

Sommaire

Liste des abréviations	1
Résumé	2
Introduction	3
I. L'entreprise	4
A. Présentation	4
B. Besoins de l'entreprise	5
II. Exemples de réponses aux besoins de l'entreprise par des SIG open source	7
A. programmation d'un outil de simulation de digues et de leurs inondations	7
B. Organisation des données sous GRASS	15
C. Obtenir MNT et MNS à partir de LAS sol et LAS surface	17
D. Modifier la topographie sur un MNT	18
E. Evolution de la topographie d'une tourbière.	21
F. Créer une carte raster représentant la topographie du sous-sol sous une couche de tourbe	22
G. Numérisation d'un relevé topographique	23
H. Création d'un outil de profils dont les données sont facilement exportables dans Excel	24
I. Esquisse de projet pour lutter contre l'assèchement et l'érosion d'une tourbière	25
III. Logiciels utilisés	27
Discussion	28
Conclusion	29
Bibliographie	30
Annexe	31

Liste des abréviations

GIS : Geographic Information System

GRASS : Geographic Resources Analysis Support System

QGIS: Quantum GIS

GDAL: Geospatial Data Abstraction Library

MNT: modèle numérique de terrain

LIDAR : light detection and ranging

NTF : Nouvelle triangulation de la France (système géodésique)

RGF93 : réseau géodésique français 1993 (système géodésique)

Résumé

Un des objectifs du stage étaient de créer une application permettant de trouver la digue la plus courte en partant d'un point de coordonnées, de mesurer la longueur de cette digue et de simuler l'inondation qu'elle provoque. Les autres objectifs étaient de résoudre des problématiques méthodologiques liées au SIG et d'organiser les données géographiques pour les différents sites sur lesquels l'entreprise travail.

Cela devait être fait avec un ou des logiciels SIG open source.

Pour cela, avec pour bagages les cours suivis du certificat complémentaire en géomatique, je me suis formé pendant cinq semaines avant le stage à l'utilisation du langage de programmation Python et familiarisé avec QUANTUM GIS et GRASS GIS.

L'application et les méthodes fonctionnent et sont utilisées par les membres de l'entreprise. Naturellement elles peuvent encore évoluer et être améliorées.

Créer une application et résoudre des problématiques méthodologiques sont des moyens très efficaces pour comprendre et apprendre.

Introduction

Les informations géographiques ont un rôle important dans la gestion et réhabilitation des milieux naturels (délimitation de réserves, de zones tampons, cartographies floristiques, faunistiques et des milieux, topographie...).

Pour pouvoir utiliser efficacement ces informations il est nécessaire de disposer de méthodologies et d'outils adaptés.

Les logiciels les plus couramment utilisés dans le domaine des Système d'Information Géographique (SIG) sont ArcGIS et Mapinfo. Ce sont des logiciels "propriétaires", c'est-à-dire qu'ils restent la propriété intellectuelle des entreprises qui les ont développées et qui, pour pouvoir faire payer son utilisation, son développement et sa maintenance, doivent protéger le code. Cette protection du code bloque la possibilité aux utilisateurs de l'améliorer et de le développer, et rend plus compliqué la création et la diffusion d'outils se basant sur ces logiciels par les utilisateurs.

Il existe des logiciels dit "open source" dont les codes sont libres d'utilisation, de modification et de diffusion. Dans ce cas l'avantage principal pour une entreprise est d'avoir accès à un nombre sans cesse croissant d'outils, de pouvoir les modifier ou même de créer soi-même des outils adaptés à ses besoins en s'appuyant sur ce qui existe déjà.

Je limite volontairement la comparaison entre logiciel propriétaire et open source à l'aspect "réponse aux besoins spécifiques par les outils proposés" pour ne pas sortir du sujet et en partant du principe que les deux types de logiciel répondent en grande partie à ces besoins. En effet dans les deux cas les outils existants sont nombreux et répondent souvent directement aux besoins mais pas à tous.

Mon travail durant le stage a été de définir parmi les outils existants des logiciels open source lesquels répondraient le mieux aux besoins de l'entreprise et si nécessaire d'agencer des outils existants ou d'en créer.

En d'autres termes et d'un point de vue plus général les questions auxquelles ce stage tente de répondre par un exemple sont :

Que peuvent apporter des logiciels open source à une entreprise impliquée dans la gestion des milieux naturels ?

Les logiciels open sources peuvent-ils répondre précisément aux besoins d'une entreprise impliquée dans la gestion des milieux naturels ?

I. L'entreprise

A. Présentation

Créé en 2001, LIN'eco est un bureau d'étude spécialisé dans la gestion conservatoire des zones humides.

L'entreprise s'occupe de produire des plans de gestion, de mener des négociations avec les acteurs locaux, de planifier des mesures de réhabilitation, d'accompagner des chantiers et d'effectuer un suivi scientifique des mesures.

LIN'eco est également partenaire de l'OFEV (Office Fédéral de l'Environnement) en tant qu'expert technique et spécialiste des inventaires des biotopes marécageux d'importance nationale en Suisse.

Le bureau se situe à Reconvilliers dans le Jura Bernois.

Il compte actuellement 2 personnes, Philippe Grosvernier son créateur et depuis fin 2012, Elisabeth Contesse.

B. Besoins de l'entreprise

LIN'eco s'occupe de la réhabilitation de milieu humide, lors d'un mandat la première étape de son travail est de comprendre ce qui ne fonctionne pas et pourquoi. Le principal problème des biotopes marécageux est l'assèchement qui peut être dû à un déficit d'alimentation en eau ou à l'accélération de l'évacuation de l'eau.

L'information sur la topographie disponible sous forme de modèle numérique de terrain (MNT) est une aide précieuse pour comprendre le fonctionnement hydrologique d'un marais. Pour l'utiliser efficacement l'entreprise a besoin de méthodes et d'outils pour manipuler et interroger les données contenues dans les MNT.

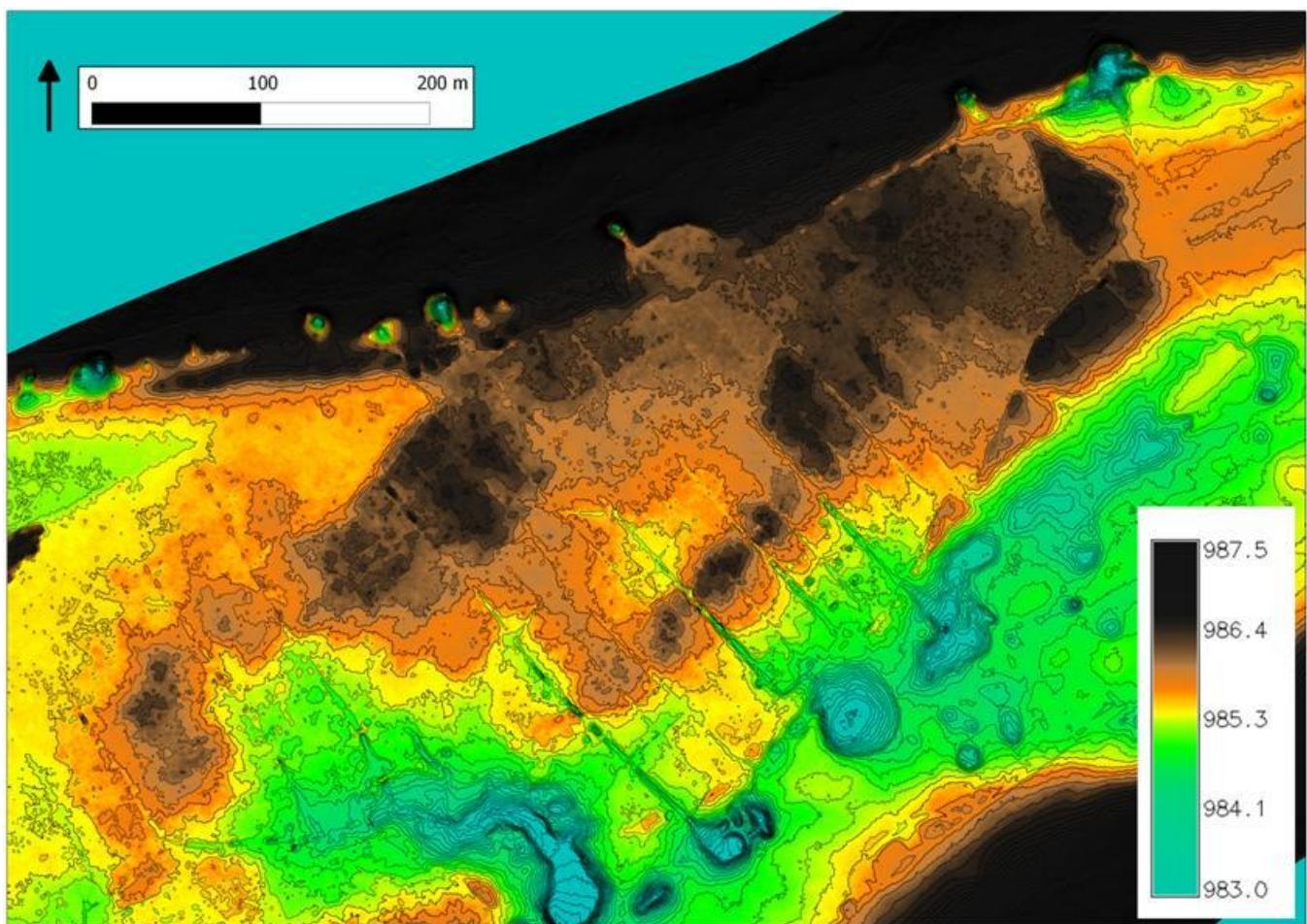


Figure 1 : représentation d'un modèle numérique de terrain avec, superposées dessus, des courbes de niveau tout les 20cm de dénivelé.
Tourbière de la Chaux des Breuleux.

