



Représentation numérique du paysage par OpenFLUID

M. Rabotin et al.

LISAH - Laboratoire d'étude des Interactions Sol-Agrosystème-Hydrosystème



This document is licensed
under Creative Commons license

Concepts de RNP

Modélisation du fonctionnement du paysage

Représentation **numérique** du paysage

- **éléments du paysage**
(parcelles, routes, fossés, nappes, ...)
- **propriétés** de ces éléments
(géométrie, propriétés physiques, ...)
- **relations/connexions** entre ces éléments (topologiques et hiérarchiques)

+

Modélisation des **processus** en **interaction**

- **dynamiques locales et spatiales** des processus
(transferts, évolutions, décisions, ...)
- **couplage** entre les processus (interactions, rétroactions)



Modélisation du fonctionnement du paysage

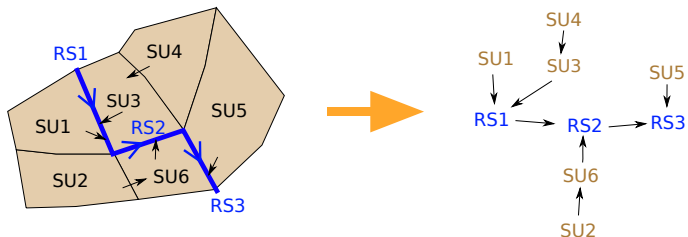
Représentation numérique du paysage

- éléments du paysage
(parcelles, routes, fossés,
nappes, . . .)
- propriétés de ces éléments
(géométrie, propriétés
physiques, . . .)
- relations/connexions entre ces
éléments (topologiques et
hiérarchiques)

Représentation de l'espace sous OpenFLUID

L'espace est représenté par OpenFLUID sous la forme d'un **graphe orienté**

- Les noeuds sont les **unités spatiales** composant l'espace, rangés par **classes d'unités spatiales**
- Les arcs orientés sont les **relations** entre les unités spatiales
- Chaque noeud porte des **attributs propres** à l'unité spatiale qu'il représente

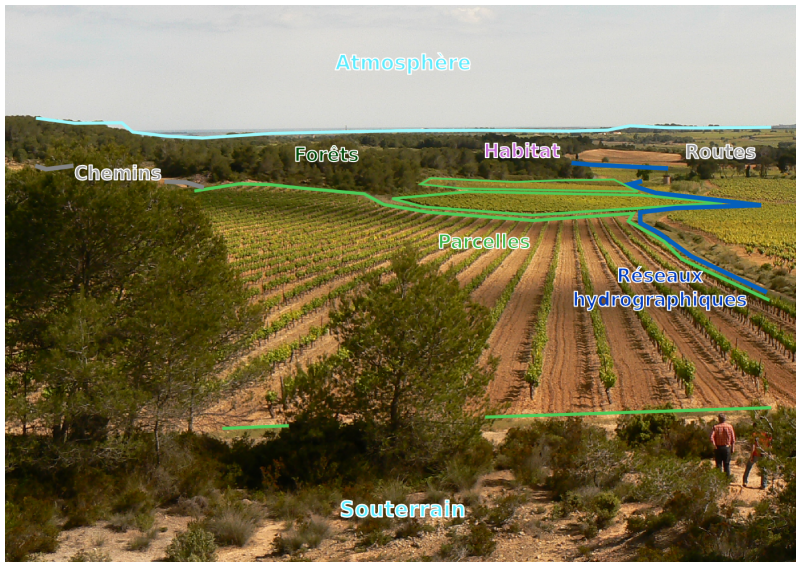


Nombreux **algorithmes** disponibles sur les graphes
(calculs de connexité, parcours, parallélisation, ...)

Du paysage à sa représentation numérique



Du paysage à sa représentation numérique



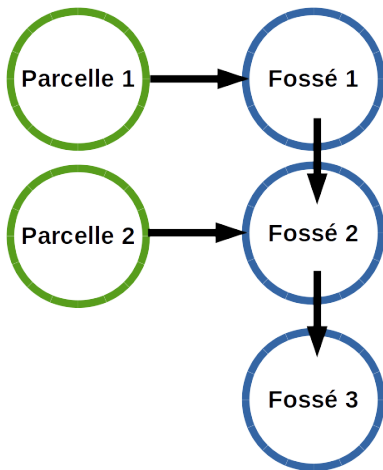
Démarche générale

- Choix des **classes** d'unités
 - classe **Parcelle**
 - classe **Fossé**
- **Instanciation** des classes d'unités
- Calcul des **connexions** entre unités
- Calcul des **attributs** des unités



