

OpenFLUID

Software Environment
for Modelling Fluxes in Landscapes

*Manuel de l'utilisateur
OpenFLUID-Engine
version 1.3*

Table des matières

Avant-propos	3
Conventions typographiques	3
Lexique	3
1 Présentation	5
1.1 Eléments conceptuels	5
1.1.1 Discrétisation de l'espace	5
1.1.2 Modélisation des processus de transfert	6
1.2 Eléments opérationnels	6
1.2.1 Le moteur de simulation : Engine	6
1.2.2 L'interface utilisateur : Builder	7
1.2.3 Le site web communautaire : Web	7
2 Simuler avec OpenFLUID-engine	9
2.1 Préparer le jeu de données	9
2.1.1 Définition du modèle de flux (<code>model.xml</code>)	9
2.1.2 Configuration de la simulation(<code>run.xml</code>)	10
2.1.3 Unités (<code>SUdefs.dat</code> , <code>RSdefs.dat</code> , <code>GUdefs.dat</code>)	10
2.1.4 Propriétés et conditions initiales distribuées (<code>*.ddata.xml</code>)	12
2.1.5 Evènements discrets distribués (<code>*.events.xml</code>)	12
2.1.6 Gestion des sorties (<code>output.xml</code>)	13
2.2 Lancer la simulation	14
2.3 Exploiter les résultats	15
3 Annexes	17
3.1 Lexique	17
3.2 Options de la ligne de commande	17
3.3 Généralités sur les formats de fichier OpenFLUID-engine	17
3.3.1 Format texte-colonnes	17
3.3.2 Format XML	18
3.3.3 Format XML-mixte	18
3.4 Formatage des dates à l'aide de la norme ANSI <code>strftime()</code>	19

Avant-propos

Ce manuel d'utilisation décrit la mise en oeuvre d'une session de modélisation avec l'environnement logiciel OpenFLUID-engine, notamment au travers d'exemples concrets. Ce manuel n'a pas pour but d'expliquer en détail la démarche et la connaissance scientifique exprimée dans OpenFLUID-engine.

Si vous souhaitez en savoir plus, nous vous invitons à lire les documentations scientifiques liées à OpenFLUID-engine et à prendre connaissance de la production bibliographique associée.

Ce manuel s'adresse aux chercheurs, ingénieurs, étudiants, doctorants qui souhaitent utiliser OpenFLUID-engine dans le cadre de leurs travaux.

Convention typographiques

Les rubriques "à noter" sont signalées comme suit :



Rubrique "à noter" précédée du sigle correspondant

Les rubriques "code" sont signalées comme suit :



Rubrique code, avec le sigle correspondant et une police à chasse fixe

Les rubriques "attention" sont signalées comme suit :



Rubrique attention, avec le sigle correspondant

Lexique

Conditions initiales distribuées : Valeurs attachées à chacune des unités homogènes constituant la représentation du paysage donnant l'état initial du système à modéliser. Elles ne peuvent être modifiées au long de la simulation.

Élément du paysage : Éléments physiques du paysage, à partir desquels sont bâties les unités homogènes représentatives.

Espace : Ensemble d'unités représentatives homogènes constituant le paysage sur lequel on exécute une simulation.

Évènement distribué : Un évènement distribué est un évènement qui survient à un instant donné sur une unité homogène constituant le paysage. Chaque évènement est caractérisé par des informations. Le nombre d'évènement, ainsi que la quantité d'informations qui les

caractérisent, ne sont pas limités. Ils ne peuvent être modifiés au long de la simulation, mais il est possible d'en générer.

Fonction de simulation : Une fonction de simulation est un code de calcul, mobilisé par le noyau du moteur de calcul, qui s'exécute à chaque pas de temps en utilisant les données mises à disposition par le noyau et les autres fonctions de simulation. Les fonctions de simulation sont les éléments de base des modèles.

ID : Identifiant unique d'une unité représentative homogène.

Modèle : Un modèle, au sens de la plate-forme OpenFLUID, est une liste ordonnée et cohérente de fonctions de simulations qui sera mobilisée par le noyau du moteur de calcul. La notion de cohérence est limitée à la disponibilité des données injectées dans le moteur de calcul, et aux échanges des données entre les fonctions de simulation.

Ordre de traitement : Ordre selon lequel une unité va être traitée (à chaque pas de temps).

Paramètre de fonction : Paramètre d'une fonction de simulation, qui lui est propre, et qui définit une valeur d'un élément de calcul ou d'équation.

Pas de temps : Le pas de temps de simulation est le pas de temps auquel se font les échanges entre les différentes fonctions de simulation.

Propriétés distribuées : Propriétés attachées à chacune des unités homogènes constituant la représentation du paysage. Elles peuvent être géométriques (longueur, largeur, surface, ...), qualitatives (Ks, ThetaR, ...), etc... Les propriétés distribuées peuvent être modifiées au long de la simulation (actuellement, sans garder trace des valeurs précédentes)

Simulation : Une simulation est l'application d'un modèle à un espace sur une période de temps, à un pas de temps donné.