L'accélération de l'évacuation de l'eau dans les marais est causée principalement par des réseaux souterrains de tuyaux de drainage, par des fossés de drainage et par la modification de la topographie résultant de l'extraction de la tourbe.

Les mesures utilisées pour contrer les effets des fossés sont la mise en place de barrages ou le remplissage.

Dans certains cas une ou plusieurs digues peuvent être une bonne solution pour bloquer ou allonger le cheminement de l'eau et ainsi la maintenir dans le système quand la topographie a été modifiée par l'extraction de la tourbe.

Lors de la réflexion et de la préparation des mesures de réhabilitation l'entreprise a besoin de méthodes et d'outils pour simuler les digues et modifier la topographie.



Figure 2 : Fossés de drainages dans la tourbière de la Chaux des Breuleux, dans le Jura Bernois.

II. Exemples de réponses aux besoins de l'entreprise par des SIG open source

Une partie des tâches m'ont été attribuées en début de stage, puis, au fur et à mesure de leur achèvement, d'autres se sont ajoutées. Pour chacune, Philippe Grosvernier me décrivait ce qu'il voulait, nous discutions pour préciser l'objectif, puis je me lançais.

A. Programmation d'un outil de simulation de digues et de leurs inondations

LIN'eco travail à la réhabilitation de biotopes marécageux dont le principal problème est l'assèchement.

Lorsque cet assèchement est du à des fossés de drainages ou à la modification de la topographie lors de l'extraction de la tourbe, il est utile de pouvoir simuler l'effet d'une ou plusieurs digues ou barrages sur le fonctionnement hydrologique du marais.

Pour cette raison un des objectifs du stage a été de créer une application qui permette de définir l'orientation et la longueur du barrage ou de la digue la plus courte à partir d'un point de coordonnées en fonction de l'altitude de l'inondation voulue puis de montrer l'étendu de l'inondation provoquée.

A.1 Ecriture du script

Pour l'écriture du script, j'ai travaillé en conceptualisant une structure théorique avec les différentes étapes/parties nécessaires à l'obtention du résultat, c'est-à-dire les longueurs et l'affichage des digues et inondations obtenues.

Puis j'ai cherché les outils dans Grass et les librairies fonctionnant avec python permettant d'effectuer ces étapes. Parallèlement, ma conception du script a évolué avec ma connaissance des librairies fonctionnant avec Python.

Dans les paragraphes qui suivent, je décris l'idée et la méthode utilisée pour les étapes principales (pour plus de détails voir le script "**ProjetDigue.py**" dans "réalisation_concrète.zip").

En résumé : dans un premier temps, le script recherche les extrémités de digues en partant du point de coordonnées de départ, choisie la plus courte, puis l'incruste dans le MNT pour calculer l'inondation, cela en boucle pour les différentes altitudes choisies, puis il ouvre une fenêtre permettant la visualisation des digues et inondations obtenues sur un fond de carte choisi dans les options par l'utilisateur. Il permet aussi d'enregistrer les couches Grass et images créées.

Obtention des digues et des inondations

Pour la gestion des input, output et options, une fenêtre d'interaction entre l'utilisateur et le script est proposée par GRASS. Les coordonnées de la digue et de la source de l'inondation sont copiées sous GRASS (r.what) et collées dans les inputs.

Une première étape du script consiste à définir l'orientation pour obtenir la digue la plus courte à partir des coordonnées et de l'altitude des inondations voulues.

Pour cela mon idée est de vérifier la valeur des pixels, en avançant pixel par pixel dans plusieurs directions, en s'arrêtant lorsque la valeur du pixel dépasse la valeur d'altitude choisie pour l'inondation. Le problème avec cette méthode est que la digue peut être arrêtée par un seul pixel ou un petit groupe avec une altitude supérieure. Si cela se produit, l'inondation fait simplement le tour de ce pixel et la digue est inutile.

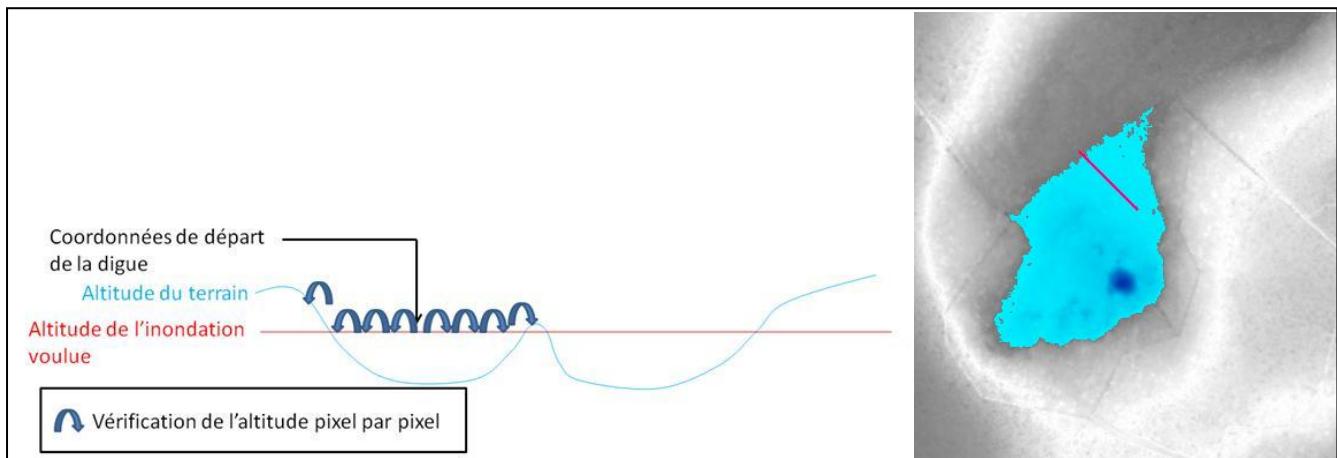


Figure 3: la digue s'est arrêtée sur un pixel dont la valeur est supérieure à l'altitude de l'inondation.

Pour remédier à cela j'ai ajouté une option qui permet de demander au script de continuer à tester la valeur des pixels tant que le nombre de pixels supérieurs à l'altitude désirée n'atteint pas le nombre fixé par l'utilisateur dans cette option. Si la valeur des pixels redévient inférieure à la valeur d'altitude de l'inondation voulue avant d'avoir atteint le nombre fixé dans cette option la recherche continue.

Si le nombre de pixels, dont la valeur est supérieure à l'altitude désirée, atteint le nombre fixé dans cette option l'extrémité de la digue sera située au premier de ces pixels.

Ensuite le script choisit le segment le plus court.

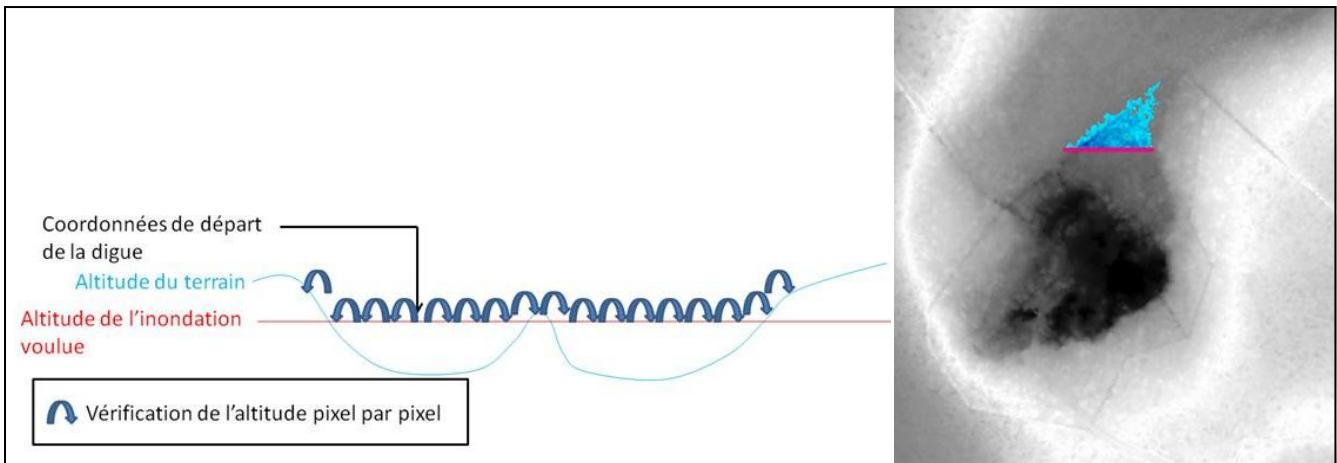


Figure 4 : la digue ne s'est pas arrêtée sur un pixel dont la valeur est supérieure à l'altitude de l'inondation car l'option a autorisé le saut du pixel, l'orientation précédente n'est plus la digue la plus courte, une digue ayant une autre orientation est donc conservée.

On obtient ainsi les coordonnées des extrémités de la digue.

La longueur de la digue est calculée à partir de ces coordonnées.

