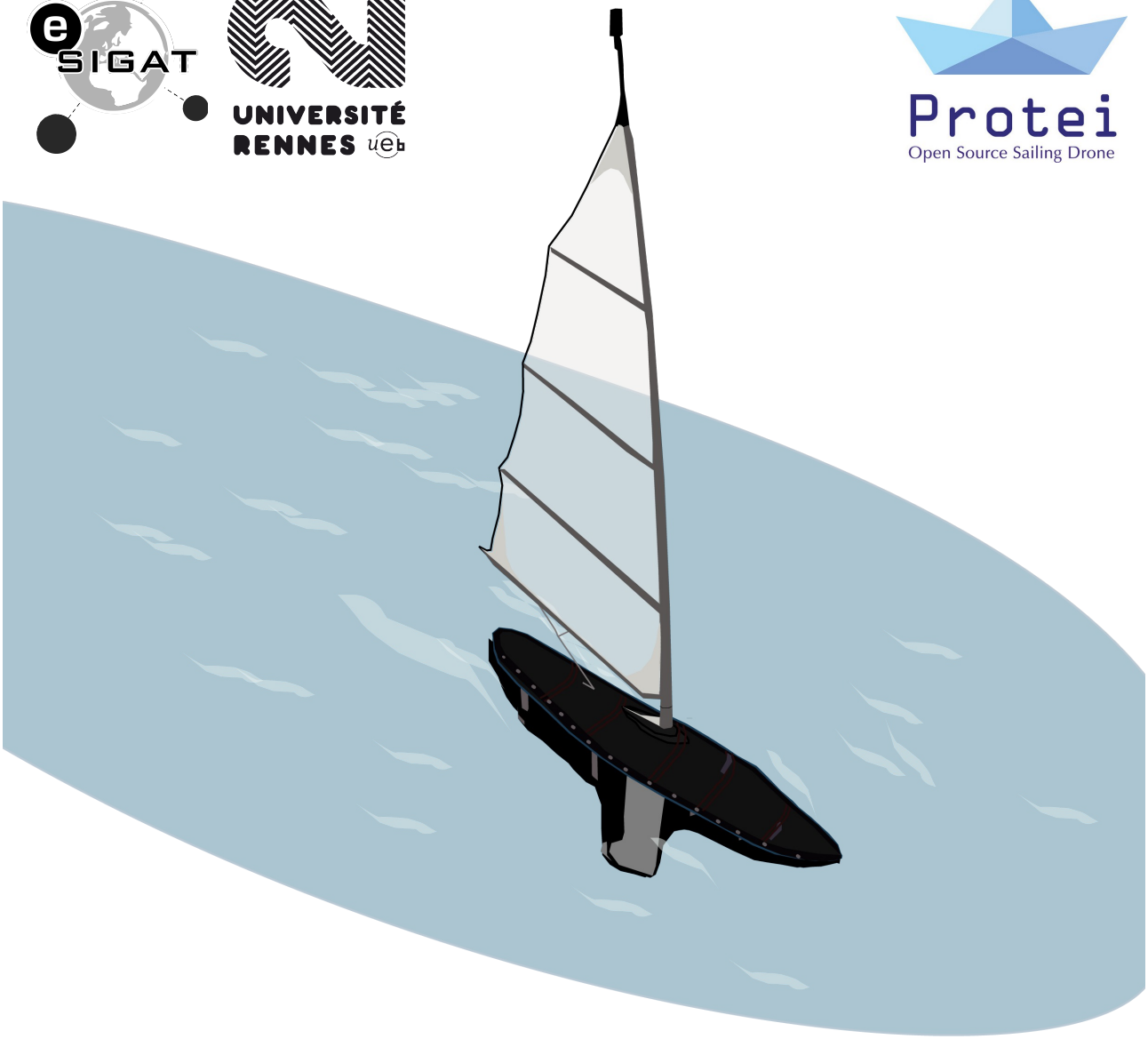


Élaboration de stratégies pour le déploiement de drones marins dans le cadre de missions environnementales

- Rapport d'étude -

Baptiste BODIN
Hugo CARRE
Guillaume PAYSANT
Enzo VENTURINI
Eric VINOUZE

Master 2 SIGAT | 2013-2014
Université Rennes 2

Résumé

Ce travail mené dans le cadre du partenariat SIGAT-Protei a pour objectif de contribuer au projet Protei afin de lui apporter une dimension spatiale avant, pendant et après les missions de nettoyage des zones maritimes polluées. Au travers des axes de conception et de développement, il vise avant tout à proposer différents outils et méthodologies issus de la géomatique pour la mise en place de stratégies de déploiement de drones marins dans le cadre de missions environnementales.

Quatre modules ont ainsi été développés pour répondre à ces enjeux : un module de couverture de zone pour générer des itinéraires de navigation théoriques, un module de préparation à la navigation afin de préparer un parcours optimal en amont du déploiement, un module de détection de pollution à partir de données relevées sur le terrain, et enfin un module de déploiement de flotte qui s'appuie sur les trois premiers modules afin de déployer stratégiquement plusieurs drones à la fois.

Ce travail s'inscrit dans le cadre de la formation Master 2 SIGAT (Systèmes d'Information Géographique pour l'Aménagement des Territoires) de l'université Rennes 2 et s'appuie sur une approche expérimentale à partir de scénarios de simulation. Protei se basant sur le concept d'Open Hardware, les outils développés sont Open Source et donc réutilisables.

Mots clés : géomatique, drone, mer, pollution, détection, bathymétrie, polaire des vents, analyse multicritères, flotte, analyse spatiale, géostatistiques, python, QGIS, PHP

Table des matières

I. Introduction

Au cours des dernières décennies, l'activité humaine s'est largement étendue sur les océans sous différentes formes (implantations de plate-formes pétrolières ou de parcs éoliens, intensification des flux maritimes) qui ont pour conséquence de nuire à cet écosystème fragile. L'augmentation de toutes ces activités accroît donc les risques de catastrophes et les rejets de déchets : dégazage de navires, fuite de matières toxiques (pétrole, fioul, méthane, etc.), explosion de plate-formes pétrolières, rejets de plastique, perte de conteneurs, etc.

Malgré l'immensité des océans, le système hydrique mondial forme un système interconnecté. Les catastrophes anthropiques n'ont pas que des conséquences au niveau local, mais bien global. La maîtrise d'une catastrophe demande donc des moyens importants qui, dans les faits, sont souvent insuffisants. Au delà de l'impact sur le milieu, les sinistres - quand ils sont médiatisés - entraînent une prise de conscience des populations qui se transforme bien souvent en sentiment d'impuissance, faute de capacité des individus à agir. Ce constat a amené Cesar Harada, le commanditaire de notre étude, à se lancer dans un projet ambitieux. Selon lui, si l'homme est capable de causer ces catastrophes, c'est aussi à lui de se charger de résoudre les problèmes.

Suite à l'explosion de la plate-forme pétrolière Deepwater Horizon dans le Golfe du Mexique, il décide de se lancer dans le projet Protei qui a pour objectif de fabriquer un drone marin capable de nettoyer les mers. Ce drone, équipé de voiles, a la particularité de pouvoir se déplacer face au vent grâce à une coque articulée. Il est donc respectueux de l'environnement et s'appuie sur les technologies biomimétiques pour se déplacer plus efficacement.

En outre, il est conçu de façon modulaire. Ainsi, plusieurs drones peuvent être mis bout à bout pour répondre à des fonctionnalités propres : géolocalisation, collecte de données environnementales, prises de photos/vidéo, traction d'un boudin de captation de pollution etc.

Enfin, ce projet s'inscrit dans une démarche Open Source/Open Hardware. Au travers d'une communauté mondiale de chercheurs, d'ingénieurs et de navigateurs, l'ensemble des travaux sont partagés et réutilisables par tous pour faire avancer le projet. Celui-ci fait donc appel à différentes thématiques où la géographie a toute sa place.

C'est cette philosophie dans laquelle s'est inscrite le partenariat entre le Master SIGAT et le projet Protei. Dans le but de détecter et de nettoyer des pollutions marines, nous avons cherché à trouver des méthodes et des outils pour être en mesure de déployer efficacement des drones sur une zone d'intervention.

II. Présentation du projet SIGAT-Protei

La collaboration du master SIGAT avec Protei nous a amené à une réflexion sur la place de la géomatique dans un tel projet. Plus largement les questionnements ont porté sur l'approche géographique dans le cadre de développement de technologies navales et environnementales. A travers la dimension spatiale, la géographie met en évidence les relations entres les entités spatiales en faisant apparaître des systèmes intelligibles. La géomatique par l'informatique apporte une puissance de calcul permettant d'effectuer des analyses complexes et de passer de la temporalité prospective à une temporalité effective. Le développement de programmes automatisés permet de rendre accessible des analyses prédéfinies à des utilisateurs non initiés.

Après un riche travail de problématisation du projet, notre axe de réflexion s'est finalement fixé sur le développement de modules automatisés centrés autour de la notion de "stratégies spatiales pour le déploiement de drones marins". Qu'il s'agisse de proposer des stratégies ou d'aider à les concevoir, la géomatique offre une approche transversale à toutes les problématiques pouvant être rencontrées.

Les modules proposés ici ont vocation à s'intégrer en amont du déploiement des drones puis entre les phases de remontée d'information issues de capteurs et de diffusion auprès des utilisateurs. Le fil conducteur de notre travail a été pensé autour de la temporalité d'une mission Protei. Cela commence par l'étude de la zone d'étude puis son appréhension et enfin sa lecture afin quelle puisse être traitée efficacement.

Deux points d'ordre méthodologique ont été transversaux à notre travail. D'une part, il y a une volonté d'interopérabilité entre les modules développés ici mais également avec d'autres outils qui s'intégreraient dans le contexte du projet. Cette interopérabilité se matérialise par l'utilisation des standards OGC concernant le partage de l'information géographique. D'autres parts les outils ont été pensés de telle manière que l'utilisateur reste le dernier décideur. Il ne s'agit pas de tout automatiser mais plutôt d'offrir des outils d'aide à l'interprétation et à la décision.

A. Se déplacer dans un espace connu a priori

La globalisation des flux maritimes et l'urbanisation des littoraux font qu'une grande partie du globe est susceptible d'être victime d'une pollution. De plus certaines zones sont extrêmement vulnérables car très fréquentées (exemple : Mer Manche, une des plus fréquentée du globe) ou concentrent les activités polluantes comme les plate-formes pétrolières. Le but du projet Protei est de pouvoir intervenir dans n'importe quelle zone marine rapidement. Les drones seront donc amenés à se déplacer dans des environnements très différents.

Dans ce contexte, la préparation de missions, notamment avec l'utilisation de drones, nécessite une bonne connaissance du terrain a priori. En effet, les pilotes de drones ne peuvent pas appréhender de leur propre yeux la zone et doivent se fier aux détecteurs embarqués.

Cependant la capacité d'embarquement du drone et les possibilités technologiques ne permettent pas de décider uniquement sur la base de ces outils. Il est pourtant indispensable de parer aux risques liés à la navigation tels que les collisions. Des connaissances sur les conditions météorologiques, notamment à l'échelle locale, sont nécessaires pour adapter au mieux les déplacements des drones. La géomatique permet de mettre en relation toute ces variables acquises à partir de différentes base de données pour les synthétiser et proposer un chemin idéal de navigation. Cette problématique a été travaillée dans le projet SIGAT-Protei au travers du module "Préparation à la navigation".

B. Couvrir une zone efficacement

Selon les cas, les nappes de pollutions peuvent se déplacer rapidement et sont parfois difficiles à localiser précisément à un instant T. Une mauvaise coordination des missions de nettoyage peut faire perdre du temps et il est donc important de suivre une logique de moindre coût en termes de couverture

spatiale. Le projet Protei a pour but de traiter de manière rapide et efficace des espaces pollués.

Il est tout d'abord important d'estimer les besoins en drones sur chaque mission en connaissant les capacités d'un seul mais également le nombre de drones nécessaires selon la manière dont ils sont coordonnés. En effet, pour compenser la capacité de nettoyage et d'observation limitée d'un drone, plusieurs peuvent être utilisés sous la forme d'une flotte sur une même zone. Néanmoins, la coordination des flottes nécessite que l'espace soit subdivisé de manière cohérente. La géomatique permet de répondre à cette problématique de partage de l'espace par des calculs géostatistiques et géométriques.

D'autre part, elle permet d'identifier des chemins théoriques de déplacements des drones en vue d'une couverture optimale. Les modules nommés "Couverture de zone" et "déploiement de flotte" se sont penchés sur cette problématique.