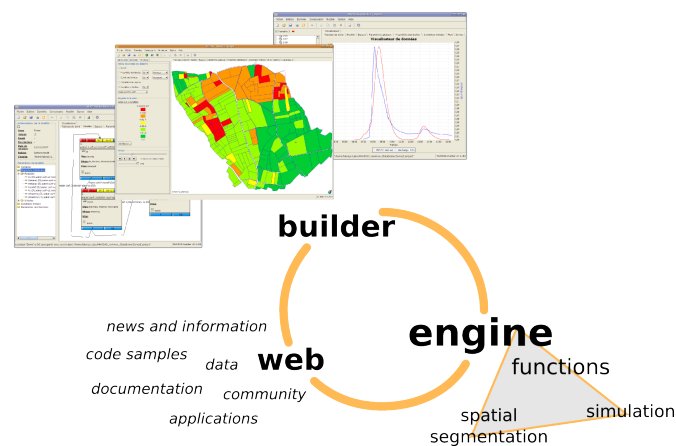
Unité homogène : Unité considérée comme représentative d'un espace et homogène pour le modèle utilisé, et qualifiée par des propriétés et conditions initiales distribuées. Actuellement, 3 types de RHU sont pris en compte : les SU (Surface Units), les RS (Reach Segement), les GU (Groundwater Units).

Variable : Une variable de simulation représente un transfert quantitatif entre deux unités, et pour laquelle une valeur est produite à chaque pas de temps.

Chapitre 1

Présentation

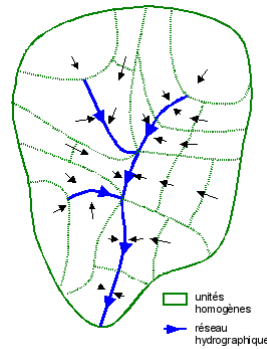
OpenFLUID– Software environment for Modelling Fluxes in Landscapes - est une plateforme ouverte de modélisation des flux de matière dans le paysage, dans l'espace et dans le temps. Elle est constituée de 3 grands composants logiciels : le moteur de simulation OpenFLUID-Engine, l'interface utilisateur OpenFLUID-Builder, et le site web communautaire OpenFLUID-Web.



1.1 Eléments conceptuels

1.1.1 Discrétisation de l'espace

OpenFLUID est basée sur une discrétisation du paysage en "unités homogènes". Actuellement, 3 classes d'unités homogènes sont prises en compte : les unités de surface ou "surface units" (SU), les unités de nappes ou "groundwater units" (GU), et le réseau hydrographique ou "reach segments" (RS). Ces unités homogènes sont déterminées en tenant compte des hétérogénéités du paysage, et des contraintes numériques liés à la représentation de processus à simuler. Elles sont reliées entre-elles par une topologie orientée. L'ensemble des unités homogènes constitue un graphe connexe orienté représentant le paysage.



1.1.2 Modélisation des processus de transfert

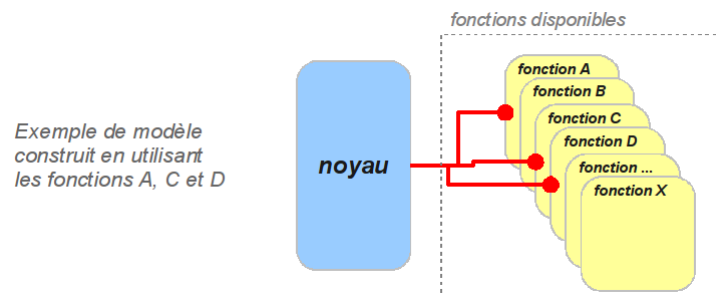
La modélisation des processus de transfert permet la simulation des flux de matière sur les différentes unités homogènes constituant le paysage. Ils s'appuient sur la topologie orientée reliant les unités, ainsi que sur leurs propriétés et conditions initiales et aux limites. OpenFLUID propose une structure permettant d'intégrer et de coupler les processus à modéliser sous la forme de fonctions de simulation. Ces fonctions sont chaînées et paramétrées pour constituer un modèle.

Un ensemble de fonctions de simulation permettant de modéliser certains flux de matière dans le paysage sont disponibles.

1.2 Eléments opérationnels

1.2.1 Le moteur de simulation : Engine

Le moteur de simulation de la plateforme est constitué d'un noyau et de fonctions de simulation. Le noyau a pour tâches principales de structurer, gérer et contrôler les données, d'exécuter et monitorer l'exécution de la simulation, de veiller à la cohérence des modèles exécutés, de prendre en charge les entrées-sorties. Les fonctions de simulation portent les codes de calcul. Elles sont mobilisées par le noyau pour réaliser le modèle à construire (au sens OpenFLUID), et exécuter la simulation. Une fonction de simulation peut-être un simple code rudimentaire, un modèle complet, ou bien un couplage fort de modèles.



Afin de pouvoir développer ses propres fonctions de simulation, et ensuite les utiliser pour construire un modèle, un Kit de Développement Source (Source Kit Développement ou SDK) a été mis en place. Il est constitué :

- d'un environnement de développement (IDE) sous Eclipse, intégrant un assistant de création de fonctions de simulation
- d'une API pour développer la fonction de simulation et bénéficier de toutes les fonctionnalités du moteur de simulation
- d'un ensemble de documentation pour les développeurs

- d'un site web communautaire où l'on peut trouver des exemples de codes, des conseils, des trucs et astuces, des liens vers des docs externes, ...