Pour l'essai du programme, je travaillais seulement sur 4 orientations (nordest/sudouest ; nord/sud ; nordouest/sudest ; est/ouest). Et c'est resté car cela convenait à Philippe Grosvernier.

La librairie GDAL répond aux besoins de cette étape.

J'ai trouvé la méthode pour obtenir la valeur d'un pixel avec la librairie GDAL sur une page du site portailsig.org écrite par Martin Laloux.

Je n'ai pas réussi à faire fonctionner les fonctions GDAL avec le fichier raster déjà importé dans GRASS, par conséquent le script exporte le raster pour travailler avec GDAL.

Les coordonnées géoréférencées sont converties en coordonnées pixels pour le travail avec GDAL en dehors de GRASS, puis les coordonnées pixels des extrémités de la digue la plus courte sont converties en coordonnées géoréférencées.

L'étape suivante est d'importer la digue dans GRASS.

Pour pouvoir importer dans GRASS la digue qui n'est pour l'instant qu'une paire de coordonnées, je fais écrire au script (open,write,close) un fichier .asc qui est un vecteur sous forme de fichier texte assez simple :

VERTI

L 2

"coordonnées 1"

"coordonnées2"

Dans la deuxième ligne, "L" signifie vecteur ligne, "2" pour 2 noeuds.

De cette manière, un vecteur de la digue la plus courte est obtenu et est importé dans GRASS.

Ensuite, avec les fonctions de GRASS, ce vecteur est rastérisé, la valeur d'altitude voulue est donnée aux pixels représentant la digue puis une couche raster est créée avec les valeurs du MNT sauf à l'emplacement de la digue, là on lui donne les valeurs de la digue.

Un MNT comprenant la digue est ainsi obtenu sur la base duquel on peut simuler une inondation.

Cela fonctionne en boucle et recommence pour chaque valeur d'altitude d'inondation que l'on veut simuler.

Les fichiers temporaires sont effacés à la fin de chaque boucle.

Les couches gardées pour chaque altitude d'inondation sont le raster représentant la digue et le raster représentant l'inondation. Ils sont exportés au format .png pour être affichés hors GRASS.

Affichage des résultats obtenus

Etant donné que les ordinateurs de l'entreprise fonctionnent sous Windows et que je ne pouvais pas décentement proposer de créer une machine virtuelle linux seulement pour faire tourner le script, plusieurs difficultés se sont imposées dès le départ. La plus grosse étant l'impossibilité d'afficher les couches dans GRASS sous Windows via une ligne de commande, et des problèmes d'interactivité (voir annexe 1 ou le paragraphe *Known issues/won't fix* sur la page http://grasswiki.osgeo.org/wiki/WinGRASS_Current_Status).

Ce qui veut dire que l'application ne pourra pas utiliser l'affichage de GRASS, il faudra qu'elle contienne sa propre fenêtre d'affichage.

Une fois les images des inondations créées, il faut pouvoir les afficher. Pour cela j'ai utilisé Wxpython qui est une boîte à outil servant à construire une interface graphique.

La principale difficulté a été de superposer trois images (digue, inondation et image de fond) et de pouvoir les faire défiler (scroll) correctement ensemble. Après de nombreuses tentatives avec des méthodes différentes qui ne permettaient pas d'obtenir à la fois l'affichage des trois images et leur défilement correct, la solution a été de faire un masque sur la couleur noire représentant les valeurs NULL.

Les images contiennent dans leur nom l'altitude qui servira de références pour les appeler (ex : digue354.png, lac354.png).

Pour passer d'une image à l'autre j'ai écrit des fonctions, appelées par une interaction de l'utilisateur avec des boutons, qui changent la valeur d'un objet contenant la valeur d'altitude qui sert de référence puis lance l'affichage des images correspondantes.

A.2 Fonctionnement de l'outil

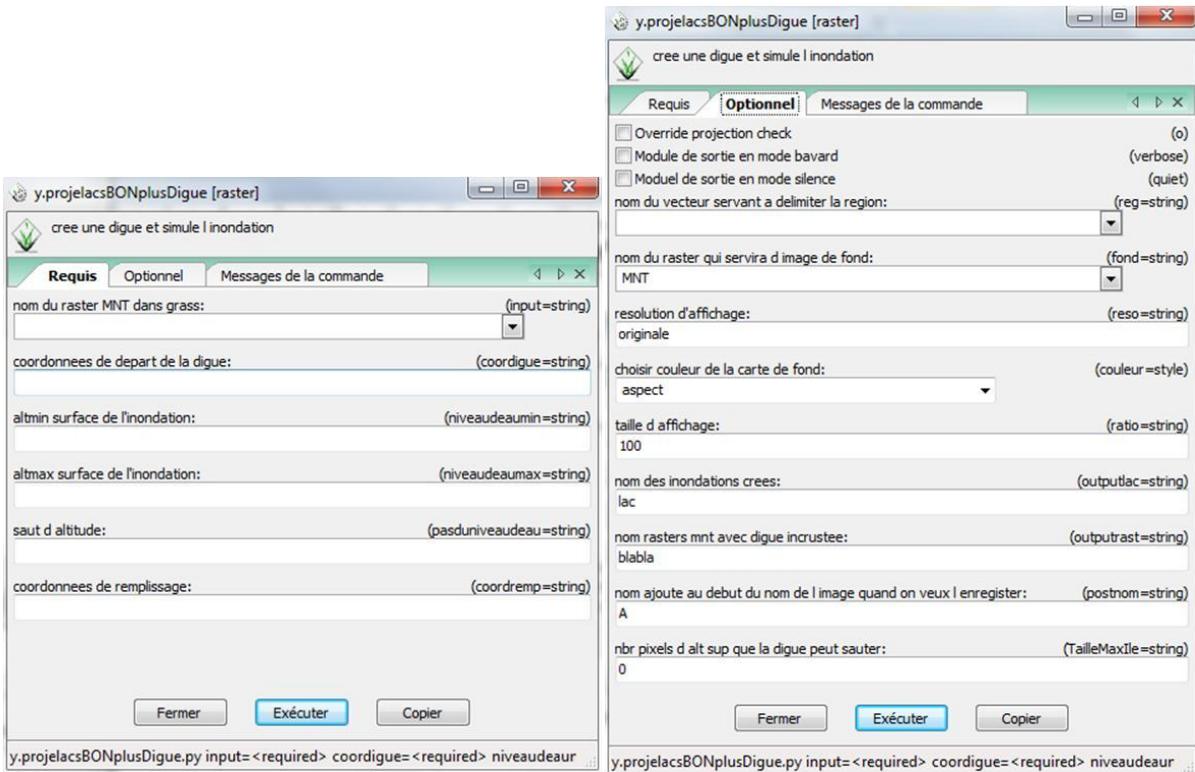


Figure 5 : fenêtre de dialogue de l'application pour entrer les input.

Les paramètres obligatoires sont:

Nom du raster dans Grass : nom du MNT dans grass à utiliser.

Coordonnées de départ de la digue : coordonnées du point de départ du calcul de la digue.

Altmin surface de l'inondation : altitude de l'inondation la plus basse.

Altmax surface de l'inondation : altitude de l'inondation la plus haute.

Saut d'altitude : saut d'altitude entre les inondations calculées.

Les inondations seront calculées aux altitudes : [Altmin inondation ; Altmin inondation +1x_Saut d'altitude ; Altmin inondation +2x_Saut d'altitude ; Altmin inondation +3x_Saut d'altitude ; ... ; Altmin inondation + n x_Saut d'altitude ; Altmax_inondation].

Coordonnées de remplissage : coordonnées de la source de l'inondation.

Les paramètres optionnels sont:

Nom du vecteur servant à délimiter la région : Pour limiter le temps de calcul on peut utiliser un polygone vecteur pour délimiter la région calculée et affichée. Par défaut le programme utilise la région du MNT en input.

Nom du raster servant d'image de fond : la digue et l'inondation doivent être calculées sur un MNT mais cette option permet de choisir l'image de fond sur laquelle les résultats seront affichés. (tout raster importé dans Grass peut être utilisé)

Résolution : résolution d'affichage en mètre, une résolution moins fine permet au programme d'aller plus vite dans les parties où il n'utilise pas la résolution originale. Par défaut le programme utilise la résolution du MNT en input.

Choisir couleur de la carte : Choix de l'aspect de la carte de fond.

Taille d'affichage : ratio gérant la taille d'affichage initial (en %).

Suivant la résolution et la taille de la région, l'image affichée peut être petite ou grande.

Pour éviter de devoir zoomer ou dézoomer dès le début on peut régler ce paramètre.

Nom des inondations créées : nom des rasters représentant l'inondation (créés dans grass).

Nom rasters mnt avec digue créées : nom des rasters où la digue est ajouté au MNT (créés dans grass).

Nom ajouté au début du nom de l'image quand on veut l'enregistrer : l'application permet d'enregistrer le rendu (digue, et inondation sur image de fond) et de choisir le dossier, cette option permet de définir le nom de base des images enregistrées, sera ensuite ajouté leur altitudes.

Nbr pixels d'alt sup que la digue peut sauter :

Pour choisir la longueur de la digue le programme avance de pixel en pixel jusqu'à ce que la valeur du pixel soit supérieure à l'altitude désirée.

Le problème de cette approche est que s'il y a qu'une « île » d'altitude supérieure, la digue va s'arrêter là et l'inondation va faire le tour rendant la digue inutile.

