } S > S_1 \end{cases} \quad (7.32)$$

avec

$$a_1 = 2h_1 - \left(\frac{dh}{dS}\right)_1 (S_1 - 1) \quad (7.33)$$

$$a_2 = h_1 - a_1 \quad (7.34)$$

où  $h_1$  et  $\left(\frac{dh}{dS}\right)_1$  sont la valeur et la dérivée de la courbe de pression de VG en  $S_1$ . On obtient ensuite par inverse l'expression suivante pour la courbe de rétention :

$$S(h) = \begin{cases} (1 + h^n)^{-m}, & \text{si } h \geq h_1 \\ 1 + \frac{2(S_1-1)h}{a_1 + \sqrt{a_1^2 + 4a_2h}}, & \text{si } h < h_1 \end{cases} \quad (7.35)$$

### 7.3.2 Modification de la courbe de conductivité

La courbe de conductivité utilisée est basée sur la formulation de Schaap-Van Genuchten (Schaap and van Genuchten, 2006). Celle-ci correspond à une modification de la courbe de conductivité Mualem-Van Genuchten modifiée près de la saturation. Cette dernière permettait d'éliminer les pentes trop fortes qui apparaissent juste avant la saturation (notamment pour des valeurs de  $n$  approchant la borne inférieure de 1 (Vogel et al., 2001)). La modification proposée par Schaap and van Genuchten (2006) permet en plus d'intégrer l'influence des écoulements préférentiels qui dominant le régime d'écoulement près de la saturation. La formulation finale permet ainsi de distinguer les régimes composés uniquement d'écoulements dans la matrice des régimes où les écoulements dans les macropores et autres fractures du sol se superposent et dominant les écoulements matriciels (près de la saturation).

La formulation de Mualem-Van Genuchten modifiée proposée par Vogel et al. (2001) est la suivante :

La courbe de rétention est modifiée comme suit :

$$\theta(h) = \begin{cases} \theta_r + \frac{\theta_m - \theta_r}{(1 - |\alpha h|^n)^m}, & \text{si } h < h_s \\ \theta_s, & \text{si } h \geq h_s \end{cases} \quad (7.36)$$

où  $h_s$  est une pression capillaire seuil, négative et proche de zéro et  $\theta_m$  est un paramètre fictif ( $\theta_m > \theta_s$ ) défini comme :

$$\theta_m = \theta_r + (\theta_s - \theta_r)(1 + |\alpha h_s|^n)^m \quad (7.37)$$

La saturation s'exprime donc

$$S(h) = \begin{cases} \frac{\theta(h)-\theta_r}{\theta_m-\theta_r}, & \text{si } h < h_s \\ \frac{\theta_s-\theta_r}{\theta_m-\theta_r}, & \text{si } h \geq h_s \end{cases} \quad (7.38)$$

Et on en déduit l'expression de la courbe de conductivité de Mualem-Van Genuchten modifiée :

$$K(S) = \begin{cases} K_{xm}x^{mp} [1 - (1 - x)^m]^2, & \text{si } h < h_s \\ K_{xm}, & \text{si } h \geq h_s \end{cases} \quad (7.39)$$

avec  $x$  défini par  $S^{1/m}$ ,  $p$  un paramètre de porosité,  $K_{xm} = \frac{K_x}{x_s^{pm}(1-x_s)^m}$ ,  $x_s$  la valeur de  $x$  à la pression  $h_s$  et  $K_x$  est la conductivité à saturation de la matrice du sol [ $LT^{-1}$ ].

Cette expression de la courbe de conductivité est alors modifiée pour prendre en compte l'influence des macropores. La gamme des pression est divisée en 3 :

- **$h < h_{mac_2}$**  : le régime est dominé par les écoulements dans la matrice.  $h_{mac_2}$  est fixé à -40cm dans le code d'origine. Pour cette gamme de pression, la conductivité hydraulique s'exprime avec la relation de Mualem-Van Genuchten modifiée :

$$K_{vGm}(h) = K_{xm}x^{mp} [1 - (1 - x)^m]^2 \quad (7.40)$$

- **$h_{mac_2} < h < h_{mac_1}$**  : le régime peut être considéré comme dominé par les écoulements dans des mésopores de taille intermédiaire (Schaap and van Genuchten, 2006). Dans le code de Ross,  $h_{mac_1} = h_s = -4$  cm.

$$K_{ms}(h) = \left( \frac{K_s}{K_{vGm}(h)} \right)^{R(h)} K_{vGm}(h) \quad (7.41)$$

où la fonction  $R(h)$  est défini comme :

$$R(h) = \frac{R_{mac1}(h - h_{mac_2})}{h_{mac_1} - h_{mac_2}} \quad (7.42)$$

et  $R_{mac1}$  fixé à 0.25.

- **$h > h_{mac_1}$**  : le régime est dominé par les écoulements dans les macropores et la conductivité s'exprime par une fonction qui décroît exponentiellement avec la pression depuis la saturation :

$$K_{mp}(h) = K_s \exp\left(\frac{-h}{h_p}\right) \quad (7.43)$$

$h_p$  est déterminé en supposant qu'à  $h = h_{mac_1}$ , on a :

$$K(h) = K_{mp}(h) = K_{ms}(h) \quad (7.44)$$

On a alors  $h_p = \frac{h_{mac_1}}{(1-R_{mac1})\ln\left(\frac{K_s}{K_x}\right)}$

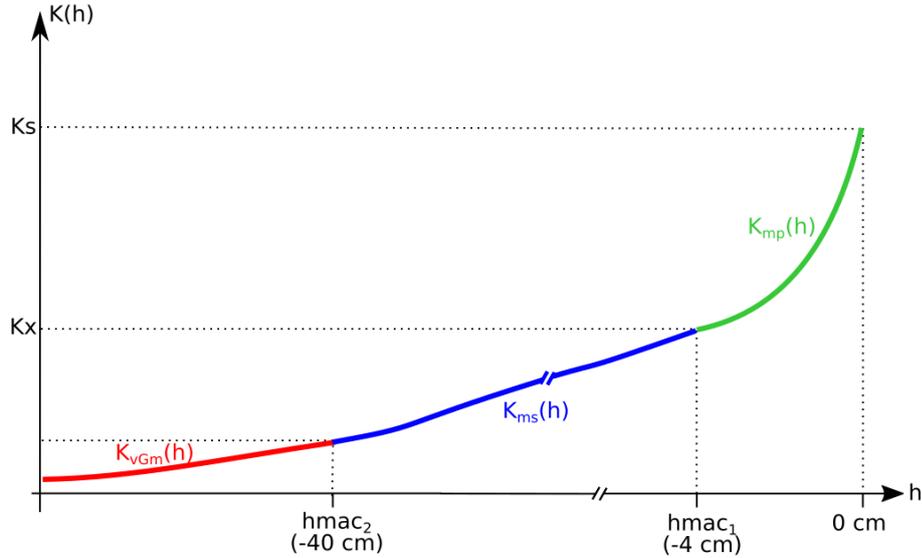


FIGURE 23 – Exemple de courbe de conductivité basée sur les paramètres de Van Genuchten utilisée dans FRER1D.

La forme finale de la courbe de conductivité est illustrée fig. 23. La principale difficulté de cette formulation est l'absence d'expression analytique simple pour le potentiel de Kirchhoff dans la gamme de pressions inférieures à  $hmac_1$ . On utilise alors une approximation de la conductivité et du potentiel basée sur une expansion en série et sur les approximatifs de Padé ainsi que des fonctions quadratiques

- Sur la gamme  $h \leq h_g$  (ou  $h \leq hmac_2$  si  $h_g > hmac_2$ ) : la conductivité  $K_{vGm}(h)$  est approximée grâce à l'approximant de  $1-(1-x)^m$  de Padé :

$$1 - (1 - x)^m \cong \frac{mx(2 - x)}{2 + (m - 2)x[1 + (m - 1)x/6]}, \text{ pour } x \leq 1/2 \quad (7.45)$$

Et l'approximant de Padé pour le potentiel de Kirchhoff donne :

$$\Phi(h) \cong h_g K_{xm} m^2 (m - 1) x^a \frac{\frac{1}{a} + \left(\frac{1}{1+a} + \frac{c}{a}\right)x + \left(\frac{c}{a+1} + \frac{b}{12}\right)}{1 + cx} \quad (7.46)$$

avec

$$a = m(p + 1) + 1; \quad b = \frac{m^2 + 11}{a + 2}; \quad c = \frac{-2(m^2 + 5)}{b(a + 3)} \quad (7.47)$$

- Pour  $h > h_g$  (ou  $h > hmac_2$  si  $h_g > hmac_2$ ) : l'intervalle de pressions est découpé en plusieurs segments sur chacun desquels  $K_{ms}(h)$  est approximée numériquement par une fonction quadratique. Ces quadratiques sont intégrées pour déterminer le potentiel de Kirchhoff correspondant.
- Pour les pressions supérieures à  $hmac_1$  : on a  $\Phi(h) = -h_p K_{mp}(h)$

## 7.4 Implémentation

### 7.4.1 module FRER1D

Le module FRER1D est un script Fortran 90 qui permet de lancer le schéma de résolution en appelant la subroutine FLOW. Il initialise les différents paramètres d'une colonne de sol et reçoit les conditions initiales en surface (volume et concentration dans le ponding). C'est dans ce module que se fait le changement de conditions limites, le calcul de l'équilibre d'adsorption puis qu'est appelé le module FLOW effectuant lui-même la résolution. L'équilibre dans la couche de mélange est enfin calculé avant de renvoyer les différentes variables au coupleur.

### 7.4.2 module FRER1D\_HEDGE

Le module FRER1D\_HEDGE fonctionne exactement de la même manière que FRER1D et permet de lancer le calcul de l'infiltration verticale sur une haie. Un module indépendant a été mis en place pour traiter ces objets car la résolution ne doit pas se faire sur une mais sur 2 colonnes de sol. Comme les scripts Fortran sont très rigoureux concernant la déclaration de variables, il aurait été assez complexe de traiter des éléments composés d'une colonne de sol et d'autres composés de 2 colonnes de sol dans un unique script FRER1D. Une solution aurait pu être d'appeler FRER1D successivement pour chaque colonne de sol composant la haie mais cette gestion aurait dû se faire à l'extérieur du script par une région de code PALM, ce qui paraissait plus complexe. Il a donc été décidé de dupliquer le code FRER1D pour en faire une version "haie" spécialement conçue pour gérer un objet avec ses colonnes de sol amont et aval. Ce module appelle donc les mêmes variables PALM que FRER1D mais est capable de distinguer les 2 composantes "up" et "down" fournies dans tous les objets PALM relatifs au fonctionnement d'une haie. Pour cela, une dimension supplémentaire est ajoutée à toutes les variables déclarées dans le script. Par exemple, la variable S (saturation) est un vecteur de dimension `n_layer` dans FRER1D et une matrice de dimensions  $(2, n\_layer)$  dans FRER1D\_HEDGE. La première ligne détermine dans quelle partie de la haie "up" ou "down" on se trouve (`S(1, :)` décrit la saturation dans la partie **amont** de la haie et `S(2, :)` dans la partie **aval**).

### 7.4.3 module FLOW et FLOW\_HEDGE

FLOW est le script dans lequel s'effectue la résolution présentée section 7.2. Le script initial a été conservé hormis la partie de calcul de la masse de solutés infiltrés. A chaque sous pas de temps de résolution, une fois déterminées la quantité d'eau infiltrée et la nouvelle hauteur de ponding, on détermine la nouvelle concentration dans le ponding  $c_0$  (s'il y a du ponding) et la masse de solutés qui ont été infiltrés pendant ce pas de temps *sinfil*. Pour cela, on distingue plusieurs cas :

**Une hauteur de ponding apparaît ou disparaît pendant le pas de temps considéré :**

S'il reste du ponding en fin de pas de temps :

— Détermination de la concentration en solutés dans le volume en eau infiltré *cav*

- $c_{av}$  est aussi la concentration en solutés de l'eau qui reste en surface  $c_0$

S'il n'y a plus de ponding à la fin du pas de temps :

- Détermination de la concentration en solutés dans le volume en eau infiltré  $c_{av}$
- $c_0 = 0$  car il n'y a plus de solutés dissous en surface

### Pas de ponding en début et en fin de pas de temps

La quantité de solutés infiltrés est exactement égale au flux en surface  $q_{prec} * c_{in} * dt$ .

Dans un second temps, le module FLOW appelle la subroutine SOLUTE qui résout en particulier la partie de transport de solutés dans la colonne de sol. Avant d'appeler cette subroutine, la variable  $m_{pond}$  est initialisée. Elle correspond à la masse de solutés dissous en surface une fois soustraite la masse adsorbée. La masse  $m_{pond}$  est initialisée comme étant la masse totale disponible pour l'infiltration :

$$m_{pond} = q_{prec} * c_{in} * area * \Delta t \quad (7.48)$$

Cette masse sera ensuite incrémentée dans le module solute : elle diminuera en cas d'infiltration et augmentera en cas d'exfiltration. S'il n'y a pas de ponding en fin de simulation,  $m_{pond}$  est forcément nulle.

FLOW\_HEDGE est strictement identique à FLOW et est utilisé. L'appel à 2 scripts séparés avait été implémenté dans un premier temps car certaines variables semblaient mal se déclarer lorsque FRER1D et FRER1D\_HEDGE appelaient simultanément FLOW. Il faudra sûrement recreuser ce point là car il n'est peut-être pas nécessaire de doubler ces scripts.

#### 7.4.4 modules HYPBC et HYPVGVG

Ces deux modules permettent la détermination des variables  $K$ ,  $S$  et  $\Phi$  en utilisant les relations de Brooks & Corey pour HYPBC et les relations de Van Genuchten modifiées pour HYPVGVG. L'appel à l'un de ces modules est déclaré au début des subroutines INIT\_FRER1D, FRER1D et FRER1D\_HEDGE et des modules FLOW et FLOW\_HEDGE. Pour le moment, PESHMELBA ne dispose pas d'un passage automatisé à l'une ou l'autre description. Lorsqu'on veut changer le type de description des propriétés du sol, il faut changer explicitement l'appel à l'un des deux modules dans la ligne "USE" des scripts fortran dans la carte d'identité du script pour INIT\_FRER1D, FRER1D et FRER1D\_HEDGE. Il faut également spécifier le type de relation utilisée dans les constantes PALM : on met à 0 la variable  $hydr\_pptes$  si on utilise les relations de Brooks & Corey, et à 2 si on utilise les relations de Van Genuchten modifiées (la variable pourra être fixée à 1 si besoin d'implémenter l'utilisation de relations composites entre Brooks & Corey et Van Genuchten, voir (Ross, 2006)).



# Chapitre 8

## Extraction racinaire

### 8.1 Introduction

PESHMELBA prétend dépasser l'approche événementielle pour modéliser les transferts hydriques advenant à l'échelle du bassin versant sur des périodes relativement longues. Dans ce cadre là, il est nécessaire de prendre en compte l'influence de la végétation (naturelle ou sous forme de culture) car sa croissance et sa transpiration permet de récupérer une partie non négligeable de l'eau infiltrée (Varado, 2004). Les phénomènes liés à la végétation qui doivent être pris en compte sont :

- l'interception de la pluie
- l'extraction racinaire
- la transpiration

De tels processus ont déjà été implémentés dans la solution numérique de Ross et sont décrits dans Varado et al. (2006b) et Varado et al. (2006a). De plus, la version de FRER1D dont nous disposons possède déjà un module d'extraction racinaire. Nous nous baserons sur l'ensemble de ces travaux pour implémenter l'interception, l'extraction racinaire et la transpiration dans PESHMELBA. La description ci-dessous s'inspire donc largement des travaux de thèse de Varado (2004) et de la description des processus qu'elle y fait.

Deux approches existent pour quantifier l'extraction racinaire. La première est une approche microscopique (Gardner, 1958) qui consiste à représenter chaque racine comme un cylindre infiniment long, de rayon uniforme et aux propriétés absorbantes. Ce modèle est écrit en coordonnées cylindriques et les flux sont représentés radialement par rapport à la racine (Li and Zhang, 2001). Cependant, il requiert de nombreuses données difficiles à acquérir, surtout à grande échelle ce qui explique qu'il soit rarement adopté dans la pratique. La seconde approche, macroscopique est plus empirique et consiste à considérer le prélèvement d'eau dans la zone racinaire dans son ensemble, sans considérer les effets des racines individuelles. L'extraction racinaire est alors évaluée en fonction du gradient de potentiel hydrique entre le sol et la plante et apparaîtra comme un terme puits dans l'équation de Richards.

Ce terme puits a été décrit par plusieurs auteurs (Feddes et al., 1978; Jarvis, 1989) et est en général représenté comme une fonction de la transpiration potentielle, de la

distribution de la densité racinaire, de la teneur en eau et parfois de la conductivité hydraulique. Le modèle d'extraction racinaire proposé dans la version du code de Ross dont nous disposons est celui de Li and Zhang (2001). Il dépend de la transpiration potentielle et permet de prendre en compte le stress hydrique en incluant une fonction de compensation qui permet de simuler une extraction d'eau dans les couches profondes lorsque celles de la surface sont sèches. Une validation de ce module a été menée par Varado et al. (2006a). L'étude compare entre autre cette approche à celle développée par Lai and Katul (2000) et conclut qu'elle est plus efficace, notamment sur des sols sableux.

On note également que le bilan d'énergie n'est pas résolu dans le modèle proposé par Varado (2004) mais que l'on se base plutôt sur l'utilisation de l'évapotranspiration potentielle (ETP) et d'un coefficient cultural. Cette approche, peut-être moins rigoureuse, nécessite cependant moins de données que la résolution du bilan d'énergie qui requiert des données climatiques (température et humidité de l'air, vitesse du vent, rayonnement solaire, etc...). Les travaux présentés par Braud et al. (2005) ont permis de montrer que l'utilisation d'une ETP au lieu de la résolution explicite du bilan d'énergie menait généralement à la sous-estimation de la transpiration réelle. Le calcul de l'ETP est pointé comme étant une étape délicate à laquelle il est conseillé d'apporter le plus grand soin.

## 8.2 Représentation des processus

### 8.2.1 Interception (précipitations et radiations)

L'ETP est dans un premier temps fractionnée en ses deux composantes : l'évaporation potentielle (EP) et la transpiration potentielle (TP). L'EP sera utilisée directement comme condition limite en surface dans FRER1D et la TP permettra de déterminer l'extraction racinaire dans la colonne de sol. Cette dernière sera également utilisée comme condition limite latérale dans FRER1D (voir chapitre 7). Pour fractionner l'ETP en ses deux composantes, on utilise une loi de type Beer-Lambert où la séparation dépend de l'indice foliaire LAI (Leaf Area Index) comme proposé par Huygen et al. (1997) :

$$\begin{aligned} TP &= ETP(1 - \exp(-a_{bl}LAI)) \\ EP &= ETP\exp(-a_{bl}LAI) \end{aligned} \quad (8.1)$$

Avec :

- $a_{bl}$  : coefficient d'extinction, régissant l'interception des radiations par la végétation. On peut considérer que l'interception des précipitations et l'interception des radiations solaires sont soumises aux mêmes lois. La valeur choisie est la même que celle fixée dans Varado (2004) :  $a_{bl} = 0.5$

Les feuilles de la végétation constituent un réservoir d'interception pour l'eau dont la capacité maximale  $w_{rmax}$  (cm) peut être exprimée en fonction du LAI (Noilhan and Planton, 1989) :

$$w_{rmax} = 0.02\sigma_f LAI \quad (8.2)$$

où  $\sigma_f$  est la fraction de radiations solaires et de précipitations interceptée par la végétation (Taconet et al., 1986) :

$$\sigma_f = 1 - \exp(-a_{bl}LAI) \quad (8.3)$$

Les précipitations atteignant le sol correspondent donc à la somme de la fraction non interceptée par la végétation et à la fraction qui déborde du réservoir d'interception :

$$P_g = P * \exp(-a_{bl}LAI) + \frac{w_{rrunoff}}{dt} \quad (8.4)$$

Avec :

- $P_g$  : pluie atteignant effectivement la surface du sol
- $P$  : intensité de pluie au dessus de la canopée
- $w_{rrunoff}$  : volume d'eau qui déborde du réservoir d'interception

La dynamique du réservoir d'interception est gouvernée par l'équation suivante (Noilhan and Planton, 1989) :

$$\frac{\partial w_r}{\partial t} = (P - P_g) - E_w \quad (8.5)$$

Avec :

- $P$  : intensité de pluie au dessus de la canopée
- $P_g$  : pluie atteignant effectivement la surface du sol
- $E_w$  : taux d'évaporation de la canopée mouillée qui dépend de la fraction de végétation mouillée ( $\delta$ ) et de la transpiration potentielle ( $TP$ ). La formulation retenue pour  $\delta$  est celle de Deardorff (1978) :

$$\delta = \frac{w_r}{w_{rmax}}^{2/3} \quad (8.6)$$

L'énergie utilisée pour cette évaporation directe à la surface des feuilles est retenue sur la transpiration potentielle et ne peut donc pas être utilisée par la transpiration des plantes. La transpiration potentielle est donc mise à jour comme suit :

$$TP_f = TP - E_w = TP(1 - \delta) \quad (8.7)$$

### 8.2.2 Extraction racinaire : le modèle de Li and Zhang (2001)

L'extraction racinaire  $Ex(Z_i, t)$  dans une cellule  $i$  apparaît comme terme puits dans l'équation de Richards :

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z} \left[ K(h) \left( \frac{\partial h}{\partial z} - 1 \right) \right] - Ex(z, t) \quad (8.8)$$

Comme pour le terme puits/source correspondant aux flux latéraux saturés, le terme apparaît dans le calcul du coefficient  $d_i$  (cf chapitre 7) :

$$d_i = \left( \frac{q_{i-1/2}Area - q_{i+1/2}Area}{\sigma} + \frac{Ex_i}{\sigma} \right) \quad (8.9)$$

Elle est représentée par le modèle de Li and Zhang (2001) dans lequel elle s'exprime comme une fonction de la distribution racinaire, du stress hydrique et possède un mécanisme de compensation. Le terme puits  $Ex(Z_i, t)$  pour la cellule numérique  $i$  est défini Feddes et al. (1978) comme :

$$Ex(Z_i, t) = \chi(\theta_i, Z_i)TP_f \quad (8.10)$$

avec TP : transpiration potentielle et  $\chi(\theta, z)$  est une fonction empirique comprise entre 0 et 1 qui peut s'exprimer comme le produit de 3 termes (Braud et al., 2005) :

$$\chi(\theta_i, Z_i) = \alpha_1(\theta_i, Z_i)\alpha_2(\theta_i, Z_i)g(Z_i) \quad (8.11)$$

où  $\alpha_1$  représente le phénomène de compensation,  $\alpha_2$  celui de stress hydrique et  $g$  la distribution racinaire dans le profil de sol.

La distribution racinaire utilisée est celle proposée par Li and Zhang (2001). Pour la cellule numérique  $i$ , elle s'exprime :

$$g(Z_i) = \frac{\ln[1 + \exp(-bZ_i)/1 + \exp(-bZ_{i+1})] + 0.5[\exp(-bZ_i) - \exp(-bZ_{i+1})]}{\ln[2/1 + \exp(bZ_r)] + 0.5[1 - \exp(bZ_r)]} \quad (8.12)$$

Avec :

- $Z_i$  : profondeur (cm)
- $Z_r$  : profondeur maximale racinaire (cm)
- $b$  : un paramètre empirique qui s'exprime à l'aide de  $Z_r$  (cm), la profondeur maximale racinaire et de  $F_{10}$ , la fraction de densité racinaire présente dans les 10% supérieurs de la zone racinaire (-) (Li et al., 1999) :

$$b = \frac{24.66F_{10}^{1.59}}{Z_r} \quad (8.13)$$

La fonction de stress hydrique  $\alpha_2$  est basée sur le modèle trapézoïdal proposé par Feddes et al. (1978) pour tenir compte de la diminution d'extraction racinaire lorsqu'il y a stress hydrique, ou, au contraire, anoxie de la plante par excès d'eau.

$$\begin{cases} \alpha_2(\theta, z) = 1 & \text{si } h \geq h_1 \text{ ou } h \leq h_4 \\ \alpha_2(\theta, z) = 0 & \text{si } h \geq h_2 \text{ ou } h \leq h_3 \end{cases} \quad (8.14)$$

et  $\alpha_2$  qui augmente linéaire entre  $h_1$  et  $h_2$  et décroît linéairement entre  $h_3$  et  $h_4$ . Comme dans Li and Zhang (2001), on prendra dans la suite  $h_1 = -10$  cm,  $h_2 = -25$  cm,  $h_3^1 = -500$  cm et  $h_3^2 = -1000$  cm ; la valeur de  $h_3$  sera interpolée entre  $h_3^1$  et  $h_3^2$  selon la valeur de la transpiration potentielle.

La fonction de compensation  $\alpha_1$  s'écrit quant à elle :

$$\alpha_1(\theta_i, Z_i) = \frac{\alpha_2(\theta_i, Z_i)g(Z_i)^\lambda}{\sum_{i=1}^n \alpha_2(\theta_i, Z_i)g(Z_i)^\lambda} \quad (8.15)$$

Le paramètre  $\lambda$  est pris égal à 0.5 comme proposé par Li and Zhang (2001) et confirmé par Braud et al. (2005).

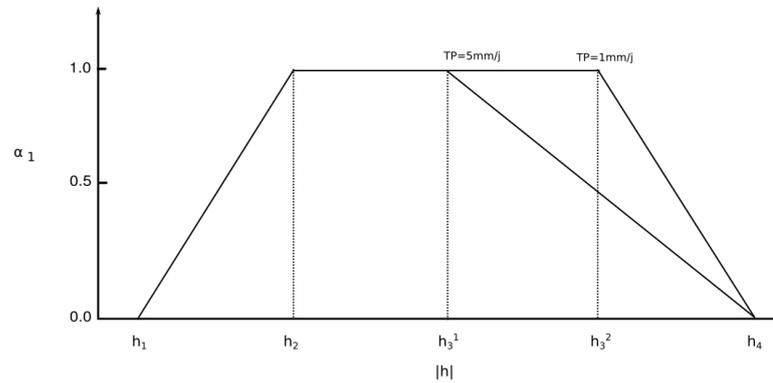


FIGURE 24 – Fonction de stress hydrique  $\alpha_2$  en fonction de la pression capillaire.

### 8.2.3 Utilisation d'un coefficient cultural

- Détailler l'utilisation du coefficient cultural comme conseillé dans la FAO pour la partition de l'ETP.
- Montrer les limites de l'utilisation d'un tel coefficient : difficultés à déterminer + incertitudes par rapport à l'utilisation d'un bilan radiatif, notamment illustrées dans la thèse de Noémie.

## 8.3 Implémentation

### 8.3.1 PARTITION dans les modules FRER1D et FRER1D\_HEDGE

Le bloc PARTITION codé dans FRER1D et FRER1D\_HEDGE permet de calculer les taux d'évaporation et de transpiration potentielles ainsi que le taux de précipitation atteignant le sol. Le bloc prend comme paramètres et variables d'entrée :

- `abl(double)` : coefficient d'extinction
- `crop(double)` : coefficient cultural
- `LAI(double)` : indice de surface foliaire
- `dt(double)` : pas de temps courant (h)
- `pet, rain(double)` : ETP, taux de précipitations au dessus de la canopée (cm/h).

L'ensemble de ces paramètres/variables sont obtenus à partir de PALM avec un tag identifiant l'élément dont il est question. Un booléen (`interception`) permet d'activer ou pas le calcul de l'interception puis la partition de l'ETP en transpiration/évaporation est calculée. Les 3 variables (précipitation atteignant le sol, évaporation et transpiration) sont ensuite données en entrée de la subroutine SOLVE et le niveau actualisé du réservoir d'interception est envoyé à PALM.

### 8.3.2 ROOTS

Le module ROOTS est un module FORTRAN qui sera appelé par le module FLOW si le calcul de l'extraction racinaire est activé. Il implémente le modèle d'extraction racinaire de Li and Zhang (2001) et se présente sous la forme de deux sous-routines :

setroots(x,F10,Zr) et qetex(Pt,h,jt,rex,rexh).

obs : la plupart des paramètres nécessaires (transpiration, paramètres pour la détermination de la fonction de stress hydrique  $\alpha_2$  et celle de compensation  $\alpha_1$ ) sont définis comme *Public*, c'est-à-dire qu'ils peuvent être initialisés dans FLOW et n'ont pas besoin d'être reprécisés dans ROOTS.

### setroots

Cette subroutine permet de définir la distribution de densité racinaire dans un profil de sol à partir de la distribution définie par Li and Zhang (2001). Les paramètres d'entrée sont :

- x(double(1 :n)) : profondeur du fond des cellules numériques (cm) de la plus proche de la surface à la plus éloignée
- F10(double) : Fraction de densité racinaire présente dans les 10% supérieurs de la zone racinaire
- Zr(double) : profondeur maximale racinaire (cm)

A partir de ces paramètres et variables d'entrée, la subroutine détermine la densité racinaire pour chaque cellule numérique qu'elle stocke dans le vecteur Fs. Pour les cellules situées au dessus de Zr, on se base sur l'équation 8.12, pour toutes les autres, l'élément du vecteur Fs est forcé à 0.

### 8.3.3 getrex

Cette subroutine permet de déterminer le taux d'extraction racinaire rex(double(1 :n)) (cf eq. 8.11) et sa dérivée par rapport à la pression h rexh(double(1 :n)) à partir des données de pression capillaire h(double(1 :n)) et de densité racinaire déterminée par setroots .

# Chapitre 9

## Écoulements dans un réseau

### 9.1 Introduction

Pour atteindre une représentation des transferts d'eau et de solutés à l'échelle du bassin versant, PESHMELBA doit intégrer une représentation des écoulements dans un réseau hydrographique (rivières et/ou fossés). Ces écoulements sont classiquement modélisés grâce aux équations de St Venant ou une de ses approximations. Les équations de Barré Saint-Venant (1871) permettent classiquement de décrire les écoulements dans un canal dans le cadre des hypothèses suivantes :

- L'eau est incompressible. Sa densité et sa viscosité sont constantes.
- La distribution de pressions est hydrostatique sur la verticale, les ligne de courant sont donc horizontales.
- L'amplitude des vagues de surface est négligeable devant leur longueur d'onde.
- La pente du lit est faible et le lit est fixe.
- S'il y a des apports latéraux, la quantité de mouvement qu'ils injectent dans le canal est faible par rapport à la quantité de mouvement au sein du canal.

Les équations de Barrés Saint Venant sont constituées d'une équation de conservation de la masse et d'une équation de conservation du moment :

$$\frac{\partial A}{\partial t} = -\frac{\partial Q}{\partial x} + q \quad (9.1)$$

$$\underbrace{\frac{\partial}{\partial t}(\rho AV) + \frac{\partial}{\partial x}(\rho AV^2)}_{\text{Onde dynamique}} + \underbrace{A \frac{\partial p}{\partial x} + \underbrace{\tau P - \rho g AS_0}_{\text{Onde cinématique}}}_{\text{Onde diffusante}} = 0 \quad (9.2)$$

où  $\rho$  est la densité de l'eau,  $A$  [L<sup>2</sup>] la section transversale,  $V$  [LT<sup>-1</sup>] la vitesse de l'écoulement,  $g$  [LT<sup>-2</sup>] l'accélération de la pesanteur,  $P$  [L] le périmètre mouillé,  $S_0$  [LL<sup>-1</sup>] la pente,  $p$  [L] la pression hydrostatique locale,  $q$  les apports latéraux [L<sup>2</sup> T<sup>-1</sup>] et  $\tau$  le coefficient de frottement du fond ( $\tau = \rho g R S_f$  où  $R$  est le rayon hydraulique [L] et  $S_f$  la pente de friction [LL<sup>-1</sup>]).

Plusieurs approximations de ces équations basées sur des simplifications de l'équation

de conservation du moment sont communément utilisées. L'approximation de l'onde diffusive néglige les termes d'inertie tandis que l'approximation de l'onde cinématique ne prend en compte que les effets de friction et de perte de charge. L'équation 9.2 se réduit alors à  $S_f = S_0$ . Bien qu'elle soit très simplifiée, l'équation de l'onde cinématique offre l'avantage d'être robuste et permet dans certains cas de s'affranchir de problèmes numériques liés à l'utilisation des équations de Saint-Venant (Weill, 2007).

De plus, des relations empiriques reliant la hauteur d'eau et la vitesse moyenne d'écoulement sont communément utilisées. C'est le cas de la relation de Manning-Strickler :

$$V = \frac{h^{2/3}}{n} \sqrt{S_f} \quad (9.3)$$

où  $n$  [ $\text{LT}^{-1/3}$ ] est le paramètre de Manning-Strickler caractérisant la rugosité du sol.

## 9.2 Représentation des écoulements dans un réseau

Pour modéliser le routage de l'eau et des solutés dans un réseau ramifié, PESHMELBA se base sur le module River1D déjà implémenté dans la plateforme LIQUID (Branger et al., 2008). River1D permet de simuler des écoulements dans un réseau ramifié avec une géométrie simplifiée (sections symétriques trapézoïdales) (Branger, 2007a). Il permet également de prendre en compte des termes puits-source pouvant correspondre par exemple aux échanges nappe-rivière, à l'intersection du ruissellement ou à des précipitations ([il faudra rajouter un second terme d'extraction correspondant à l'évaporation](#)). Pour être intégré à PESHMELBA, le module initial de LIQUID a été réadapté et recodé en Python. En effet, les bassins versants simulés dans PESHMELBA, les réseaux considérés sont complexes puisqu'ils peuvent posséder des confluences et des diffluences, notamment dans le cas de réseaux de fossés et la gestion de ces configurations a été intégrée dans RIVER1D. A noter que la version actuelle de River1D ne permet pas de simuler un débordement ni un assèchement de cours d'eau. Le module devra être modifié ultérieurement pour intégrer ces situations. La section qui suit est largement inspirée du rapport technique de River1D (Branger, 2007a).

### 9.2.1 Devenir des pesticides dans un fossé ou une rivière

Comme dans une colonne de sol, le devenir des pesticides dans un fossé est régi par la combinaison de plusieurs processus. Dans un premier temps, on ne représente que le transport par convection (la dispersion et la diffusion ne sont pour le moment pas représentées), la dégradation ainsi que l'adsorption/désorption.

Les flux advectés le long du réseau sont d'abord calculés à partir des flux d'eau et des concentrations dissoutes. Puis, à la fin du pas de temps, les quantités dégradées sont calculées à partir des concentrations dissoutes et de la durée du pas de temps. Pour finir, on calcule l'équilibre d'adsorption. Contrairement à une colonne de sol, l'eau et les solutés qu'elle contient n'est en contact avec le substrat solide permettant l'adsorption que sur la surface mouillée du fossé. Dans un premier temps, on suppose que l'adsorption ne peut se faire que sur une bande de sol à la hauteur paramétrable ( $h_{ads}$ ) au fond du fossé et dont il faudra fournir la masse volumique ( $\rho_b$ ) et la teneur en eau à saturation ( $\theta_s$ ). On considère un mélange complet, c'est-à-dire que sur l'épaisseur de sol  $h_{ads}$  le

sol/substrat peut être considéré comme en suspension dans l'eau du fossé, autorisant une adsorption complète et instantanée. D'après Dollinger et al. (2016), l'épaisseur d'adsorption  $h_{ads}$  peut être fixée à 2 cm **Plus tard, on essaiera de prendre en compte les bords verticaux du fossé pour l'adsorption. Il faudra à ce moment-là réfléchir à la désorption pour s'assurer que lorsque la hauteur d'eau baisse, le stock de solutés disponible pour la désorption ne tient plus compte de la surface à présent hors de l'eau.**

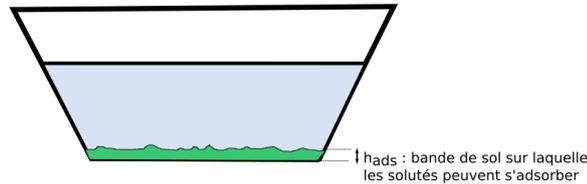


FIGURE 25 – Bande de sol disponible pour l'adsorption

## 9.2.2 Domaine de modélisation, représentation d'un réseau ramifié

### Propriétés d'un brin

Dans PESHMELBA comme dans LIQUID, un réseau est composé de brins (ou tronçons) interconnectés formant un réseau ramifié avec un exutoire unique. Dans PESHMELBA, un brin correspond à un élément linéaire élémentaire déterminé par GeOMELBA. Chaque brin possède une section trapézoïdale symétrique définie par une largeur au fond  $w$  [L] et un fruit des berges  $m$  [L/L] (voir Figure 26).

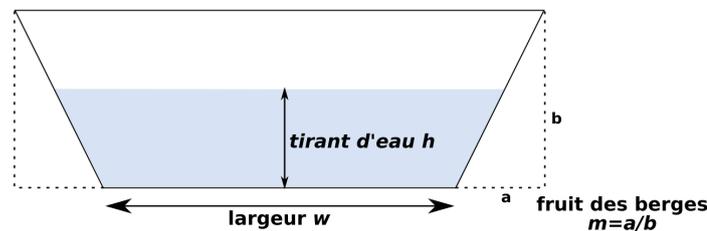


FIGURE 26 – Caractéristiques géométriques d'une section trapézoïdale de brin.

Le brin est également défini par une pente moyenne  $S$  [L/L]. On rappelle les différentes grandeurs géométriques qui peuvent être exprimées en fonction du tirant d'eau  $h$  [L] :

- Section mouillée :  $h(w + hm)$
- Largeur au miroir :  $w + 2hm$
- Périmètre mouillé :  $w + 2h\sqrt{1 + m^2}$
- Rayon hydraulique :  $\frac{h(w+hm)}{w+2h\sqrt{1+m^2}}$

### Ordre de parcours

Dans la méthode de résolution de River1D, les brins sont parcourus de l'amont vers l'aval et il faut définir le sens de parcours du/des réseau(x) pour procéder à la résolution

numérique. Un bassin versant peut contenir plusieurs réseaux (fossés, rivières,...), chacun caractérisé par d'éventuelles confluences et/ou diffluences. Il est donc nécessaire de procéder à une première étape d'analyse du paysage et des réseaux pour :

- **Déterminer les réseaux indépendants** : on considère qu'un ensemble de brins constitue un réseau indépendant s'il possède un exutoire unique et si tous les brins sont de la même nature (rivières ou fossés). Ainsi, un réseau de fossés se jetant dans un brin de rivière sera simulé comme un réseau indépendant puis pris en compte dans le réseau de rivière comme un apport latéral. On se base sur la table de connexions longitudinales fournie par GeoMELBA pour déterminer ces réseaux
- **Déterminer un ordre de parcours dans chaque réseau indépendant identifié** : GeoMELBA fournit une altitude moyenne pour chaque brin et se base sur cette information pour établir les connexions longitudinales entre les différents éléments linéaires. On se base également sur cette altitude pour déterminer un ordre de parcours dans le réseau : les brins sont parcourus depuis les altitudes les plus hautes vers les altitudes les plus basses.

Un exemple est présenté Figure 27.

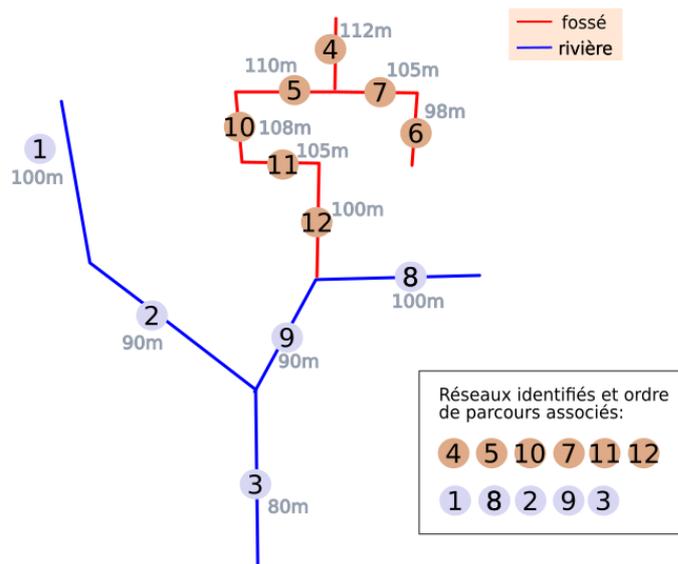


FIGURE 27 – Exemple de réseaux et ordre de parcours associés.

### 9.3 Schéma de résolution

On rappelle que River1D se base sur l'approximation de l'onde cinématique, laquelle néglige tous les termes de l'équation dynamique mis à part le terme de pertes de charges dues au frottements. Cela revient à assimiler l'écoulement à une succession d'états permanents uniformes. La vitesse est ensuite calculée en fonction du rayon hydraulique à l'aide de la formule de Manning :

$$V = \frac{1}{n} \sqrt{i} R_H^{\frac{2}{3}} \quad (9.4)$$

où  $V$  est la vitesse d'écoulement [L/T],  $n$  le coefficient de rugosité de Manning,  $i$  [L/L] la pente du fond du lit et  $R_H$  le rayon hydraulique [L].

La résolution se fait alors pour chaque brin et la variable de calcul utilisée est la section mouillée. Pour chaque brin, entre et  $t$  et  $t + \Delta t$ , le bilan de masse s'écrit :

$$\frac{\Delta S}{\Delta t} = \frac{Q_{in} + Q_{out} + q_{lat}}{L} \quad (9.5)$$

où  $L$  est la longueur du brin [L]. Le débit entrant  $Q_{in}$  est égal à la somme des débits sortants des brins amont. Le débit sortant est égal au produit de la section mouillée et de la vitesse dans le brin ( $Q_{out} = SxV$ ).  $q_{lat}$  regroupe l'ensemble des terme puits-source latéraux, positifs ou négatifs : précipitations, connexion à l'exutoire d'un autre réseau, interception du ruissellement, échanges nappe-rivière, évaporation ([à implémenter](#)). La résolution est explicite et fait appel à l'algorithme de Runge-Kutta au quatrième ordre (détaillé en Annexe B). L'équation de continuité est ré-écrite comme suit :

$$\begin{aligned} \frac{\partial S}{\partial t} &= q_{apport} - \frac{\partial Q}{\partial x} \\ \Rightarrow \frac{\partial S}{\partial h} \frac{\partial h}{\partial t} &= q_{apport} - \frac{\partial Q}{\partial x} \\ \Rightarrow \frac{\partial h}{\partial t} &= \frac{\partial h}{\partial S} \left\{ q_{apport} - \frac{\partial Q}{\partial x} \right\} \end{aligned}$$

On résout finalement :

$$\frac{\Delta h}{\Delta t} = \frac{\partial h}{\partial S} \left\{ \frac{Q_{in} + Q_{out} + Q_{lat}}{L} \right\} \quad (9.6)$$

La résolution se fait d'abord dans les réseaux de fossés pour que le débit à l'exutoire puisse être considéré comme un terme source dans un brin de rivière s'ils sont effectivement connectés (comme dans l'exemple de la Figure 27).

## 9.4 Implémentation

### 9.4.1 Analyse de réseau

L'analyse du réseau permet de déterminer les réseaux d'écoulement indépendants et l'ordre de parcours des tronçons dans chacun de ces réseaux, du plus élevé au plus bas comme décrit section 9.2.2. Ceci est fait par la fonction `get_order_reach` du script `landscape_analyse.py`. Cette fonction prend en entrée :

- `param_table(0:n_layer,2)` : table contenant un ensemble d'indices de linéaires (colonne 1) et leur altitude moyenne associée (colonne 2).
- `connect_table(0:nb_connect,2)` : table contenant les indices des linéaires connectés. La première colonne indique l'indice de l'élément amont et la seconde, celle de l'élément aval.

A partir de ces informations, la fonction identifie d'abord les réseaux indépendants en parcourant la table de connections à partir des têtes de bassin. Les éléments de chaque réseaux sont ensuite triés de l'altitude la plus élevée vers l'altitude la plus basse. Si aucun réseau n'est identifié, la fonction crée autant de réseaux indépendants qu'il y a d'éléments. Ceci pour être sûr que chaque tronçon sera traité par River1D même s'il ne

fait parti d'aucun réseau.

En sortie, la fonction renvoie une liste composée d'autant de listes que de réseaux indépendants. Les éléments de chacune de ces listes sont triés par altitude décroissante.

### 9.4.2 Classe Reach

La classe Reach permet de traiter chaque tronçon du réseau en particulier. Celle-ci hérite de la classe Linear et les méthodes qu'elle contient permettent de calculer les grandeurs géométriques caractérisant le tronçon et d'appliquer la méthode de résolution de Runge-Kutta-Cash-Karp sur chaque tronçon indépendant.

#### \_\_init\_\_

Une instance de la classe Reach s'initialise avec les mêmes éléments que la classe Linear ainsi qu'avec les attributs suivants :

- width\_bottom (scal) : largeur du fond du lit
- slope (scal) : pente moyenne du fond de la rivière
- batter (scal) : fruit des berges
- adsorption\_thickness (scal) : épaisseur de la couche d'adsorption en fond de lit
- bd (scal) : masse volumique du sol du fond du lit
- ths (scal) : teneur en eau volumique à saturation en fond de lit
- manning\_coeff (scal) : coefficient de rugosité de Manning
- hpond (scal) : hauteur de ponding autorisée
- DownstreamElements(0 :n) : vecteur contenant les indices des linéaires connectés en longitudinal à l'aval du tronçon initialisé

Les attributs width\_move (scal) et pond\_full (bool) sont également initialisés. Le premier permet de calculer la largeur du lit au dessus de la couche de ponding qui reste immobile quelque soit les circonstances. Le second est un booléen qui passe à 1 lorsque la hauteur d'eau dans le tronçon dépasse la hauteur de ponding.

⚠ On note que le ponding ne fonctionne pas de la même manière que sur les éléments surfaciques. Pour les linéaires, seule l'eau située au dessus de la couche de ponding pourra se mettre en mouvement. Sur les éléments surfaciques, le ruissellement ne se déclenche que lorsque la hauteur d'eau dépasse la hauteur de ponding et dans ce cas, c'est l'ensemble de la colonne d'eau qui se met en mouvement !

### GeometrySimple

A partir d'un tirant d'eau donné en entrée **WaterLevel**, la méthode GeometrySimple calcule les différentes grandeurs géométriques caractéristiques de la section trapézoïdale : section mouillée, largeur au miroir, périmètre mouillé, rayon hydraulique dont les formules sont données Section 9.2.2. On calcule également la dérivée du niveau d'eau par rapport à la section mouillée (grandeur nécessaire au schéma de résolution) :

$$\frac{1}{\sqrt{w^2 + 4 * WSec * m}} \quad (9.7)$$

où  $W_{Sec}$  est la section mouillée.

L'ensemble de ces grandeurs sont calculées soit en considérant le véritable fond du brin, si le booléen **isponded** est Faux, soit en considérant que le fond du tronçon se situe au dessus de la hauteur de ponding si le booléen est vrai.

### WaterContent2WaterLevel

La méthode `WaterContent2Waterlevel` permet de convertir un volume d'eau en une hauteur d'eau à partir d'une géométrie connue. L'objectif de cette méthode étant de fournir la hauteur d'eau qui sera utilisée pour la résolution de l'équation de l'onde cinématique, il est nécessaire de ne considérer que l'eau qui pourra effectivement se déplacer dans le réseau. C'est-à-dire l'eau située au dessus de la hauteur de ponding renseignée par l'utilisateur. Cette méthode prend comme argument d'entrée le volume d'eau disponible dans le tronçon et renvoie en sortie une hauteur d'eau calculée à partir d'un fond virtuel au niveau de la hauteur maximale de ponding **WaterLevel\_RO** et une hauteur d'eau calculée depuis le vrai fond du tronçon **WaterLevel\_TOTAL**. Ces deux variables sont illustrées figure 28.

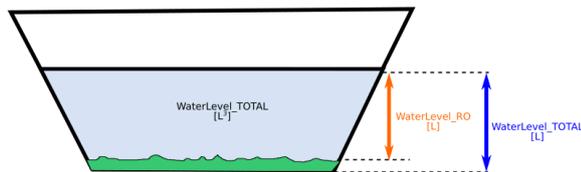


FIGURE 28 – Variables `WaterContent`, `WaterLevel_TOTAL`, `WaterLevel_RO`

### RKCK\_RightHandSide

Cette méthode permet de calculer le membre de droite de l'équation 9.6. Elle prend en entrée une hauteur d'eau **WaterLevel** laquelle sera utilisée pour appeler la méthode `GeometrySimple`.

### RungeKuttaCashKarp

Cette méthode met en oeuvre la méthode de résolution de Runge-Kutta à l'ordre 4 avec les coefficients de Cash (voir Annexe B). On rappelle que la variable de résolution est une hauteur d'eau, initialisée à **Y0** et le membre de droite de l'équation de continuité 9.6 ainsi que le pas de temps utilisé pour la résolution sont également donnés en entrée.

## 9.4.3 Classe Network

La classe `Network` réunit plusieurs tronçons de rivière ou fossé pour former un réseau sur lequel est résolue l'équation de l'onde cinématique. Elle est construite par association [association ou... ? je ne sais jamais lequel](#) d'instances de la classe `Reach`. Chaque réseau se caractérise également par un ordre de parcours des tronçons qui est déterminé dans la section 9.4.1. On précise aussi le type d'éléments qui composent le réseau (un réseau ne peut être constitué que d'un seul type d'élément).

### `__init__`

Une instance de la classe `Network` s'initialise à partir de :

- `idx(scal)` : un indice pour identifier le réseau ;
- `nature(string)` : une chaîne de caractères indiquant si le réseau est composé de fossés ("ditch") ou de rivières ("river") ;
- `ReachDic(dictionnaire)` : un dictionnaire contenant **m** instances de la classe `Reach` ;
- `RoutingOrder(0 :m)` : un vecteur ordonné qui définit l'ordre de parcours des tronçons dans le réseau ;
- `Outlet(0 :2,0 :l)` : un tableau contenant l'ensemble des connexions entre réseaux. Pour chaque connexion, la première colonne contient l'indice du tronçon exutoire du réseau amont connecté à un autre réseau aval. La seconde colonne contient l'indice du tronçon de l'autre réseau dans lequel se jette le tronçon repéré dans la première colonne. Par exemple, dans le cas de la Figure 10, le tableau `Outlet` contiendrait une seule ligne : "12 ; 9".
- `WaterOutlet(scal)` : le volume d'eau cumulé qui a traversé l'exutoire identifié pour le réseau donné. Cette variable est incrémentée à condition que l'exutoire du réseau soit effectivement connecté à un autre réseau en aval. Cette connexion est renseignée dans le tableau **Outlet**.
- `SoluteOutlet(1 :n_solute)` : pour chacun des soluté simulé, la masse cumulée qui a traversé l'exutoire identifié pour le réseau donné. A nouveau, cette variable est incrémentée à condition que l'exutoire du réseau soit effectivement connecté à un autre réseau en aval.

### `handle_RKCK_Resolution`

C'est la méthode principale de la classe `Network`. Elle permet de résoudre l'équation de l'onde cinématique sur le réseau en utilisant la méthode de RKCK. Durant un pas de temps, le niveau d'eau dans chaque tronçon du réseau est mis à jour ; plusieurs fois, si le pas de temps déterminé pour la résolution le nécessite. Les opérations suivantes sont effectuées jusqu'à arriver à la fin du pas de temps :

- Les tronçons sont parcourus dans l'ordre et le débit d'eau entrant et sortant de chacun d'eux est déterminé en fonction du niveau d'eau et des éléments situés en amont et en aval. Le débit à destination d'un autre réseau éventuellement connecté à l'aval est stocké.
- Le pas de temps pour la méthode RKCK est déterminé de manière à garder l'erreur de la solution inférieure à un seuil fixé par la variable **Tolérance**. A noter que dans ce bloc de code, le pas de temps ne peut-être qu'augmenter si la contrainte sur l'erreur n'est pas atteinte et que la solution est "trop" précise. Une fois qu'un pas de temps suffisamment grossier a été identifié, on appelle une autre méthode `EvalTimeStep`, décrite plus bas pour vérifier spécifiquement que le pas de temps permet toujours d'obtenir une solution suffisamment précise. [Je n'ai pas bien compris pourquoi ces deux étapes sont séparées et ne peuvent pas se faire en un seul bloc. J'ai repris cette structure du code de LIQUID après m'être rendue compte qu'une structure plus simple, en un bloc ne conduisait pas à des résultats satisfaisants...à creuser.](#)

- Les tronçons sont parcourus dans l'ordre et le débit de solutés entrant et sortant de chacun d'eux est déterminé en fonction des éléments situés en amont et en aval. A nouveau, le débit à destination d'un autre réseau éventuellement connecté à l'aval est stocké.
- Le transport advectif de solutés est calculé en faisant un bilan de masse sur chaque tronçon. La masse disponible dans chaque tronçon est mise à jour.
- La résolution de RKCK est lancée sur chaque tronçon à partir des débits d'eau entrants, sortants. Pour cela, la méthode RungeKuttaCashKarp de la classe Reach est appelée. Ceci permet de mettre à jour la variable hauteur d'eau (au dessus de la hauteur de ponding) puis d'actualiser le volume d'eau total dans le tronçon ainsi que l'état du réservoir de ponding (plein ou non).
- A partir de ces nouveaux volumes d'eau mis à jour dans chaque tronçon, la quantité de solutés dissous est elle aussi mise à jour dans chaque tronçon.
- Enfin, les variables de forçage cumulé sont mises à jour.

### **EvalTimeStep**

Cette méthode détermine le pas de temps nécessaire à la résolution RKCK permettant de respecter le seuil de qualité fixé par la variable **Tolerance**. Pour cela, une première résolution à partir d'un pas de temps de base est simulée sur tout le réseau. On appelle pour cela la méthode RungeKuttaCashKarp de la classe Reach. On stocke alors l'erreur obtenue conformément aux recommandations faites dans (Press et al., 1992) et re-détaillées en Annexe B. En comparant l'erreur obtenue à la variable **Tolerance**, on adapte alors le pas de temps en le raffinant si l'erreur sur la solution n'est pas satisfaisante. Le processus est itératif jusqu'à obtenir un résultat satisfaisant.

### **adsorption**

L'équilibre d'adsorption pour tous les tronçons d'un réseau est calculé par cette méthode. Pour chacun des tronçons, la méthode récupère ou détermine le volume d'eau total et le volume d'eau en contact avec la bande d'adsorption. A nouveau, on fait l'hypothèse que dans la bande d'adsorption, le substrat/sol est en suspension dans l'eau du fossé. L'équilibre d'adsorption est alors déterminé dans cette bande d'adsorption.

### **degradation**

Pour chacun des solutés simulés, la quantité de solutés dégradée est déterminée, les stocks sont mis à jour en alimentant éventuellement les stocks de métabolites.

### **9.4.4 Module use\_launch\_RIVER1D**

Le module use\_launch\_RIVER1D permet de créer toutes les instances des classes Reach et Network permettant la résolution de l'onde cinématique sur les différents réseaux du bassin versant simulé. C'est également le module qui fait le lien avec le reste du modèle en récupérant puis transmettant toutes les variables et paramètres stockés dans des variables PALM. Pour cela, les étapes suivantes sont effectuées :

- Réception de toutes les données relatives à la description du BV : liste des linéaires, des fossés, des rivières, etc... ;
- Chargement des données décrivant chaque tronçon de fossé et initialisation d'une instance de la classe Reach pour chacun d'entre eux ;
- Chargement des données décrivant chaque tronçon de rivière et initialisation d'une instance de la classe Reach pour chacun d'entre eux ;
- Identification des différents réseaux de fossés indépendants et de l'ordre de parcours qui les caractérise. Pour cela, on fait appel à la fonction d'analyse du paysage spécialement développée **get\_order\_reach**. A partir d'une table de connexions entre tronçons et d'une information sur l'altitude de ces tronçons, cette fonction renvoie le nombre de réseaux indépendants identifiés et une liste contenant des listes ordonnées d'indices de tronçons pour chaque réseau. Chacune de ces listes ordonnées d'indices correspond à l'ordre de parcours du réseau.
- Identification des différents réseaux de rivières indépendants et de l'ordre de parcours qui les caractérise ;
- Initialise de la simulation : pour cela, il faut initialiser le niveau d'eau et la vitesse de Manning dans chaque tronçon de fossé et de rivière. ;
- Lancement de la simulation entre  $t$  et  $t+dt\_PALM$ . La résolution se fait d'abord sur les réseaux de fossés. Ainsi, cela permet d'obtenir un éventuellement terme source correspondant à un réseau de fossés dont l'exutoire serait relié à la rivière. Ce terme source déjà calculé pourrait alors être injecté au même pas de temps lors de la résolution pour les réseaux de rivière.

# Chapitre 10

## Transferts latéraux de subsurface

### 10.1 Considérations générales sur l'implémentation des lignes d'échanges

L'objectif de PESHMELBA n'est pas seulement de représenter des processus à l'échelle de chaque élément du paysage individuellement mais aussi, et principalement, de représenter les interactions entre ces éléments. Ces interactions sont intégrées sous forme de **lignes d'échanges** que l'on distingue selon qu'il s'agisse d'une ligne d'échanges de surface ou de subsurface (voir Figure 17).

Lors de la construction du modèle, on implémente progressivement différents types de lignes d'échanges (parcelle-parcelle, puis parcelle-fossé-parcelle, puis parcelle-haie, etc...) et l'ensemble se doit de rester suffisamment flexible pour toujours permettre l'intégration de nouveaux éléments et de nouvelles connexions entre eux. Ces objectifs de flexibilité et de généricité justifient l'utilisation de notions simples de Programmation Orientée Objet et d'un diagramme UML structuré pour guider l'élaboration des différents modules. On définit ainsi des classes génériques `SurfaceInterface` et `SubsurfaceInterface` qui peuvent connecter n'importe quels types d'éléments. De ces classes génériques, héritent ensuite un ensemble de classes plus spécialisées qui décrivent spécifiquement chaque type de ligne d'échanges : `ColumnColumnExchangeLine`, `ColumnDitchColumnExchangeLine`, `ColumnRiverExchangeLine`, `ColumnBCSEExchangeLine`, `ColumnHedgeExchangeLine`, etc... Ce chapitre décrit l'ensemble de ces classes à partir du fonctionnement des lignes d'échanges auxquelles elles réfèrent. Puisque chacune de ces classes hérite de la classe `SubsurfaceInterface`, elles se basent entre autre sur son constructeur pour initialiser leurs propres instances. Toute instance de la classe `SubsurfaceInterface`, ou d'une classe héritant de cette dernière (sous-classe) est initialisée à partir des attributs suivants :

- `idx` : un index (scalaire entier) pour repérer la ligne d'échanges ;
- `length` : une longueur [L] ;
- `x,y` : une longitude et une latitude [L] ;
- `z` : une altitude [L] ;
- `UpstreamInstance` : une instance d'une sous-classe de la classe `Element`. Il s'agit de l'élément situé à l'amont de la ligne d'échanges.
- `DownstreamInstance` : une instance d'une sous-classe de la classe `Element`. Il s'agit de l'élément situé à l'aval de la ligne d'échanges.

- `InterceptInstance` : une instance d'une sous-classe de la classe `Element` qui constituera un potentiel élément intercepteur entre l'élément amont et l'élément aval de la ligne d'échanges. Pour le moment, cet attribut est seulement utilisé pour coder la ligne d'échanges `ColumnDitchColumnExchangeLine`.

Que se passe-t-il dans des cas du genre "Parcelle-fossé-talus-Parcelle" ? A décrire quand ces cas seront implémentés.

En plus de ce constructeur, la classe `SubsurfaceInterface` contient deux méthodes abstraites : `update_state_variables()` et `calculate_flux_interface()`. Il s'agit de méthodes que chaque sous-classe doit aussi contenir et décrire explicitement car on attend un comportement différent selon le type de ligne d'échanges manipulée. Dans la classe `SubsurfaceInterface`, ces méthodes abstraites jettent seulement l'exception `NotImplementedError`. Elles constituent un exemple simple de polymorphisme en Python.

L'intérêt d'une telle implémentation est illustré dans la dernière section de ce chapitre. Cette dernière détaille l'implémentation du module `EXCHANGES_SUBSURFACE.py` qui apparaît dans le schéma PALM. C'est un module relativement générique qui exploite les concepts d'héritage et de polymorphisme. Ce module est capable de gérer tous les types de lignes d'échanges de subsurface.

## 10.2 Transferts entre deux parcelles

### 10.2.1 Rôle de la ligne d'échanges `ColumnColumnExchangeLine`

- La ligne d'échanges `ColumnColumnExchangeLine` doit assurer le calcul d'écoulements latéraux saturés entre deux colonnes de sols. En plus des flux d'eau, elle doit permettre de calculer les flux de solutés advectés.
- La ligne d'échanges doit gérer des nappes en fond de profil mais aussi des nappes perchées.
- Idéalement, le calcul du flux latéral entre deux zones saturées se fait à partir du gradient hydraulique entre les deux éléments (Loi de Darcy). Cependant, pour simplifier on impose un sens de parcours lié à la topographie (comme cela est fait dans le modèle TOPOG (Vertessy et al., 1993)). Le calcul du flux latéral est fait à partir du gradient hydraulique mais celui-ci sera effectif seulement s'il est compatible avec le sens du gradient topographique. Sinon, on considère le transfert comme nul (cf fig 29). On s'interdit ainsi les cas où l'eau "remonte" la pente mais il est peu probable de rencontrer une telle situation sur les bassins étudiés.