1.2.2 L'interface utilisateur : Builder

TODO

1.2.3 Le site web communautaire : Web

TODO

Chapitre 2

Simuler avec OpenFLUID-engine

2.1 Préparer le jeu de données

Un jeu de données de simulation doit être composé des fichiers normalisés suivants :

- `model.xml` : définition du modèle
- `run.xml` : configuration de la simulation
- `SUdefs.dat`, `RSdefs.dat`, `GUdefs.dat` : géométrie-topologie des objets spatiaux
- `*.ddata.xml` : propriétés et conditions initiales distribuées
- `*.events.xml` : évènements discrets distribués
- `output.xml` : définition des sorties

Par défaut, le jeu de données en entrée doit se trouver dans un répertoire `.openfluid/engine/OPENFLUID.IN` se situant dans le répertoire-maison (home directory) de l'utilisateur. De la même manière, les résultats de simulation se trouveront, en fin de run, dans un répertoire `.openfluid/engine/OPENFLUID.OUT` se situant dans le répertoire-maison de l'utilisateur. Il est possible de modifier ce comportement par défaut via les options de la ligne de commande (voir en annexe).

2.1.1 Définition du modèle de flux (`model.xml`)

Le fichier de définition du modèle de flux `model.xml` décrit la schéma du simulation qui sera mis en oeuvre. On dit qu'il définit un modèle OpenFLUID-engine. Il décrit :

- les paramètres globaux de calcul et de fonctionnement
- l'enchaînement des fonctions de simulation constituant le modèle
- les paramètres globaux propres à chacune des fonctions de simulation

Ce fichier est au format XML. Il contient une section unique `<model>`. Cette section décrit l'ordonnement des fonctions de simulation qui vont être utilisées. Chaque fonction sera décrite par une balise `<function>` contenant un attribut obligatoire : `fileID`. Cet attribut définit l'identifiant du fichier de la fonction (nom du fichier binaire sans l'extension). S'il y a lieu, chaque fonction de simulation peut définir ses paramètres globaux particuliers au travers d'une ou plusieurs sous-balises `<param>` contenant deux attributs : `name` contenant le nom du paramètre, `value` contenant sa valeur.



l'ordre de description des fonctions utilisées est important : elles seront chaînées ainsi lors de l'exécution de la simulation.



```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <model>
    <function fileID="mseytoux" >
      <param name="resstep" value="0.000005" />
    </function>
    <function fileID="hayamisu">
      <param name="maxsteps" value="100" />
      <param name="meancel" value="0.045" />
      <param name="meansigma" value="500" />
    </function>
    <function fileID="hayamirs">
      <param name="maxsteps" value="100" />
      <param name="meancel" value="0.8" />
      <param name="meansigma" value="500" />
    </function>
  </model>
</openfluid>
```

2.1.2 Configuration de la simulation(run.xml)

La configuration de la simulation consiste principalement à définir la période de simulation (date de début, date de fin) ainsi que le pas de temps d'échange (δt) entre les fonctions de simulation. Les dates de début et de fin précisant la période sont données à la seconde près sous la forme "AAAA-MM-JJ hh :mm :ss". Si la date de fin demandée ne respecte pas la formule $fin = dbut + k * \delta t$ (avec k entier représentant le nombre de pas de temps), la date de fin est automatiquement recalculée pour englober la date de fin demandée. Il est également possible dans ce fichier de forcer la valeur de l'identifiant de simulation généré automatiquement.

Ce fichier est au format XML. Il contient une section unique `<run>`. Dans cette section `run`, on trouve 2 balises obligatoires `<deltat>` et `<period>`, et une balise facultative `<simid>`. La balise `<deltat>` définit le pas de temps d'échange. La balise `<period>` définit la période de simulation au travers de ses attributs obligatoires `begin` et `end`. La balise `<simid>` permet de forcer la valeur de l'identifiant de simulation.



```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <run>
    <deltat>60</deltat>
    <period begin="1997-06-05 04:00:00" end="1997-06-05 16:23:00" />
  </run>
</openfluid>
```

2.1.3 Unités (SUdefs.dat, RSdefs.dat, GUdefs.dat)

Les fichiers de description des unités définissent l'ensemble des unités représentant le domaine spatial de simulation. Ils décrivent leur géométrie, leur topologie (connectivité), ainsi que l'ordre dans lequel elles devront être traitées par le modèle de flux. Ces fichiers sont au format texte-colonnes. Leur format est fixé et chaque colonne a une signification précise.

Ces fichiers sont au nombre de 3 et chacun définit un type d'objet. Si un de ces fichiers est absent le modèle considèrera qu'il n'a pas à traiter les objets de ce type là.

Le fichier `SUdefs.dat` décrit la surface sous la forme d'un ensemble d'unités de surface (SU). Il est composé de 8 colonnes exactement.