Pour éviter ça on peut autoriser le programme tester si le nombre de pixel supérieur à l'altitude désirée ne dépasse pas le nombre fixé dans ce paramètre et à continuer s'il est inférieur.

Une fois les inputs remplis et le bouton "exécuté" cliqué, après un temps de calcule, la fenêtre suivante s'affiche :

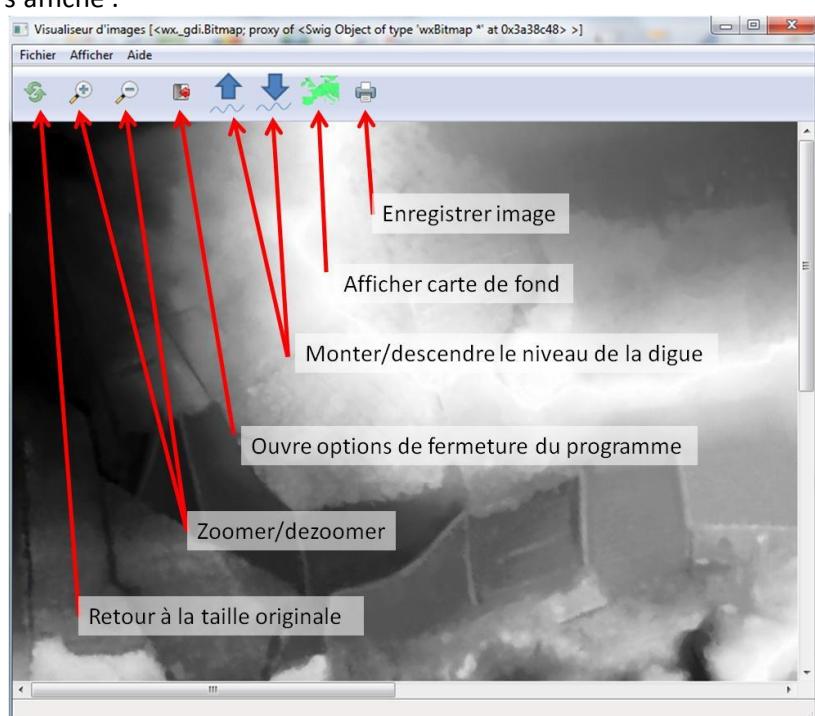


Figure 6 : Fenêtre d'affichage des résultats.

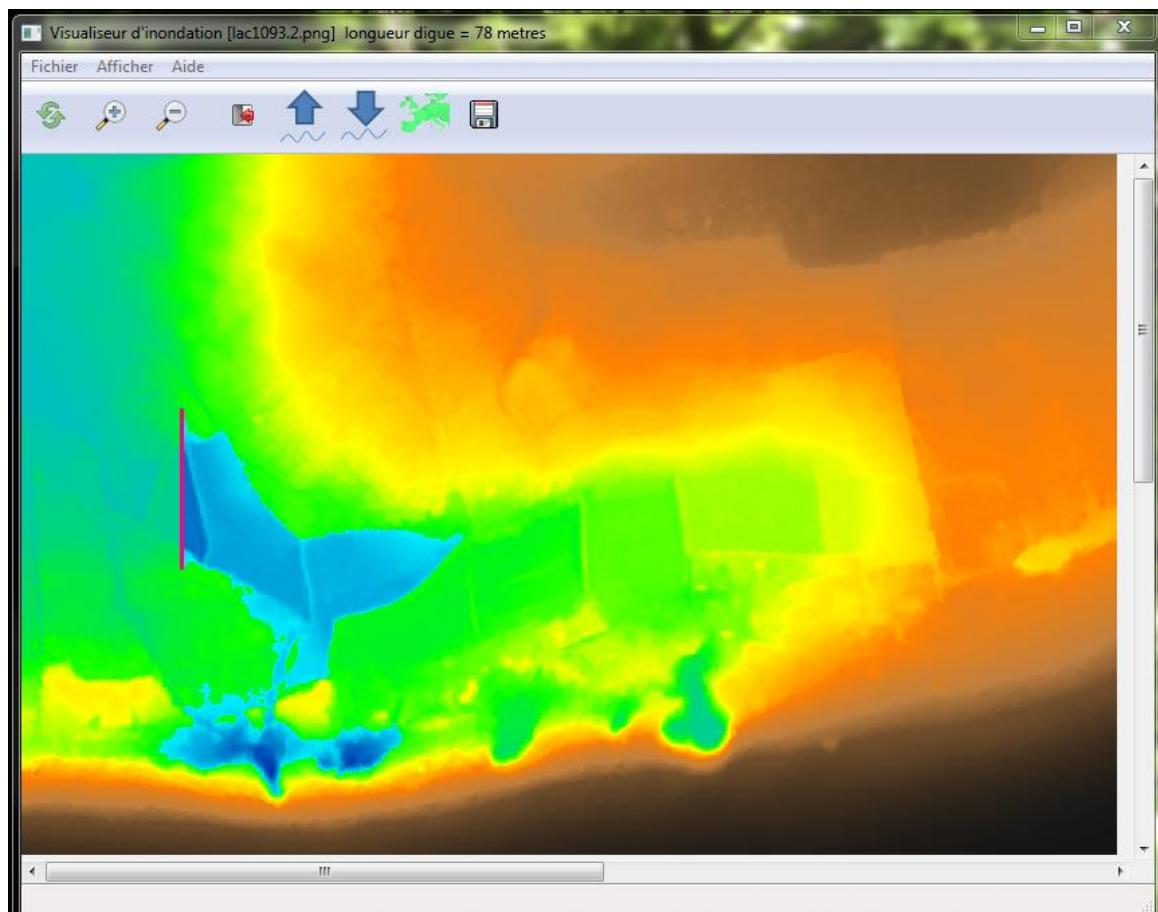
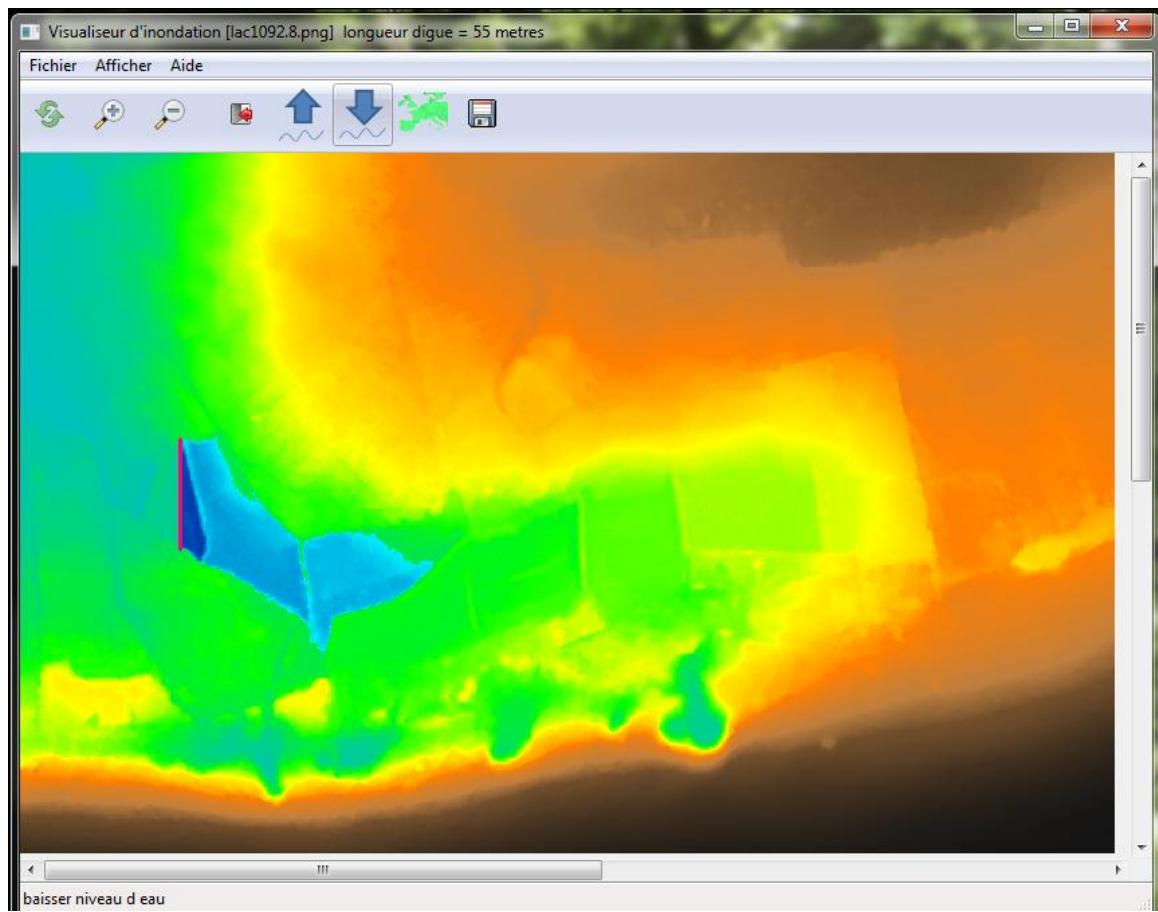


Figure 7 : exemples d'affichage de la digue (rouge) et de l'inondation (bleu), dans le titre de la fenêtre sont notées l'altitude de l'inondation et la longueur de la digue.