C. Comprendre un environnement inconnu en l'explorant

Si les catastrophes environnementales importantes mobilisent des moyens conséquents de surveillance, un grand nombre d'événements mineurs, comme le dégazage de navires, restent ignorés ou connus tardivement. Quelle que soit l'importance de la catastrophe, le suivi temporel et la connaissance fine à haute résolution pose des problèmes. Protei développe des drones amenés à être présents en continu sur les zones de pollution, y compris celles plus mineures où le suivi par télédétection est parfois trop onéreux, ou celle ayant une mauvaise ou une absence de couverture aérienne ou satellitaire à cause de la résolution temporelle et de la mauvaise qualité de prise de vue par temps nuageux. Grâce à ses détecteurs, un drone Protei peut appréhender l'environnement où il se trouve pour comprendre l'ampleur, la dynamique et la cause d'une pollution. La difficulté réside dans le fait que ces détecteurs embarquent des technologies avec une portée limitée qui leur donne ainsi une vision "Bird eye".

La géomatique, au travers de méthodologies de géostatistiques, permet de répondre à cette problématique en offrant des outils d'interprétation et d'évaluation des données terrain. Par la mise en relation de données spatio-temporelles provenant de plusieurs drones, il est possible d'avoir une vision globale de l'espace qu'un drone ne peut avoir à lui seul pour couvrir l'ensemble de la zone. Cette connaissance permet ainsi de réagir plus efficacement et d'apporter des connaissances plus fiables au grand public sur l'ampleur d'une catastrophe. Le module "détection" s'est focalisé sur cette problématique.

III. Concepts-clés

L'homme se repère depuis toujours de façon relative, c'est-à-dire par rapport à des points de repère existants et connus sur le territoire. Il est dit par exemple : « j'habite derrière le cinéma », où le repère se fait à partir d'un lieu connu et environnant, comme dans l'exemple cité. Rapidement, ce moyen de se repérer a ses limites lorsqu'il s'agit de se repérer de façon absolue, autrement dit sans se préoccuper de l'environnement extérieur. Ce sont les coordonnées géographiques qui permettent de se repérer dans l'espace, qu'importe sa localisation sur terre. Ce repérage est très précis mais il implique des notions fondamentales pour bien être appréhendé.

Se repérer à la surface d'une sphère est une tâche relativement aisée, mais la Terre est loin d'être une sphère, mais a plutôt la forme d'un géoïde. En d'autres termes, elle a une forme légèrement écrasée sur les pôles et avec de très grosses déformations à sa surface.

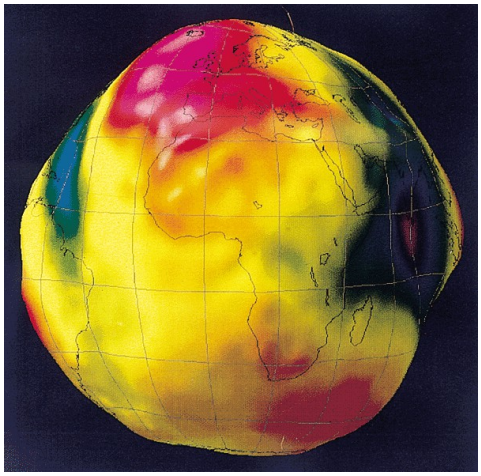


Illustration 1: Représentation symbolique d'un géoïde

La difficulté était alors de mettre en place un système afin de se repérer sur une forme complexe. Il a donc été nécessaire de créer un modèle de la Terre servant de base à la mise en place des coordonnées géographiques : l'ellipsoïde. Il en existe plusieurs types dont chacun possède un système de coordonnées distinct. Certains sont globaux c'est-à-dire qu'ils couvrent l'intégralité de la Terre, tandis que d'autres sont locaux et ne couvrent qu'une zone définie. Ces derniers sont logiquement plus précis mais leur caractère non global rend leur utilisation plus complexe. L'ellipsoïde global WGS84 est le plus utilisé, notamment par le biais du système GPS dont le système de coordonnées porte le même nom.

Les coordonnées sont des mesures d'angles exprimées en degrés. La longitude est l'angle entre le méridien principal à Greenwich et le point voulu par rapport au centre de la terre. La latitude est l'angle formé entre l'équateur et le point voulu par rapport au centre de la terre.

À ce stade tout ce qui a été évoqué relève d'une réflexion en 3 dimensions et ne peut pas être appliqué au cas de cartes papier. En effet, pour passer à une cartographie en 2 dimensions, il faut projeter l'ellipsoïde mobilisé sur une surface plane. Cette projection permet de représenter la Terre sur une carte mais induit des déformations. On peut relever 3 types de projections : les projections dites "conformes" qui déforment les angles mais conservent les distances, les projections "équivalentes", qui conservent les angles mais déforment les longueurs, ou encore les projections aphylactiques qui ne conservent ni les angles ni les longueurs.

Dans le cadre du travail présenté ici, toutes les études ont été menées en prenant comme système de référence le Système de coordonnées WGS84 (code EPSG 4326), ou système GPS qui est un système de coordonnées non projeté exprimé en degrés décimaux. Cela implique un mode opératoire particulier : la distance entre deux points étant exprimée en degrés décimaux dans ce système, il est nécessaire de projeter le résultat des calculs pour obtenir une mesure en mètres, plus parlante.

Pour aller plus loin : *http://fr.wikipedia.org/wiki/Coordonn%C3%A9es_g%C3%A9ographiques*

IV.Module « Stratégies de couverture de zone »

A. Contexte et présentation générale

Le module de stratégie de couverture de zone a pour objectif de définir le trajet idéal d'un drone afin de couvrir une zone définie par l'utilisateur à partir de 4 coordonnées GPS. Cela va se traduire par la création d'un pattern de navigation qui est composé d'un fichier de coordonnées GPS dont certaines sont définies comme des points de passage obligatoires. Ce fichier contient les points de passage théoriques par lesquels le drone est censé passer pour couvrir la zone d'étude. Néanmoins, ce pattern de navigation ne prend pas en compte les obstacles ou les conditions de navigation.

Le pattern est donc un modèle idéal de navigation pour couvrir une zone dans sa globalité en étant le plus efficace possible. La prise en compte du vent et de la bathymétrie intervient dans *Module « Préparation à la navigation »*. Ce module intervient donc en amont de la chaîne de traitements qui vise à l'adapter à la réalité du terrain, mais le modèle théorique peut tout de même être tracé de façon optimale en suivant plusieurs paramètres. L'utilisateur peut choisir une forme générale qui doit s'adapter au mieux à la mission ou aux conditions de navigations particulières. Le chemin théorique généré peut ensuite être testé dans la chaîne de traitements globale afin d'obtenir le meilleur résultat en sortie.

Le module se présente sous la forme d'un script développé en PHP5 couplé à un formulaire écrit en HTML5.

Deux versions du module ont été développées :

- La première version permet à un pattern d'être orienté selon 4 directions différentes., c'est à dire selon l'axe Nord/Sud et l'axe Est/Ouest. Cette version du script est totalement fonctionnelle.
- La version 2 du script permet à l'utilisateur de dessiner une forme plus libre (parallélogramme au lieu d'un carré) ce qui induit une orientation à 360° du pattern. Cette seconde version n'est pas totalement fonctionnelle et devra donc être finalisée avant son utilisation.

B. Utilisation d'une bibliothèque de calcul géographique

Afin de calculer des distances entre deux coordonnées en WGS84, il faut utiliser une fonction de calcul comprise dans une bibliothèque de calculs géographiques. Deux fonctions principales sont disponibles pour calculer une distance, la formule *haversine* et la formule de *vincenty*. Ces formules ont pour objectif de transformer deux coordonnées en degrés décimaux en une longueur en mètres.

Pour la formule *haversine*, le calcul s'effectue avec comme postulat de base que la Terre est une sphère de 6 378 km de rayon. La distance est une approximation de la distance réelle obtenue en projetant les deux points sur un arc les reliant entre eux.

Or la Terre n'est pas une sphère car elle est plus aplatie sur les pôles. Cette différence est négligeable pour des distances calculées à proximité de l'équateur mais s'accroît en allant vers les pôles. Les drones Protei pouvant potentiellement être déployés à différentes latitudes, Il est préférable de choisir la formule de *Vincenty* qui prend comme paramètre supplémentaire l'aplatissement de la terre sur les pôles.

Bien que la formule *haversine* soit plus rapide en temps de calcul, le gain de précision est important avec l'utilisation de la formule de *Vincenty*.

C. Création d'un pattern de navigation - Version 1 (4 orientations possibles)

Module stratégie de couverture de zone - Version 1 (4 orientations possibles)

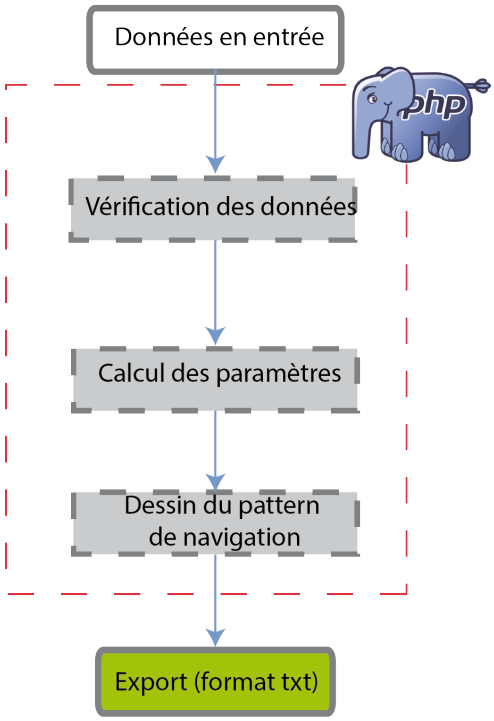


Illustration 2: Stratégies de couverture - chaîne de traitements modèle v1

i. Données en entrée

Les données en entrée du script sont récupérées au moyen d'un formulaire.

Ce formulaire comprend :

- Deux coordonnées en WGS84 (en degrés décimaux) : ces points définissent l'emprise du pattern de navigation. Cette emprise prend ici la forme d'un carré (deux points d'une diagonale).

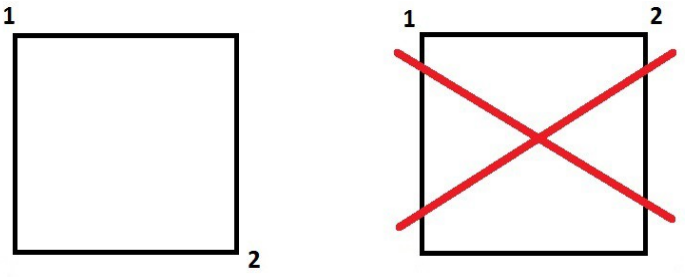


Illustration 3: Stratégies de couverture : définition de l'emprise

- Une forme de pattern : Deux formes sont disponibles, Zig Zag carré et une forme en Escargot.

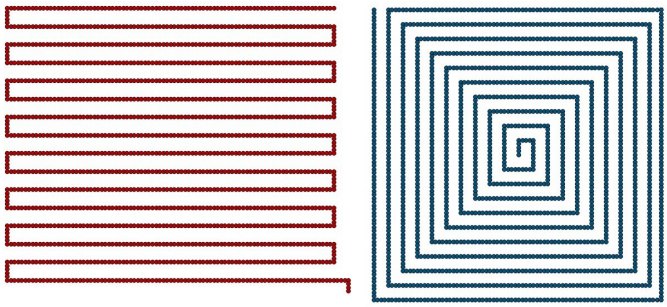


Illustration 4: Stratégies de couverture - Formes possibles de chemins théoriques

- Une orientation : celle-ci définit le sens dans lequel la zone est parcourue

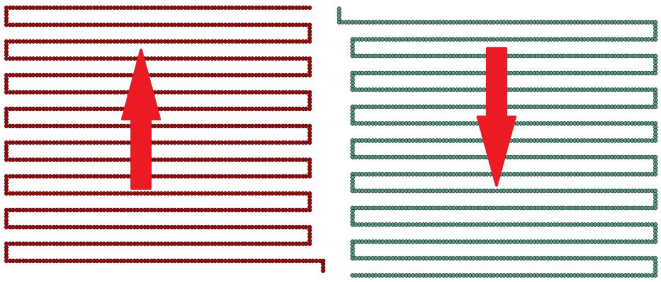


Illustration 5: Stratégies de couverture - Orientations possibles de déplacement

- Une taille de pattern : celle-ci est la longueur maximale en mètres entre deux passages du drone

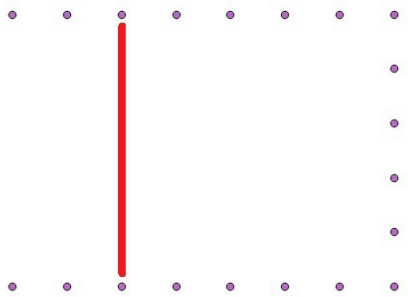


Illustration 6: Stratégies de couverture : taille de pattern

- Un intervalle de point qui définit une longueur en mètres entre la prise de 2 points GPS.

ii. Données en sortie

La sortie du script est un fichier texte comprenant l'ensemble des points GPS du pattern. Chaque point GPS contient les informations suivantes :

- un identifiant
- ses coordonnées X et Y en WGS84 en degrés décimaux
- un attribut égal à '1' si c'est un point de passage obligatoire

iii. Étape 1 : La récupération des informations issues du formulaire.