Fichier de description des unités de surface SÜdefs.dat*Type : texte colonnes*

Nom	Type	Description
ID	entier	Identifiant unique de l'objet
Area	réel	Aire (m ²)
Slope	réel	Pente (m/m)
FlowCode	caract.	Type de l'objet avec lequel se font les échanges de surface : R pour un bief, S pour une autre SU
FlowID	entier	ID de l'objet avec lequel se font les échanges de surface
FlowDistance	réel	Distance entre centroïdes des objets pour les échanges de surface(m)
ExchangeGU	entier	ID de la GU avec lequel se font les échanges surface-souterrain
ProcessOrd	entier	ordre de traitement de l'objet



```
% définition des unités de surfaces (SU)
%ID AREA SLOPE FCODE FID FDIST EXCHGU PCSSORD
1 4813.344 0.06265 R 330 14.366 2 1
2 293.982 0.27129 R 1 18.468 18 2
3 1038.097 0.16391 S 2 50.15 18 1
4 498.407 0.14065 R 113 4.906 10 3
5 1610.964 0.10073 S 10 23.031 24 1
6 1282.989 0.05443 S 4 30.497 10 2
7 2212.718 0.35908 R 112 15 18 1
8 3315.03 0.02693 S 6 35 10 2
9 1806.479 0.11358 S 169 15.231 18 3
10 2153.719 0.21005 S 9 13.901 18 2
11 4471.305 0.00487 R 350 10 18 4
12 201.064 0.24028 R 7 13.484 18 1
13 2043.269 0.19111 S 11 28.779 18 3
14 1672.532 0.10392 S 13 21.074 18 2
15 1625.557 0.0001 S 14 10 18 1
16 3191.879 0.05893 R 138 33.599 18 1
17 2672.699 0.17241 R 139 15.834 24 1
```

Le fichier **RSdefs.dat** décrit le réseau hydrographique au travers d'un ensemble de tronçons de biefs formant une arborescence. Il est composé de 10 colonnes exactement.

Fichier de description des segments de bief RSdefs.dat*Type : texte colonnes*

Nom	Type	Description
ID	entier	Identifiant unique de l'objet
UpNode	entier	Identifiant du noeud haut
DownNode	entier	Identifiant du noeud bas
LowRS	entier	Identifiant du RS aval
Length	réel	Longueur (m)
Width	réel	Largeur (m)
Height	réel	Hauteur (m)
Slope	réel	Pente (m/m)
ExchangeGU	entier	ID de la GU avec lequel se font les échanges bief-souterrain
ProcessOrd	entier	ordre de traitement de l'objet

Le fichier **GUdefs.dat** décrit le souterrain sous la forme d'un ensemble d'unités souterraines ou aquifères (GU). Il est composé de 4 colonnes exactement.

Fichier de description des unités souterraines GUdefs.dat

Type : texte colonnes

Nom	Type	Description
ID	entier	Identifiant unique de l'objet
SubstLevel	réel	Altitude absolue du substratum (m)
ExchangeGU	entier	ID de la GU avec lequel se font les échanges souterrain-souterrain
ProcessOrd	entier	ordre de traitement de l'objet



Lorsque le lien topologique requis n'existe pas, l'identifiant de l'objet lié doit être -1. Ceci s'applique notamment au champ FlowID du fichier SUdefs.dat, aux champs LowRS et ExchangeGU du fichier RSdefs.dat, ainsi qu'au champ ExchangeGU du fichier GUdefs.dat.

2.1.4 Propriétés et conditions initiales distribuées (*.ddata.xml)

Les fichiers des propriétés distribuées et des conditions initiales des unités homogènes définissent les propriétés et conditions initiales propres à chacune des unités, et doivent avoir pour extension `.ddata.xml`. Il peut y avoir autant de fichiers `.ddata.xml` que souhaité. Les propriétés ou conditions initiales définies pour des unités qui n'existent pas sont ignorées.

Les fichiers `.ddata.xml` sont au format XML-mixte décrit en annexe. Ils doivent comporter une section `distridata`, avec un attribut `unitclass` précisant la classe d'unités à laquelle s'appliquent les données, ainsi qu'un attribut `datacat` précisant la catégorie de données (`ini` pour des conditions initiales, `param` pour des propriétés). Dans la section `distridata`, doit se trouver une section `columns` avec un attribut `order` qui permet de définir l'ordre dans lequel se trouvent les colonnes de données, séparées par des " ; ". La section `data` contient les données en colonnes, la première colonne étant obligatoirement réservée à l'identifiant de l'unité.



```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <distridata unitclass="SU" datacat="param">
    <columns order="ks;hc;betaMS;thetares;thetasat;nmaning" />
    <data>

1 0.000001 0.1 1.3 0.02 0.36 0.05
2 0.000001 0.1 1.3 0.02 0.36 0.05
3 0.000001 0.1 1.3 0.02 0.36 0.05
4 0.000001 0.1 1.3 0.02 0.36 0.05
5 0.000001 0.1 1.3 0.02 0.36 0.05
6 0.000001 0.1 1.3 0.02 0.36 0.05
7 0.000001 0.1 1.3 0.02 0.36 0.05
8 0.000001 0.1 1.3 0.02 0.36 0.05
9 0.000001 0.1 1.3 0.02 0.36 0.05
10 0.000001 0.1 1.3 0.02 0.36 0.05
11 0.000001 0.1 1.3 0.02 0.36 0.05
12 0.000001 0.1 1.3 0.02 0.36 0.05
13 0.000001 0.1 1.3 0.02 0.36 0.05
14 0.000001 0.1 1.3 0.02 0.36 0.05
15 0.000001 0.1 1.3 0.02 0.36 0.05

    </data>
  </distridata>
</openfluid>
```