Pour quitter il faut cliquer sur le bouton ‘quitter’ ou dans le menu, cela ouvre une fenêtre avec les options de fermeture.

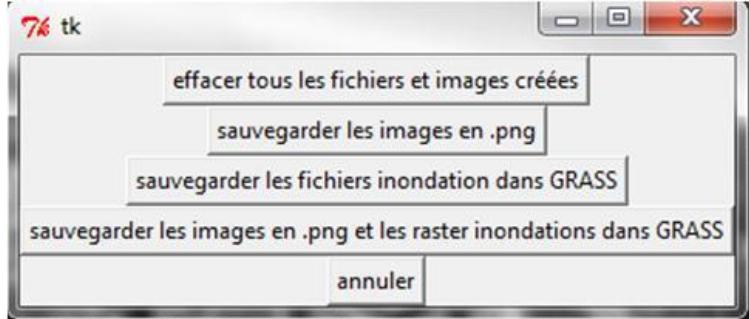


Figure 8 : Options de fermeture

Les fichiers sauvegarder dans Grass le seront sous le nom définit dans les options de l’application.

Pour la sauvegarde des images une fenêtre s’ouvrira pour choisir le dossier dans lequel sauvegarder, sous le nom définit dans les options de l’application.

B. Organisation des données sous GRASS

Objectif :

Obtenir un ensemble de jeux de données clairs et utilisables, et créer une fiche par site décrivant les couches importantes.

Données disponibles :

Un secteur GRASS déjà créé et des fichiers sous formes diverses (.shp, .asc , .xyz, .tif, .las, etc.) répartis dans des dossiers organisés par sites.

Travail effectué :

Une première partie du travail était de comprendre le fonctionnement des bases de données GRASS.

Puis nous avons discuté des données existantes, lesquelles garder, renommer, ou archiver.

J'ai ensuite créé un secteur GRASS par site dans lequel j'ai importé ou "créé" les couches.

Puis j'ai écrit une fiche par site expliquant les caractéristique de la région (le système de projection, l'entendue, la résolution des rasters) et décrivant brièvement le contenu des couches.

Les secteurs GRASS des sites français et leurs couches utilisaient le système géodésique NTF et la projection Lambert II étendu. Elisabeth Contesse a importé des nouvelles couches reçues et s'est rendu compte qu'elles utilisent le système géodésique RGF93 et la projection Lambert93. En effet en France le système légal est le RGF 93 depuis 2001 mais cela n'est concrètement entré en vigueur qu'en 2009 et beaucoup de données sont encore en NTF (Harmel A. ,2009). J'ai donc recréé les secteurs GRASS des sites français et converti les coordonnées des couches en Lambert II étendu vers des coordonnées en Lambert93.

Résultat :

Le résultat est une base de données unique pour tous les sites avec un secteur GRASS pour chacun.

voir "B.fichesSecteursGrass.pdf" dans "réalisation_concrète.zip".



Figure 9: arborescence de la base de données GRASS et fenêtre d'ouverture de GRASS.

C. Obtenir MNT et MNS à partir de LAS sol et LAS surface

Objectif :

Obtenir un MNT et MNS à partir des fichiers LAS contenant les points sol et des fichiers LAS contenant les points surface. Écrire un tutoriel pour expliquer la démarche.

Données disponibles :

Fichiers LAS contenant les points sol et fichiers LAS contenant les points surface.

Travail effectué :

L'idée est d'importer le fichier LAS sous forme raster dans GRASS et d'interpoler toute la surface à partir des points.

Le problème est que nos ordinateurs ne sont pas assez puissants pour faire l'interpolation sur les surfaces et avec la résolution spatiale demandées.

J'ai donc importé le LAS sous forme raster puis j'ai fais l'interpolation par petites zones.

Comme pour le fichier sur lequel je travaillais cela durait environ quatre heures et que c'est répétitif j'ai écrit un script python s'ouvrant sous GRASS qui prend en charge les étapes du travail.

Voir "**C.LASinterpol.py**" dans "réalisation_concrète.zip".

Voir "**C.Tuto_import_interp_Las.pdf**" dans "réalisation_concrète.zip".

Pour "découper" le LAS importé dans GRASS sous forme raster, j'ai fait en sorte que le script déplace la région de travail de GRASS pour faire l'interpolation par petites zones puis qu'il assemble les dalles ainsi obtenues.

D. Modifier la topographie sur un MNT.

Objectif :

Trouver une méthode qui permette de modifier la topographie d'un MNT.

Première méthode

Travail fait :

Résumé de la méthode : obtenir les courbes de niveau du terrain actuel et s'en servir de référence pour créer des courbes de niveau correspondant à la topographie voulue que l'on rasterise puis interpole pour obtenir la nouvelle topographie.

Voir "D. Modifier la topographie d un MNTsans le script.pdf" dans "réalisation_concrète.zip".

Voir "D. Modifier la topographie d un MNT avec script.pdf" dans "réalisation_concrète.zip".

Voir "D.Script_Modif_topo.py" dans "réalisation_concrète.zip".

Exemples d'utilisations :

On extrait les courbes de niveau du MNT, sur la figure 10 les courbes de niveau sont tout les 10cm de dénivelé.

Puis on crée un nouveau vecteur auquel on donne un attribut contenant les altitudes et dans lequel on dessine les nouvelles courbes de niveau correspondant à la topographie souhaité (en rouge ci-dessous) en s'aidant des courbes de niveau extraites du MNT (en gris ci-dessous).

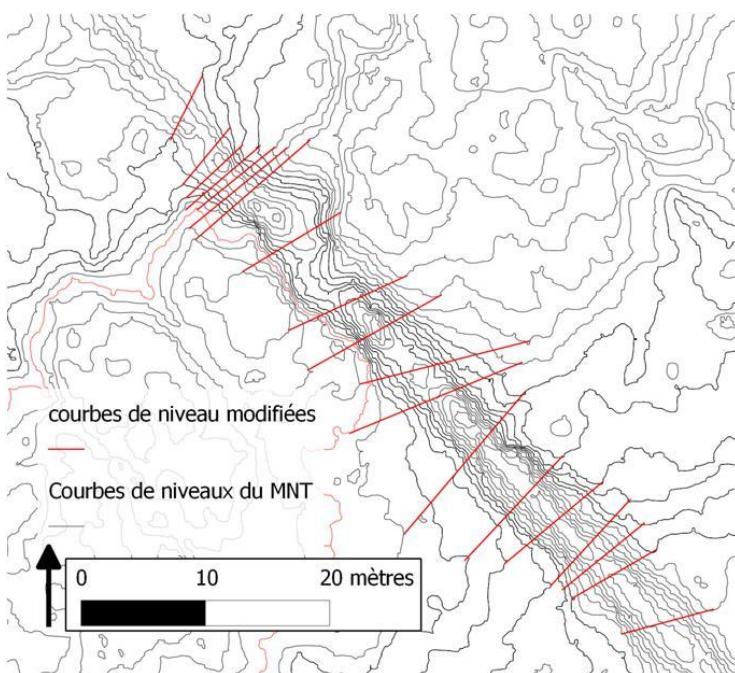


Figure 10: exemple de modification des courbes de niveau comme étape dans la modification de la topographie d'un MNT.

Puis on rasterise les nouvelles courbes de niveau en donnant aux pixels les valeurs d'altitudes.

Ensuite on interpole le reste de la surface à partir de ces courbes de niveau rasterisées en utilisant un masque qui délimitera exactement la zone du MNT que l'on veut modifier.

Enfin on remplace sur le MNT les zones que l'on a modifiées.

Le résultat sera toujours un peu lisse mais la finesse dépend du dénivelé choisi entre les courbes de niveau.