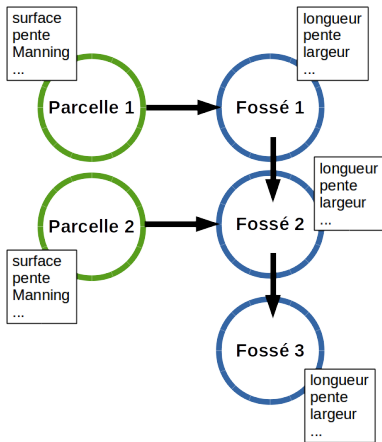
Démarche générale

- Choix des **classes** d'unités
 - classe **Parcelle**
 - classe **Fossé**
- **Instanciation** des classes d'unités
- Calcul des **connexions** entre unités
- Calcul des **attributs** des unités



Démarche générale

- Choix des **classes** d'unités
 - classe **Parcelle**
 - classe **Fossé**
- **Instanciation** des classes d'unités
- Calcul des **connexions** entre unités
- Calcul des **attributs** des unités



Pour construire le graphe

- Approche par **expertise terrain**
- Approche par **automatisation**

Approche par expertise terrain

Exemple de création de paysage

Bassin versant de **St Bauzille de la Sylve**

Objectifs : modéliser le fonctionnement **hydrologique** de surface



Pas de données spatiales

Visite terrain pour **acquérir** les informations

Identification des classes d'unités

Classe d'unités : **Parcelle**



Identification des classes d'unités

Classe d'unités : **Fossé**



Identification des connexions

Ecoulements hydrologiques de surface



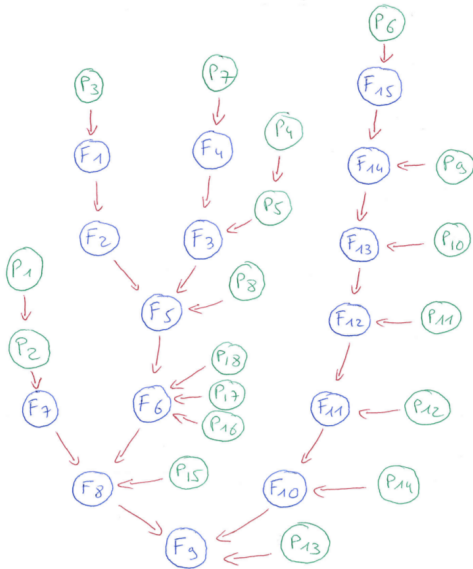
Acquisition des attributs

Attributs **géométriques**

Attributs en lien avec la **thématique**



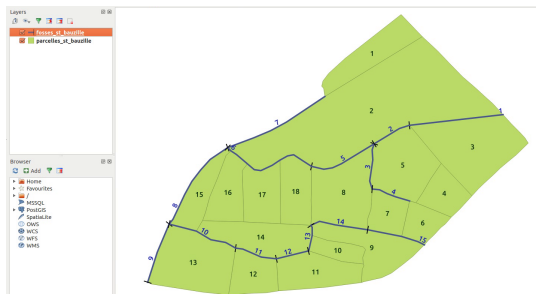
Construction du graphe d'espace



Avec des données spatiales

Acquisition de données spatiales : GPS, cadastre...
Données SIG **support** lors de l'import dans OpenFLUID.

- **Géométrie** utilisée pour calcul surface, longueur...
- Table **attributaire** utilisée pour les **connexions** et les attributs



Avec des données spatiales

Acquisition de données spatiales : GPS, cadastre...

Données SIG support lors de l'import dans OpenFLUID.