2.1.5 Evènements discrets distribués (*.events.xml)

Les fichiers des évènements discrets distribués contiennent les calendriers d'évènements qui surviennent à moment donné sur une unité donnée, ainsi que leur description. Ils doivent avoir

pour extension `.events.xml`. Il peut y avoir autant de fichiers `.events.xml` que souhaité.



```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <calendar>

    <event unitclass="SU" unitid="127" date="2008-02-26 09:51">
      <info key="depth" value="0.15" />
      <info key="followslope" value="1" />
      <description>this event is a tillage... blablabla.</description>
    </event>

    <event unitclass="SU" unitid="7" date="2008-02-22 09:00">
      <info key="product" value="roundup" />
      <info key="molecule" value="glyphosate" />
      <info key="qty" value="0.92" />
    </event>

    <event unitclass="RS" unitid="372" date="2008-05-12 00:00">
      <info key="molecule" value="diuron" />
      <info key="qty" value="0.12" />
    </event>

  </calendar>
</openfluid>
```

2.1.6 Gestion des sorties (output.xml)

La gestion des sorties se fait au travers du fichier `output.xml`. Ce fichier de définition des sorties est au format XML. Il permet de définir l'ensemble des résultats qui seront extraits en fin de simulation sous la forme de fichiers au format texte-colonnes contenant pour chaque pas de temps de simulation un ensemble `[date;variable1;variable2;;variablen]`. La date est considérée dans son ensemble comme une seule et même colonne. Ces fichiers contenant les résultats de simulation peuvent être générés selon 2 modes : le mode "auto" et le mode "custom". A l'heure de la rédaction de ces lignes, seul le mode "auto" est fonctionnel.

Le fichier de définition des sorties contient une section définie par la balise `<autooutfiles>`. Cette balise peut contenir les attributs suivants

- `colsep` qui définit le séparateur entre les colonnes. Par défaut si l'attribut n'est pas présent, c'est le caractère tabulation qui est le caractère séparateur.
- `dtformat` qui définit le format de la date. Le format peut être défini en utilisant la nomenclature ANSI `strftime` (cf annexes), ou bien en utilisant les formats prédéfinis `6cols` ou `iso`. Par défaut si l'attribut n'est pas présent, c'est le format `iso` qui sera utilisé.
- `commentchar` qui définit le caractère de mise en commentaire. Par défaut si l'attribut n'est pas présent, c'est le caractère `%` (pourcent) qui est le caractère de mise commentaire.

Dans la section `<autooutfiles>`, il peut être défini autant de catégories de sorties que souhaité à l'aide des balises `<SUout>` pour extraire les données simulées sur les les unités de surface, `<RSout>` pour les données simulées sur les tronçons de biefs, `<GUout>` pour les données simulées sur les unités souterraines. Chacune de ces balises va donner lieu à la génération d'une série de fichiers contenant les résultats de simulation et qui seront nommées selon la nomenclature `ObjetID_Suffixe.dat`. Ces balises peuvent comporter 3 attributs :

- `filesuffix` qui viendra compléter le nom de fichier lors de la génération
- `selection` qui définit la liste des identifiants des objets dont on veut les données simulées. Chaque identifiant d'objet doit être séparé par un `;`. Si on souhaite tous les objets, on met une `*`. Par défaut si l'attribut n'est pas présent, les données sont générées pour tous les objets.
- `vars` qui définit la liste des variables que l'on souhaite extraire. Chaque nom de variable doit être séparé par un `;`. Si on souhaite toutes les variable, on met une `*`. Les variables de type vecteur doivent être suffixées avec les caractères `[]`. Par défaut si l'attribut n'est pas