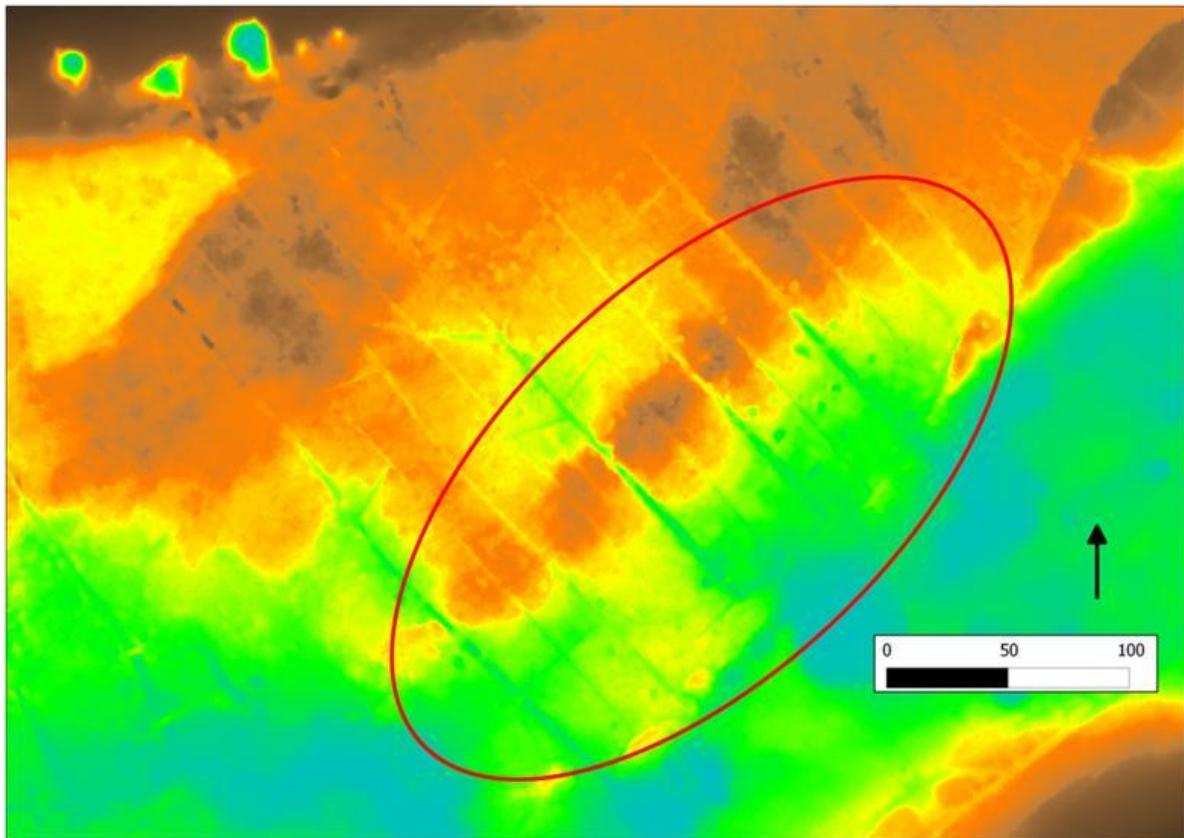


Figure 11: MNT avant modification.

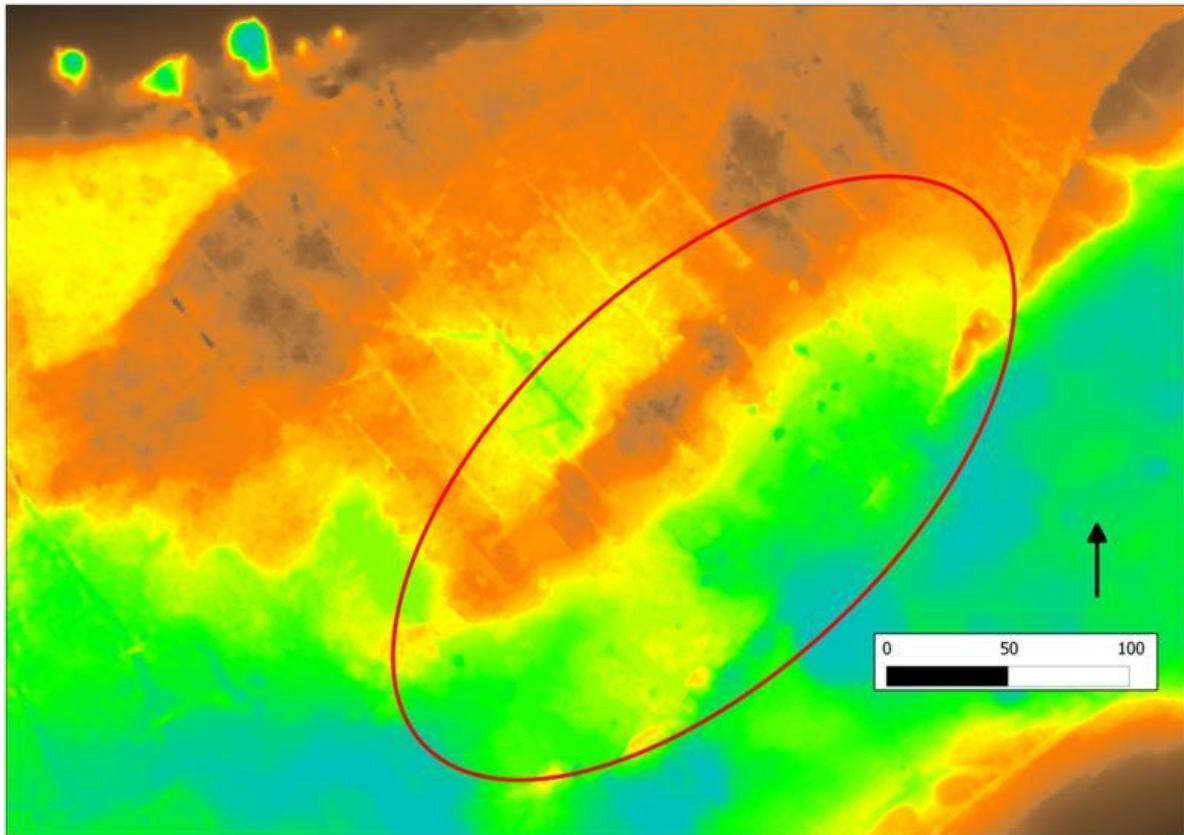


Figure 12: MNT après modification, simulation du remplissage des fossés de drainage dans la zones entourée en rouge.

J'ai aussi utiliser cette méthode pour calculer des volumes de remplissages dans un autre projet où l'on veut remonter le niveau de la nappe d'eau dans une tourbière en créant des barrages sur les drains ou en les remplissant suivant la topographie (voir le chapitre I : Ebauche de projet pour lutter contre l'assèchement et l'érosion d'une tourbière).

Deuxième méthode

Lorsque l'on veut faire des modifications plus simples j'ai aussi développé deux méthodes très proches et écris leur script pour en faciliter l'utilisation.

Les deux méthodes consistent à modifier les valeurs du MNT sur la surface d'un vecteur dessiné par l'utilisateur.

La première méthode consiste à rasteriser le vecteur en lui donnant une valeur d'altitude qui remplacera les valeurs du MNT sur la surface dessinée.

La seconde méthode consiste aussi à rasteriser le vecteur mais en lui donnant une valeur que l'on additionnera aux valeurs du MNT sur la surface dessinée. Cela creusera le MNT si la valeur est négative ou le surélèvera si la valeur est positive. C'est une méthode très rapide pour creuser un canal ou créer une surélévation qui suit les variations du terrain.

Voir "[D.modifAltMNTviaVecteur.py](#)" et "[D.EpaissiCreuseMNTviaVecteur.py](#)" dans "réalisation_concrète.zip".

E. Evolution de la topographie d'une tourbière.

Objectif :

Pouvoir comparer la topographie mesurée en 1984 avec la topographie en 2011, pour évaluer l'affaissement de la tourbière lié à l'assèchement et pour évaluer l'érosion causée par un canal de drainage dans la tourbe.

Données disponibles :

Deux vecteurs des courbes de niveau de 1984, l'un avec les courbes de niveau tous les 5 mètres de dénivelé, et l'autre qui le complète pour obtenir une courbe de niveau tous les mètres.

Un raster de modèle numérique de terrain (MNT) obtenu par LIDAR en 2011.

Travail effectué :

J'ai répondu à cette demande de deux manières :

- en créant les courbes de niveau à partir du MNT de 2011 (r.contour), en les comparant avec les courbes de niveau de 1984. On a un aperçu rapide de l'évolution de la topographie.
- en interpolant un MNT à partir des courbes de niveau de 1984 et en créant un raster représentant la différence entre les altitudes 2011 et ceux interpolées de 1984.

Pour cela j'ai combiné les deux vecteurs des courbes de niveau de 1984 (v.patch) puis j'ai converti le vecteur obtenu en raster (v.to.rast) à partir duquel j'ai fait interpoler les valeurs d'altitudes pour la surface étudiée (r.surf.contour). De cette manière on obtient un MNT de 1984.

Pour obtenir un raster représentant la différence entre 2011 et 1984, je soustraie les valeurs de 1984 à celles de 2011 (r.mapcalc).

Résultats :

L'affaissement de la tourbière lié à l'assèchement est bien observable sur le raster représentant la différence d'altitude entre 2011 et 1984 mais l'érosion causée par un canal de drainage dans la tourbe n'est pas observable car les relevés topographiques effectués pour créer les courbes de niveau en 1984 n'ont pas pris en compte les drains.

F. Créer une carte raster représentant la topographie du sous-sol sous une couche de tourbes

Objectif :

Créer une carte raster représentant la topographie du sous-sol sous une couche de tourbe.

Données disponibles :

- Un MNT représentant l'altitude de surface de la tourbe
- Un raster dont les valeurs représentent l'épaisseur de tourbe, **les valeurs sont négatives** car obtenues par géoradar déplacé sur la surface de la tourbière.

Travail fait :

En observant le raster d'épaisseur de tourbe j'ai remarqué des valeurs positives ce qui est incohérent vu la méthode pour obtenir les données. J'ai corrigé cela en transformant les valeurs positives en NULL (r.nul).

J'ai ensuite additionné le MNT au raster d'épaisseur de tourbe corrigé (r.mapcalc) pour obtenir le raster représentant la topographie du sous-sol minéral.

Sur cette couche on retrouve les drains, c'est un artefact dû à la méthode de déplacement du géoradar qui n'allait pas au fond des drains mais passait dessus. Il y a donc une surestimation de l'épaisseur de tourbe au niveau des drains qui est reportée à la profondeur du sous sol minéral (voir "E.Carte_sous-sol_mineral.png" dans réalisation concrète).