Cette étape permet de vérifier l'intégrité des données pour que le résultat obtenu soit conforme aux attentes de l'utilisateur. Cette partie aborde les problèmes pouvant être rencontrés par l'utilisateur qu'ils bloquent l'exécution du processus de traitement ou non.

Le script s'appuie sur un formulaire HTML qui envoie plusieurs paramètres au script. On retrouve notamment les 4 points GPS avec leurs coordonnées X et Y. Ces coordonnées doivent être au format décimal (champ texte pour X et pour Y). Le formulaire comprend aussi la forme du pattern (liste déroulante), la largeur du pattern (champ texte) et l'intervalle de point (champ texte) ainsi que l'orientation souhaitée (liste déroulante).

Après les vérifications d'usage comme le format du paramètre entré (texte ou numérique) il convient de vérifier que les coordonnées sont effectivement en WGS 84 et en degrés décimaux et qu'elles sont en cohérence avec l'emprise définie pour la mission du drone afin d'éviter toute erreur dans la transcription des coordonnées par l'utilisateur.

La première vérification concerne le format des coordonnées. Les coordonnées doivent être en degrés décimaux qui est l'unité utilisée par ce système de coordonnées. Des valeurs différentes signifient donc que l'utilisateur a commis une erreur de saisie ou une erreur dans le système de coordonnées. Un message vient alors le prévenir que le calcul du pattern est peut être erroné et donne des informations sur la nature de l'erreur si possible.

Par exemple si les coordonnées fournies vont jusqu'à des valeurs assez élevées, il peut s'agir d'une erreur due à l'utilisation de coordonnées projetées. Si au contraire le système de coordonnées semble être le bon mais que l'étendue semble trop importante alors il peut s'agir d'une erreur lors de la recopie des paramètres.

Pour connaître l'étendue de la zone, le script s'appuie sur la formule de Vincenty ce qui permet d'afficher le résultat à l'utilisateur pour l'informer de la nature de l'opération qu'il est entrain de réaliser. Cela peut lui permettre d'effectuer des corrections en cas d'incohérence avec la zone qu'il souhaite initialement traiter.

De ce fait, si des coordonnées erronées sont entrées en paramètres, le script ne s'exécutera pas et affichera un message d'erreur demandant à l'utilisateur de vérifier ses paramètres.

A l'inverse le script ne sera pas en mesure de détecter une erreur si l'utilisateur saisit les coordonnées dans un ordre différent de celui indiqué dans les paramètres en entrée.

iv. Étape 2 : La fonction de dessin du pattern

A ce stade de développement, deux formes de pattern théorique sont disponibles. Chacune des formes est une chaîne de traitements particulière construite avec le même modèle.

Le modèle consiste à calculer, avant d'initier le traitement, le nombre de mouvements que le drone devra effectuer dans une direction et de définir la direction suivante. Ensuite le script s'exécutera en boucle tant qu'il n'aura pas dépassé les limites de l'emprise.

Pour chaque mouvement effectué sur le pattern, le script fait appel à une fonction correspondante. Si l'utilisateur souhaite générer un pattern avec 4 mouvements vers le nord puis 10 mouvements vers l'Est, le script fera 4 appels à la fonction de mouvement vers le nord et 10 appels à la fonction de mouvement vers l'Est.

Afin de faciliter le traitement, l'intégralité du script s'appuie sur une grille dont le nombre de colonnes et de lignes est défini à la fois en fonction de la longueur de la zone et de l'intervalle défini. Chaque case est numérotée dans le sens de lecture.

Prenons par exemple le cas d'une grille de 10 par 10 en partant du point 5. Un déplacement vers l'ouest signifie donc un déplacement de la case de départ (5) vers une case située sur sa gauche (4). Un déplacement vers le sud signifie un changement de ligne, la case d'arrivée sera alors la case numéro 15.

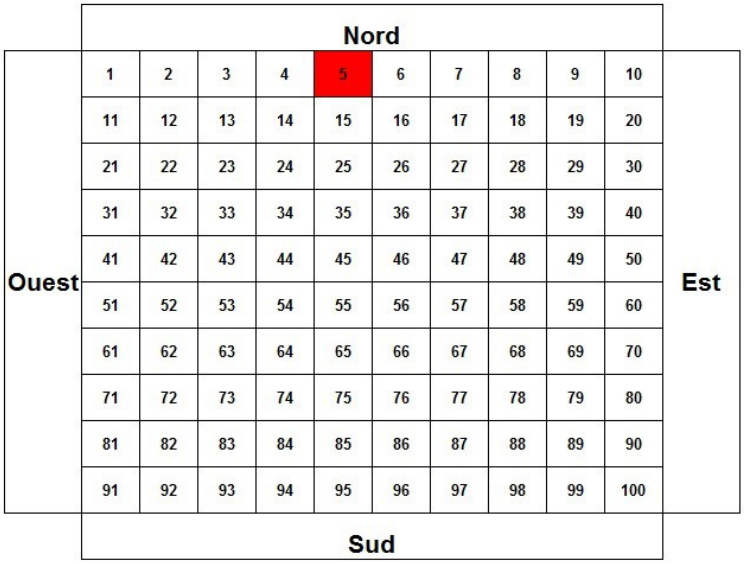


Illustration 7: Stratégies de couverture : exemple de grille

Chacune de ces fonctions est définie à l'image de l'exemple illustré ci-dessous :

```
function ouest(){
    global $case, $trajet, $mouvement, $x, $y, $x_array, $y_array, $pas_x_NS,
$pas_y_NS, $col_max, $pas_x_EO, $pas_y_EO;
    $case--;
    $trajet[$case]=$mouvement;
    $mouvement++;
    $x = $x-$pas_x_EO;
    $y = $y-$pas_y_EO;
    $x_array[$case]=$x ;
    $y_array[$case]=$y ;
}
```

Dans un premier temps tous les paramètres sont ajoutés dans la fonction. Ensuite les listes des coordonnées et du numéro de mouvement sont complétées avec les nouveaux paramètres. Les coordonnées de la position du drone sont alors modifiées suivant un pas Est-Ouest ou un pas Nord-Sud selon la direction du mouvement.

D. Création d'un pattern de navigation théorique - Version 2 (orientation à 360°)

La version 2 du module de stratégies de couverture de zone est une évolution de la première version. La plupart des étapes présentes dans la version 1 sont reprises dans la version 2. Ce rapport se concentre ici sur l'explicitation des étapes ajoutées.

L'objectif est ici de permettre à l'utilisateur de dessiner une emprise "plus libre". Cela implique d'effectuer des traitements spécifiques pour réadapter la forme de l'emprise à une forme gérée par le script (un parallélogramme) mais aussi d'intégrer une fonction capable de reconnaître la position des points dans l'espace.

Module stratégie de couverture de zone - Version 2 (Orientation à 360°)

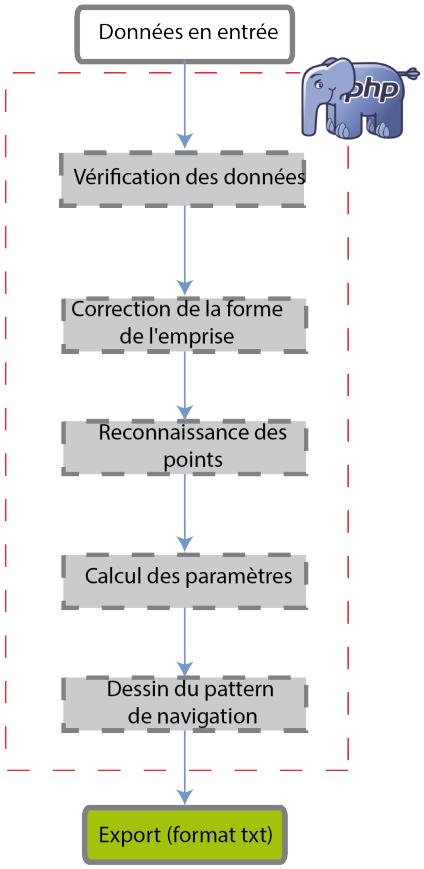


Illustration 8: Stratégies de couverture - chaîne de traitements modèle v2

i. Données en entrée

Les données en entrée sont les mêmes que pour la version 1, mis à part la nécessité d'entrer 4 points GPS au lieu de seulement 2 points au départ :

- 4 coordonnées GPS en WGS84 (degrés décimaux) : ces points définissent l'emprise du pattern de navigation. Cette emprise doit se rapprocher d'un parallélogramme dont les 4 points sont consécutifs.

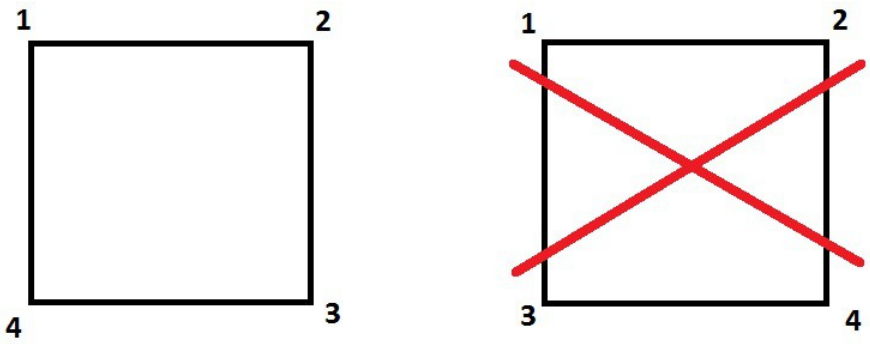


Illustration 9: Stratégies de couverture : emprise d'un parallélogramme

ii. Données en sortie

La sortie du script est un fichier texte comprenant l'ensemble des points GPS du pattern dont les éléments constitutifs sont identiques à la première version du modèle

iii. Étape 1 : La récupération des informations issues du formulaire.

La vérification est identique à la version 1 du modèle mais est ici appliquée au 4 points GPS demandés dans les paramètres.

iv. Étape 2 : La fonction de correction de la forme de l'emprise

Le script s'appuie sur une emprise dessinée à partir des quatre points GPS paramétrés par l'utilisateur. Afin de laisser le plus de liberté à l'utilisateur, le choix s'est porté sur la forme du parallélogramme.

Ce choix induit cependant des limites. En effet, cela demande à l'utilisateur d'entrer des coordonnées qui correspondent aux 4 coins d'un parallélogramme, ce qui peut être difficile si les points GPS ne sont pas précisément relevés.

Ainsi, nous avons décidé de laisser libre les points GPS en entrée et d'effectuer des corrections au sein du script pour obtenir un parallélogramme. Ces corrections sont effectuées sur un seul point. Les coordonnées de ce point sont ensuite modifiées pour correspondre à la forme désirée. Le point corrigé est celui qui apporte le moins de modifications à l'emprise. En effet, il est souhaitable que la surface corrigée soit la plus faible possible pour se rapprocher au mieux de la zone souhaitée par l'utilisateur.

Si l'utilisateur entre des points GPS assez proches de la forme d'un parallélogramme alors les corrections seront minimes. A l'inverse, si la forme est très différente d'un parallélogramme, les corrections peuvent être importantes et la forme une fois corrigée risquera de ne plus correspondre à la forme voulue par l'utilisateur. Dans l'exemple ci dessous le point 2 est déplacé pour que le tracé puisse correspondre à un parallélogramme (en bleu).

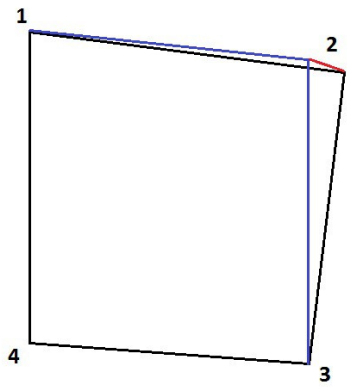


Illustration 10: Stratégies de couverture : correction d'un point

La fonction de correction de l'emprise se décompose en deux parties :

- La première partie consiste à effectuer la correction sur chacun des points, c'est à dire à modifier sa position pour la faire correspondre à la forme souhaitée.
- La seconde partie est une comparaison entre les 4 modifications possibles pour sélectionner celle qui change le moins la forme du périmètre d'intervention.

a) Correction sur le premier point

L'objectif de la première partie est de modifier les coordonnées d'un des 4 points GPS à partir des coordonnées des 3 autres.