- Géométrie utilisée pour calcul surface, longueur...
- Table attributaire utilisée pour les connexions et les attributs

| | altitude | OFLD_ID | Hc | Ks | betaMS | manning | thetaini | thetara | thetasat | topology |
|----|----------|---------|----------|---------------|---------|----------|----------|----------|----------|------------|
| 0 | 82.30000 | 1 | 0.100000 | 0.00000258... | 1.29000 | 0.060000 | 0.280000 | 0.018000 | 0.350000 | parcelle#2 |
| 1 | 79.20000 | 2 | 0.100000 | 0.00000258... | 1.29000 | 0.060000 | 0.320000 | 0.021000 | 0.350000 | fosse#2 |
| 2 | 80.20000 | 3 | 0.090000 | 0.00000732... | 1.27000 | 0.050000 | 0.330000 | 0.021000 | 0.350000 | fosse#1 |
| 3 | 79.30000 | 4 | 0.090000 | 0.00000732... | 1.31000 | 0.050000 | 0.320000 | 0.019000 | 0.350000 | parcelle#5 |
| 5 | 78.50000 | 5 | 0.110000 | 0.00000258... | 1.30000 | 0.061000 | 0.280000 | 0.021000 | 0.360000 | fosse#3 |
| 17 | 79.40000 | 6 | 0.100000 | 0.00000258... | 1.30000 | 0.050000 | 0.290000 | 0.014000 | 0.360000 | fosse#15 |
| 4 | 79.00000 | 7 | 0.110000 | 0.00000258... | 1.31000 | 0.050000 | 0.320000 | 0.018000 | 0.350000 | fosse#4 |
| 6 | 77.50000 | 8 | 0.090000 | 0.00000225... | 1.30000 | 0.060000 | 0.300000 | 0.020000 | 0.360000 | fosse#5 |
| 11 | 77.60000 | 9 | 0.120000 | 0.00000258... | 1.29000 | 0.050000 | 0.310000 | 0.017000 | 0.360000 | fosse#14 |
| 13 | 78.40000 | 10 | 0.100000 | 0.00000225... | 1.31000 | 0.050000 | 0.280000 | 0.020000 | 0.370000 | fosse#13 |
| 14 | 79.80000 | 11 | 0.100000 | 0.00000732... | 1.28000 | 0.060000 | 0.270000 | 0.020000 | 0.350000 | fosse#12 |
| 15 | 79.20000 | 12 | 0.110000 | 0.00000732... | 1.28000 | 0.060000 | 0.310000 | 0.015000 | 0.350000 | fosse#11 |
| 16 | 77.00000 | 13 | 0.090000 | 0.00000258... | 1.30000 | 0.050000 | 0.280000 | 0.014000 | 0.360000 | fosse#9 |
| 12 | 74.10000 | 14 | 0.080000 | 0.00000225... | 1.31000 | 0.050000 | 0.300000 | 0.018000 | 0.370000 | fosse#10 |
| 10 | 73.60000 | 15 | 0.110000 | 0.00000258... | 1.29000 | 0.060000 | 0.290000 | 0.016000 | 0.360000 | fosse#8 |
| 9 | 73.70000 | 16 | 0.100000 | 0.00000732... | 1.27000 | 0.060000 | 0.280000 | 0.014000 | 0.340000 | fosse#6 |
| 8 | 76.00000 | 17 | 0.090000 | 0.00000732... | 1.27000 | 0.050000 | 0.300000 | 0.160000 | 0.340000 | fosse#6 |
| 7 | 76.90000 | 18 | 0.120000 | 0.00000225... | 1.30000 | 0.050000 | 0.300000 | 0.018000 | 0.330000 | fosse#6 |

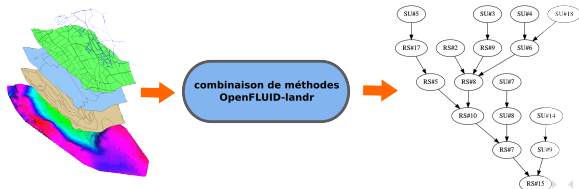
Principes

Mettre à disposition des utilisateurs des **briques élémentaires** de traitements géomatiques pour construire des traitements **spécifiques**.

Caractéristiques :

- S'appuie sur des bibliothèques conformes aux **standards** de l'OGC (OGR/GDAL, GEOS),
- Utilise les formats de données cartographiques **existants**,
- N'est pas ciblé sur une thématique particulière.

Des données **spatiales** au **graphe**



Fonctionnalités et structures proposées

Des **objets**

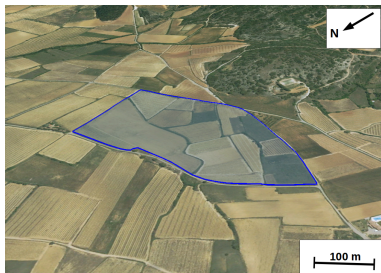
- permettant l'intégration des données spatiales vectorielles **polygones** et **lignes**,
- proposant des structures de stockage **topologique**,
- pouvant être liés à des données **raster**.