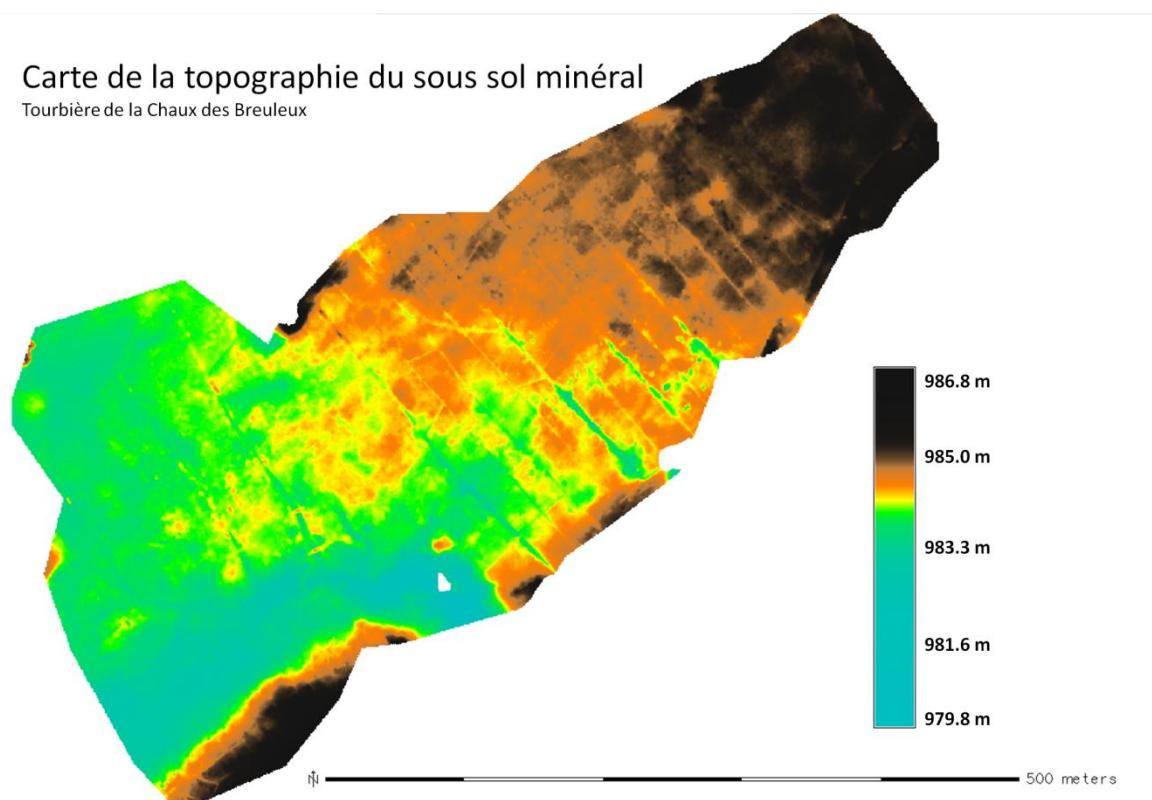


Figure 13: Carte de la topographie du sous sol minéral d'une partie de la tourbière de la Chaux des Breuleux dans le Jura Bernois.

G. Numérisation d'un relevé topographique

Objectif :

Obtenir un MNT à partir d'un relevé topographique.

Données disponibles :

Notes et schémas du relevé topographique.

Toute la méthodologie m'a été fournie par Philippe Grosvernier.

Travail effectué :

Participation au relevé topographique à l'aide d'une simple lunette de visée, j'ai tenu la mire.

Changer les valeurs pour qu'elles se réfèrent au point d'altitude de référence.

Numérisation sous QGIS des points relevés et interpolation d'un MNT très simple:

J'ai défini le système de coordonnées de référence du projet Qgis pour qu'il soit le même que celui du secteur GRASS dans lequel se trouve les couches sur lesquelles je me base pour placer les points.

J'ai placé les points qui pouvaient être repérables sur le MNT, puis avec les notes prises sur le terrain et les outils de distance et d'angle de QGIS j'ai placé les lignes des transepts relevés puis les points.

J'ai ensuite rempli la table attributaire avec les altitudes relatives au point de référence.

H. Création d'un outil de profils dont les données sont facilement exportables dans Excel

Objectif :

Création d'un outil qui génère des profils sur des rasters dont les valeurs représentent des altitudes ou des épaisseurs et qui permet d'exporter facilement les données dans Excel.

Travail effectué :

Etant donné que l'on travaille beaucoup avec des vecteurs lignes qui représente des digues ou la visualisation des profils sur la carte, j'ai écrit le script qui fait un profil à partir d'un vecteur ligne.

Le script crée un vecteur point dont l'espacement maximal entre les points est choisi par l'utilisateur.

Puis il copie dans un fichier les valeurs du raster à l'emplacement des points et dans un autre fichier la distance à partir du premier point.

Puis, suivant l'option choisie, il ouvrir une fenêtre Tkinter affichant le graphique du profil ou/et ouvre Excel avec les données du profil.

Voir "[H.outil_profiles.py](#)" dans réalisation concrète".

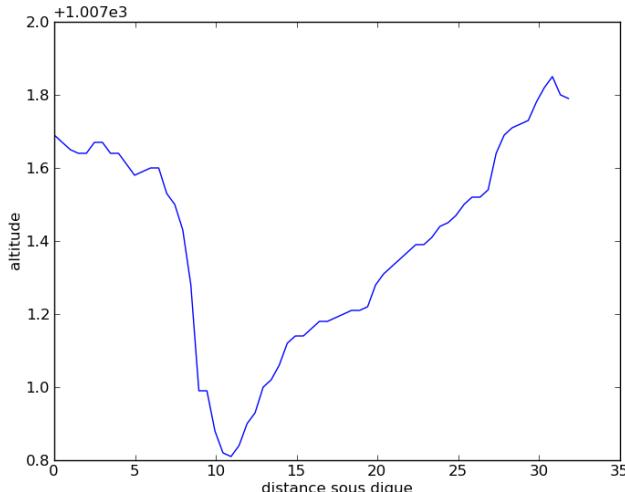


Figure 14 : Exemple d'un graphique d'un profil produit par l'outil.

I. Esquisse de projet pour lutter contre l'assèchement et l'érosion d'une tourbière

Objectif :

Choisir les bons emplacements et définir le type d'aménagement à réaliser pour lutter contre l'assèchement et l'érosion d'une tourbière causés par un réseau de fossés de drainage.

Travail effectué :

Tout le long du stage, Philippe Grosvernier m'a énormément appris sur le fonctionnement des tourbières et j'ai aussi lu sur le sujet. Cela a été la première étape du travail avant même d'avoir ce travail à faire.

Ensuite j'ai travaillé à comprendre le fonctionnement hydrologique actuel de la tourbière puis le fonctionnement "idéal".

L'idée est de contrer les effets des fossés de drainage, c'est à dire l'accélération de la sortie de l'eau du système, la baisse du niveau de la nappe d'eau et de l'augmentation de l'amplitude du battement de la nappe d'eau.

Pour cela on peut créer des barrages qui bloquent l'écoulement de l'eau dans les fossés et créent un plan d'eau ou on peut combler les fossés, ce qui annule leur fonctionnement.

Les barrages sont proposés principalement dans la partie haute de la tourbière, là où la pente est faible.

Les autres sont proposés lorsque le terrain a un faible dénivelé aux limites de l'inondation créée, permettant ainsi de remonter le niveau de la nappe près de la surface, de rendre diffus l'écoulement de l'eau et de l'orienter vers les zones souhaitées.

Lorsque les fossés sont trop encaissés dans le terrain ou/et la pente trop importante, les remplissages sont proposés.

Pour les barrages j'ai utilisé l'outil de simulation des digues puis un outil très proche dans lequel la digue est un vecteur que l'on a dessiné, et l'outil servant à faire des profils.

Pour les remplissages, j'ai utilisé la méthode de modification de la topographie.

Pour obtenir les volumes j'ai créé une couche résultant de la soustraction du MNT original au MNT modifié avec "r.mapcalc".

Puis avec l'outil "r.sum" j'obtiens la somme des valeurs des pixels, ce qui me donne la différence de volume en mètres carrés si la résolution des pixels est de 1 mètre sur 1 mètre, sinon il faut diviser le nombre obtenu par le nombre de pixels par mètre carré.

Par exemple si la résolution est de 0.5 m sur 0.5 m il faut diviser le nombre par 4.