Sachant que ce script s'appuie sur des points consécutifs, la formule pour obtenir les coordonnées du

4ème point sera :

```
|(x_p_1) – (x_p_2)| et |(y_p_1) – (y_p_2)|
```

où x_p_1 est l'abscisse du point 1 et y_p_1 l'ordonnée du points 1. Cela revient à calculer la valeur absolue de la différence entre les coordonnées des deux premiers points.

Ensuite deux options sont possibles :

- soit il faut ajouter cette valeur au point 3 pour obtenir les coordonnées du point 4
- soit il faut les soustraire.

Pour répondre à cette interrogation, les deux possibilités sont testées. La solution retenue est celle dont les longueurs entre le point 1 et le point 4 sont égales à la longueur entre le point 2 et le point 3. Les deux calculs de longueur sont effectués avec la formule de *Vincenty*.

Par ailleurs, au vu de la modification des données entrées par l'utilisateur, il est important de l'informer de ces modifications. Pour comprendre et quantifier ces modifications aisément et de manière efficace, une mesure de déplacement du point (en m ou en km) est calculée et affichée à titre informatif.

b) Comparaison entre les corrections

La seconde partie de la fonction de correction de la forme consiste à refaire le même traitement sur tous les autres points pour ensuite comparer l'indice de déplacement.

En sortie, le scénario retenu est celui où le déplacement est le moins important afin de modifier les coordonnées du point correspondant.

v. Étape 3 : La fonction de reconnaissance des points

A ce niveau, les 4 points n'ont pas de position définie les uns par rapport aux autres. Il faut donc les nommer pour fixer selon leurs positions, les éléments qui vont permettre le dessin du pattern. L'objectif est ici de passer de 4 points indifférents à 4 points nommés selon leurs positions respectives.

Schématiquement il y a 6 types de formes possibles de parallélogrammes. La forme en carré/rectangle parfaitement orienté Nord/Sud est la forme la plus simple à identifier. C'est la seule dont les deux points les plus au nord ont les mêmes coordonnées en Y (le test aurait également pu être effectué sur les autres cotés du carré/rectangle).

La forme en losange suit le même principe, le point le plus au nord et le point le plus au sud ont les mêmes coordonnées en abscisse (X).

Pour les autres formes, la fonction interroge l'abscisse des deux points les plus au nord pour la comparer à l'abscisse de deux points les plus au sud. Cela permet de différencier les emprises légèrement ou fortement orientées vers l'Ouest ou l'Est.

vi. Étape 4 : La fonction de dessin du pattern

Cette fonction est identique à celle développée dans la version 1 du module (*Création d'un pattern de navigation - Version 1 (4 orientations possibles)*).

E. Perspective de développement

Ce module n'a pas été développé par un programmeur de métier mais par un géomaticien. Cela implique donc que de très grosses évolutions dans la structure, l'optimisation ou dans l'écriture même du code peuvent être envisagées.

De même, certaines fonctionnalités ont été développées mais n'ont pas été étendues à toutes les

possibilités. La version 1 du module n'accepte que 2 patterns différents et le pattern en Zig Zag ne peut être orienté que vers le nord ou le sud. Il faudrait compléter avec d'autres patterns et intégrer pour chacun d'eux de nouvelles orientations.

La version 2 du module n'est pas finalisée car la gestion des pas de déplacement pour une orientation donnée n'est pas encore stable. Le point de blocage qui ne permet pas de finir le développement se situe au niveau de la définition des pas sur les axes Nord/Sud et Est/Ouest qui définissent en degrés pour chaque déplacement la distance au prochain point. Actuellement ce pas n'a pas pu être totalement testé et cela peut provoquer en sortie des patterns déformés qui ne prennent pas en compte la forme définie par l'utilisateur.

Toutefois le script a été conçu de façon modulaire. Il est donc possible d'ajouter d'autres patterns de navigation qui pourraient être plus intéressants selon une expertise effectuée en fonction de missions précises ou des caractéristiques de navigation particulières du drone.

Actuellement l'ajout de nouvelles formes (un triangle par exemple) est pensé pour être réalisé au début du *Module « Préparation à la navigation »* en croisant une couche de bathymétrie avec le pattern de navigation et la zone restreinte pour le pattern de navigation. Cela permet notamment de pouvoir diviser la zone d'étude efficacement selon les méthodes décrites dans le *Module « Déploiement d'une flotte »*.

La prise en compte d'une forme indéfinie pour créer un pattern implique de changer le fonctionnement du module mais serait la prochaine évolution majeure à apporter à ce module afin qu'il puisse s'intégrer directement et automatiquement entre le *Module « Déploiement d'une flotte »*. et le *Module « Préparation à la navigation »*.

V. Module « Préparation à la navigation »

Le module de préparation à la navigation est un outil d'aide à la décision destiné aussi bien à la cellule de coordination des opérations qu'aux pilotes de drones.

Il permet d'avoir une connaissance des caractéristiques de la zone d'étude et la possibilité de déterminer un itinéraire de navigation optimisé pour le ou les drones, avant même qu'ils ne soient déployés sur zone.

Dans le présent rapport, deux modèles sont présentés : une première version reposant sur une approche matricielle qui ne prend en compte que l'aspect bathymétrique et une seconde établie sur un principe de parcours de graphes qui intègre à la fois la bathymétrie et un facteur de vent.

A. Présentation générale du module

Le but principal de l'outil se focalise sur la possibilité de permettre au pilote de programmer un itinéraire pour les drones sur la zone d'étude à couvrir. Cet itinéraire se décline selon les paramètres suivants :

- la zone d'étude à couvrir : celle-ci est définie par l'emprise d'une couche d'information géographique, définissant ainsi l'étendue maximale de la zone : photo satellite, couche de bathymétrie, etc.
- le niveau de profondeur porté par la couche de données bathymétriques : selon le gabarit du drone, un niveau de profondeur trop faible peut l'abîmer, et par conséquent mettre en péril la mission. Toute partie de la zone d'étude dont la profondeur est supérieure au seuil passé en paramètres est définie comme un obstacle pour le drone.
- les conditions de vent : ce critère s'applique uniquement au *Modèle v2 : analyse de graphe selon la bathymétrie et le vent* et permet, à partir d'une vitesse et d'une orientation de vent, de définir le comportement du drone selon son orientation de déplacement.
- un itinéraire théorique de couverture de zone : cet itinéraire, généré à partir du *Module « Stratégies de couverture de zone »* permet de définir un parcours théorique pour le drone afin de couvrir l'ensemble de la zone d'étude.

Dans le cadre de ce projet, un travail exploratoire a permis d'aboutir au développement de deux modèles différents :

- une première version qui s'appuie sur une logique de recherche de plus court chemin en mode matriciel et qui ne prend en compte que la bathymétrie. Ce modèle est une première approche permettant de mettre en avant des méthodologies en géomatique qui pourront être réutilisables par la suite, mais qui ne permettent pas pour l'instant d'approcher des conditions réelles de navigation d'un drone à voile
- une deuxième version qui s'appuie sur une logique de plus court chemin en mode vecteur et qui prend également en compte des conditions de vent simulées. Ce modèle, par la suite couplé à des référentiels de données météorologiques, permettra de se rapprocher des conditions réelles de navigation.

B. Concepts-clés

i. Analyse multicritères

L'analyse multicritères (AMC) consiste à croiser plusieurs couches de données thématiques au format raster (ex : couche bathymétrique, zones de vents, etc.) en attribuant pour chacune d'elles un facteur de pondération. Celle-ci permet d'avantager ou de contraindre les facteurs d'influence selon leur importance considérée dans l'analyse. L'objectif est de produire en sortie une carte synthétique permettant de déterminer les contraintes qui s'exercent sur le territoire étudié.

ii. Plus court chemin (raster)

Pour calculer le chemin le plus court dans une image raster, chaque pixel a une valeur qui correspond à un poids. Plus la valeur d'un pixel est élevée, moins il est probable que l'itinéraire généré ne passe par celui-ci. Autrement dit, l'algorithme du plus court chemin cherche à emprunter l'ensemble des pixels contigus qui ont une somme de valeur la plus faible entre un point d'origine et un point de destination.

iii. Plus court chemin (vecteur) ou parcours de graphe

Les recherches de "plus court chemin" fait appel à la théorie des graphes. L'objectif est de calculer une route entre des sommets d'un graphe qui minimise ou maximise une certaine fonction économique. Dans le cas de l'algorithme de Dijkstra, il permet, par exemple, de déterminer le plus court chemin pour se rendre d'une ville à une autre connaissant le réseau routier d'une région. Il s'applique à un graphe connexe dont le poids lié aux arêtes est un réel positif.

C. Modèle v1 : analyse bathymétrique par chemin de coût matriciel

La première version du module d'analyse multicritères présentée ici repose sur une approche matricielle. Cette version permet de définir un itinéraire de navigation optimal en prenant en compte d'une part un motif de navigation idéal pré-défini, et d'autre part la bathymétrie de la zone d'étude. L'approche matricielle consiste à réaliser l'analyse multicritères à partir d'un algèbre de couches.

La couche bathymétrique mobilisée en entrée permet de définir l'emprise géographique de la zone sur laquelle s'applique le chemin de moindre coût, à l'inverse du pattern de navigation, qui présente un itinéraire théorique idéal.

i. Paramètres de navigation pris en compte dans le modèle

Pour cette première version de modèle, seule la couche raster de bathymétrie est prise en compte. Elle permet à la fois :

- de définir une emprise géographique de la zone d'étude
- de déterminer tout obstacle pouvant nuire à la navigation du drone pour couvrir la zone

ii. Méthodologie et implémentation

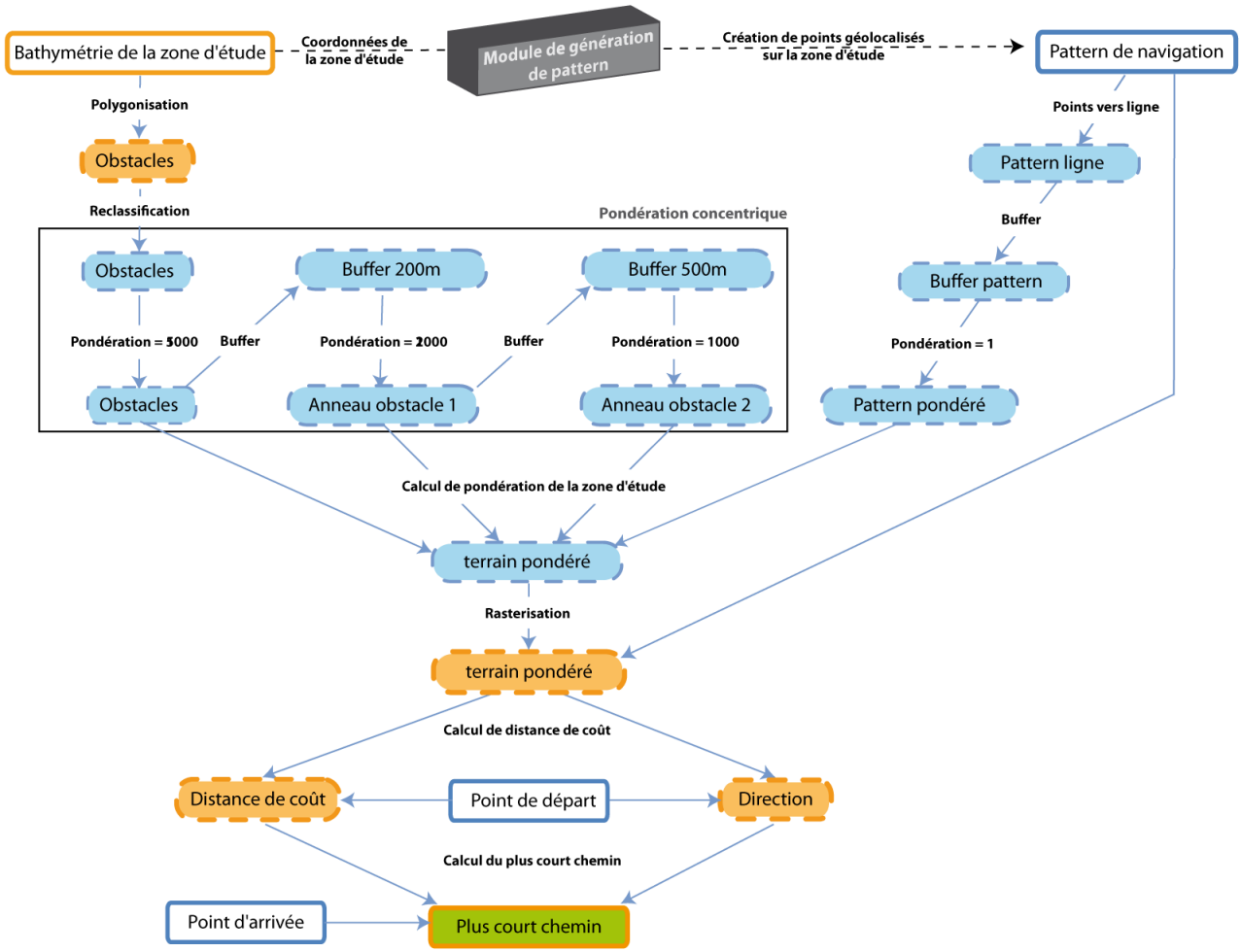


Illustration 11: Préparation à la navigation - chaîne de traitements du modèle v1

Le modèle implémenté a pour objectif de programmer l'itinéraire du drone de telle sorte qu'il suive le plus fidèlement possible le pattern de navigation tout en évitant de s'approcher trop près des obstacles potentiels qu'il peut rencontrer en chemin. Cela se traduit donc par une pondération très faible du tracé du pattern de navigation et d'une pondération élevée à proximité des zones émergées ou peu profondes.