présent, les données sont générées pour toutes les variables.



```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <output>
    <autooutfiles colsep=" " dtformat="%d/%m/%Y;%H:%M:%S" commentchar="#">

      <SUout filesuffix="full" selection="*" vars="runoff;infiltration;qoutput" />

      <RSout filesuffix="qoutput" selection="*" vars="qoutput" />

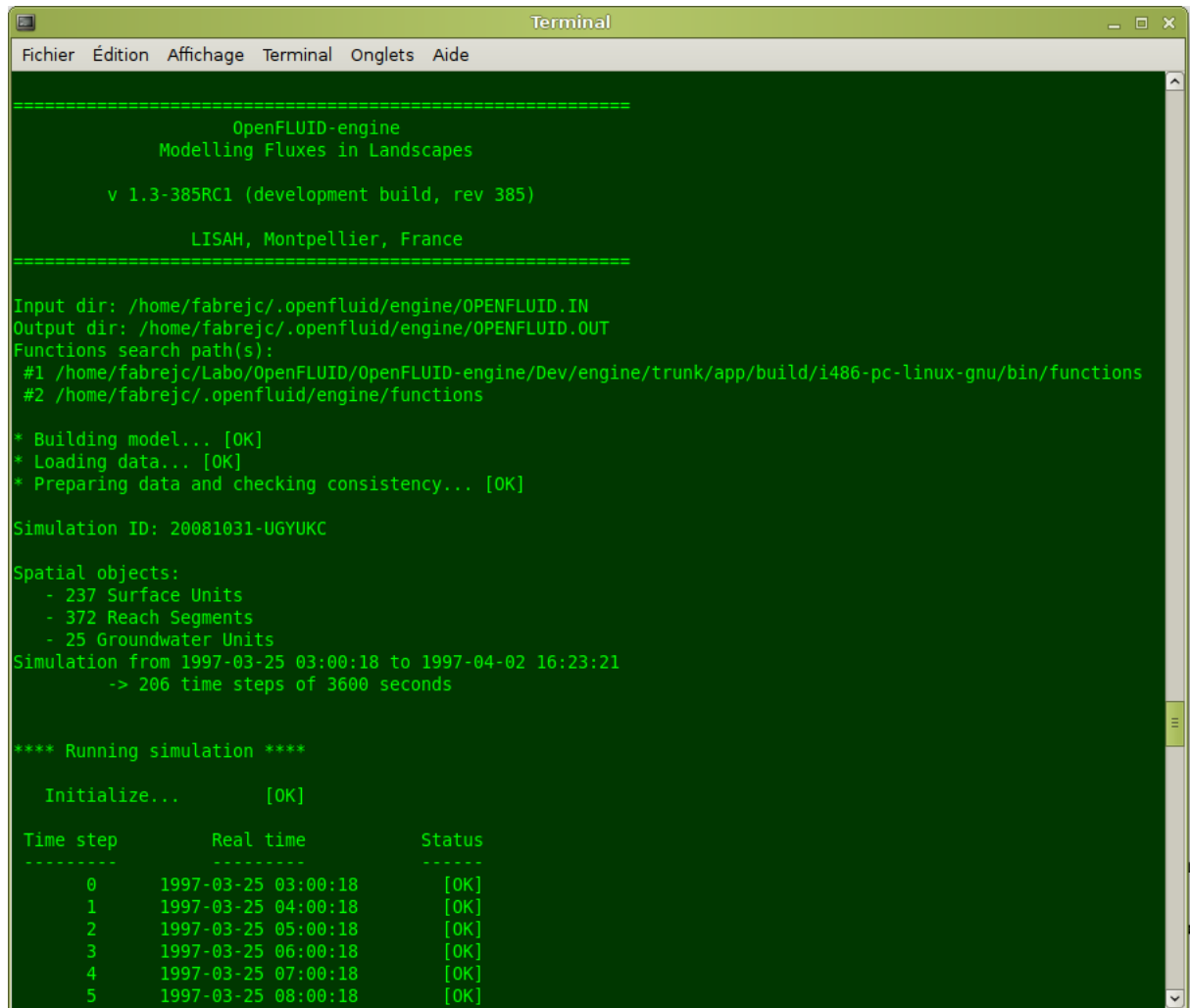
      <GUout filesuffix="fullfull" selection="*" vars="*" />

      <SUout filesuffix="prod" selection="1;18;152" vars="runoff;infiltration" />

    </autooutfiles>
  </output>
</openfluid>
```

2.2 Lancer la simulation

Pour lancer la simulation, il faut lancer la commande `openfluid-engine` dans un shell (sous Unix/Linux) ou exécuter le programme `openfluid-engine.exe` (sous Windows).



```
-----
                OpenFLUID-engine
            Modelling Fluxes in Landscapes

                v 1.3-385RC1 (development build, rev 385)

                LISAH, Montpellier, France
            -----

Input dir: /home/fabrejc/.openfluid/engine/OPENFLUID.IN
Output dir: /home/fabrejc/.openfluid/engine/OPENFLUID.OUT
Functions search path(s):
#1 /home/fabrejc/Labo/OpenFLUID/OpenFLUID-engine/Dev/engine/trunk/app/build/i486-pc-linux-gnu/bin/functions
#2 /home/fabrejc/.openfluid/engine/functions

* Building model... [OK]
* Loading data... [OK]
* Preparing data and checking consistency... [OK]

Simulation ID: 20081031-UGYUKC

Spatial objects:
- 237 Surface Units
- 372 Reach Segments
- 25 Groundwater Units
Simulation from 1997-03-25 03:00:18 to 1997-04-02 16:23:21
-> 206 time steps of 3600 seconds

**** Running simulation ****

    Initialize...      [OK]

Time step      Real time      Status
-----
    0      1997-03-25 03:00:18      [OK]
    1      1997-03-25 04:00:18      [OK]
    2      1997-03-25 05:00:18      [OK]
    3      1997-03-25 06:00:18      [OK]
    4      1997-03-25 07:00:18      [OK]
    5      1997-03-25 08:00:18      [OK]
```

2.3 Exploiter les résultats

Les résultats de la simulation se trouveront, en fin de run, dans un répertoire `.openfluid/engine/OPENFLUID.OUT` se situant dans le répertoire-maison de l'utilisateur. Il est possible de modifier ce comportement par défaut via les options de la ligne de commande (voir en annexe).

Chapitre 3

Annexes

3.1 Lexique

3.2 Options de la ligne de commande

<code>-a, -auto-outdir</code>	génère automatiquement un répertoire contenant un les données résultats
<code>-c, -clean-outdir</code>	vide le répertoire avant l'écriture des données résultats
<code>-f, -functions-list</code>	liste les fonctions de simulation disponibles (n'exécute pas le modèle)
<code>-h, -help</code>	affiche l'aide
<code>-i, -inputdir=<str></code>	spécifie le répertoire contenant les données d'entrée
<code>-k, -openfluid-version</code>	affiche le numéro de version du SDK utilisé
<code>-m, -trace-dir=<str></code>	spécifie le répertoire pour les données trace (en mode trace uniquement)
<code>-o, -outputdir=<str></code>	spécifie le répertoire pour les données résultats
<code>-q, -quiet</code>	désactive l'affichage pendant la simulation
<code>-r, -functions-report</code>	affiche un rapport détaillé sur les fonctions de simulation disponibles (n'exécute p
<code>-s, -no-simreport</code>	ne génère pas de rapport de simulation
<code>-t, -trace</code>	active le mode trace
<code>-v, -verbose</code>	active l'affichage détaillé pendant la simulation
<code>-version</code>	affiche la version du moteur de calcul
<code>-x, -xml-functions-report</code>	affiche un rapport détaillé au format XML sur les fonctions de simulation disponi
<code>-z, -no-results</code>	ne génère pas de fichier de sortie résultats
<code>-no-varname-check</code>	désactive la vérification du nommage des variables

3.3 Généralités sur les formats de fichier OpenFLUID-engine

Les formats de fichiers utilisés par OpenFLUID-engine sont de 3 types : texte-colonnes, XML, XML-mixte.



Les exemples pris ci-dessous pour illustrer les différents formats de fichiers ne sont pas tous tirés de jeux de données issus d'OpenFLUID-engine. Il ne faut donc pas s'attacher au contenu mais plutôt au format des exemples.

3.3.1 Format texte-colonnes

Les fichiers au format texte-colonne contiennent des données en format texte qui sont rangées en colonnes, séparées par au moins un espace ou une tabulation. Toute ligne commençant par le caractère % est ignorée.



```
Pierre 18 15 16
Béatrice 12 15 13
% Yves 7 12 11
Valérie 19 17 15
```

Dans l'exemple ci-dessus, la ligne "Yves" est ignorée.

3.3.2 Format XML

Le format XML est un format de données auto-descriptif, hiérarchique, utilisant les notions de balises et d'attributs. Pour obtenir des informations plus complètes sur ce format largement utilisé, nous vous invitons à vous rendre sur un des sites suivants :

- <http://www.w3.org/XML/1999/XML-in-10-points.fr.html>
- <http://www.commentcamarche.net/xml/xmlintro.php3>
- <http://www.chez.com/xml/initiation/>
- <http://fr.wikipedia.org/wiki/XML>



les fichiers xml ont obligatoirement pour entête `<?xml version="1.0" standalone="yes" ?>`
 les fichiers xml OpenFLUID-Engine ont pour balise racine `<openfluid>`
 pour insérer un commentaire dans un fichier XML ou désactiver une partie du fichier, il faut encadrer la partie souhaitée par `<!--` au début et `-->` en fin.

3.3.3 Format XML-mixte

Le format XML-mixte est basé sur XML. Il est utilisé ici pour la descriptions de propriétés ou de conditions initiales distribuées dans l'espace.

Ce format permet de définir des données en colonnes avec la possibilité d'avoir un nombre de colonnes variables ainsi qu'un ordre et une signification "à la carte" de ces colonnes. Il est appelé XML-mixte car on utilise les caractéristiques du XML pour la description du contenu, mêlées à la facilité d'utilisation du format texte colonne. Le format XML-mixte est un format XML classique avec la particularité d'une zone balisée dans laquelle se trouve les données en colonnes.

Les fichiers XML-mixte contiennent une section principale définie par la balise qui va donner la type d'objet spatial (SU, RS ou GU) ainsi que le type de données (propriétés ou condition initiales). Cette balise peut donc être `<SUprops>`, `<SUini>`, `<RSprops>`, `<RSini>`, `<GUprops>` ou `<GUini>` et contient 2 balises :

- La balise `<columns>` qui, à travers de son attribut `order`, définit l'ordre des colonnes contenues dans la partie données du fichier.
- La balise `<data>` qui contient les données en colonnes. Attention : la première colonne n'est pas décrite dans la balise `<columns>` car elle est obligatoire et définit l'identifiant de l'objet spatial auquel sont rattachées les valeurs de la ligne.



```
<?xml version="1.0" standalone="yes"?>
<openfluid>
  <RSprops>
    <columns order="ks;nmaning" />
    <data>

1 0.003 0.2
2 0.003 0.2
3 0.003 0.2
4 0.003 0.2
5 0.003 0.2

    </data>
  </RSprops>
</openfluid>
```

3.4 Formatage des dates à l'aide de la norme ANSI strftime()

La fonction strftime() permet de transformer une date en une chaîne de caractère en utilisant une chaîne de formatage composée d'indicateurs standardisés représentant les champs d'une date. Ces indicateurs de champs commencent par le caractère % (voir ci-après). Les caractères ordinaires présent dans la chaîne de formatage seront reproduits tels quels. Une tabulation est représentée par le code \t, une fin de ligne par le code \n, le caractère % par le code %%.

Par exemple, la date 19 avril 2007 à 10h11m25s formatée à l'aide des chaînes de formatage suivantes :

- "%d/%m/%Y %H:%M:%S" donnera "19/04/2007 10:11:25"
- "%Y-%m-%d %H.%M" donnera "2007-04-19 10.11"
- "%Y\t%m\t%d\t%H\t%M\t%S" donnera "2007 04 19 10 11 25"

- %a Le nom abrégé du jour de la semaine, en fonction de la localisation en cours.
- %A Le nom complet du jour de la semaine, en fonction de la localisation en cours.
- %b Le nom abrégé du mois, en fonction de la localisation en cours.
- %B Le nom complet du mois, en fonction de la localisation en cours.
- %c La représentation classique de la date et de l'heure, suivant la localisation en cours.
- %C Le siècle (année / 100) sous forme de nombre de deux chiffres. (SU)
- %d Le quantième du mois sous forme de nombre décimal (entre 01 et 31).
- %D Equivalent à %m/%d/%y. Seulement pour les américains. Ces derniers doivent noter que les autres pays préfèrent le format %d/%m/%y. Dans un contexte international ce format est ambigu et doit être évité. (SU)
- %e Comme %d, le quantième du mois sous forme décimal, mais l'éventuel zéro en tête est remplacé par un espace. (SU)
- %E Modificateur : Utiliser un format de rechange. Voir plus bas. (SU)
- %G L'année ISO 8601 avec le siècle comme nombre décimal. L'année sur quatre chiffres correspond au numéro de la semaine ISO (voir %V). C'est le même format que %y, sauf que si le numéro de semaine appartient à l'année suivante ou précédente, c'est celle-ci qui est affichée à la place. (TZ)
- %g Comme %G sans le siècle, c'est à dire avec une année sur deux chiffres (00-99). (TZ)

- %h Equivalent à %b. (SU)
- %H L'heure, sur 24 heures, sous forme de nombre décimal (entre 00 et 23).
- %I L'heure, sur 12 heures, sous forme de nombre décimal (entre 01 et 12).
- %j Le numéro du jour dans l'année (entre 001 et 366). %k L'heure (sur 24 heures) sous forme de nombre décimal (intervalle 0 à 23). Les chiffres uniques sont précédés d'un blanc (voir aussi %H). (TZ).
- %l L'heure (sur 12 heures) sous forme de nombre décimal (intervalle 1 à 12). Les chiffres uniques sont précédés d'un blanc (voir aussi %I). (TZ).
- %m Le numéro du mois (entre 01 et 12).
- %M La minute, sous forme de nombre décimal (00 à 59).
- %n Un caractère saut-de-ligne. (SU)
- %O Modificateur : Utiliser un format de rechange. Voir plus bas. (SU)
- %p L'une des deux chaînes 'AM' ou 'PM' en fonction de l'heure, ou la traduction de ces chaînes en fonctions de la localisation en cours. Midi est traité comme 'PM' et Minuit comme 'AM'.
- %P Comme %p mais en minuscule : 'am' ou 'pm' ou une chaîne locale équivalente. (GNU)
- %r L'heure en notation a.m. ou p.m. Dans la localisation POSIX, c'est équivalent à '%I :%M :%S %p'. (SU)
- %R L'heure en format 24 heures (%H :%M). (SU). Pour avoir les secondes voir %T.
- %s Le nombre de secondes écoulées depuis le 1er Janvier 1970 à 00 :00 :00 UTC. (TZ).
- %S La seconde, sous forme de nombre décimal. (00-61)
- %t Un caractère Tab. (SU)
- %T L'heure en notation 24 heures (%H :%M :%S). (SU)
- %u Le jour de la semaine sous forme décimal, de 1 (Lundi) à 7. Voir aussi %w. (SU)

- %U Le numéro de la semaine dans l'année, sous forme de nombre décimal (00-53), en commençant le comptage au premier Dimanche de l'année. Voir aussi %V et %W.
- %V Le numéro de semaine ISO 8601 :1988 de l'année en cours sous forme de nombre décimal dans l'intervalle 01-53, où la semaine 1 est la première ayant au moins 4 jours dans l'année en cours, et où Lundi est le premier jour de la semaine. Voir aussi %U et %W. (SU)
- %W Le numéro de la semaine dans l'année, sous forme de nombre décimal (00-53), en commençant le comptage au premier Lundi de l'année.
- %w Le numéro du jour de la semaine, sous forme décimale (0-6), Dimanche valant 0. Voir aussi %u.
- %x La représentation usuelle de la date, sans l'heure, dans la localisation en cours.
- %X La représentation usuelle de l'heure, sans la date, dans la localisation en cours.
- %y L'année, sous forme de nombre décimal, sans le siècle (entre 00 et 99).
- %Y L'année, sous forme de nombre décimal, incluant le siècle.
- %z Le fuseau horaire sous forme de décalage GMT. Nécessaire pour créer des dates conformes à la RFC 822 (avec "%a, %d %b %Y %H :%M :%S %z"). (GNU)
- %Z Le nom ou l'abréviation du fuseau horaire.
- %+ La date et l'heure au format `date(1) . (TZ)`
- %% Un caractère '%'.