Des **méthodes** pour

- l'**intégration** des données vecteur et raster,
- la **discrétisation** de l'espace en **unités spatiales** selon les besoins du modélisateur,
- la vérification et la modification de la **géométrie** des unités spatiales,
- la construction de la **connectivité orientée** entre ces unités spatiales,
- le **paramétrage** de ces unités spatiales.

Exemple thématique : Création du paysage

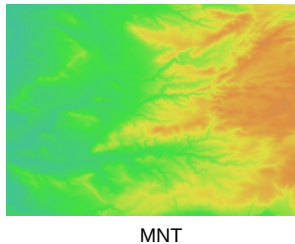
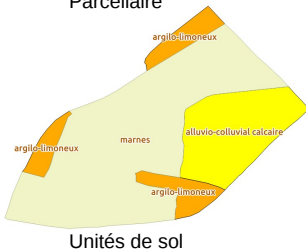
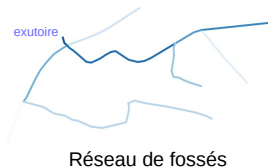
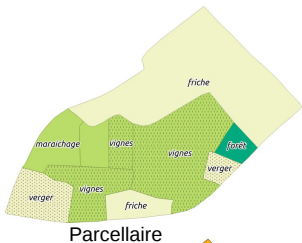
Bassin versant de St Bauzille de la Sylve



Représentation du paysage pour le modèle hydrologique utilisé :

- discrétisation de l'espace en unités spatiales **surfaci**ques et **liné**aires,
- construction de la **connectivité** des unités spatiales,

Données spatiales disponibles



Discrétisation du paysage en unités spatiales

Intersections **successives** des données vectorielles



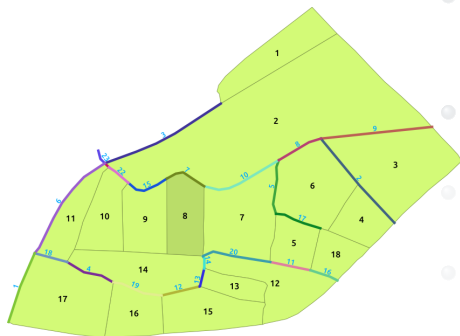
Création des unités spatiales **surfacentriques** et **linéaires**
Nettoyage si nécessaire des **géométries** non conformes
avec vérification de la validité **topologique**.

Discrétisation par OpenFLUID-landr - pseudo code

```
//Definition d'un objet MyLandscape :  
//qui peut contenir des objets PolygonGraph et LineStringGraph  
MyLandscape();  
  
//integration des couches Parcelles, Sols et Fosses  
MyLandscape.addLayer(Parcelles)  
MyLandscape.addLayer(Sols)  
MyLandscape.addLayer(Fosses)  
  
// intersections successives  
MyLandscape.segment()  
  
// nettoyage des polygones non conformes :  
// par des criteres de surface  
MyLandscape.mergePolygonEntities(  
    getPolygonEntitiesByMinArea(Threshold))  
  
// et des criteres de forme  
MyLandscape.mergePolygonEntities(  
    getPolygonEntitiesByCompactness(Threshold))
```

Modularité du calcul de connectivité

Exemple pour l'entité surfacique **8**



Différentes règles de connectivité possibles

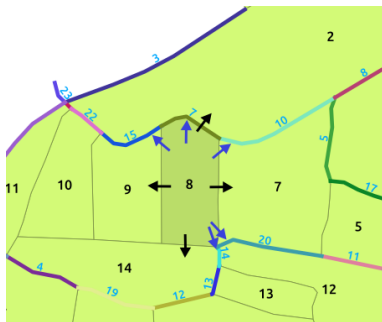
- voisins polygones + voisins lignes en contact par au moins un **sommet**,
- voisins polygones + voisins lignes avec contact **latéral**,
- prise en compte d'éléments linéaires **bloquants**,
- orientation par la **pente** calculée à partir d'un MNT,

Modularité du **nombre** et de l'**ordre** des méthodes.

Modularité du calcul de connectivité

Exemple pour l'entité surfacique **8**

Différentes règles de connectivité possibles

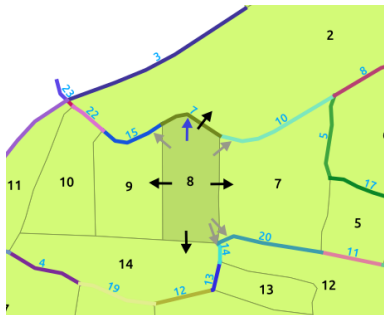


- voisins polygones + voisins lignes en contact par au moins un **sommet**,
- voisins polygones + voisins lignes avec contact **latéral**,
- prise en compte d'éléments linéaires **bloquants**,
- orientation par la **pente** calculée à partir d'un MNT,

Modularité du **nombre** et de **l'ordre** des méthodes.

Modularité du calcul de connectivité

Exemple pour l'entité
surfactive **8**



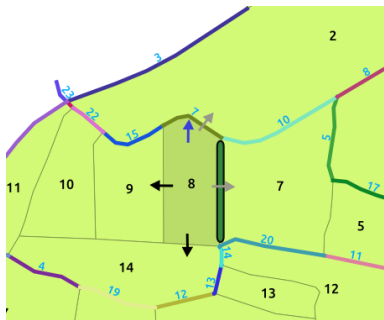
Différentes règles de
connectivité possibles

- voisins polygones + voisins lignes en contact par au moins un **sommet**,
- voisins polygones + voisins lignes avec contact **latéral**,
- prise en compte d'éléments linéaires **bloquants**,
- orientation par la **pente** calculée à partir d'un MNT,

Modularité du **nombre** et de l'**ordre** des méthodes.

Modularité du calcul de connectivité

Exemple pour l'entité
surfactive **8**



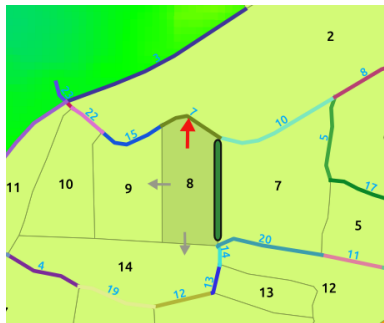
Différentes règles de
connectivité possibles

- voisins polygones + voisins lignes en contact par au moins un **sommet**,
- voisins polygones + voisins lignes avec contact **latéral**,
- prise en compte d'éléments linéaires **bloquants**,
- orientation par la **pente** calculée à partir d'un MNT,

Modularité du **nombre** et de l'**ordre** des méthodes.

Modularité du calcul de connectivité

Exemple pour l'entité
surfactive **8**



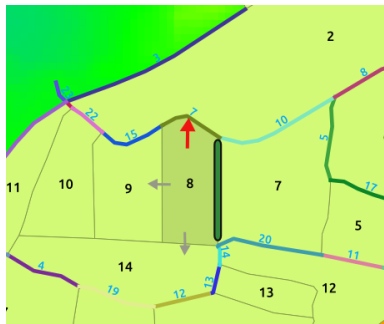
Différentes règles de
connectivité possibles

- voisins polygones + voisins lignes en contact par au moins un **sommet**,
- voisins polygones + voisins lignes avec contact **latéral**,
- prise en compte d'éléments linéaires **bloquants**,
- orientation par la **pente** calculée à partir d'un MNT,

Modularité du **nombre** et de l'**ordre** des méthodes.

Modularité du calcul de connectivité

Exemple pour l'entité surfacique **8**



Différentes règles de connectivité possibles

- voisins polygones + voisins lignes en contact par au moins un **sommet**,
- voisins polygones + voisins lignes avec contact **latéral**,
- prise en compte d'éléments linéaires **bloquants**,
- orientation par la **pente** calculée à partir d'un MNT,

Modularité du **nombre** et de l'**ordre** des méthodes.

Connectivité par OpenFLUID-landr - pseudo code

```
// Identification des voisins polygones
MyLandscape.computePolygonNeighbour

// Identification des voisins lignes
MyLandscape.computeLineStringNeighbour(
  openfluid::landr::LandRTools::Relationship, Threshold)

// Plusieurs possibilites de voisinage polygone - ligne
//Relationship :contains, touches, intersects, ...

// Prise en compte des barrieres lineaires
MyLandscape.computeNeighbourWithBarriers(
  LineStringGraph Barriers)

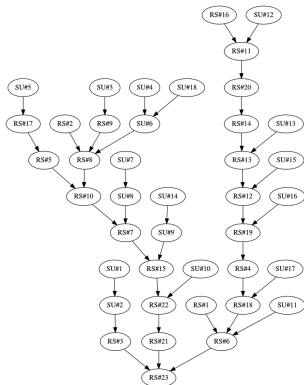
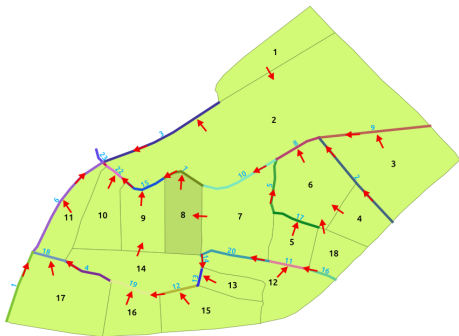
// Integration du MNT
MyLandscape.addLayer(MNT)

// Identification du voisin connecte
// par la methode de plus grande pente
MyLandscape.computeNeighbourWithSlope()
```

Exemple de graphe de connectivité obtenu

Calcul de connectivité sur toutes les unités spatiales

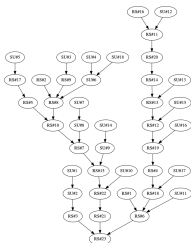
```
MyLandscape.computeLineStringNeighbour(
    openfluid::landr::LandRTools::Relationship Touches, Threshold)
```



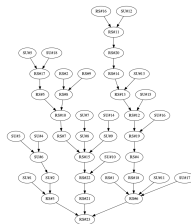
Impact des règles de connectivité sur le graphe

Avec une **même discrétisation** du paysage

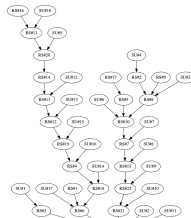
computeLineStringNeighbour(Relationship **Touches**)



computeLineStringNeighbour(Relationship **Intersects**)



computeLineStringNeighbour(Relationship **HigherProcessOrder**)



Exemple d'un outil en production

Objectifs:

- Création de domaines spatiaux pour le modèle hydrologique **MHYDAS**
- Utilisation au travers d'une interface graphique (extension à OpenFLUID-Builder)

Démarche:

- identification des **spécificités de la représentation du paysage** pour le modèle MHYDAS,
- développement de **traitements géomatiques spécifiques** basés sur une **combinaison** de méthodes OpenFLUID-landr,
- création d'une **interface graphique** permettant le paramétrage et l'enchaînement de ces traitements.

Exemple d'un outil en production

Objectifs:

- Création de domaines spatiaux pour le modèle hydrologique **MHYDAS**
- Utilisation au travers d'une interface graphique (extension à OpenFLUID-Builder)

Démarche:

- identification des **spécificités de la représentation du paysage** pour le modèle MHYDAS,
- développement de **traitements géomatiques spécifiques** basés sur une **combinaison** de méthodes OpenFLUID-landr,
- création d'une **interface graphique** permettant le paramétrage et l'enchaînement de ces traitements.

Geo-MHYDAS 15.xx

Utilisation de l'extension **Geo-MHYDAS**

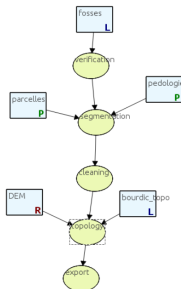
Vidéo de démonstration :

http://youtu.be/Ly_BkuJwkdk

Perspectives

- Intégration des objets vectoriels de type **points**,
- Faciliter les approches à dominante **raster**,
- Travail sur une nouvelle **ergonomie**.

Organigramme d'actions



Références



J.C. Fabre, X. Louchart, R. Moussa, C. Dagès, F. Colin, M. Rabotin, D. Raclot, P. Lagacherie, and Voltz M.

OpenFLUID: a software environment for modelling fluxes in landscapes.

In *LANDMOD2010*, INRA, CIRAD, page 13pp, Montpellier, France, 2010. Quae.



J.-C. Fabre, M. Rabotin, D. Crevoisier, A. Libres, C. Dagès, R. Moussa, Ph. Lagacherie, D. Raclot, and M. Voltz.

OpenFLUID: an open-source software environment for modelling fluxes in landscapes.

In *Vol. 15, EGU2013-8821-1, EGU General Assembly 2013, Vienna (Austria)*, 7-12 avril 2013.



M. Rabotin, J.-C. Fabre, A. Libres, P. Lagacherie, D. Crevoisier, and R. Moussa.

Using graph approach for managing connectivity in integrative landscape modelling.

In *Vol. 15, EGU2013-8851, EGU General Assembly 2013, Vienna (Austria)*, 7-12 avril 2013.