Pour être en mesure de répondre à cet enjeu, le modèle se décompose en 12 étapes implémentées dans le Model Builder d'ArcGIS, logiciel SIG propriétaire.

L'ensemble de ces étapes peut être regroupé en quatre phases principales :

- Calcul de la pondération en fonction du pattern de navigation
- Calcul de la pondération en fonction de la bathymétrie
- Croisement des pondérations sur l'ensemble de la zone d'étude
- Calcul du plus court chemin

a) Calcul de la pondération en fonction du pattern de navigation

L'objectif de cette phase est de calculer la pondération de la zone d'étude en fonction du pattern de navigation. Celui-ci est généré à partir du *Module « Stratégies de couverture de zone »* et de l'emprise géographique de la couche raster de bathymétrie.

Ce pattern de navigation, exprimé par un ensemble de points géolocalisés, est transformé en couche vectorielle linéaire afin d'obtenir une seule entité continue pour le tracé du plus court chemin.

Pour des raisons techniques, ce linéaire est ensuite élargi pour être en mesure de lancer l'algorithme de plus court chemin en dernière phase (la largeur du linéaire est tellement faible qu'il ne comble sinon pas la taille d'un pixel). Cela permet également de laisser un minimum de liberté au drone pour se

déplacer sur ce couloir.

Enfin, une pondération très faible est affectée à cette ligne pour que le drone soit incité à emprunter ce chemin.

b) Calcul de la pondération en fonction de la bathymétrie

Le calcul de pondération s'effectue en plusieurs étapes successives qui se basent sur l'ensemble des pixels dont la valeur de profondeur est estimée comme insuffisante pour naviguer.

Cette valeur est ici, dans le cas d'un drone mesurant 7 mètres de long, fixée arbitrairement à 10m, mais il est tout à fait envisageable de faire évoluer le modèle en paramétrant cette valeur avant l'exécution du modèle.

A partir de la couche de bathymétrie, l'opération de classification permet de ne conserver que les zones considérées comme des obstacles. Deux zones tampon successives sont ensuite appliquées autour des obstacles afin d'y affecter une pondération décroissante. cela permet de simuler un effet répulsif croissant autour des zones non navigables pour les drones.

En sortie de phase, trois couches surfaciques concentriques sont générées avec un gradient de pondération décroissant autour des obstacles.

c) Croisement des pondérations sur l'ensemble de la zone d'étude

Le croisement des valeurs de chaque couche se fait par deux opérations successives.

La première consiste à combiner les différentes couches vectorielles pour n'obtenir qu'une unique couche où l'ensemble des entités géométriques se superposent.

La seconde opération effectue la conversion de la couche vectorielle au format matriciel.

Toute surface qui n'est pas couverte par une entité géométrique n'aura pas de pondération. Cela signifie que la couche matricielle n'aura pas de pixel pour ces zones.

De plus, cette conversion applique en priorité le poids de plus forte valeur lorsque plusieurs entités se superposent.

Ces deux aspects permettent au drone d'emprunter le pattern théorique dans la limite où il ne passe pas au dessus d'un obstacle.

d) Calcul du plus court chemin

La phase finale du modèle s'appuie sur le calcul du plus court chemin entre deux points d'origine et de destination matérialisés par deux couches vectorielles distinctes. Elle se décline en deux étapes.

La première étape consiste à effectuer un calcul de distance de coût qui génère deux couche en sortie :

- une couche de distance de coût qui exprime le coût cumulé depuis le point de départ jusqu'à n'importe quel pixel de la zone d'étude. Par conséquent, cette opération ne se résume pas à une distance euclidienne, mais prend en compte la pondération qui est issue de la phase 2 du modèle
- une couche de direction qui indique la direction à privilégier à partir du point de départ. Cette opération peut être comparée à un modèle d'écoulement depuis un point de source, où le liquide emprunte la pente la plus proche sous l'effet de la gravité.

La seconde étape reprend les deux couches matricielles qui viennent d'être créées pour calculer le chemin le plus court en fonction du point d'arrivée défini.

La couche matricielle en sortie trace ainsi le chemin le plus court en fonction des conditions bathymétriques.

Entrées :
Bathymétrie (couche raster)
Pattern de navigation (couche vectorielle en ponctuel)
Point de départ (couche vectorielle en ponctuel)

Point d'arrivée (couche vectorielle en ponctuel)
Sortie :
Chemin le plus court (couche raster)

iii. Limites et perspectives

Ce modèle est une première approche de l'analyse multicritères et a été développé à titre expérimental. En effet, il permet de prendre en compte les données bathymétriques d'une zone d'étude pour y adapter un schéma de navigation théorique qui tient compte de ses contraintes physiques et ainsi éviter la collision avec un obstacle naturel ou l'échouage du drone.

Cependant, les capacités du modèle restent très limitées puisqu'il n'intègre que l'aspect bathymétrique et ne tient pas compte des autres paramètres influençant fortement la navigation à voile : vents, courants, etc...

En outre, ce modèle a été implémenté sous ArcGIS 10.1, un logiciel propriétaire sous licence payante et ne correspond donc pas à la logique Open Source du projet Protei.

Il est néanmoins possible d'exploiter le potentiel de cette version en la croisant avec la seconde. En effet, l'approche matricielle en tant que telle n'est pas satisfaisante pour un paramètre de vent unidirectionnel puisque cela revient a avoir une couche de directions contenant les mêmes valeurs pour chaque pixel. Dès lors, il est plus simple de pondérer l'ensemble des données directement. Si l'on souhaitait affecter un coût en fonction de la direction pouvant être prise par le drone, il serait nécessaire d'adopter une logique combinatoire intégrant 8 couches de coûts, soit une par direction.

L'approche matricielle devient intéressante à partir du moment où la zone d'étude peut être subdivisée en plusieurs zones de vent ou de courants. En effet, la mise en œuvre d'une couche de directions de flux sur ces deux paramètres permettrait de discriminer les différentes zones identifiées et de préparer leur intégration dans le processus d'analyse du chemin de moindre coût.

Ces améliorations peuvent être élaborées dans une version Open Source du modèle à partir de logiciels tels que QGIS, GRASS et SAGA combinés au SGBD PostgreSQL/PostGIS.

Ainsi, le potentiel de cette version ne réside pas dans son développement en propre mais bien dans la perspective d'une fusion entre les deux approches proposées dans les versions 1 et 2 du modèle.

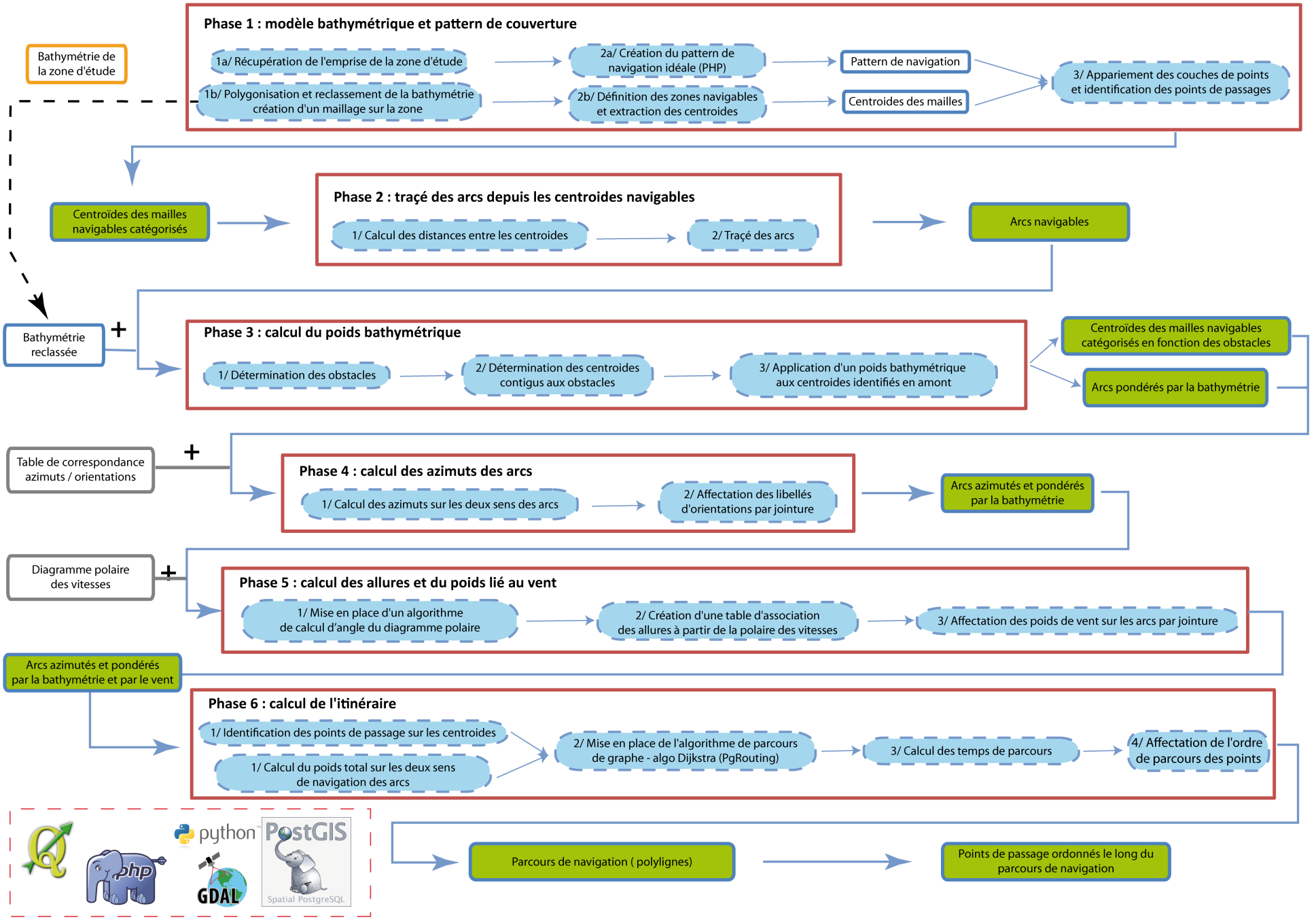
D. Modèle v2 : analyse de graphe selon la bathymétrie et le vent

En raison des limites exposées dans la version 1 du modèle, nous avons réfléchi à l'élaboration d'une nouvelle approche géomatique permettant d'intégrer des paramètres de vent dans l'analyse multicritères. Étant donné que les drones sont uniquement équipés d'une voile et non d'un moteur, le vent est alors un facteur primordial à prendre en compte dans le module.

Cette seconde version repose donc sur le principe du parcours de graphe*(approche vecteur). Elle permet de prendre en compte trois principaux facteurs :

- un premier facteur environnemental sur la localisation des obstacles, et plus globalement des zones non navigables
- un deuxième facteur environnemental sur la direction et la force du vent. Pour simplifier le modèle, ce vent est considéré comme constant et unidirectionnel le temps d'une simulation.
- la stratégie de couverture de zone sélectionnée avant le lancement d'une simulation.

Ainsi, au travers de ces différents facteurs, le modèle permet de définir un itinéraire de navigation optimal sur la zone d'étude avec un résultat plus proche de la réalité que la première version proposée.