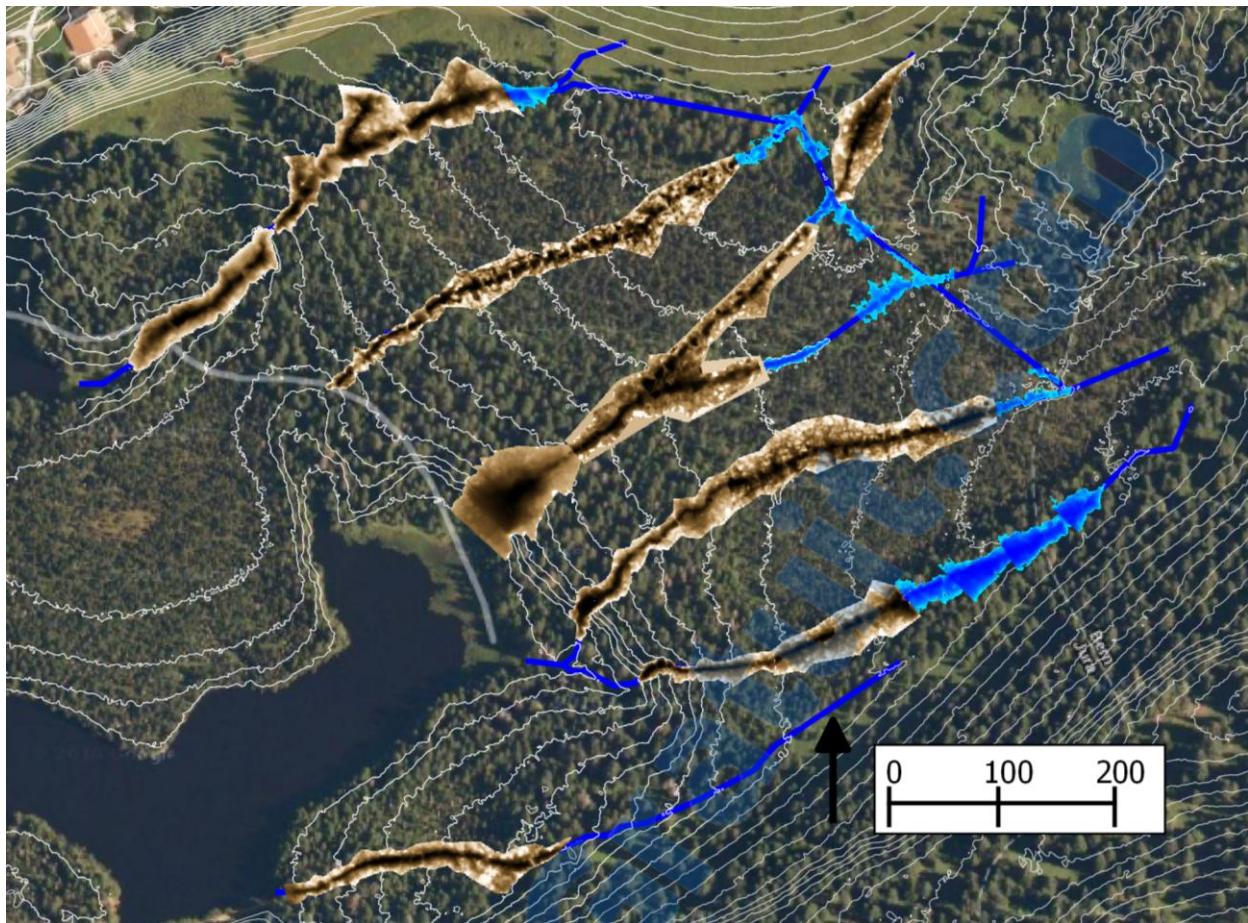


Figure 15: Vue d'ensemble des mesures proposées, les traits bleus foncés sont les drains existant, les surfaces bleues sont les inondations proposées et les surfaces sépias sont les propositions de remplissages des drains.
Les courbes de niveaux, représentées en blancs, sont tous les 1m de dénivélées.

Voir " I. Esquisse de projet pour lutter contre l'assèchement et l'érosion d'une tourbière.pdf" dans "réalisation_concrète.zip".

III. Les logiciels utilisés

Les logiciels utilisés pendant le stage sont des logiciels open source ce qui signifie que leur code est disponible, que tout le monde peut apporter une contribution à leur amélioration et qu'ils sont libres de redistribution.



GRASS GIS : "Geographic Resources Analysis Support System" est un Système d'information géographique qui propose un moteur de traitement raster, vecteur, et traitement géospatial regroupé dans un logiciel. GRASS intègre des outils pour la modélisation spatiale, la visualisation de rasters et de vecteurs, la gestion et l'analyse de données géospatiales, et le traitement de données d'images satellites et images aériennes. Il est traduit dans 20 langues et supporte un très grand nombre de formats de données.

Le logiciel GRASS intègre nativement plus de 400 modules d'analyses. De plus, plus de 100 modules et outils fournis par la communauté sont disponibles gratuitement à partir du site wiki de la communauté. Après près de 30 ans de développements continus, GRASS est parmi les plus vieux et plus gros projets SIG libres disponibles. Il est facilement scriptable et assez polyvalent pour être utilisé tant pour traiter des données par lots sur des supercalculateurs massivement parallélisés que pour être utilisé sur des SIG de bureau.



A l'origine Quantum GIS était destiné à n'être qu'un outil de visualisation des données de GRASS (Geographic Resources Analysis Support System). Aujourd'hui, ce SIG généraliste peut lire et modifier des données géographiques, faire des analyses thématiques simples et les mettre en page avec "Map composer" (logiciel de mise en page intégré).

Moins axé sur l'analyse que GRASS, le nombre de ses fonctionnalités en font un bon outil de SIG bureautique, proche de la plupart des standards payants et d'autant plus intéressant qu'il est compatible en lecture et en écriture avec de nombreux formats "classiques" de données géographiques. Il est aussi possible de se connecter à des serveurs de données type WMS ou WFS.

Discussion

Initialement Philippe Grosvernier voulait un outil montrant en direct la largeur de la digue et l'étendue de l'inondation lorsque l'on déplace un curseur sur une droite qui serait placée sur un drain. Cela pose des problèmes. Les principaux sont, le temps de calcul et la définition du niveau de l'inondation. En effet pour définir la longueur d'une digue il faut le niveau de l'inondation voulue qui doit être choisie par l'utilisateur et qui si l'on se déplace devra varier car les terrains sont souvent en pente. L'interactivité était aussi un gros travail (cf. annexe 1). Evidemment des solutions sont possibles mais elles nécessiteraient un temps de travail trop important pour le stage et pour les bénéfices qu'apporterait l'outil à l'entreprise. Après réflexion et discussion avec Philippe Grosvernier j'ai proposé l'outil présenté dans ce rapport qui est un bon compromis et répond en plus à la problématique de l'orientation de la digue.

Conclusion

Les logiciels SIG propriétaire et open source offrent un panel impressionnant d'outils et de méthodes utilisables, ce que l'on attendait de moi était d'en faciliter l'utilisation et de proposer des solutions à des besoins particuliers en utilisant uniquement des logiciels open source.

Une partie des méthodes développées pendant ce stage est utilisable avec les deux types de logiciels mais les logiciels open source permettent une plus grande plasticité grâce à la possibilité d'accéder au code et à la profusion de librairies utilisables pour la création d'outils. Cela est très appréciable lorsque l'on cherche à obtenir des outils répondant de la manière la plus personnalisée possible aux besoins d'une entreprise.

Les logiciels open sources on permit de répondre aux besoins de l'entreprise.

Pour ma part le stage s'est déroulé à merveille, j'ai été très bien accueilli par Philippe Grosvernier et Elisabeth Contesse et ils se sont toujours montrés disponibles.

Pendant la formation du certificat complémentaire en géomatique et le stage à LIN'eco j'ai énormément appris dans les domaines de la programmation, de la géomatique, de la restauration des milieux humides et du secteur de l'environnement en Suisse. Il me reste encore beaucoup à apprendre et à découvrir dans ces mêmes domaines et cela me stimule.

Bibliographie

Manneville O., Vergne V., Villepoux O. ;1999 ; le monde des tourbières et des marais ; ed. Delachaux et Niestlé

Grosvernier Ph. & Staubli P. ; (réd.) 2009: Régénération des hauts-marais. Bases et mesures techniques.

L'environnement pratique n° 0918. Office fédéral de l'environnement, Berne. 96 p.

Rapinel S. ;2012 ; Contribution de la télédétection à l'évaluation fonctionnelle des zones humides : de l'observation à la modélisation prospective ; thèse Université Rennes 2

"Apprendre à programmer avec Python" de Gérard Swinnen :

<http://python.developpez.com/cours/TutoSwinnen/>

Librairie Python Tkinter :

<http://docs.python.org/2/library/tkinter.html>

http://fr.wikibooks.org/wiki/Programmation_Python/Tkinter

Librairie wxpython :

<http://www.wxpython.org/>

Laloux M. ; Utilisation des fonctions Gdal avec python :

<http://www.portalsig.org/content/python-utilisation-des-couches-vectorielles-et-matricielles-dans-une-perspective-geologique->

Tutoriaux pour GRASS :

http://grass.fbk.eu/gdp/grass5tutor/HTML_fr/book1.html

http://grass.fbk.eu/gdp/grass5tutor/grass50_tutorial_fr.pdf

Problèmes GRASS connus :

http://grasswiki.osgeo.org/wiki/WinGRASS_Current_Status

Harmel A. ;"Le nouveau système réglementaire Lambert 93" Géomatique Expert - N° 68 - Avril-Mai 2009

www.geomag.fr/sites/default/files/68_91.pdf

Laloux M. ;Utilisation de GDAL pour obtenir diverses informations sur un raster ;mars 2013

"<http://www.portalsig.org/content/python-utilisation-des-couches-vectorielles-et-matricielles-dans-une-perspective-geologique->"

Annexe 1

Problèmes GRASS 4 connus :

http://grasswiki.osgeo.org/wiki/WinGRASS_Current_Status

Extrait de la page , chapitre 4 " **Known Issues** " :

Won't fix (at least not immediately)

Issues listed here are rather impossible to fix due to the different nature of native Windows. Or, lend us a hand and let's try harder!

- No monitors

This means that you cannot launch any monitor launched with d.mon (x0, PNG, PS, etc). The only way you can render is directly to a file. But you cannot directly display to screen from the command line. This will be solved in GRASS 7 with a new rendering system and possibly via the new wxgrass GUI. So no work will probably be put into this until then.

The absence of monitors also makes impossible the use of interactive modules based on these monitors, such as: