

Chaîne « Risque » – WaterHeight

Table des matières

I. Présentation du script.....	2
I.1. Description.....	2
I.2. Traitements.....	3
I.3. Dépendances.....	4
II. Utilisation du script.....	5
II.1. Paramètres.....	5
II.2. Lancement.....	6
III. Exemples de résultats.....	7

Index des illustrations

Illustration 1: Flux de données du script WaterHeight.....	3
Illustration 2: Dépendances du script WaterHeight avec la chaîne interne.....	4

I. Présentation du script

I.1. Description

Note : Ce script est en partie issu de la traduction d'un plugin QGIS : 'cartoZI', développé par le SPC Loire Cher Indre, et porté par le SCHAPI¹.

Il a pour but de **produire une donnée de classes de hauteurs d'eau**, par croisement d'une emprise inondée et d'un Modèle Numérique de Terrain. Le fichier de sortie peut-être au format raster et/ou vecteur.

Ces classes de hauteurs d'eau sont paramétrées par l'utilisateur.

Il peut être généré autant de classes que l'on souhaite, sans que les intervalles ne soient forcément égaux.

Dans le paramétrage du script, elles devront respecter un format particulier : en chaîne de caractères, sans espace, séparateur décimal en point, valeurs séparées par une virgule (ex par défaut : '0,0.5,1,1.5,2').

La première valeur correspond à la valeur basse de la première classe, et sert à éliminer toutes les valeurs inférieures. Il s'agit généralement de 0, pour éliminer les valeurs négatives (sur de légères sur-élévations par exemple, où l'altitude du MNT est supérieure à celle de la nappe d'eau).

La dernière valeur correspond à la valeur basse de la dernière classe, et contient toutes les valeurs supérieures.

Intervalles utilisés par défaut : $[0 ; 0,5[- [0,5 ; 1[- [1 ; 1,5[- [1,5 ; 2[- [2 ; +\infty[$

Dans le fichier final, les classes sont représentées en valeur entière, de 1 pour le premier intervalle, jusqu'au nombre de seuils définis par l'utilisateur (de 1 à 5 par défaut).

C'est un script très complet, qui fait appel à de nombreux outils : GDAL/OGR, OTB, GRASS, SAGA

¹ <http://wikhydro.developpement-durable.gouv.fr/index.php/CartoZI>

I.2. Traitements

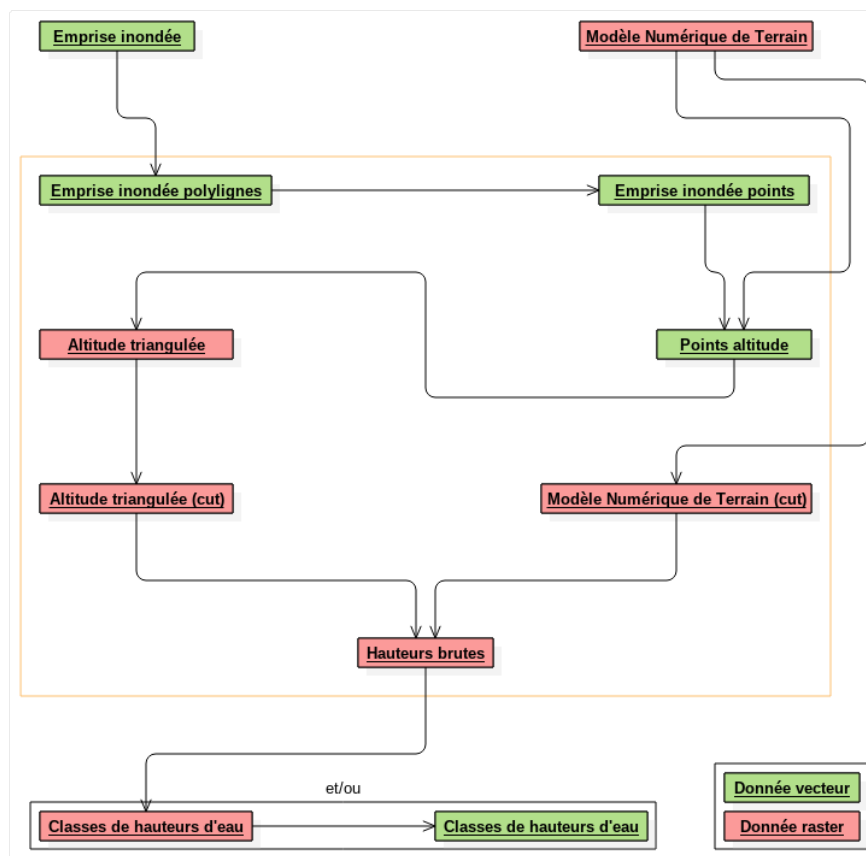


Illustration 1: Flux de données du script WaterHeight.
Les fichiers temporaires sont dans le cadre orange.

La 1^{re} étape consiste à transformer le polygone d’emprise inondée en créant des points le long du périmètre. Le MNT est ensuite échantillonné sous ces points pour récupérer l’altitude (la hauteur d’eau sur ces secteurs en bordure de la zone inondée étant de 0, l’altitude du MNT et du haut de la colonne d’eau est la même). Cette altitude est alors triangulée sur le reste de l’emprise inondée, pour obtenir une nappe d’eau plane. Puis, la différence d’altitude est faite entre le MNT et le sommet de la nappe d’eau (la hauteur triangulée), pour obtenir la hauteur brute pour chaque pixel. C’est ici qu’interviennent les seuils de classes définis par l’utilisateur, pour classer les hauteurs brutes. Enfin, si cela est demandé, une vectorisation du raster des classes de hauteurs d’eau peut être réalisée.

I.3. Dépendances

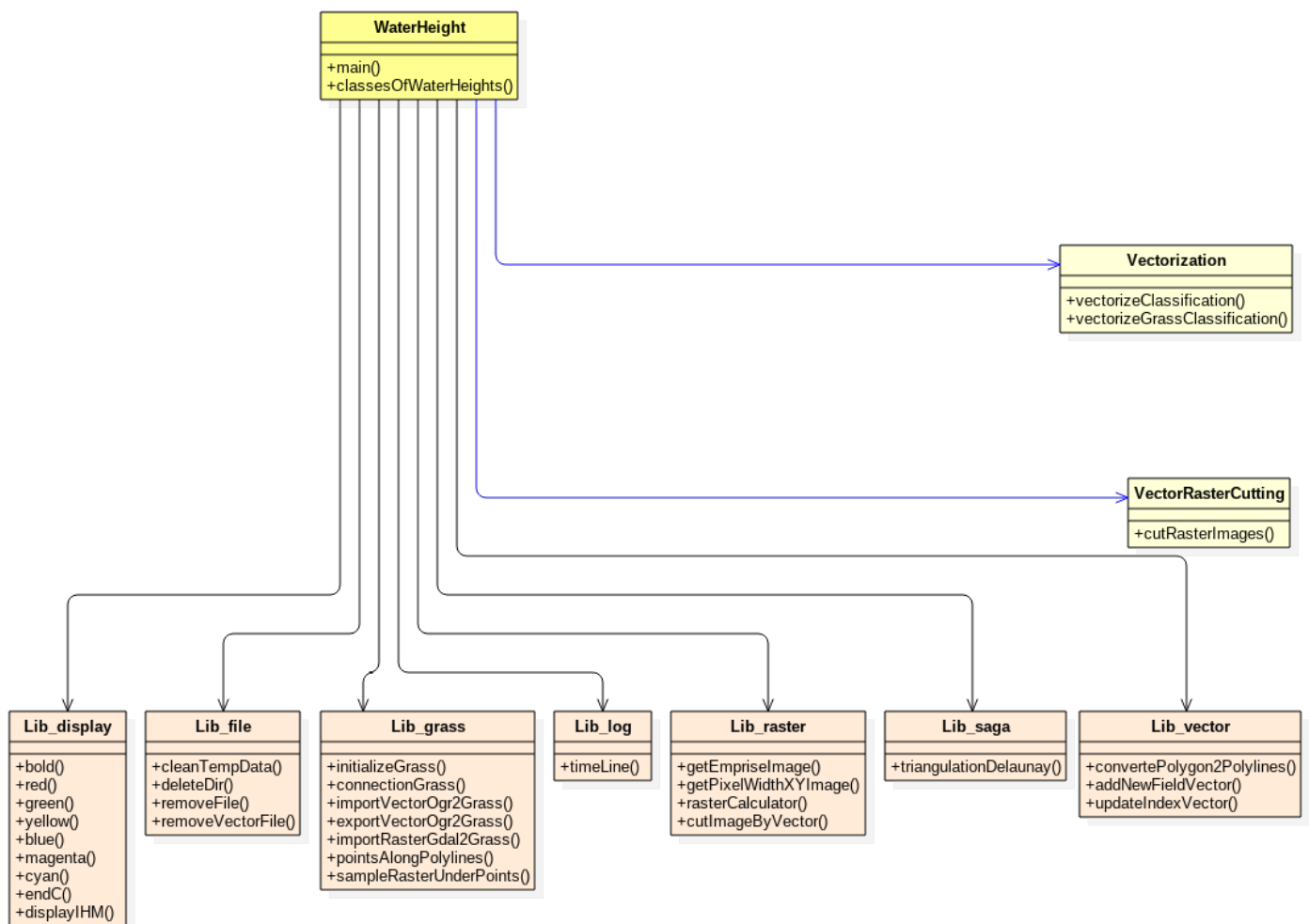


Illustration 2: Dépendances du script WaterHeight avec la chaîne interne.

En plus de ces dépendances au reste de la chaîne interne, il fait appel aux bibliothèques ‘os’, ‘argparse’, ‘ogr’.

II. Utilisation du script

II.1. Paramètres

Paramètres spécifiques du script :

- **input_flooded_areas_vector (-infld)** : fichier de l'emprise inondée (vecteur entrée)
- **input_digital_elevation_model_file (-indem)** : fichier du MNT (raster entrée)
- output_heights_classes_file (-outr) : fichier des classes de hauteurs d'eau (raster sortie)
- output_heights_classes_vector (-outv) : fichier des classes de hauteurs d'eau (vecteur sortie)
- heights_classes (-hcla) : classes de hauteurs d'eau à générer, en mètres. Par défaut : '0,0.5,1,1.5,2'

Seuls les paramètres en gras sont obligatoires.

Il faut renseigner au moins un des deux fichiers de sortie.

Paramètres liés à GRASS :

- grass_gisbase (-ggis) : variable d'environnement GRASS. Par défaut : `os.environ['GISBASE']`
- grass_gisdb (-gdbn) : nom de la géodatabase GRASS. Par défaut : 'GRASS_database'
- grass_location (-gloc) : nom du secteur GRASS (= *location*). Par défaut : 'LOCATION'
- grass_mapset (-gmap) : nom du jeu de cartes GRASS (= *mapset*). Par défaut : 'MAPSET'

En plus de ces paramètres spécifiques et ceux liés à GRASS, le script fait appel à des paramètres liés aux fichiers (projection, valeur NoData, format/extension des rasters/vecteurs) et aux paramètres généraux (fichier log, sauvegarde des fichiers temp, ré-écriture sur fichier existant, niveau de debug).

II.2. Lancement

Exemples d'utilisation avec 5 classes (par défaut) : $[0 ; 0,5[- [0,5 ; 1[- [1 ; 1,5[- [1,5 ; 2[- [2 ; +\infty[$.

Via le séquenceur :

Il s'agit de la tâche 10_RSQ dans le séquenceur.

C'est un script autonome, qui n'a normalement besoin d'aucune dépendance avec d'autres applications.

```
<Task10_RSQ_WaterHeight_List>
  <Task10_RSQ_WaterHeight>
    <InputFloodedAreasVector>/mnt/RAM_disk/WaterHeight/emprise_inondee.shp</InputFloodedAreasVector>
    <InputDigitalElevationModelFile>/mnt/RAM_disk/WaterHeight/MNT.tif</InputDigitalElevationModelFile>
    <OutputHeightsClassesFile>/mnt/RAM_disk/WaterHeight/WaterHeight.tif</OutputHeightsClassesFile>
    <OutputHeightsClassesVector>/mnt/RAM_disk/WaterHeight/WaterHeight.shp</OutputHeightsClassesVector>
    <HeightsClasses>0,0.5,1,1.5,2</HeightsClasses>
    <Grass>
      <EnvironmentVariable>GISBASE</EnvironmentVariable>
      <DatabaseName>GRASS_database</DatabaseName>
      <Location>LOCATION</Location>
      <Mapset>MAPSET</Mapset>
    </Grass>
  </Task10_RSQ_WaterHeight>
</Task10_RSQ_WaterHeight_List>
```

En ligne de commande :

Il se lance comme tout autre script Python :

```
python3 -m WaterHeight
-infld /mnt/RAM_disk/WaterHeight/emprise_inondee.shp
-indem /mnt/RAM_disk/WaterHeight/MNT.tif
-outr /mnt/RAM_disk/WaterHeight/WaterHeight.tif
-outv /mnt/RAM_disk/WaterHeight/WaterHeight.shp
-hcla 0,0.5,1,1.5,2
```

Dans un souci de compréhension, seuls les paramètres propres au script sont renseignés.

Appel en tant que fonction :

S'agissant d'un indicateur indépendant, il n'a pas vocation à se faire appeler comme fonction dans un autre script Python. L'appel peut néanmoins se faire comme toute autre fonction :

```
from WaterHeight import classesOfWaterHeights

classesOfWaterHeights('/mnt/RAM_disk/WaterHeight/emprise_inondee.shp',
  '/mnt/RAM_disk/WaterHeight/MNT.tif',
  '/mnt/RAM_disk/WaterHeight/WaterHeight.tif',
  '/mnt/RAM_disk/WaterHeight/WaterHeight.shp',
  heights_classes='0,0.5,1,1.5,2')
```

Dans un souci de compréhension, seuls les paramètres propres à la fonction sont renseignés.

III. Exemples de résultats