### 10.2.2 Cahier des charges

#### Repérage des nappes

- Nappe en fond de profil :  
Si une nappe existe en fond de profil de la colonne amont, on regarde s'il existe une nappe en fond de profil de la colonne aval. Si c'est le cas et que le gradient hydraulique respecte le sens de la pente, on peut effectivement calculer le flux latéral. S'il existe bien une nappe en fond de profil de la nappe aval mais que le

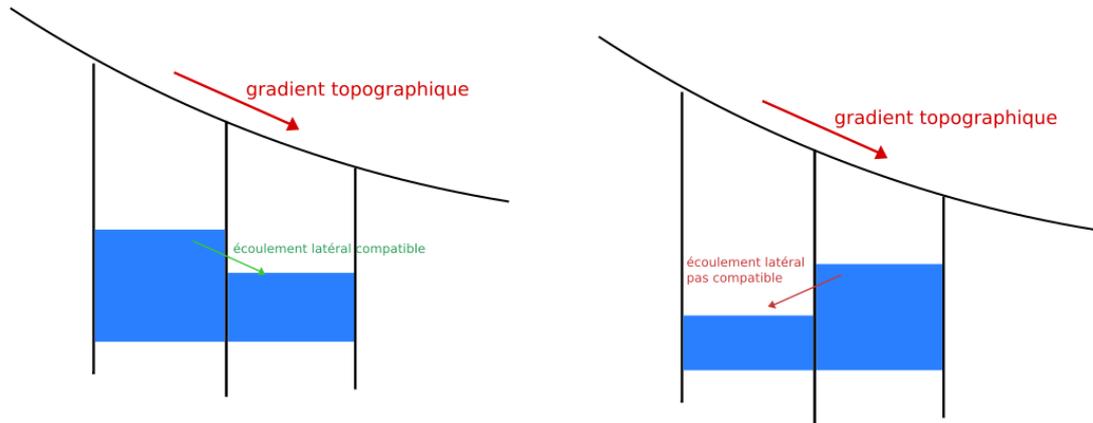


FIGURE 29 – Gradient hydraulique vs gradient topographique. Gauche : gradient hydraulique compatible avec le sens de la pente. Droite : gradient hydraulique non compatible avec le sens de la pente.

sens du gradient hydraulique est opposé au sens du gradient topographique, on ne calcule pas de flux latéral. Si la colonne aval ne présente pas de nappe en fond de profil, le flux latéral correspondant à la zone saturée amont sera routé vers la dernière cellule de la colonne aval.

- Nappe perchée :

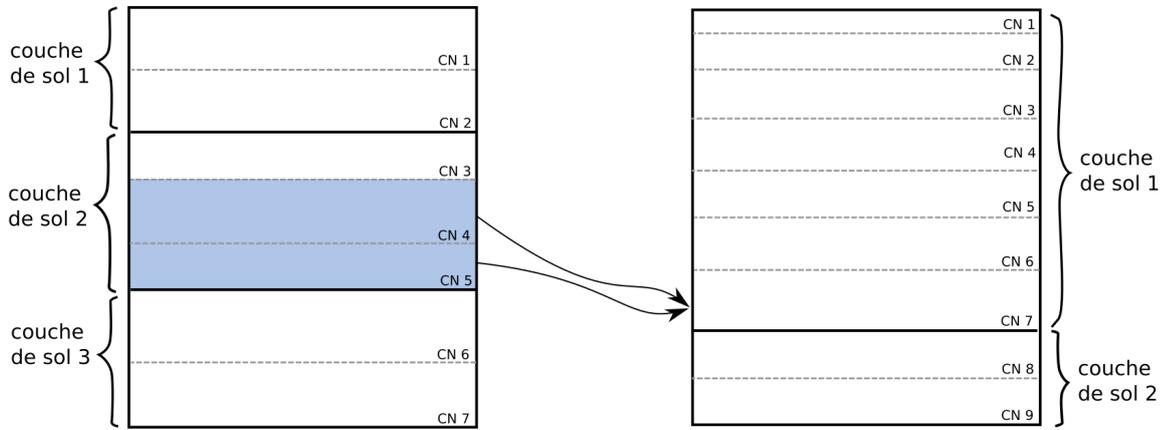
Une zone saturée est considérée comme une nappe perchée si le fond de nappe est "posé" sur le fond d'un horizon de sol. Dans le cas contraire, on considère qu'il s'agit d'une nappe fugace, qui ne doit pas être transférée, les flux latéraux associés à cette nappe étant négligés par rapport au gradient gravitaire appliqué sur la colonne.

On raisonne comme pour les nappes en fond de profil. S'il existe une nappe perchée dans la colonne amont, le flux calculé sera routé vers le fond de l'horizon de sol le plus proche (mais toujours plus profond) de la colonne aval. S'il existe déjà une zone saturée à ce niveau de la colonne aval, on vérifie que le gradient hydraulique calculé est compatible avec le sens de la pente pour effectivement calculer un flux latéral (fig 30).

Si la nappe perchée amont ne peut pas s'écouler car la colonne aval est saturée depuis le fond jusqu'à une hauteur supérieure, sans discontinuité, on considère en effet que l'écoulement est bloqué. Cependant si l'écoulement ne peut se faire à cause d'une nappe perchée plus haute située à l'aval, on transfère quand même l'eau de la nappe amont vers la dernière cellule du prochain horizon de sol rencontré, de sorte que l'eau ne soit pas "coincée" comme présenté fig 31.

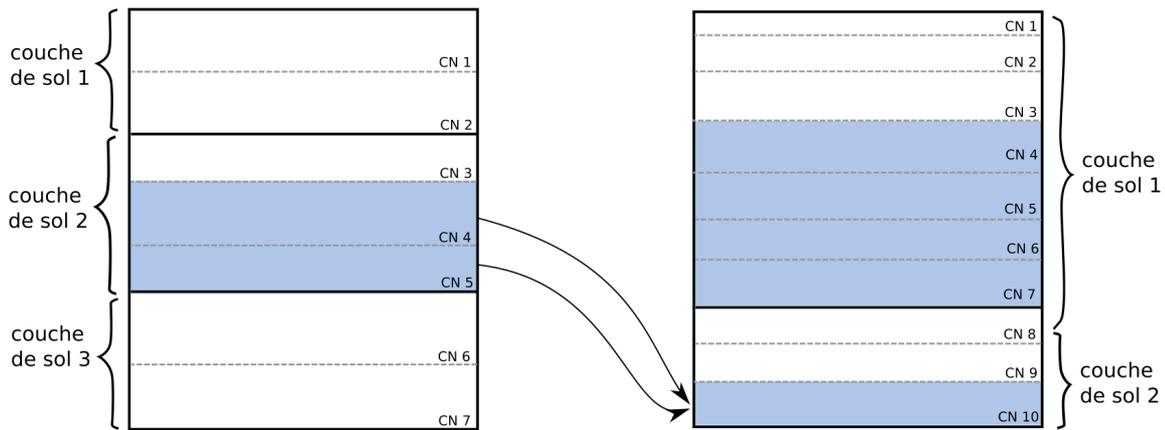
Obs : Chaque colonne de sol peut avoir une altitude différente en accord avec le MNT mais l'épaisseur totale des colonnes restera constante sur le bassin versant modélisé. Ainsi, on ne peut pas utiliser le fond des colonnes comme référence commune pour l'axe vertical. On choisit comme niveau de référence pour l'axe vertical le niveau à partir

duquel sont calculées les altitudes des centroïdes des unités homogènes, soit le niveau de la mer. Les niveaux de nappes et charges hydrauliques à l'intérieur de la colonne de sol seront donc calculés à partir de cette référence.



CN : Couche numérique

FIGURE 30 – Exemple d'une nappe perchée et de la répartition du flux sortant.



CN : Couche numérique

FIGURE 31 – Exemple de la répartition du flux sortant en cas de nappe perchée plus haute à l'aval.

### Détermination du flux latéral

Le flux latéral entre deux colonnes est déterminé à partir de la loi de Darcy :

$$Q_{x \rightarrow y} = K_{int} S \times gradH \quad (10.1)$$

où  $K_{int}$  est la conductivité hydraulique (horizontale) à l'interface [ $LT^{-1}$ ],  $S$  est la surface d'échange à travers la ligne d'échanges [ $L^2$ ] et  $gradH$  le gradient de charge entre A et B [-].

La conductivité hydraulique horizontale à l'interface est déterminée comme la moyenne harmonique entre les conductivités hydrauliques horizontales de chacune des colonnes (les distances  $d_x$  et  $d_y$  sont illustrées sur la fig 32).

$$\frac{d_x + d_y}{K_{int}} = \frac{d_x}{K_x} + \frac{d_y}{K_y} \quad (10.2)$$

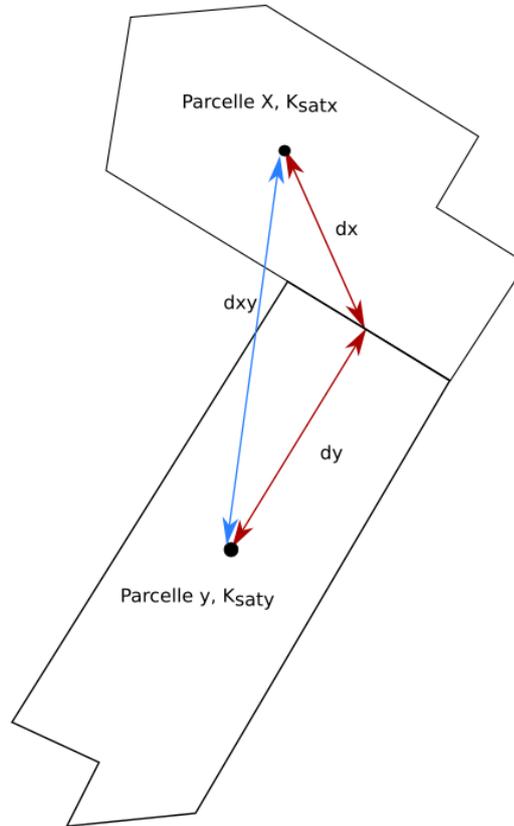


FIGURE 32 – Distances utilisées pour calculer la conductivité hydraulique à l'interface ( $d_x$  et  $d_y$ ) et le gradient de charge ( $d_{xy}$ ).

Dans le cas d'un profil de sol hétérogène, la conductivité hydraulique à saturation horizontale équivalente d'une zone saturée  $WT_n$  répartie sur  $m$  cellules numérique dans une colonne de sol est égale :

$$K_{WT_n} = \frac{\sum_{i=1}^m z_i \times k_i}{\sum_{i=1}^m z_i} \quad (10.3)$$

où  $z_i$  est la hauteur de la cellule  $i$  de sol saturé [L] et  $k_i$  la conductivité hydraulique horizontale de cette cellule [ $LT^{-1}$ ].

Si la colonne aval ne contient aucune nappe, la conductivité hydraulique à saturation pour cette colonne sera égale à celle de la cellule non saturée vers laquelle sera routé le flux latéral.

Dans de nombreux cas, la conductivité hydraulique horizontale n'est pas égale à la conductivité hydraulique verticale, on parle alors d'anisotropie. Cependant, de manière générale, on ne dispose pas des valeurs exactes des conductivités hydrauliques horizontales et verticales pour chacun des horizons de sol mais plutôt d'une mesure de la conductivité hydraulique verticale et d'une hypothèse sur le ratio  $\frac{Ksat_{xy}}{Ksat_z}$  avec  $Ksat_{xy}$ , conductivité hydraulique horizontale [LT<sup>-1</sup>] et  $Ksat_z$ , conductivité hydraulique verticale [LT<sup>-1</sup>].

Pour la suite, on notera simplement  $K = K_{xy}$  pour la conductivité hydraulique horizontale.

### Test sur la teneur en eau de la colonne

Si la méthodologie présentée fonctionne bien pour 2 colonnes de sol isolées, elle doit être adaptée au cas où une parcelle sera reliée en aval à plusieurs parcelles.

En effet, dans ce cas, le flux latéral total sortant d'une colonne amont est calculé comme la somme algébrique des flux latéraux sortant vers chaque colonne aval. Cependant, ces contributions sont calculées "par couple", c'est-à-dire sans prendre en compte le fait qu'une colonne peut avoir plusieurs voisins. Ainsi, dans le cas d'une colonne amont reliée à plusieurs colonnes aval, pour chacune de ces voisines aval, on calcule le flux de Darcy à partir de la même hauteur de nappe de la colonne amont (celle au début du pas de temps). Rien n'assure donc que lorsqu'on calcule un flux sortant vers la dernière voisine aval, la colonne amont n'est pas déjà été vidée par les contributions partielles déjà calculées.

Il faut donc une étape de test une fois le flux latéral total sortant de la colonne amont calculé : si le flux latéral total sortant  $Q_{lat}$  multiplié par le pas de temps  $dt$  est supérieur à la quantité d'eau dans la zone saturée  $W_{sat}$  de la colonne considérée, ce flux latéral est réévalué à :

$$Q_{lat} = \frac{W_{sat}}{dt}$$

et redistribué entre les colonnes aval au prorata de la longueur d'interface. Ceci implique donc de devoir fournir pour chaque colonne, la longueur totale de la ligne d'échanges qu'elle possède avec ses voisines situées en aval.

### Répartition du flux entre couches de calcul

Pour retirer effectivement le flux  $Q$  de la colonne amont et l'affecter à la colonne aval, il existe plusieurs manières d'inclure ce flux dans l'équation du bilan de masse des cellules des colonnes amont et aval. S'il existe également une zone saturée dans la colonne aval, on peut :

- Appliquer le flux latéral au fond de la zone saturée, c'est-à-dire comme un terme source pour une seule cellule ;
- Appliquer le flux latéral au dessus de la zone saturée, c'est-à-dire comme terme source pour la première cellule non saturée au dessus de la nappe ;
- Répartir le flux latéral entre les différentes cellules saturées.

On choisit de répartir ce flux entrant/sortant entre les différentes cellules numériques saturées des colonnes amont et aval. Ainsi, on espère avoir moins d'eau ajoutée ou retirée par cellule de calcul ce qui peut diminuer le risque d'instabilités.

1. On va essayer d'effectuer la répartition en prenant en compte des « conductivités hydrauliques relatives » de chaque cellule. De cette manière, on pourra retranscrire des comportements du type « une cellule faisant partie d'une couche plus imperméable recevra moins d'eau qu'une cellule d'une couche plus perméable ». Pour cela, à chaque cellule de calcul, on associe un poids correspondant à une conductivité relative. Pour la cellule numérique  $i$  de la nappe  $n$ , on a :

$$weight_n(i) = \frac{\frac{1}{K_i}}{\frac{1}{K_{eq}(n)}} = \frac{\frac{z_i}{K_i}}{\sum \frac{z_i}{K_i}}. \quad (10.4)$$

De sorte que la somme des poids soit égale à 1 pour une nappe donnée.

2. Cette ligne d'échanges devant également assurer le transfert de solutés d'une colonne à l'autre, il est judicieux de réfléchir à une manière de conserver le gradient de concentration qui peut apparaître au sein de la colonne. On peut donc imaginer une méthode de répartition qui ne répartisse pas le flux global  $weight_x(i)*Q$  qui sort de la cellule  $i$  de la colonne  $x$  entre toutes les cellules de la zone saturée de  $y$  en fonction des poids de chacun mais plutôt qui privilégie les cellules les plus proches géographiquement.

Le flux  $weight_x(1)*Q$  qui sortira de la première cellule de  $x$  ira de préférence vers la première cellule de la zone saturée de  $y$ . Si on atteint la capacité maximale  $weight_y(1)*Q$ , le flux restant sera attribué à la seconde cellule. Si au contraire, tout le flux  $weight_x(1)*Q$  sortant de la cellule 1 de  $x$  ne suffit pas à atteindre la capacité maximale de la cellule 1 de  $y$ , on affecte le flux  $weight_x(2)*Q$  sortant de la cellule de  $x$  à la couche 1 de  $y$  (fig. 33).

En appliquant cette même répartition pour les concentrations de solutés, on peut respecter les gradients de concentration qui peuvent exister au sein de la zone saturée (important surtout dans le cas de nappes perchées). Par contre, on ne représente ainsi que la convection et pas la diffusion ni la dispersion. Des tests plus poussés devront être menés pour savoir si cette simplification est suffisante ou non.

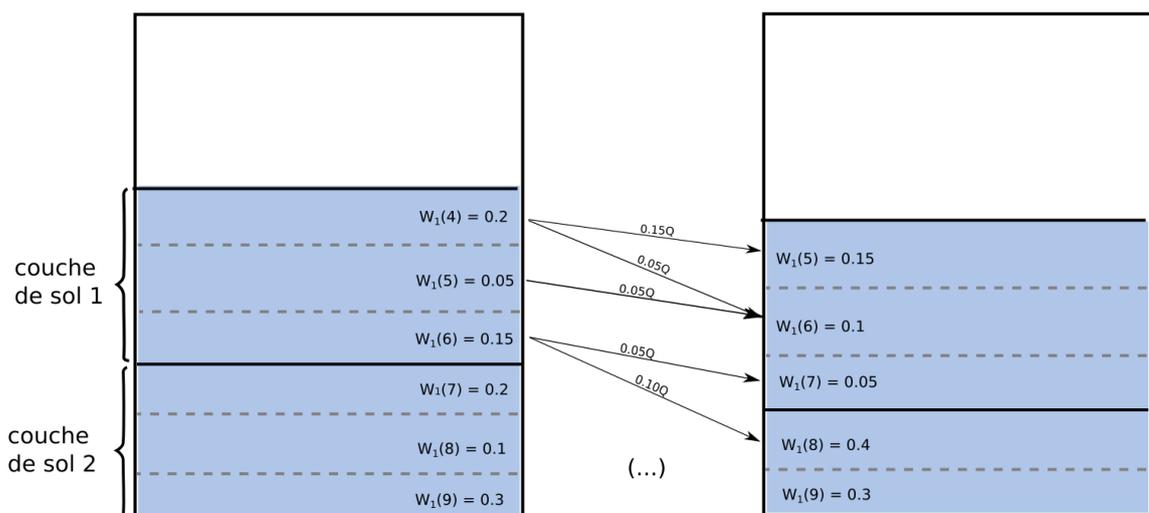


FIGURE 33 – Exemple de répartition du flux de Darcy  $Q$  entre 2 colonnes avec le système de poids.

### 10.2.3 Implémentation de la ligne d'échanges Colonne-Colonne

La ligne d'échanges est codée sous la forme d'une classe `ColumnColumnExchangeLine`. Celle-ci hérite de la classe `SubsurfaceInterface` et s'initialise entre autre à partir de deux instances de la classe `Column`.

#### Classe `Column`

La classe `Column` permet d'initialiser une colonne de sol, ses coordonnées, caractéristiques géométriques, propriétés hydrologiques et concentration en solutés dissous avant d'identifier les nappes éventuellement présentes dans la colonne ainsi que les conductivités hydrauliques associées. Elle hérite de la classe `HomogeneousUnit` qui elle-même hérite de la classe `Element`.

Par défaut, on suppose être en situation d'isotropie et la valeur donnée pour la conductivité hydraulique à saturation est valable pour les composantes verticales et horizontales. Pour intégrer de l'anisotropie, on peut donner une valeur au paramètre optionnel "ratio"  $= \frac{K_{sat_{xy}}}{K_{sat_z}}$ . Dans ce cas, le `Ksat` donnée en paramètre d'entrée obligatoire correspond à la conductivité hydraulique à saturation verticale, utilisée par `FRER1D` et la conductivité hydraulique horizontale est calculée grâce au paramètre "ratio" [-].

La classe `Column` est composée des méthodes suivantes :

- **constructeur `__init__()`**

La classe `Column` est initialisée à partir des attributs suivants :

- `idx` (double) : index de la colonne
- `x,y` (double) : latitude et longitude de la colonne [L]
- `z` (double) : altitude de la surface de la colonne [L]
- `area` (double) : surface de la colonne [L<sup>2</sup>]
- `soil_profile` : une instance de classe `SoilProfile` décrivant le type de sol associé à la colonne
- `Lint` (double) : longueur d'échange entre la colonne et des colonnes aval [L]
- `head` (double(1 :n)) : vecteur contenant la pression capillaire dans chaque cellule numérique [L]
- `wat` (double(1 :n)) : vecteur contenant la quantité d'eau contenue dans chaque cellule numérique [L<sup>3</sup>]
- `cw` (double(1 :ns,1 :n)) : vecteur contenant la concentration en solutés dissous dans chaque cellule numérique et pour chaque espèce [M.L<sup>-3</sup>]
- `soiltype` : dictionnaire python décrivant la position de chaque horizon de sol dans la colonne. Pour chaque type de sol `j`, on crée deux entrées dans le dictionnaire : `{j,nb_cel}` contenant le nombre de cellules numériques étant de ce type de sol et `{j,interface}` donnant la position de l'interface avec l'horizon de sol situé en dessous (plus exactement l'index de la cellule située au dessus de cette interface)

On rappelle que le vecteur `dx`, attribut de la classe `SoilProfile` fournit la profondeur de chaque cellule numérique par rapport à la **surface**. Cependant, les altitudes et

les fonds des colonnes peuvent être différents d'une colonne à l'autre et ils sont calculés par rapport à un niveau de référence (niveau de la mer). Il faut faire attention à ce point pour le calcul des altitudes de chaque cellule numérique par rapport au niveau de la mer.

- **get\_h\_table()**

Permet de repérer les zones saturées éventuellement présentes dans une colonne de sol en identifiant les cellules pour lesquelles la condition suivante est vérifiée :

$$h_i - h_e > 0 \quad (10.5)$$

où  $h_i$  est la pression dans la cellule  $i$  [L] et  $h_e$ , la pression d'entrée de l'air dans cette même cellule [L]. La nappe la plus près de la surface est identifiée comme numéro 1.

Pour chacune de ces nappes identifiées, un dictionnaire `water_table` est créé, contenant ses caractéristiques géométriques. Par exemple pour la nappe 1, on a :

```
self.water_table={ (1,bottom) :200.0, (1,sat_cells) :[8,9], (1,thick) :80.0, (1,height) :80.0
}
```

où 'bottom' désigne la position du bas de la nappe par rapport au niveau de référence commun, 'sat\_cells' donne la liste des indices des cellules saturées pour cette nappe, 'thick' donne l'épaisseur de la nappe et 'height', la hauteur de charge (toujours par rapport au niveau de référence commun). L'indice  $n$  dans les couples ( $n,xxx$ ) fait référence au numéro de la nappe dont sont énumérées les caractéristiques.

L'attribut `nb_table` (integer) est également créé et contient le nombre de nappes que contient la colonne de sol.

⚠ Comme précisé dans le cahier des charges, toute zone saturée ne sera pas considérée comme une nappe qui donnera lieu à un transfert latéral. Une zone saturée sera comptabilisée dans le dictionnaire `water_table_dic` seulement si sa cellule saturée la plus basse correspond à une interface entre deux horizons (ainsi, s'il ne s'agit pas d'une nappe de fond de profil, on s'assure que c'est réellement une nappe perchée et pas une nappe fugace).

De plus, pour plus de facilité, on a décidé dans un second temps de raisonner par horizon de sol. Cela signifie qu'une zone saturée s'étendant sur  $n$  horizons de sol sera séparée en  $n$  nappes différentes. Le test précédent sera appliqué sur chacune de ces nappes pour vérifier qu'elles puissent bien être intégrées au dictionnaire.

Dans le code, les indices des cellules numériques correspondant à une zone saturée sont stockés dans la variable `group` puis séparés selon les horizons de sol dans la variable `group_split`.

- **get\_Ksat\_eq()**

Méthode qui enrichit le dictionnaire `water_table` de la conductivité hydraulique (latérale) à saturation équivalente 'Keq' pour chaque zone saturée. Celle-ci est

calculée comme indiqué par la formule 10.3 :

$$K_{int} = \frac{\sum z_i \times k_i}{\sum z_i} \quad (10.6)$$

où  $z_i$  est la longueur de la couche  $i$  de sol saturé [L] et  $k_i$  la conductivité hydraulique horizontale de cette couche de sol [LT<sup>-1</sup>].

Dans le cas d'anisotropie, les  $k_i$  font référence aux conductivités hydrauliques horizontales, calculée à partir des données d'entrée de conductivités hydrauliques verticales et du paramètre "ratio" comme suit :

$$K_{int} = \frac{\sum z_i \times k_{z_i} \times ratio}{\sum z_i} \quad (10.7)$$

- **get\_weight()** :

Méthode qui permet de créer l'attribut `weight` (`double(1 :n)`) contenant les pondérations nécessaires au calcul de flux. Ces pondérations sont égales à 1 dans le cas de cellules non saturées (comme présenté plus haut, l'intégralité du flux provenant d'une nappe située en amont sera routé vers une unique cellule non saturée en aval s'il n'existe pas déjà de nappe). Dans le cas de cellules saturées, la pondération est calculée comme indiqué par la formule 10.4.

### Classe `ColumnColumnExchangeLine`

- **constructeur `__init__()`** :

Une instance de la classe `ColumnColumnExchangeLine` est initialisée avec les mêmes attributs que la classe `SubsurfaceInterface` :

- `idx` : un index (scalaire entier) pour repérer la ligne d'échanges ;
- `length` : une longueur [L] ;
- `x,y` : la longitude et la latitude du milieu de la ligne d'échanges[L] ;
- `z` : une altitude [L] ;
- `UpstreamInstance` : une instance de la classe `Column`. Il s'agit de l'élément situé à l'amont de la ligne d'échanges.
- `DownstreamInstance` : une instance de la classe `Column`. Il s'agit de l'élément situé à l'aval de la ligne d'échanges.

En plus d'appeler le constructeur de la classe `SubsurfaceInterface`, les distances entre le centroïde des colonnes amont (et aval) et le centroïde de la ligne d'échanges sont initialisées. C'est l'attribut `dist_up` (resp. `dist_down`). La distance entre les centroïdes des éléments amont et aval est également initialisée comme l'attribut `dist_updown`.

- **update\_state\_variables()** :

Cette méthode permet la mise à jour des variables d'état des colonnes amont et aval au début de chaque pas de temps. Pour cela, elle récupère les variables PALM contenant le vecteur de pression capillaire, de volume d'eau et de concentration en solutés dans chaque cellule numérique et met à jour les attributs de la classe `Column` correspondants. La méthode `get_h_table` de classe `Column` est également appelée pour finaliser cette mise à jour.

- **calculate\_Kint(K\_up,K\_down)** :  
Méthode qui calcule la conductivité hydraulique à l'interface selon la formulation 10.2. Les distances `dist_up` et `dist_down` entre chaque centroïde et le centre de la ligne d'échanges ont été initialisées par la méthode `_init_()` et ne sont pas données comme argument d'entrée. Par contre, cette méthode nécessite de donner en entrée les conductivités hydrauliques des colonnes amont et aval. Ceci permet l'appel successif de la méthode avec plusieurs conductivités équivalentes relatives aux mêmes colonnes mais pour des nappes différentes.

- **calculate\_Hint(H\_up,H\_down)** :  
Cette méthode permet le calcul de la hauteur mouillée à l'interface, laquelle entre dans le calcul de  $Q_{x \rightarrow y}$  (voir équation 10.1). En effet, la surface  $S$  est définie comme le produit de la longueur  $L$  de la ligne d'échanges  $[L]$  et de la hauteur mouillée  $H_{int}$   $[L]$ .  
Pour le calcul de  $H_{int}$ , on reprend la méthode adoptée par Branger (2007b) qui calcule  $H_{int}$  à chaque pas de temps par interpolation linéaire entre les niveaux de nappe de la colonne amont et aval :

$$H_{int} = \frac{H_{down} - H_{up}}{d_{up} + d_{down}} \times d_{up} + H_{down} \quad (10.8)$$

où  $H_{up}$ ,  $H_{down}$  sont les charges hydrauliques dans les colonnes amont et aval  $[L]$  et  $d_{up}$ ,  $d_{down}$ , les distances des centroïdes des colonnes au centre de la ligne d'échanges  $[L]$ .

La formule 10.8 est valable telle quelle dans le cas d'une nappe en fond de profil mais doit être adaptée au cas des nappes perchées. Dans ce cas (nappes perchées dans les 2 colonnes), on applique la formule 10.8 pour trouver par interpolation les limites inférieures et supérieures de la hauteur mouillée à l'interface. La hauteur mouillée est ensuite calculée par différence.

En l'absence de nappe dans la seconde colonne, la hauteur mouillée à l'interface est calculée comme précédemment en prenant des limites inférieures et supérieures toutes deux égales à la hauteur du fond de la cellule non saturée qui recevra le flux latéral dans la colonne aval. De cette manière, la hauteur mouillée résultante est nulle.

Comme précédemment, on donne en argument à cette fonction les niveaux de nappe dans chaque colonne permettant ainsi le calcul de plusieurs hauteurs mouillées à l'interface pour une colonne en fonction des nappes considérées.

- **calculate\_flux\_interface()** :  
Cette méthode identifie les flux possibles entre les deux colonnes. Après calcul, les différents flux sont sommés dans 2 vecteurs `UpstreamInstance_lat_flux` et `DownstreamInstance_lat_flux`, attributs des instances `Column`, contenant la quantité d'eau algébrique totale qui sera retirée ou ajoutée à chaque couche numérique. De la même manière, les flux de solutés advectés avec l'eau sont calculés suivant le système de pondération détaillé partie 10.2.2 en sommant algébriquement les différentes contributions dans les vecteurs `UpstreamInstance_sol_flux` et `DownstreamInstance_sol_flux`.

Les vecteurs `column_up.lat_flux` et `column_down.lat_flux` sont d'abord initialisés comme un vecteur nul, de taille égale au nombre de couches numériques de chaque colonne. Puis, pour chaque nappe de la colonne amont, la méthode recherche la cellule située au fond de l'horizon de sol le plus proche (mais qui est plus profond) de la colonne aval. L'indice de cette cellule est stocké dans la variable `idx_cell_tofill`. Si cet indice est présent dans le vecteur contenant tous les indices des cellules saturées de la colonne aval, on recherche la nappe à laquelle il appartient. Sinon la méthode `handle_water_table` est appelée avec comme arguments pour la colonne aval, la cellule la plus basse de l'horizon le plus proche, une charge nulle et des caractéristiques  $K_{int}$  et  $H_{int}$  calculées en considérant la hauteur de nappe nulle dans la colonne.

Dans le cas où `idx_cell_tofill` fait bien partie d'une nappe, après que cette dernière ait été repérée, un test est fait pour vérifier que le gradient hydraulique est bien compatible avec la pente et qu'il existe une différence de hauteur significative entre les charges hydrauliques des 2 colonnes (pour le moment, le delta est fixé à 1 cm, cette valeur pourra être ajustée par la suite (comme constante PALM de la simulation par exemple) avant de lancer le calcul effectif du flux. Si les conditions sont réunies, la méthode `handle_water_table` est appelée avec comme arguments pour la colonne aval, les caractéristiques de la nappe repérée.

Un traitement particulier est réservé au cas d'une nappe perchée en amont qui doit se déverser dans une nappe perchée en aval mais pour lesquelles le gradient hydraulique n'est pas compatible (cas illustré par la figure 31). Dans ce cas-là, si l'intégralité du profil en dessous de la nappe perchée aval n'est pas saturé, on réinitialise la variable `idx_cell_tofill` avec la valeur de l'interface entre horizons de sol suivant rencontrée en profondeur et on relance la démarche précédente (le tout étant contenu dans une boucle `while`) : `idx_cell_tofill` est-il saturé? Si oui, on repère sa nappe, si non on relance le calcul du flux avec cette cellule non saturée, etc... L'algorithme est résumé ci-dessous :

Initialiser les vecteurs de flux latéraux pour les 2 colonnes

Pour chaque nappe de la colonne amont FAIRE:

Rechercher la cellule aval située à l'interface entre 2 horizons ET  
la plus proche de la dernière cellule de la nappe amont  
Mettre l'indice de cette cellule dans `idx_cell_tofill`

`iter=1`

TANT QUE `iter<2`:

SI `cellule_aval[idx_cell_tofill]` est saturée :

Trouver la nappe à laquelle appartient `cellule_aval[idx_cell_tofill]`

SI (la charge de la nappe aval est inférieure à celle de la nappe amont

ET que le gradient de charge est supérieur au seuil):

Lancer `calculate_flux` entre les 2 nappes  
`Iter = Iter + 1`

SINON-SI (la charge de la nappe aval identifiée est supérieure à celle de la nappe amont  
 ET la nappe amont est une nappe perchée  
 ET le fond de la nappe aval repérée n'arrive pas au fond du profil):

Trouver dans la colonne aval l'interface entre 2 horizons la plus proche  
 Mettre l'indice de cette cellule dans `idx_cell_tofill`

SINON :  
 Ne pas calculer de flux  
`Iter = Iter + 1`

SI `idx_cell_tofill` n'est pas saturée:

Lancer `calculate_flux` avec la cellule `cellule_aval[idx_cell_tofill]`  
`Iter = Iter + 1`

- **`handle_water_table(Ki,Hi,H_up,H_down,cells_up,cells_down)`** :  
 Méthode qui calcule un flux entre 2 zones saturées. La méthode prend les arguments suivants :
  - `Ki` : Conductivité hydraulique à l'interface ;
  - `Hi` : Hauteur mouillée à l'interface,
  - `H_up` : Charge dans la colonne amont,
  - `H_down` : Charge dans la colonne aval,
  - `cells_up` : index de la (des) cellule(s) contenant la zone saturée dans la colonne amont,
  - `cells_down` : index de la (des) cellule(s) contenant la zone saturée dans la colonne aval,
  - `L_int` : longueur de la ligne d'échanges,
  - `L_int_down` : somme des longueurs des lignes d'échanges entre la colonne amont et les colonnes aval.

### *Calcul du flux*

Dans un premier temps, le flux est calculé comme indiqué par la formule 10.1. Il est nécessaire à ce niveau d'effectuer un test sur la teneur en eau de la zone saturée pour vérifier que l'on ne vide pas trop la colonne. Dans le cas d'une parcelle ayant une unique voisine en aval, on peut appliquer le raisonnement décrit section

10.2.2. Cependant, dans le cas d'une parcelle  $P_{up}$  reliée à plusieurs parcelles aval  $P_{down_i}$ , chaque  $P_{down_i}$  n'a pas (et ne doit pas avoir) accès aux propriétés des autres voisines aval de  $P_{up}$ . Elle ne saura donc pas si le flux total sortant  $Q_{lat}$  (et donc la contribution vers  $P_{down_i}$ ) a du être réévalué ou pas. Pour contourner le problème, on choisit la formulation suivante pour calculer le flux associé à chaque nappe de la colonne amont :

$$q_{lat} = \min \left( q_{Darcy}, \frac{\sum_{i=1}^m water\_cell_i \times L\_int_{P_{up}-P_{down}}}{dt * L\_int\_down} \right) \quad (10.9)$$

pour une ligne d'échanges de longueur  $L\_int$  entre la colonne amont  $P_{up}$  contenant une nappe s'étalant sur  $m$  cellules numériques et la colonne aval  $P_{down}$ .  $L\_int\_down$  est la longueur totale d'interface entre  $P_{up}$  et l'ensemble des colonnes aval avec lesquelles  $P_{up}$  partage une ligne d'échanges.

### Répartition entre couches numériques

Dans un second temps, la répartition détaillée dans la section 10.2.2 est appliquée en parcourant chaque élément de  $cells\_up$ , en récupérant son poids  $weight_i$  et en regardant combien de cellules de la colonne aval peuvent être remplies grâce à lui, en commençant par celle du haut. Chaque contribution est alors incrémentée aux vecteurs  $lat\_flux$  comme flux entrant pour la colonne aval et flux sortant pour la colonne amont.

En parallèle, les vecteurs  $sol\_flux$  sont incrémentés avec les solutés advectés non plus à partir des flux d'eau  $Q$  mais avec les flux de solutés  $Q * c_{wi}$  ( $g.h^{-1}$ ) où  $c_{wi}$  est la concentration en solutés dissous dans la cellule numérique  $i$  ( $g.cm^{-3}$ ). Les figures 34 et 35 montrent un exemple de répartition de flux latéral, la première pour l'eau seulement et la seconde avec des solutés dans la colonne amont.

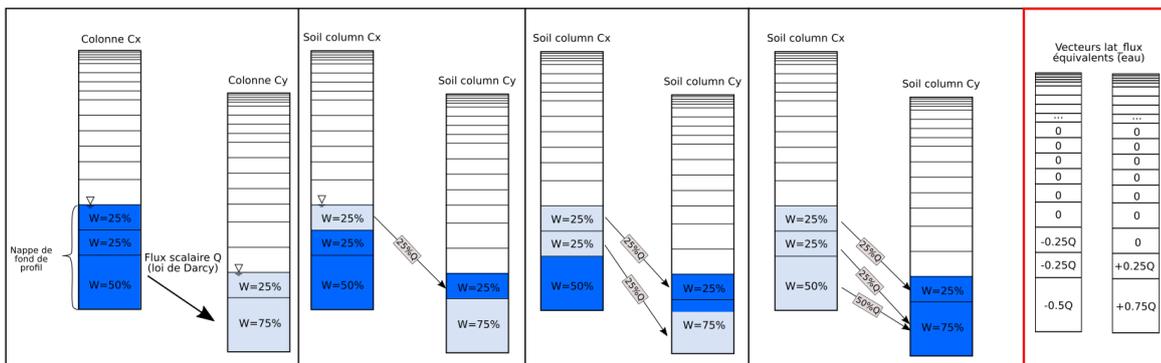


FIGURE 34 – Exemple de répartition d'un flux d'eau  $Q$  scalaire entre une colonne amont et une colonne aval. Les coefficients  $W$  sont les poids de répartition  $weight$ .

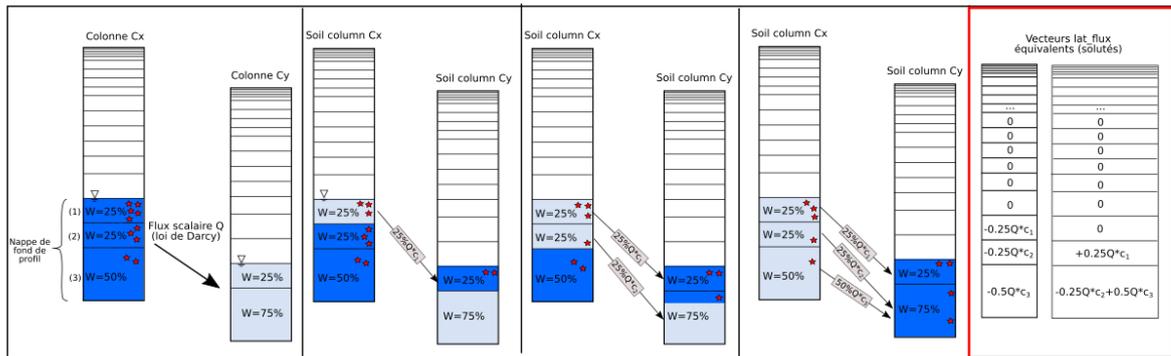


FIGURE 35 – Exemple de calcul d’échanges latéraux de solutés entre une colonne amont et une colonne aval. Les coefficients  $W$  sont les poids de répartition *weight*, les indices (1), (2) et (3) repèrent les cellules saturées de la colonne amont et les étoiles rouges représentent schématiquement la masse de solutés disponible dans chaque cellule numérique.

### 10.3 Transferts entre deux parcelles séparées par un fossés

#### 10.3.1 Introduction : fonctionnement hydrologique d’un fossé

Les fossés peuvent avoir une fonction de court-circuit ou au contraire constituer des éléments de pondération de la pollution favorisant la rétention des polluants, leur dégradation éventuelle et ainsi limiter la contamination des ressources en eau (Margoum et al., 2001). Le fonctionnement des fossés agricoles diffère selon la pente du terrain, et surtout le niveau de nappe par rapport au niveau de l’eau dans le fossé. Ainsi, un fossé peut avoir un rôle drainant ou infiltrant. Dans le cas d’un fossé en travers de la pente, la collecte du ruissellement et des écoulements subsurface domine alors que pour un fossé parallèle à la ligne de plus grande pente, c’est le rabattement latéral de la nappe et le transfert vers l’aval qui l’emportent. L’interception des écoulements subsurface et souterrains (nappe) est fonction des profondeurs relatives du fond du fossé et du toit de la nappe et de l’existence et du développement des écoulements subsurface (Adamiade, 2004). Les différents fonctionnements d’un fossé sont résumés fig. 36.

Dans un premier temps, on ne représentera que des fossés qui interceptent les écoulements de surface et une partie de la subsurface provenant de l’amont et transfèrent le reste (cas d de la fig.36). Pour ce mode de fonctionnement, une partie de l’écoulement et potentiellement les solutés qu’il transporte sont interceptés et peuvent être transférés très rapidement le long du réseau de fossés. En plus des flux provenant des parcelles ou des tronçons amont, les fossés sont également soumis aux forçages climatiques (pour le moment, seulement de la pluie, pas d’évaporation).

#### 10.3.2 Représentation

On rappelle qu’un fossé est représenté par un réservoir unique, à section trapézoïdale (Figure 5) et qu’il possède les attributs suivants :

- Longueur  $L$  (cm) ;

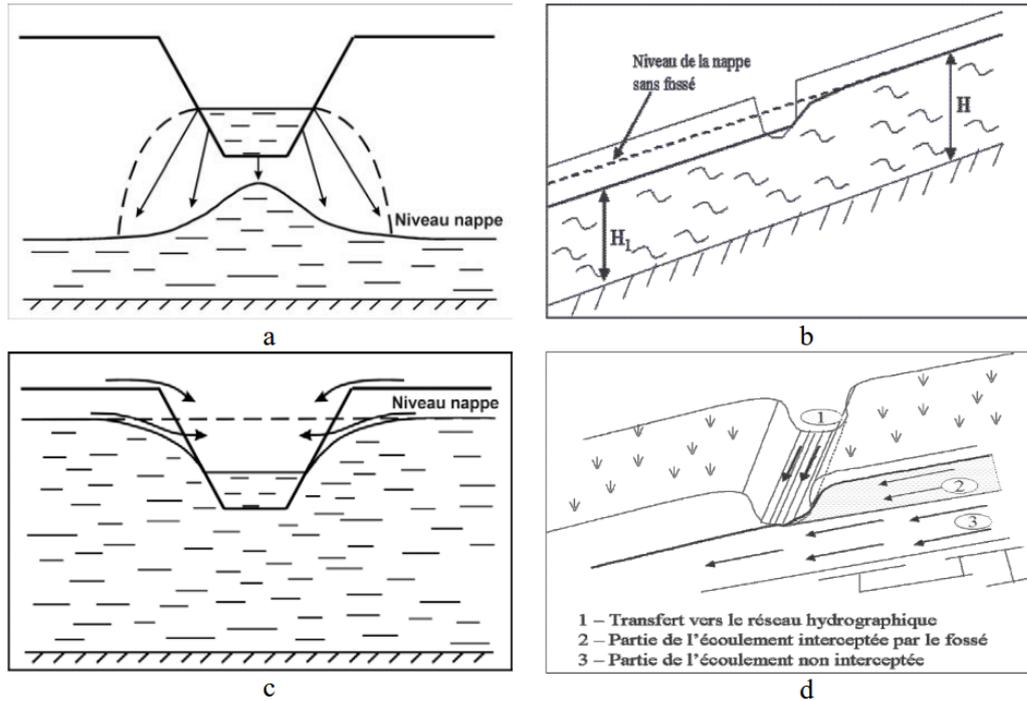


FIGURE 36 – Fonctionnement hydrologique d'un fossé. a : fossé qui infiltre, b : fossé qui intercepte, c : fossé qui draine, d : fossé qui intercepte et qui transfère. Tiré de Adamiade (2004)

- Largeur de fond  $l$  (cm),
- Profondeur (cm),
- Fruit des berges  $\alpha$  (deg),
- Pente moyenne longitudinale (deg)
- Substrat.

La variable d'état choisie pour représenter l'évolution de l'état du fossé est la hauteur d'eau, calculée à partir du volume d'eau  $V$  contenu dans le réservoir à chaque pas de temps :

$$h_w = \frac{\left(\frac{-l}{\tan\alpha} + \sqrt{\left(\frac{l}{\tan\alpha}\right)^2 + \frac{4V}{\tan\alpha L}}\right)}{2} \quad (10.10)$$

Dans le cas de section trapézoïdale, le rayon hydraulique peut-être estimé simplement à partir de la surface de la section ( $S$ ) et du périmètre mouillé ( $P$ ) :

$$d = \frac{S}{P} = \frac{(l + \tan(\alpha)h_w)h_w}{l + 2h_w\sqrt{1 + \tan(\alpha)^2}} \quad (10.11)$$

On autorise également une hauteur de ponding  $H_{\text{pond}}$  paramétrable, dans le fond du fossé, afin de ralentir les transferts de solutés parfois trop rapides. Lorsqu'une hauteur de ponding non nulle est spécifiée, seul le volume d'eau situé au dessus de cette hauteur seuil peut se déplacer dans le réseau de fossés et le volume restant reste immobile.

### 10.3.3 Implémentation de la ligne d'échanges entre 2 colonnes séparées par un fossé

La ligne d'échanges est codée sous la forme de la classe `ColumnDitchColumnExchangeLine`. Celle-ci hérite de la classe `SubsurfaceInterface` et s'initialise entre autre à partir de deux instances de la classe `Column` et d'une instance de la classe `Ditch`. La classe `Column` est décrite section 10.2.3 et les autres classes sont décrites ci-après :

#### Classe `Ditch`

La classe `Ditch` initialise une instance de fossé telle qu'on en a besoin dans la ligne d'échanges `ColumnDitchColumnExchangeLine`. Cette initialisation se fait à partir d'une instance de la classe `Reach` (voir description dans la section 9.4.2). En plus d'un constructeur, la classe `Ditch` contient des méthodes d'initialisation du flux latéral algébrique entrant/sortant de l'élément (un scalaire pour le flux d'eau, un vecteur(0 :n\_solute) pour les solutés). La classe `Ditch` contient aussi les méthodes `update_waterlatflux` et `update_solutelatflux`. Contrairement à des setters classiques qui remplacent des variables existantes par une nouvelle valeur, ces méthodes permettent de mettre à jour la variable `waterlatflux` (resp. `solutelatflux`) en incrémentant la variable existante avec un stock additionnel `increment_waterflux` (resp. `increment_solutelatflux`).

#### Classe `ColumnDitchColumnExchangeLine`

- `__init__()`

Cette méthode permet d'initialiser une instance de la classe `ColumnDitchColumnExchangeLine` avec les mêmes attributs que la classe `SubsurfaceInterface` :

- `idx` : un index (scalaire entier) pour repérer l'interface ;
- `length` : une longueur [L] ;
- `x,y` : une longitude et une latitude [L] ;
- `z` : une altitude [L] ;
- `UpstreamInstance` : une instance de la classe `Column`. Il s'agit de l'élément situé à l'amont de l'interface.
- `DownstreamInstance` : une instance de la classe `Column`. Il s'agit de l'élément situé à l'aval de l'interface.
- `InterceptInstance` : une instance de la classe `Ditch` qui interceptera une partie des transferts entre les colonnes amont et aval.

- `update_state_variables()`

Cette méthode permet la mise à jour des variables d'état des colonnes amont et aval au début de chaque pas de temps. Pour cela, elle récupère les variables PALM contenant le vecteur de pression capillaire, de volume d'eau et de concentration en solutés dans chaque cellule numérique et met à jour les attributs de la classe `Column` correspondants. La méthode `get_h_table` de la classe `Column` est également appelée pour finaliser cette mise à jour. La concentration en solutés est enfin initialisée pour l'instance de fossé interceptant une partie des transferts.

- **split\_watertable(ColumnInstance,limit,limit\_idx)**

Split\_watertable est une fonction qui prend en entrée les variables suivantes :

- ColumnInstance : une instance de la classe Column ;
- limit : une profondeur limite (scalar) ;
- limit\_idx : l'index de la cellule au fond de laquelle la profondeur limite *limit* est atteinte (integer).

Cette fonction permet de modifier le dictionnaire water\_table\_dic pour qu'il ne contienne que les nappes situées en dessous de la hauteur *limit*. En réalité, les autres nappes sont conservées dans le dictionnaire mais tous leurs attributs sont fixés à **None**. Ces nappes ainsi que leurs attributs sont par ailleurs stockés dans le dictionnaire wt\_2linear\_dic.

Les sorties de la fonction sont composées des dictionnaires water\_table\_dic et wt\_2linear\_dic et du vecteur sat\_cl\_above contenant les index des cellules situées au dessus de la limite *limit*.

- **calculate\_flux\_interface()**

La méthode calculate\_flux\_interface gère le calcul du flux latéral dans le cas d'une ligne d'échanges Colonne-**Fossé**-Colonne.

Dans un premier temps, on suppose qu'un fossé se comporte toujours comme un fossé qui intercepte et transfère (d de la figure 36). La ligne d'échanges correspondante fonctionne de la manière suivante :

- Le flux latéral (eau et solutés) sortant de la colonne amont (UpstreamInstance) est d'abord calculé comme s'il n'existait pas de linéaire mais simplement une colonne aval (DownstreamInstance). A partir des instances de la classe Column UpstreamInstance et DownstreamInstance, on peut initialiser une instance de la classe ColumnColumnExchangeLine.
- Si un fossé se trouve entre les deux colonnes, les écoulements latéraux provenant de cellules numériques situées au dessus du fond du fossé seront envoyés vers le fossé (écoulements interceptés). Ceux provenant de cellules numériques situées au dessous du fond du fossé seront transmises à la colonne aval comme si le fossé n'existait pas (cf fig. 37).

⚠ Il est possible de se trouver dans le cas où le fond du fossé ne correspond pas exactement à la limite entre 2 cellules numériques de la colonne de sol amont. Dans ce cas-là, la profondeur du fossé est remplacée par la profondeur de la cellule numérique la plus proche (variable **dx\_lin12**) pour éviter de devoir répartir l'eau d'une cellule numérique entre linéaire et colonne aval.

Comme détaillé dans la partie relative à l'interface ColumnColumnExchangeLine, pour calculer les flux d'eau sortant de la colonne amont vers la colonne aval, la méthode ColumnColumnExchangeLine.calculate\_flux\_interface explore les nappes d'eau contenues dans le dictionnaire water\_table\_dic afin de déterminer si elles seront à l'origine d'un flux et quelles cellules numériques de la colonne aval seront concernées.

Pour effectuer ces deux premières étapes et être capable de distinguer l'eau qui sera transmise au fossé de celle transmise à la colonne aval, on se base sur le

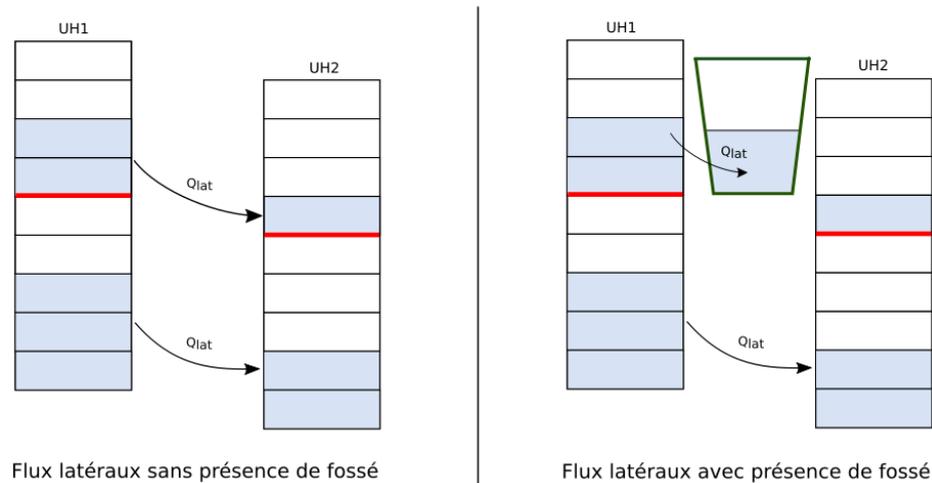


FIGURE 37 – Répartition des flux latéraux sortant de la colonne amont avec et sans fossé.

dictionnaire `water_table_dic` : on déplace toutes les nappes dont le fond est situé au dessus du fond du fossé dans un second dictionnaire `wt_2linear_dic`. Dans le cas d'une nappe située à cheval sur un fossé (fond de la nappe inférieur au fond du fossé mais fond du fossé inférieur au haut de la nappe), elle sera coupée en 2 pour apparaître dans les 2 dictionnaires, comme illustré fig. 38. Ce découpage des nappes est assuré par la fonction `split_watertable`.

Dans un premier temps, on ne découpait que les nappes de la colonne amont pour différencier l'eau qui allait entrer dans le fossé de celle qui allait entrer dans la colonne aval. Par la suite, on s'est rendu compte qu'il fallait également découper les nappes de la colonne aval afin de ne garder que celles situées plus bas que le fossé s'assurant ainsi que l'eau de la nappe amont ne "remontera" pas dans la colonne aval.

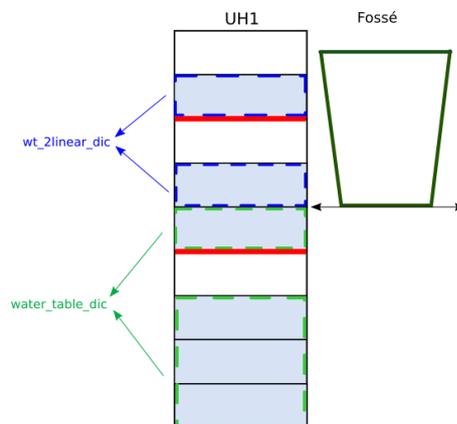


FIGURE 38 – Répartition des nappes dans les dictionnaires `water_table_dic` et `wt_2linear` en fonction de leur position par rapport au fossé.

⚠ Attention, dans le code, lorsque l'on manipule le dictionnaire `water_table_dic`. Lorsqu'une nappe est "supprimée" du dictionnaire car le flux sortant sera dirigé vers un linéaire et non pas vers la colonne aval, les entrées du dictionnaire correspondantes (`water_table_dic[indice,bottom]`, `water_table_dic[indice,sat_cells]`, `water_table_dic[indice,thick]`, `water_table_dic[indice,height]` et `water_table_dic[indice,Keq]`) ne sont pas réellement supprimées mais plutôt passées à **None**. Ceci car, si l'on supprime ces entrées et que l'on diminue le nombre de nappes représentées dans le dictionnaire, on ne s'assure pas de passer en revue toutes les nappes existantes par la suite dans `handle_water_table` :

Par exemple, si l'on dispose au départ d'un dictionnaire contenant 2 nappes et que l'on efface toutes les entrées relatives à la numéro 1 tout en diminuant de 1 le nombre de nappes dans le dictionnaire, la méthode `ColumnColumnExchangeLine.calculate_flux_interface` va effectuer les opérations suivantes :

- Chercher le nombre de nappes à explorer = 1
- Explorer la première nappe, soit la numéro 1
- Or, la nappe numéro 1 n'existe plus, c'est la numéro 2 qui existe encore dans le dictionnaire !

Ainsi, pour éviter ce problème, on garde bien les 2 nappes dans le dictionnaire et on rajoute un test dans la méthode `ColumnColumnExchangeLine.calculate_flux_interface` : une nappe ne sera étudiée que si elle contient au moins une cellule saturée.

D'un point de vue du code détaillé, l'algorithme se présente comme suit :

- Appel de la méthode `split_watertable(UH,limit,limit_idx)` pour séparer les nappes et modifier `water_table_dic`.
- Appel de la méthode `ColumnColumnExchangeLine.calculate_flux_interface` pour calculer les flux correspondant aux nappes qui ne seront pas redirigées vers le fossés (pointillés verts sur la fig. 38).
- Remplacement de l'attribut `water_table_dic` par `wt_2linear` (pointillés bleus sur la fig. 38) et second appel de la méthode `ColumnColumnExchangeLine.calculate_flux_interface` (ainsi, on calcule séparément les flux allant vers la colonne aval ou vers le fossé ce qui devrait éviter les confusions).
- Mise à jour des attributs `lat_flux` (eau) et `sol_flux` (solutés) pour les instances `UpstreamInstance`, `DownstreamInstance` et `InterceptInstance` en rajoutant les différentes contributions calculées :  
`UpstreamInstance_lat_flux` est incrémenté grâce au vecteur des flux sortants attribués aux nappes de `water_table_dic` et `wt_2linear`. `Intercept_instance_lat_flux` est incrémenté à partir de l'opposé du flux sortant attribués aux nappes de `wt_2linear_dic` (on ne veut qu'un scalaire). Enfin `DownstreamInstance_lat_flux` est incrémenté grâce au vecteur des flux entrant attribués aux nappes de `water_table_dic`. Le calcul de ces différentes contributions est illustré fig. 39. Le fonctionnement est le même pour le calcul des contributions aux flux de solutés.

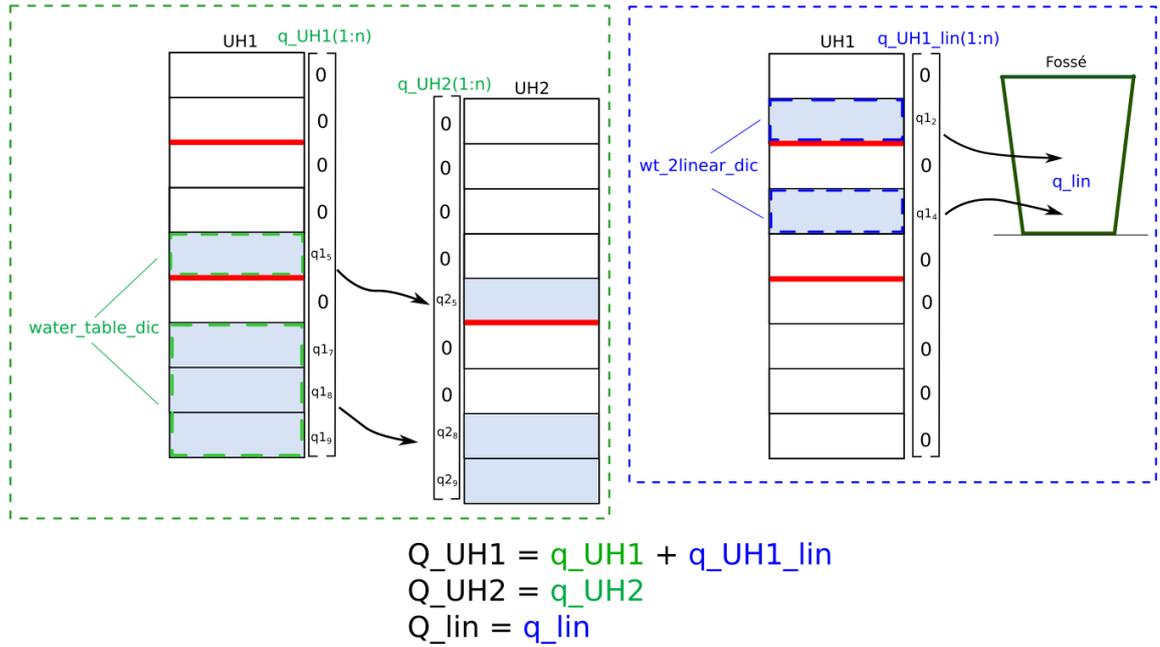


FIGURE 39 – Décomposition du calcul des vecteurs  $Q\_UH1$ ,  $Q\_UH2$  et du scalaire  $Q\_lin$ .

## 10.4 Transferts entre une parcelle et la rivière

### 10.4.1 Fonctionnement hydrologique de la ligne d'échanges

Les flux latéraux de subsurface entre parcelle et rivière constituent également une voie de transfert des contaminants et, à ce titre, doivent être représentés dans PESHMELBA. Plusieurs approches existent pour représenter de tels échanges et sont évoquées dans Dehotin (2007). L'approche choisie dans PESHMELBA est une approche simplifiée basée sur les équations de Miles (Miles, 1985) et déjà utilisée par Dehotin (2007) dans la plateforme LIQUID. Celle-ci se base sur la perte de charge entre la nappe et la rivière pour déterminer le flux à l'interface :

$$Q_u = C_m K_s \Delta h \quad (10.12)$$

où  $Q_u$  est le flux entre la rivière et la zone saturée [ $L^3T^{-1}L^{-1}$ ],  $\Delta h$ , la perte de charge entre la rivière et la nappe [ $L$ ],  $K_s$  la conductivité hydraulique du fond de la rivière [ $LT^{-1}$ ] et  $C_m$ , le coefficient de Miles, calculé par la formule suivante :

$$C_m = \frac{5[0.25(w_s + w) + h + s]}{D_i + h + s} \quad (10.13)$$

où  $w_s$  et  $w$  sont respectivement la largeur au miroir [ $L$ ] et la largeur au radier [ $L$ ] du tronçon de rivière  $h$ , le tirant d'eau dans la rivière [ $L$ ],  $D_i$  est la distance entre le fond de la rivière et l'imperméable de la zone saturée [ $L$ ] et  $s$  la distance de suintement [ $L$ ] (Figure 40).

Tout comme dans LIQUID, un flux est effectivement calculé uniquement lorsque le niveau de nappe est au dessus du fond de la rivière, connectant effectivement la nappe à la rivière (Figure 41 , configurations 1 et 2).

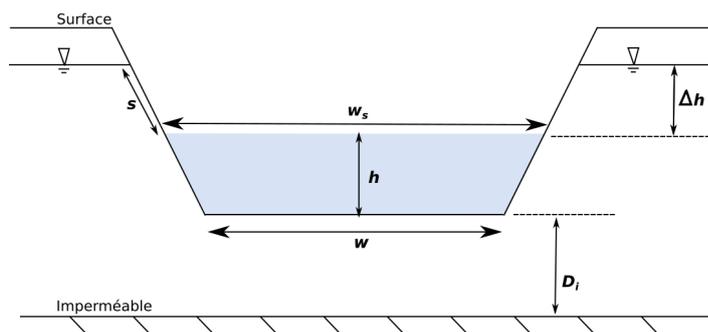


FIGURE 40 – Variables utilisées dans les équations de Miles.

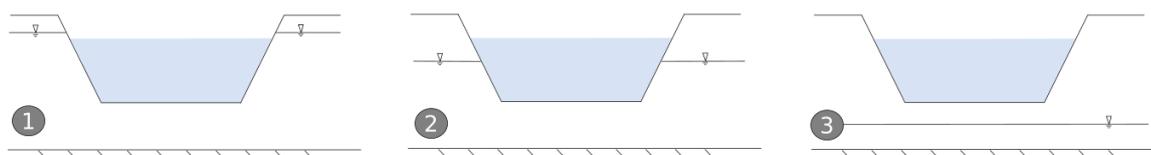


FIGURE 41 – Configurations possibles pour le positionnement de la nappe par rapport au niveau de la rivière.

On note que cette formulation permet de représenter un flux dirigé vers la rivière mais aussi de considérer qu'il y a infiltration depuis la rivière vers la colonne de sol (configuration 2 sur la Fig. ??). De plus, si la parcelle connectée à la rivière contient une nappe perchée, dont le fond est situé au dessus du niveau d'eau dans la rivière, un flux dirigé vers la rivière est calculé en considérant une surface de suintement tout le long de la hauteur de nappe perchée.

### 10.4.2 Implémentation

La ligne d'échanges est codée sous la forme d'une classe `ColumnRiverExchangeLine` qui hérite de la classe `SubsurfaceInterface`. Elle s'initialise avec une instance de la classe `Column` (décrite section 10.2.3) et une instance de la classe `River` (décrite ci-après).

#### Class River

La classe `River` permet d'initialiser un objet `River` avec tous les attributs nécessaires au calcul des échanges entre la nappe et la rivière. Cette initialisation se fait à partir d'une instance de la classe `Reach` (voir description dans la section 9.4.2), de la distance du fond de la rivière à l'imperméable  $d_i$  [L] et d'une conductivité hydraulique à saturation  $ksat$  [L.T<sup>-1</sup>]. En plus d'un constructeur, la classe `River` contient les méthodes suivantes :

- **watercontent2waterlevel()** : Cette méthode renvoie une hauteur d'eau  $waterlevel$  [L] correspondant à un volume d'eau  $watercontent$  [L<sup>3</sup>] donné en entrée calculé en considérant la section trapézoïdale de la rivière.
- **simplegeometry()** : A partir d'un niveau d'eau  $waterlevel$  [L] et d'une perte de charge  $deltaH$  [L], la méthode `simplegeometry` permet de déterminer :

- `topwidth [L]` : la largeur à la surface, appelée  $W_s$  Figure 40 ;
- `seepage_dist [L]` : la distance entre la berge et la surface de la rivière disponible pour le suintement, appelée  $s$  (Fig. 40).
- **MilesCoeff()** :  
 Cette méthode détermine les paramètres indispensables au calcul du coefficient de Miles (formule 10.4.1) avant de le calculer. Les paramètres relatifs à la géométrie et le coefficient de Miles sont renvoyés en sortie.
- **update\_waterlatflux et update\_solutelatflux** :  
 Contrairement à des setters classiques qui remplacent des variables existantes par une nouvelle valeur, les méthodes `update` permettent de mettre à jour la variable `waterlatflux` (resp. `solutelatflux`) en incrémentant la variable existante avec un stock additionnel `increment_waterflux` (resp. `increment_solutelatflux`).

### Class `ColumnRiverExchangeLine`

- **`__init__()`**

La classe `ColumnRiverExchangeLine` est initialisée intégralement avec le constructeur de la classe `SubsurfaceInterface` et à partir des attributs suivants :

- `idx` : un index (scalaire entier) pour repérer l'interface ;
- `length` : une longueur [L] ;
- `x,y` : une longitude et une latitude [L] ;
- `z` : une altitude [L] ;
- `UpstreamInstance` : une instance de la classe `Column`. Il s'agit de l'élément situé à l'amont de l'interface.
- `DownstreamInstance` : une instance de la classe `River`. Il s'agit de l'élément situé à l'aval de l'interface.
- **`update_state_variables()`**  
 Cette méthode permet la mise à jour des variables d'état de la colonne amont au début de chaque pas de temps. Pour cela, elle récupère les variables PALM contenant le vecteur de pression capillaire, de volume d'eau et de concentration en solutés dans chaque cellule numérique et met à jour les attributs de la classe `Column` correspondants. La méthode `get_h_table` de la classe `Column` est également appelée pour finaliser cette mise à jour. Le volume d'eau et la concentration en solutés est également initialisée pour l'instance de `River` située à l'aval de l'interface.
- **`repartition_river()`**  
 Cette méthode permet de passer d'un flux scalaire à un vecteur de flux. En effet, la rivière est représentée par un réservoir unique qui reçoit ou envoie un flux scalaire. Ce flux est reçu ou envoyé vers une colonne de sol composée de plusieurs cellules numériques. La méthode fonctionne donc comme `Column.handle_water_table` en permettant de répartir ce flux scalaire entre les différentes cellules numériques en se basant sur l'attribut `weight` (voir section 10.2.2).

- **calculate\_flux\_interface()**

Cette méthode calcule le flux traversant l'a ligne d'échanges entre l'instance de Column et l'instance de River. Pour cela, elle comment par appeler la méthode `get_h_table_unified` de l'instance de Column amont pour déterminer toutes les nappes et leur position dans cet élément là. La méthode détermine dans quelle configuration de la Figure 41 se trouve chacune de ces nappes par rapport au niveau de la rivière.

S'il s'agit des configurations 1 ou 2, le flux est calculé par la formule de Miles, s'il s'agit de la configuration 3 (suintement), on calcule un flux de suintement en considérant la différence de charge entre la nappe de la colonne et la pression atmosphérique. Dans tous les cas, la méthode `repartition_river` permet ensuite de transformer ce flux scalaire en un vecteur de flux sortant ou entrant pour chaque cellule numérique de la colonne amont. Comme pour tous les types de lignes d'échanges, les flux d'eau et de solutés entrants/sortants calculés sont stockés dans les attributs `Upstream_lat_flux`, `Downstream_lat_flux`, `Upstream_sol_flux` et `Downstream_sol_flux`.

## 10.5 Transferts entre une parcelle et une haie

### 10.5.1 Introduction : fonctionnement hydrologique d'une haie

Le fonctionnement global d'une haie, d'une haie sur talus ou d'un talus nu situés entre 2 parcelles est décrit ici. On se restreint pour le moment à cette configuration mais dans les versions futures de PESHMELBA, il faudra également pouvoir traiter les cas où la haie/talus est située en bordure d'un fossé ou en bordure quasi immédiate d'un cours d'eau.

Pour un emplacement entre 2 parcelles, les haies/talus/haies sur talus sont des structures qui :

- interceptent plus ou moins le ruissellement en transférant une partie **le long de leur tracé** (en fonction de leur orientation par rapport à la pente, de leur pente et de leur rugosité) et une partie **vers la parcelle aval** (si on est en présence d'une haie simple ou si la partie talus est submergée) ;
- ont une capacité d'infiltration accrue par rapport aux parcelles avoisinantes avec possiblement une infiltration rapide jusqu'à la nappe (structuration du sol riche en matière organique, racines...) et avec une capacité d'adsorption et de dégradation accrue ;
- n'interfèrent pas avec les transferts latéraux de subsurface hormis grâce à une capacité d'évapotranspiration accrue par rapport aux parcelles avoisinantes.

Ce fonctionnement est résumé Figure 42.

### 10.5.2 Représentation adaptée aux processus de subsurface

#### Processus verticaux

L'infiltration matricielle est représentée par le module FRER1D (chapitre 7) et l'évapotranspiration par le bloc PARTITION et le module ROOT (chapitre 8). Un

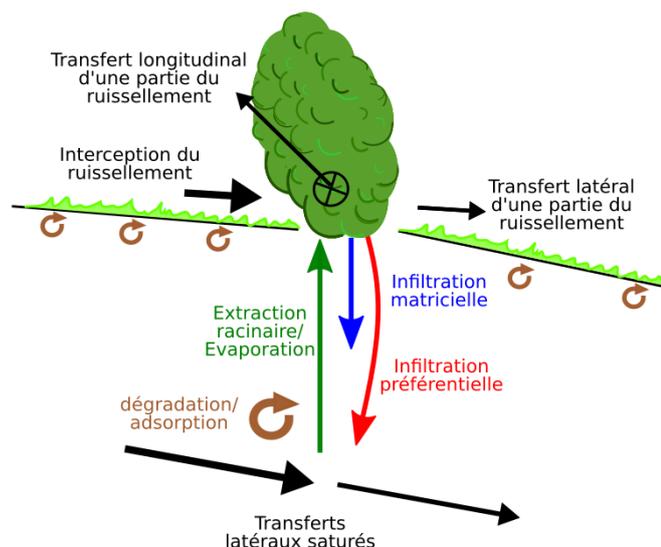


FIGURE 42 – Fonctionnement complet d'une haie.

paramétrage adapté du LAI et de la densité racinaire permettront directement de rendre compte des capacités accrues d'évapotranspiration par rapport aux parcelles alentour. De même, un paramétrage incluant un taux élevé de matière organique dans le sol permettra de rendre compte d'une adsorption importante au niveau de la haie. Il faut noter que l'évapotranspiration accrue due à la présence d'une haie a fréquemment lieu sur une largeur bien supérieure à l'emprise au sol du linéaire. Cependant, il est difficile de rendre compte de cette largeur d'influence dans PESHMELBA sous peine d'extraire trop d'eau. En effet, en considérant une largeur d'influence de la haie qui empiète sur les éléments voisins, cela conduirait à une surestimation des prélèvements en eau sur la surface de la largeur d'influence qui se superpose aux éléments voisins.

### Transferts latéraux de subsurface

Les échanges latéraux de subsurface avec les parcelles avoisinantes ne sont pas perturbés par la présence de la haie. Il a donc été choisi de reprendre la base développée pour l'interface ColumnColumnExchangeLine pour les décrire. On rappelle en effet qu'une haie est modélisée en subsurface par 2 colonnes de sol. Elle peut donc être connectée aux parcelles amont et en aval par des interfaces de type ColumnColumnExchangeLine. Une interface de ce type là permet également de décrire les échanges latéraux entre les parties supérieure et inférieure de la haie.

### Écoulements préférentiels

Le pied de haie a été identifié comme un lieu privilégié d'écoulements verticaux rapides par macropores. La représentation des écoulements préférentiels reste encore un défi d'un point de vue modélisation et des modèles complexes sont nécessaires pour les représenter correctement (double porosité, double perméabilité). Cet aspect devra être traité rigoureusement dans PESHMELBA mais dans sa version actuelle, une représentation très simplifiée de tels écoulements a été choisie. En surface, une portion des apports en eau et en solutés est stockée dans un réservoir d'infiltration

préférentielle. Ce volume d'eau et les solutés dissous qu'il contient est déterminé par la partie d'interface latérale de surface (section 11.3.4) grâce à la capacité d'infiltration préférentielle caractérisant la haie. Ensuite, ce stock est utilisé comme forçage latéral dans une des cellules numériques décrivant la subsurface de la haie en profondeur. S'il existe une nappe en fond de profil, la cellule située juste au dessus de la nappe est sélectionnée pour recevoir le volume d'eau et la masse de soluté provenant du réservoir d'infiltration préférentielle. S'il n'y a pas de nappe en fond de profil, la cellule numérique la plus profonde est sélectionnée.

Sur la partie supérieure de la haie, cette contribution latérale est agrégée au flux latéral déjà calculé par l'interface avec la parcelle amont. Sur la partie inférieure de la haie, elle est agrégée au flux latéral calculé entre les 2 parties de la haie (Figure 43).

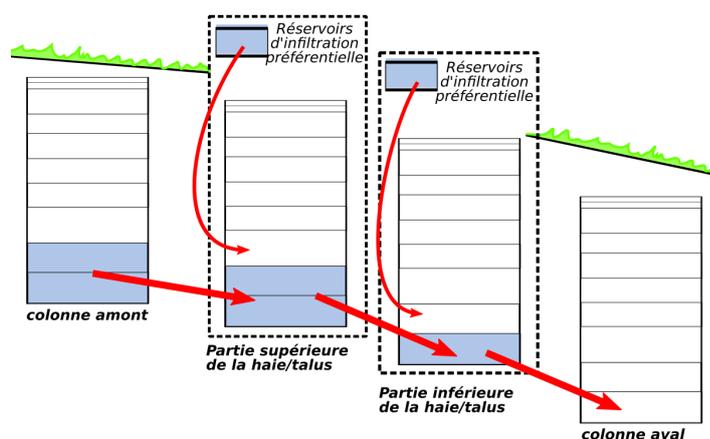


FIGURE 43 – Représentation des échanges latéraux de subsurface dans une haie entre 2 parcelles.

### 10.5.3 Implémentation

Les échanges latéraux en subsurface entre colonne de sol voisine et haie ou à l'intérieur d'une haie sont calculés de la même manière que pour une ligne d'échange entre 2 parcelles. Pour cela, une classe `HedgeSubsurface` a été développée pour représenter la subsurface d'un élément de type haie, composée de 2 colonnes de sol. Ensuite, la classe `ColumnHedgeExchangeLine` permet de gérer les transferts latéraux saturés "Colonne de sol vers Haie" ou "Haie vers Colonne de sol". Enfin, la classe `HedgeInternExchangeLine` permet de gérer les transferts latéraux en subsurface ayant lieu depuis la partie amont de la haie vers la partie aval. Cette classe permet également d'inclure la composante de macropores dans les parties amont et aval de la haie.

#### Classe `HedgeSubsurface`

- **constructeur init :**

La classe `HedgeSurface` dérive de la classe `Linear`. Elle s'initialise à partir d'un indice, une position, une altitude, une longueur, une largeur et des attributs **ColumnUp** et **ColumnDown**. Ce sont des instances de la classe `Column` qui représentent respectivement les parties amont et aval d'un tronçon de haie.

- **update\_waterlatflux et update\_solutelatflux :**

Contrairement à des setters classiques qui remplacent des variables existantes par une nouvelle valeur, les méthodes update permettent de mettre à jour la variable waterlatflux (resp. solutelatflux) en incrémentant la variable existante avec un stock additionnel increment\_waterflux (resp. increment\_solutelatflux). Dans le cas des haies, l'argument updown permet de choisir si l'on met à jour le stock d'eau (resp. de solutés) dans la colonne de sol amont (updown="up") ou aval de la haie (updown="down"). Le stock propre de l'instance de HedgeSubsurface est aussi mis à jour, avec ses composantes amont et aval à partir des argument ColumnUp.waterlatflux et ColumnDown.waterlatflux. Même si ça peut paraître un peu lourd (on remet à jour les 2 composantes de waterlatflux et solutelatflux pour l'instance de HedgeSubsurface même si une seule a effectivement changé), cette structure est nécessaire pour garder un ensemble générique et traiter indifféremment des lignes d'échanges ColumnHedgeExchangeLine et HedgeInternExchangeLine.

### Classe ColumnHedgeExchangeLine

Comme les lignes d'échanges Colonnes-Haie fonctionnent comme des lignes d'échanges Colonne-Colonne, la classe ColumnHedgeExchangeLine hérite de la classe ColumnColumnExchangeLine et ne sert basiquement qu'à appeler les méthodes de cette dernière. Le constructeur de ColumnHedgeExchangeLine appelle le constructeur de ColumnColumnExchangeLine en distinguant les cas "Colonne C1 vers Haie H1" et "Haie H1 vers Colonne C1". Dans le premier cas, une instance de la classe ColumnColumnExchangeLine est initialisée entre C1 et la colonne amont de H1 et dans le second cas, l'instance est initialisée entre la colonne aval de H1 et la colonne C1. De même, la méthode update\_state\_variables initialise les différentes variables d'état en distinguant les 2 types de connexions puisque les objets PALM appelés ne sont pas les mêmes selon qu'on souhaite mettre à jour une variable d'une colonne ou d'une des colonnes d'une haie.

### Classe HedgeInternExchangeLine

La classe HedgeInterneExchangeLine hérite aussi de la classe ColumnColumnExchangeLine. Une ligne d'échange est toujours initialisée par le constructeur entre les partcolonnesies amont et aval de la haie. La méthode update\_state\_variable fonctionne exactement comme pour ColumnColumnExchangeLine sauf qu'elle appelle des objets PALM de type "hedge" pour mettre à jour ses variables. La méthode calculate\_flux\_interface a été surchargée : on fait appel à la méthode calculate\_flux\_interface de la classe mère puis à la méthode calculate\_macroporous\_flux pour rajouter la contribution des écoulements préférentiels. Cette dernière constitue la seule spécificité de la classe HedgeInternExchangeLine. Les stocks d'eau et de solutés alloués aux écoulements préférentiels dans les parties amont et aval de la haie sont d'abord récupérés depuis PALM (wat2macropore et sol2macropore). On récupère aussi les stocks d'eau et de solutés disponibles en surface de la haie (hors stock alloué aux écoulements préférentiels). Dans un second temps, on fait appel à la méthode get\_h\_table\_unified de la classe Column pour identifier une potentielle nappe en fond de profil dans les parties amont et aval de la haie. Si une telle nappe existe, la variable idx\_cell\_to\_fill est associée à

l'indice de la cellule numérique située juste au dessus du toit de cette nappe. On note qu'en appelant la méthode `get_h_table_unified` au lieu de la méthode `get_h_table`, le dictionnaire des nappes présentes dans chaque colonne de sol ne divise pas les nappes si elles s'étalent sur plusieurs horizons. Cela facilite l'identification de la première cellule non saturée au dessus de la nappe de fond de profil. S'il n'y a pas de nappe dans une des 2 colonnes de sol de la haie, on stocke dans `idx_cell_to_fill` l'indice de la cellule numérique la plus basse de la colonne. Ensuite, les variables `Upstream_lat_flux` et `Downstream_lat_flux` (et `Upstream_sol_flux` et `Downstream_sol_flux`) sont mises à jour en rajoutant pour la cellule `idx_cell_to_fill` la contribution `wat2macropore/dt` et `sol2macropore/dt`. Si l'une des colonnes de la haie est entièrement saturée, on ne rajoute pas la contribution des macropores car `FRER1D` n'est plus capable de gérer une condition de flux latéral entrant et le bilan de masse final n'est pas bon (eau prélevée du stock de macropores mais pas allouée en subsurface). Dans ce cas, on réintègre les stocks alloués aux macropores dans `met` à jour les niveaux d'eau et les masses de solutés disponibles en surface.

## 10.6 Transferts vers/depuis un élément de condition limite (BCS)

### 10.6.1 Fonctionnement d'un élément de type BCS

Les conditions limites en subsurface disponibles pour les éléments BCS (Boundary Condition Storage) sont toutes de type "Flux" afin d'être compatibles avec le module `FRER1D`. Elles sont résumées dans la Figure 44.

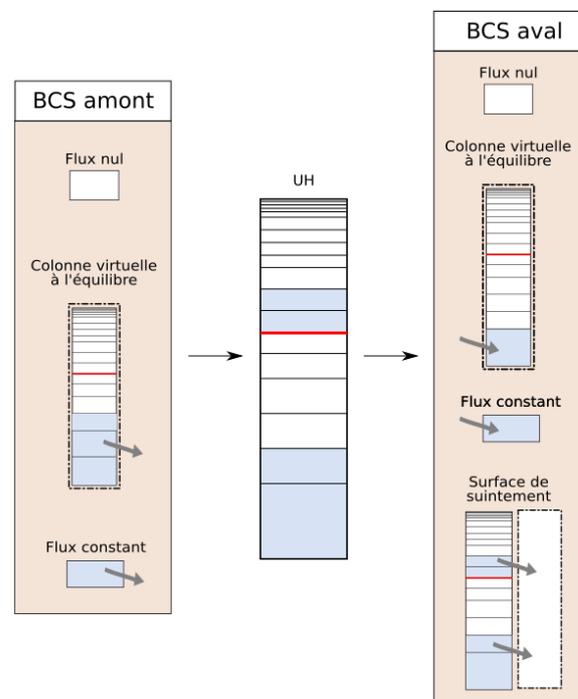


FIGURE 44 – Types de conditions limites en subsurface qu'il est possible de simuler avec un élément du type BCS.

### Condition amont

Si l'élément BCS est connecté en amont d'une colonne, il peut lui fournir un flux nul ou un flux entrant, provenant d'une colonne virtuelle amont ou défini par l'utilisateur. Dans le cas d'un flux constant provenant d'une colonne virtuelle amont, celui-ci est calculé à partir du niveau de nappe indiqué comme paramètre de la colonne virtuelle. Si cette hauteur de nappe ne permet pas d'échange avec la colonne aval (gradient non compatible avec la topographie), le flux résultant est nul. Dans le cas contraire, les échanges se font de la même manière que pour les échanges Parcelle-Parcelle (voir chapitre 10.2).

Dans le cas d'un flux constant défini par l'utilisateur, celui-ci est toujours non nul et égal à la valeur définie en entrée. Cependant, il est seulement utilisé pour recharger une éventuelle nappe de fond de profil. Même si la colonne de sol comporte plusieurs horizons de sol, c'est toujours le dernier horizon qui reçoit le flux entrant. En l'absence d'une nappe de fond de profil, la dernière cellule numérique de la colonne reçoit l'intégralité du flux entrant.

Dans les deux cas, en plus de la valeur du flux d'eau, l'utilisateur définit également une concentration en solutés qui permet de déduire un flux de solutés entrant. Un BCS peut également fournir une condition limite amont en surface. Ce point est détaillé dans la section relative aux écoulements de surface (section 11.3.5).

### Condition aval

Si l'élément BCS est connecté en aval d'une colonne de sol, la condition limite associée peut être un flux nul, une condition de suintement sur toute la hauteur de la colonne de sol ou un flux sortant. Ce flux sortant peut-être calculé en considérant une colonne virtuelle à l'aval définie par un niveau de nappe ou peut-être considéré comme constant, défini par l'utilisateur.

Dans le cas d'un flux sortant vers une colonne virtuelle aval, celui-ci est également déterminé à partir du gradient hydraulique entre la colonne amont et le BCS-colonne virtuelle aval. Si le gradient hydraulique n'est pas compatible avec le gradient topographique, le flux est nul.

De même, dans le cas d'un flux constant sortant, celui-ci résulte forcément d'une nappe de fond de profil. En cas d'absence de zone saturée dans la colonne de sol amont, le flux est nul.

La condition de suintement quant à elle, est calculée sur l'ensemble de la colonne de sol, ce qui signifie que le flux sortant résultant peut être associé à une nappe de fond de profil et à des nappes perchées. Le calcul du flux de suintement se fait de la même manière que dans l'interface Parcelle-Rivière-River (voir section 10.4).

Dans les 2 cas, le flux de solutés sortant est calculé de la même manière que dans le cas d'une interface ColumnColumn ou ColumnRiver, à partir des concentrations dans les cellules numériques concernées par le flux d'eau sortant.

## 10.6.2 Implémentation

La ligne d'échanges est codée sous la forme d'une classe `ColumnBCSExchangeLine`. Celle-ci hérite de la classe `SubsurfaceInterface` et s'initialise entre autre à partir d'une

instance de la classe Column et une instance de la classe BCS.

### Classe Boundary Condition Storage (BCS)

- **constructeur `__init__()`**

Les BCS et leurs propriétés sont implémentés à travers la classe BCS. Celle-ci dérive de la classe PunctualElement et est initialisée à partir des attributs suivants :

- `idx`, `lon`, `lat`, `alt` : attributs permettant d'initialiser la classe Element
- `updown` (float) : nature du BCS (sink : 0 ou source : 1). [a supprimer dans le code car on en a pas besoin](#)
- `typeBC` (float) : type de condition limite associée au BCS (0-flux : 0, virtual column : 1, constant flux : 2, seepage face : 3)
- `subsurfaceBC` (float) : valeur du flux constant en subsurface si `typeBC=2`. La valeur est **positive** pour un flux sortant et **négative** pour un flux entrant.
- `surfaceBC` (float) : condition limite en surface ( 0 : pas de flux ,  $x>0$  :  $x\text{-cm}^3/\text{h}$  flux )
- `cwsurf/cwsubsurf` (facultatif, float(1 :n\_solute)) : dans le cas d'un BCS source, concentration en solutés dans le flux d'eau entrant en surface/subsurface
- `wtable_height` (facultatif, float) : Dans le cas d'un BCS colonne virtuelle, profondeur de la nappe à partir du niveau de référence qu'est la surface (positive).

- **`update_state_variables()`**

L'état d'un élément de type BCS peut être mis à jour par cette méthode. Elle permet d'accéder aux variables PALM et de mettre à jour les attributs suivants :

- `wqlat_bcs` (double) : le flux d'eau entrant/sortant du BCS [ $\text{L}^3.\text{T}^{-1}$ ];
- `sqlat_bcs` vect(1 :n\_solute) : le flux de solutés entrant/sortant du BCS [ $\text{M}.\text{T}^{-1}$ ];
- `wcontent_subsurf_bcs` (double) : le volume d'eau qui a été reçu/envoyé du BCS en subsurface [ $\text{L}^3$ ];
- `wcontent_surf_bcs` (double) : le volume d'eau qui a été reçu/envoyé du BCS en surface [ $\text{L}^3$ ];
- `smass_subsurf_bcs` (vect(1 :n\_solute)) : la masse de solutés qui a été reçue/envoyée du BCS en subsurface [M];
- `smass_surf_bcs` (vect(1 :n\_solute)) : la masse de solutés qui a été reçue/envoyée du BCS en surface [M];

- **`get_h_vector(x)`**

Cette méthode est utilisée lorsque l'élément BCS est de type colonne virtuelle. Elle permet de définir un vecteur de pression capillaire à chaque profondeur contenue dans un vecteur d'entrée `x(1 :n_layer)`. La pression capillaire est calculée en considérant le niveau de nappe `wth` et un équilibre hydrostatique.

- **`update_waterlatflux` et `update_solutelatflux`**

Contrairement à des setters classiques qui remplacent des variables existantes par

une nouvelle valeur, les méthodes `update` permettent de mettre à jour la variable `waterlatflux` (resp. `solutelatflux`) en incrémentant la variable existante avec un stock additionnel `increment_waterflux` (resp. `increment_solutelatflux`).

### ColumnBCSExchangeLine

La classe `ColumnBCSExchangeLine` permet de construire une ligne d'échanges entre une Colonne et un BCS situé en amont ou en aval. Elle est composée d'une instance de la classe `Column`, d'une instance de la classe `BCS` et d'attributs caractérisant l'interface (un indice d'identification, des coordonnées géographiques, etc...).

- **constructeur `__init__()` :**

Une instance de la classe `ColumnBCSExchangeLine` est initialisée à partir de :

- `idx` (int) : une indice d'identification de l'interface.
- `lon`, `lat` (int) : les latitude et longitude de l'interface.
- `length` (int) : la longueur de l'interface.
- `UH`, `bcs` : les instances des classes `Column` et `BCS` qui sont connectées à travers l'interface construit.
- `dt` : le pas de temps auquel sera calculé le flux résultant.

On initialise également les indices de la colonne et du BCS car on ne connaît pas jusque là pas leur position (lequel des deux est à l'amont et à l'aval).

- **`update_state_variables`**

- **`calculate_flux_interface`** La méthode `calculate_flux_interface` permet de déterminer les flux entrant/sortant de l'UH et du BCS en fonction du type de condition limite. La méthode vérifie le type de l'élément BCS, sa compatibilité avec le flux entré par l'utilisateur (par ex, un `tpe=1` doit forcément correspondre à un flux positif) avant de calculer le flux correspondant. Les variables `QUH`, `QsolUH`, `Qbcs` et `Qsolbcs` sont alors mises à jour avec les flux calculés.

La méthode de calcul du flux dépend du type de condition limite associé au BCS :

- **Flux nul** : les variables `QUH`, `QsolUH`, `Qbcs` et `Qsolbcs` sont toutes fixées à 0 (scalaire ou vecteur).
- **Colonne virtuelle** : afin de calculer le flux saturé entre les deux colonnes, il est nécessaire de créer une instance virtuelle de la classe `Colonne` associée au BCS. Celle-ci est entre autre initialisée avec les coordonnées du BCS, le vecteur de pression hydrostatique déterminé par la méthode `get_h_vector`. L'ensemble des autres attributs sont copiés à partir de ceux de l'UH composant l'interface. Cela signifie qu'une colonne virtuelle possède la même géométrie, le même maillage et les mêmes propriétés de sol que l'UH à laquelle est connectée. Une instance de la classe `interface_subsurface` est ensuite initialisée pour permettre le calcul du flux saturé entre les deux colonnes. Les échanges se font alors le long de la ligne d'échanges de longueur `length`. On note que l'on sépare les cas où le BCS est à l'aval ou à l'amont car l'interface entre 2 colonnes n'est pas initialisée de la même manière.
- **Surface de suintement** Chaque zone saturée de l'UH est scannée pour déterminer un flux sortant de suintement associé (scalaire). Celui-ci est ensuite transformé en vecteur par la méthode `split_flux` (voir ci-dessous).

- **split\_flux** La méthode `split_flux` de la classe `ColumnBCSEExchangeLine` est à rapprocher de la méthode `repartition_river` de la classe `PlotRiverExchangeLine`. Elle est appelée par la méthode `calculate_flux_interface` dans le cas d'une surface de suintement et permet de répartir un flux scalaire  $Q$  sortant de l'UH entre les cellules numériques *cells* de la zone saturée associées à ce flux. La méthode se base sur les poids de répartition de la classe `Colonne` (attribut `weight`) pour faire cette répartition et incrémente les variables `QUH` et `QsolUH`.

## 10.7 Module EXCHANGES\_SUBSURFACE

EXCHANGES\_SUBSURFACE est le seul module relatif aux échanges latéraux de subsurface qui apparaît explicitement dans le schéma PALM. Il réutilise toutes les classes décrites dans ce chapitre ainsi que plusieurs fonctions de service pour déterminer les transferts latéraux de subsurface, d'eau et de solutés. Afin d'optimiser l'application finale, ce module a été redivisé : EXCHANGES\_SUBSURFACE contient la partie d'initialisation des différentes instances et c'est ensuite dans le script `use_EXCHANGES_SUBSURFACE`, qui est appelée à chaque pas de temps, que sont déterminés les transferts latéraux.

### 10.7.1 Initialisation d'instances d'éléments du paysage et de lignes d'échanges entre éléments

EXCHANGES\_SUBSURFACE se base sur l'utilisation de structures de type dictionnaire pour regrouper dans un même objet plusieurs objets de natures différentes. Le dictionnaire *Element\_dic* contient tous les éléments impliqués dans des transferts latéraux de subsurface sous la forme d'instances de classes représentant ces éléments. Ce dictionnaire regroupe ainsi des instances de la classe `Column`, `Ditch`, `River`, `HedgeSubsurface` et `BCS`. Pour chaque type d'élément, l'ensemble des paramètres nécessaires au constructeur de la classe associée est chargé depuis des variables ou des objets PALM. Chaque instance est alors initialisée puis stockée dans le dictionnaire *Element\_dic*. Ensuite, avec l'analyse de la table *lateral\_connection\_complete* contenant une description de toutes les connexions latérales entre éléments, le dictionnaire *ExchangeLine\_dic* est aussi initialisé. Cette initialisation est faite par la fonction `initiaize_interface_subsurface` décrite dans la section 10.7.4 à partir de la table des connexions latérales *lateral\_connection\_complete* et du dictionnaire d'éléments *Element\_dic*. Pour chaque ligne d'échanges identifiée, une instance de type `XXXExchangeLine` est créée. Que ça soit pour les objets du dictionnaire *Element\_dic* ou ceux de *ExchangeLine\_dic*, cette étape d'initialisation se fait seulement à partir de paramètres des objets, lesquels ne varient pas au cours du temps. Elle n'a donc lieu qu'une seule fois en début de simulation et les dictionnaires sont utilisés, après mise à jour des variables d'état, à chaque pas de temps.

### 10.7.2 Calcul des flux latéraux entre éléments

A chaque pas de temps, les dictionnaires *Element\_dic* et *ExchangeLine\_dic* sont ensuite transmis à la fonction `use_EXCHANGE_SUBSURFACE`. A partir de la description des éléments du bassin et des lignes d'échanges les reliant, on peut calculer les flux

latéraux à chaque pas de temps. Le vecteur contenant tous les pas de temps *time\_vect* et l'indice du pas de temps courant *time\_index* sont d'abord chargés. Les variables *waterlatflux* et *solutelatflux* sont aussi initialisées pour chaque élément du dictionnaire *Element\_dic*. Grâce à la notion de polymorphisme ceci est fait par une simple boucle sur les éléments d'*Element\_dic* qui appelle les méthodes *initialize\_subsurf\_waterlatflux* et *initialize\_subsurf\_solutelatflux* sans avoir à se soucier du type d'élément manipulé. Sur le même principe, le calcul du flux à travers les lignes d'échanges est ensuite lancé pour chaque élément du dictionnaire *ExchangeLine\_dic*. C'est fait par la méthode *calculate\_flux\_interface(dt)*. Chaque instance contenue dans le dictionnaire possède cette méthode même si les actions qui lui sont associées peuvent être différentes d'une classe à l'autre. Une fois de plus, cette application de la notion de polymorphisme permet de garder un haut degré de généralité dans le module final.

⚠ Il est important de noter qu'en plus des notions d'héritage et de polymorphisme, la structuration des codes adoptée dans PESHMELBA se base sur une troisième notion, spécifique à Python qui est celle du passage de variable par référence. En effet, dans Python lorsqu'on fait une déclaration du type :

```
a = [1, 2, 3, 4]
b = a
b.append(4)
```

L'objet **a** a aussi été transformé et une impression de **a** donne [1,2,3,4]. En effet, dans Python tous les objets se passent toujours par référence. Ceci peut souvent poser souci mais ici on exploite cette spécificité pour obtenir un code encore plus générique. Dans le cas du module EXCHANGES\_SUBSURFACE, chaque interface initialisée l'est avec des objets provenant d'*Element\_dic*. Si un objet d'*Element\_dic* est modifié dans le cadre d'une interface, l'objet sera aussi modifié dans toutes les instances d'interface dans lesquelles il apparaît. Ainsi, cela garantit une mise à jour continue de l'état de tous les objets.

### 10.7.3 Envoi des variables PALM

Dans un dernier temps, pour chaque élément du dictionnaire *Element\_dic*, les flux latéraux d'eau et de solutés calculés sont renvoyés vers le coupleur PALM. Contrairement à ce qui a été fait avant, l'envoi se fait grâce à un test sur la nature de l'élément concerné. Selon sa classe, la variable à envoyer est nommée différemment et il était ici impossible de n'utiliser qu'un nom générique. En effet, selon la nature de l'élément concerné, on renvoie vers PALM un vecteur ou un scalaire et il faut décrire très précisément l'espace associé à l'objet à envoyer. Il paraissait donc plus judicieux de faire ici une description explicite.

### 10.7.4 Fonction de service : *initialize\_interface\_subsurface*

La fonction *initialize\_interface\_subsurface* a pour objectif d'initialiser un dictionnaire contenant toutes les lignes d'échanges en subsurface d'un bassin. Elle prend en entrée *lateral\_connection\_table*, l a table de connexions latérales entre éléments et *Element\_dic*, le dictionnaire contenant des instances représentant tous les éléments du

```

ExchangeLine_to_class_map = {
    ConnectedElement(up=Column,down=Column,intercept=None): ColumnColumnExchangeLine ,
    ConnectedElement(up=Column,down=Column,intercept=Ditch): ColumnDitchColumnExchangeLine ,
    ConnectedElement(up=Column,down=River,intercept=None): ColumnRiverExchangeLine ,
    ConnectedElement(up=Column,down=BCS,intercept=None): ColumnBCSExchangeLine ,
    ConnectedElement(up=BCS,down=Column,intercept=None): ColumnBCSExchangeLine ,
    ConnectedElement(up=Column,down=HedgeSubsurface,intercept=None): ColumnHedgeExchangeLine ,
    ConnectedElement(up=HedgeSubsurface,down=Column,intercept=None): ColumnHedgeExchangeLine ,
}

```

FIGURE 45 – Dictionnaire `ExchangeLine_to_class_map` dans la fonction `initialize_interface_subsurface`

bassin modélisé. Pour chaque ligne de la table `lateral_connection_table`, la méthode permet d’analyser la nature des éléments amont et aval et éventuellement d’un élément interceptant les transferts. En fonction de la nature de ces éléments, la méthode sélectionne le type de ligne d’échanges à initialiser.

Cette sélection est faite par l’approche de la Factory Method : un dictionnaire générique `ExchangeLine_to_class_map` est créé. Les clés sont constituées d’un triplet `[up,down,intercept]` contenant les combinaisons de classes possibles pour les éléments amont, aval et interceptant d’une interface. La valeur associée à chacun de ces triplets est la classe de la ligne d’échanges qu’il faut créer pour chaque combinaison. Par exemple, si le triplet donné en clé est `[up=Column,down=Column,intercept=None]`, la valeur associée est la classe **ColumnColumnExchangeLine**. Pour le triplet `[up=Column,down=Column,intercept=Ditch]` la valeur associée est la classe **ColumnDitchColumnExchangeLine**. Toutes les associations actuellement codées dans PESHMELBA sont présentées Figure 45.

Ainsi, chaque ligne de la table `lateral_connection_table` est analysée. Les indices des éléments `up`, `down` et `intercept` sont utilisés pour aller explorer le dictionnaire `Element_dic` et pour retrouver la classe dont ils sont une instance. Ces classes sont utilisées pour déterminer le triplet `[up,down,intercept]` correspondant et ainsi le type de ligne d’échanges à initialiser. Les ligne d’échanges créées sont regroupées dans le dictionnaire `ExchangeLine_dic` qui est retourné en sortie de la fonction.

⚠ A chaque nouveau type d’interface implémentée dans le modèle, une nouvelle correspondance doit être créée dans le dictionnaire `ExchangeLine_to_class_map`. Si une combinaison `[up,down,intercept]` trouvée dans la table `lateral_connection_table` ne correspond à aucune clé du dictionnaire `ExchangeLine_to_class_map`, une `ValueError` est levée.

# Chapitre 11

## Écoulements de surface

### 11.1 Introduction

Une représentation réaliste du transfert de solutés à l'échelle du bassin versant ne peut se faire sans la prise en compte des écoulements de surface. Le ruissellement est en effet un des vecteurs majeurs du transfert rapide des pesticides (sous forme dissoute ou adsorbée) des parcelles jusqu'aux cours d'eau, notamment pendant les événements pluvieux (Peyrard, 2016). Il est associé à deux processus :

- Le ruissellement hortonien (Horton, 1933) généré par un dépassement de la capacité d'infiltration du sol.
- Le ruissellement par saturation qui se déclenche lorsque des précipitations tombent sur des surfaces saturées. C'est notamment le cas pour des sols dont la nappe phréatique est proche de la surface, et sur les sols dans lesquels une couche imperméable existe à faible profondeur (socle rocheux ou horizon très argileux) favorisant l'apparition de nappes perchées.

### 11.2 Représentation du ruissellement

Le ruissellement est classiquement représenté par une des approximations des équations de St Venant (section 9.2). Ainsi, l'onde diffusive est utilisée dans les modèles Geo-MHYDAS (Lagacherie et al., 2010), CMF (Kraft et al., 2012) ou ZIN-AgriTra (Gassmann et al., 2013) et l'onde cinématique est utilisée dans le modèle openLISEM (Roo et al., 1996). Dans PESHMELBA, l'approximation de l'onde cinématique est utilisée car elle ne nécessite pas de définir de condition limite aval permettant une simplification de la procédure de calcul. On se base également sur la formule empirique de Manning-Strickler (éq. 9.3) pour relier hauteur de la lame d'eau et vitesse moyenne de l'écoulement.

#### 11.2.1 Couplage surface-subsurface

Si les techniques de modélisation des écoulements de surface et de subsurface permettent aujourd'hui des représentations robustes et réalistes, le couplage entre les deux reste encore délicat, notamment à cause des différentes échelles de temps et de la

non-linéarité des processus rentrant en jeu (Spanoudaki et al., 2009).

Une première approche consiste à adopter une représentation très simplifiée de ces interactions en se basant par exemple sur une approche capacitive. Dans TOPMODEL (Beven and Kirkby, 1979) par exemple, le ruissellement correspond à la portion de précipitations arrivant à la surface du sol et dépassant la capacité du réservoir d'infiltration situé au dessous. Dans le modèle SACADEAU (Tortrat, 2005), le partage entre infiltration et ruissellement pour un volume d'eau arrivant en surface d'une parcelle est fait à partir de "tableaux experts" nécessitant une grande quantité de données de terrain.

Une approche plus physique mais encore simplifiée consiste à implémenter l'interaction surface/subsurface dans le modèle de surface en considérant l'écoulement de subsurface comme un terme puits à partir du modèle d'infiltration. L'équation de Green et Ampt est alors communément utilisée pour estimer le terme puits d'infiltration. Le modèle ZIN-AgriTra par exemple évalue la quantité d'eau infiltrée à partir de l'équation de Green et Ampt puis résout les équations de Saint-Venant en surface en se basant sur l'approximation de l'onde diffusive.

Enfin, pour une représentation plus fine, les modèles à base physique s'attachent à résoudre les équations gérant les écoulements sur un maillage fin en surface et en subsurface (les modèles les plus détaillés s'appuient sur l'équation de Richard's 3D en subsurface et sur une approximation des équations de Saint-Venant en surface). Les interactions surface/subsurface sont alors gérées par différentes méthodes de couplage : changement de condition à la limite comme dans les modèles CATHY (Camporese et al., 2010) ou SHE (Abbott et al., 1986), couplage du premier ordre (CMF), ou encore d'autres approches plus innovantes comme l'approche darcéenne multidomaine (Weill, 2007). Ces approches sont bien sûr les plus réalistes d'un point de vue physique mais se heurtent souvent à des difficultés en terme de stabilité, convergence et temps de calcul, notamment lors du passage à l'échelle du bassin versant.

### 11.2.2 Interception du ruissellement par les éléments linéaires

Le ruissellement est particulièrement sensible à l'organisation spatiale du paysage. Les écoulements de surface se dirigent systématiquement vers l'aval topographique mais peuvent être redirigés par les éléments du réseau de linéaires (haies, fossés, route, talus) qui peuvent accélérer ou ralentir le transfert vers les cours d'eau selon leur orientation. Si le ruissellement est intercepté par un fossé, l'eau et les solutés qu'elle contient sont transportés dans le réseau vers le cours d'eau (Figure 46). Cette partie est décrite dans le chapitre 9.

Si le ruissellement est intercepté par une structure de type haie, talus ou haie sur talus, l'écoulement est entièrement ou partiellement bloqué. Une partie de l'eau et des solutés qu'elle contient peut être infiltrée, une partie peut être redirigée en surface vers les tronçons connectés à l'aval en longitudinal et une partie peut traverser la structure si la hauteur (ou l'absence) de talus le permet (voir Figure 47).

Cette répartition dépend de l'orientation de la structure par rapport à la ligne de plus grande pente, de sa rugosité en pied de haie et de la hauteur du talus.

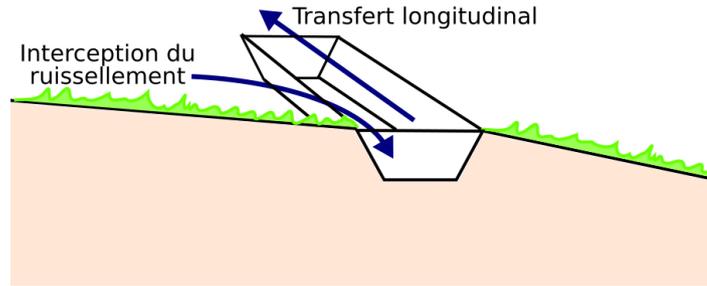


FIGURE 46 – Devenir du ruissellement intercepté par un fossé

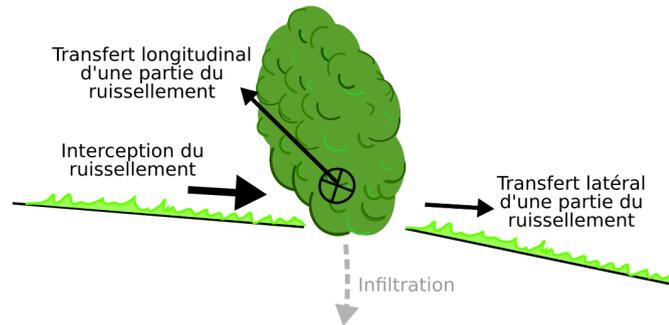


FIGURE 47 – Devenir du ruissellement intercepté par une haie/talus

## 11.3 Représentation du ruissellement dans PESH-MELBA

### 11.3.1 Contraintes et problématiques

Dans PESHMELBA, le module de ruissellement doit répondre aux attentes suivantes :

- Représenter les écoulements de surface à l'échelle du bassin versant ;
- S'appuyer sur un maillage à l'échelle de l'élément (on rappelle que l'infiltration est modélisée à l'échelle de la parcelle comme une unique colonne, avec un maillage seulement vertical) ;
- Prendre en compte explicitement le rôle d'interception de l'écoulement des linéaires ;
- Être robuste et efficace en terme de temps de calcul (pour répondre aux objectifs opérationnels d'un tel outil).

De plus, alors que les transferts de subsurface sont représentés dans PESHMELBA à un pas de temps assez grossier (par exemple, 1h en période sèche, 30 min en période de pluie ou de récession de nappe), les transferts de surface devront être représentés à un pas de temps plus fin afin de ne pas "ralentir" artificiellement cette composante du modèle. En effet, à chaque pas de temps l'eau ruisselant qui sort d'un élément ne pourra être transmise qu'à son(ses) voisin(s) direct(s). Il faudra attendre le pas de temps de ruissellement suivant pour transmettre ce volume d'eau aux éléments "voisins".

Ce dernier point concernant les différences de pas de temps entre modélisation des écoulements de surface et subsurface est particulièrement délicat et aucune approche de la sorte n'a pu être trouvée dans la littérature (hormis pour le modèle SACADEAU qui détermine le ruissellement à un pas de temps horaire puis en déduit l'infiltration à un pas de temps journalier comme la différence entre le cumul de ce qui arrive sur une parcelle et ce qui en sort).

### 11.3.2 Choix adoptés

Les choix détaillés ci-dessous sont valables pour l'instant pour tous les éléments surfaciques (parcelles et bandes enherbées). A terme, il n'est pas exclu de changer cette représentation pour type d'élément surfacique pour lequel des processus particuliers mériteraient d'être pris en compte en surface.

#### Aspect spatial

Au vu du maillage utilisé pour la modélisation de l'infiltration (1 parcelle = 1 maille horizontale) et des informations dont on dispose concernant la topographie (homogène à l'échelle de la parcelle), on choisit de garder le même maillage en surface qu'en subsurface. Ce choix aboutira forcément à une représentation très simplifiée du ruissellement relevant plus du routage de surface mais permet de contourner certaines difficultés majeures, notamment celle du couplage surface-subsurface particulièrement délicat dans le cas où les processus sont modélisés sur des maillages différents. De plus, on prend ainsi le parti de conserver l'aspect modélisation "objet" de PESHMELBA en calculant le ruissellement centre à centre.

#### Aspect temporel

Une solution naturelle pour représenter la différence de cinétique entre les processus d'infiltration et de ruissellement est d'adopter un pas de temps spécifique, suffisamment fin pour représenter les écoulements de surface. Cependant, pour rester cohérent avec la résolution spatiale présentée ci-dessus, il semble judicieux de ne pas trop raffiner ce pas de temps.

On rappelle d'autre part que le ruissellement est calculé entre un élément et son(s) voisin(s) direct(s) à chaque sous pas de temps. Cela signifie que le choix de ce sous pas de temps influencera grandement la vitesse de ruissellement à l'échelle du bassin versant. Celui-ci pourrait être par exemple un indicateur de la concentration des écoulements qui existe sur le bassin versant mais devra dans tous les cas faire l'objet d'un travail de calage permettant de le régler en fonction de la résolution spatiale des éléments du bassin, de la topographie, etc...

#### Couplage avec la subsurface

A la fin d'un pas de temps, on dispose avec le module FRER1D d'une information précise concernant la quantité d'eau en surface n'ayant pas réussi à s'infiltrer (ruissellement hortonien ou par saturation) ou s'étant exfiltrée. Cette quantité d'eau est alors disponible pour le ruissellement et peut être utilisée pour représenter les écoulements de surface ayant eu lieu au cours du pas de temps juste achevé, comme représenté par

le chronogramme de la fig. 48 : le ruissellement est calculé sur un pas de temps pour déterminer une nouvelle répartition des volumes d'eau en surface sur l'ensemble des parcelles ; ces volumes étant alors disponibles pour l'infiltration au pas de temps suivant.

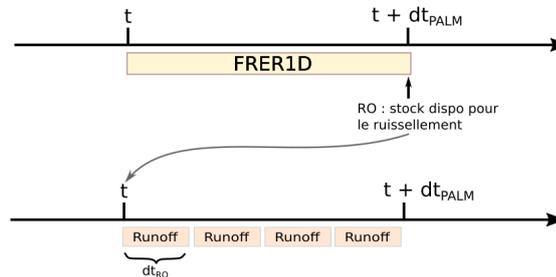


FIGURE 48 – Couplage surface-subsurface dans PESHMELBA.  $dt_{PALM}$  désigne le pas de temps de base du modèle et  $dt_{RO}$  désigne le sous pas de temps de ruissellement.

De plus, au lieu de transférer la totalité du stock de ruissellement (RO) en début de pas de temps comme indiqué sur la fig. 48, on peut imaginer transférer à chaque sous pas de temps ( $dt_{RO}$ ) un certain pourcentage de ce stock. Par exemple, si un pas de temps  $dt_{PALM}$  se décompose en 4 sous pas de temps  $dt_{RO}$ , on transfère à chaque sous pas de temps 25% du stock RO. Cette solution, illustrée fig. 49 permettrait d'éviter de trop grosses discontinuités mais aussi de déplacer trop tôt de la pluie qui ne serait pas encore tombée.

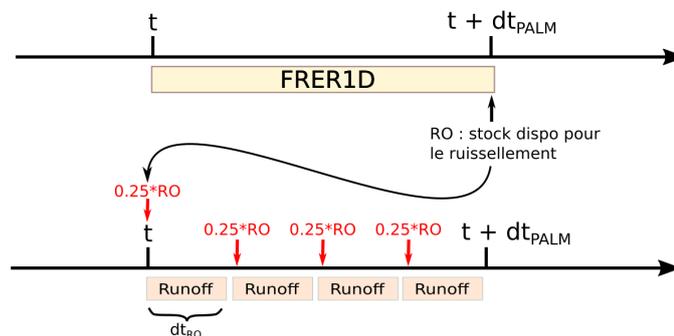


FIGURE 49 – Exemple de partage du stock disponible pour le ruissellement

Le module de ruissellement doit pouvoir intégrer les deux options ("one at a time" vs "equipartition") pour pouvoir au moins les tester et les comparer.

### Représentation du ruissellement

Le ruissellement est représenté par l'équation de l'onde cinématique dont la vitesse est estimée par la formule de Manning-Strickler. Le débit sortant [ $L^3T^{-1}$ ] d'un élément est donné par la formule 11.1 :

$$q = \frac{Ah^{2/3}}{n} \sqrt{S_0} \quad (11.1)$$

où A [ $L^2$ ] est la section transversale de la surface de l'écoulement (ou section mouillée).

La détermination de la section transversale n'est pas évidente puisqu'elle nécessite

de connaître la largeur sur laquelle se fait l'écoulement et donc la direction d'écoulement principale. Or, la formule de Manning-Strickler nous permet d'obtenir un débit ruisselé mais aucune information n'est donnée sur la direction. Cette information n'étant pas transmise par GeoMelba, il nous faut donc l'estimer à partir des données disponibles (altitudes et distances entre les parcelles, pente moyenne sur un élément, position des centroïdes et des interfaces...). Peu d'informations sont disponibles pour caractériser la géométrie d'une parcelle (seulement une pente moyenne et une surface) et ces caractéristiques peuvent être très variables d'une parcelle à l'autre. On fait donc le choix de déterminer la largeur d'écoulement en estimant d'abord une longueur caractéristique de ruissellement. On remontera ensuite à la largeur d'écoulement en divisant la surface de la parcelle par cette grandeur.

Une parcelle peut recevoir du ruissellement de plusieurs parcelles amont et peut également posséder plusieurs exutoires susceptibles de correspondre à plusieurs directions de ruissellement. Ainsi, pour chaque parcelle, on estime une longueur caractéristique de ruissellement en cherchant le voisin amont et aval (surfacique ou linéaire) associé à la plus forte pente.<sup>1</sup> La longueur d'écoulement sur la parcelle est alors calculée comme la somme d'une longueur amont ; longueur entre le centroïde de la parcelle et le centre de l'interface avec le voisin responsable de la plus forte pente entre éléments, et d'une longueur aval calculée sur le même schéma. Le calcul est illustré fig. 50.

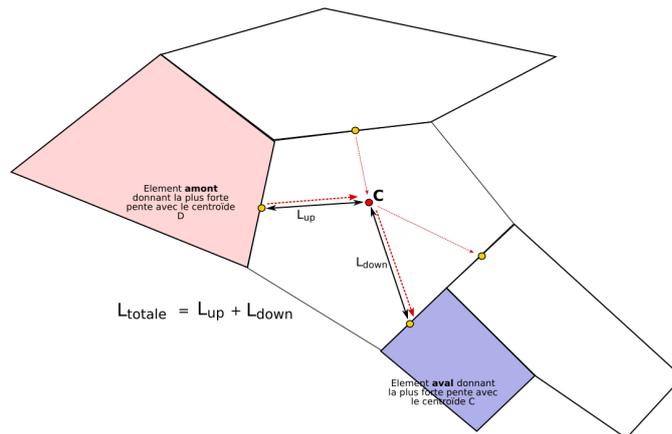


FIGURE 50 – Estimation d'une longueur caractéristique de ruissellement

Obs : en réalité, pour déterminer précisément la direction principale d'écoulement à l'échelle d'un élément il faudrait se baser sur les données d'un MNT mais aussi considérer le travail du sol et le sens du labour qui influencent les directions de ruissellement sur les bassins agricoles aux pentes faibles. Dans un première version de PESHMELBA on ne prend pas en compte ces composantes mais le script de ruissellement devra rester suffisamment modulaire pour se laisser la possibilité de les intégrer par la suite.

1. La pente est calculée entre le centroïde de la parcelle et le centroïde de l'interface avec chaque voisin. Ceci assure que la direction de ruissellement identifiée ne dépend que des caractéristiques de la parcelle. On peut alors comparer des scénarios où les caractéristiques des éléments aval changent sans influencer le ruissellement sortant de la parcelle.

Enfin, la formule de Manning-Strickler permet de déterminer un débit sortant mais celui-ci devra être réparti si la parcelle possède en réalité plusieurs exutoires. La détermination d'une unique direction d'écoulement amène naturellement l'hypothèse que l'on peut ne considérer qu'un seul exutoire en surface pour chaque parcelle. Cependant, cette hypothèse est trop réductrice dans des configurations où les pentes sont du même ordre de grandeur pour tous les éléments aval (parcelle aval coupée en deux par exemple) ou lorsque ce sont plutôt les longueurs d'interface qui sont discriminantes pour la répartition de l'écoulement sortant. Ainsi, il est plus cohérent d'autoriser les configurations multisorties pour lesquelles il faut définir un système de pondération pour assurer la répartition du débit sortant.

Deux facteurs principaux influençant la répartition du ruissellement ont été identifiés et constitueront le système de pondération mis en place. Il s'agit du gradient d'altitude entre le centroïde de la parcelle et celui de son voisin aval (pente  $s$ ) et la longueur d'interface entre les deux ( $L_{int}$ ). Ces deux facteurs peuvent alors intervenir dans l'expression d'un coefficient de pondération normalisé. Dans un premier temps, on choisit une expression basique pour ces coefficients  $\alpha^i$  [-] :

$$\alpha^i = \frac{s^i (L_{int}^i)^{\frac{2}{3}}}{\sum_{j=0}^N s^j (L_{int}^j)^{\frac{2}{3}}} \quad (11.2)$$

où  $N$  correspond au nombre d'éléments situés à l'aval d'une parcelle donnée  $i$ .

Sur la Figure 51 et dans les tables 11.1 et 11.2 est présenté un exemple de répartition du ruissellement sortant d'un élément UH1 vers trois éléments UH2, UH3 et Lin11. Les coefficients  $\alpha$  correspondent aux coefficients de répartition spatiale du ruissellement décrits ci-dessus.

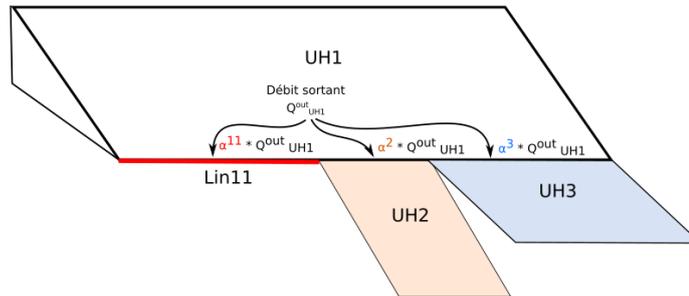


FIGURE 51 – Exemple de configuration et calcul des débits associés

Index de l'élément	Volume sortant	Volume entrant
UH1	$(\alpha^{11} + \alpha^2 + \alpha^3) dt_{PALM} Q_{UH1}^{out}$	0
UH2	0	$\alpha^2 dt_{PALM} Q_{UH1}^{out}$
UH3	0	$\alpha^3 dt_{PALM} Q_{UH1}^{out}$

TABLE 11.1 – Structure de la table surface\_in\_out associée à la configuration de la fig. 51

Index de l'élément	Volume entrant
Lin1	$\alpha^{11} dt_{RO} * Q_{UH1}^{out}$

TABLE 11.2 – Structure de la table `linear_in_out` associée à la configuration de la fig. 51. Cette table étant transmise à chaque sous pas de temps, le volume indiqué correspond bien au sous pas de temps  $dt_{RO}$ .

### 11.3.3 Transfert de solutés en surface, couplage surface-subsurface

On rappelle que l'infiltration des solutés disponibles dans une hauteur d'eau en surface vers la colonne de sol se fait dans le module FRER1D. La couche de mélange permet aussi de remobiliser des solutés disponibles dans les premiers cm de sol (section 7.2.9). Ensuite, dans le module de ruissellement, les solutés sont advectés avec les écoulements de surface. Il n'y a pas de représentation de diffusion ce qui pourra conduire à de fortes discontinuités de concentration entre les différents éléments. Puis, en fin de chaque pas de temps de ruissellement, l'équilibre d'adsorption est calculé par un isotherme linéaire et Freundlich et la quantité de solutés dégradés pendant le pas de temps est déterminée à partir du stock de solutés dissous disponible.

### 11.3.4 Fonctionnement d'une haie/talus

Les choix concernant la représentation du ruissellement sur des structures de type haie/talus sont les mêmes que ceux faits pour les éléments surfaciques : l'eau disponible pour le ruissellement sur une haie/talus en début de pas de temps correspond à l'eau qui n'a pas pu s'infiltrer ou qui s'est exfiltrée pendant ce même pas de temps. On rappelle en effet que la subsurface d'une haie/talus est modélisée par 2 colonnes de sol, amont et aval, sur lesquelles peut être appliqué FRER1D. Le stock de ruissellement initial peut aussi être réparti temporellement de la même manière que pour les surfaciques (Figure 49). De plus, la question de la largeur sur laquelle le ruissellement se transmet à l'aval en cas de transfert multi-exutoire ne se pose pas car un élément de type linéaire ne peut avoir qu'un seul exutoire en latéral. La section transversale intervenant dans la formule de Manning est calculée à partir de la **longueur totale** de l'élément et il n'y a pas besoin de mettre en place des coefficients de répartition.

Par contre, lorsque du ruissellement est reçu dans la partie supérieure de la haie il faut établir une chronologie pour la gestion du stock d'eau et de solutés afin d'effectuer la répartition évoquée section 11.2.2. Une partie doit être réservée pour l'infiltration préférentielle, une est redirigée dans la direction longitudinale le long du réseau de linéaires et le stock d'eau restant peut potentiellement traverser la structure. Pour cela, la représentation d'une haie/talus sous la forme de deux réservoirs en surface a été adoptée (Figure 6). Elle permet de gérer plus facilement les transferts longitudinaux en amont de la haie et le transfert potentiel de ruissellement de la partie amont vers la partie inférieure aval de la haie. Les opérations réalisées pendant un pas de temps  $dt_{RO}$  sont les suivantes :

- Une portion du ruissellement disponible après le calcul de l'infiltration par FRER1D est affectée aux réservoirs de surface amont et aval de la haie (pour la partie aval, il s'agit d'eau ne s'étant pas infiltrée ou ruisselée depuis la partie amont de la haie).

- Dans la partie supérieure de la haie, le réservoir destiné à l'infiltration préférentielle est rempli avec un volume déterminé à partir de la conductivité préférentielle. Cette eau est retirée du réservoir de surface.
- Un flux de ruissellement et un volume associé  $W_{kw}$  sont déterminés à partir de la formule de Manning. ils sont divisés en une composante longitudinale et latérale à partir de l'orientation de la structure par rapport à la ligne de plus grande pente ( $\beta$ ). Le volume d'eau transmis en longitudinal au réservoir de surface supérieur de la haie connectée à l'aval est  $\cos(\beta) * W_{kw}$ .
- Si le volume d'eau restant dans le réservoir de surface supérieur de la haie courante dépasse la capacité du réservoir autorisée par la hauteur du "mur" séparant les 2 parties de la haie, le trop plein d'eau est transmis au réservoir de surface inférieur de la haie. Dans le cas d'une haie sans talus, c'est l'intégralité du volume d'eau qui est transmis au réservoir inférieur.
- Dans la partie inférieure de la haie, le réservoir destiné à l'infiltration préférentielle est rempli avec un volume déterminé à partir de la conductivité préférentielle. Cette eau est retirée du réservoir de surface.
- S'il reste de l'eau dans le réservoir de surface inférieur, la formule de Manning permet de déterminer le volume d'eau qui peut être transmis à l'élément connecté à l'aval.

Ces différentes actions sont résumées Figure 52.

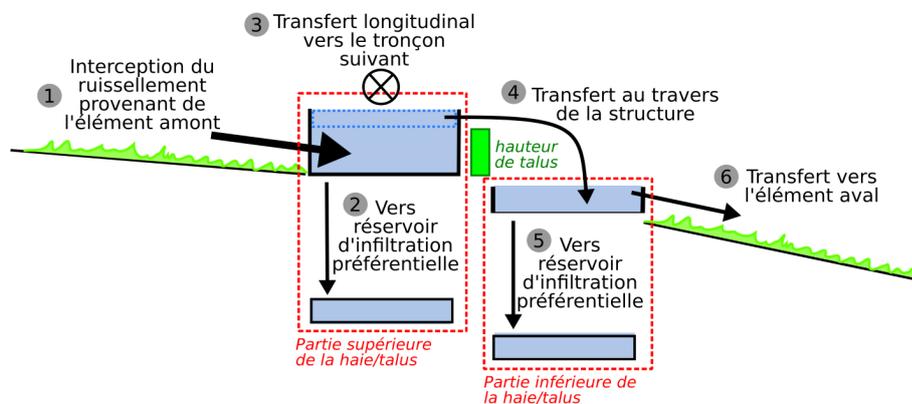


FIGURE 52 – Représentation des processus de transfert hydrique en surface d'une haie.

### 11.3.5 Connexion avec un élément de condition limite (BCS)

Les connexions potentielles d'un élément du bassin avec un élément de condition limite (BCS) doivent aussi être gérées pour les écoulements de surface. En surface, la condition limite donnée par un BCS à l'amont peut être un flux nul ou un flux constant défini par l'utilisateur. A ce flux d'eau est associé un flux de solutés calculé à partir du flux d'eau et d'une concentration donnée. A l'aval, un BCS sert seulement à récupérer le ruissellement (eau et solutés) de la parcelle amont. De tels éléments doivent être intégrés lors de la détermination des écoulements de surface entre les différents types d'éléments.

## 11.4 Implémentation dans PESHMELBA

De la même manière que pour les transferts latéraux de subsurface, les écoulements de surface d'un élément à l'autre sont représentés par un ensemble de **lignes d'échanges**. Selon le type d'élément considéré, différentes lignes d'échanges ont été implémentées sous forme de classes, toutes héritant de la classe `SurfaceInterface`. Il faut cependant noter, que contrairement à la subsurface, les lignes d'échanges de surface n'ont besoin de considérer que l'élément amont pour déterminer la composante de ruissellement. En effet, en se basant sur l'onde cinématique, on considère que le flux d'eau sortant d'un élément ne dépend que de l'état de cet élément sans considérer le niveau d'eau sur l'élément aval. Les classes implémentées sont donc conditionnées par ce choix et décrivent les lignes d'échanges par rapport à leur élément amont, sans distinguer les cas où les éléments aval sont de natures différentes. C'est-à-dire qu'une ligne d'échanges de surface parcelle->haie sera implémentée de la même manière qu'une ligne d'échanges parcelle->parcelle ou parcelle->fossé. Pour le moment, deux classes ont été implémentées : `HomogeneousSurfaceExchangeLine` et `TankExchangeLine`. Les éléments de condition limite de surface auraient aussi pu faire l'objet d'une interface à part mais leur traitement a été intégré aux autres types d'interface pour plus de commodité. Puisque chacune de ces classes hérite de la classe `SurfaceInterface`, elles se basent aussi sur son constructeur. Chaque instance de la classe `SurfaceInterface` est initialisée comme suit :

- `idx (int)` : un index pour repérer l'interface. En général, il s'agit de l'indice de l'élément situé à l'amont de la ligne d'échanges ;
- `x,y (double)` : une longitude et une latitude [L] ;
- `z (double)` : une altitude [L] ;
- `main_inst_idx (int)` : l'indice de l'élément principal composant l'interface. Cela correspond à celui situé à l'amont de la ligne d'échanges.
- `weight_runoff` : un dictionnaire contenant pour chaque élément situé à l'aval de la ligne d'échanges la proportion de ruissellement qui lui est affectée comme décrit section 11.3.2 ;
- `downsurface_idx (vect(int))` : les indices des éléments de type `Surface` qui se situent à l'aval de la ligne d'échanges ;
- `downtank_idx (vect(int))` : les indices des éléments de type `Tank` qui se situent à l'aval de la ligne d'échanges ;
- `downreach_idx (vect(int))` : les indices des éléments de type `Reach` qui se situent à l'aval de la ligne d'échanges ;
- `downBCS_idx (vect(int))` : les indices des éléments de type `BCS` qui se situent à l'aval de la ligne d'échanges ;
- `upBCS_idx (vect(int))` : les indices des éléments de type `BCS` qui se situent à l'amont de la ligne d'échanges.

Comme pour les lignes d'échanges de subsurface, la classe `SurfaceInterface` contient aussi des méthodes abstraites (`update_state_variables()`, `calculate_runoff_flux()`, `flux_spreading()` et `ponding_bcs_update()`). On rappelle qu'il s'agit de méthodes que chaque sous classe doit aussi contenir et décrire explicitement car on attend un comportement différent selon le type d'interface manipulé. Dans la classe `SubsurfaceInterface`, ces méthodes abstraites jettent seulement l'exception `NotImplementedError`.

### 11.4.1 classe `HomogeneousSurface`

La classe `HomogeneousSurface` décrit un objet de surface possédant une aire (c'est-à-dire, pas un linéaire ni un objet ponctuel). Il décrit donc par exemple la partie de surface d'une parcelle ou d'une bande enherbée. C'est une sous classe de `HomogeneousUnit` qui est initialisée à partir des attributs suivants :

- `idx` (double) : index de l'élément
- `x,y` (double) : latitude et longitude de l'élément [L]
- `z` (double) : altitude de l'élément [L]
- `area` (double) : surface de l'élément [L<sup>2</sup>]
- `manning` (double) : coefficient de Manning [LT<sup>-1/3</sup>]
- `Loff` (double) : longueur caractéristique de ruissellement [L]
- `hpond` (double) : hauteur de ponding autorisée sur la surface avant de déclencher le ruissellement [L]
- `adsorp_thick` (double) : hauteur de la bande de sol sur laquelle est autorisée l'adsorption [L]. On rappelle qu'il n'y a pas de correspondance avec la hauteur de la couche de mélange.
- `bd` (double) : masse volumique apparente en surface [M.L<sup>-3</sup>]
- `ths` (double) : teneur en eau à saturation en surface [L<sup>3</sup>.L<sup>-3</sup>]
- `isotype` (string(1 :n\_solute)) : type d'isotherme à utiliser pour calculer l'adsorption pour chaque soluté
- `kads` (double(1 :n\_solute)) : constante d'adsorption pour chaque soluté
- `nads` (double(1 :n\_solute)) : constante d'adsorption pour chaque soluté

Ses méthodes permettent principalement la mise à jour des variables d'état mais aussi le calcul de l'adsorption :

- **`calculate_adsorption()`**

Pour calculer l'adsorption en surface, il faut d'abord définir quelle portion d'eau est en contact avec la hauteur de terre où est autorisée l'adsorption `adsorp_thick`. Dans cette bande, on considère alors un mélange complet, c'est-à-dire que le sol/substrat peut être considéré comme en suspension dans l'eau, autorisant une adsorption complète et instantanée. On suppose aussi que les solutés sont uniformément repartis dans tous le volume d'eau pour calculer la masse de solutés en contact avec le sol et potentiellement adsorbables. La méthode `calculate_adsorption` permet donc d'abord le calcul du volume d'eau dans lequel l'adsorption peut avoir lieu `volume_soil`, puis la masse de solutés contenue dans ce volume `mass_soil` (voir Figure 53). On suppose qu'à l'état initial tous les solutés sont en solution et que la masse adsorbée est nulle.

La fonction externe `adsorption` est alors appelée pour chaque soluté déterminant le nouvel équilibre d'adsorption à partir des constantes d'adsorption et de l'isotherme spécifié en initialisation de l'objet `HomogeneousSurface`.

A partir de ce nouvel équilibre, les attributs contenant la concentration de solutés, les masses adsorbée et dissoute sont mis à jour.

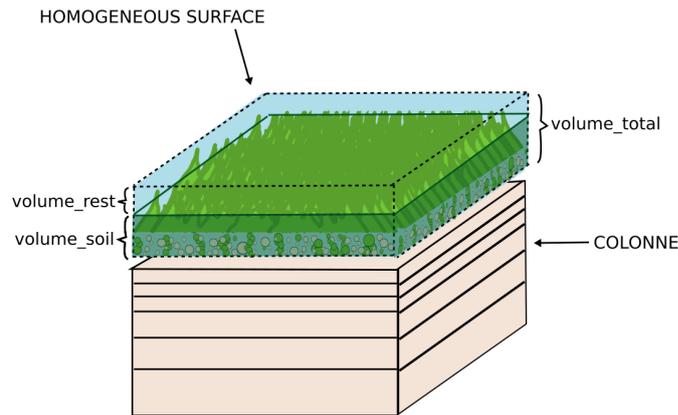


FIGURE 53 – Bande de sol disponible pour l’adsorption sur l’objet Homogeneous Surface

- **calculate\_degradation()**

La méthode `calculate_degradation` permet de déterminer pour chaque soluté, la quantité dégradée sur l’élément en un pas de temps. Une partie du stock de solutés dégradés peut être attribué au stock de métabolite si un indice de métabolite a été donné. Cette masse dépend aussi du taux de transformation en métabolites donnée en entrée.

- **update\_state\_variables()**

Cette méthode permet la mise à jour de l’état d’un élément `HomogeneousSurface` en important depuis PALM les variables suivantes :

- `wheight_surface` : le volume d’eau disponible sur l’élément [ $L^3$ ];
- `wroff_cumul_surface` : le volume d’eau reçu/envoyé en cumulé sur l’élément [ $L^3$ ];
- `scumul_surface` : la masse de solutés reçue/envoyée en cumulé sur l’élément [M];
- `soluteconc_surface` : la concentration en solutés [ $M.L^{-3}$ ];
- `smass_ads`, `smass_diss`, `smass_total` : les masses adsorbée, dissoute et totale en surface de l’élément [M].

- **runoff\_update()**

Cette méthode permet la mise à jour du volume d’eau et de la masse de solutés en surface. En effet, les attributs `surface_waterlatfluxIN_dtRO`, `surface_waterlatfluxOUT_dtRO`, `surface_solutelatfluxIN_dtRO` et `surface_solutelatfluxOUT_dtRO` permettent de stocker pour un pas de temps de ruissellement `dt_RO` des volumes d’eau/masses de solutés provenant de différents éléments amonts ou envoyés vers différents éléments aval sans toucher aux variables d’état `pond_volume` et `smass_diss`. Ces variables doivent être explicitement actualisées pour intégrer ces contributions. Cette méthode permet aussi d’incrémenter à chaque pas de temps `dt_RO` les variables `runofftransfer` et `solute_runofftransfer` qui contiennent le volume d’eau et la masse de solutés reçus algébriquement par l’élément en un pas de temps `dt_PALM`.

### 11.4.2 Classe SurfaceTank

La classe SurfaceTank hérite de la classe ElementaryLinear. Elle permet de définir un réservoir de surface virtuel qui peut contenir un volume d'eau, une masse de solutés et dans lequel les solutés peuvent être adsorbés et dégradés. En plus des arguments du constructeur de la classe ElementaryLinear, le constructeur de SurfaceTank initialise ses instances à partir de :

- area (double) : surface (conceptuellement virtuelle car il s'agit d'un linéaire) [ $L^2$ ];
- slope (double) : pente transversale sur l'élément [ $L^1.L^{-1}$ ];
- DT50 (double(1 :n\_solute)) : temps de demi-vie pour chaque solutés simulé sur le linéaire [ $T^{-1}$ ];
- adsorp\_thick (double) : épaisseur de sol sur laquelle peut être calculée l'adsorption des solutés [ $L$ ];
- bd (double) : masse volumique apparente du sol [ $M.L^{-3}$ ];
- ths (double) : teneur en eau à saturation dans le sol du réservoir [ $L^3.L^{-3}$ ];
- Kads (double(1 :n\_solute)) : constante d'adsorption;
- nads (double(1 :n\_solute)) : constante d'adsorption;
- isotype (string(1 :n\_solute)) : type d'isotherme appliqué pour déterminer l'équilibre d'adsorption.

La classe SurfaceTank contient aussi la méthode calculate\_adsorption qui permet de déterminer l'équilibre d'adsorption sur l'élément. Cette méthode fonctionne de la même manière que sur l'ensemble des autres éléments : détermination du volume d'eau et de la masse de solutés en contact avec la bande d'adsorption, application d'un isotherme linéaire ou de Freundlich sur cette bande et mise à jour des quantités adsorbée et dissoute dans l'ensemble du volume d'eau disponible. On se reporte à la section 11.4.1 pour plus de détails.

Enfin, la méthode calculate\_degradation permet de déterminer la quantité de solutés dégradés sur l'élément en un pas de temps. On rappelle qu'une partie du stock de solutés dégradés peut être attribué au stock de métabolite si un indice de métabolite a été donné. Cette masse dépend aussi du taux de transformation en métabolites donnée en entrée.

### 11.4.3 classe HedgeSurface

Contrairement aux surfaciques pour lesquels une seule classe, HomogeneousSurface, décrit à la fois la surface de parcelles ou de bandes enherbées ou de tout autre élément surfacique, une classe spéciale a été créée ici pour décrire la surface de linéaires de type haie, talus ou haie sur talus. Ceci car la représentation choisie des haies/talus en 2 parties est propre à son fonctionnement hydrologique et qu'il ne serait pas forcément judicieux de la transposer à tous les types de linéaires. Une instance de la classe HedgeSurface hérite de la classe Linear et se compose de 2 réservoirs de surface de type SurfaceTank. Elle est initialisée à partir des attributs suivants :

- idx (double) : index de l'élément;
- x,y (double) : latitude et longitude de l'élément [ $L$ ];
- z (double) : altitude du milieu de la haie [ $L$ ];

- length (double) : longueur du linéaire [L];
- width (double) : largeur totale du linéaire [L];
- slope (double) : pente longitudinale moyenne sur le linéaire (deg);
- alpha (double) : orientation de la haie en longitudinal par rapport à la ligne de niveau (deg);
- manning (double) : coefficient de Manning  $[LT^{-1/3}]$ ;
- height (double) : hauteur du talus séparant les parties supérieure et inférieure de la haie [L];
- macroporous\_infiltration\_rate (double) : conductivité à saturation régissant la vitesse d'infiltration préférentielle  $[L.T^{-1}]$ ;
- UpTank : instance de la classe SurfaceTank représentant le réservoir de surface **amont** de la haie;
- DownTank : instance de la classe SurfaceTank représentant le réservoir de surface **aval** de la haie;

On note que hormis les paramètres liés aux instances de SurfaceTank (c'est à dire leur largeur et les paramètres nécessaires au calcul de l'adsorption), tous les autres paramètres décrivant la haie sont les mêmes pour les parties supérieures et inférieures.

- **calculate\_adsorption()**

La méthode calculate\_adsorption permet de déterminer l'équilibre d'adsorption sur les parties amont et aval de la haie. Pour cela, elle appelle successivement les méthodes calculate\_adsorption des instances UpTank et DownTank de la classe SurfaceTank.

- **calculate\_degradation()**

La méthode calculate\_degradation permet de déterminer la quantité de solutés dégradée sur les parties amont et aval de la haie. Pour cela, elle appelle successivement les méthodes calculate\_degradation des instances UpTank et DownTank de la classe SurfaceTank.

- **runoff\_update()**

Cette méthode permet seulement la mise à jour du volume d'eau et de la masse de solutés dans les 2 réservoirs composant la haie/talus à partir des volumes reçus ou envoyés aux éléments voisins. Les attributs surface\_waterlatfluxIN\_dtRO, surface\_waterlatfluxOUT\_dtRO, surface\_solutelatfluxIN\_dtRO et surface\_solutelatfluxOUT\_dtRO permettent de stocker pour un pas de temps de ruissellement dt\_RO des volumes d'eau/masses de solutés provenant de différents éléments amonts ou envoyé vers un potentiel élément aval sans toucher aux variables d'état. Les variables d'état décrivant les niveaux d'eau dans la haie sont ainsi mises à jour : les contributions reçues des éléments amont surface\_waterlatfluxIN\_dtRO et surface\_solutelatfluxIN\_dtRO sont ajoutées dans le réservoir de surface de la partie **supérieure** de la haie et les contributions envoyées vers l'élément aval surface\_waterlatfluxIN\_dtRO et surface\_solutelatfluxIN\_dtRO sont retirées au réservoir de surface de la partie **inférieure** de la haie.

### 11.4.4 classe `SurfaceInterface`

Toutes les classes de lignes d'échanges de surface héritent de la classe `SurfaceInterface`. Celle-ci est principalement composée d'un constructeur qui permet d'initialiser une instance à partir des attributs suivants :

- `x,y` : une longitude et une latitude [L] ;
- `main_inst_idx` : un index (scalaire entier) pour repérer la ligne d'échanges. Il s'agit de l'indice de l'élément situé à l'amont de la ligne d'échanges ;
- `weight_runoff` : l'ensemble des poids de ruissellement pour tous les éléments situés à l'aval de la ligne d'échanges. Il s'agit d'un dictionnaire qui contient pour clés les indices des éléments connectés et pour valeurs les poids correspondant à chaque indice.
- `downsurface_idx` : un vecteur contenant les indices des éléments de type surface connectés à l'aval ;
- `downtank_idx` : un vecteur contenant les indices des éléments de type réservoirs de surface connectés à l'aval ;
- `downreach_idx` : un vecteur contenant les indices des éléments de type tronçons connectés à l'aval ;
- `downBCS_idx` : un vecteur contenant les indices des éléments de type BCS connectés à l'aval ;
- `upBCS_idx` : un vecteur contenant les indices des éléments de type BCS connectés à l'amont ;
- `nb_connect_downstream` : le nombre de connexions avec des éléments à l'aval.

La classe `SurfaceInterface` contient aussi des méthodes abstraites `update_state_variables`, `ponding_bcs_update`, `calculate_runoff_flux` et `flux_spreading`. Ces méthodes sont aussi présentes dans chaque classe qui hérite de `SurfaceInterface` et sont décrites explicitement dans ces classes car on attend un comportement différent selon le type de ligne d'échanges manipulé. Dans la classe `SurfaceInterface`, ces méthodes jettent seulement l'exception `NotImplementedError`. Comme pour les échanges de subsurface, l'intérêt d'une telle méthode est illustré dans le module `EXCHANGES_SURFACE` : dans la fonction `use_EXCHANGES_SURFACE` on appelle pour chaque objet impliqué dans des transferts de surface ces trois méthodes. Selon la nature de l'objet manipulé, les actions effectuées par ces méthodes seront différentes.

### 11.4.5 classe `HomogeneousSurfaceExchangeLine`

- **constructeur** `__init__()`

Le constructeur de la classe `HomogeneousSurfaceInterface` permet d'initialiser une instance avec les mêmes attributs que ceux de la classe `SurfaceInterface` :

- `idx` (int) : un index pour repérer l'interface. En général, il s'agit de l'indice de l'élément situé à l'amont de l'interface ;
- `x,y` (double) : une longitude et une latitude [L] ;
- `z` (double) : une altitude [L] ;

- `main_inst_idx` (int) : l'indice de l'élément principal composant l'interface. Cela correspond à celui situé à l'amont de l'interface. Il s'agit ici d'un élément de la classe `HomogeneousSurface`.
  - `weight_runoff` : un dictionnaire contenant pour chaque élément situé à l'aval de l'interface la proportion de ruissellement qui lui est affectée ;
  - `downsurface_idx` (vect(int)) : les indices des éléments de type `Surface` qui se situent à l'aval de l'interface ;
  - `downtank_idx` (vect(int)) : les indices des éléments de type `Tank` qui se situent à l'aval de l'interface ;
  - `downreach_idx` (vect(int)) : les indices des éléments de type `Reach` qui se situent à l'aval de l'interface ;
  - `downBCS_idx` (vect(int)) : les indices des éléments de type `BCS` qui se situent à l'aval de l'interface ;
  - `upBCS_idx` (vect(int)) : les indices des éléments de type `BCS` qui se situent à l'amont de l'interface.
- **`ponding_bcs_update()`**  
 Cette méthode permet une première mise à jour du volume d'eau sur l'élément `HomogeneousSurface`. Comme détaillé dans la section 11.3.2, en début de pas de temps le volume **`pondvolume`** disponible en surface d'un élément est initialisé à zéro. Toute l'eau effectivement disponible après la fin du calcul de `FRER1D` est stockée dans la variable **`runoffvolume`**. A chaque sous pas de temps de ruissellement `dt_RO`, une portion du volume d'eau contenu dans la variable **`runoffvolume`** est attribuée à la portion **`pondvolume`**. Cette répartition dépend des coefficients **`Wrunoff`** donnés en entrée de la méthode `ponding_bcs_update`. En plus de cette première initialisation, le volume d'eau **`pondvolume`** et la masse de solutés dissous en surface **`smass_diss`** sont mis à jour en prenant en compte la présence d'éléments de condition limite `BCS` en amont. Ceci n'est pas très rigoureux car les éléments `BCS` auraient du faire l'objet d'une ligne d'échanges séparée mais cela simplifiait l'écriture du bilan de masse. Il pourra être envisagé à terme de modifier cette partie pour faire en sorte que les transferts entre une `BCS` amont et une `HomogeneousSurface` fassent l'objet d'une interface à part entière.
  - **`calculate_runoff_flux()`**  
 La méthode `calculate_runoff_flux()` calcule la composante de ruissellement sortant à partir de la hauteur d'eau sur l'élément. Le ruissellement n'est effectivement calculé que sur le volume d'eau situé au dessus de la hauteur limite de `ponding` **`hpond`**. Le flux puis le volume d'eau correspondant sont déterminés à partir de la formule de Manning pour une hauteur d'eau re-évaluée en soustrayant la hauteur **`hpond`**. On utilise une largeur de ruissellement estimée `width_runoff` calculée à partir de l'aire de l'élément et de sa longueur de ruissellement caractéristique :  $width_{runoff} = \frac{surface}{L_{off}}$ . La méthode retourne à la fois le flux `Q_kw` et le volume correspondant `W_kw` pour un pas de temps donné en entrée.
  - **`flux_spreading()`**  
 La méthode `flux_spreading` permet de répartir un volume d'eau `W_kw` donné en entrée et les solutés qu'il contient entre plusieurs éléments situés à l'aval. La répartition se fait entre tous les éléments dont les indices sont données dans les

vecteurs `downsurface_idx`, `downreach_idx`, `downBCS_idx` et `downtank_idx` en se basant sur les poids contenus dans le dictionnaire `weight_runoff`. Pour chacun de ces éléments, l'instance correspondante dans le dictionnaire `Element_dic` est mise à jour.

### 11.4.6 classe `TankExchangeLine`

- **constructeur `__init__()`**

Le constructeur de la classe `TankExchangeLine` permet d'initialiser une instance avec les mêmes attributs que ceux de la classe `SurfaceInterface` :

- `idx (int)` : un index pour repérer l'interface. En général, il s'agit de l'indice de l'élément situé à l'amont de l'interface ;
- `x,y (double)` : une longitude et une latitude [L] ;
- `z (double)` : une altitude [L] ;
- `main_inst_idx (int)` : l'indice de l'élément principal composant l'interface. Cela correspond à celui situé à l'amont de l'interface. Il s'agit ici d'un élément de la classe `HedgeSurface`.
- `weight_runoff` : un dictionnaire contenant pour chaque élément situé à l'aval de l'interface la proportion de ruissellement qui lui est affectée. Dans le cas d'une interface `HedgeInterface`, le dictionnaire contient au maximum un élément affecté du poids 1 car un élément linéaire peut être connecté au plus à un élément aval en latéral.
- `downsurface_idx (vect(int))` : l'indice de l'élément de type `Surface` qui se situe potentiellement à l'aval de la ligne d'échanges ;
- `downtank_idx (vect(int))` : l'indice de l'élément de type `Tank` qui se situe potentiellement à l'aval de l'interface. Le ruissellement peut ainsi être routé vers les éléments connectés latéralement mais aussi longitudinalement ;
- `downreach_idx (vect(int))` : l'indice de l'élément de type `Reach` qui se situe potentiellement à l'aval de la ligne d'échanges ;
- `downBCS_idx (vect(int))` : l'indice de l'élément de type `BCS` qui se situe potentiellement à l'aval de la ligne d'échanges ;
- `upBCS_idx (vect(int))` : l'indice de l'élément de type `BCS` qui se situe potentiellement à l'aval de la ligne d'échanges ;
- `longtank_idx` : les indices des éléments de type `Tank` qui se situent potentiellement à l'aval de l'élément longitudinalement.

- **`ponding_bcs_update()`**

Tout comme pour les interfaces `HomogeneousSurface`, cette méthode permet une première mise à jour du volume d'eau sur l'élément en affectant une portion du volume d'eau contenu dans la variable `runoffvolume` à la portion `pondvolume`. Cette affectation est faite pour les réservoirs de surface `UpTank` et `DownTank` en fonction des coefficients `Wrunoff` donnés en entrée de la méthode.

- **`calculate_runoff_flux()`**

La méthode `calculate_runoff_flux()` calcule la composante de ruissellement sortant de l'élément à partir d'une hauteur d'eau. Le calcul est fait pour les réservoirs

supérieur et inférieur et les variables renvoyées (flux et volume d'eau correspondant pour un pas de temps donné en entrée) sont des vecteurs dont la première composante est relative au réservoir supérieur et la seconde au réservoir inférieur.

- **flux\_spreading()**

La méthode `flux_spreading` gère la répartition du stock de ruissellement (eau et solutés) disponible en surface de la haie pendant un pas de temps de ruissellement `dt_PALM`. La chronologie des actions est celle décrite dans la section 11.3.4. Dans un premier temps, la partie supérieure de la haie est gérée en alimentant d'abord le réservoir d'infiltration préférentielle. Le volume d'eau à déplacer est calculé à partir de la conductivité d'infiltration préférentielle et du pas de temps `dt_RO`. La masse de solutés est déterminée à partir de la concentration dissoute. A chaque fois, un test est fait pour être sûr de ne pas déplacer plus d'eau ou de solutés que le stock disponible en surface.

Ensuite, un flux de ruissellement est à nouveau calculé avec la hauteur d'eau mise à jour dans le réservoir supérieure. Ce flux est partagé en une composante longitudinale et latérale. Un volume d'eau (resp. une masse de solutés) est alors affectée à la variable `surface_waterlatfluxIN_dtRO` (resp. `surface_solutelatfluxIN_dtRO`) des réservoirs supérieurs des tronçons connectés à l'aval en longitudinal (indices contenus dans le vecteur `longtank_idx`) et le niveau d'eau est mis à jour pour le réservoir supérieur de l'élément courant.

Si le niveau d'eau encore disponible dans ce réservoir dépasse la hauteur du "mur" du talus, l'eau dépassant cette hauteur est déplacée du réservoir supérieur vers le réservoir inférieur et l'infiltration préférentielle est calculée sur la partie inférieure de la haie.

Enfin, s'il reste encore de l'eau dans le réservoir inférieur de la haie, le volume d'eau disponible pour ruisseler est mis à jour par l'appel à la fonction `calculate_runoff_flux` et le transfert est fait vers l'élément aval s'il existe.

## 11.5 Module EXCHANGES\_SURFACE

`EXCHANGES_SURFACE` est le script Python qui apparaît dans le schéma PALM pour gérer les transferts de surface. Il réutilise les classes détaillées dans ce chapitre pour décrire les écoulements de surface sur un bassin et leur interception par les différents éléments du paysage. Les différentes actions réalisées par ce module sont résumées Figure 54. Dans un souci d'optimisation de l'application finale, le script `EXCHANGE_SURFACE` ne contient que la partie d'initialisation des objets. La partie de calcul du ruissellement est faite par un script `use_EXCHANGES_SURFACE` qui est appelé à chaque pas de temps PALM.

### 11.5.1 Initialisation des instances représentant les éléments du paysage et des interfaces entre eux

Comme pour le module `EXCHANGES_SUBSURFACE`, `EXCHANGES_SURFACE` se base sur l'utilisation de dictionnaires pour regrouper des objets de natures différentes (de type `HomogeneousSurface`, `BCS`, `HedgeSurface` et `Reach`). Les paramètres nécessaires à l'initialisation de chacun de ces éléments sont chargés depuis des variables et objets

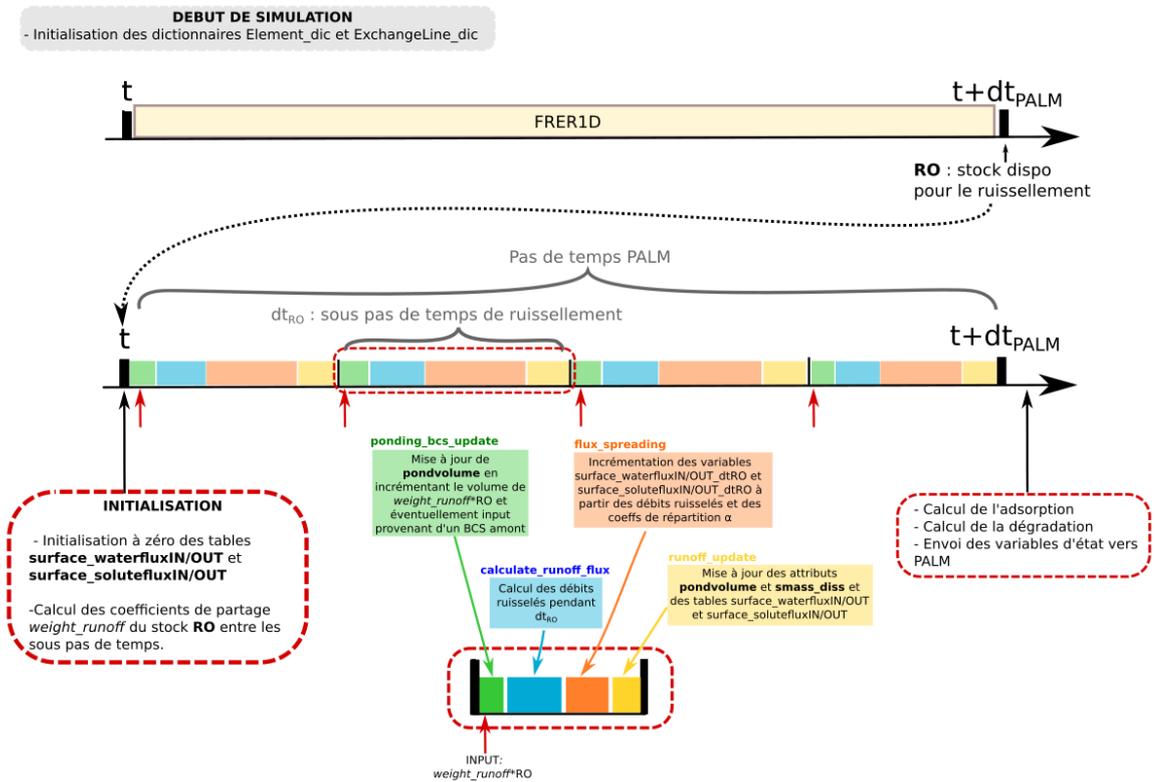


FIGURE 54 – Déroulement des étapes de calcul du script EXCHANGES\_SURFACE

PALM et le dictionnaire `Element_dic` est rempli avec tous les objets initialisés. Ensuite, pour chaque objet de type `HomogeneousUnit` ou `HedgeSurface`, une ligne d'échanges est initialisée et stockée dans le dictionnaire `ExchangeLine_dic`. Cette étape est réalisée par la fonction de service `initialize_interface_surface` donc le fonctionnement est à rapprocher de celui de la fonction `initialize_interface_subsurface` (description dans la section 10.7.4). Pour que le constructeur de la classe `SurfaceInterface` puisse être appelé, il faut posséder en entrée une description des connexions entre l'objet principal de l'interface et les éléments connectés à l'aval. C'est pour cela que les objets PALM `weight_runoff_HU`, `connect_to_reach_HU`, `connect_to_HU`, `connect_from_bcs` et `connect_to_bcs` et `connect_to_tanklat` sont chargés.

### 11.5.2 Calcul du ruissellement sortant et transfert vers les éléments voisins

Le module `EXCHANGE_SURFACE` appelle à chaque pas de temps PALM la fonction `use_EXCHANGE_SURFACE` qui prend en entrée les dictionnaires `Element_duc` et `ExchangeLine_dic` ainsi qu'une liste de tous les éléments contenus sur le bassin et des paramètres décrivant les solutés. Ainsi, à chaque pas de temps `use_EXCHANGES_SURFACE` charge le vecteur contenant tous les pas de temps `alltime_vect` et l'indice du pas de temps courant `time_index` puis les actions suivantes sont effectuées :

1. la fonction `get_weight` détermine des coefficients de "partage" pour les volumes d'eau qui seront appliqués en entrée d'un élément à chaque sous pas de temps. Ces

poinds sont regroupés sous la forme d'un vecteur `weight_runoff_temporal` pour être utilisés en entrée de la fonction `ponding_bcs_update`. Ce vecteur contient autant d'éléments que de sous pas de temps `dt_RO` composant le pas de temps `dt_PALM`.

Pour la première option "one at a time", on obtient le vecteur :

$$weight\_runoff\_temporal = \underbrace{[1 \quad 0 \quad 0 \quad \dots \quad 0]}_{N \text{ elements ; } N = \lfloor \frac{dt\_PALM}{dt\_RO} \rfloor + 1} \quad (11.3)$$

Et pour la deuxième "equirepartition" :

$$weight\_runoff\_temporal = \underbrace{\left[ \frac{dt\_RO}{dt\_PALM} \quad \dots \quad \frac{dt\_RO}{dt\_PALM} \right]}_{N-1} \quad 1 - (N-1) \frac{dt\_RO}{dt\_PALM} \quad (11.4)$$

2. les variables `surface_waterlatfluxIN`, `surface_waterlatfluxOUT`, `surface_solutelatfluxIN` et `surface_solutelatfluxIN` sont initialisées à zéro pour tous les éléments du dictionnaire `Element_dic`. Ces variables sont incrémentées à chaque pas de temps `dt_RO` pour stocker le volume d'eau et la masse de solutés entrant et sortant d'un élément donné durant un pas de temps `dt_PALM`. A la fin du pas de temps `PALM`, ces variables serviront à mettre à jour les volume d'eau et les masses de solutés dissous sur chaque élément.
3. Les variables d'état de chaque élément d'`Element_dic` sont mises à jour en appelant la méthode `update_state_variables`. Cette méthode a été explicitement définie pour les éléments de type `SurfaceHedge` ou `HomogeneousSurface`. Pour les objets `Reach`, la fonction est aussi implémentée dans la sous classe `Reach` mais ne comporte qu'une seule instruction "pass". En effet, pour le moment on considère qu'il n'y a pas d'écoulement de surface qui peut être généré à l'intérieur d'un tronçon et nous excluons donc d'éventuels débordements. Les éléments `Reach` ne sont inclus dans ce module que pour intercepter le ruissellement généré par les éléments amont.
4. Le dictionnaire `Element_dic` contenant les éléments mis à jour est associé explicitement à chacune des interfaces. Grâce au passage de variables par référence caractéristique du langage de Python, on s'assure qu'une modification à l'intérieur d'un interface donné d'un élément d'`Element_dic` sera aussi effective dans toutes les autres instances d'interface. Ceci assure une mise à jour continue des variables d'état pendant la simulation. Ainsi, à la fin du pas de temps `dt_PALM`, si on accède directement aux éléments du dictionnaire `Element_dic`, on s'assure qu'ils auront été bien mis à jour au travers de toutes les lignes d'échanges dans lesquelles il apparaît.
5. Une boucle `while t < tfinish_PALM` est lancée pendant un pas de temps `dt_PALM` et sera activée autant de fois que le ratio `dt_PALM/dt_RO` le permet. A l'intérieur de cette boucle, les actions suivantes sont réalisées :
  - l'initialisation à zero des variables `surface_waterlatfluxIN_dtRO`, `surface_waterlatfluxOUT_dtRO`, `surface_solutelatfluxIN_dtRO` et `surface_solutelatfluxIN_dtRO` ;
  - la mise à jour des volumes d'eau en surface des `HomogeneousSurface` et des `HedgeSurface` en allouant la part de `ponding` initialement présente et en

prenant en compte les potentielles contributions de BCS amont. Ceci est fait par l'appel de la méthode `ponding_bcs_update`.

- pour chaque interface, la détermination explicite du calcul du volume d'eau sortant en ruissellement ( $W_{kw}$ ) de chaque `HomogeneousSurface` et `HedgeSurface` par l'appel à la méthode `calculate_runoff_flux`.
- pour chaque interface, la répartition du volume d'eau  $W_{kw}$  et des solutés qu'il transporte entre les éléments situés à l'aval. Pour cela les variables `surface_waterlatfluxIN_dtRO`, `surface_waterlatfluxOUT_dtRO`, `surface_solutelatfluxIN_dtRO` et `surface_solutelatfluxIN_dtRO` sont incrémentées dans le dictionnaire `Element_dic` rattaché à chaque interface.
- pour chaque élément du bassin, la mise à jour du volume d'eau et de la masse de soluté disponible en surface à partir des variables `surface_waterlatfluxIN_dtRO`, `surface_waterlatfluxOUT_dtRO`, `surface_solutelatfluxIN_dtRO` et `surface_solutelatfluxIN_dtRO`.

### 11.5.3 Calcul de l'adsorption et envoi des variables vers PALM

A la fin du pas de temps  $dt_{PALM}$ , l'équilibre d'adsorption puis la quantité de solutés dégradés sont calculés sur tous les éléments de type `HomogeneousSurface` et `HedgeSurface` en appelant la méthode `calculate_adsorption` de leurs classes respectives. Puis les variables d'état décrivant le niveau d'eau et la répartition en solutés pour chaque élément du paysage sont renvoyées vers PALM.



## Troisième partie

### Mise en oeuvre du couplage : Implémentation dans OpenPALM



# Chapitre 12

## Introduction

Chacun des modules décrits dans la partie II peut être développé et validé de manière indépendante mais un outil doit permettre l'échange de variables entre les différents éléments, permettant ainsi le passage à l'échelle du bassin versant. Comme schématisé fig. 55, chaque module correspond à une brique élémentaire qu'il faut inclure au modèle complet et c'est le rôle du coupleur OpenPALM (Fouilloux and Piacentini, 1999).

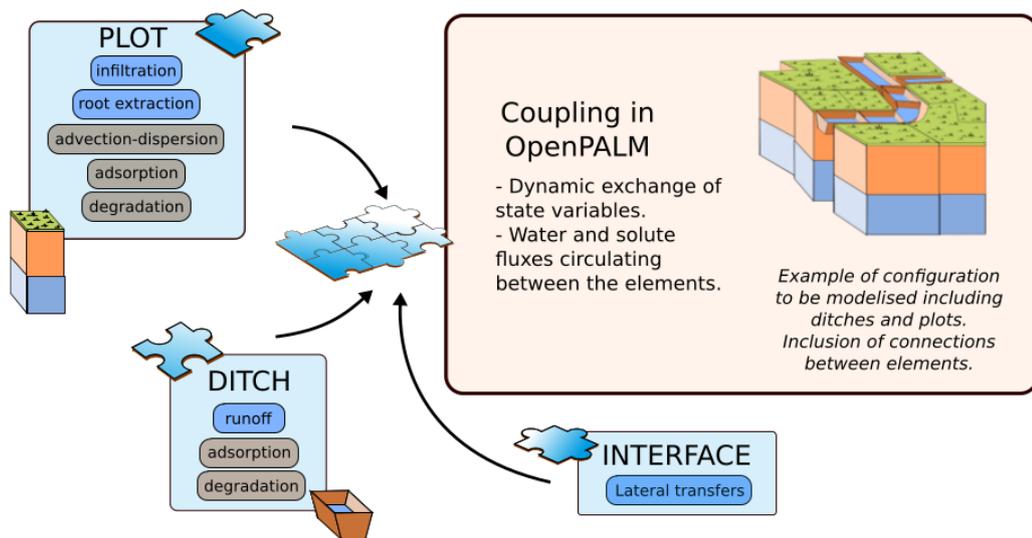


FIGURE 55 – Exemple d'assemblage des modules développés dans un modèle complet à l'échelle du bassin versant.

Les principes fondamentaux de PALM (notions de module, d'unité, de communication,...) sont décrits de manière exhaustive dans le guide technique d'utilisation (Morel et al., 2013) et ne seront pas décrits en détails ici. Cette partie propose plutôt une description chronologique des différents développements faits dans OpenPALM et notamment de l'intégration successive des différents modules afin de garder en tête les difficultés rencontrées et ainsi comprendre certains choix présentés dans la version finale de PESHMELBA.



# Chapitre 13

## Intégration du module FRER1D dans OpenPALM

### 13.1 Implémentation basique

La première étape consiste à tester le module FRER1D dans OpenPALM afin de vérifier qu'il fonctionne de manière autonome. Le coupleur PALM ne sert alors qu'à lancer le module Fortran. Les temps de début et de fin de simulation sont donnés en entrée au module ROSS03\_FORPALM, ainsi que le nom du fichier contenant tous les paramètres de la simulation. Pour tester le processus de communication des unités, une unité PRINT\_OUTPUT est implémentée. Celle-ci reçoit simplement les variables head et S (saturation) fournies par ROSS\_FORPALM.

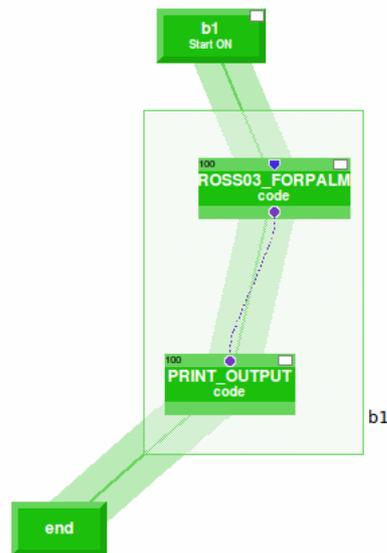


FIGURE 56 – Branche OpenPALM correspondant à l'implémentation d'un module FRER1D

On vérifie que les résultats obtenus après implémentation dans PALM sont exactement les mêmes que ceux obtenus par le module Fortran indépendamment de l'implémentation dans PALM.

## 13.2 Gestion du pas de temps externalisée dans PALM

A terme, le pas de temps de la simulation sera géré par PALM pour permettre la synchronisation des simulations sur chaque élément du paysage. Dans cette optique, il est nécessaire d'adapter la branche PALM et le module FRER1D en séparant l'initialisation des variables et la résolution à un seul pas de temps. Alors que le module FRER1D prenait en entrée les temps de début et de fin de la simulation entière, il doit maintenant pouvoir recevoir en entrée les temps de début et de fin d'un pas de temps et tourner pour chaque pas de temps comme s'il s'agissait d'une simulation indépendante. Ainsi, PALM doit maintenant lancer le module FRER1D autant de fois qu'il y a de pas de temps, en lui fournissant des temps de fin et de simulation à chaque fois. PALM doit aussi être capable de stocker en mémoire les variables d'état obtenues en sortie de FRER1D pour les re-fournir au pas de temps suivant. Pour ce faire, on se base sur l'utilisation du buffer, la mémoire permanente de PALM. Les variables d'état de chaque UH seront envoyées à chaque pas de temps dans le buffer. Pour pouvoir retrouver les variables de la bonne UH au bon pas de temps, il est donc nécessaire d'identifier les variables envoyées dans le buffer par un TAG d'identification et d'identifier les variables produites à chaque pas de temps par un tag TIME.

La structure adoptée est présentée fig. 57. Le module ROSS\_FORPALM a été coupé en 2 modules :

### INITIALISATION\_ROSS

INITIALISATION\_ROSS récupère les paramètres de la parcelle dans un fichier d'entrée dont le nom lui est donné en entrée (variable PALM 'name') et initialise les variables S (vecteur de saturation S dans chaque couche numérique de la colonne de sol) et head (pression dans chaque couche numérique). Les variables S, head ainsi que les forçages en quantités cumulées et les quantités de solutés (en prévision) sont stockées dans des variables PALM avec un tag time égal à 0. On stocke également dans des objets de type dérivés toutes les propriétés hydrauliques et géométriques de la parcelle. Au lieu d'ouvrir à nouveau le fichier 'name', les autres modules récupéreront les informations nécessaires dans ces structures. **Sur ce dernier point, il faudra voir d'un point de vue mémoire / optimisation du temps de calcul si c'est le plus pertinent.**

### SOLVE\_ROSS

SOLVE\_ROSS qui applique la méthode de résolution proposée par Ross sur la colonne de sol entre les temps  $t$  et  $t+dt$ . SOLVE\_ROSS prend en entrée les variables S et heads dotées du TIME TAG `ib_do` et renvoie les variables S et heads tagées temporellement `ib_do+1`.

Le module SOLVE\_ROSS est contenu dans une boucle WHILE qui permet l'incrément du temps de la simulation et la mise à jour des tags temporels.

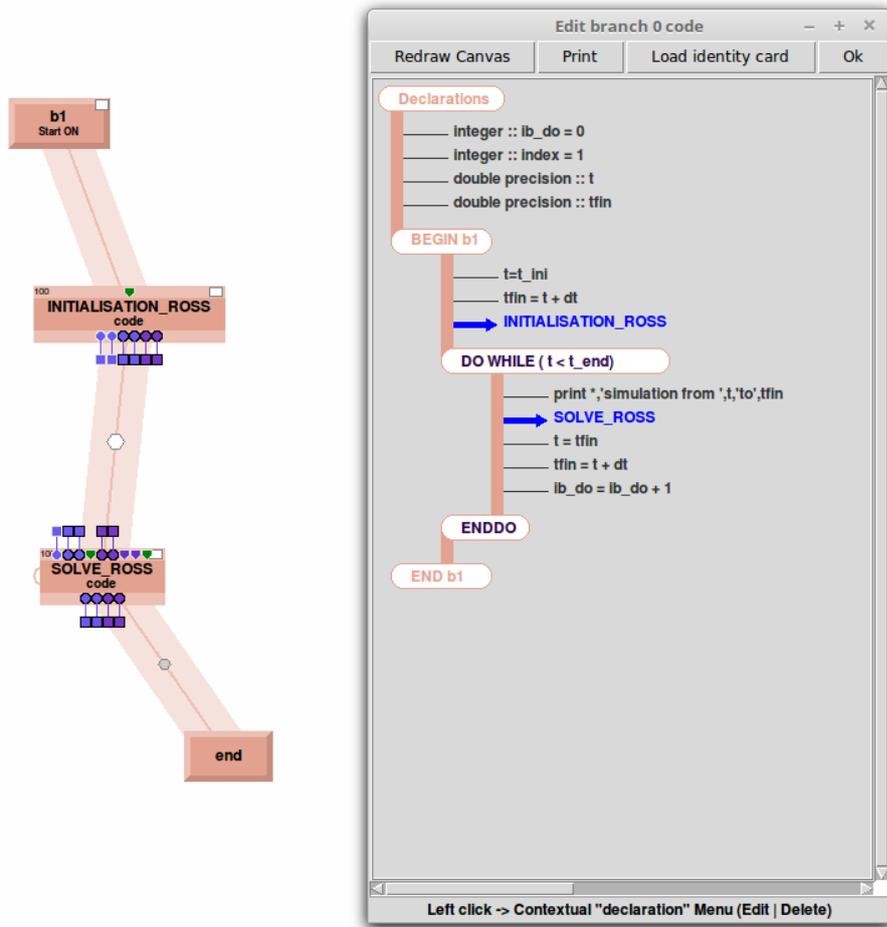


FIGURE 57 – Branche OpenPALM correspondant à l'adaptation du module FRER1D



# Chapitre 14

## Couplage de modules FRER1D par l'interface Plot-Plot-Interface

Pour coupler 2 UH de type parcelle, il est nécessaire de calculer les flux latéraux provenant du gradient de charge hydraulique entre les 2 colonnes de sol. C'est le rôle de l'interface PPI (renommée ColumnColumnInterface par la suite). Dans le cadre de la modélisation à l'échelle du bassin versant où chaque UH peut-être reliée à une ou plusieurs voisines, il faut intégrer une étape permettant l'identification des parcelles connectées et la récupération de leur propriétés dans les tables fournies par GeoMELBA. Pour une parcelle donnée, on se base sur l'implémentation présentée dans la section 13.2 pour laquelle le repérage des éléments voisins et le calcul du flux latéral à chaque pas de temps sont ensuite intégrés.

### 14.1 Modules utilisés

#### INITIALISATION\_ROSS et SOLVE\_ROSS

cf section précédente.

#### GET\_UPDOWN\_COLUMNS

Pour une unité homogène d'identifiant  $i$ , le module GET\_UPDOWN\_COLUMNS identifie les unités homogènes qui lui sont connectées. A partir de l'identifiant  $i$ , le module GET\_UPDOWN\_COLUMNS parcourt la table des connexions fournie par GeoMelba et repère toutes les lignes où l'indice  $i$  apparaît, que ça soit en amont ou en aval. Tous les éléments connectés sont alors stockés dans un objet "connection". Cet objet est ensuite envoyé à PALM.

#### USE\_WTI

Le module use\_WTI correspond à une version simplifiée du module PPI décrit dans la partie 2. Dans cette version basique, le module ne traite qu'un élément surfacique. Il reçoit en entrée un indice d'UH  $i$ , un entier  $ib\_do$  permettant de savoir à quel pas de

temps il travaille et le vecteur "connections" correspondant à toutes les unités connectées à l'UH  $i$ . Pour chaque élément d'identifiant  $j$  du vecteur "connections", `use_WTI` récupère les variables d'état/propriétés relatives à cet objet (grâce aux tags  $j$  pour identifier l'UH et `ib_do` pour récupérer les variables au bon pas de temps) et les utilise pour initialiser une instance de la classe `Colonne`. Une instance de la classe `Interface` est alors initialisée à partir de la colonne courante  $i$  et cette UH connectée  $j$ . Il résulte de cette interface un flux latéral (qui peut-être nul s'il n'y a pas besoin de rééquilibrer). Une fois calculé, ce flux latéral (positif, négatif ou nul) est incrémenté au flux latéral total qui sera attribué à la colonne courante d'identifiant  $i$ .

Par exemple, pour une UH 1, connectées aux UH 3, 10 et 7 :

$$Qlat_1 = qlat_{1\leftrightarrow 3} + qlat_{10\leftrightarrow 1} + qlat_{7\leftrightarrow 1}$$

où chacun des termes  $qlat_{1\leftrightarrow 3}$ ,  $qlat_{10\leftrightarrow 1}$  et  $qlat_{7\leftrightarrow 1}$  ont été calculés par la méthode `calculate_flux` de la classe `interface`. La somme permettant d'obtenir  $Qlat_1$  est ensuite faite par le module `use_WTI`.

On note que l'on raisonne avec des quantités algébriques permettant d'exprimer si chaque contribution au flux est entrante ou sortante. En sommant ces contributions, le flux total est positif si les parcelles connectées ont permis au total de faire rentrer de l'eau dans la parcelle, négatif sinon.

## 14.2 Assemblage dans une branche PALM

L'ensemble des modules sont ensuite assemblés dans une branche PALM représentant le fonctionnement d'une parcelle (fig. 58) :

- ✓ Les variables de branches `ib_do` (tag temporel des variables), `index` (numéro de la branche), `t` (temps de début de simulation), `tfin` (temps de fin de simulation), `dt` (pas de temps) sont initialisées dans la branche PALM.
- ✓ Le temps de fin de simulation est initialisé.
- ✓ Les variables relatives à l'exécution de FRER1D sont initialisées au travers du modèle `INITIALISATION_ROSS`.
- ✓ Début de la boucle TANT QUE  $t < t\_end$  (temps de la simulation TOTALE).
- ✓ Lancement du module `SOLVE_ROSS` entre  $t$  et  $t\_fin = t + dt$ . Les variables sont stockées dans le buffer avec le tag temporel `ib_do` et le tag général `index`.
- ✓ Incrémentation de la variable `ib_do`.

- ✓ Récupération des indices des éléments connectés par le module GET\_UPDOWN\_COLUMNS.
- ✓ Calcul du flux latéral total entrant/sortant dans cette colonne à partir des contributions des colonnes amont ou aval.
- ✓ *A l'origine, on avait choisi d'effectuer la phase de rééquilibrage ici, dans un module Solve\_Ross\_lateral sans forçage climatique, et de manière "instantanée". Finalement, cette approche a été abandonnée car elle implique qu'en même temps que la nappe se rééquilibre, le reste du profil continue aussi à se rééquilibrer également de manière "instantanée". On applique donc le flux latéral calculé à ce pas de temps au pas de temps suivant en même temps que les forçages climatiques dans le module SOLVE\_ROSS.*
- ✓ Dans le schéma de branche présenté fig. 58, une étape de modification du pas de temps est prévue s'il existe un flux latéral non nul (latflag!= 0). Cette approche sera abandonnée par la suite car posant des problèmes pour la synchronisation des branches.
- ✓ Incrémentation des temps de début et de fin de simulation.
- ✓ Fin de la boucle TANT QUE.

## 14.3 Mise en place du couplage

### Duplication des branches

Pour coupler deux UH de type parcelle, une première idée est de dupliquer la branche b1 présentée fig. 58 en une seconde branche b2 avec pour unique différence la valeur de l'index déclaré dans les variables en haut/début de branche. La figure 59 présente une telle implémentation.

### Schéma plus générique

A l'échelle des bassins versants de la Morcille et de Fontaine du Theil, on compte plusieurs centaines de parcelle à modéliser. Il n'est donc pas envisageable de créer une branche par UH à modéliser. Il est nécessaire que toutes les parcelles puissent être gérées par une seule branche, soit dans une boucle, soit en parallèle. Lorsque l'on duplique une branche comme dans l'exemple de la fig. 59, la seule variable indiquant qu'il s'agit d'une parcelle différent est la variable index. En incrémentant cette variable dans une boucle FOR, on peut ainsi faire tourner chaque module pour toutes les parcelles en choisissant un système d'indexation adapté (cf fig. 60).

Fig. 60, on note la présence de 2 boucles FOR à la suite l'une de l'autre. Ceci pour s'assurer que les modules GET\_UPDOWN\_COLUMNS et use\_WTI ne se déclencheront qu'une fois que les profils de saturation de TOUTES les colonnes auront

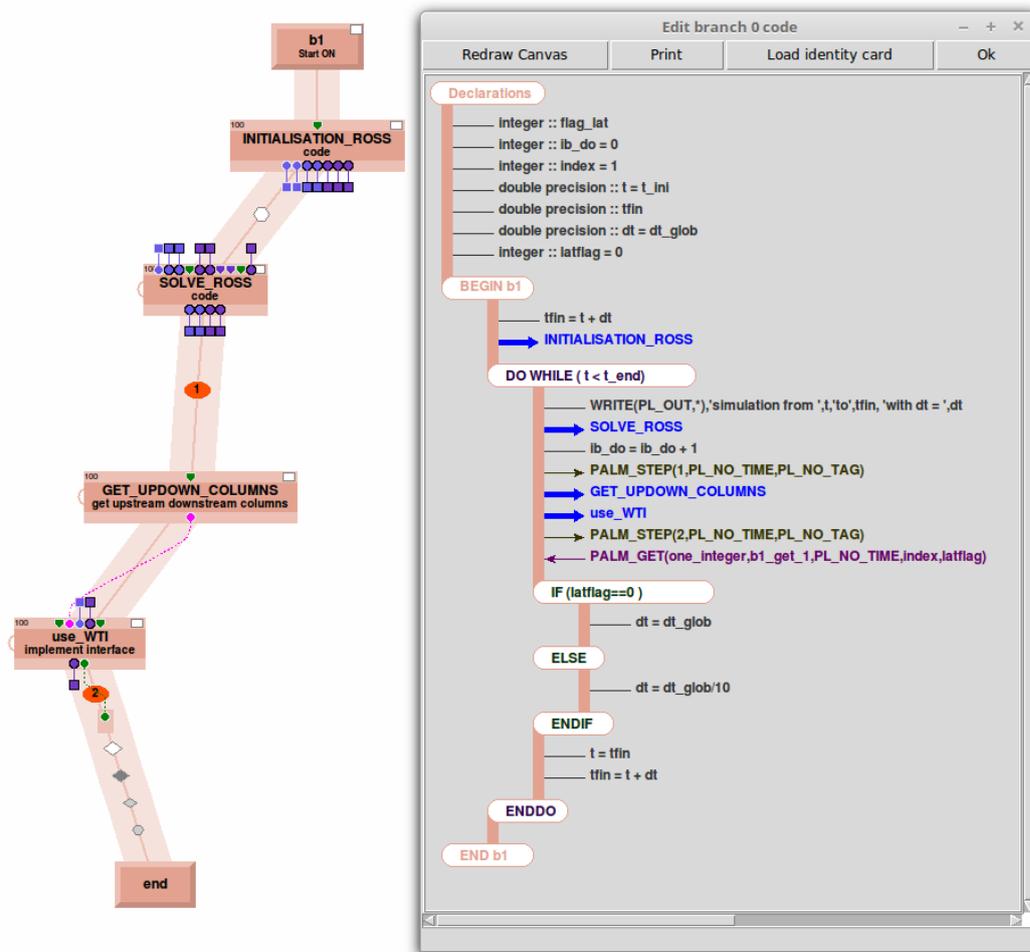


FIGURE 58 – Branche OpenPALM correspondant au fonctionnement d'une UH de type parcelle

été calculés au temps  $t$  et ne pas créer de confusions ou décalages dans le calcul des flux latéraux de rééquilibrage.

Au final, la fig 61 présente une version plus propre et allégée du schéma de couplage. Une première branche d'initialisation et de synchronisation est d'abord implémentée. Elle permet d'initialiser toutes les UH grâce au module INITIALISATION\_ROSS placé à l'intérieur d'une boucle FOR qui est incrémentée sur les indices des parcelles. Les temps de début et de fin de simulation sont calculés grâce au pas de temps  $dt$  donné dans les constantes de PALM et les forçages climatiques au temps  $t$  de début de simulation sont calculés à partir de chroniques de forçages climatiques par le module get\_forcing. La boucle b1 gérant toutes les parcelles est enfin activée. la branche synchro lui transmet les temps de début et de fin de simulation (fin et début d'un pas de temps). On ne passe au temps suivant que lorsque la boucle b1 a fini de tourner pour toutes les parcelles (présence du step).

## Test sur la teneur en eau des cellules

Lors des premiers tests avec un pas de temps assez grossier, on se rend compte que la simulation se retrouve souvent bloquée car on enlève trop d'eau à la colonne ayant

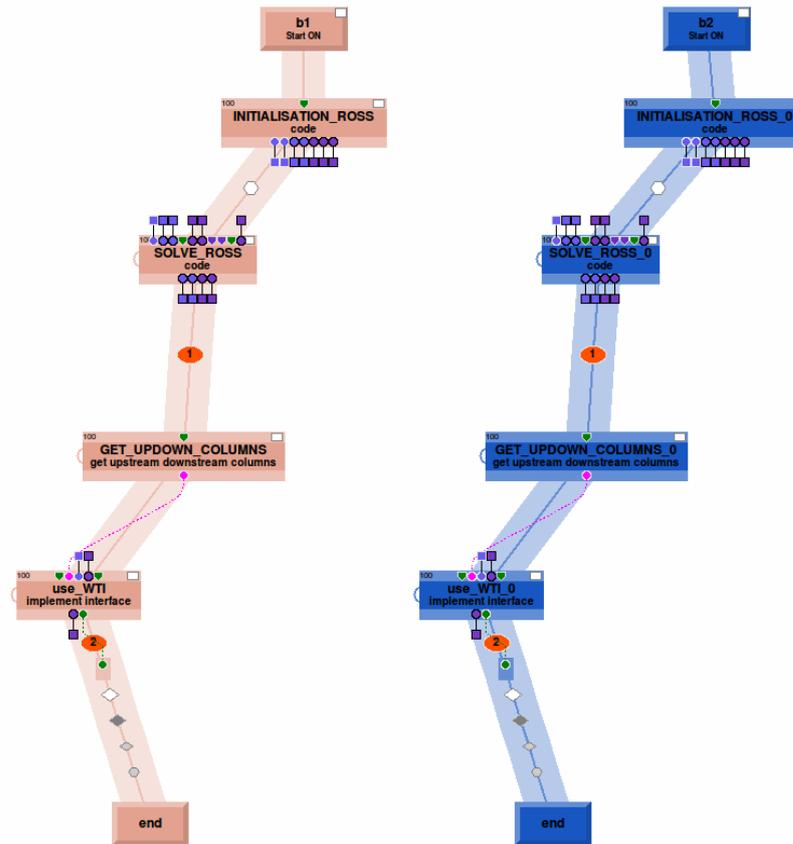


FIGURE 59 – Branche OpenPALM correspondant au fonctionnement de 2 UH de type parcelle

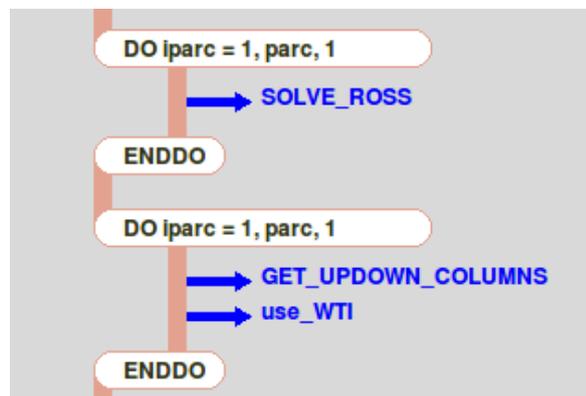


FIGURE 60 – Boucles FOR de la branche b1

un flux latéral sortant. Ceci s'explique car le module use\_wti calcul un débit en fin de boucle mais le pas de temps suivant n'est pas adapté pour que la quantité d'eau effectivement enlevée soit inférieure à la quantité d'eau dans la nappe à rééquilibrer. Une solution pourrait être de raffiner le pas de temps pour la colonne considérée. Cette solution est viable et devrait se baser sur les capacités d'interpolation temporelle dans le buffer offerte par PALM. Comme ce n'est pas l'option la plus facile, on part dans un premier temps sur une autre solution : ajouter une étape de test dans use\_WTI

qui vérifie que la quantité d'eau que l'on enlève à une colonne n'est pas supérieure à sa capacité totale. Si c'est le cas, on enlève seulement la quantité d'eau contenue dans la cellule en la répartissant éventuellement entre les colonnes aval s'il y en a plusieurs.

## 14.4 Optimisation

Le schéma présenté dans la partie précédente fonctionne correctement cependant, en augmentant le nombre de parcelles et la durée de la simulation, on se rend compte que les performances sont assez mauvaises (time CPU et "elapsed time") : plus de 180s (CPU) pour une simulation de 48h avec 5 parcelles. Suite à une discussion avec Thierry, un ensemble de modifications a pu être apporté pour améliorer ces performances :

1. Les scripts python ont été regroupés au maximum. En effet, PALM doit lancer l'interpréteur python à chaque fois qu'une unité est appelée ce qui est coûteux en terme de temps d'exécution. Ainsi, le module `get_updown_column` a été intégré au module `use_wti` (renommé PPI par la suite).
2. Dans cette même logique de regroupement, la boucle faite sur les parcelles dans la branche `b1` (cf fig. 61) a elle aussi été intégrée au script PPI.
3. Pour limiter les entrées/sorties et notamment la lecture des tables fournies par GeoMELBA, ces dernières sont lues une unique fois dans la branche `synchro` au travers d'un nouveau module python `init_CONF` puis stockées dans des variables PALM auxquelles accéderont les autres unités PALM lorsqu'elles en auront besoin. De plus, cette unité PALM a également pour rôle de charger les propriétés des différents types de sol (stockées dans des objets `soil_profile_o`), la géométrie de chaque UH du bassin versant (stockée dans des objets `geometrie_o`), la pression capillaire initiale dans chaque cellule numérique pour chaque UH (stockée dans la variable PALM `heads`), la géométrie et les caractéristiques de tous les linéaires (objet `linear_o`) et les forçages cumulés pour faire le bilan de masse à chaque pas de temps (objet `forcing_linear_o`).

Le nouveau schéma PALM intégrant ces améliorations est présenté fig. 62 et les tests de comparaison avec le schéma initial permette de confirmer le gain de temps associé : pour une simulation avec 5 parcelles pendant 48h, on passe de 180 s CPU à environ 40s (CPU). Pour la suite, il faudra donc essayer au maximum de regrouper les unités python et insérer les boucles dans les scripts et non pas les faire faire à PALM pour ne pas perdre en efficacité (du moins pour les scripts python). En ce qui concerne les unités Fortran, on peut se tourner vers l'usage de blocs qui peut permettre d'optimiser également l'usage des ressources et le temps de calcul.

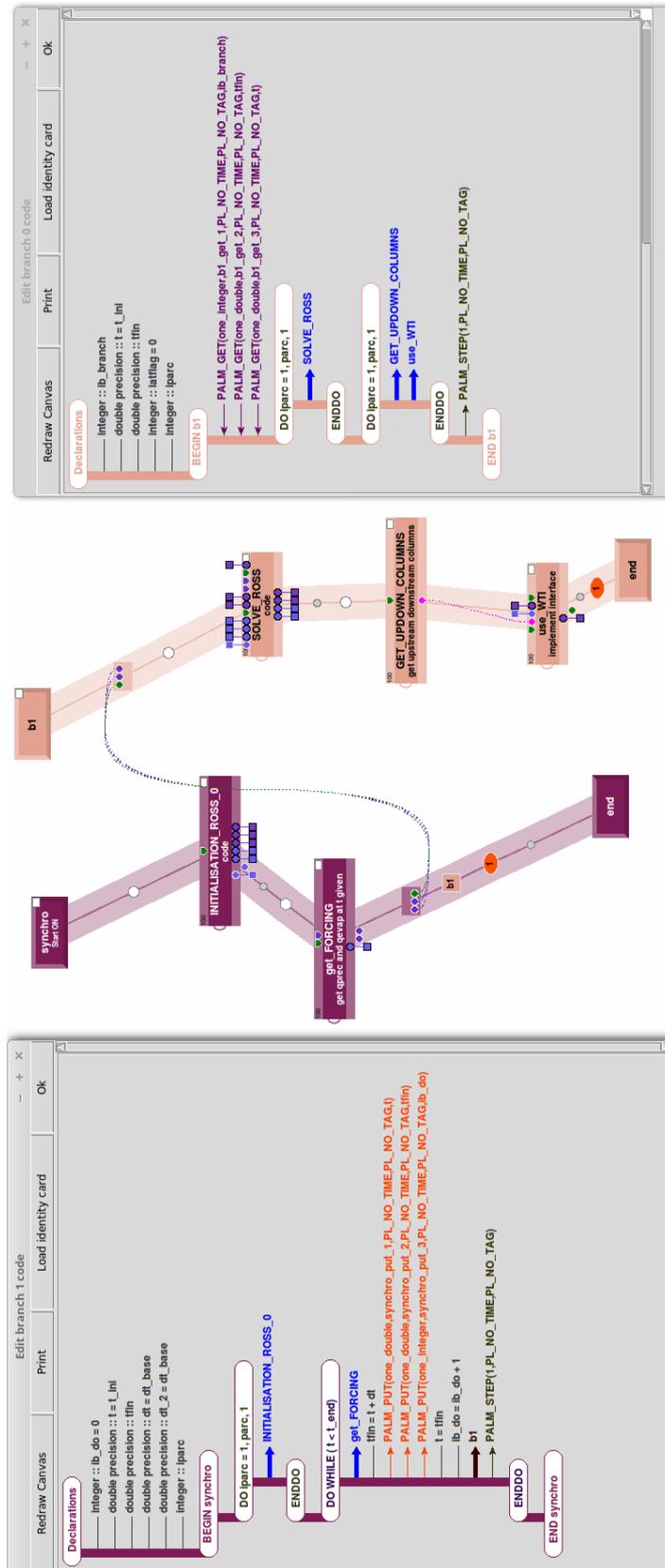


FIGURE 61 – Implémentation PALM gérant l'intégralité des UH de type parcelle sur le bassin versant

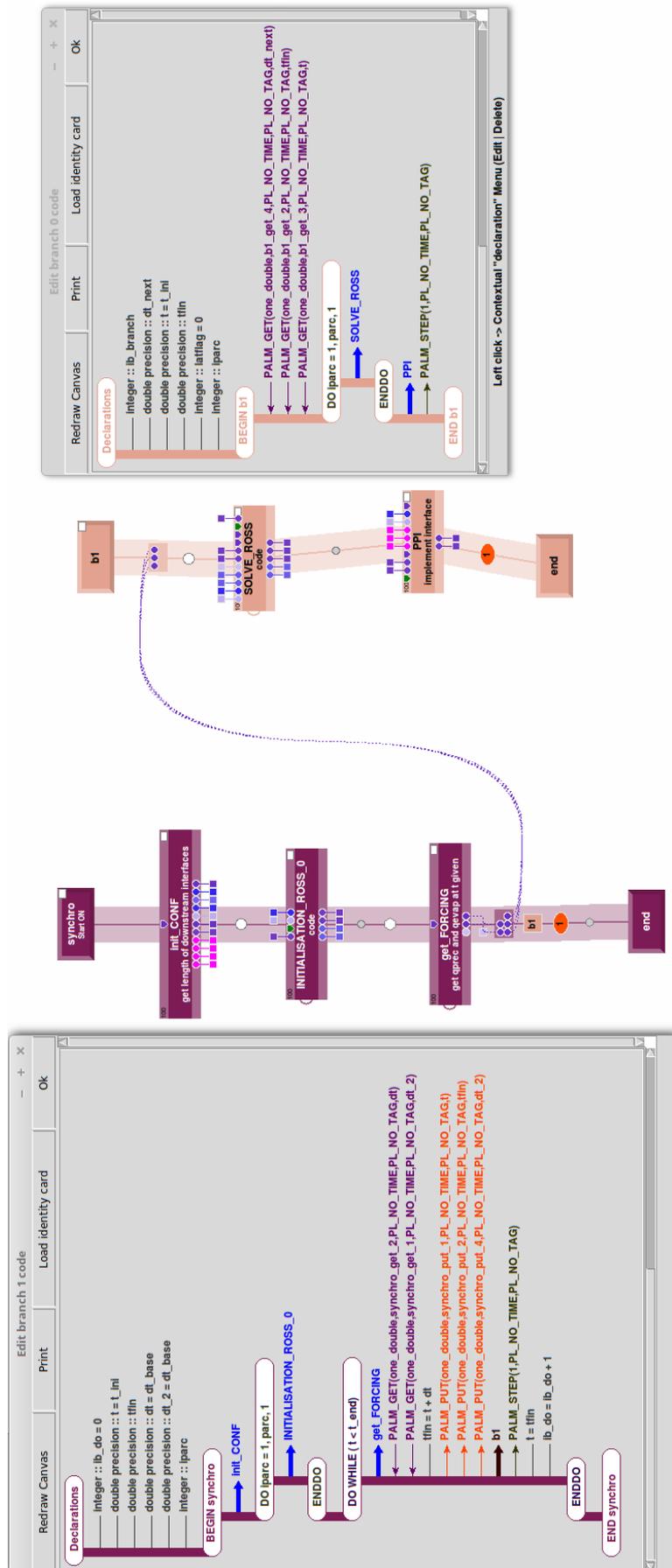


FIGURE 62 – Schéma PALM intégrant les optimisations.

# Chapitre 15

## Gestion des linéaires dans le schéma PALM

Comme détaillé dans la partie II, les linéaires sont modélisés dans un premier temps par 2 unités PALM : l'unité *PPI*, enrichie pour gérer les interfaces surfaciques-linéaires en latéral puis l'unité *solve\_linear* permettant de gérer les interfaces linéaire-linéaire en longitudinal et d'actualiser les volumes d'eau contenus dans ces éléments à chaque pas de temps.

Pour inclure ces unités dans le schéma PALM (ou du moins l'unité *solve\_linear*; l'unité *PPI* étant déjà incluse), on choisit de créer une autre branche "linear". Ainsi, les branches b1 et linear pourront tourner en même temps ce qui permet de gagner du temps. Cependant, il faut prévoir plusieurs steps de synchronisation entre ces 2 branches et la branches synchro pour être sûr que toutes les variables ont bien été calculées avant de passer au mises à jour des variables d'état puis au pas de temps suivant.

Le schéma PALM incluant les linéaires est présenté fig. 63. On note l'utilisation des deux steps suivant :

1. Le step 1, apparaît sur les branches b1, synchro et linear. Il permet de vérifier que tous les calculs sur les UH et les linéaires ont été effectués dans les branches b1 et linear avant que la branche synchro passe au pas de temps suivant.
2. Le step 2 apparaît sur la branche b1 et linear. Comme la branche linear s'appuie sur les flux latéraux provenant des UH, il faut être sûr d'utiliser les contributions latérales associées au bon pas de temps. On rappelle qu'au pas de temps n, la branche linear intègre dans les volumes d'eau les contributions latérales du pas de temps n-1. Ainsi le step 2 permet de s'assurer que tous ces calculs sont faits avant que b1 ne lance l'unité *PPI* qui met à jour ces flux latéraux.

D'autre part, comme pour la branche b1, la branche synchro transmet à b1 les variables de début et de fin de simulation qui seront ensuite demandées par l'unité *solve\_LINEAR*.

En annexe, on propose un tableau récapitulant tous les "PUT" et les "GET" des unités PALM (à compléter au fur et à mesure) apparaissant à ce stade.

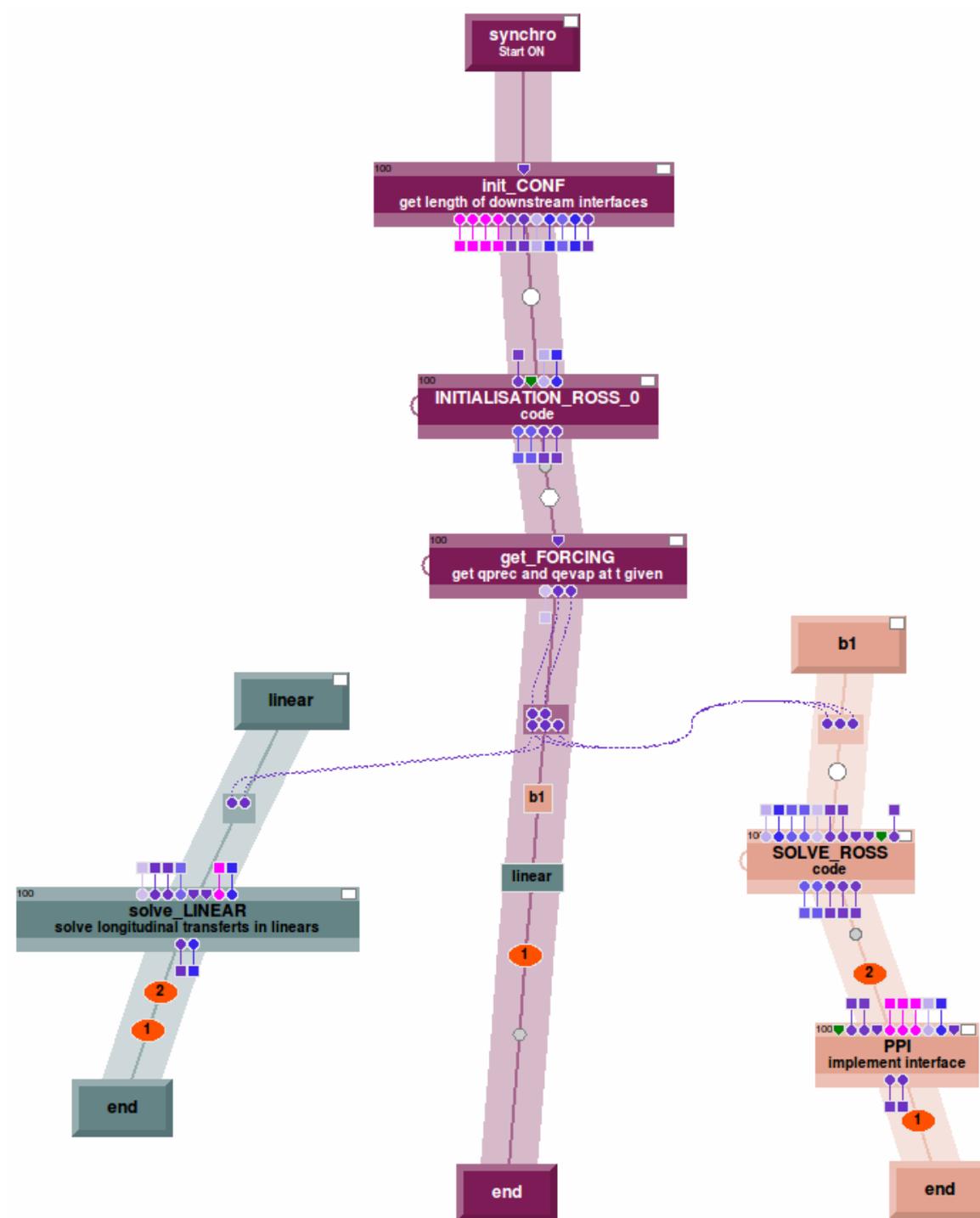


FIGURE 63 – Schéma PALM incluant les linéaires.

# Chapitre 16

## Insertion du ruissellement dans le schéma PALM

Le ruissellement est codé sous la forme d'un module indépendant (cf partie II) et c'est OpenPALM qui permet le couplage surface-subsurface en gérant les échanges de variables entre les modules FRER1D (représentation de la subsurface) et SURROM (représentation de la surface). On rappelle (voir chapitre 11) que lors d'une simulation, le module FRER1D tourne à chaque pas de temps PALM ( $dt_{PALM}$ ), transmet les niveaux d'eau en surface au module SURROM qui tourne au même pas de temps pour simuler le ruissellement avant de retransmettre à FRER1D les nouvelles hauteurs d'eau sur chaque élément. Ces nouvelles hauteurs d'eau seront alors utilisées en entrée du module FRER1D au pas de temps suivant. De plus, un pas de temps PALM est divisé en plusieurs sous pas de temps de ruissellement ( $dt_{RO}$ ) et à chaque sous pas de temps, le module SURROM transmet en début de sous pas de temps du module solve\_LINEAR des volumes d'eau entrant dans chaque linéaire. L'ensemble de ces étapes est représenté fig. 64

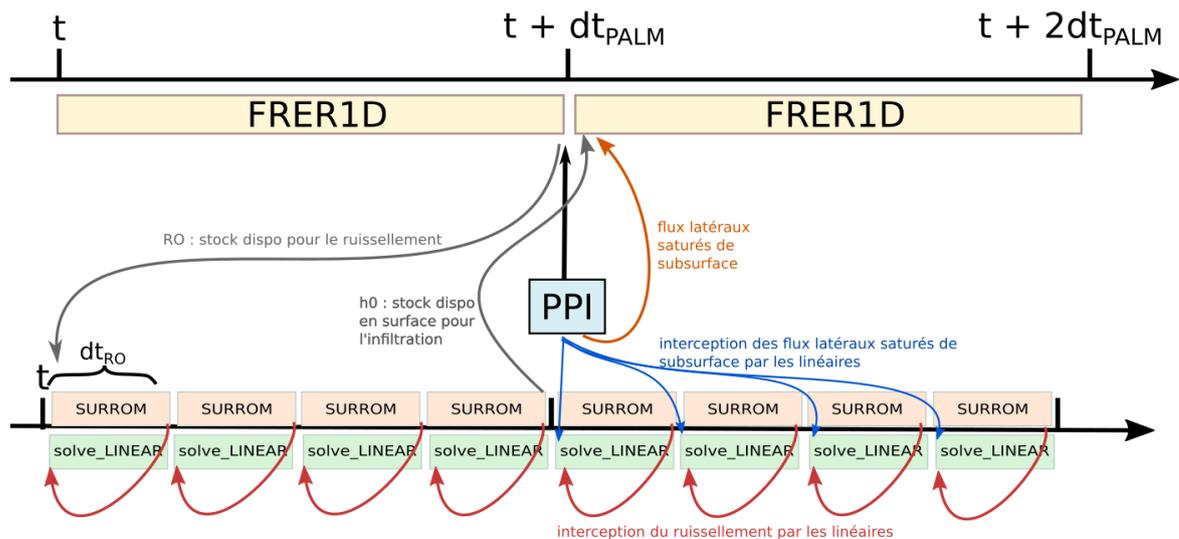


FIGURE 64 – Chronogramme de couplage entre ruissellement sur les parcelles, transferts de surface dans les linéaires, infiltration et transferts latéraux saturés en subsurface dans les parcelles.

D'un point de vue synchronisation dans PALM, cela signifie :

1. que SURROM ne peut pas commencer à tourner entre  $t$  et  $t+dt_{PALM}$  avant que FRER1D ait fini de tourner entre ces mêmes pas de temps ;
2. que solve\_LINEAR ne s'enclenche à un sous pas de temps  $dt_{RO}$  qu'une fois que SURROM a fini de tourner à ce sous pas de temps.

Chacun de ces points est géré de manière différente dans PALM :

1. Pour l'aspect 1, on choisit d'activer la branche surface contenant le module SURROM au travers de la branche b1 (subsurface). Ainsi, la branche surface et le module qu'elle contient ne seront lancés qu'une fois que l'infiltration aura fini de tourner sur toutes les parcelles. L'ajout du step de synchronisation 1 déjà décrit plus haut à la fin de la branche surface permet de s'assurer que la simulation au pas de temps suivant ne pourra pas être lancée tant que la branche surface n'aura pas fini de tourner au pas de temps courant.
2. Pour l'aspect 2, on utilise les propriétés des fonctions PUT et GET d'OpenPALM. En effet, la fonction GET d'OpenPALM a pour propriété d'être bloquante, c'est-à-dire que le script ne peut pas aller plus loin tant qu'il n'a pas reçu la variable attendue par le GET. En branchant un des GET de solve\_LINEAR directement à la sortie du module SURROM, on s'assure que solve\_LINEAR ne pourra pas calculer les transferts dans les linéaires avant d'avoir reçus les volumes d'eau interceptés en surface à ce même pas de temps. De plus, cette configuration permet également d'ajouter la synchronisation inverse en transmettant un potentiel débordement de fossé calculé par solve\_LINEAR au module de ruissellement SURROM.

Une fois assemblés et "branchés", les modules s'articulent comme présenté fig. 65.

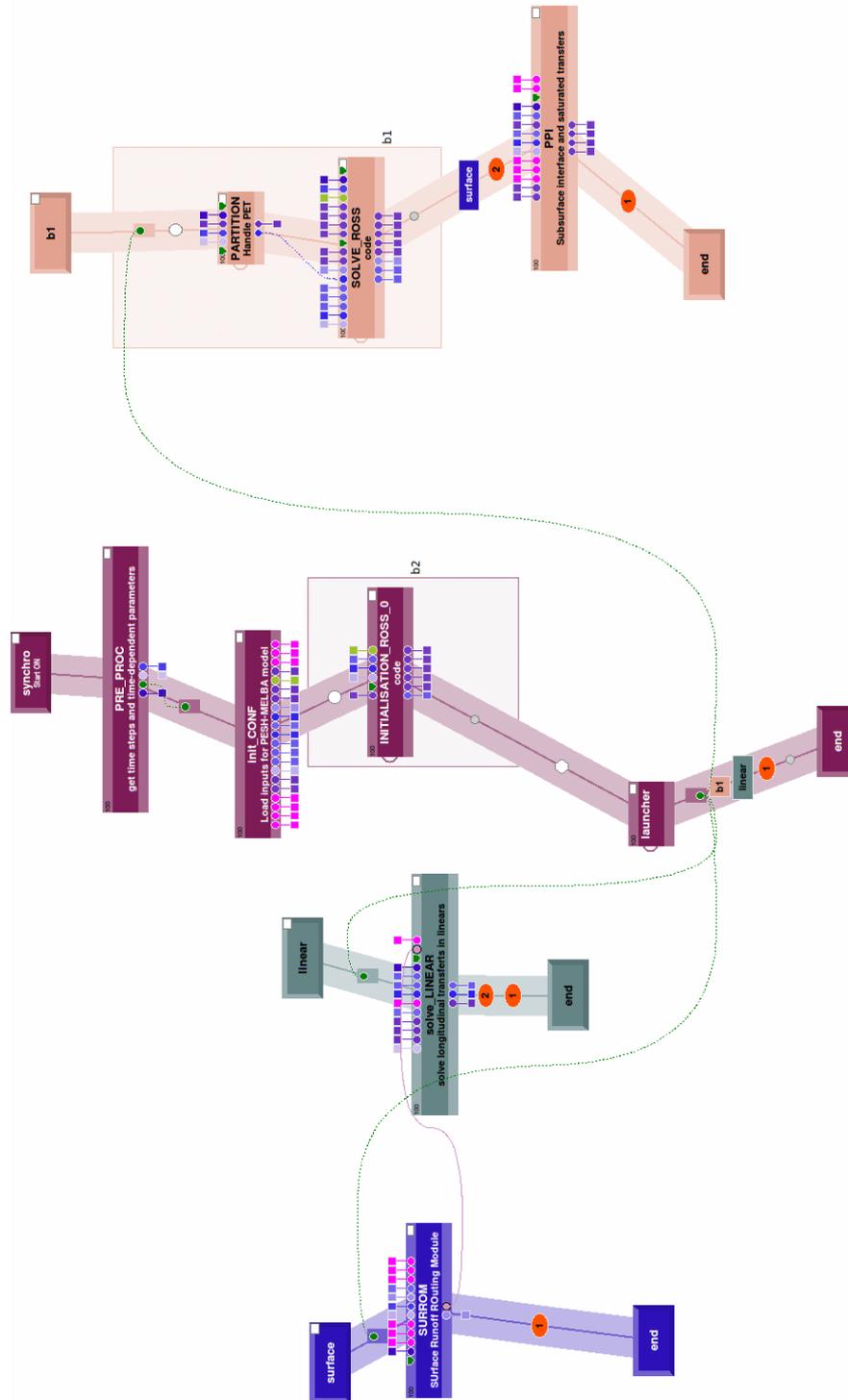


FIGURE 65 – Schéma PALM incluant le module de ruissellement



# Chapitre 17

## Intégration de River1D, gestion des modules Python et boucles infinies

Dans les dernières versions de PESHMELBA, le module River1D a remplacé `solve_LINEAR` pour représenter les écoulements dans un réseau de linéaires de type fossés ou rivières. Cependant, en intégrant ce module au schéma PALM, nous avons été confrontés à des difficultés de temps de calcul ; d'abord car la méthode de résolution de River1D n'est pas spécialement performante mais aussi car l'usage d'un script Python sur une branche supplémentaire est coûteux. En effet, dans les schémas PALM élaborés jusqu'à présent, chaque branche autre que la branche synchro était appelée à chaque pas de temps et les scripts Python était relancés à chaque fois. Or, pour appeler un script Python, OpenPALM est obligé de lancer un **interpréteur** dont l'appel est très coûteux en terme de performances. Pour contourner cette difficulté, des boucles infinies ont été mises en place pour permettre de n'appeler qu'une seule fois chaque script Python et de les laisser ouverts tout au long de la simulation. L'idée est de remplacer chaque module Python du schéma par une boucle infinie qui scrute en permanence l'arrivée de signaux (ou flags) lui indiquant qu'une action doit être effectuée. Ceci est fait par une commande PALM de type `Get` permettant l'interruption du script jusqu'à ce qu'il reçoive la variable attendue (on dit que le `Get` est "bloquant"). En recevant un signal, la boucle infinie lance une fonction contenant elle-même toutes les instructions initiales du module Python. Lorsque toutes ces actions ont été effectuées, la fonction renvoie un booléen qui sera transmis à la boucle infinie qui le transmettra à la branche synchro pour permettre la synchronisation de l'application finale. La boucle infinie attend alors de recevoir le prochain flag. L'évolution entre l'ancienne syntaxe et la nouvelle est résumée sur la Figure 66.

Cette nouvelle structure implique également de modifier les techniques de synchronisation de l'application. L'utilisation du `steps` de synchronisation de fin de pas de temps de la branche synchro (step 1 sur Figure 65) est remplacé par un module de synchronisation qui attend les booléens renvoyés de tous les modules Python. En effet, les `steps` de synchronisation sont activés lorsqu'une unité PALM a fini de tourner. Dans le cas des boucles infinies, les unités PALM codées en Python ne s'arrêtent de tourner qu'à la fin de la simulation. Le step 2 présenté sur la Figure 65 permettait d'activer l'unité PPI seulement lorsque l'unité `solve_LINEAR` avait fini de tourner. Ce step a été remplacé par un booléen directement transmis de River1D (anciennement `solve_LINEAR`) vers PPI pour autoriser son lancement.

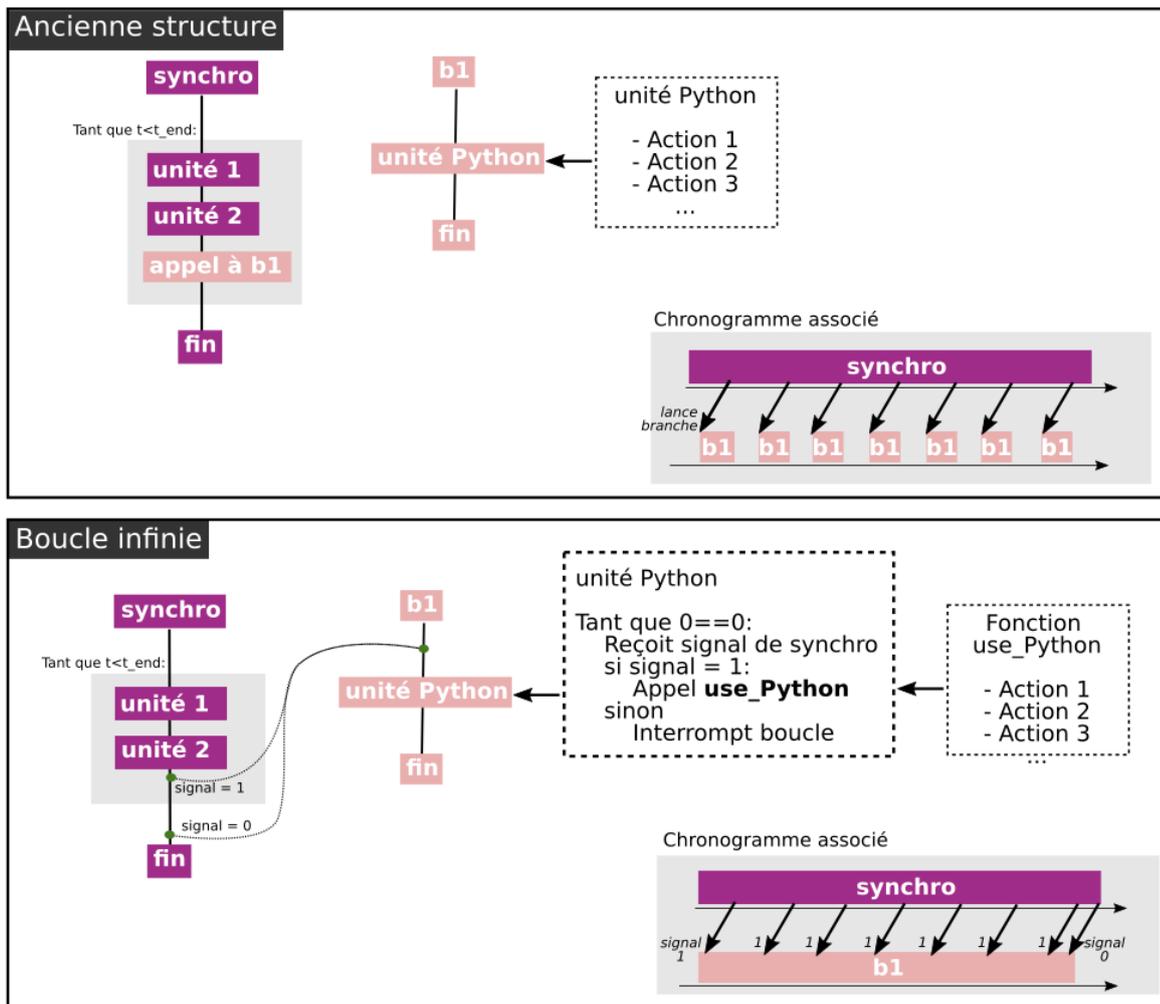


FIGURE 66 – Différence de fonctionnement entre l’ancienne syntaxe et l’optimisation incluant l’utilisation d’une boucle infinie.

⚠ Cette structure est limitée par le nombre de processeurs accessibles sur la machine car chaque branche qui tourne simultanément utilise un processeur. La machine sur laquelle a été développée la version actuelle de PESHMELBA a limité le nombre de branches à 4.

Le nouveau schéma PALM est présenté figure 67. On note la présence de 4 branches. La branche **synchro** gère l’initialisation des variables (INIT\_CONF, PRE\_PROC et INITIALISATION\_ROSS) et le lancement de la boucle temporelle permettant le déroulement de la simulation. Dans cette boucle, est aussi lancé le calcul de l’infiltration verticale (SOLVE\_ROSS) et de l’évapotranspiration (PARTITION). La branche **surface\_UH** gère le calcul des flux de surface entre éléments surfaciques (unité SURROM). La branche **network** gère le lancement de RIVER1D pour le calcul des écoulements dans le réseau et la branche **subsurface\_UH** lance le calcul des flux latéraux en subsurface entre tous les types d’éléments (unité PPI). Hormis **synchro**, les 3 autres branches sont composées d’un unique script Python contenant une boucle infinie.

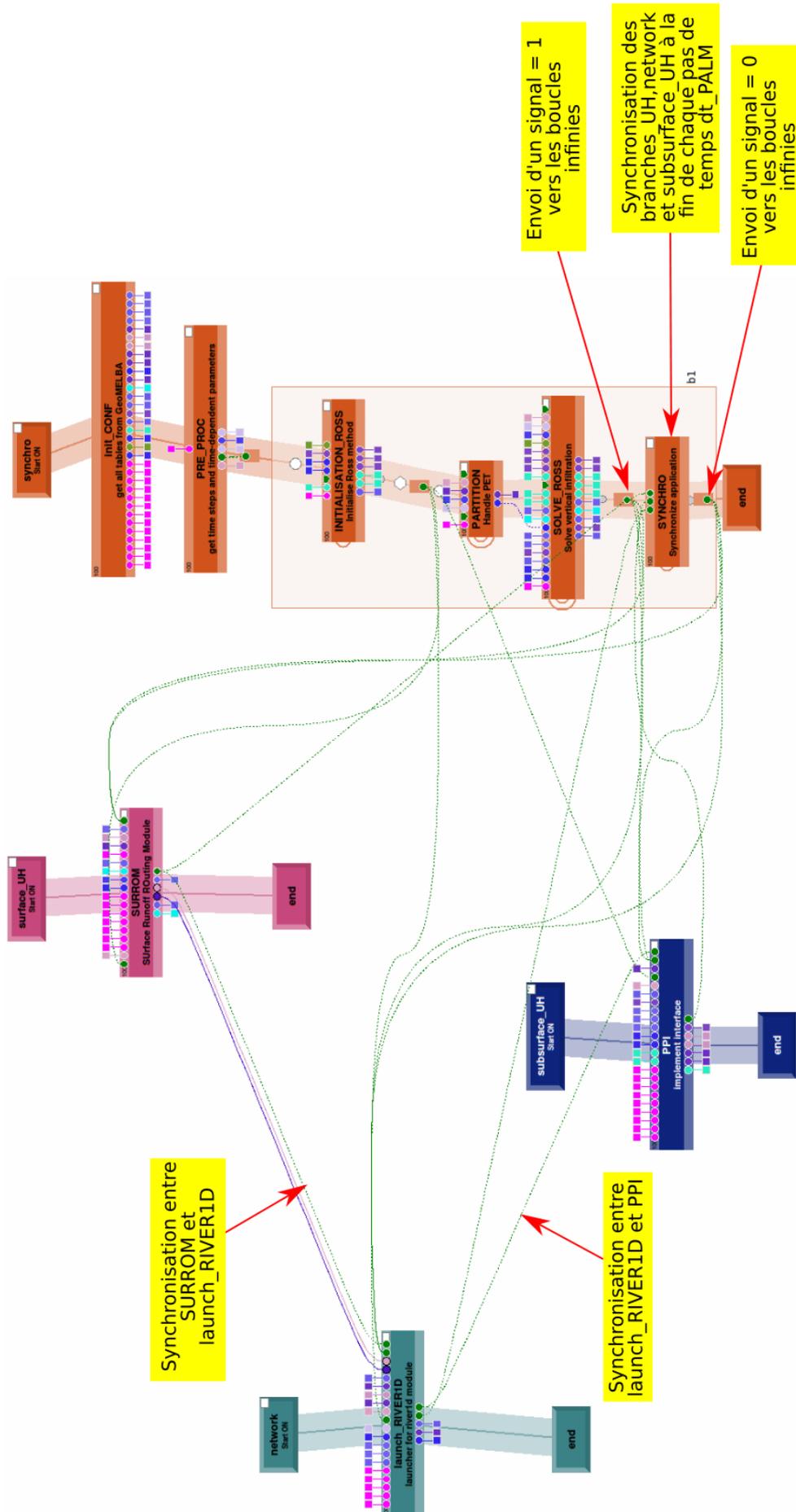


FIGURE 67 – Schéma PALM après optimisation avec les boucles infinies page 147



# Chapitre 18

## Schéma PALM définitif de l'application

Le schéma PALM a été modifié une dernière fois pour intégrer les derniers développements et optimiser l'appel aux unités PALM. Le schéma PALM actualisé (Figure 68) inclut les modifications suivantes :

- Insertion de l'unité `FRER1D_HEDGE` permettant le calcul de l'infiltration verticale dans les objets de type haie ;
- Agrégation des unités `PRE_PROC` et `INIT_CONF` pour limiter le nombre de scripts Python lancés ;
- Agrégation des scripts `PARTITION` et `FRER1D` et `PARTITION` et `FRER1D_HEDGE` pour limiter l'appel à des unités ;
- Changement des noms des branches et des modules pour être plus générique. La branche **surface** et son module `EXCHANGES_SURFACE` permettent en effet de gérer la génération et le routage de ruissellement sur tous les éléments de surface et pas seulement les UHs. C'est également le cas pour la branche **subsurface** qui permet de gérer toutes les interfaces latérales.

La dernière modification effectuée consiste en l'optimisation des boucles infinies. Dans la modification présentée section 17, le script Python qui apparaissait dans le schéma PALM final ne contenait qu'une boucle infinie récupérant et envoyant des flags ainsi qu'un appel à une fonction contenant toutes les instructions que devait réaliser l'unité. Ce schéma était un peu redondant car à chaque appel de la fonction, certaines actions étaient relancées alors que ce n'était pas nécessaire. Par exemple, les premières instructions de cette fonction pouvaient être de récupérer des objets PALM contenant des paramètres définissant les objets manipulés. Ces paramètres concernant par exemple la géométrie ou les propriétés de sol sont fixés et ils n'évoluent pas au cours de la simulation. Ils pourraient être chargés une fois pour toutes en début de simulation par la boucle infinie et transmis à la fonction correspondante à chaque appel de celle-ci. Cette modification a été appliquée aux unités `EXCHANGES_SUBSURFACE` et `EXCHANGES_SURFACE` pour lesquelles la structure de base permettait particulièrement ce genre d'adaptation. Dans les scripts contenant la boucle infinie, les différents éléments du paysage et les interfaces entre eux sont initialisés en récupérant leurs propriétés depuis des variables PALM. Ces objets sont stockés dans les dictionnaires `Element_dic` et `ExchangeLine_dic`. La boucle infinie est alors lancée pour attendre les flags de la branche synchro. Lorsqu'un

tel flag est reçu, la fonction contenant le reste des instructions est appelée avec comme arguments les dictionnaires `Element_dic` et `ExchangeLine_dic`. Les fonctions appelées dans `EXCHANGES_SUBSURFACE` et `EXCHANGES_SURFACE` sont appelées respectivement `use_EXCHANGES_SUBSURFACE` et `use_EXCHANGES_SURFACE`. La fonction ne contient plus que les instructions à réaliser pendant un pas de temps : mise à jour des variables d'état, calcul des flux entre éléments et envoi des variables vers PALM. Lorsque toutes ces actions sont réalisées, la fonction retourne en sortie un autre flag qui sera renvoyé depuis la boucle infinie vers les autres branches du schéma PALM pour synchroniser l'application. On note que les branches surface et subsurface sont lancées en même temps que la branche synchro. C'est-à-dire qu'elles peuvent potentiellement essayer de récupérer des variables PALM pour initialiser leurs objets avant même que les modules `INIT_CONF` et `INIT_FRER1D` aient créé ces variables. Il faut donc rajouter un signal de synchronisation à la fin de `INIT_FRER1D` qui sera envoyé vers les branches contenant des boucles infinies pour leur stipuler qu'elles peuvent lancer la récupération de variables PALM. Le chronogramme 69 synthétise le lancement des différents processus avec l'utilisation des variables de synchronisation pour un pas de temps donné.

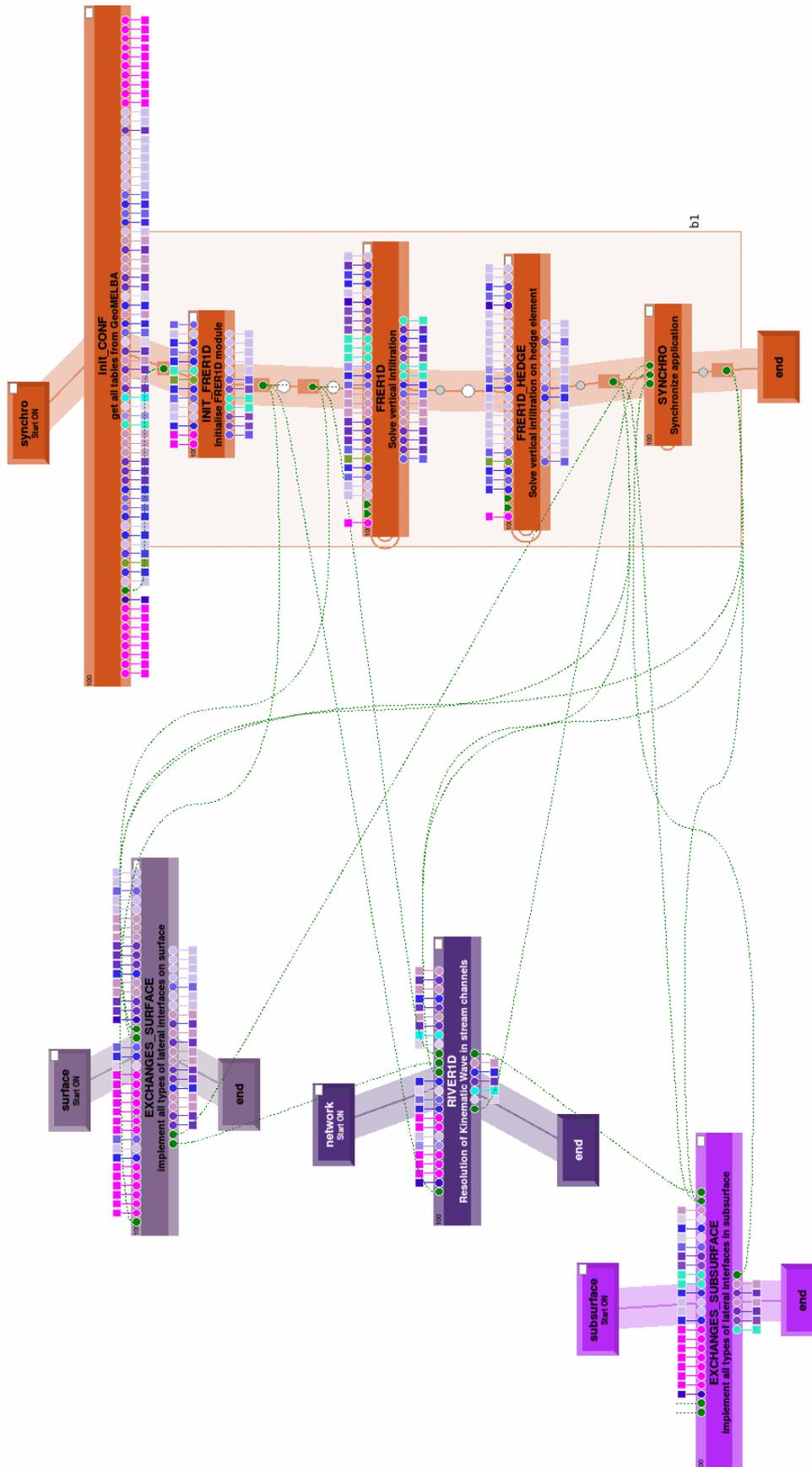


FIGURE 68 – Schéma PALM final



# Bibliographie

- Abbott, M., Bathurst, J., Cunge, J., O’Connell, P., and Rasmussen, J. (1986). An introduction to the european hydrological system — systeme hydrologique europeen, “she”, 2 : Structure of a physically-based, distributed modelling system. *Journal of Hydrology*, 87(1) :61 – 77.
- Adamiade, V. (2004). *Influence d’un fossé sur les écoulements rapides au sein d’un versant*. PhD thesis.
- Beven, K. J. and Kirkby, M. J. (1979). A physically based, variable contributing area model of basin hydrology. *Hydrological Sciences Bulletin*, 24(1) :43–69.
- Branger, F. (2007a). River1d : module simplifié d’écoulement en rivière. documentation du module dans la plateforme liquid. Technical report, Cemagref.
- Branger, F. (2007b). *Use of a modelling framework for representing the influence of landscape management practices on water and pesticides contamination. Application to the Fontaine du Theil experimental catchment*. PhD thesis, Université Joseph-Fourier - Grenoble I.
- Branger, F., Braud, I., Viallet, P., and Debionne, S. (2008). Modelling the influence of landscape management practices on the hydrology of a small agricultural catchment. In *Proceedings of the 8th International Conference on HydroSciences and Engineering*, pages 586–594.
- Braud (2006). Inclusion of a net lateral flux in the saturated zone in the frer1d (fast 1d richards equation resolution) module. Technical report, CEMAGREF.
- Braud, I., Varado, N., and Oliosio, A. (2005). Comparison of root water uptake modules using either the surface energy balance or potential transpiration. *Journal of Hydrology*, 301(1–4) :267 – 286.
- Buis, S., Piacentini, A., and Déclat, D. (2006). PALM : a computational framework for assembling high-performance computing applications. *Concurrency and Computation : Practice and Experience*, 18(2) :231–245.
- Calvet, R. (2005). *Les pesticides dans le sol : conséquences agronomiques et environnementales*. Référence scientifique. Editions France Agricole.
- Camporese, M., Paniconi, C., Putti, M., and Orlandini, M. (2010). Surface-subsurface flow modeling with path-based runoff routing, boundary condition-based coupling, and assimilation of multisource observation data. *Water Resources Research*, 46(2).
- Carluer, N. (1998). *Vers une modélisation hydrologique adaptée à l’évaluation des pollutions diffuses : prise en compte du réseau anthropique. Adaptation au bassin versant de Naizin (Morbihan)*. PhD thesis.

- Deardorff, J. W. (1978). Efficient prediction of ground surface temperature and moisture, with inclusion of a layer of vegetation. *Journal of Geophysical Resources*, 83(C4) :1889–1903.
- Dehotin, J. (2007). *Prise en compte de l'hétérogénéité des surfaces continentales dans la modélisation hydrologique spatialisée. Application sur le haut-bassin de la Saône*. PhD thesis, Institut National Polytechnique de Grenoble, France, Europe.
- Djabekhir, K. (2015). *Computer modeling of pesticide fate at hillslope scale within a hydrological modeling framework taking into account macroporosity*. PhD thesis, Université Grenoble Alpes.
- Dollinger, J., Dagès, C., Negro, S., Bailly, J.-S., and Voltz, M. (2016). Variability of glyphosate and diuron sorption capacities of ditch beds determined using new indicator-based methods. *Science of The Total Environment*, 573 :716 – 726.
- Feddes, R. A., Kowalik, P. J. (Piotr J.), j. a., and Zaradny, H, j. a. (1978). *Simulation of field water use and crop yield*. Wageningen : Pudoc for the Centre for Agricultural Publishing and Documentation.
- Fouilloux, A. and Piacentini, A. (1999). The PALM project : MPMD paradigm for an oceanic data assimilation software. In *Euro-Par'99 Parallel Processing : 5th International Euro-Par Conference Toulouse, France, August 31 - September 3, 1999 Proceedings*, pages 1423–1430. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Freundlich, H. (1909). *Kapillarchemie, eine Darstellung der Chemie der Kolloide und verwandter Gebiete, von Dr. Herbert Freundlich,...* akademische Verlagsgesellschaft.
- Gardner, R. W. (1958). Some steady-state solutions of the unsaturated moisture flow equation with application to evaporation from a water table. *Soil Science*, 85 :228–232.
- Gassmann, M., Stamm, C., Olsson, O., Lange, J., Kümmerer, K., and Weiler, M. (2013). Model-based estimation of pesticides and transformation products and their export pathways in a headwater catchment. *Hydrology and Earth System Sciences*, 17 :5213–5228.
- Gatel, L., Lauvernet, C., Carlier, N., Weill, S., Tournebize, J., and Paniconi, C. (2018). Global evaluation and sensitivity analysis of a physically based flow and reactive transport model on a laboratory experiment. *Environmental Modelling and Software*.
- Horton, R. E. (1933). The role of infiltration in the hydrologic cycle. *Eos, Transactions American Geophysical Union*, 14(1) :446–460.
- Huygen, J., Van Dam, J., Kroes, J., and Wesseling, J. (1997). *SWAP 2.0 : input and output manual*. Wageningen Agricultural University.
- Jarvis, N. (1989). A simple empirical model of root water uptake. *Journal of Hydrology*, 107(1) :57 – 72.
- Kraft, P., Vache, K. B., Frede, H.-G., and Breuer, L. (2012). CMF : A hydrological programming language extension for integrated catchment models. *Environmental Modelling and Software*, 26(6) :828–830.
- Lagacherie, P., Rabotin, M., Colin, F., Moussa, R., and Voltz, M. (2010). Geo-mhydas : A landscape discretization tool for distributed hydrological modeling of cultivated areas. 36(8) :1021–1032.

- Lai, C.-T. and Katul, G. (2000). The dynamic role of root-water uptake in coupling potential to actual transpiration. *Advances in Water Resources*, 23(4) :427–439.
- Li, K. Y., Boisvert, J. B., and Jong, R. D. (1999). An exponential root-water-uptake model. *Canadian Journal of Soil Science*, 79(2) :333–343.
- Li, Z. and Zhang, J. (2001). Calculation of field manning's roughness coefficient. *Agricultural Water Management*, 49(2) :153 – 161.
- Margoum, C., Gouy, V., Williams, R., and Smith, J. (2001). Le rôle des fossés agricoles dans la dissipation des produits phytosanitaires. *Ingénieries - EAT*, 1 :55–65.
- Martinez, A. (2007). Modélisation numérique d'écoulements en sols saturés/non saturés : Comparaison de différentes formulations des equations de richards 1d ; introduction à une formulation des equations de richards 2d et 3d. Master's thesis, Université Bordeaux 1, INRIA.
- Miles, J. (1985). The representation of flows to partially penetrating rivers using groundwater flow models. *Journal of Hydrology*, 82 :341–355.
- Morel, T., Duchaine, F., Thévenin, A., Piacentini, A., Kirmse, M., and Quémerais, E. (2013). Coupleur open-palm vesion 4.1.5, manuel utilisateur et de formation. Technical report, CERFACS.
- Noilhan, J. and Planton, S. (1989). A simple parameterization of land surface processes for meteorological models. *Monthly Weather Review*, 117(3) :536–549.
- Peyrard, X. (2016). Transfert de produits phytosanitaires par les écoulements latéraux en proche surface dans le beaujolais de coteaux : suivi sur parcelle exploitée, expérimentation de travage in situ et modélisation. pages –.
- Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical recipies in Fortran 77. Second edition. The art of scientific computing*. Press Syndicate of the University of Cambridge.
- Roo, A., , A., Wesseling, C., Jetten, V., and Ritsema, C. (1996). LISEM : A physically-based hydrological and soil erosion model incorporated in a GIS. In *In : K. Kovar & H.P. Nachtnebel (eds.), Application of geographic information systems in hydrology and water resources management. Wallingford (UK), IAHS, 1996. IAHS Publ. 235, pp. 395-403.*
- Ross, P. (2003). Modeling soil water and solute transport - fast, simplified numerical solutions. *Agronomy Journal*, 95(6) :1352–1361.
- Ross, P. (2006). Fast solution of richards' equation for flexible soil hydraulic property descriptions. Technical report, CSIRO.
- Schaap, M. G. and van Genuchten, M. T. (2006). A modified mualem-van genuchten formulation for improved description of the hydraulic conductivity near saturation. *Vadose Zone Journal*, 5 :27–34.
- Spanoudaki, K., Stamou, A., and Nanou-Giannarou, A. (2009). Development and verification of a 3-d integrated surface water-groundwater model.
- Taconet, O., Bernard, R., and Vidal-Madjar, D. (1986). Evapotranspiration over an agricultural region using a surface flux/temperature model based on NOAA-AVHRR data. *Journal of Applied Meteorology*, 25(3) :284–307.

- Tortrat, F. (2005). *Modélisation orientée décision des processus de transfert par ruissellement et subsurface des herbicides dans les bassins versants agricoles*. PhD thesis.
- Varado, N. (2004). *Contribution to the development of a distributed hydrological modelling. Application to the Donga catchment, in Benin*. PhD thesis, Institut national polytechnique de Grenoble.
- Varado, N., Braud, I., and Ross, P. (2006a). Development and assessment of an efficient vadose zone module solving the 1d Richards' equation and including root extraction by plants. *Journal of Hydrology*, 323(1-4) :258–275.
- Varado, N., Braud, I., Ross, P., and Haverkamp, R. (2006b). Assessment of an efficient numerical solution of the 1d Richards' equation on bare soil. *Journal of Hydrology*, 323(1–4) :244 – 257.
- Vertessy, R. . . ., Hatton, T. . . ., O'Shaughnessy, P. . . ., and Jayasuriya, M. . . . (1993). Predicting water yield from a mountain ash forest catchment using a terrain analysis based catchment model. *Journal of Hydrology*, 150(2-4) :665–700–.
- Vogel, T. and Císlarová, M. (1998). On the reliability of unsaturated hydraulic conductivity calculated from the moisture retention curve. *Transport in Porous Media*, 3 :1–15.
- Vogel, T., Van Genuchten, M. T., and Císlarová, M. (2001). Effect of the shape of the soil hydraulic functions near saturation on variably-saturated flow predictions. *Advances in Water Resources*, 24(2) :133–144.
- Weill, S. (2007). *Modélisation des échanges surface/subsurface á l'échelle de la parcelle par une approche darcéenne multidomaine*. PhD thesis, École Nationale Supérieure des Mines de Paris.

# Annexe A

## Description des variables transmises dans OpenPALM

Objet PALM	Élément	Description
application_o	smass	masse de solutés appliquée à chaque pas de temps
application_o	wvolume	volume d'eau appliqué à chaque pas de temps
bcs_o	conc_subsurf	concentration en solutés dans le réservoir de subsurface d'un BCS
bcs_o	conc_surf	concentration en solutés dans le réservoir de surface d'un BCS
bcs_o	subsurf_bc	flux potentiel en subsurface d'un BCS (>0 si élément source, <0 si élément puits)
bcs_o	surf_bc	série temporelle de flux en surface pour chaque échéance de temps pour un BCS
bcs_o	type	type de condition limite d'un BCS : 0 : flux nul, 1 : colonne virtuelle, 2 : flux constant, 3 : surface de suintement
bcs_o	updown	position du BCS : 0 : objet puits, 1 : objet source
bcs_o	wtable_height	hauteur de la nappe si l'objet est du type colonne virtuelle
climato_by_time_o	pet_vect	série temporelle d'évapotranspiration potentielle pour chaque temps de la simulation
climato_by_time_o	rain_vect	série temporelle de pluie pour chaque temps de la simulation

column_o	cellvol	volume de chaque cellule numérique d'un objet Column
column_o	id_soil	Indice du type de sol attaché à un objet Column
element_o	lon	longitude d'un objet Element
element_o	lat	latitude d'un objet Element
element_o	alt	altitude d'un objet Element
head_hedge_o	up	pression dans les cellules numériques de la colonne supérieure d'un objet Hedge
head_hedge_o	down	pression dans les cellules numériques de la colonne inférieur d'un objet Hedge
hedge_o	adsorp_thick	épaisseur de la bande d'adsorption en surface d'un objet Hedge
hedge_o	angle_to_leveline	orientation de l'objet Hedge par rapport à la ligne de niveau
hedge_o	bd	masse volumique apparente en surface de l'objet Hedge
hedge_o	cellvol_down	volume des cellules numériques de la colonne inférieure de l'objet Hedge
hedge_o	cellvol_up	volume des cellules numériques de la colonne supérieure de l'objet Hedge
hedge_o	column_height	hauteur des colonnes de sol inférieures et supérieures de l'objet Hedge
hedge_o	downstream_width	largeur de la partie inférieure de l'objet Hedge
hedge_o	upstream_width	largeur de la partie supérieure de l'objet Hedge
hedge_o	hedge_height	hauteur du talus potentiel de l'objet Hedge
hedge_o	manning	coefficient de Manning sur l'objet Hedge
hedge_o	preferential_conductivity	taux d'infiltration préférentielle sur l'objet Hedge
hedge_o	soil_idx	indice du type de sol des colonnes de sol de l'objet Hedge
hedge_o	ths	teneur en eau à saturation en surface de l'objet Hedge

hedge_o	vegetation_idx	indice du type de végétation modélisée sur l'objet Hedge
HU_o	area	surface d'un objet HU
HU_o	Lint	longueur totale d'interface avec des objets aval d'un objet HU
HU_o	Loff	longueur caractéristique de ruissellement d'un objet HU
linear_o	length	longueur d'un objet Linear
linear_o	slope	pente moyenne sur un objet Linear
linear_o	width	largeur moyenne d'un objet Linear
linear_o	Kads	coefficient d'adsorption des solutés modélisés sur un objet Linear
linear_o	nads	coefficient d'adsorption des solutés modélisés sur un objet Linear
linear_o	isotype	type d'isotherme d'adsorption à utiliser pour chaque soluté modélisé sur un objet Linear
reach_o	adsorp_thick	épaisseur de la bande d'adsorption sur un objet Reach
reach_o	bank_angle	fruit des berges d'un objet Reach
reach_o	depth	profondeur d'un objet Reach
reach_o	bd	masse volumique apparente dans le sol d'un objet Reach
reach_o	hpond	hauteur de ponding maximale pour un objet Reach
reach_o	manning	coefficient de Manning pour un objet Reach
reach_o	ths	teneur en eau à saturation sur un objet Reach
river_o	dist_impervious	distance entre le fond de la rivière et l'imperméable de la zone saturée pour un objet River
river_o	ksat	conductivité hydraulique à saturation dans le lit d'un objet River
river_o	qin	débit entrant dans un objet River
-	runoff_linear	volume d'eau reçu par ruissellement sur un objet Linear
sconc_surface_hedge_o	up	concentration en solutés sur la surface supérieure d'un objet Hedge
sconc_surface_hedge_o	down	concentration en solutés sur la surface inférieure d'un objet Hedge
scumul_surface_hedge_o	up	quantité de solutés cumulée stockée sur la surface supérieure d'un objet Hedge

scumul_surface_hedge_o	down	quantité de solutés cumulée stockée sur la surface inférieure d'un objet Hedge
sflw_surface_hedge_o	up	flux de solutés en surface (ruissellement) sur la surface supérieure d'un objet Hedge
sflw_surface_hedge_o	down	flux de solutés en surface (ruissellement) sur la surface inférieure d'un objet Hedge
smass_column_o	smass_total	masse totale de solutés dans chaque cellule numérique d'un objet Column
smass_column_o	smass_diss	masse de solutés dissous dans chaque cellule numérique d'un objet Column
smass_column_o	smass_ads	masse de solutés adsorbés dans chaque cellule numérique d'un objet Column
smass_column_hedgeø	smass_total_up	masse totale de solutés dans chaque cellule numérique de la colonne supérieure d'un objet Hedge
smass_column_hedgeø	smass_diss_up	masse de solutés dissous dans chaque cellule numérique de la colonne supérieure d'un objet Hedge
smass_column_hedgeø	smass_ads_up	masse de solutés adsorbés dans chaque cellule numérique de la colonne supérieure d'un objet Hedge
smass_column_hedgeø	smass_total_down	masse totale de solutés dans chaque cellule numérique de la colonne inférieure d'un objet Hedge
smass_column_hedgeø	smass_diss_down	masse de solutés dissous dans chaque cellule numérique de la colonne inférieure d'un objet Hedge
smass_column_hedgeø	smass_ads_down	masse de solutés adsorbés dans chaque cellule numérique de la colonne inférieure d'un objet Hedge
smass_surface_hedge_o	smass_total_up	masse totale de solutés sur la surface supérieure d'un objet Hedge
smass_surface_hedge_o	smass_diss_up	masse de solutés dissous sur la surface supérieure d'un objet Hedge
smass_surface_hedge_o	smass_ads_up	masse de solutés adsorbés sur la surface supérieure d'un objet Hedge

smass_surface_hedge_o	smass_total_down	masse totale de solutés sur la surface inférieure d'un objet Hedge
smass_surface_hedge_o	smass_diss_down	masse de solutés dissous sur la surface inférieure d'un objet Hedge
smass_surface_hedge_o	smass_ads_down	masse de solutés adsorbés sur la surface inférieure d'un objet Hedge
smass_surface_o	smass_total	masse totale de solutés sur un objet Surface
smass_surface_o	smass_diss	masse de solutés dissous sur un objet Surface
smass_surface_o	smass_ads	masse de solutés adsorbés sur un objet Surface
smass_linear_o	smass_total	masse totale de solutés sur un objet Linear
smass_linear_o	smass_diss	masse de solutés dissous sur un objet Linear
smass_linear_o	smass_ads	masse de solutés adsorbés sur un objet Linear
soil_profile_o	bd	masse volumique apparente dans chaque horizon d'un objet SoilType
soil_profile_o	dis	dispersivité dans chaque horizon d'un objet SoilType
soil_profile_o	dx	profondeur de chaque cellule numérique par rapport à la surface d'un objet SoilType
soil_profile_o	he	pression d'entrée d'air dans chaque cellule numérique d'un objet SoilType
soil_profile_o	Kads	coefficient d'adsorption pour chaque soluté simulé dans chaque horizon d'un objet SoilType
soil_profile_o	Ksat	conductivité hydraulique à saturation dans chaque horizon d'un objet SoilType
soil_profile_o	lambda	coefficient de Brooks and Corey dans chaque horizon d'un objet SoilType
soil_profile_o	nads	coefficient d'adsorption pour chaque soluté simulé dans chaque horizon d'un objet SoilType
soil_profile_o	nb_soils	nombre d'horizons présents dans un objet SoilType

soil_profile_o	p	porosité dans chaque horizon d'un objet SoilType
soil_profile_o	position_int	position des interfaces entre les différents horizons d'un objet SoilType
soil_profile_o	thr	teneur en eau résiduelle dans chaque horizon d'un objet SoilType
soil_profile_o	ths	teneur en eau à saturation dans chaque horizon d'un objet SoilType
soil_profile_o	x	épaisseur de chaque cellule numérique d'un objet SoilType
solute_o	halflife_time	temps de demi-vie un soluté
solute_o	metabolite_idx	indice du métabolite associé à un soluté
solute_o	occurence_frac	fraction d'apparition du métabolite associé au soluté donné en TAG
solutecumul_column_hedge	simput_cumul_up	quantité de solutés cumulée appliquée sur la colonne supérieure d'un objet Hedge
solutecumul_column_hedge	simfil_cumul_up	quantité de solutés cumulée infiltrée sur la colonne supérieure d'un objet Hedge
solutecumul_column_hedge	sodrain_cumul_up	quantité de solutés cumulée drainée sur la colonne supérieure d'un objet Hedge
solutecumul_column_hedge	slatflw_cumul_up	quantité de solutés cumulée reçue en flux latéral de subsurface sur la colonne supérieure d'un objet Hedge
solutecumul_column_hedge	sodeg_cumul_up	quantité de solutés cumulée dégradée sur la colonne supérieure d'un objet Hedge
solutecumul_column_hedge	sometacumul_up	quantité de solutés cumulée apparue sous la forme de métabolites sur la colonne supérieure d'un objet Hedge
solutecumul_column_hedge	simput_cumul_down	quantité de solutés cumulée appliquée sur la colonne inférieure d'un objet Hedge
solutecumul_column_hedge	simfil_cumul_down	quantité de solutés cumulée infiltrée sur la colonne inférieure d'un objet Hedge

solutecumul_column_hedge	sdrain_cumul_down	quantité de solutés cumulée drainée sur la colonne inférieure d'un objet Hedge
solutecumul_column_hedge	slatflw_cumul_down	quantité de solutés cumulée reçue en flux latéral de subsurface sur la colonne inférieure d'un objet Hedge
solutecumul_column_hedge	sdeg_cumul_down	quantité de solutés cumulée dégradée sur la colonne inférieure d'un objet Hedge
solutecumul_column_hedge	smeta_cumul_down	quantité de solutés cumulée apparue sous la forme de métabolites sur la colonne inférieure d'un objet Hedge
solutecumul_HU_o	sinput_cumul	quantité de solutés cumulée appliquée sur un objet HU
solutecumul_HU_o	sinfil_cumul	quantité de solutés cumulée infiltrée sur un objet HU
solutecumul_HU_o	sdrain_cumul	quantité de solutés cumulée drainée sur un objet HU
solutecumul_HU_o	slatflw_cumul	quantité de solutés cumulée reçue en flux latéral de subsurface dans un objet HU
solutecumul_HU_o	sdeg_cumul	quantité de solutés cumulée dégradée dans un objet HU
solutecumul_HU_o	smeta_cumul	quantité de solutés cumulée apparue sous la forme de métabolites dans un objet HU
solutecumul_linear_o	sdeg_cumul	quantité de solutés cumulée dégradée dans un objet Linear
solutecumul_linear_o	splat_cumul	quantité de solutés cumulée reçue en flux latéral de subsurface dans un objet Linear
solutecumul_linear_o	slongiflw_cumul	quantité de solutés cumulée reçue en flux longitudinal dans un objet Linear
solutecumul_linear_o	smeta_cumul	quantité de solutés cumulée apparue sous la forme de métabolites dans un objet Linear
solutecumul_linear_o	srunoff_cumul	quantité de solutés cumulée reçue par ruissellement sur un objet Linear

solute_cumul_linear_o	ssource_cumul	quantité de solutés cumulée reçue d'un terme source sur un objet Linear
—	isotype_soil	type d'isotherme d'adsorption appliqué pour chaque soluté modélisé pour chaque horizon de sol d'un objet SoilType
—	solute_runoff_linear	masse de solutés reçue par ruissellement sur un objet Linear
surface_o	manning	coefficient de Manning sur un objet Surface
surface_o	hpond	hauteur de ponding maximale autorisée sur un objet Surface
surface_o	slope	pente moyenne sur un objet Surface
surface_o	adsorp_thick	épaisseur de la bande d'adsorption dans le sol d'un objet Surface
surface_o	bd	masse volumique apparente dans le sol d'un objet Surface
surface_o	ths	teneur en eau à saturation dans le sol d'un objet Surface
surface_o	isotype	isotherme d'adsorption utilisé pour chaque soluté modélisé dans le sol d'un objet Surface
surface_o	Kads	coefficient d'adsorption pour chaque soluté modélisé dans le sol d'un objet Surface
surface_o	nads	coefficient d'adsorption pour chaque soluté modélisé dans le sol d'un objet Surface
—	lateral_connection	table des connexions latérales
—	lateral_connection_complete	table des connexions latérales et de leurs propriétés
—	longitudinal_connection	table des connexions longitudinales donnée avec l'élément amont et aval pour chaque ligne
—	connect_to_HU	liste des connexions latérales à l'aval avec des objets de type HU
—	connect_ditch_ditch	liste des connexions longitudinales à l'aval avec des objets de type Ditch
—	connect_river_river	liste des connexions longitudinales à l'aval avec des objets de type River

–	weight_runoff_HU	table contenant la proportion de ruissellement reçu par chaque objet connecté en latéral à l’aval (tous types d’objets)
–	connect_to_reach_HU	table des connexions latérales à l’aval avec des objets de type Reach et avec éventuellement un HU à l’aval du tronçon repéré
–	connect_to_bcs	liste des connexions longitudinales à l’aval avec des objets de type BCS
–	connect_to_tanklat	liste des connexions latérales à l’aval avec des objets de type SurfaceTank
–	connect_to_tanklon	liste des connexions longitudinales à l’aval avec des objets de type SurfaceTank
–	list_bcs	liste des éléments de type BCS sur le bassin à modéliser
–	list_HU	liste des éléments de type HU sur le bassin à modéliser
–	list_linear	liste des éléments de type Linear sur le bassin à modéliser
–	list_hedge	liste des éléments de type Hedge sur le bassin à modéliser
–	list_river	liste des éléments de type River sur le bassin à modéliser
–	list_ditch	liste des éléments de type Ditch sur le bassin à modéliser
–	alltime_vect	liste de tous les temps de simulation
–	wcontent_column	volume d’eau dans chaque cellule numérique d’un objet de type Column
–	saturation_column	saturation dans chaque cellule numérique d’un objet de type Column
–	head_colmun	pression dans chaque cellule numérique d’un objet de type Column
–	wqlat_column	flux d’eau latéral reçu dans chaque cellule numérique d’un objet de type Column
–	soluteconc_surface	concentration en solutés sur un objet Surface
–	scumul_surface	quantité de solutés cumulée stockée sur un objet Surface

–	sflw_surface	flux de solutés en surface (ruissellement) sur un objet Surface
–	sconc_dissolved_linear	concentration en solutés sur un objet Linear
–	wcontent_bcs	volume d'eau contenu dans un objet BCS
–	smass_subsurf_bcs	masse de solutés en subsurface dans un objet BCS
–	smass_surf_bcs	masse de solutés en surface dans un objet BCS
–	squat_bcs	flux latéral de solutés depuis/vers un objet BCS
–	squat_linear	flux latéral de solutés depuis/vers un objet Linear
–	solute_runoff_linear	flux de solutés intercepté depuis du ruissellement amont dans un objet Linear
–	squat_column	flux de solutés latéral reçu dans chaque cellule numérique d'un objet de type Column
–	sconc_column	concentration en solutés dans chaque cellule numérique d'un objet de type Column
–	sconcderv_column	dérivée de la concentration en solutés par rapport à ... dans chaque cellule numérique d'un objet de type Column
vegetation_o	crop	coefficient cultural d'un objet Vegetation
vegetation_o	extinc	coefficient d'extinction d'un objet Vegetation
vegetation_o	F10	...
vegetation_o	h_transp	...
vegetation_o	LAI	Leaf Area Index d'un objet Vegetation
vegetation_o	lamnbdaroot	...
vegetation_o	transp	...
vegetation_o	Zr	...
watercumul_HU_o	wevapo_cumul	volume d'eau cumulé évaporé depuis un objet HU
watercumul_HU_o	wextract_cumul	volume d'eau retiré par extraction racinaire d'un objet HU

watercumul_HU_o	wprec_cumul	volume d'eau cumulé reçu par précipitations sur objet HU
watercumul_HU_o	wdrain_cumul	volume d'eau cumulé drainé d'un objet HU
watercumul_HU_o	winfl_cumul	volume d'eau cumulé infiltré sur un objet HU
watercumul_HU_o	wqlat_cumul	volume d'eau cumulé reçu/retiré par transferts latéraux depuis un objet HU
watercumul_linear_o	wevap_cumul	volume d'eau cumulé évaporé depuis un objet Linear
watercumul_linear_o	wqlat_cumul	volume d'eau cumulé reçu/retiré par transferts latéraux dans un objet Linear
watercumul_linear_o	wrunoff_cumul	volume d'eau cumulé reçu/retiré par ruissellement dans un objet Linear
watercumul_linear_o	wlongiflw	volume d'eau cumulé reçu/retiré par transferts longitudinaux dans un objet Linear
watercumul_linear_o	wprec_cumul	volume d'eau cumulé reçu par précipitations sur un objet Linear
watercumul_linear_o	wsource_cumul	volume d'eau cumulé reçu par un terme source sur un objet Linear
watercumul_linear_o	splat_cumul	solutés cumulés reçu/retirés par transferts latéraux sur un objet Linear
wcontent_surface_hedge_o	down	volume d'eau sur la surface inférieure d'un objet Hedge
wcontent_surface_hedge_o	up	volume d'eau sur la surface supérieure d'un objet Hedge
wintercept_hedge_o	up	volume d'eau contenu dans le réservoir d'interception par la végétation sur la partie supérieure d'un objet Hedge
wintercept_hedge_o	down	volume d'eau contenu dans le réservoir d'interception par la végétation sur la partie inférieure d'un objet Hedge
wroff_cumul_surface_hedge_o	up	volume d'eau cumulé reçu/retiré par ruissellement sur la surface supérieure d'un objet Hedge

wroff_cumul_surface_hedge_owndown		volume d'eau cumulé reçu/retiré par ruissellement sur la surface inférieure d'un objet Hedge
wroff_flw_surface_hedge_oup		flux d'eau reçu/retiré par ruissellement sur la surface supérieure d'un objet Hedge
wroff_flw_surface_hedge_owndown		flux d'eau reçu/retiré par ruissellement sur la surface inférieure d'un objet Hedge
–	weight_surface	volume d'eau sur un objet Surface
–	wintercept_vegetation	volume d'eau contenu dans le réservoir d'interception par la végétation
–	wqlat_bcs	volume d'eau reçu/retiré par transfert latéraux de subsurface d'un objet BCS
–	wqlat_linear	volume d'eau reçu par transfert latéraux de subsurface sur un objet Linear
–	wroff_cumul_surface	volume d'eau cumulé reçu/retiré par ruissellement sur un objet Surface
–	wroff_flw_surface	flux d'eau reçu/retiré par ruissellement sur un objet Surface

# Annexe B

## Algorithme de Runge-Kutta au second et quatrième ordre

L'algorithme est décrit dans le chapitre 6 de Press et al. (1992). Les méthodes de Runge-Kutta sont des schémas numériques à un pas qui permettent de résoudre les équations différentielles ordinaires. Elles font partie des méthodes les plus populaires de part leur facilité de mise en œuvre et leur précision ([citer biblio/Autre/Resolution\\_numerique](#)).

### B.1 Introduction

On considère une équation différentielle ordinaire du premier ordre du type :

$$\begin{cases} \frac{dy}{dt} = f(t, y(t)), & 0 \leq t \leq T \\ y(0) = y_0 \end{cases} \quad (\text{B.1})$$

où  $y(t)$  est la fonction que l'on cherche,  $y_0$  sa valeur initiale et  $f$  une fonction connue (suffisamment régulière pour que l'existence et l'unicité de la solution ne pose pas de problème). Tout comme dans la méthode d'Euler, celles de Runge-Kutta sont des schémas numériques à un pas basées sur la discrétisation de la variable  $t$ . On note  $h$  ce pas et  $y_n$  la valeur approchée de  $y(t_n)$  pour les différents instants ( $t_n = nh$ ). En intégrant l'équation différentielle entre  $t_n$  et  $t_{n+1}$ , on a la relation :

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_n+h} f(t, y(t)) dt \quad (\text{B.2})$$

et l'idée est d'approcher cette intégrale plus précisément que ne le fait la méthode d'Euler.

### B.2 Runge-Kutta d'ordre 2

La méthode de Runge-Kutta d'ordre 2 consiste à approximer l'intégrale B.2 en calculant l'aire d'un trapèze au lieu de celle d'un rectangle comme fait dans la méthode d'Euler (voir Figure 70).

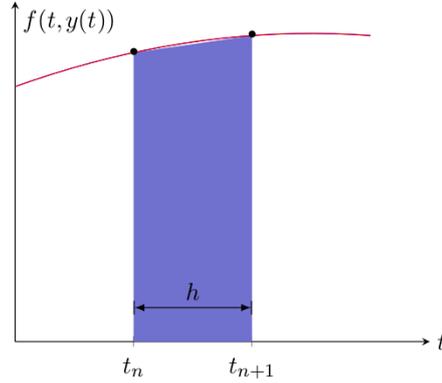


FIGURE 70 – Méthode du trapèze, tirée de [ref runge-kutta-methode](#)

La méthode du trapèze consiste en l'approximation suivante :

$$\int_a^b f(x) dx \simeq \frac{b-a}{2} [f(a) + f(b)] \quad (\text{B.3})$$

et appliquée à l'intégrale B.2, cela donne :

$$\int_{t_n}^{t_{n+h}} f(t, y(t)) dt \simeq \frac{h}{2} [f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+1}))] \quad (\text{B.4})$$

Cette intégrale dépendant des valeurs de  $y_n$  et de  $y_{n+1}$ , la méthode est implicite. On utilise la méthode d'Euler (explicite) afin d'estimer la valeur de  $y_{n+1}$  qui intervient dans  $f(t_{n+1}, y(t_{n+1}))$  (voir [ref intro B.1](#) pour plus de détails) :

$$y_{n+1} = y_n + hf(t_n, y_n) \quad (\text{B.5})$$

Appliqué à B.4, on obtient le schéma itératif suivant :

$$y_{n+1} = y_n + h\left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right) \text{ avec } \begin{cases} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + h, y_n + hk_1) \\ y_0 = y(0) \end{cases} \quad (\text{B.6})$$

L'erreur produite à chaque étape se comporte comme  $C^{te}h^3$  de sorte que l'erreur globale pour une durée  $T$  se comporte comme  $h^2$ .

### B.3 Runge-Kutta d'ordre 4

La méthode de Runge-Kutta d'ordre 4 se base sur la méthode de Simpson au lieu de la méthode des trapèzes pour le calcul de l'intégrale B.2. Elle consiste à remplacer la fonction intégrée par une parabole passant par les points extrêmes et le point milieu. On a :

$$\int_a^b f(x) dx \simeq \frac{b-a}{6} [f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)] \quad (\text{B.7})$$

Appliquée à l'intégrale B.2, cela donne

$$\int_{t_n}^{t_{n+1}} f(t, y(t)) dt \simeq \frac{h}{6} [f(t_n, y(t_n)) + 4f(t_{n+1/2}, y(t_{n+1/2})) + f(t_{n+1}, y(t_{n+1}))] \quad (\text{B.8})$$

d'où la relation

$$y_{n+1} = y_n + \frac{h}{6}[f(t_n, y_n) + 4f(t_{n+1/2}, y_{n+1/2}) + f(t_{n+1}, y_{n+1})] \quad (\text{B.9})$$

L'équation présente encore 2 inconnues :  $y_{n+1/2}$  et  $y_{n+1}$ . Pour rendre le schéma explicite, il faut estimer  $4f(t_{n+1/2}, y_{n+1/2})$  et  $f(t_{n+1}, y_{n+1})$  à partir de  $y_n$ ,  $t_n$  et  $h$ .  $4f(t_{n+1/2}, y_{n+1/2})$  peut s'exprimer comme  $2f(t_{n+1/2}, y_{n+1/2}) + 2f(t_{n+1/2}, y_{n+1/2})$ . Dans le premier terme, on remplace  $y_{n+1/2}$  par sa valeur déduite de la méthode d'Euler explicite ; à savoir  $y_{n+1/2}^a = y_n + \frac{h}{2}f(t_n, y_n)$ . Dans le deuxième terme, on remplace  $y_{n+1/2}$  par sa valeur déduite de la méthode d'Euler implicite :  $y_{n+1/2}^b = y_n + \frac{h}{2}f(t_{n+1/2}, y_{n+1/2})$  que l'on va approcher par  $y_n + \frac{h}{2}f(t_{n+1/2}, y_{n+1/2}^a)$ . Pour résumer, on a :

$$4f(t_{n+1/2}, y_{n+1/2}) = 2k_2 + 2k_3 \text{ avec } \begin{cases} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1) \\ k_3 = f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2) \end{cases} \quad (\text{B.10})$$

Quant au terme  $f(t_{n+1}, y_{n+1})$ , on l'approche en estimant  $y_{n+1}$  par la méthode du point milieu, c'est-à-dire en appliquant la méthode du rectangle au milieu :

$$y_{n+1} \simeq y_n + hf(t_{n+1/2}, y_{n+1/2}) \simeq y_n + hf(t_{n+1/2}, y_{n+1/2}^b) \quad (\text{B.11})$$

Finalement, on obtient le schéma explicite, dit de Runge-Kutta d'ordre 4 :

$$y_{n+1} = y_n + h[\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4] \text{ avec } \begin{cases} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1) \\ k_3 = f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2) \\ k_4 = f(t_n + h, y_n + hk_3) \end{cases} \quad (\text{B.12})$$

## B.4 Pas de temps adaptatif pour la méthode de Runge-Kutta, formules emboîtées de Runge-Kutta

inspiré en partie de <http://www.f-legrand.fr/scidoc/docmml/numerique/rungekutta/rkemboit/rkemboit.html>

Press et al. (1992) décrit une amélioration de la méthode de Runge-Kutta (RK) d'ordre 4. Il s'agit de la méthode de Runge-Kutta à formules emboîtées qui est utilisée pour adapter le pas  $h$ . Il s'agit d'utiliser 2 formules de RK simultanément pour chaque pas, l'une à l'ordre  $p$  et l'autre à l'ordre  $p + 1$ . Pour la méthode RK d'ordre 5, cela donne :

$$y_{n+1} = y_n + c_1k_1 + c_2k_2 + c_3k_3 + c_4k_4 + c_5k_5 + c_6k_6 + O(h^6) \quad (\text{B.13})$$

La formule emboîtée d'ordre 4 donne :

$$y_{n+1}^* = y_n + c_1^*k_1 + c_2^*k_2 + c_3^*k_3 + c_4^*k_4 + c_5^*k_5 + c_6^*k_6 + O(h^5) \quad (\text{B.14})$$

L'intérêt de ces 2 formules emboîtées est de permettre une estimation de l'erreur locale pour la formule d'ordre 4 :

$$\Delta = y_{n+1} - y_{n+1}^* = \sum_{i=1}^6 (c_i - c_i^*) k_i \quad (\text{B.15})$$

Press et al. (1992) préconise l'utilisation des coefficients de Cash et Karp [ref!](#) pour estimer les coefficients  $c_i$  et  $c_i^*$ . Par rapport à la formule d'ordre 4 classique, la méthode nécessite 2 évaluations en plus. Cependant, l'évaluation de l'erreur est indispensable pour adapter le pas  $h$  à la précision souhaitée.

L'adaptation du pas de temps consiste ensuite à modifier le pas  $h$  de manière à maintenir l'erreur inférieure à une limite fixée.

⚠ Le contrôle de l'erreur est local (à chaque pas) et l'erreur cumulée après  $N$  pas ne peut être contrôlée directement.

L'erreur  $\Delta$  se comporte comme  $h^5$  d'après B.13-B.14. Si l'on considère un pas  $h_1$  qui produit une erreur  $\Delta_1$  et que l'on souhaite déterminer le pas  $h_0$  qui produit une erreur désirée  $\Delta_0$ , on a la relation :

$$h_0 = h_1 \frac{\|\Delta_0\|^{0.2}}{\|\Delta_1\|} \quad (\text{B.16})$$

L'équation B.16 indique si le pas  $h$  doit être diminué ou augmenté : Si  $\|\Delta_1\|$  est supérieur à  $\|\Delta_0\|$ , l'équation nous dit comment le pas doit être diminué. Le cas échéant, elle nous indique comment le pas peut être augmenté pour le prochain pas. L'erreur désirée  $\Delta_0$  est en réalité un vecteur d'erreurs désirées pour chacune des équations (dans notre cas pour chacun des brins) et on adapte le pas en fonction du "pire des brins". La norme utilisée est alors le plus souvent  $\|\Delta_0\| = \text{Max}(|\Delta_0(i)|)$

L'erreur  $\Delta_0(i)$  doit être définie avec soin et dépend en grande partie de l'application visée. Elle peut être exprimée comme la somme d'une erreur absolue et d'une erreur relative :  $\Delta_0 = \varepsilon_a + \varepsilon_r |y_n|$  où  $\varepsilon_a$  est l'erreur absolue et  $\varepsilon_r$  l'erreur relative. Si l'on souhaite seulement une erreur relative, l'on posera la première très faible mais non nul pour éviter une division par zéro lorsque  $y_n$  s'annule. On peut par exemple également poser pour l'erreur relative à la  $i$ -ème équation :

$$\Delta_0^i = \varepsilon \times y_{scal}^i \text{ avec } y_{scal}^i = |y_i| + |h * \frac{dy}{dx}| \quad (\text{B.17})$$

[Repréciser détermination de  \$\Delta\_0\$  + Une autre possibilité est décrite par Press et réutilisée dans l'algorithme de River1D : à redétailler](#)

# Annexe C

## Données d'entrée du modèle PESHMELBA

### C.1 LANDSCAPE : organisation du paysage

#### liste\_UH.csv

Ce fichier csv contenant la liste des unités homogènes ainsi que leurs propriétés dans l'ordre suivant :

- Idn element** : indice de l'élément
- Type** : type d'unité homogène (1 si parcelle, 2 si zone tampon)
- type\_soil** : indice du type de sol. Cet indice doit être le même que celui indiqué dans le titre du fichier soilX\_propertiesXX.in contenu dans le répertoire SOIL\_PROPERTIES correspondant
- surface** : surface de l'élément (cm<sup>2</sup>)
- centroid\_alti** : altitude du centroïde de l'élément (cm depuis un niveau de référence commun à tous les éléments)
- centroid\_x** : longitude du centroïde de l'élément (cm depuis un niveau de référence commun à tous les éléments)
- centroid\_y** : latitude du centroïde de l'élément (cm depuis un niveau de référence commun à tous les éléments)
- mean\_slope** : pente moyenne sur l'élément (⚠ donné en m.m<sup>-1</sup> ce qui signifie qu'une pente de 2% sera rentrée sous la forme 0.02 m/m dans le fichier d'entrée.)
- adsorp\_soil\_thick** : épaisseur de sol sur laquelle est calculée l'adsorption (cm)
- bulk\_density** : masse volumique en surface (i.e. dans l'épaisseur de sol en surface où est calculée d'adsorption, g.cm<sup>-3</sup>)
- thetas** : teneur en eau à saturation en surface (i.e. dans l'épaisseur de sol en surface où est calculée d'adsorption, cm<sup>3</sup>.cm<sup>-3</sup>)
- Hpond** : hauteur de ponding sur l'élément (cm)

#### linear.csv

Ce fichier décrit les propriétés des tronçons élémentaires dans l'ordre suivant :

- Idn element** : indice du tronçon élémentaire
- Type** : type de tronçon élémentaire (1 si fossé, 2 si rivière, 3 si haie/talus)
- Lon** : longitude du centroïde de l'élément (cm depuis un niveau de référence commun à tous les éléments)
- Lat** : latitude du centroïde de l'élément (cm depuis un niveau de référence commun à tous les éléments)
- Altitude** : altitude du centroïde de l'élément (cm depuis un niveau de référence commun à tous les éléments)
- Length** : longueur du tronçon élémentaire (cm)

## ditch.csv

Ce fichier contient les paramètres des tronçons de fossé dans l'ordre suivant :

- Idn element** : indice du tronçon de fossé
- Depth** : profondeur du fossé (cm)
- Manning\_Coeff** : coefficient de Manning dans le fossé. Il est donné en  $s.m^{1/3}$  car il s'agit de l'unité classiquement utilisée dans la littérature et qu'il sera plus simple d'effectuer la conversion par la suite que de contraindre les utilisateurs à fournir ce paramètre en  $h.cm^{1/3}$ .
- Bottom\_width** : largeur du fond du fossé (cm)
- Bank\_angle** : fruit des berges (deg)
- Slope** : pente longitudinale ( $\Delta$  donnée en  $m.m^{-1}$  ce qui signifie qu'une pente de 2% sera rentrée sous la forme 0.02 m/m dans le fichier d'entrée.)
- Initial\_water\_content** : volume d'eau dans l'élément en début de simulation ( $cm^3$ ) **C'est une condition initiale, il ira à termes dans un fichier à part dans le répertoire INITIAL CONDITIONS**
- adsorp\_soil\_thick** : épaisseur de sol sur laquelle est calculée l'adsorption dans le fond du fossé (cm)
- bulk\_density** : masse volumique dans le fond du fossé (i.e. dans l'épaisseur de sol en surface où est calculée d'adsorption,  $g.cm^{-3}$ )
- thetas** : teneur en eau à saturation dans le fond du fossé (i.e. dans l'épaisseur de sol en surface où est calculée d'adsorption,  $cm^3.cm^{-3}$ )
- Hpond** : hauteur de ponding dans le fond du fossé (cm)

## river.csv

Ce fichier décrit les tronçons de rivière grâce aux paramètres suivants :

- Idn element** : indice du tronçon de rivière
- Depth** : profondeur de la rivière (cm)
- Manning\_Coeff** : coefficient de Manning dans la rivière. Il est donné en  $s.m^{1/3}$  car il s'agit de l'unité classiquement utilisée dans la littérature et qu'il sera plus simple d'effectuer la conversion par la suite que de contraindre les utilisateurs à fournir ce paramètre en  $h.cm^{1/3}$ .

- Bottom\_width** : largeur du lit de la rivière (cm)
- Bank\_angle** : fruit des berges (deg)
- Slope** : pente longitudinale ( $\Delta$  donnée en  $m.m^{-1}$  ce qui signifie qu'une pente de 2% sera rentrée sous la forme 0.02 m/m dans le fichier d'entrée.)
- Initial\_water\_content** : volume d'eau dans l'élément en début de simulation ( $cm^3$ ) **C'est une condition initiale, il ira à termes dans un fichier à part dans le répertoire INITIAL CONDITIONS**
- Input flow** potentiel débit en amont de la rivière ( $cm^3/h$ )
- adsorp\_soil\_thick** : épaisseur de sol sur laquelle est calculée l'adsorption dans le fond du fossé (cm)
- bulk\_density** : masse volumique dans le fond du fossé (i.e. dans l'épaisseur de sol en surface où est calculée d'adsorption,  $g.cm^{-3}$ )
- thetas** : teneur en eau à saturation dans le fond du fossé (i.e. dans l'épaisseur de sol en surface où est calculée d'adsorption,  $cm^3.cm^{-3}$ )
- Hpond** : hauteur de ponding dans le fond du fossé (cm)
- Dist\_bed\_imperm** : distance entre le fond du lit de la rivière et l'aquifère (cm)
- ksat\_bed** : conductivité hydraulique à saturation dans le lit de rivière ( $cm/h$ )

## hedge.csv

Ce fichier décrit les tronçons de haie/talus grâce aux paramètres suivants :

- Idn element** : indice du tronçon haie/talus
- Vegetation\_idx** : indice du type de végétation présent sur la haie/talus. Cet indice doit être le même que ceux indiqués dans les titres des fichiers LAI\_vegetalX.in, roots\_vegetalX.in et vegetalX.in (répertoire VEGETAL) correspondants.
- Soil\_Type** : Indice du type de sol. Cet indice doit être le même que celui indiqué dans le titre du fichier soilX\_propertiesXX.in contenu dans le répertoire SOIL\_PROPERTIES correspondant
- Preferential infilt. Conductivity** : Taux d'infiltration utilisé pour calculer l'infiltration dans les macropores ( $cm/h$ )
- Upstream width** : largeur de la partie amont de la haie (cm)
- Downstream width** : largeur de la partie aval de la haie (cm)
- Hedge\_height** : hauteur du talus potentiel séparant les partie amont et aval de l'élément (cm). S'il s'agit d'une simple haie, cette hauteur est fixée à zéro.
- Column\_height** : hauteur des colonnes de sol représentant la subsurface des parties amont et aval de la haie/talus (cm).
- Angle\_to\_levelline** : Inclinaison longitudinale de la haie par rapport à la ligne de niveau (deg).
- Longitudinal\_slope** Pente longitudinale de l'élément ( $m.m^{-1}$ ).
- Lateral\_slope\_UP** Pente transversale dans la partie amont de l'élément ( $m.m^{-1}$ ).
- Lateral\_slope\_DOWN** Pente transversale dans la partie aval de l'élément ( $m.m^{-1}$ ).

- adsorp\_soil\_thick** : épaisseur de sol sur laquelle est calculée l'adsorption (cm)
- bulk\_density** : masse volumique dans l'épaisseur de sol en surface où est calculée d'adsorption,  $\text{g.cm}^{-3}$
- thetas** : teneur en eau à saturation dans l'épaisseur de sol en surface où est calculée d'adsorption,  $\text{cm}^3.\text{cm}^{-3}$
- Manning\_Coeff\_UP** : coefficient de Manning sur la partie amont de l'élément. Il est donné en  $\text{s.m}^{1/3}$  car il s'agit de l'unité classiquement utilisée dans la littérature et qu'il sera plus simple d'effectuer la conversion par la suite que de contraindre les utilisateurs à fournir ce paramètre en  $\text{h.cm}^{1/3}$ .
- Manning\_Coeff\_DOWN** : coefficient de Manning sur la partie aval de l'élément ( $\text{s.m}^{1/3}$ )

## bcs.csv

Ce fichier décrit les BCS grâce aux paramètres suivants :

- Idn element** : indice d'identification de l'élément. On rappelle que chaque élément de la simulation doit posséder un identifiant unique.
- Lon** : longitude de l'élément (cm depuis un niveau de référence commun à tous les éléments)
- Lat** : latitude (cm depuis un niveau de référence commun à tous les éléments)
- Altitude** : altitude (cm depuis un niveau de référence commun à tous les éléments)
- Updown** : indice indiquant la nature de l'élément BCS. Updown est égal à 0 si le BCS est à l'aval d'un élément surfacique et qu'il correspond à un terme **puits**. Il est égal à 1 s'il est à l'amont d'un BCS et qu'il s'agit d'un terme **source**.
- Type** : indice indiquant le type de condition limite en subsurface associée à l'élément BCS :
  - 0 : Flux nul
  - 1 : Colonne virtuelle
  - 2 : Flux constant défini par l'utilisateur
  - 3 : Surface de suintement (valable seulement pour un BCS de type 0 ; terme puit, à l'aval d'une UH)
- Subsurface BC** : Valeur du flux constant entrant ou sortant du BCS ( $\text{cm}^3/\text{h}$ ) dans le cas d'une CL de type 2. Une valeur arbitraire peut être renseignée dans le cas d'autre type de condition limite (0,1 ou 3).
 

 Pour les bcs situés en amont d'une UH, ce scalaire est **positif**. En cas de bcs situé à l'aval d'une UH, ce scalaire est **négatif**.
- Surface BC** : condition limite en surface associée à l'élément ( $\text{cm}^3/\text{h}$ ). Pour un élément situé à l'amont d'une UH, l'utilisateur spécifie la valeur du flux d'eau entrant en surface. On renseigne la valeur 0 en cas de flux nul. A l'aval, la valeur est fixée à 0 par défaut car le ruissellement sortant de l'UH est calculé à partir de ses caractéristiques et de la hauteur d'eau.

<u>time (h)</u>	<u>RO (cm<sup>3</sup>/h)</u>
0	0
1.99	0
3.01	1363201.2
4.03	14324688
5.01	4237884
5.97	4264992
6.99	1386100.8
7.92	0
11.99	0

FIGURE 71 – Exemple de fichier runoff\_bcsXXX.csv

- conc subsurface** : concentration en solutés dans le compartiment subsurface (g/cm<sup>3</sup>). Cet attribut n'a de sens que pour des éléments source puisque les termes puits reçoivent des quantités de solutés déterminées par les conditions de l'UH amont. Attention, il faut donner autant de valeurs qu'il y a de solutés simulés.
- conc surface** : concentration en solutés dans le compartiment surface (g/cm<sup>3</sup>) : Cet attribut n'a de sens que pour des éléments source puisque les termes puits reçoivent des quantités de solutés déterminées par les conditions de l'UH amont. Attention, il faut donner autant de valeurs qu'il y a de solutés simulés.
- Water Table Height** : Attribut utilisé lorsque le BCS est une colonne virtuelle. Cet attribut définit une profondeur de la nappe dans la colonne (cm, profondeur inférieure à 0, calculée depuis la surface (cm)).

### runoff\_bcsXXX.csv

Ce fichier doit être donné en entrée pour les éléments de type bcs de type source dont la condition limite en surface est donnée par un flux. Le fichier fournit une chronique de débits de ruissellement associés à des temps de simulation, à partir de t= 0, le début de la simulation (voir exemple Fig. 71).

### connection.csv :

Ce fichier décrit les lignes d'échanges latérales entre UH et TE (tronçons élémentaires) et contient les colonnes suivantes :

- From** : indice de l'élément à l'amont de la ligne d'échanges
- To** : indice de l'élément à l'aval de la ligne d'échanges
- Length** : longueur de la ligne d'échanges (cm)
- Lon** : longitude du milieu de la ligne d'échange (cm depuis un niveau de référence commun à tous les éléments)
- Lat** : latitude du milieu de la ligne d'échange (cm depuis un niveau de référence commun à tous les éléments)
- Altitude** : altitude du milieu de la ligne d'échange (cm depuis un niveau de référence commun à tous les éléments)

## connection\_longitudinal\_linear.csv :

Ce fichier contient une description des connexions longitudinales entre tronçons élémentaires. Ces dernières sont décrites en donnant en première colonne l'élément amont et en seconde colonne l'élément aval.

## C.2 INITIAL\_CONDITIONS : conditions initiales dans les éléments

Il sera nécessaire de déplacer tous les paramètres d'adsorption dans d'autres fichiers car ne concerne pas l'initialisation

### initial\_heads\_HU.csv

Ce fichier csv contient la pression capillaire (heads) dans chaque cellule numérique de chaque UH. Le fichier contient autant de lignes que d'UH et une ligne contient les éléments suivants :

Idn element	heads dans la cellule 0	heads dans la cellule 1	heads dans la cellule ...	heads dans la cellule n
-------------	-------------------------	-------------------------	---------------------------	-------------------------

On note que l'ordre des éléments est important, la cellule 0 correspond à celle qui est la plus proche de la surface, la cellule n correspond au fond du profil de sol.

### initial\_heads\_hedge.csv

Ce fichier csv contient la pression capillaire (heads) dans chaque cellule numérique de chaque élément de type haie/talus. Une haie/talus est composée de 2 colonnes de sol qui sont initialisées avec le même profil de pression. Le fichier contient autant de lignes que d'éléments et une ligne contient les éléments suivants :

Idn element	heads dans la cellule 0	heads dans la cellule 1	heads dans la cellule ...	heads dans la cellule n
-------------	-------------------------	-------------------------	---------------------------	-------------------------

On note que l'ordre des éléments est important, la cellule 0 correspond à celle qui est la plus proche de la surface, la cellule n correspond au fond du profil de sol.

### initial\_sm\_plot.csv

Ce fichier permet d'initialiser les quantités de solutés dans les UH ainsi que leurs paramètres d'adsorption et dégradation en surface. Il contient les informations suivantes :

- Column** : indice de l'UH
- Solute** : indice du soluté
- Surface isotype** code de l'isotherme à utiliser pour calculer l'équilibre d'adsorption en surface(sous la forme d'un code de 4 caractères "line", "freu" ou "none")
- Surface Kads** coefficient d'adsorption (linéaire ou de Freundlich, en  $\text{cm}^3\text{g}^{-n}$  attention aux unités : bien vérifier qu'on a ce système d'unités avant d'utiliser le Kd)

- Surface n coeff** si besoin, exposant de l'isotherme de Freundlich. Cette dernière case est à remplir même si on utilise un isotherme linéaire, dans ce dernier cas, elle ne sera juste pas utilisée.
- DT50** : temps de demi-vie (h)
- unit mass in soil layer** : sur n colonnes, masse de solutés initialement présente dans la colonne de sol (g)

### **initial\_sm\_hedge.csv**

Ce fichier permet d'initialiser les quantités de solutés dans les éléments de type haie/talus ainsi que leurs paramètres d'adsorption et dégradation en surface. Il contient les informations suivantes :

- Column** : indice du tronçon élémentaire
- Solute** : indice du soluté
- UPDOWN** : indice de la position (0 si colonne amont, 1 si colonne aval)
- Surface isotype** code de l'isotherme à utiliser pour calculer l'équilibre d'adsorption en surface(sous la forme d'un code de 4 caractères "line", "freu" ou "none")
- Surface Kads** coefficient d'adsorption (linéaire ou de Freundlich, en  $\text{cm}^{3n}\text{g}^n$   
**attention aux unités : bien vérifier qu'on a ce système d'unités avant d'utiliser le Kd**)
- Surface n coeff** si besoin, exposant de l'isotherme de Freundlich. Cette dernière case est à remplir même si on utilise un isotherme linéaire, dans ce dernier cas, elle ne sera juste pas utilisée.
- DT50** : temps de demi-vie (h)
- unit mass in soil layer** : sur n colonnes, masse de solutés initialement présente dans la colonne de sol (g)

### **initial\_water\_hedge.csv**

Ce fichier permet d'initialiser les volumes d'eau initialement présent dans les réservoirs de surface des éléments de type haie/talus et contient les colonnes suivantes :

- Idn element** : indice du tronçon de haie/talus
- Initial wat\_cont\_UP** : volume d'eau initial contenu dans le réservoir de la partie amont de la haie ( $\text{cm}^3$ )
- Initial wat\_cont\_DOWN** : volume d'eau initial contenu dans le réservoir de la partie aval de la haie ( $\text{cm}^3$ )

### **initial\_sm\_ditch.csv**

Ce fichier permet d'initialiser les quantités de solutés dans chaque fossé ainsi que leurs paramètres d'adsorption et de dégradation en relation avec le type de sol :

- id\_linear** : indice du linéaire
- id\_solute** : indice du soluté

- initial\_mass** : masse de solutés en début de simulation (g)
- isotype** : code de l'isotherme à utiliser pour calculer l'équilibre d'adsorption (sous la forme d'un code de 4 caractères "line", "freu" ou "none")
- Kads** : coefficient d'adsorption (linéaire ou de Freundlich, en  $\text{cm}^3\text{g}^{-1}$  **attention aux unités : bien vérifier qu'on a ce système d'unités avant d'utiliser le Kd**)
- n coeff** : si besoin, exposant de l'isotherme de Freundlich. Cette dernière case est à remplir même si on utilise un isotherme linéaire, dans ce dernier cas, elle ne sera juste pas utilisée.
- DT50** : temps de demi-vie (h)

### initial\_sm\_river.csv

Ce fichier permet d'initialiser les quantités de solutés dans chaque tronçon de rivière ainsi que leurs paramètres d'adsorption et de dégradation en relation avec le type de sol :

- id\_linear** : indice du linéaire
- id\_solute** : indice du soluté
- initial\_mass** : masse de solutés en début de simulation (g)
- isotype** : code de l'isotherme à utiliser pour calculer l'équilibre d'adsorption (sous la forme d'un code de 4 caractères "line", "freu" ou "none")
- Kads** : coefficient d'adsorption (linéaire ou de Freundlich, en  $\text{cm}^3\text{g}^{-1}$  **attention aux unités : bien vérifier qu'on a ce système d'unités avant d'utiliser le Kd**)
- n coeff** : si besoin, exposant de l'isotherme de Freundlich. Cette dernière case est à remplir même si on utilise un isotherme linéaire, dans ce dernier cas, elle ne sera juste pas utilisée.
- DT50** : temps de demi-vie (h)

## C.3 VÉGÉTAL : paramètre décrivant le couvert végétal

Pour le calcul de la composante d'extraction racinaire, on doit fournir en entrée un certain nombre de paramètres caractéristiques du couvert végétal. Les paramètres qui restent constants dans le temps sont regroupés dans un fichier **vegetalX.in**, X étant l'indice du couvert végétal. Ce fichier contient :

- Un coefficient cultural (à terme, on pourra envisager d'avoir un coefficient cultural qui varie au fil de la simulation)
- Un coefficient d'extinction
- Un paramètre  $\lambda$  pour le calcul de la fonction de compensation
- Des pressions capillaires (cm)  $h_1, h_2, h_3^1, h_3^2, h_4$
- Des transpirations potentielles (cm/h) **et non pas en mm/j comme dans l'article de Li and Zhang (2001) ou Varado (2004)** associées aux pressions capillaires  $h_3^1$  et  $h_3^2$

```

vegetal1.in ✖
Parameters for vegetal type 1 : soybean
Crop coefficient Kc
1.0
Extinction coefficient abl
0.4
lambda root
0.5
h1,h2,h3a,h3b,h4 (cm)
-10 -25 -500 -1000 -14000
e1,e2 (cm/h)
0.004166666666666667 0.020833333333333332

```

FIGURE 72 – Exemple de paramètres contenus dans le fichier vegetal1.in.

Un exemple d'un tel fichier est donné fig.72.

Des séries temporelles sont ensuite fournies pour l'indice de surface foliaire dans un fichier **LAI\_vegetalX.csv** et pour les paramètres de densité racinaire (Zr et F10) dans un fichier **roots\_vegetalX.csv**. On peut avoir des séries d'observations non régulières voire une unique observation (attention pour les paramètres racinaires, on ne peut pas avoir une observation de Zr à une date sans observation de F10 pour la même date ou inversement). On donne les dates d'observation (on suppose qu'il s'agit du début de l'observation) dans leur format complet : "YYYY-MM-DD" (les variations sont supposées être de l'ordre du jour plutôt que de l'heure, on estime donc inutile de préciser une heure d'observation). Des exemples des deux fichiers sont présentés fig. 73 et fig. 74.

	A	B	
1	<u>Ini_date</u>	<u>LAI</u>	
2	2017-07-17	0.5	
3	2017-07-19	0.8	
4	2017-07-21	1.0	
5	2017-07-23	2.0	
6	2017-07-25	3.0	
7	2017-07-27	2.8	
8	2017-07-29	2.8	
9			
10			

FIGURE 73 – Exemple de paramètres contenus dans le fichier LAI\_vegetal1.in.

	A	B	C	
1	<u>Ini_date</u>	<u>F10 (Fraction of the P</u>	<u>Zr (Max root depth cm)</u>	
2	2017-07-17	0.01		10
3	2017-07-19	0.01		10
4	2017-07-21	0.01		10
5	2017-07-23	0.01		10
6	2017-07-25	0.01		10
7				
8				
9				

FIGURE 74 – Exemple de paramètres contenus dans le fichier roots\_vegetal1.in.

Dans le fichier **crop\_itinerary.csv**, on recense l'indice du couvert végétal associé à chaque parcelle de manière à pouvoir faire le lien avec les fichiers ci-dessus. On donne

également une date (date de début) afin de pouvoir préciser plusieurs occupations du sol pour une même parcelle au cours d'une simulation (cf fig.75).

	A	B	C	D
1	Plot	Inj_date	Vegetal_type	
2	1	2017-07-17	1	
3	2	2017-07-17	1	
4	3	2017-07-17	1	
5	4	2017-07-17	1	
6	5	2017-07-17	1	
7	6	2017-07-17	1	
8	7	2017-07-17	1	
9				
10				

FIGURE 75 – Exemple de paramètres contenus dans le fichier crop\_itinerary.csv

## C.4 SOIL\_PROPERTIES : types de sol

Dans les tables liste\_UH.csv et hedge.csv, la colonne "type de sol" contient un indice qui renvoie à un ensemble de propriétés décrivant un type de sol et contenu dans un fichier du type **soilX\_propertiesXX.in**. Les paramètres de sol peuvent être décrits grâce aux modèles de Brooks and Corey ou de Schaap-Van Genuchten. Selon le cas, PESHMELBA lira les paramètres nécessaires dans le fichier soilX\_propertiesBC.in (Brooks and Corey) ou soilX\_propertiesVG.in (Schaap-Van Genuchten) où X est l'indice du type de sol décrit. La description des deux modèles est faite dans le chapitre 7 relatif à FRER1D.

### Utilisation du modèle de Brooks & Corey

Si on utilise le modèle de Brooks & Corey, le fichier décrivant les paramètres de sol contient les éléments suivants :

- nombre d'horizons contenus dans le profil de sol (scalaire : nt)
- profondeur des cellules numériques par rapport à la surface (comptée positivement) (cm) / double(1 :n)
- teneur en eau résiduelle  $\theta_r$  pour chaque horizon / double(1 :nt)
- teneur en eau à saturation  $\theta_s$  pour chaque horizon / double(1 :nt)
- pression d'entrée de l'air  $h_e$  pour chaque horizon (cm) / double(1 :nt)
- conductivité hydraulique à saturation  $K_s$  pour chaque horizon (cm/h) / double(1 :nt)
- paramètre de Brooks & Corey  $\lambda$  pour chaque horizon / double(1 :nt)
- porosité  $p$  pour chaque horizon / double(1 :nt)
- masse volumique du sol  $bd$  pour chaque horizon ( $\text{g.cm}^{-3}$ ) / double(1 :nt))
- dispersivité  $dis$  pour chaque horizon (cm) / double(1 :nt))
- position des interfaces entre types de sol /integer(1 :n). Par exemple, le vecteur 1 1 1 2 2 2 3 3 3 indique qu'il s'agit d'une colonne de sol divisée en 10 cellules numériques avec 3 horizons de sol et des interfaces entre les 4<sup>ème</sup> et 5<sup>ème</sup> cellules puis entre les 7<sup>ème</sup> et 8<sup>ème</sup> cellules.

- type d'isotherme d'adsorption à utiliser *isotype* : "none" si pas d'adsorption, "line" si isotherme linéaire, "freu" si isotherme de Freundlich) pour chaque soluté et pour chaque type de sol / character(4). On indique ces paramètres sous forme de matrice avec autant de lignes que de solutés et autant de colonnes que de types de sol. Par exemple pour une colonne avec 3 horizons de sol et 2 solutés :
 

line	line	line
freu	freu	freu

 permettra de décrire le comportement du lier soluté avec un isotherme linéaire dans tous les horizons de sol alors que le second sera décrit par un isotherme de Freundlich.
  
- coefficient d'adsorption linéaire ou de Freundlich  $Kd$  pour chaque soluté et pour chaque type de sol / double(1 :ns,1 :nt). On indique ces paramètres sous forme de matrice avec autant de lignes que de solutés et autant de colonnes que de types de sol
  
- exposant de l'isotherme d'adsorption de Freundlich  $n$  pour chaque soluté et pour chaque type de sol / double(1 :ns,1 :nt). On indique ces paramètres sous forme de matrice avec autant de lignes que de solutés et autant de colonnes que de types de sol. Pour avoir une structure générique, on demande aussi de remplir cette ligne pour un isotherme linéaire, même si elle ne sera pas utilisée par la suite.
  
- temps de demie-vie  $DT50$  pour chaque soluté et pour chaque type de sol (h) / double(1 :ns,1 :nt).

Un exemple de fichier soilx\_propertiesBC.in est présenté fig. 76.

```

soil1_propertiesBC.in ✖
Parameters for soil profile 1 (G1)
number of soil types nt
3
Depth to bottom of layer (/!\ cm)
0.5 1.0 3.0 5.0 10.0 30.0 50.0 70.0 90.0 110.0 130.0 150.0 170.0 190.0 210.0 250.0 300.0 350.0 400.0
Residual water content thetar
0.14 0.00 0.05
Saturation water content thetas
0.55 0.55 0.41
Scale parameter he (cm)
-160.12 -38.4 -48.9
Saturation hydraulic conductivity Ks (cm/h)
0.64 5.4 1.6
lambda
0.64197 0.2714 0.3793
p
1.0 1.0 1.0
bd
1.4 1.4 1.4
dis
20.0 20.0 20.0
position of interface : vector with type of soil
1 1 1 1 1 2 2 3 3 3 3 3 3 3 3 3 3 3 3
isotype ("none":no adsorption, "line":linear, "freu":Freundlich, "Lang":Langmuir, "Lali":Langmuir-Linear)
freu freu none
freu freu none
Coefficient of adsorption (Klin or Kfr, or Kla,...)
3.0977752 0.598288 0.0
3.7325652 0.720888 0.0
Exposant in Fr equation
1.25 1.25 1.25
1.11 1.11 1.11
Half-life time (h)
288 288 288
792 792 792

```

FIGURE 76 – Fichier de paramètres soil\_propertiesBC pour une colonne de sol à 3 horizons contenant et avec 2 solutés.

⚠ Les propriétés des types de sol sont transmises, pour chaque parcelle à PALM sous la forme d'objets dérivés dont la taille est définie en début de simulation et constante. Qu'il s'agisse d'un sol avec 3 horizons de sol ou un seul, ses caractéristiques seront transmises dans des tables de même taille. Pour cela, on initialise toutes les tables à partir du nombre d'horizons maximal rencontré dans les différents fichiers `soil_properties` (variable `max_hor`) quitte à remplir les lignes/colonnes restantes avec des NULL ou 9999 si besoin. Ceci est fait dans le script `init_CONF` lors de l'importation des données depuis les fichiers d'entrée du répertoire `SOIL_PROPERTIES`.

## Utilisation du modèle de Schaap - Van Genuchten

Si on utilise le modèle de Schaap - Van Genuchten, le fichier décrivant les paramètres de sol contient les éléments suivants :

- nombre d'horizons contenus dans le profil de sol (scalaire : nt)
- profondeur des cellules numériques par rapport à la surface (comptée positivement) (cm) / double(1 :n)
- teneur en eau résiduelle  $\theta_r$  pour chaque horizon / double(1 :nt)
- teneur en eau à saturation  $\theta_s$  pour chaque horizon / double(1 :nt)
- paramètre de Van Genuchten  $h_g$  pour chaque horizon (cm) / double(1 :nt)
- conductivité hydraulique à saturation totale  $K_s$  pour chaque horizon (cm/h) / double(1 :nt)
- conductivité hydraulique à saturation dans la matrice de sol  $K_x$  pour chaque horizon (cm/h) / double(1 :nt)
- porosité  $p$  pour chaque horizon / double(1 :nt)
- masse volumique du sol  $bd$  pour chaque horizon (g.cm<sup>-3</sup>) / double(1 :nt))
- dispersivité  $dis$  pour chaque horizon (cm) / double(1 :nt))
- position des interfaces entre types de sol /integer(1 :n). Par exemple, le vecteur 1 1 1 2 2 2 3 3 3 indique qu'il s'agit d'une colonne de sol divisée en 10 cellules numériques avec 3 horizons de sol et des interfaces entre les 4<sup>ème</sup> et 5<sup>ème</sup> cellules puis entre les 7<sup>ème</sup> et 8<sup>ème</sup> cellules.
- type d'isotherme d'adsorption à utiliser *isotype* : "none" si pas d'adsorption, "line" si isotherme linéaire, "freu" si isotherme de Freundlich) pour chaque soluté et pour chaque type de sol / character(4). On indique ces paramètres sous forme de matrice avec autant de lignes que de solutés et autant de colonnes que de types de sol. Par exemple pour une colonne avec 3 horizons de sol et 2 solutés :
 

line	line	line	
freu	freu	freu	permettra de décrire le comportement du 1ier soluté avec un isotherme linéaire dans tous les horizons de sol alors que le second sera décrit par un isotherme de Freundlich.
- coefficient d'adsorption linéaire ou de Freundlich  $Kd$  pour chaque soluté et pour chaque type de sol / double(1 :ns,1 :nt). On indique ces paramètres sous forme de matrice avec autant de lignes que de solutés et autant de colonnes que de types de sol

- exposant de l'isotherme d'adsorption de Freundlich  $n$  pour chaque soluté et pour chaque type de sol / double(1 :ns,1 :nt). On indique ces paramètres sous forme de matrice avec autant de lignes que de solutés et autant de colonnes que de types de sol. Pour avoir une structure générique, on demande aussi de remplir cette ligne pour un isotherme linéaire, même si elle ne sera pas utilisée par la suite.
  - temps de demie-vie  $DT50$  pour chaque soluté et pour chaque type de sol (h) / double(1 :ns,1 :nt).
  - paramètre de Van Genuchten  $mn$  pour chaque horizon / double(1 :nt)
- On note que le paramètre  $mn$  arrive en fin de fichier ce qui ne paraît pas cohérent avec l'ordre des paramètres présentés. Cet ordre a été implémenté afin de garder le maximum de lignes en commun avec les fichiers décrivant les sols avec le modèle de Brooks & Corey. Cela permet une lecture plus aisée par le script INIT\_CONF. Un exemple de fichier soilx\_properties.in est présenté fig. 77.

```
Parameters for soil profile 1 (Test with VG parameters)
number of soil types nt
2
Depth to bottom of layer (/ \ cm)
0.5 1.0 3.0 5.0 10.0 30.0 50.0 70.0 90.0 110.0 130.0 150.0 170.0 190.0 210.0 250.0 300.0 350.0 400.0
Residual water content thetar
0.14 0.14
Saturation water content thetas
0.4 0.5
hg (cm)
-20.0 -80.0
Saturation hydraulic conductivity Ks (cm/h)
8.0 0.4
Saturation hydraulic conductivity Kx for the soil matrix (cm/h)
2.0 0.2
p
1.0 1.0
bd
1.4 1.4
dis
20.0 20.0
position of interface : vector with type of soil
1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
isotype ("none":no adsorption, "line":linear, "freu":Freudlich, "Lang":Langmuir, "Lali":Langmuir-Linear)
freu freu
freu freu
Coefficient of adsorption (Klin or Kfr, or KLa,...)
3.0977752 3.0977752
3.7325652 3.7325652
Exposant in Fr equation
1.25 1.25
1.11 1.11
Halflife time (h)
288 288
792 792
Van Genuchten mn parameter
1.2 0.3
```

FIGURE 77 – Fichier de paramètres soil\_propertiesVG pour une colonne de sol à 2 horizons contenant 2 solutés.

## C.5 PRACTICES : pratiques agricoles

Pour décrire les pratiques en terme d'application de pesticides et volume d'eau associé, un fichier practice\_plotX.csv est fourni pour chaque UH X de type parcelle. Celui-ci contient un ensemble de dates/heures une information sur la masse de solutés appliquée (g/ha), le volume d'eau associé (L) et l'indice du soluté qui a été appliqué. Un exemple est présenté Fig. 78.

<u>date/hour</u>	<u>mass (g/ha)</u>	<u>water volume (L)</u>	<u>Solute index</u>
14/06/2003 10:00:00	1000	1400	1
14/06/2003 10:00:00	0	0	2

FIGURE 78 – Exemple de fichier practice\_plot1.csv

## C.6 Autres

### Degradation.csv

Les données relatives à la dégradation des solutés sont disponibles dans le fichier **degradation.csv**. Ce fichier renseigne pour chaque soluté l'indice du soluté qui est son métabolite principal (999 s'il n'y en a pas) et le taux de transformation associé (entre 0 et 1). Ainsi, on modélise dès le début de la simulation le devenir du soluté et du métabolite et la masse dégradée d'un soluté peut venir incrémenter le stock de son métabolite au lieu d'être simplement éliminée. Par exemple, pour modéliser 3 solutés tels que #1 possède comme métabolite principal #2 et que #3 n'a pas de métabolite, on aura par exemple le fichier degradation.csv présenté fig. 79.

Solute #	Metabolite	Occurrence fraction
1	2	1
2	999	1

FIGURE 79 – Exemple d'un fichier degradation.csv.

⚠ La table degradation.csv doit être triée par ordre croissant d'indice de solutés et pour le moment l'ensemble d'indices est forcément le vecteur [1,2,...n]. Par la suite, il faudra s'assurer que PESHMELBA s'en sort si ce n'est pas le cas.

### forcing\_data.in

Enfin, on fournit également des séries temporelles de précipitations et de PET dans le fichier **forcing\_data.in**. Pour le moment, les dates y sont données en nombre d'heures écoulées depuis le début de la simulation, à modifier pour avoir également des dates complètes, au format "YYYY-MM-DD hh :mm :ss".

# From agricultural catchment to management scenarios : a modular tool to assess effects of landscape features on water and pesticide behavior

Emilie Rouzies<sup>a,\*</sup>, Claire Lauvernet<sup>a</sup>, Christine Barachet<sup>a</sup>, Thierry Morel<sup>b</sup>, Flora Branger<sup>a</sup>, Isabelle Braud<sup>a</sup>, Nadia Carluier<sup>a</sup>

<sup>a</sup>*Irstea, UR RIVERLY, 5 rue de la Doua, CS 20244, 69625 Villeurbanne Cedex, France*

<sup>b</sup>*CERFACS, 42 avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France*

---

## Abstract

Pesticide transfers and fate are highly influenced by the presence of discontinuities such as grass strips, slopes, hedgerows or roads that can accelerate or slow down and dissipate water and contaminant fluxes. That is why those landscape elements must be integrated into watershed management plans. It implies taking them into account when modeling water and contaminant fluxes at the small catchment scale. However, if the influence of landscape elements has already been widely explored at field scale, models generally do not reach the catchment scale. The project PESHMELBA aims at developing a new modeling tool of water and contaminants circulation and fate at the scale of small catchments in order to optimize landscape organization. The model explicitly takes into account spatial organization of landscape by representing existing elements, their locations and shapes. The final aim of this modeling tool is to efficiently test and rank different development scenarios in order to assess the influence of agricultural practices, land uses and landscape management strategies on water quality. In PESHMELBA, dominating processes ruling water and contaminants circulation and dissipation for each element type are mainly represented by existing and validated models. New components have also been developed when no suitable model was found in the literature. All these models present different levels of conceptualization and are used as modeling units ensuring a modular structure. Then, the different units are gathered and connected in the OpenPALM coupler (Fouilloux and Piacentini, 1999) in order to implement the spatial and temporal couplings. This innovative approach leads to a spatialized model of the whole catchment. Applications cases are tested with an increasing complexity, from a case with two plots to the hillslope scale with several plots, ditches and rivers. They show that PESHMELBA is a promising tool to compare scenarios considering water and pesticide fate in different complex landscapes.

*Keywords:* pesticide behavior, spatialized modeling, agricultural catchment, landscape scenarios, contamination management

---

## 1. Introduction

Occurrence of pesticides in surface and groundwater is a major concern regarding water quality in agricultural watersheds. At the catchment scale, pesticide transfers depend on soil characteristics, slopes and rainfall patterns, but are also influenced by agricultural and landscape management practices (Campbell et al., 2004). Indeed,

---

\*Corresponding author

Email address: [emilie.rouzies@irstea.fr](mailto:emilie.rouzies@irstea.fr) (Emilie Rouzies)

5 landscape elements can generate discontinuities of water and pesticide fluxes, depending on their nature and their characteristics. For example, grass buffer zones reduce surface velocity limiting runoff and enhancing infiltration, pesticide trapping and degradation (Muñoz-Carpena et al., 1999; Poletika et al., 2009) and can thus be used as Best Management Practices (BMPs) (Reichenberger et al., 2007; Gevaert et al., 2008; Alix et al., 2017; Carluer et al., 2017a). On the opposite, ditches can accelerate transfers to the outlet by collecting and transmitting runoff  
10 (Buchanan et al., 2013). However, depending on the slope, the substrate and the vegetation implantation, ditches may also have a positive effect, increasing retention capacities (Elsaesser et al., 2013; Stehle et al., 2011; Margoum et al., 2001; Dollinger et al., 2016). Similarly, roads can prevent overland flow downslope but they can also speed up transfers of water and dissolved contaminants toward stream channels (Leopold & Dunne, 1978; Jones et al., 2000). As a result, all the elements present in the watershed, their position, their design and their specific functioning  
15 must be considered all together, as an ensemble for setting mitigation strategies to preserve water resources.

What's more, in this objective of preservation of water resources, it is not only necessary to take into account the landscape organization but also to consider specific subsurface functioning of the catchment. In that way, catchments where surface runoff and shallow subsurface flow strongly interact and where subsoil is impervious rock require special attention. They can be characterized for instance by the formation of shallowly perched water table  
20 near the surface. Those configurations are quite common in Europe (Dubus & Surdyk, 2006) and they make the water resources particularly vulnerable as it is located close to the surface. This knowledge should also be considered when implementing preventing actions.

Computer modeling is a relevant tool for determining such prevention and mitigation plans. Indeed, it makes it possible to set and to simulate different scenarios, such as adding some new buffer zones in the watershed or changing  
25 agricultural practices. These scenarios can then be evaluated in order to choose the most suitable mitigation actions. Many modeling tools account for solute transport and transfer at different scales (from field-scale to catchment-scale), taking different processes into account with different levels of complexity.

On the one hand, pesticide fate modeling at the local scale makes it possible to explore the function of specific landscape elements regarding solute fate. A large range of models were developed to simulate pesticide transport  
30 and fate at the plot scale, in particular for regulatory purposes. Among them, MACRO (Larsbo & Jarvis, 2003) is a 1D mechanistic model dedicated to pesticide leaching and movement to drains. Its dual-permeability representation of the soil takes into account preferential flows, an important pathway for pesticide transport (Beven & Germann, 1982; Jarvis, 2007; Beven & Germann, 2013). Conceptual models also exist such as PRZM (Carsel & Baldwin, 2000) or Pestdrain (Branger et al., 2009) that are dedicated to surface runoff and tile-drainage flow respectively  
35 with simplified approach. Additionally, some models represent water circulation and pesticide fate on other specific landscape elements such as buffer zones or ditches. VFSSMOD (Muñoz-Carpena et al., 1999, 2018; Lauvernet & Muñoz-Carpena, 2018; Sabbagh et al., 2009) simulates water, sediment and pesticides on vegetative filter strips considering rainfall event, incoming sediment and overland flows. TOXSWA (Adriaanse, 1997; Beltman & Adriaanse, 1999; Adriaanse et al., 2013) is a mechanistic model used for pesticide European certification that simulates the  
40 evolution of solute concentration within the water of a ditch. It takes into account a series of biogeochemical processes that can attenuate solute transfers (Dollinger, 2016). However, the potential mitigation power of such landscape elements regarding pesticide transfer depends on their position in the catchment and their interactions

with other elements. The models that represent solute transport and transfer at the field scale do not intrinsically take into account interactions neither with neighboring elements nor with the rest of the catchment, and those interactions can only be taken into account by setting locally adapted boundary conditions or inputs (Lauvernet & Muñoz-Carpena, 2018). Some studies include for example some VFSMOD simulations into a watershed scale tool to assess buffer performance and placement at the watershed level (Dosskey et al., 2006; Tomer et al., 2009; Carluer et al., 2017b). However, the fact remains that such methods do not allow dynamical coupling nor interactions in space and time.

In addition to local-scale models, a variety of 3D models are broadly used to simulate water flow pathways and pesticide transport at the catchment scale with different degrees of conceptualization. Some models, widely used for operational purposes, either fully-distributed such as annAGNPS (Bingner, 2011) or semi-distributed such as SWAT (Arnold et al., 1998) are able to take into account different land management practices. ZIN-AgriTra (Gassmann et al., 2013) is another physically-based model that simulates pesticide and transformation product fate on small agricultural catchments. In addition to simulating solutes reactive transport and transfers, ZIN-AgriTra also includes preferential flows. Original approaches also exist such as LISEM (De Roo et al., 1996), an event-based hydrological and soil erosion model that is completely included in a raster Geographical Information System. However, such models generally represent the subsurface in a quite simplified way and cannot account for shallowly perched water tables. These can be a significant pathway for pesticide transfers, especially on catchments where surface runoff and shallow subsurface flow interact. Hence, they need to be included when modeling pesticide fate, as shown for example on vegetative filter strips that are located along the hydrographic network (Simpkins et al., 2002; Lacas et al., 2005; Muñoz-Carpena et al., 2018). Additional limits can also arise in those models if temporal resolution is not detailed enough. Temporal variability is an important feature of pesticide transfers at the catchment scale (Ippolito & Fait, 2019) and it can be missed if the temporal resolution is too coarse. In order to accurately describe all relevant processes at the catchment scale with a high temporal resolution, Integrated Surface-Subsurface Hydrological Models (ISSHMs) can also be used. Among them, ParFlow (Ashby & Falgout, 1996; Jones & Woodward, 2001; Kollet & Maxwell, 2006) or CATHY (Paniconi & Putti, 1994; Camporese et al., 2010) give an accurate representation of water and solute surface/subsurface fluxes and interactions. Such models can also integrate a representation of reactive solute transfers (Weill et al., 2011; Beisman et al., 2015; Gatel et al., 2018). However, these 3D, surface-subsurface, physically based models are often based on a fixed mesh. It implies that they require to entirely build a new mesh when modifying one element in the modeling domain. They are thus limited for integrating different landscape elements with proper geometry, properties and processes. As a result, they can hardly constitute an exploration tool for different landscape organization scenarios.

Considering the difficulties of the existing modeling tools acting at the catchment scale to take into account the diversity of landscape elements, we may think about a different approach in order to set a model able to test different landscape management practices. As a lot of models for diverse elements and diverse processes already exist at a local scale, the following issue arises: can all of this existing knowledge be combined efficiently in order to reach the catchment scale?

One way is to rely on a modeling framework that couples models standing for different processes or elements. That is the base for integrated modeling (Hutchings et al., 2002). Many projects are already based on this approach:

JAMS (Kralisch & Krause, 2006), WaterCAST (Argent et al., 2009), LIQUID (Branger et al., 2010) or ECHSE (Kneis, 2015) for instance. Among them, only a few particularly address the question of pesticide fate at the hillslope scale. OpenFluid (Fabre et al., 2010) is a framework designed for modeling water and solute flows in agricultural landscapes. Nevertheless, landscape elements representation is currently limited to plots and ditches and shallow lateral subsurface flows are not accurately represented. The CMF framework (Catchment Modeling Framework) (Kraft et al., 2012) was also adapted by Djabelkhir et al. (2017) to simulate pesticide transfers at the hillslope scale. However, this toolkit is designed to simulate water fluxes or advection and does not allow an easy inclusion of reactive processes or specific landscapes features. In all, there is a real need for a flexible, modular and open-source tool that allows an easy coupling between units. This conception of modeling using coupling inside hydrological platforms is also strongly encouraged in the literature (Buytaert et al., 2008; Kraft et al., 2012; Fatichi et al., 2016) and appears to be a relevant solution to reuse existing models. However, these platforms are mainly dedicated to hydrology and hardly adaptive to reactive solutes such as pesticides (Djabelkhir, 2015). An alternative approach is the use of a code coupler such as OpenPALM (Fouilloux & Piacentini, 1999; Buis et al., 2006) which is highly flexible and able to couple complex and high dimension models whatever the variables it is coupling (see for example, Dauplain et al., 2008; Barthélémy et al., 2018).

In this paper, we propose an innovative use of the OpenPALM code coupler to build the PESHMELBA model (Pesticides and hydrology: modeling at the catchment scale) that represents water circulation and pesticide fate at the scale of small watersheds. The final objective of PESHMELBA is to represent pesticide fate and the role of elements on their transfers in a realistic way in order to compare and to rank scenarios for decision-making. We hypothesize that the landscape organization and its composing elements are highly important when modeling water circulation and pesticide transfers and fate. That is the reason why the meshing used in PESHMELBA is designed accordingly to landscape layout. The aim of the paper is first to present the building of the model, based on (1) preexisting and validated models that stand for different processes or landscape elements and (2) additional units newly developed. All elements are thereafter coupled within the OpenPALM framework. The second part of the paper is application-oriented. Simple scenarios involving plots and ditches are proposed to analyze the first results of the coupling and to illustrate the new tool's potential considering water and solutes. In order to be consistent and to reach realistic orders of magnitude, all scenarios are based on geometry, soil properties and climate forcing taken from Kervidy, a real catchment in North-West of France (Cheverry, 1998), mostly covered with cereal plots, ditches, and some buffer zones.

## 2. Material and methods

### 2.1. Preprocessing

PESHMELBA is running on a specific meshing that relies on homogeneous surface and linear components, and not on regular elements as it is commonly done. Each surface component is characterized by a unique nature (plot, grass buffer strip, etc.), a soil type and a land use or crop itinerary and each linear reach is characterized by its nature (ditch, road, river, bank, etc.). The resulting meshing is composed of surface and linear elements which size can vary significantly as shown in the example, Figure 1 (left). Additionally to these elementary components, the

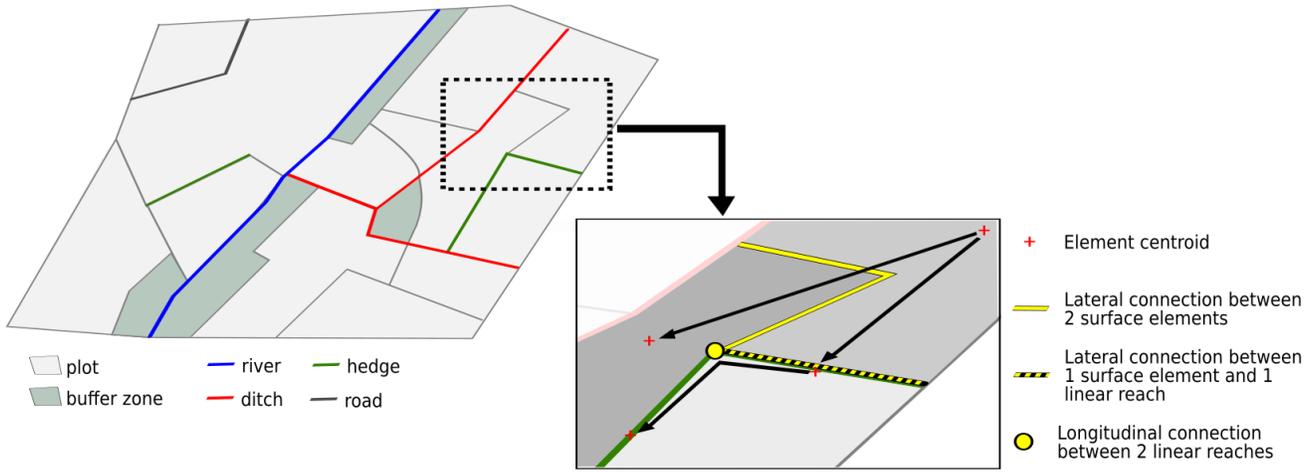


Figure 1: Surface and linear elements in a virtual catchment with example of longitudinal and lateral connections between them.

PESHMELBA meshing is also composed of the connections between elements. Connecting interfaces can be lines, typically for lateral connections between two surface elements, between two linear elements or between one surface and one linear element. Interfaces can also be points for longitudinal connections between two portions of linear (see  
 120 some examples of connections Figure 1, right). Each element is one-way connected to the surrounding lower ones. However, whether it is for lateral or for longitudinal connections, each element can be connected to several elements upstream and downstream. The spatial description of all the types of elements and the connections between them in the whole catchment is the preprocessing required by PESHMELBA for any simulation.

## 2.2. Modeling of processes and landscape elements

The PESHMELBA model relies on an ensemble of elementary units that stand for one or several processes  
 125 concerning water and pesticide fluxes at surface and subsurface on a landscape element. Those units are written in Fortran90 or Python, chosen among existing models or developed for the need of the project. They rely on different approaches, levels of complexity and discretization. They are presented individually in this section with some details in the appendix. All units are coupled to build the PESHMELBA model as schematized in Figure 2.  
 130 This coupling within the OpenPALM framework is described in section 2.3.

### 2.2.1. Vertical infiltration and unsaturated transfers for water and solutes on plots

Vertical infiltration and fluxes in the unsaturated zone on plots are represented by the FRER1D unit (see  
 Appendix B.1). This unit implements the fast and simplified numerical scheme proposed by Ross (2003) and extensively tested and validated by Varado et al. (2006b) and Crevoisier et al. (2009). A plot is represented by a  
 135 single column of soil divided in horizontal cells and can be composed of different soil horizons with distinct properties. FRER1D represents vertical infiltration of water in the soil, formation and evolution of saturated zones, evaporation on bare soil in response to rain and potential evapotranspiration. Surface boundary conditions are determined by the PARTITION unit that takes into account rain interception by the vegetation and plant transpiration. Inclusion of a source/sink term has also been added in order to represent lateral saturated transfers or root extraction by

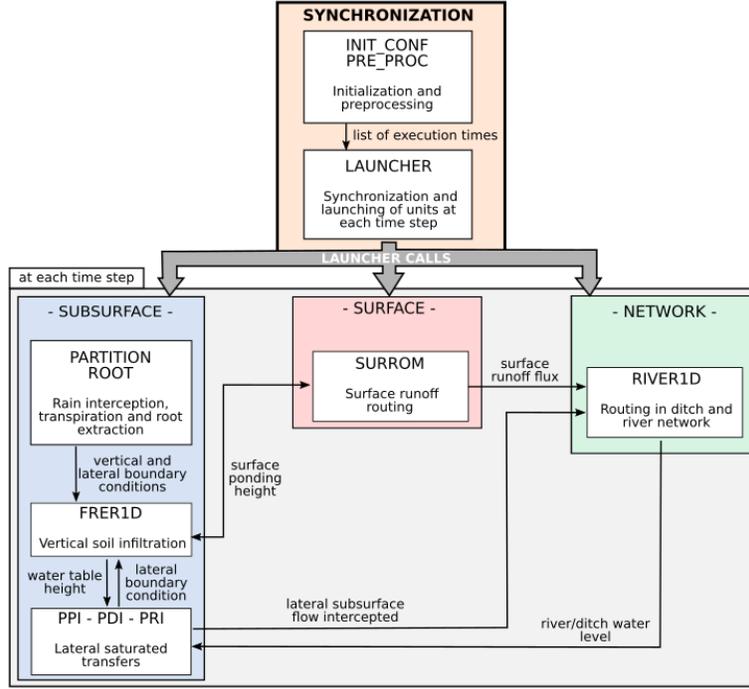


Figure 2: Schematic illustration of the PESHMELBA model structure. Colored rectangles stand for branches, white rectangles for units, arrows for the main variables exchanged between units.

140 coupling FRER1D with other dedicated units : unit PPI for lateral saturated transfers (see section 2.2.2) and unit ROOT for root extraction (see Appendix B.2).

Transfer and fate of several solutes can be simulated at the same time so that it is, for example, possible to simulate the evolution of a given solute and one metabolite. For each solute, adsorption on the soil matrix is represented by linear (eq.1) or Freundlich (eq.2) adsorption isotherms:

$$q_{ads} = K_d c \quad (1)$$

$$q_{ads} = K_f c^{n_f} \quad (2)$$

with  $q_{ads}$  the mass of solute adsorbed per mass of soil at the equilibrium [MM<sup>-1</sup>],  $K_d$  the linear coefficient of adsorption [L<sup>3</sup>M<sup>-1</sup>],  $K_f$  the Freundlich coefficient of adsorption [(L<sup>3</sup>)<sup>n<sub>f</sub></sup>M<sup>1-n<sub>f</sub></sup>M<sup>-1</sup>] and  $n_f$  the exponent of Freundlich equation [-]. Influence of soil properties on adsorption is taken into account by correcting the adsorption coefficient with soil organic carbon content  $f_{oc}$  [-]:  $K_d = K_{doc} * f_{oc}/100$  (resp.  $K_f = K_{foc} * f_{oc}/100$ ) where  $K_{doc}$  (resp.  $K_{foc}$ ) is the linear (resp. non-linear) organic carbon-water partition coefficient [-]. Degradation is integrated according to first-order kinetics:

$$c(t) = c_0 \exp(-kt) \quad (3)$$

where  $c_0$  is the concentration at time 0 [ML<sup>-3</sup>] and  $k$  the degradation rate [T<sup>-1</sup>]. The degradation rate is linked to the half-life  $DT_{50}$  [T] by the relation :  $k = \frac{\ln(2)}{DT_{50}}$ . If a metabolite has been specified, the degraded quantity is added to the metabolite mass in each column cell.

The PPI (Plot-Plot-Interface) unit has been developed to handle lateral transfers between two plots and more generally between two soil columns. As commonly done, it is assumed that lateral transfers are mainly saturated and that unsaturated flows can be neglected (Abbott et al., 1986; Chiew et al., 1992; Beckers & Frind, 2000). PPI is based on the Darcy law (eq.4) and on a simplified parameter setting firstly established for the WTI (Water Table Interface) unit within the LIQUID framework (Branger et al., 2010). The LIQUID unit has been adapted to the specific needs of PESHMELBA, mostly linked to solutes. For example, shallow water tables must be taken into account as pesticide fate is fully dependent on the transport and transformation processes that take place in the near-surface soil layer (Flury et al., 1995). The PPI unit is therefore designed to handle heterogeneous soils composed of several horizons and multi-water-table cases with soil columns that simultaneously include groundwater and shallow water tables. Considering two connected soil columns  $Col_x$  and  $Col_y$ , for each water table in  $Col_x$ , the closest but lower water table in  $Col_y$  is identified (the bottom of the water table is taken as reference). If the hydraulic gradient between the two plots is compatible with the topographic gradient, a lateral saturated flow is determined between those water tables (Figure 3, (b) and (d)). Otherwise it is set to zero. If there is no water table in  $Col_y$ , the saturated flow is calculated between the water table of  $Col_x$  and the lower cell of the closest but lower horizon in  $Col_y$  (Figure 3, (c)).

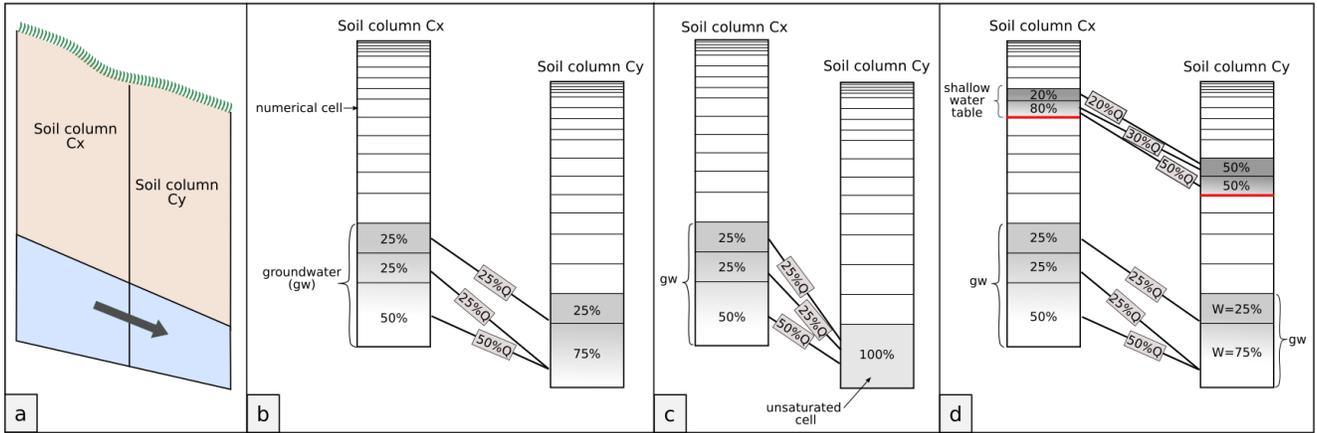


Figure 3: (a) Schematic view of two soil columns with water table. (b,c,d) Examples of distribution of the saturated flow calculated by PPI between saturated layers, (b) with transfer involving groundwater in the 2 plots, (c) with groundwater in one plot only, or (d) with groundwater and shallow water table. Red lines stand for interfaces between soil horizons and grey gradient stands for concentration gradient. Percentages within the numerical cells of each column are weights  $w_{WT}^i$  that distribute the outgoing flow  $Q$  toward each cell downstream. Percentages on connections between cells illustrate the contributions of each cell upstream to spread the outgoing flow  $Q$  toward each cell downstream.

For each water table in  $Col_x$ , the associated volumetric flow  $Q_{x \rightarrow y}$  [ $L^3 T^{-1}$ ] is expressed as:

$$Q_{x \rightarrow y} = K_{s,h}^{int} * A * \frac{H_x - H_y}{d_x - d_y} \quad (4)$$

where  $A$  is the contact area between the columns  $Col_x$  and  $Col_y$  [ $L^2$ ],  $H_x$  and  $H_y$  the water table levels in columns  $Col_x$  and  $Col_y$  [ $L$ ],  $d_x$  and  $d_y$  the distances between the centers of the columns and the interface [ $L$ ] and where  $K_{s,h}^{int}$  is the horizontal hydraulic conductivity at saturation at the interface [ $LT^{-1}$ ] (see Appendix Appendix B.3.1

for detailed calculation of  $K_{s,h}^{int}$ ). The total saturated flow from  $Col_x$  to  $Col_y$  is the sum of the volumetric flows associated to each water table in  $Col_x$  and determined by eq. 4. In what follows, this flow will be considered positive from  $Col_x$  to  $Col_y$ .

For each water table in  $Col_x$ , a resulting solute flow is calculated standing for advective transport:

$$Q_{x \rightarrow y}^{sol} = c * Q_{x \rightarrow y} \quad (5)$$

where  $Q_{x \rightarrow y}^{sol}$  is the solute mass flow [MT<sup>-1</sup>] and  $c$  the concentration of a given pesticide in the dissolved phase [ML<sup>-3</sup>]. PPI is based on the same vertical meshing than FRER1D so that the different lateral contributions will be used as sink/source term and handled by FRER1D without further adaptation when coupling the two units. As for vertical infiltration calculation, each soil column is divided in horizontal cells and an original coefficients system is applied to handle the spreading of the resulting flow  $Q_{x \rightarrow y}$  between the saturated cells of the two columns. For each saturated transfer identified between  $Col_x$  and  $Col_y$ , the coefficients system defines which portion of  $Q_{x \rightarrow y}$  flows out from each saturated cell in  $Col_x$  and enters in each saturated cell in  $Col_y$ . Those coefficients  $w_{WT}^i$  are derived from the hydraulic conductivity and the thickness of each layer in order to prioritize the transfers between numerical cells with higher hydraulic conductivity or higher thickness. They are calculated for each water table so that the sum is equal to 1 within the water table (eq. B.13). Besides, if a water table exists in  $Col_y$ , each cell from  $Col_x$  that contributes to the outgoing flow is associated to the closest cells of this water table in  $Col_y$ . Such a technique should avoid complete mixing of a potential vertical gradient of concentration during lateral transfers.

### 2.2.3. Surface runoff on plots

Surface runoff corresponds to the water that did not have time to infiltrate or that exfiltrated within a time step. This quantity is determined by the FRER1D unit before being routed across the landscape by the newly-developed SURROM (SURface ROuting Module) unit. Classical approaches to handle surface runoff routing across a catchment are based on solutions of the kinematic wave equation (Beven, 1979; Li et al., 1975) or solutions of the diffusive wave equation (Moussa & Bocquillon, 1996; Kraft et al., 2012; Gassmann et al., 2013; Camporese et al., 2010). However, the originality of the approach of PESHMELBA is the explicit inclusion of the linear elements of the landscape. Those elements intercept or redirect surface runoff along the watershed and would represent many tricky discontinuities to handle in a classical resolution of runoff routing. Moreover, as detailed previously, water available at surface for runoff for each plot is computed from FRER1D and a single height of water is affected to the plot. Regarding the constraints linked to the presence of discontinuities and to the meshing imposed by FRER1D, a sequential approach based on the kinematic wave approximation has been adopted. After running, FRER1D provides an excess surface water  $h_0$  that did not infiltrate during the time step of the application (hereafter called  $dt\_PALM$ ). This water level is transmitted to SURROM that also runs during  $dt\_PALM$ . However, as infiltration and surface runoff processes contrast by their characteristic duration, SURROM uses a smaller time step than FRER1D (called  $dt\_RO$ ). So SURROM runs several times between  $t$  and  $t + dt\_PALM$  at its own time step  $dt\_RO$ . At each time step  $dt\_RO$ , the water volume  $\frac{dt\_PALM}{dt\_RO} \times h_0$  is applied on the plot. Then, flux is calculated from water level with the Manning-Strickler equation. A corresponding outgoing water volume is determined and

this volume has to be allocated to one or several downstream elements. The distribution between them is done considering a weighting system based on the length of the interface and the altitude gradient between the centroids of these two elements. If a given plot has  $n$  downstream connections with surface or linear elements, the outgoing flow towards each downstream element  $i \in \{1, \dots, n\}$  is :

$$Q_i = w_i \times \frac{L_i h_w^{5/3}}{n} \sqrt{S_0} \text{ with } w_i = \frac{s^i L_i^{2/3}}{\sum_{j=1}^n s^j (L_j)^{2/3}} \quad (6)$$

Where  $L_i$  is the length of interface with the downstream element  $i$  [L],  $h_w$  the water level on the plot [L],  $n$  the Manning coefficient [L T<sup>-1/3</sup>],  $S_0$  the slope of the plot [L L<sup>-1</sup>] and  $s^i$  the altitude gradient between the centroid of the plot and the connected element  $i$  [-]. Then, water level is updated considering the outgoing volume towards downstream elements (linear or surface) and the potential incoming volumes from upstream. Such a sequential  
 185 approach does not require any calculation order as outgoing volume flows are calculated on all plots before updating water levels with downstream and upstream contributions.

In addition, the SURROM unit accounts for advective transport of solutes. Reactive transfers are also taken into account through simplified representations of degradation and adsorption. Degradation is simulated in the whole dissolved phase by a first-order degradation law (Equation 3). Adsorption isotherms are applied in a mixing layer  
 190 in order to represent sorption to the soil particles. Such mixing layer includes the water height on surface and the upper part of the soil layer with a configurable height. The upper part of the soil that is considered in the mixing layer is assumed to be in suspension with the water, allowing a complete and instantaneous adsorption. The adsorption is calculated at the end of each time step, allowing remobilization of solute previously adsorbed. Ponding can also be taken into account with a customizable height. As it delays surface runoff activation, water  
 195 residence time increases enhancing pesticide trapping on plot.

#### 2.2.4. Routing in the hydrographic network

Routing of water and solutes in ditch and river networks is represented by an adapted version of the River1D unit (Branger et al., 2010), which is based on the 1D kinematic wave approximation resolved by a fourth order Runge-Kutta method (Cash & Karp, 1990). This unit deals with ramified networks, confluences and has also been  
 200 modified to handle diffluences, since those configurations can happen in complex ditch networks. The different reaches of ditches and rivers obtained from landscape analysis (cf section 2.1) are firstly gathered into independent networks. Each reach is represented with a simplified geometry based on a trapezoidal section. It receives lateral subsurface and surface input flows from connected neighboring elements. Additionally, solute transport is simulated in the network by solving the advection equation. Dispersion has not been included yet. Reactive solute transfers are  
 205 represented by a first-order degradation law in the river and ditch reaches. Adsorption processes are also considered to the various materials of the ditch bed. It is assumed that sorption processes take place in a mixing layer of 2 cm at the bottom of the ditch. This is a strong but common hypothesis assumption in modeling approaches (Focus, 2007; Dollinger, 2016).

### 2.2.5. Lateral exchanges between plots and linears

Linear elements interact with the neighboring plots differently depending on their nature and their properties. A specific unit has to be designed for each type of linear element and its specific functioning. So far on, it is considered that ditches interact with the connected plots by intercepting surface runoff and lateral saturated transfers. A newly-developed unit PDI (Plot-Ditch-Interface) was implemented in PESHMELBA to simulate the interception of subsurface flow between two plots separated by a reach of ditch (Figure 4). The lateral subsurface flow between two plots is firstly computed without considering the intersection by a linear element, as described Figure 3. Then, the portion of flow generated in the numeric layers above the bottom of the ditch is driven to the ditch. The rest of the flow is transferred to the plot downstream. This unit handles interception of flows from shallow water tables and from groundwater. Solutes dissolved in the water are similarly allocated as they are advected by the water flow.

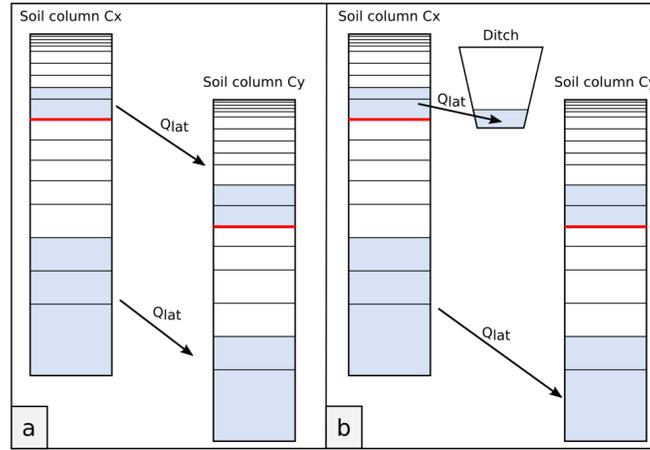


Figure 4: Procedure of calculation of flow interception by a ditch in the PDI unit. Red lines stand for interfaces between soil horizons.  $Q_{lat}$  is the lateral subsurface flow between two saturated zones. (a) Lateral subsurface flows are firstly calculated without being intercepted by the ditch. (b) Then, part of the flows are allocated to the ditch.

It is also considered that river reaches intercept surface runoff generated on the plots upstream. Exchanges between groundwater and the river are expressed in the PRI unit (Plot-River-Interface) using the Miles relation (Miles, 1985):

$$Q_{pr} = C_m K_s \Delta h \quad (7)$$

where  $Q_{pr}$  is the flow between the aquifer and the river [ $L^3 T^{-1}$ ].  $Q_{pr}$  is directed to the river if superior to zero (respectively directed to the plot if inferior to zero).  $K_s$  is the saturated hydraulic conductivity of the aquifer [ $L T^{-1}$ ],  $\Delta h$  is the head loss between the groundwater and the river [ $L$ ] and  $C_m$  the Miles coefficient adapted for trapezoidal section (Dehotin, 2007):

$$C_m = \frac{0.5(0.25(W_b + W_t) + H_r + s)}{D + H_r + s} \quad (8)$$

with  $H_r$  is the river stage level [ $L$ ],  $D$  the saturated thickness of the aquifer below the river bed [ $L$ ],  $W_b$  the bottom width [ $L$ ],  $W_t$  the water surface width [ $L$ ] and  $s$  the height of the seepage face [ $L$ ].  $Q_{pr}$  is converted into a sink (respectively source) term for each numerical cell of the water table in the plot. Such calculation is based on the same splitting system as presented in Section 2.2.2. Solutes are also advected in direction to the river (respectively

in direction to the plot). In case of a flow directed to the river, the outgoing solute flow from each cell is calculated using the water flow from this cell and the corresponding solute concentration. In case of a flow directed towards the plot, such lateral flow is determined using the water flow to each cell and the solute concentration in the river. Finally, shallow water tables can also flow towards the river through a seepage face. The outgoing flow is then determined using the lateral pressure gradient between the plot and the atmosphere.

### 2.3. Coupling of processes and landscape elements with the OpenPALM coupler

The elementary units presented in Section 2.2 are coupled in the OpenPALM software in order to build the PESHMELBA model (Figure 2). OpenPALM is an open-source dynamic coupler allowing for data exchanges between different independent code components through MPI (Message Passing Interface) communications (Duchaine et al., 2015). A specific feature of this tool is its ability to deal with complex execution scheduling and thus to address complex modeling issues. For the PESHMELBA building, the units are integrated within the framework according to their specific language (Fortran90 for PARTITION, ROOT, FRER1D and Python for PPI, PDI, SURROM, RIVER1D) and gathered depending on the type of element they refer to. Communications are set between units in order to transfer state variables, fluxes and forcing. Two time steps are used in a PESHMELBA simulation, all specified by the user. The first one is the global time step  $dt\_PALM$  that can vary all along the simulation depending on weather conditions. This adaptive time step is finer during or after rainy periods (Figure 5) and it is used by all units relative to subsurface processes (PARTITION, ROOT, FRER1D, PPI and PDI). Other units that represent processes with distinct characteristic durations, such as SURROM or RIVER1D are solved on a second time step  $dt\_RO$ , smaller than  $dt\_PALM$ . The coupling is simplified by running the units with distinct time steps sequentially: first, the units PARTITION, ROOT and FRER1D are run between  $t$  and  $t + dt\_PALM$ . Then, the final ponding volume is transferred from FRER1D to SURROM. The SURROM unit is launched between  $t$  and  $t + dt\_PALM$  and runs several times as long as its characteristic time step  $dt\_RO$  is smaller than  $dt\_PALM$  (Figure 5). At each time step  $dt\_RO$ , a fraction of the ponding volume recovered from FRER1D is allocated to SURROM to calculate surface runoff. Sequentially, the surface runoff variable is sent to the RIVER1D unit to calculate the flow in the different networks. RIVER1D also runs several times between  $t$  and  $t + dt\_PALM$  with its characteristic time step  $dt\_RO$ . Once the execution of the SURROM and the RIVER1D units between  $t$  and  $t + dt\_PALM$  are over, the PPI and PDI units calculate the lateral subsurface flows that will be used at the following time step ( $t + dt\_PALM$ ).

## 3. PESHMELBA application cases

### 3.1. Scenarios set up

In order to assess the coupling, PESHMELBA was tested on 4 contrasted scenarios of growing complexity (see figure 6). Scenario 1 was based on a setup including two 1-ha plots connected through a 100-m interface. Scenario 2 integrated a reach of ditch between Plot 1 and Plot 2, and a second reach, longitudinally connected to the first one. A second set of scenarios aimed at evaluating the model in more complex scenarios involving more connections at surface and subsurface. For that purpose, a virtual hillslope of 11.2 ha was set. Scenario 3 was first composed of plots and rivers and Scenario 4 considered the presence of a network of ditches on the same

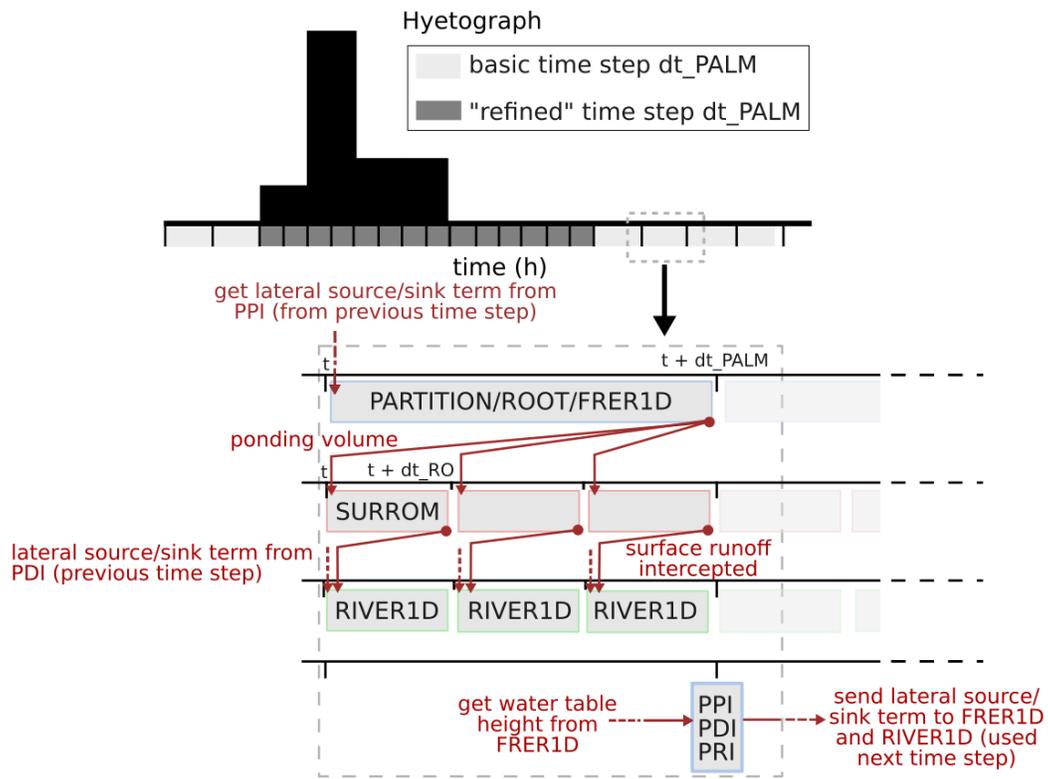


Figure 5: Timeline with time steps nesting and variable exchanges as implemented in PESHMELBA. Top: schematic hyetograph illustrating the refinement of  $dt\_PALM$  considering the rainfall characteristics. Bottom : zoom on the units organization during one time step  $dt\_PALM$  and on the use of an internal time step for surface runoff. Each grey rectangle stands for one unit execution and red arrows stand for variable exchanges within a time step  $dt\_PALM$ .

configuration. Simulations were run with very mobile, moderately mobile and slightly mobile solutes in order to investigate PESHMELBA capacities for representing transport and fate of solutes with contrasted properties. They involved Bromide (hereinafter referred to as  $\text{Br}^-$ ), Isoproturon (IPU) and Diflufenican (DFF) respectively.

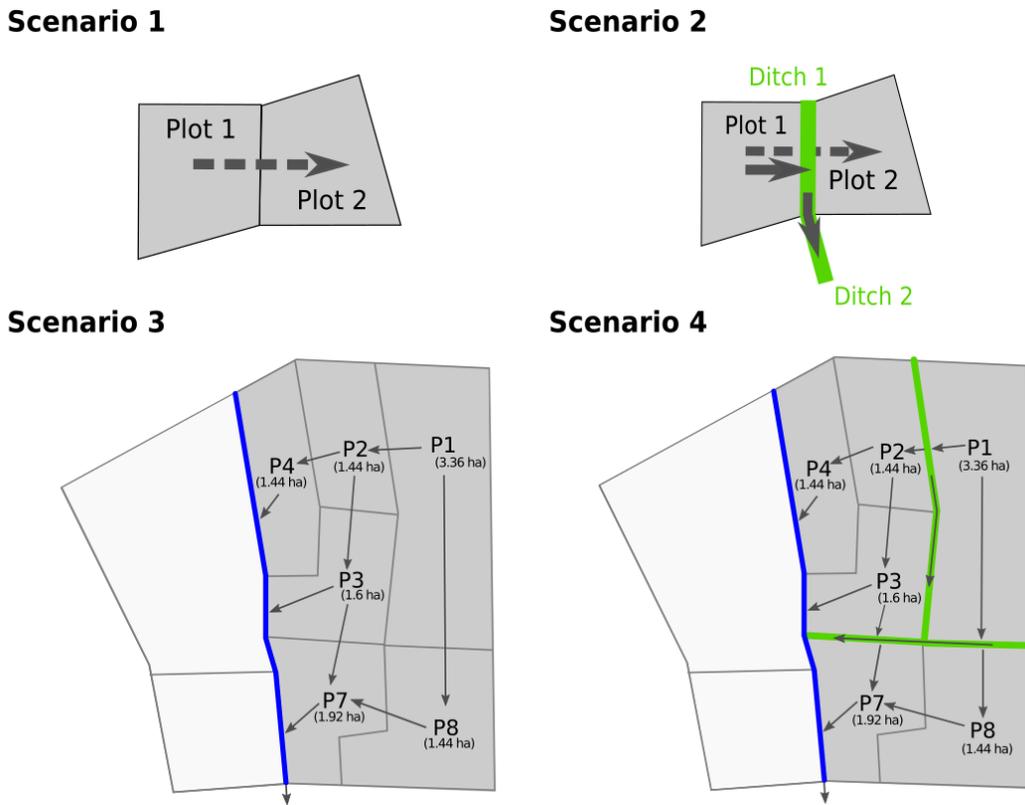


Figure 6: All configurations tested to assess PESHMELBA model locally (Scenarios 1 and 2) and then at the hillslope scale (Scenarios 3 and 4). Grey polygons stand for plots, blue lines stand for rivers and green lines stand for ditches. In Scenarios 3 and 4, the hillslope is the right part of the domain.

For Scenario 1 and 2, results analysis focused in details on vertical information (saturation and solutes mass profiles) for each element and a complete mass balance was set to assess transfers between them. We particularly intended to investigate lateral exchanges involving shallowly perched water table. Comparisons between Scenario 1 and 2 were performed to assess the impact of a ditch implantation on solute transfers at a local scale but also the behavior of contrasted molecules in these two configurations. In Scenario 2, we also aimed at examining interception of subsurface flow by a ditch and longitudinal transfer of water and solutes along a network of ditches. Considering the number of elements involved in Scenarios 3 and 4, the model application was assessed through spatialized results. Mass balance at the hillslope scale and exportation rate at the outlet were also provided to assess the impact of the ditch network on pesticide exportation. Again, simulations were run with IPU, DFF and  $\text{Br}^-$  to discuss the influence of solute characteristics on pathways and exported masses. Finally, for all scenarios, mass balance error was carefully examined as it is a basic but crucial indicator of model performances.

Parameter	Unit	Ditch	River
Depth	m	0.5	2.0
Bank angle	°	30	10
Bottom width	m	0.5	2.0
Slope	%	2/0.2	0.2
Soil thickness for adsorption	m	0.04	0.02
Saturated hydraulic conductivity of the aquifer	m/s	-	$5.6 \cdot 10^{-9}$
Thickness of the aquifer below the river bed	m	-	2.0
Bulk density	g/cm <sup>3</sup>	1.4	1.4
$\theta_s$	-	0.36	0.41
Manning coefficient	s/m <sup>1/3</sup>	0.03	0.033
Input flow	m/s	-	0.08

Table 1: Parameters of simulation for ditch and river reaches.

### 3.2. General model setup

The scenarios involved landscape elements which characteristics are based on the Kervidy catchment (Cheverry, 1998; Carluer & Marsily, 2004) in order to keep the application cases realistic. Although in PESHMELBA each element can be characterized by a distinct set of parameters, in these applications, the parameters were mostly the same for each element type. Each plot was discretized into 19 vertical cells of growing thickness from 0.5 cm at the top of the column to 50 cm at the bottom. Soil columns were 4 m high with a 2 % surface slope and a ponding height threshold of 0.25 cm on surface. Soil type was an hydromorphic soil (Carluer & Marsily, 2004), with a 2-cm soil crusting layer characterized by lower permeability and higher bulk density (Table 2). Anisotropy between vertical and horizontal hydraulic conductivity at saturation was also considered ( $\gamma = K_{sat}^h/K_{sat}^v = 10$ ). Ditches were 50-cm-depth, with an average slope equal to 2 % if they were oriented in the flow direction and equal to 0.2 % if they intercepted the flow direction. River reaches were all characterized by a 2m-depth and a slope of 0.2 %. Ditches and river characteristics are gathered in Table 1.

Atmospheric forcing was set according to Kervidy climatic conditions in the early spring. A strong stormy rainfall event of 3 hours had been extracted from Carluer & Marsily (2004). It was characterized by a maximum rate of precipitation of 26.6 mm/h ( $7.4 \cdot 10^{-6}$  m/s) reached during 1 hour and a total amount of 45.9 mm. The evaporation rate was set at 0.2 mm/h ( $4.7 \cdot 10^{-8}$  m/s). All simulations started 9 hours before the beginning of the rainfall event. Wheat crop in spring season was considered, corresponding to a 3-leaves stage of development. The parameter settings relative to vegetation processes were based on (Campbell, 1985; Li et al., 2001; Lammoglia et al., 2017) and described in detail Table C.7. All simulations ran for 10 days with an hourly time step. This time step was refined to 0.5 hour during the rain event and during 3 more hours after. IPU was firstly simulated and occurrence of Desmethylisoproturon (DM-IPU), the first metabolite of IPU (Gaillardon & Sabar, 2006; Lewis et al., 2016) was considered. Then, simulations were run considering application of DFF and Br<sup>-</sup>. Solutes properties are

Parameter	Unit	Horizons				
		0.02	0.1	0.3	0.7	4.0
Lower limit (from surface)	m	0.02	0.1	0.3	0.7	4.0
$K_s$	m/s	$6.9 \cdot 10^{-7}$	$2.4 \cdot 10^{-5}$	$1.0 \cdot 10^{-6}$	$1.7 \cdot 10^{-6}$	$1.3 \cdot 10^{-6}$
$K_s$	mm/h	2.5	86.4	3.6	6.1	4.7
$\theta_r$	-	0.14	0.14	0.00	0.00	0.00
$\theta_s$	-	0.55	0.55	0.40	0.44	0.43
$h_e$	m	-1.00	-1.00	-0.63	-0.53	-1.00
$\lambda$	-	0.55	0.55	0.25	0.16	0.29
Bulk density	$\text{g/cm}^3$	1.6	1.4	1.4	1.4	1.4
Organic carbon content	%	2.36	2.36	0.17	0.26	-
Dispersivity	m	0.2	0.2	0.2	0.2	0.2

Table 2: Hydrodynamic properties of the hydromorphic soil profile used on plots.

	Units	Isoproturon	Desmethyisoproturon	Difufenican
$DT_{50}$	d	12	33	94.5
$K_{foc}$	$\text{mL/g}^1$	122	147	2215
$n_f$	-	1.25	1.11	1.15

Table 3: Properties of solute simulated provided by the Pesticide Properties Database - PPDB (Lewis et al., 2016).

300 gathered in Table 3.

### 3.3. Initial and boundary conditions

For all scenarios, hydrostatic equilibrium was assumed as initial condition for plots and an initial 2-m-height groundwater was considered. Ditches initially did not contain water and initial level of water was 0.12 m in rivers. No solute was initially present in plots nor in ditches or rivers. For Scenario 1 and Scenario 2, upstream boundary condition was a constant flow on the subsurface ( $5.2 \cdot 10^{-6} \text{ m}^3/\text{s}$ ). Downslope lateral boundary was a Dirichlet condition given by connecting a virtual column with a fixed hydraulic head (2m-deep water table). For Scenarios 3 and 4, zero-flux was considered as boundary condition. Initial and boundary conditions for Scenarios 1 and 2 are gathered in Figure 7. For all scenarios, 1000 g/ha of Isoproturon (IPU) was applied on Plot 1 at the beginning of the simulation (in Scenarios 3 and 4, it is equivalent to a total amount of 3360 g). Note that for all scenarios, 310 the same doses were considered for DFF and  $\text{Br}^-$ , even if they are less realistic than for IPU, in order to compare simulations.

<sup>1</sup> $\text{mL/g}$  is the unit commonly used to describe Freundlich adsorption coefficient but it is actually a notation abuse and the actual unit is  $(\text{mL/g})^{n_f}$

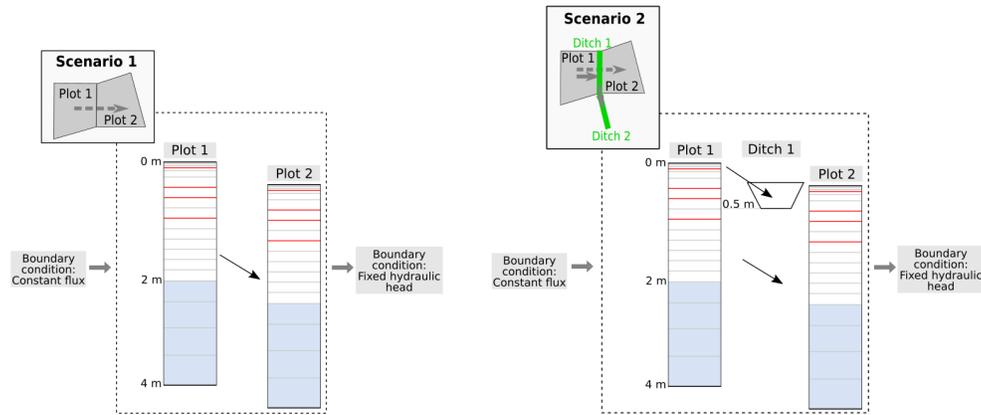


Figure 7: Vertical view and details on boundary and initial conditions for Scenarios 1 and 2.

### 3.4. Results

#### 3.4.1. Scenarios 1 & 2

Table 4 reports the cumulative mass balance for Scenarios 1 and 2. The model was able to simulate contrasted dynamics of the two scenarios with an error on mass balance inferior to  $10^{-10} \text{ m}^3$  for water and inferior to  $10^{-4} \text{ g}$  for all solutes. In Scenario 2, the presence of a ditch reach didn't modify infiltration, evaporation and root extraction amounts on Plot 1 when comparing with Scenario 1. However, water and solute amounts that infiltrated on Plot 2 decreased with the presence of the ditch. Indeed, surface runoff generated due to the crusting layer on Plot 1 was transferred to Plot 2 in Scenario 1 whereas in Scenario 2, it was intercepted by Ditch 1. As shown in Figure 8, a shallow groundwater persisted in Plot 1 for both Scenarios 1 and 2 over the entire simulation leading to saturated subsurface transfers. The presence of a soil horizon with very low permeability between 10 cm and 30 cm in the soil profile also led to accumulation of infiltrated water. After 13.5h of simulation, it led to formation of a shallowly perched water table associated to lateral transfers in both scenarios. What's more, this water table formed near the surface, in an area of high concentration in solutes (infiltrated from the surface). As a result, even if water volume transferred through this perched water table was quite low, a high quantity of solutes was transmitted. In Scenario 1, subsurface transfers were exclusively directed to Plot 2 but in Scenario 2, Ditch 1 intercepted water and solute transferred from shallowly perched water table. As a result, in Scenario 1, the solute profiles were characterized by high concentration near the surface for Plot 1 and Plot 2 (Figure 9, left and Figure 10, left). In Scenario 2, Plot 1 was characterized by the same profile evolution whereas Plot 2 only presented low solutes mass in the groundwater (Figure 9, right). In that case, small amounts of solutes percolated to groundwater in Plot 1 and were transmitted to Plot 2. As shown when comparing solute mass profiles (Figure 10) or mass balance (Table 4), mobile solutes such as  $\text{Br}^-$  are mainly transferred both vertically and laterally whereas non-mobile solutes such as DFF remain concentrated close to the area of application.

#### 3.4.2. Scenarios 3 & 4

In both cases, the small hillslope reacted quite similarly in terms of water discharge. It responded very quickly to the rain event, with a peak of discharge of nearly  $0.18 \text{ m}^3/\text{s}$  of water 2 hours after the beginning of the stormy event. However, solute transfer to the outlet was highly increased by the presence of the ditch network. Not only

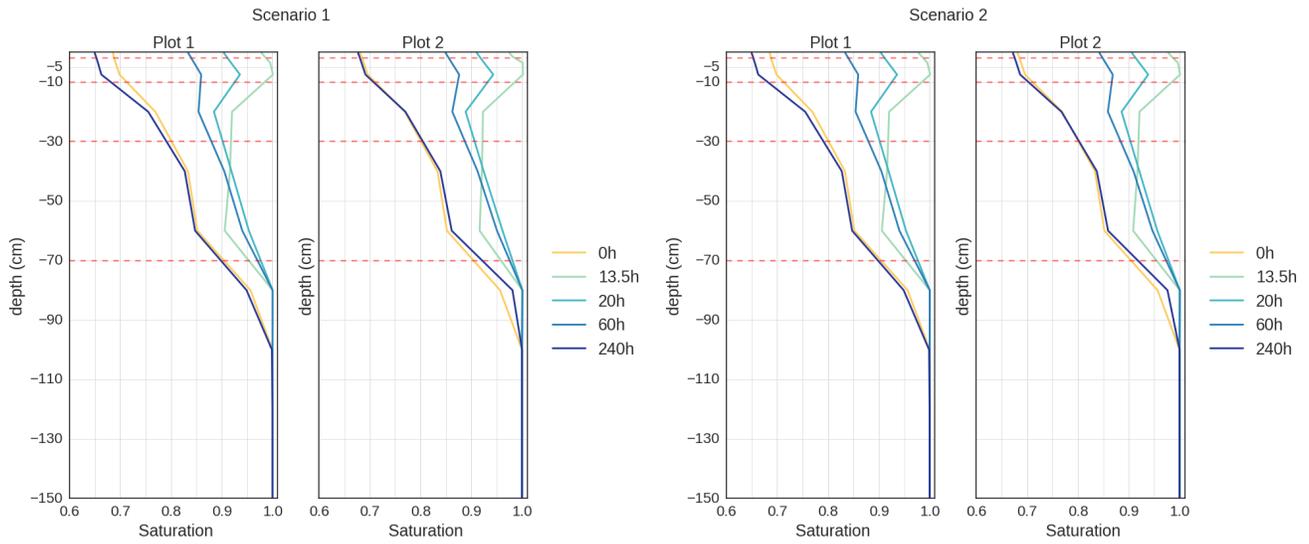


Figure 8: Saturation profile in the 2 plots at 13.5 h, 20 h, 60 h, and 240 h after the beginning of the simulation. Horizontal, dotted lines stand for interfaces between soil horizons.

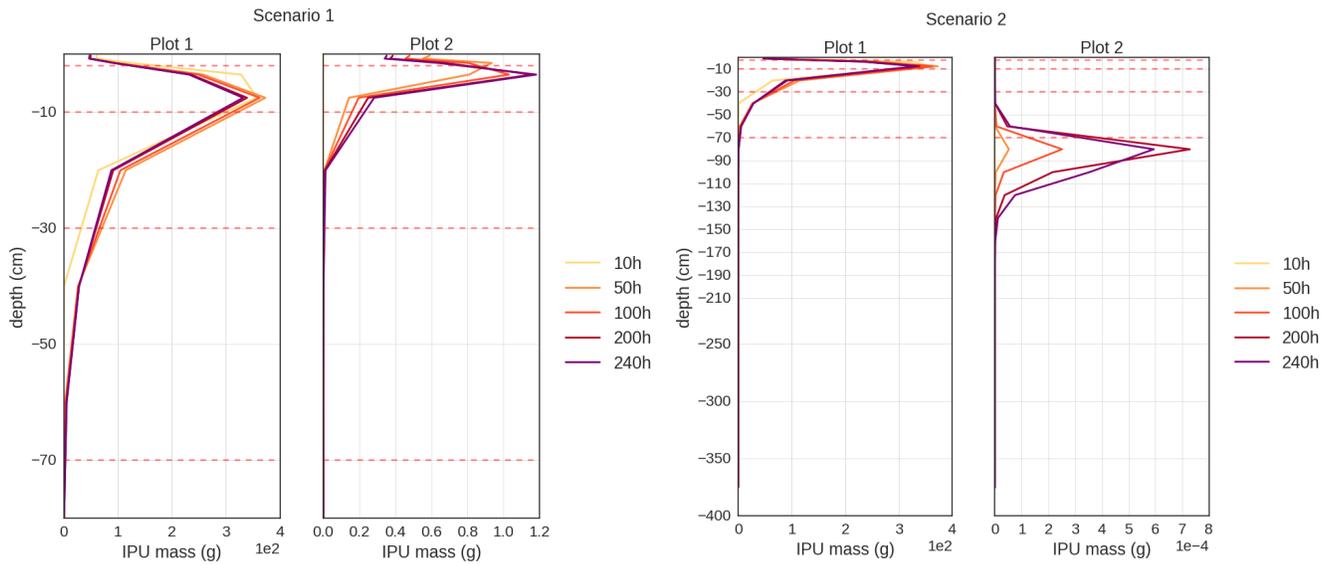


Figure 9: IPU mass profile in the 2 plots at 10 h, 50 h, 100 h, 200 h and 240 h after the beginning of the simulation. Horizontal, dotted lines stand for interfaces between soil horizons. To make the figure more readable, the vertical and horizontal axis change from a profile to another.

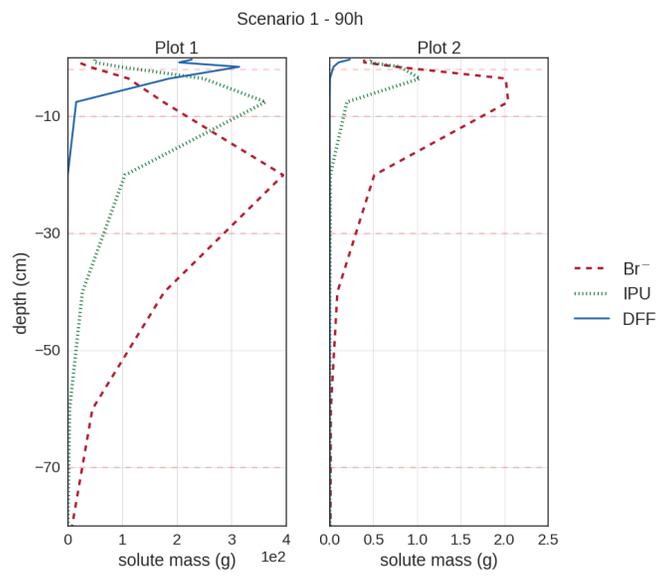


Figure 10: Comparison of mass solute profiles in Plots 1 and 2 after 90h of simulation for IPU, DFF and  $\text{Br}^-$  for Scenario 1.

		Unit	Infiltration	Evaporation	Root Extraction	Surface Runoff	Subsurface transfer		Longitudinal transfer
							Shallowly perched water table	Groundwater	
Scenario 1	Water	m <sup>3</sup>	346.0 (P1)	92.7 (P1)	338.4 (P1)	22.2	3.8 10 <sup>-2</sup>	74.1	—
			359.0 (P2)	92.7 (P2)	338.3 (P2)	(P1 →P2)	(P1 →P2)	(P1 →P2)	
	IPU	g	1003.5 (P1)	—	—	3.1	1.1 10 <sup>-2</sup>	1.4 10 <sup>-3</sup>	—
			3.1 (P2)			(P1 →P2)	(P1 →P2)	(P1 →P2)	
	DM-IPU	g	7.5 10 <sup>-3</sup> (P1)	—	—	7.4 10 <sup>-3</sup>	2.7 10 <sup>-5</sup>	3.2 10 <sup>-4</sup>	—
7.4 10 <sup>-3</sup> (P2)					(P1 →P2)	(P1 →P2)	(P1 →P2)		
DFE	g	963.4 (P1)	—	—	1.2	1.4 10 <sup>-4</sup>	9.4 10 <sup>-9</sup>	—	
		0.7 (P2)			(P1 →P2)	(P1 →P2)	(P1 →P2)		
Br <sup>-</sup>	g	1007.2 (P1)	—	—	6.2	3.5 10 <sup>-2</sup>	3.8 10 <sup>-2</sup>	—	
		6.2 (P2)			(P1 →P2)	(P1 →P2)	(P1 →P2)		
Scenario 2	Water	m <sup>3</sup>	346.0 (P1)	92.7 (P1)	338.4 (P1)	22.2	3.4 10 <sup>-2</sup>	74.3	24.0
			347.0 (P2)	92.7 (P2)	338.4 (P2)	(P1 →D1)	(P1 →D1)	(P1 →P2)	(D1 →D2)
	IPU	g	1003.5 (P1)	—	—	3.1	1.1 10 <sup>-2</sup>	1.4 10 <sup>-3</sup>	2.7
			0.0 (P2)			(P1 →D1)	(P1 →D1)	(P1 →P2)	(D1 →D2)
	DM-IPU	g	7.5 10 <sup>-3</sup> (P1)	—	—	7.3 10 <sup>-3</sup>	2.4 10 <sup>-5</sup>	3.2 10 <sup>-4</sup>	9.0 10 <sup>-3</sup>
			0.0 (P2)			(P1 →D1)	(P1 →D1)	(P1 →P2)	(D1 →D2)
	DFE	g	963.4 (P1)	—	—	1.2	1.2 10 <sup>-4</sup>	9.4 10 <sup>-9</sup>	1.05
			0.0 (P2)			(P1 →D1)	(P1 →D1)	(P1 →P2)	(D1 →D2)
	Br <sup>-</sup>	g	1007.2 (P1)	—	—	6.2	3.3 10 <sup>-2</sup>	3.8 10 <sup>-2</sup>	5.8
			0.0 (P2)			(P1 →D1)	(P1 →D1)	(P1 →P2)	(D1 →D2)

Table 4: Mass balance for Scenarios 1 and 2. P1 stands for Plot 1, P2 for Plot 2, D1 for Ditch 1 and D2 for Ditch 2.

the linear network dramatically increased the total amount of solutes transferred to the outlet ( $3.6 \cdot 10^{-3}$  g of IPU in Scenario 3 vs 8.5 g in Scenario 4) but it also affected the contributions of the different transfer paths to the discharge. In Scenario 3 the discharge was characterized by 2 contributions with different orders of magnitude (Figure 11). The first peak corresponds to a small amount of solutes that transferred to the river through surface runoff. Thereafter the discharge was mainly driven by slower lateral subsurface transfers from perched water table. The contribution of surface runoff remained relatively low due to ponding that slowed down surface transfers and promoted infiltration.

In Scenario 4 (Figure 11), the discharge was firstly characterized by a very strong peak of discharge associated to transfers through the ditch network. Then, all over the simulation solutes kept on being transferred to the river in smaller quantities. Such releases were associated to much slower subsurface transfers by shallow groundwater. In this latter case, solutes firstly infiltrated in plot P1 until reaching the shallow groundwater. Once they reach the water table, they laterally transferred below the ditch network to the river. It led to a longer transfer time than for solutes from surface that were intercepted by ditches and directly routed through the network. In the two cases, the exported mass to the outlet remains quite high and can be explained by the spatial scale that does not allow attenuation on such an event.

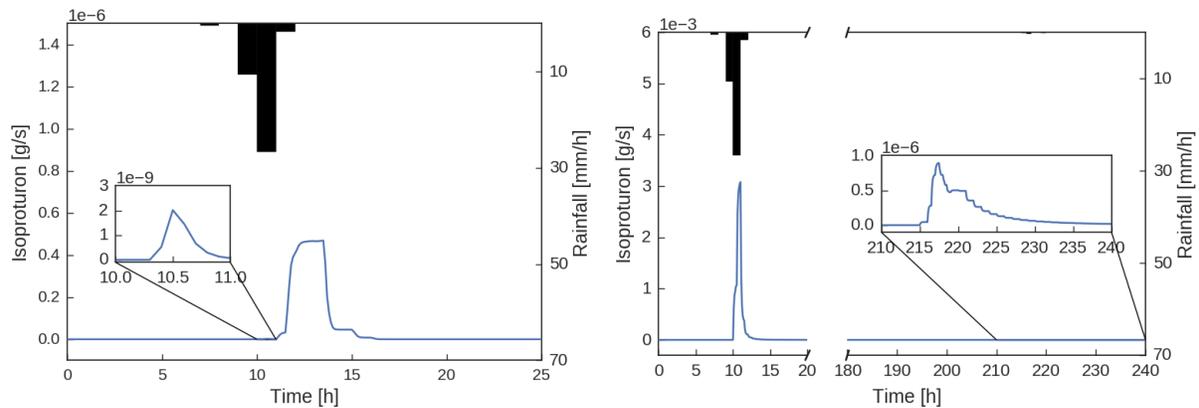


Figure 11: IPU discharge at the outlet of hillslope for Scenario 3 (left) and Scenario 4 (right). Rectangles in the figures represent zooms on the discharge for given periods of time.

Figure 12 reports for each scenario the evolution of IPU mass in each plot and ditch reach over time at surface and subsurface and all relative cumulative amounts are reported in Table 5. Scenarios 3 and 4 ran with a final error on mass balance less than  $10^{-8}$  m<sup>3</sup> for water and less than  $10^{-4}$  g for all solutes. Global mass balances and spatialized views of solute spreading over time show that transfer paths were also highly modified with the implantation of ditches. In Scenario 3, solutes spread quickly at the surface and the neighbouring plots were highly contaminated shortly after the beginning of the rain. The mass at surface decreased along time as the IPU infiltrated or was degraded. In the subsurface, IPU spread more slowly, mostly due to shallowly perched water tables that appeared because of the impermeable soil layer between 10 cm and 30 cm. As no diffusion is considered when modeling lateral transfers between plots, it led to strong temporal and spatial discontinuities for solute masses. On the contrary, in Scenario 4, all surface runoff or subsurface transfer from shallowly perched water table generated in Plot 1 was intercepted by the ditch network. Most of solutes intercepted by the ditch network were quickly transmitted to

the river while a small fraction remained adsorbed at the bottom of the ditch. Small amounts still transferred to the rest of the hillslope by lateral transfers through groundwater. Figure 13 shows distinct exportation patterns between IPU, DFF and Br<sup>-</sup> for scenario 4. As for scenario 1, mobile solutes such as Br<sup>-</sup> are quickly transferred over the hillslope whereas non-mobile solutes such as DFF are poorly transferred from the zone of application. Mass balances gathered in Table 5 also illustrate such contrasted behaviors. For example, for all runs of Scenario 4, lateral flow intercepted at surface or subsurface by the ditch network mainly contributed to solute export to the outlet. However, amount of Br<sup>-</sup> exported to the outlet is nearly twice as much as IPU exported amount and it is nearly six times as much as DFF exported amount.

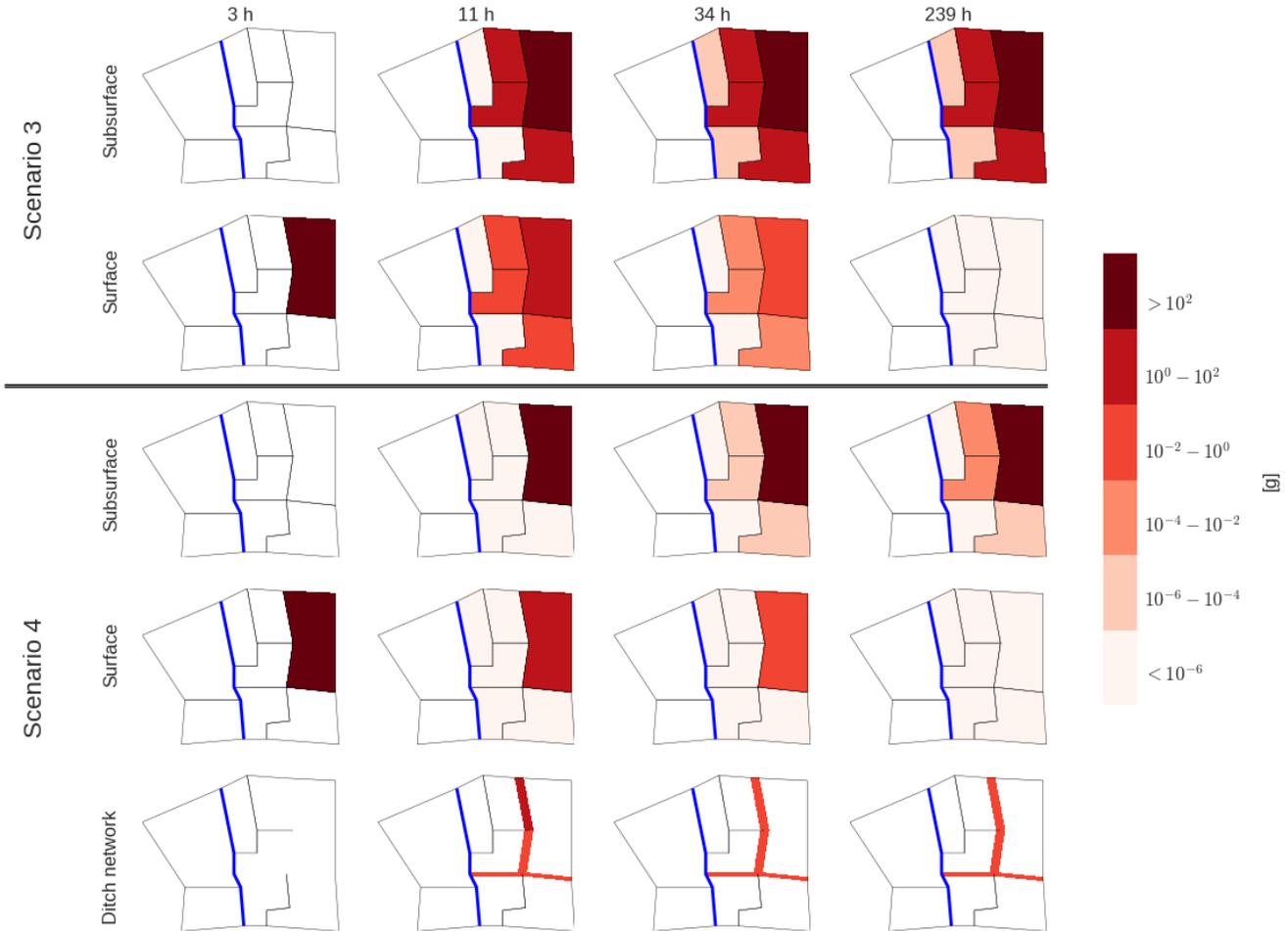


Figure 12: IPU mass along time over the hillslope for Scenarios 3 and 4.

#### 4. Discussion

In these first simulations, the main processes observed on the Kervidy catchment are successfully represented: formation of shallowly perched water tables and interception of subsurface transfers by ditch. Contrasted dynamics are simulated and show a clear effect of the presence of ditches that redirect and accelerate surface but also subsurface solute fluxes. Adsorption and degradation are properly represented by the model. They are simulated here with

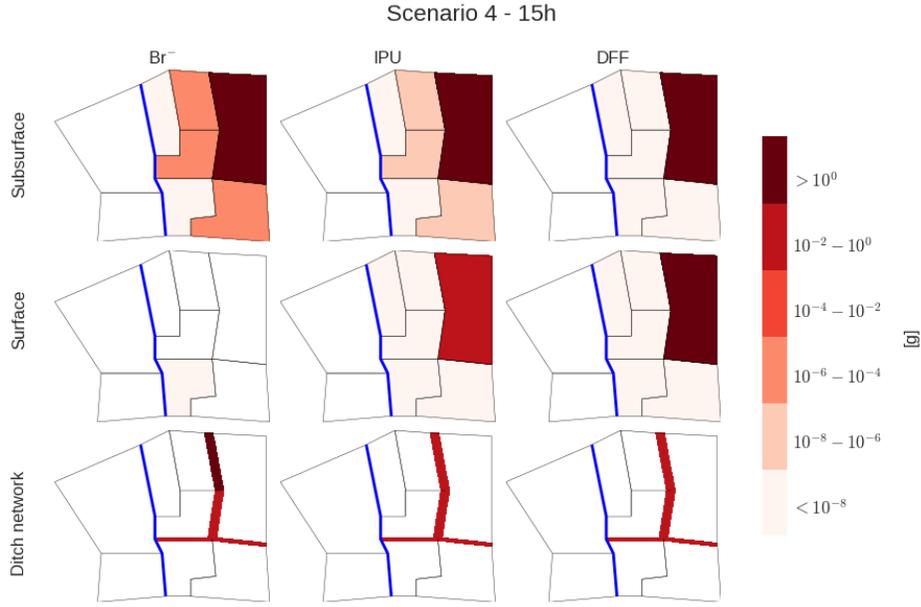


Figure 13: Comparison of mass spreading over the hillslope after 15h of simulation for IPU, DFF and  $\text{Br}^-$  for Scenario 4.

		Unit	Surface runoff to river	Lateral saturated transfer to river	Transmitted by ditch network	Transmitted to the outlet
Scenario 3	Water	$\text{m}^3$	71.5	$9.7 \cdot 10^{-2}$	–	$7.1 \cdot 10^4$
	Mass of IPU	g	$1.8 \cdot 10^{-6}$	$3.0 \cdot 10^{-3}$	–	$3.6 \cdot 10^{-3}$
	Mass of DM-IPU	g	$1.1 \cdot 10^{-9}$	$3.4 \cdot 10^{-6}$	–	$3.4 \cdot 10^{-6}$
	Mass of DFF	g	$5.7 \cdot 10^{-9}$	$2.1 \cdot 10^{-6}$	–	$2.1 \cdot 10^{-6}$
	Mass of $\text{Br}^-$	g	$2.2 \cdot 10^{-5}$	$8.4 \cdot 10^{-2}$	–	$8.4 \cdot 10^{-2}$
Scenario 4	Water	$\text{m}^3$	63.9	$9.6 \cdot 10^{-2}$	77.2	$7.1 \cdot 10^4$
	Mass of IPU	g	0	$5.6 \cdot 10^{-7}$	8.5	8.5
	Mass of DM-IPU	g	0	$1.7 \cdot 10^{-7}$	$2.7 \cdot 10^{-2}$	$2.7 \cdot 10^{-2}$
	Mass of DFF	g	0	$3.8 \cdot 10^{-12}$	3.0	3.0
	Mass of $\text{Br}^-$	g	0	$1.7 \cdot 10^{-5}$	17.6	17.6

Table 5: Cumulative amounts of water and solutes transferred to outlet through the different pathways at the end of simulations for Scenarios 3 and 4.

parameters from the Pesticide Properties Database - PPDB (Lewis et al., 2016) as field data were lacking. However, such parameters are obtained in laboratory or for given field conditions and do not allow for taking into account specific soil properties from the field of application. If field data are available, users can include them in the model for better catching soil-dependency and spatial variability of those processes. In the same way, formation of metabolite is successfully represented. For these simulations, DM-IPU was considered as the only metabolite with a 100%-occurrence fraction to test the feasibility of integrating metabolite fate in the model. This simplistic representation can evolve in more realistic application cases depending on the user's knowledge about metabolite formation and transformation. Finally, solutes with distinct properties and contrasted behaviors can also be simulated by the model but much longer simulations should be performed to assess properly DFF degradation that is characterized by a long half-life time.

Still, some processes and elements are currently represented in a simplistic way or are just missing in the model. Such choices are justified to illustrate dominant processes on the catchment chosen for the first simulations and they can evolve thanks to PESHMELBA modular structure. Yet, one should keep in mind that such a tool aims at comparing and ranking landscape management scenarios for decision making rather than precisely predicting solute amounts at any point of the catchment. Actually, to our knowledge, no model is currently able to simulate those interactive processes at such a scale and under real field conditions. As a result, efforts made to improve the simulations should focus mainly on dominating processes that rule solute behavior at the catchment scale. For example, description of degradation in PESHMELBA currently does not take into account influence of temperature or soil moisture. The code can easily be modified to include such relations as the ones from Larsbo & Jarvis (2003) for example. Additional developments should also depend on what the user knows about the functioning of the targeted catchment. In the application cases described in this paper, ditches functioning is limited to surface and subsurface flow interception. Such hypothesis is relevant in the case of the Kervidy catchment where ditches functioning is mainly drainage of water during wet season. It may be necessary to modify it in other contexts and to include infiltration within the ditch depending on its characteristics. More generally, other processes that play a key role on pesticide transfers should also be added such as macropore flow in soils (Beven & Germann, 2013; Djabelkhir et al., 2017) or sediment transport (Muñoz-Carpena et al., 1999; Warren et al., 2003). Some of them can be included with mechanistic additive units, but they may also be added more simply with transfer functions (Jury & Horton, 2004). It must be pointed out however that the model structure is limited to include diffusion although it would probably limit the discontinuities in solute mass due to spatial discretization.

Before reaching a complete representation of an agricultural watershed, other elements also need to be implemented. The implementation of buffer zones such as vegetative filter strips, hedges or artificial wetlands will be the next step to test different options for best management practices for transfers attenuation. Drained plots will also be included as they are commonly set up on agricultural watersheds. To that aim, we will use existing modeling knowledge such as Branger et al. (2009, 2010) for tile-drained plots, Krone-Davis et al. (2013) for wetlands or Lauvernet & Muñoz-Carpena (2018) for buffer zones. Metamodeling approaches (Villa-Vialaneix et al., 2012; Lauvernet et al., 2016) can also be used to simplify units based on mechanistic models, to limit the number of input parameters and to make the numerical cost reasonable. Indeed, these application cases show that the calculation time of the model may become a problem when making more complex scenarios (calculation time increased from 108s in Scenario 1

415 to 1285s in Scenario 4). In order to explore different scenarios of landscape organization efficiently, it is crucial to optimize the calculation time of the model. The River1D unit that handles routing in the network has been identified as the most limiting process in term of calculation performances. It should probably be improved by considering the resolution scheme for this unit.

## 5. Conclusion

420 In this paper, an original approach for modeling water and pesticide transfer and fate in small agricultural landscapes has been proposed with some application cases comparing management scenarios of a hillslope. PESHMELBA makes it possible to take into account the diversity of landscape elements and to assess their influence on transfers by easily adding or removing new components, but also modifying the description of the processes. The landscape is described as a combination of linear and surface elements connected in surface and subsurface. Relevant 425 processes for water circulation and pesticide fate are represented on each element individually with different levels of complexity. The coupling performed by the OpenPALM coupler is innovative and very efficient, giving high flexibility to handle spatially distributed modeling and complex time management, and to include distinct elements and processes with different conceptualizations. The scenarios tested in this paper showed that PESHMELBA is able to simulate water and solute fate in various configurations that are easily implemented. However, some elements still 430 need to be included before reaching applications on real agricultural catchments, in particular grass buffer strips, hedges, artificial wetlands or drained plots. Future research should focus on the evaluation of PESHMELBA at the catchment scale. It should be performed by applying PESHMELBA on a real agricultural catchment with comprehensive experimental datasets such as the Kervidy catchment. Evaluation on real catchments with contrasted characteristics should also be targeted. Global sensitivity analysis methods is recommended to analyze the influence 435 of input factors in such a modular tool, but also to evaluate the associated uncertainty (Fu et al., 2019). Finally, to reduce this uncertainty, data assimilation will make it possible to combine satellite or field observations with this physico-conceptual approach and to constrain the model to respect observations. Although many improvements may be developed, this study proves the feasibility and the facilities to add new elements and to build such a model. PESHMELBA is a new and promising tool that could be used in the next future for contamination risk management 440 or for local agricultural watershed management dedicated to improve surface water quality.

## Acknowledgement

This research was supported by the French Agency for the Biodiversity.

## Appendix A. Nomenclature

---

$a_{bl}$	[-]	radiation interception by the vegetation coefficient
$c$	[ML <sup>-3</sup> ]	solute concentration in water
$c_0$	[ML <sup>-3</sup> ]	solute concentration in solution at time 0
$\Delta h$	[L]	head loss between groundwater and the river

$\Delta_i$	[L]	thickness of water table within the soil horizon $i$
$\delta$	[-]	fraction of wet vegetation
$D$	[L <sup>2</sup> T <sup>-1</sup> ]	hydrodynamic dispersion coefficient
$DT50$	[T <sup>-1</sup> ]	half-life
$\eta$	[-]	Brooks and Corey hydraulic conductivity shape parameter
$E_p$	[LT <sup>-1</sup> ]	potential evaporation
$E_w$	[LT <sup>-1</sup> ]	energy used to evaporate water from the interception tank
$F_{10}$	[-]	fraction of the root length density in the top 10% of the root zone
$f_{oc}$	%	soil organic carbon content
$\gamma$	[-]	anisotropy coefficient between horizontal and vertical hydraulic conductivity
$H_r$	[L]	river stage level
$h$	[L]	pressure head
$h_e$	[L]	capillary pressure (bubbling pressure)
$h_w$	[L]	water level on a plot
$K_h$	[LT <sup>-1</sup> ]	vertical hydraulic conductivity
$K_d$	[L <sup>3</sup> M <sup>-1</sup> ]	linear coefficient of adsorption
$K_f$	[(L <sup>3</sup> ) <sup><math>n_f</math></sup> M <sup>1-<math>n_f</math></sup> M <sup>-1</sup> ]	Freundlich coefficient of adsorption
$K_{doc}$	[L <sup>3</sup> M <sup>-1</sup> ]	linear organic carbon-water partition coefficient
$K_{foc}$	[(L <sup>3</sup> ) <sup><math>n_f</math></sup> M <sup>1-<math>n_f</math></sup> M <sup>-1</sup> ]	non-linear organic carbon-water partition coefficient
$K_{s,h}$	[LT <sup>-1</sup> ]	horizontal hydraulic conductivity at saturation
$K_{s,v}$	[LT <sup>-1</sup> ]	vertical hydraulic conductivity at saturation
$K_{s,h}^{int}$	[LT <sup>-1</sup> ]	horizontal hydraulic conductivity at saturation at the interface
$\lambda$	[-]	Brooks and Corey shape parameter (pore size index)
$LAI$	[-]	Leaf Area Index
$L_i$	[-]	length of interface with a given element $i$ connected downstream
$n_f$	[-]	exponent of Freundlich equation for adsorption equilibrium
$n$	[LT <sup>1/3</sup> ]	Manning coefficient
$P$	[LT <sup>-1</sup> ]	rain intensity above the canopy
$P_g$	[LT <sup>-1</sup> ]	actual rainfall rate that reaches the soil surface
$PET$	[LT <sup>-1</sup> ]	potential evapotranspiration
$q_{ads}$	[M M <sup>-1</sup> ]	mass of solute adsorbed per mass of soil at the equilibrium
$q_z$	[LT <sup>-1</sup> ]	vertical pore velocity
$\sigma_f$	[-]	fraction of solar radiation and precipitation intercepted by the vegetation
$S_0$	[LL <sup>-1</sup> ]	slope on the plot
$s$	[L]	length of seepage face
$s_i$	[LL <sup>-1</sup> ]	altitude gradient between the centroid of the plot and a given element $i$ connected downstream
$\theta$	[L <sup>3</sup> L <sup>-3</sup> ]	volumetric water content

$\theta_s$	[L <sup>3</sup> L <sup>-3</sup> ]	volumetric water content at saturation
$T_r$	[LT <sup>-1</sup> ]	real transpiration
$T_p$	[LT <sup>-1</sup> ]	potential transpiration
$W_b$	[L]	bottom width
$W_t$	[L]	top width
$w_i$	[-]	splitting weight for runoff toward a given element $i$ connected downstream
$w_{WT}^i$	[-]	weight associated to numerical cell $i$ within the water table WT for lateral saturated transfer
$w_{rmax}$	[L]	maximal height of leaves interception tank
$w_{ro}$	[L]	water height that overflows from the interception tank
$Z_R$	[L]	rooting depth
$z_i$	[L]	thickness of numerical cell $i$

## Appendix B. Details on PESHMELBA units

### 445 Appendix B.1. FRER1D

The FRER1D (Fast 1D Richards Equation Resolution) model solves the vertical Richards' equation (Richards, 1931) (eq.B.1) on homogeneous and heterogeneous soils, that is:

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z} [K_v(h) \left( \frac{\partial h}{\partial z} - 1 \right)] + R \quad (\text{B.1})$$

where  $\theta$  is the volumetric water content [L<sup>3</sup>L<sup>-3</sup>],  $h$  the pressure head [L],  $K_v$  the vertical hydraulic conductivity [LT<sup>-1</sup>] and  $R$  a potential source/sink term [L<sup>3</sup>L<sup>-3</sup>T<sup>-1</sup>].

450 In this version of PESHMELBA, the numerical method relies on the use of the Brooks and Corey soil coefficients (Brooks & Corey, 1964) (eq.B.2) which allows an analytical formulation of the Kirchhoff transform proposed by Gardner (1958) and Campbell (1985) (eq.B.3) for each soil horizon:

$$\frac{\theta}{\theta_s} = \left( \frac{h}{h_e} \right)^{-\lambda}$$

$$\frac{K_v}{K_{s,v}} = \begin{cases} \left( \frac{\theta}{\theta_s} \right)^\eta, & \text{si } h < h_e \\ 1, & \text{si } h \geq h_e \end{cases} \quad (\text{B.2})$$

$$\phi = \int_{-\infty}^h K_v(h) dh \quad (\text{B.3})$$

where  $h_e$  is the air entry pressure [L],  $K_{s,v}$  the vertical hydraulic conductivity at saturation [LT<sup>-1</sup>], and  $\lambda$  and  $\eta$  the shape parameters of the Brooks and Corey model [-]. Regarding solute transport and transformation, the original version of FRER1D solves the 1D advection-dispersion equation based on the same numerical scheme than for water transport:

$$D \frac{\partial^2 c}{\partial z^2} - q_z \frac{\partial c}{\partial z} + r = \frac{\partial c}{\partial t} \quad (\text{B.4})$$

where  $D$  is the coefficient of hydrodynamic dispersion [ $L^2T^{-1}$ ] (the contribution of diffusion to solute dispersion is ignored in this solution),  $c$  the pesticide concentration in solution at the equilibrium [ $ML^{-3}$ ],  $q_z$  the vertical pore velocity [ $LT^{-1}$ ] and  $r$  a potential source/sink term [ $ML^{-3}T^{-1}$ ].

#### 460 Appendix B.2. PARTITION and ROOT

Potential evaporation and rain rate are converted into rain reaching the soil surface and plant transpiration in the PARTITION module as previously done by Varado et al. (2006a). First, the potential evapotranspiration  $PET$  [ $LT^{-1}$ ] is split between potential evaporation and potential transpiration using a Beer-Lambert law and the LAI [-] (Leaf Area Index) (Huygen et al., 1997):

$$T_p = PET(1 - \exp(-a_{bl}LAI)) \quad (B.5)$$

$$E_p = PET \exp(-a_{bl}LAI) \quad (B.6)$$

where  $a_{bl}$  [-] is a coefficient that accounts for interception of the radiation by the vegetation.  $E_p$  is directly used in FRER1D as a surface boundary condition. Leaves constitute an interception tank which maximal height  $w_{rmax}$  [L] is a function of the LAI (Noilhan & Planton, 1989):

$$w_{rmax} = 2.10^{-4} \sigma_f LAI \quad (B.7)$$

where  $\sigma_f$  is the fraction of solar radiation and precipitation intercepted by the vegetation [-] (Taconet et al., 1986):

$$\sigma_f = 1 - \exp(-a_{bl}LAI) \quad (B.8)$$

The actual rainfall rate  $P_g$  [ $LT^{-1}$ ] that reaches the soil surface is deduced by adding the rain that is not intercepted by the vegetation and the water that overflows from the interception tank:

$$P_g = P * \exp(-a_{bl}LAI) + \frac{w_{ro}}{dt} \quad (B.9)$$

465 where  $P$  is the rain intensity above the canopy [ $LT^{-1}$ ] and  $w_{ro}$  the water height that overflows from the interception tank [L].

Finally, water root extraction is determined for each numerical cell of the soil column in the ROOT unit as done in Li et al. (2001).  $R_{ex}$  is expressed as a linear function of the potential transpiration and three empirical coefficients:

$$R_{ex} = \alpha \beta g T_p \quad (B.10)$$

where  $\alpha$  stands for the compensation mechanism,  $\beta$  for water stress and  $g$  for the root density. Such coefficients are function of the rooting depth  $Z_r$  [L] and  $F_{10}$  the fraction of the root length density in the top 10% of the root zone. They are extensively described in Li et al. (2001) and Braud et al. (2005).

#### 470 Appendix B.3. PPI

##### Appendix B.3.1. Hydraulic conductivity calculation

$K_{s,h}^{int}$  is calculated as the harmonic mean between  $K_{s,h}^x$  and  $K_{s,h}^y$ , the horizontal conductivities at saturation associated to the cells involved in the transfer for each column (Bouwer, 1969):

$$\frac{d_x + d_y}{K_{s,h}^{int}} = \frac{d_x}{K_{s,h}^x} + \frac{d_y}{K_{s,h}^y} \quad (B.11)$$

Where  $d_x$  and  $d_y$  are the distances from the centroids of the columns to the interface [L]. For a given column  $Col_x$ , if a water table extends on M soil horizons, the associated horizontal hydraulic conductivity  $K_{s,h}^x$  is expressed as :

$$K_{s,h}^x = \frac{\sum_{i=1}^M \Delta^i K_{s,h}^i}{\sum_{i=1}^M \Delta^i} \quad (\text{B.12})$$

Where  $\Delta_i$  is the thickness of the water table within the soil horizon  $i$  [L] and  $K_{s,h}^i$  the horizontal hydraulic conductivity at saturation associated to the soil horizon  $i$  [ $\text{LT}^{-1}$ ]. Various authors have reported anisotropy of hydraulic conductivity with higher conductivity in the horizontal direction than in the vertical direction (Kanwar et al., 1990; Zhang, 1996). In PESHMELBA, the user provides a vertical hydraulic conductivity and anisotropy is taken into account with an anisotropy coefficient  $\gamma$  ( $\gamma = K_{s,h}/K_{s,v}$ ).

### Appendix B.3.2. Coefficients for flow splitting

For a given water table  $WT$  spread on N numerical cells, the weight  $w_{WT}^i$  for cell  $i$ ,  $i \in \{1, \dots, N\}$  is :

$$w_{WT}^i = \frac{\frac{z^i}{K_{s,h}^i}}{\sum_{j=1}^N \frac{z^j}{K_{s,h}^j}} \quad (\text{B.13})$$

where  $z^i$  is the thickness of the numerical cell  $i$  [L] and  $K_{s,h}^i$  its horizontal hydraulic conductivity at saturation [ $\text{LT}^{-1}$ ]. Upstream, those coefficients are equal to 0 for non saturated cells.

## Appendix C. Parameters for vegetation used in application tests

Crop coefficient	-	1.0
Extinction coefficient	-	0.5
Pressure heads that determine the $\alpha$ sink term in Li et al. (2001) root extraction formulation h1, h2, h3a, h3b, h4	m	-0.1/-0.25/-5.0/-10.0/-150.0
Associated potential transpiration e1, e2	m/s	$1.16 \cdot 10^{-6}$ / $5.79 \cdot 10^{-6}$
LAI	-	Increases linearly between 2.2 at t=0h and 4 at h=240h
Zr	m	Increases linearly between 0.15 at t=0h and 0.471 at h=240h
F10	-	Increases linearly between 0 at t=0h and 0.29 at h=240h

Table C.7: . Set of vegetation parameters used in the simulations.

## References

- Abbott, M., Bathurst, J., Cunge, J., O'Connell, P., & Rasmussen, J. (1986). An introduction to the european hydrological system — systeme hydrologique europeen, “she”, 2: Structure of a physically-based, distributed modelling system. *Journal of Hydrology*, *87*, 61 – 77. URL: <http://www.sciencedirect.com/science/article/pii/0022169486901150>. doi:10.1016/0022-1694(86)90115-0.
- Adriaanse, P. (1997). Exposure assessment of pesticides in field ditches: The TOXSWA model. *Pesticide Science*, *49*, 210–212. doi:10.1002/(SICI)1096-9063(199702)49:2<210::AID-PS496>3.0.CO;2-1.
- Adriaanse, P., Boesten, J., & Crum, S. (2013). Estimating degradation rates in outdoor stagnant water by inverse modelling with toxswa: A case study with prosulfocarb. *Pest Management Science*, *69*, 755–767. doi:10.1002/ps.3435.
- Alix, A., Brown, C., Capri, E., Goerlitz, G., Golla, B., Knauer, K., Laabs, V., Mackay, N., Vasile, A., Alonso Prados, E., Reinert, W., Streloke, M., & Poulsen, V. (2017). *Mitigating the Risks of Plant Protection Products in the Environment: MAgPIE*. SETAC.
- Argent, R., Perraud, J.-M., Rahman, J., Grayson, R., & Podger, G. (2009). A new approach to water quality modelling and environmental decision support systems. *Environmental Modelling & Software*, *24*, 809 – 818. URL: <http://www.sciencedirect.com/science/article/pii/S1364815208002351>. doi:10.1016/j.envsoft.2008.12.010.
- Arnold, J., Srinivasan, R., Muttiah, R., & Williams, J. (1998). Large area hydrologic modeling and assessment - Part 1: Model development. *Journal of the American Water Resources Association*, *34*, 73–89. doi:10.1111/j.1752-1688.1998.tb05961.x.
- Ashby, S., & Falgout, R. (1996). A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. *Nuclear Science and Engineering*, *124*, 145–159. doi:10.13182/NSE96-A24230.
- Barthélémy, S., Ricci, S., Morel, T., Goutal, N., Pape, E. L., & Zaoui, F. (2018). On operational flood forecasting system involving 1D/2D coupled hydraulic model and data assimilation. *Journal of Hydrology*, *562*, 623 – 634. doi:10.1016/j.jhydrol.2018.05.007.
- Beckers, J., & Frind, E. (2000). Simulating groundwater flow and runoff for the oro moraine aquifer system. part i. model formulation and conceptual analysis. *Journal of Hydrology*, *229*, 265 – 280. URL: <http://www.sciencedirect.com/science/article/pii/S0022169400001670>. doi:10.1016/S0022-1694(00)00167-0.
- Beisman, J. J., Maxwell, R. M., Navarre-Sitchler, A. K., Steefel, C. I., & Molins, S. (2015). Parcrunchflow: an efficient, parallel reactive transport simulation tool for physically and chemically heterogeneous saturated subsurface environments. *Computational Geosciences*, *19*, 403–422. URL: <https://doi.org/10.1007/s10596-015-9475-x>. doi:10.1007/s10596-015-9475-x.
- Beltman, W., & Adriaanse, P. (1999). *User's manual TOXSWA 1.2; Simulation of pesticides fate in small surface waters*. Technical document 54 Winand Staring Centre, Wageningen.

- Beven, K. (1979). On the generalized kinematic routing method. *Water Resources Research*, *15*, 1238–1242. doi:10.1029/WR015i005p01238.
- Beven, K., & Germann, P. (1982). Macropores and water flow in soils. *Water Resources Research*, *18*, 1311–1325. doi:10.1029/WR018i005p01311.
- 520 Beven, K., & Germann, P. (2013). Macropores and water flow in soils revisited. *Water Resources Research*, *49*, 3071–3092. doi:10.1002/wrcr.20156.
- Bingner, T. F. Y. Y., R.L. (2011). *AnnAGNPS technical processes documentation, Version 5.2*. Technical Report United States Department of Agriculture.
- Bouwer, H. (1969). Planning and interpreting soil permeability measurements. *Journal of the Irrigation and*  
525 *Drainage Division*, *95*, 391–402.
- Branger, F., Braud, I., Debionne, S., Viallet, P., Dehotin, J., Henine, H., Nedelec, Y., & Anquetin, S. (2010). Towards multi-scale integrated hydrological models using the LIQUID® framework. overview of the concepts and first application examples. *Environmental Modelling and Software*, *25*, 1672–1681. doi:10.1016/j.envsoft.2010.06.005.
- 530 Branger, F., Tournebize, J., Carluer, N., Kao, C., Braud, I., & Vauclin, M. (2009). A simplified modelling approach for pesticide transport in a tile-drained field: The PESTDRAIN model. *Agricultural Water Management*, *96*, 415–428. doi:10.1016/j.agwat.2008.09.005.
- Braud, I., Varado, N., & Olioso, A. (2005). Comparison of root water uptake modules using either the surface energy balance or potential transpiration. *Journal of Hydrology*, *301*, 267 – 286. doi:10.1016/j.jhydro1.2004.06.033.
- 535 Brooks, R. H., & Corey, A. T. (1964). *Hydraulic Properties of Porous Media*. Technical Report 1 Colorado State University, Fort Collins.
- Buchanan, B. P., Falbo, K., Schneider, R. L., Walter, M. T., & Easton, Z. M. (2013). Hydrological impact of roadside ditches in an agricultural watershed in Central New York: implications for non-point source pollutant transport. *Hydrological Processes*, *27*, 2422–2437. doi:10.1002/hyp.9305.
- 540 Buis, S., Piacentini, A., & Déclat, D. (2006). PALM: a computational framework for assembling high-performance computing applications. *Concurrency and Computation: Practice and Experience*, *18*, 231–245. doi:10.1002/cpe.914.
- Buytaert, W., Reusser, D., Krause, S., & J.-P., R. (2008). Why can't we do better than Topmodel? *Hydrological Processes*, *22*, 4175–4179. doi:10.1002/hyp.7125.
- 545 arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/hyp.7125>.
- Campbell, G. (1985). *Soil Physics With Basic Transport Models for Soil-Plant Systems* volume 14. Elsevier.
- Campbell, N., D'Arcy, B., Frost, A., Novotny, V., & Sampson, A. (2004). *Diffuse Pollution: An Introduction to the Problems and Solutions*. IWA Publishing.

- 550 Camporese, M., Paniconi, C., Putti, M., & Orlandini, M. (2010). Surface-subsurface flow modeling with path-based runoff routing, boundary condition-based coupling, and assimilation of multisource observation data. *Water Resources Research*, *46*. doi:10.1029/2008WR007536.
- Carluer, N., Catalogne, C., Dages, C., & Tournebize, J. (2017a). Aménager le territoire et gérer les aménagements : les zones tampons sèches et humides, les fossés pour lutter contre les pollutions diffuses par les produits phytosanitaires dans les aires de captage. *Innovations Agronomiques*, *57*, 117–139.
- 555 Carluer, N., Lauvernet, C., Noll, D., & Muñoz-Carpena, R. (2017b). Defining context-specific scenarios to design vegetated buffer zones that limit pesticide transfer via surface runoff. *Science of The Total Environment*, *575*, 701 – 712. doi:10.1016/j.scitotenv.2016.09.105.
- Carluer, N., & Marsily, G. D. (2004). Assessment and modelling of the influence of man-made networks on the hydrology of a small watershed: implications for fast flow components, water quality and landscape management. 560 *Journal of Hydrology*, *285*, 76 – 95. doi:10.1016/j.jhydro1.2003.08.008.
- Carsel, J. E., & Baldwin, J. E. (2000). *PRZM-3, A model for predicting Pesticide and nitrogen fate in the crop root and unsaturated soil zones*. Technical Report Environmental Protection Agency.
- Cash, R. J., & Karp, A. (1990). A variable order RungeKutta method for value problems with rapidly varying right-hand sides. *ACM Transactions on Mathematical Software*, *16*, 201–222. doi:10.1145/79505.79507.
- 565 Cheverry, C. (1998). *Agriculture intensive et qualité des eaux*. Editions Quae.
- Chiew, F., McMahon, T., & O’Neill, I. (1992). Estimating groundwater recharge using an integrated surface and groundwater modelling approach. *Journal of Hydrology*, *131*, 151 – 186. URL: <http://www.sciencedirect.com/science/article/pii/002216949290216I>. doi:10.1016/0022-1694(92)90216-I.
- Crevoisier, D., Chanzy, A., & Voltz, M. (2009). Evaluation of the Ross fast solution of Richards’ equation in 570 unfavourable conditions for standard finite element methods. *Advances in Water Resources*, *32*, 936–947. doi:10.1016/j.advwatres.2009.03.008.
- Dauplain, A., Favier, J., & Bottaro, A. (2008). Hydrodynamics of ciliary propulsion. *Journal of Fluids and Structures*, *24*, 1156 – 1165. doi:10.1016/j.jfluidstructs.2008.06.007.
- De Roo, A., Wesseling, C., Jetten, V., & Ritsema, C. (1996). LISEM: A physically-based hydrological and soil 575 erosion model incorporated in a GIS. In *In: K. Kovar & H.P. Nachtnebel (eds.), Application of geographic information systems in hydrology and water resources management. Wallingford (UK), IAHS, 1996. IAHS Publ. 235, pp. 395-403* (pp. 395 – 403).
- Dehotin, J. (2007). *Prise en compte de l’hétérogénéité des surfaces continentales dans la modélisation hydrologique spatialisée. Application sur le haut-bassin de la Saône..* Ph.D. thesis Institut National Polytechnique de Grenoble 580 France, Europe.

- Djabekhir, K. (2015). *Computer modeling of pesticide fate at hillslope scale within a hydrological modeling framework taking into account macroporosity (in French)*. Ph.D. thesis Université Grenoble Alpes.
- Djabekhir, K., Lauvernet, C., Kraft, P., & Carlier, N. (2017). Development of a dual permeability model within a hydrological catchment modeling framework: 1d application. *Science of The Total Environment*, 575, 1429 – 1437. doi:10.1016/j.scitotenv.2016.10.012.
- 585
- Dollinger, J. (2016). *Analyse et modélisation des transferts et de la rétention de pesticides dans les fossés agricoles infiltrants en lien avec les stratégies d'entretien*. Ph.D. thesis Laboratoire d'étude des Interactions Sol - Agrosystème - Hydrosystème (LISAH).
- Dollinger, J., Dagès, C., Negro, S., Bailly, J.-S., & Voltz, M. (2016). Variability of glyphosate and diuron sorption capacities of ditch beds determined using new indicator-based methods. *Science of The Total Environment*, 573, 716 – 726. doi:10.1016/j.scitotenv.2016.08.168.
- 590
- Dosskey, M. G., Helmers, M. J., & Eisenhauer, D. E. (2006). An approach for using soil surveys to guide the placement of water quality buffers. *Journal of Soil and Water Conservation*, 61, 344–354. doi:61(6):344-354.
- Dubus, I., & Surdyk, N. (2006). State-of-the-art review on pesticide fate models and environmental indicators. URL: [http://www.eu-footprint.org/downloads/FOOTPRINT\\_DL4.pdf](http://www.eu-footprint.org/downloads/FOOTPRINT_DL4.pdf).
- 595
- Duchaine, F., Jauré, S., Poitou, D., Quémerais, E., Staffelbach, G., Morel, T., & L.Y.M., G. (2015). Analysis of high performance conjugate heat transfer with the openpalm coupler. *Computational Science & Discovery*, 8. doi:10.1088/1749-4699/8/1/015003.
- Elsaesser, D., Stang, C., Bakanov, N., & Schulz, R. (2013). The Landau stream mesocosm facility: Pesticide mitigation in vegetated flow-through streams. *Bulletin of Environmental Contamination and Toxicology*, 90, 640–645. doi:10.1007/s00128-013-0968-9.
- 600
- Fabre, J.-C., Louchart, X., Colin, F., Dagès, C., Moussa, R., Rabotin, M., Raclot, D., Lagacherie, P., & Voltz, M. (2010). OpenFLUID : a software environment for modelling fluxes in landscapes. In *LandMod 2010: International Conference on Integrative Landscape Modelling . 2010; International Conference on Integrative Landscape Modelling, Montpellier, FRA, 2010-02-03-2010-02-05, 1-13* (pp. –).
- 605
- Fatichi, S., Vivoni, E., Ogden, F., Ivanov, V., Mirus, B., Gochis, D., Downer, C., Camporese, M., Davison, J., Ebel, B., Jones, N., Kim, J., Mascaro, G., Niswonger, R., Restrepo, P., Rigon, R., Shen, C., Sulis, M., & Tarboton, D. (2016). An overview of current applications, challenges, and future trends in distributed process-based models in hydrology. *Journal of Hydrology*, 537, 45–60. doi:10.1016/j.jhydro1.2016.03.026.
- 610
- Flury, M., Leuenberger, J., Studer, B., & Flühler, H. (1995). Transport of anions and herbicides in a loamy and a sandy field soil. *Water Resources Research*, 31, 823–836. doi:10.1029/94WR02852.
- Focus (2007). *Landscape and mitigation factors in aquatic risk assessment. Volume 2: detailed technic*. Technical Report European Commission.

- Fouilloux, A., & Piacentini, A. (1999). The PALM project: MPMD paradigm for an oceanic data assimilation software. In *Euro-Par'99 Parallel Processing: 5th International Euro-Par Conference Toulouse, France, August 31 - September 3, 1999 Proceedings* (pp. 1423–1430). Berlin, Heidelberg: Springer Berlin Heidelberg.
- 615 Fu, B., Merritt, W. S., Croke, B. F., Weber, T. R., & Jakeman, A. J. (2019). A review of catchment-scale water quality and erosion models and a synthesis of future prospects. *Environmental Modelling & Software*, *114*, 75–97. URL: <http://www.sciencedirect.com/science/article/pii/S1364815218307461>. doi:10.1016/j.envsoft.2018.12.008.
- 620 2018.12.008.
- Gaillardon, P., & Sabar, M. (2006). Changes in the concentration of isoproturon und its degradation products in soil und soil solution during incubation at two different temperatures. *Weed Research*, *34*, 243–250. doi:10.1111/j.1365-3180.1994.tb01992.x.
- Gardner, W., R (1958). Some steady-state solutions of the unsaturated moisture flow equation with application to evaporation from a water table. *Soil Science*, *85*, 228–232. doi:10.1097/00010694-195804000-00006.
- 625 evaporation from a water table. *Soil Science*, *85*, 228–232. doi:10.1097/00010694-195804000-00006.
- Gassmann, M., Stamm, C., Olsson, O., Lange, J., Kümmerer, K., & Weiler, M. (2013). Model-based estimation of pesticides and transformation products and their export pathways in a headwater catchment. *Hydrology and Earth System Sciences*, *17*, 5213–5228. doi:10.5194/hess-17-5213-2013.
- Gatel, L., Lauvernet, C., Carluer, N., Weill, S., Tournebize, J., & Paniconi, C. (2018). Global evaluation and sensitivity analysis of a physically based flow and reactive transport model on a laboratory experiment. *Environmental Modelling and Software*, .
- 630 *Environmental Modelling and Software*, .
- Gevaert, V., Van Griensven, A., Holovet, K., Seunjtjens, J., & P.A., V. (2008). SWAT developments and recommendations for modelling agricultural pesticide mitigation measures in river basins. *Hydrological Sciences Journal*, *53*, 1075–1089. doi:10.1623/hysj.53.5.1075. arXiv:<https://doi.org/10.1623/hysj.53.5.1075>.
- 635 Hutchings, C., Struve, S., J. Westen, Millard, K., & Fortune, D. (2002). *HarmonIT work package 1 : State of the art review*. Technical Report HR Wallingford.
- Huygen, J., Van Dam, J., Kroes, J., & Wesseling, J. (1997). *SWAP 2.0: input and output manual*. Wageningen Agricultural University.
- Ippolito, A., & Fait, G. (2019). Pesticides in surface waters: from edge-of-field to global modelling. *Current Opinion in Environmental Sustainability*, *36*, 78–84. URL: <http://www.sciencedirect.com/science/article/pii/S1877343518300290>. doi:doi.org/10.1016/j.cosust.2018.10.023. Environmental Change Assessment.
- 640 *in Environmental Sustainability*, *36*, 78–84. URL: <http://www.sciencedirect.com/science/article/pii/S1877343518300290>. doi:doi.org/10.1016/j.cosust.2018.10.023. Environmental Change Assessment.
- Jarvis, N. (2007). A review of non-equilibrium water flow and solute transport in soil macropores: Principles, controlling factors and consequences for water quality. *European Journal of Soil Science*, *58*, 523–546. doi:10.1111/j.1365-2389.2007.00915.x.
- 645 Jones, J., & Woodward, C. (2001). Newton-Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems. *Advances in Water Resources*, *24*, 763–774. doi:10.1016/S0309-1708(00)00075-0.

- Jones, J. A., Swanson, F. J., Wemple, B. C., & Snyder, K. U. (2000). Effects of roads on hydrology, geomorphology, and disturbance patches in stream networks. *Conservation Biology*, *14*, 76–85. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1523-1739.2000.99083.x>. doi:10.1046/j.1523-1739.2000.99083.x.
- 650 Jury, W. A., & Horton, R. (2004). *Soil Physics*. (6th ed.). New York: John Wiley & Sons.
- Kanwar, R., A. Rizvi, H., Ahmed, M., & Horton, R. (1990). Measurement of field-saturated hydraulic conductivity by using guelph and velocity permeameters. *American Society of Agricultural Engineers*, *32*. doi:10.13031/2013.31239.
- Kneis, D. (2015). A lightweight framework for rapid development of object-based hydrological model engines. 655 *Environmental Modelling & Software*, *68*, 110 – 121. URL: <http://www.sciencedirect.com/science/article/pii/S1364815215000584>. doi:10.1016/j.envsoft.2015.02.009.
- Kollet, S., & Maxwell, R. (2006). Integrated surface-groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model. *Advances in Water Resources*, *29*, 945–958. doi:10.1016/j.advwatres.2005.08.006. Cited By 321.
- 660 Kraft, P., Vache, K. B., Frede, H.-G., & Breuer, L. (2012). CMF: A hydrological programming language extension for integrated catchment models. *Environmental Modelling and Software*, *26*, 828–830. doi:10.1016/j.envsoft.2010.12.009.
- Kralisch, S., & Krause, P. (2006). Jams - a framework for natural resource model development and application. In *Proceedings of the iEMSs 3rd Biennial Meeting, " Summit on Environmental Modelling and Software*. 665 URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84858643466&partnerID=40&md5=cb2f0342fb5b341a9046a98e22bcae0b> cited By 33.
- Krone-Davis, P., Watson, F., Huertos, M. L., & Starner, K. (2013). Assessing pesticide reduction in constructed wetlands using a tanks-in-series model within a bayesian framework. *Ecological Engineering*, *57*, 342 – 352. doi:10.1016/j.ecoleng.2013.04.053.
- 670 Lacas, J.-G., Voltz, M., Gouy, V., Carluer, N., & Gril, J.-J. (2005). Using grassed strips to limit pesticide transfer to surface water: A review. *Sustainable Agriculture*, (pp. 471–491). doi:10.1007/978-90-481-2666-8\_30.
- Lammoglia, S.-K., Moeys, J., Barriuso, E., Larsbo, M., Marín-Benito, J.-M., Justes, E., Alletto, L., Ubertosi, M., Nicolardot, B., Munier-Jolain, N., & Mamy, L. (2017). Sequential use of the STICS crop model and of the MACRO pesticide fate model to simulate pesticides leaching in cropping systems. 675 *Environmental Science and Pollution Research*, *24*, 6895–6909. doi:10.1007/s11356-016-6842-7.
- Larsbo, M., & Jarvis, N. (2003). *MACRO 5.0: A Model of Water Flow and Solute Transport in Macroporous Soil : Technical Description*. Emergo (Uppsala). Department of Soil Sciences, Swedish University of Agricultural Sciences.

- Lauvernet, C., Helbert, C., Catalogne, C., Carlier, N., & Muñoz-Carpena, R. (2016). Sensitivity analysis and  
680 metamodeling methods for designing buffer strips to protect water from pesticide transfers. In *International  
conference on sensitivity analysis of model output, 30/11/2016-03/12/2016, La Réunion, FRA / Sensitivity  
Analysis of Model Output (SAMO) PROCEEDINGS* (pp. 58–60). Le Tampon: University of Reunion Island.
- Lauvernet, C., & Muñoz-Carpena, R. (2018). Shallow water table effects on water, sediment, and pesticide transport  
in vegetative filter strips – Part 2: model coupling, application, factor importance, and uncertainty. *Hydrology  
685 and Earth System Sciences, 22*, 71–87. doi:10.5194/hess-22-71-2018.
- Leopold, L. B., & Dunne, T. (1978). *Water in environmental planning*. Freeman, W. H.
- Lewis, K. A., Tzilivakis, J., Warner, D. J., & Green, A. (2016). An international database for pesticide  
risk assessments and management. *Human and Ecological Risk Assessment: An International Journal, 22*,  
1050–1064. URL: <https://doi.org/10.1080/10807039.2015.1133242>. doi:10.1080/10807039.2015.1133242.  
690 arXiv:<https://doi.org/10.1080/10807039.2015.1133242>.
- Li, K., De Jong, R., & Boisvert, J. (2001). An exponential root-water-uptake model with water stress compensation.  
*Journal of Hydrology, 252*, 189–204. doi:10.1016/S0022-1694(01)00456-5.
- Li, R.-M., B. Simons, D., & A. Stevens, M. (1975). Nonlinear kinematic wave approximation for water routing.  
*Water Resources Research, 11*, 245–252. doi:10.1029/WR011i002p00245.
- 695 Margoum, C., Gouy, V., Williams, R., & Smith, J. (2001). Le rôle des fossés agricoles dans la dissipation des  
produits phytosanitaires. *Ingénieries - EAT, 1*, 55–65. doi:10.7202/705514ar.
- Miles, J. (1985). The representation of flows to partially penetrating rivers using groundwater flow models. *Journal  
of Hydrology, 82*, 341–355. doi:10.1016/0022-1694(85)90026-5.
- Moussa, R., & Bocquillon, C. (1996). Criteria for the choice of flood-routing methods in natural channels. *Journal  
700 of Hydrology, 186*, 1–30. doi:10.1016/S0022-1694(96)03045-4.
- Muñoz-Carpena, R., Lauvernet, C., & Carlier, N. (2018). Shallow water table effects on water, sediment, and  
pesticide transport in vegetative filter strips – Part 1: nonuniform infiltration and soil water redistribution.  
*Hydrology and Earth System Sciences, 22*, 53–70. doi:10.5194/hess-22-53-2018.
- Muñoz-Carpena, R., Parsons, J. E., & Gilliam, J. (1999). Modeling hydrology and sediment transport in vegetative  
705 filter strips. *Journal of Hydrology, 214*, 111–129. doi:10.1016/S0022-1694(98)00272-8.
- Noilhan, J., & Planton, S. (1989). A simple parameterization of land surface processes for meteorological models.  
*Monthly Weather Review, 117*, 536–549. doi:10.1175/1520-0493(1989)117<0536:ASPOLS>2.0.CO;2.
- Paniconi, C., & Putti, M. (1994). A comparison of Picard and Newton iteration in the numer-  
ical solution of multidimensional variably saturated flow problems. *Water Resources Research, 30*,  
710 3357–3374. URL: [http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.  
0-0028666563&lang=fr&site=eds-live](http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-0028666563&lang=fr&site=eds-live). doi:10.1029/94WR02046.

- Poletika, N., Coody, P., Fox, G., J Sabbagh, G., C Dolder, S., & White, J. (2009). Chlorpyrifos and atrazine removal from runoff by vegetated filter strips: Experiments and predictive modeling. *Journal of Environmental Quality*, 38, 1042–52. doi:10.2134/jeq2008.0404.
- 715 Reichenberger, S., Bach, M., Skitschak, A., & Frede, H.-G. (2007). Mitigation strategies to reduce pesticide inputs into ground- and surface water and their effectiveness; a review. *Science of The Total Environment*, 384, 1–35. doi:10.1016/j.scitotenv.2007.04.046.
- Richards, L., A (1931). Capillary conduction of liquids in porous mediums. *Physics*, 1, 318 – 333. doi:10.1063/1.1745010.
- 720 Ross, P. (2003). Modeling soil water and solute transport - fast, simplified numerical solutions. *Agronomy Journal*, 95, 1352–1361. doi:10.2134/agronj2003.1352.
- Sabbagh, G., Fox, G., Kamanzi, A., Roepke, B., & Tang, J.-Z. (2009). Effectiveness of vegetative filter strips in reducing pesticide loading: Quantifying pesticide trapping efficiency. *Journal of Environmental Quality*, 38, 762–71. doi:10.2134/jeq2008.0266.
- 725 Simpkins, W., Wineland, T., Andress, R., Johnston, D., Caron, G., Isenhardt, T., & Schultz, R. (2002). Hydrogeological constraints on riparian buffers for reduction of diffuse pollution: examples from the Bear Creek watershed in Iowa, USA. *Water Science and Tech.*, 45, 61–68. URL: <http://www.iwaponline.com/wst/04509/wst045090061.htm>.
- Stehle, S., Elsaesser, D., Gregoire, C., Imfeld, G., Niehaus, E., Passeport, E., Payraudeau, S., Schäfer, R. B., 730 Tournebize, J., & Schulz, R. (2011). Pesticide risk mitigation by vegetated treatment systems: A meta-analysis. *Journal of Environmental Quality*, 40, 1068–80. doi:10.2134/jeq2010.0510.
- Taconet, O., Bernard, R., & Vidal-Madjar, D. (1986). Evapotranspiration over an agricultural region using a surface flux/temperature model based on NOAA-AVHRR data. *Journal of Applied Meteorology*, 25, 284–307. doi:10.1175/1520-0450(1986)025<0284:E0AARU>2.0.CO;2.
- 735 Tomer, M. D., Dosskey, M. G., Burkart, M. R., James, D. E., Helmers, M. J., & Eisenhauer, D. E. (2009). Methods to prioritize placement of riparian buffers for improved water quality. *Agroforestry Systems*, 75, 17–25. doi:10.1007/s10457-008-9134-5.
- Varado, N., Braud, I., & Ross, P. (2006a). Development and assessment of an efficient vadose zone module solving the 1d Richards' equation and including root extraction by plants. *Journal of Hydrology*, 323, 258–275. doi:10. 740 1016/j.jhydro1.2005.09.015.
- Varado, N., Braud, I., Ross, P., & Haverkamp, R. (2006b). Assessment of an efficient numerical solution of the 1d Richards' equation on bare soil. *Journal of Hydrology*, 323, 244 – 257. doi:10.1016/j.jhydro1.2005.07.052.
- Villa-Vialaneix, N., Follador, M., Ratto, M., & Leip, A. (2012). A comparison of eight metamodelling techniques for the simulation of N<sub>2</sub>o fluxes and N leaching from corn crops. *Environmental Modelling and Software*, 34, 745 51–66. doi:10.1016/j.envsoft.2011.05.003.

Warren, N., Allan, I., Carter, J., House, W., & Parker, A. (2003). Pesticides and other micro-organic contaminants in freshwater sedimentary environments—a review. *Applied Geochemistry*, 18, 159 – 194. URL: <http://www.sciencedirect.com/science/article/pii/S0883292702001592>. doi:[https://doi.org/10.1016/S0883-2927\(02\)00159-2](https://doi.org/10.1016/S0883-2927(02)00159-2).

750 Weill, S., Mazzia, A., Putti, M., & Paniconi, C. (2011). Coupling water flow and solute transport into a physically-based surface–subsurface hydrological model. *Advances in Water Resources*, 34, 128 – 136. doi:10.1016/j.advwatres.2010.10.001.

Zhang, H. (1996). Anisotropic variation of saturated hydraulic conductivity of a variously grazed salt marsh soil. *Journal of Plant Nutrition and Soil Science*, 159, 129–135. doi:10.1016/j.still.2008.05.004.