Phase 1 : modèle bathymétrique et pattern de couverture
Bathymétrie de la zone d'étude
1a/ Récupération de l'emprise de la zone d'étude
1b/ Polygonisation et reclassement de la bathymétrie création d'un maillage sur la zone
2a/ Création du pattern de navigation idéale (PHP)
2b/ Définition des zones navigables et extraction des centroides
Pattern de navigation
Centroides des mailles
3/ Appariement des couches de points et identification des points de passages
Centroïdes des mailles navigables catégorisés
Phase 2 : traçé des arcs depuis les centroides navigables
1/ Calcul des distances entre les centroides
2/ Traçé des arcs
Arcs navigables
Bathymétrie reclassée
+
Phase 3 : calcul du poids bathymétrique
1/ Détermination des obstacles
2/ Détermination des centroides contigus aux obstacles
3/ Application d'un poids bathymétrique aux centroides identifiés en amont
Centroïdes des mailles navigables catégorisés en fonction des obstacles
Arcs pondérés par la bathymétrie
Table de correspondance azimuts / orientations
+
Phase 4 : calcul des azimuts des arcs
1/ Calcul des azimuts sur les deux sens des arcs
2/ Affectation des libellés d'orientations par jointure
Arcs azimutés et pondérés par la bathymétrie
Diagramme polaire des vitesses
+
Phase 5 : calcul des allures et du poids lié au vent
1/ Mise en place d'un algorithme de calcul d'angle du diagramme polaire
2/ Création d'une table d'association des allures à partir de la polaire des vitesses
3/ Affectation des poids de vent sur les arcs par jointure
Arcs azimutés et pondérés par la bathymétrie et par le vent
Phase 6 : calcul de l'itinéraire
1/ Identification des points de passage sur les centroides
1/ Calcul du poids total sur les deux sens de navigation des arcs
2/ Mise en place de l'algorithme de parcours de graphe - algo Dijkstra (PgRouting)
3/ Calcul des temps de parcours
4/ Affectation de l'ordre de parcours des points
python
PostGIS
Spatial PostgreSQL
GDAL
Parcours de navigation (polylignes)
Points de passage ordonnés le long du parcours de navigation

Remarque importante : l'ensemble des requêtes SQL décrites dans ce module sont embarquées dans le modèle sous la forme de scripts pyQGIS. Néanmoins, lorsque les requêtes doivent prendre en compte un paramètre d'entrée, le script pyQGIS est décrit plutôt que la requête SQL en elle-même pour préciser cet aspect.

i. Phase 1 : vectorisation de la zone d'étude

La première phase du module d'analyse multicritères repose sur une couche matricielle de bathymétrie qui définit l'emprise géographique de la zone d'étude. L'enjeu est ici de passer d'un format matriciel à un format vectoriel pour effectuer l'ensemble des étapes du module.

Module d'analyse multicritères - phase 1 : extraction des centroides navigables

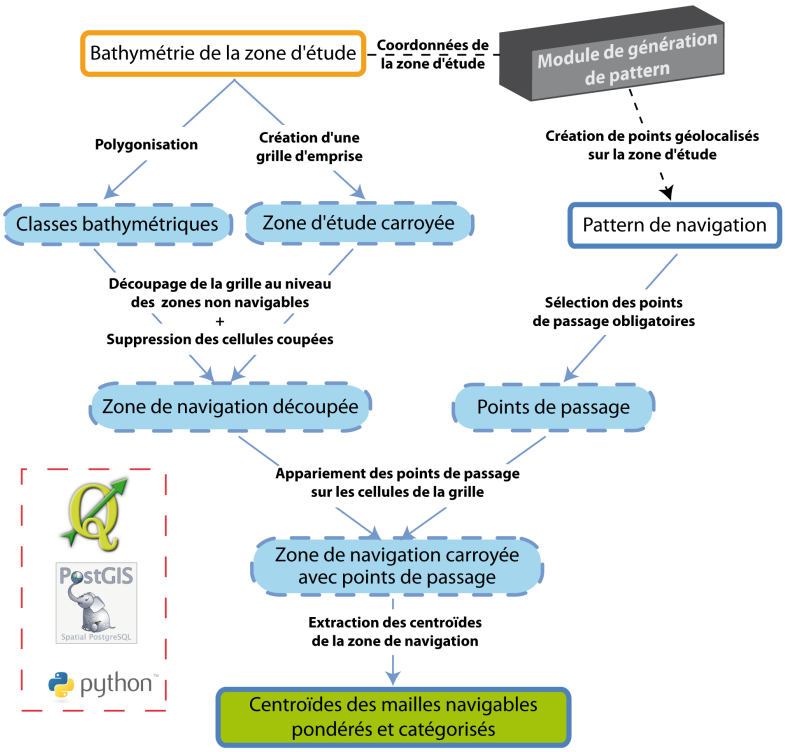


Illustration 12: Préparation à la navigation : chaîne de traitements phase 1

a) *Étape 1 : GDAL - Polygonize*

Cette première étape consiste à vectoriser, à partir de la couche bathymétrique en entrée, les classes de profondeur en multipolygones. Tous les pixels ayant la même valeur sont polygonisés au sein d'une même entité polygonale.

L'opération est assurée par l'outil "Polygonize" fourni par la librairie GDAL.

Entrée : couche raster "Bathy"
Sortie : couche vecteur polygone "Bathy_classes"

b) *Étape 2 : pyQGIS - createGrid from raster*

Cette étape consiste à traduire l'emprise géographique de la couche matricielle en une couche vectorielle sous la forme d'une grille. Ainsi, chaque pixel de la couche matricielle se traduit par une maille

carrée de la grille qui sert ensuite de support pour le reste du modèle.

L'opération est assurée par un script développé en pyQGIS.

Entrée : couche raster "Bathy"
Sortie : couche vecteur polygone "grille_bathy"

c) Étape 3 : PostGIS - import des couches générées aux étapes 1 et 2

Cette étape permet d'importer les couches vectorielles générées lors des deux premières étapes dans la base de données PostgreSQL/PostGIS.

L'opération est assurée par deux scripts d'import dans la base de données "Protei"

Entrée : couches vecteur polygone "Bathy_classes" et "grille_bathy"
Sortie : néant

d) Étape 4 : PostGIS - requêtes SQL de découpage de la grille de navigation

L'hypothèse principale consiste à définir comme non navigable toute zone dont la profondeur est inférieure à une valeur passée en paramètre. Ces surfaces sont alors assimilées à des obstacles.

Cette étape permet ainsi d'effectuer un découpage de la couche "grille_bathy" par les zones de la couche "bathy_classes" pour lesquelles les valeurs de profondeur sont supérieures à la valeur du paramètre. La couche créée récupère également pour chaque maille de la grille les informations suivantes :

- l'id
- la longitude
- la latitude
- la valeur de la profondeur (%profondeur_min%)

Cette étape vérifie également si les valeurs de profondeur fournies en entrée sont positives ou négatives afin de s'assurer de ne sélectionner que les zones navigables.
En effet, certains fournisseurs de données définissent la profondeur avec des valeurs positives (ex : l'institut français IFREMER) alors que d'autres la définissent avec des valeurs négatives (ex : l'institut anglais BODC)

L'opération est assurée par le script pyQGIS suivant :

```
#Creation des obstacles en selectionnant les cellules de la grille issue de la
bathymetrie dont la valeur de profondeur :
# est superieure a celles de la grille de navigation dans le cas de donnees
bathymetriques negatives
# est inferieure a celles de la grille de navigation dans le cas de donnees
bathymetriques positives
statement = ""
if(valeurs_negatives):
    statement = "< " + profondeur_min
else:
    statement = "> " + profondeur_min
print statement

outputs_4=processing.runalg("gspg:postgisexecutesql", 'Protei', "DROP TABLE IF
EXISTS grille_navigation; \
CREATE TABLE grille_navigation AS \
SELECT grille_bathy.the_geom AS the_geom, grille_bathy.id AS id, longitude,
latitude, bathy_classes.depth \
FROM grille_bathy, bathy_classes \
```

```
WHERE ST_Within(ST_Centroid(grille_bathy.the_geom), bathy_classes.the_geom) \
AND depth " + statement + "; \
UPDATE grille_navigation set the_geom = ST_SetSRID(the_geom, 4326)")
```

Entrées : grille_bathy, bathy_classes (depuis PostGIS)
Sortie : grille_navigation (dans PostGIS)

e) Étape 5 : PostGIS - Récupération des points de passage du pattern et extraction des centroïdes

- **Récupération de la localisation des points de passage**

Il s'agit de récupérer les points de passage du pattern de navigation qui se situent sur la zone navigable. A partir de la couche vectorielle "grille_navigation", tous les points intermédiaires du pattern (dont l'attribut 'point_inte' = 1) se situant à l'intérieur sont considérés comme des points de passage. La requête suivante permet de récupérer les identifiants des points de passage du pattern de navigation pour les associer aux centroïdes navigables afin de forcer le drone à passer par ces points :

```
-- Ajout de la colonne 'pt_passage'
UPDATE grille_navigation gn
SET pt_passage = g.pid
FROM (
  SELECT gn.id, p.id AS pid
  FROM grille_navigation gn, pattern p
  WHERE ST_Within(p.the_geom, gn.the_geom)
  AND point_inte = 1
  ) AS g
WHERE gn.id = g.id
```

Entrée : grille_navigation (depuis PostGIS)
Sortie : néant

- **Création des centroïdes de la grille de navigation**

Cette dernière étape de la phase 1 permet d'obtenir l'ensemble des centroïdes de la grille de navigation par lesquels le drone peut naviguer.

L'opération est assurée par les requêtes SQL suivantes :

```
-- Suppression éventuelle de la table
DROP TABLE IF EXISTS centroïdes_navigables;
-- Création des centroïdes à partir de la grille de navigation
CREATE TABLE centroïdes_navigables AS
SELECT ST_Centroid(the_geom) AS the_geom, id, longitude, latitude, depth,
pt_passage
FROM grille_navigation;
-- Mise à jour du SRID
UPDATE centroïdes_navigables set the_geom = ST_SetSRID(the_geom, 4326)
```

Entrée : grille_navigation (depuis PostGIS)
Sortie : centroïdes_navigables (dans PostGIS)

ii. Phase 2 : création des arcs de navigation

Cette phase consiste à tracer l'ensemble des arcs reliant les points de la couche "centroïdes_navigables" générée à la fin de la phase 1 (*Étape 5 : PostGIS - Récupération des points de passage du pattern et extraction des centroïdes*). La couche créée récupère les informations suivantes à partir de la couche en entrée :

- l'id du centroïde de départ de l'arc
- l'id du centroïde d'arrivée de l'arc

Module d'analyse multicritères - phase 2 : création des arcs depuis les centroides

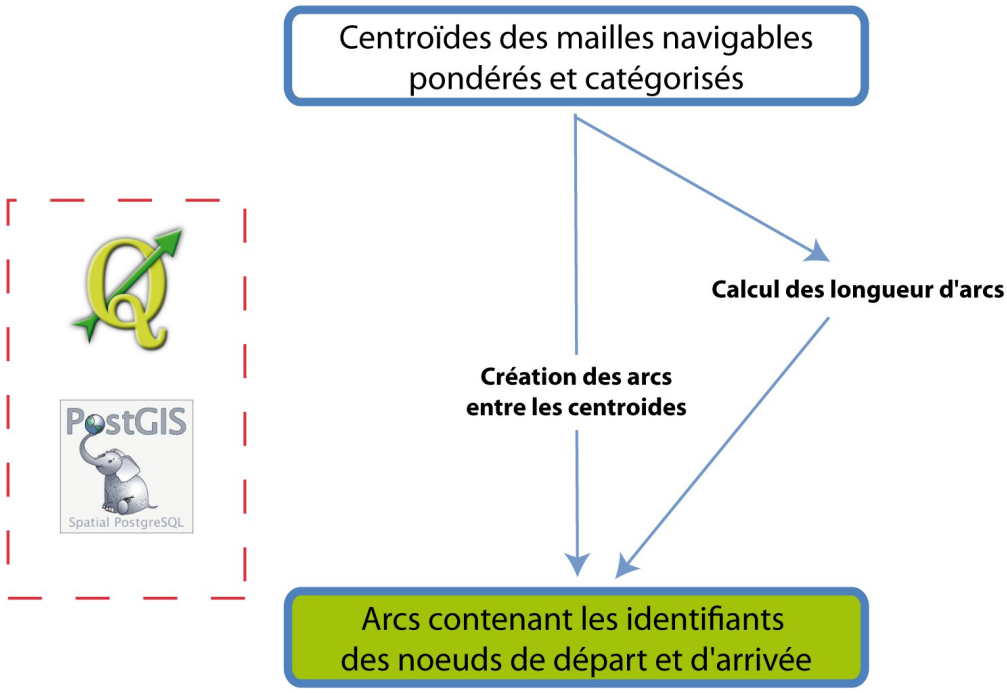


Illustration 13: préparation à la navigation - chaîne de traitements phase 2

L'opération est assurée par les requêtes SQL suivantes :

```
/*Creation des arcs a partir des centroïdes navigables. Chaque centroïde sera
rattache a 8 arcs, sauf dans le cas ou le
centroïde sur situe proche d'un obstacle ou un bord de la zone d'etude*/
DROP TABLE IF EXISTS arcs;
CREATE TABLE arcs AS
SELECT ST_Makeline(g1.the_geom, g2.the_geom) AS geom, g1.id AS id_dep, g2.id AS
id_dest, ST_Distance(g1.the_geom, g2.the_geom) AS dist_deg
FROM centroïdes_navigables g1, centroïdes_navigables g2
WHERE ST_Distance(g1.the_geom, g2.the_geom) <=
(SELECT MIN(ST_Distance(c1.the_geom, c2.the_geom))
    FROM centroïdes_navigables c1, centroïdes_navigables c2
    WHERE c1.id = c2.id+1)*sqrt(2)*1.0001
AND g1.id != g2.id
AND g1.id < g2.id;
UPDATE arcs set geom = ST_SetSRID(geom, 4326);
ALTER TABLE arcs ADD gid serial NOT NULL primary key;
```

Ces requêtes permettent de :

1. supprimer la table si elle existe pour la regénérer
2. créer la nouvelle table des arcs permettant de se déplacer sur la zone d'étude, avec la prise en compte des obstacles et des points de passage obligatoires
3. mettre à jour le référentiel géographique
4. créer un identifiant unique pour chaque arc de navigation

La valeur de la longueur maximale d'un arc entre deux centroïdes est fixée par trois facteurs :

- La valeur minimum entre deux centroïdes qui correspond à une longueur de maille
- La racine carrée de 2 correspondant au facteur d'hypoténuse d'une maille carrée
- Un facteur de tolérance de 0.001% pour les éventuelles différences de distances entre centroïdes lors de la génération de la couche de sortie de la phase 1

Pour éviter d'avoir des arcs "jumeaux" qui se superposent (un arc allant du centroïde 1 vers le centroïde 2 et inversement), une astuce consiste à ne tracer que les arcs pour lesquels l'id de départ est inférieur à l'id de destination.

Entrée : centroïdes_navigables (depuis PostGIS)
Sortie : arcs (dans PostGIS)

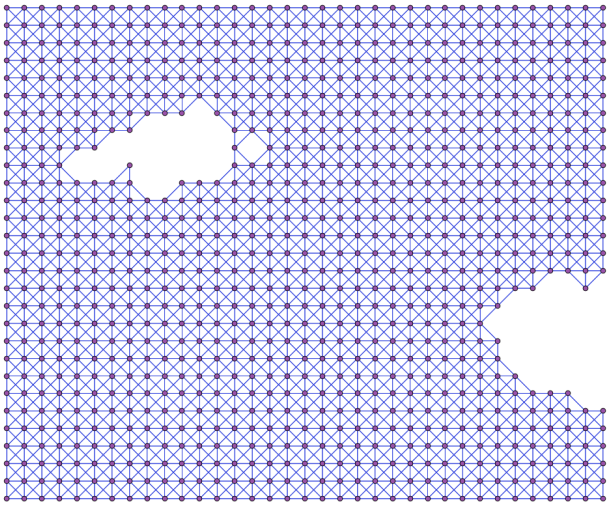


Illustration 14: Préparation à la navigation - résultats de génération d'arcs

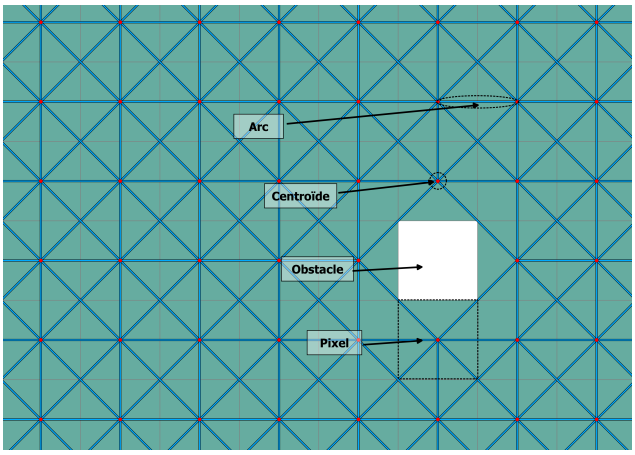


Illustration 15: Préparation à la navigation - description des éléments constitutifs de la zone de navigation

iii. Phase 3 : calcul de pondération induite par la bathymétrie

Cette phase permet de prendre en compte le facteur de proximité des centroïdes navigables aux zones non navigables afin de limiter les déplacements du drone sur ces espaces pour des raisons de sécurité. En effet, il parait essentiel d'éviter qu'un drone navigue trop près d'obstacles pour éviter de l'abîmer et de compromettre ainsi la mission.

Le modèle sous QGIS présente ici trois étapes :

1. Détermination des obstacles
2. Détermination des centroïdes contigus à ces obstacles
3. Application d'un poids sur les arcs ayant ces centroïdes comme centroïdes de destination

Module d'anlyse multicritères- phase 3 : affectation des poids bathymétriques

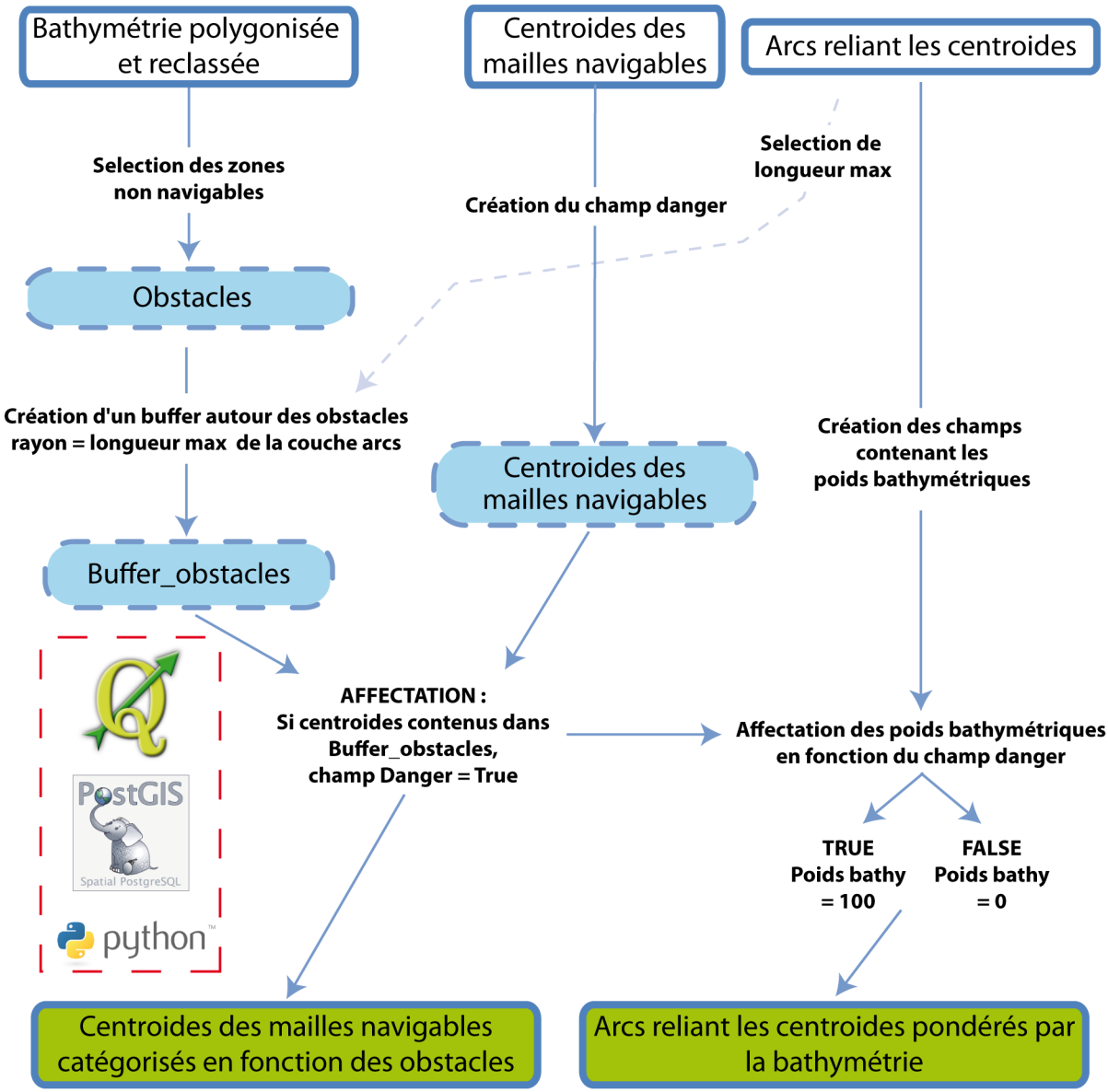


Illustration 16: Préparation à la navigation - chaîne de traitements phase 3

Ces trois étapes sont incluses au sein d'un bloc unique dans le Model Builder pour lequel on applique plusieurs requêtes SQL sous PostGIS sur deux couches :

- la couche des centroïdes obtenues à l'issue de la phase 1

- la couche des arcs obtenue à l'issue de la phase 2

a) Principe d'attribution d'un poids sur les arcs

L'attribution du poids des arcs dépend du poids du nœud d'arrivée de l'arc pour chaque sens.

Ainsi, chaque arc aura deux poids différents selon la direction prise par le drone :
- le poids du centroïde 'A' si le drone se déplace sur l'arc du nœud 'B' vers le nœud 'A'
- le poids du centroïde 'B' si le drone se déplace en sens inverse

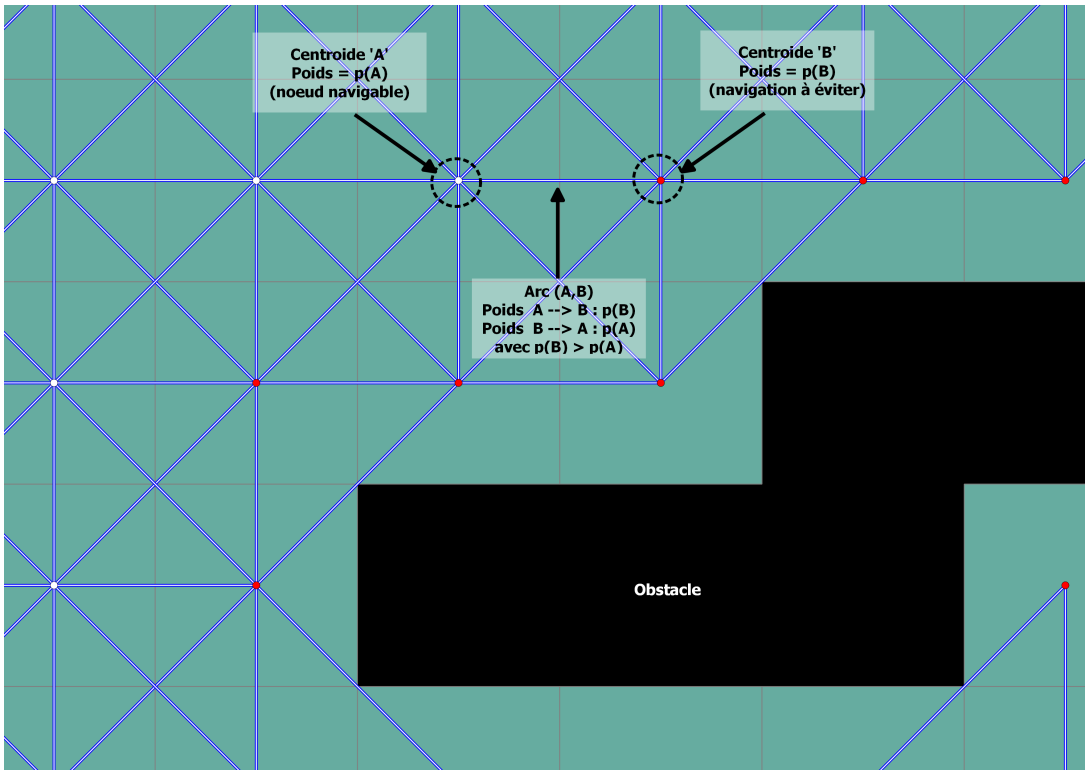


Illustration 17: Préparation à la navigation - principe de la pondération bathymétrique

b) Détermination des obstacles

Cette étape revient à déterminer la zone complémentaire de la couche "grille_navigation" par rapport à la couche "grille_bathy" générées lors de la *Phase 1 : vectorisation de la zone d'étude*.

L'opération est assurée par le script pyQGIS suivant :

```
#Creation des obstacles en selectionnant les cellules de la grille issue de la
bathymetrie dont la valeur de profondeur :
# est superieure a celles de la grille de navigation dans le cas de donnees
bathymetriques negatives
# est inferieure a celles de la grille de navigation dans le cas de donnees
bathymetriques positives

#Gestion des valeurs positives ou negatives des donnes bathymetriques
statement = ""
if(valeurs_negatives):
    statement = "> (SELECT MAX(depth)"
else:
    statement = "< (SELECT MIN(depth)"

outputs_0=processing.runalg("gspg:postgisexecutesql", 'Protei', "DROP TABLE IF
EXISTS obstacles; \
```

```
CREATE TABLE obstacles AS \
SELECT ST_Union(the_geom) AS the_geom \
FROM bathy_classes \
WHERE depth " + statement + " \
    FROM grille_navigation);")
```

Entrées : grille_navigation, bathy_classes (depuis PostGIS)
Sortie : obstacles (dans PostGIS)

c) Détermination des centroïdes navigables contigus aux obstacles

Cette étape permet de déterminer les centroïdes navigables qui sont les plus proches des obstacles. Par l'application d'un poids supérieur à celui des autres centroïdes, le drone aura moins de chance d'emprunter les arcs qui mènent aux zones moins sécurisées car peu profondes.

Pour effectuer cette opération, une zone tampon (ou buffer) est appliquée autour des obstacles afin de connaître les centroïdes navigables qui se situent à proximité des zones dangereuses. La distance tampon est déterminée par la plus grande distance entre deux centroïdes, autrement dit par l'arc ayant la longueur la plus grande par mesure de sécurité.

Les requête SQL suivantes permettent d'effectuer cette opération :

```
-- Création du champ 'danger';
DO $$
    BEGIN
        BEGIN
            ALTER TABLE centroïdes_navigables ADD COLUMN danger boolean;
       EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <danger> already
exists in <centroïdes_navigables>.';
        END;
    END;
$$;
```

Remarque: cette étape permet de créer un champ s'il n'existe pas encore dans la table 'arcs'. C'est une condition nécessaire pour exécuter le script de la phase 3 indépendamment des phases précédentes si elles ont déjà été exécutées.

```
/* Creation d'une fonction permettant de marquer les centroids consideres comme
trop proches des obstacles */
CREATE OR REPLACE FUNCTION centroïdes_dangereux() RETURNS void  AS $$
DECLARE

    dist_max float;
BEGIN
SELECT MAX(dist_deg) INTO dist_max FROM arcs;
DROP TABLE IF EXISTS  buffer_obstacles;
CREATE TABLE buffer_obstacles AS
SELECT ST_Buffer(o.the_geom, dist_max) AS the_geom
       FROM obstacles o;

UPDATE centroïdes_navigables c1
SET danger = true
    WHERE c1.id IN (
    SELECT c2.id
    FROM centroïdes_navigables c2, buffer_obstacles b
    WHERE ST_Intersects(c2.the_geom, b.the_geom)
    );
END;
$$ LANGUAGE plpgsql;
SELECT centroïdes_dangereux()
```

Les centroïdes répondant à ces conditions sont donc identifiés dans la table afin d'appliquer un poids sur les arcs qui y sont reliés lors de l'étape suivante.

Entrée 1 : arcs (depuis PostGIS)
Sortie 1 : buffer_obstacles (dans PostGIS)

Entrées 2 : centroïdes_navigables, buffer_obstacles (depuis PostGIS)
Sortie 2 : champ 'danger' de la table centroïdes_navigables (dans PostGIS)

d) Application des poids sur les arcs

Cette dernière étape permet d'appliquer un poids à l'arc relié à chaque centroïde considéré comme dangereux, et ce qu'il soit dans le sens direct ou indirect

Ce transfert d'information aboutit à la création de deux nouveaux champs dans la table "arcs" : 'pds_direct' et 'pds_indir' pour distinguer les deux sens de navigation pour chaque arc.

L'opération est assurée par les requêtes SQL suivantes :

```
-- Création des champs 'pds_direct' et 'pds_indir';
DO $$
    BEGIN
        BEGIN
            ALTER TABLE arcs ADD COLUMN pds_direct float;
UPDATE arcs SET pds_direct = 0;
        EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <pds_direct> already
exists in <arcs>.';
        END;
        BEGIN
            ALTER TABLE arcs ADD COLUMN pds_indir float;
UPDATE arcs SET pds_indir = 0;
        EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <pds_indir> already
exists in <arcs>.';
        END;
    END;
$$;
-- récuperation des poids de b dans le sens direct a --> b;
UPDATE arcs
    SET pds_direct = 100
    WHERE id_dest IN
       (SELECT id
       FROM centroïdes_navigables c1
       WHERE danger = true
       );
-- récuperation des poids de a dans le sens indirect b --> a;
UPDATE arcs
    SET pds_indir = 100
    WHERE id_dep IN
       (SELECT id
       FROM centroïdes_navigables c1
       WHERE danger = true
       );
```

Entrées : centroïdes_navigables(depuis PostGIS)
Sortie : champs de poids d'obstacles de la table arcs (dans PostGIS)

iv. Phase 4 : calcul des allures du drone

Cette phase permet d'intégrer le critère vent dans le modèle d'analyse multicritères. Il s'agit donc de déterminer l'orientation et le sens des arcs que parcourt le drone.

Le modèle sous QGIS présente ici deux étapes :

1. Calcul de l'azimut pour chaque arc en fonction de la direction prise par le drone
2. Direction du drone en fonction de l'azimut

Module d'analyse multicritères - phase 4 : calcul des allures et des poids de vent

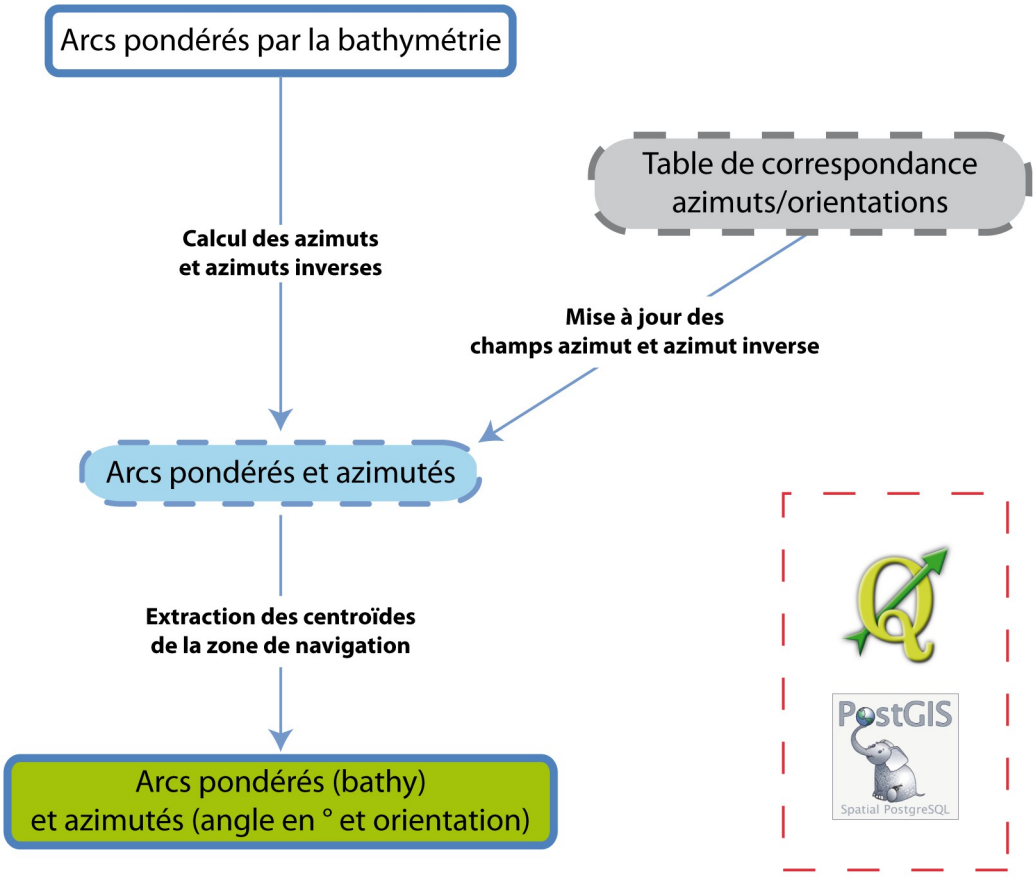


Illustration 18: Préparation à la navigation - chaîne de traitements phase 4

Ces étapes sont incluses au sein d'un bloc unique du modeler dans lequel on applique plusieurs requêtes SQL sous PostGIS sur la couche des arcs obtenue à l'issue de la phase 2

a) Étape 1 : PostGIS - calcul des azimuts

Cette étape permet de définir l'azimut (en degrés) de chaque arc selon le sens de navigation du drone. Il est donc nécessaire de calculer deux valeurs d'azimut par arc.

L'azimut servira ensuite à définir l'allure du drone en fonction de l'orientation du vent.

Le calcul des azimuts est assuré par les requêtes SQL suivantes :

```
-- Création des champs 'azimut' et 'azimut_inv'
DO $$
    BEGIN
        BEGIN
            ALTER TABLE arcs ADD COLUMN azimut int;
        EXCEPTION
```

```
            WHEN duplicate_column THEN RAISE NOTICE 'column <azimut> already
exists in <arcs>.';
        END;
         BEGIN
            ALTER TABLE arcs ADD COLUMN azimut_inv int;
        EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <azimut_inv> already
exists in <arcs>.';
        END;
    END;
$$;
-- orientation des arcs (azimut)
UPDATE arcs
SET
    azimut = round((ST_azimuth (g1.the_geom, g2.the_geom)/
(2*pi())*360)::numeric, 0),
    azimut_inv = round((ST_azimuth (g2.the_geom, g1.the_geom)/
(2*pi())*360)::numeric, 0)
    FROM centroïdes_navigables g1, centroïdes_navigables g2
    WHERE g1.id = arcs.id_dep
    AND g2.id = arcs.id_dest
```

Entrées : centroïdes_navigables, arcs (depuis PostGIS)
Sortie : champs d'azimuts dans la couche arcs (dans PostGIS)

b) Étape 2 : PostGIS - table de correspondance entre azimut et direction

Pour des raisons de praticité et de facilité de lecture, nous avons défini les différents caps du drone en fonction des champs 'azimut' et 'azimut_inverse'. La table de correspondance est structurée comme suit :

Azimut (degrés)	Orientation (N-S-E-W)
0	S-N
45	SW-NE
90	W-E
135	NW-SE
180	N-S
225	NE-SW
270	E-W
315	SE-NW

Après avoir créé les champs 'direction' et 'direction_inverse' dans la table 'arcs', il est nécessaire d'implémenter une fonction plpgsql pour y affecter les valeurs de la table de correspondance.

```
-- Création des champs 'dir' et 'dir_inv'
DO $$
    BEGIN
        BEGIN
            ALTER TABLE arcs ADD COLUMN dir character(5);
        EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <dir> already exists
in <arcs>.';
        END;
         BEGIN
```

```
            ALTER TABLE arcs ADD COLUMN dir_inv character(5);
        EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <dir_inv> already
exists in <arcs>.';
        END;
    END;
$$;
```

Remarque : dans le script du modèle, ces deux champs sont créés respectivement lors du calcul des champs 'azimut' et 'azimut_inverse' pour des raisons de lisibilité de la table "arcs".

```
-- Création de la table de correspondance
DROP TABLE IF EXISTS azimut;
CREATE TABLE azimut
(
  id integer,
  az_value integer,
  az_label character(5)
)
WITH (
 OIDS = FALSE
);
INSERT INTO azimut
VALUES
(1, 0, 'S-N'),
(2, 45, 'SW-NE'),
(3, 90, 'W-E'),
(4, 135, 'NW-SE'),
(5, 180, 'N-S'),
(6, 225, 'NE-SW'),
(7, 270, 'E-W'),
(8, 315, 'SE-NW');
-- Mise à jour des champs 'dir' et 'dir_inv'
UPDATE arcs
    SET
        dir = azimut.az_label
    FROM azimut
    WHERE azimut = azimut.az_value;
UPDATE arcs
    SET
        dir_inv = azimut.az_label
    FROM azimut
    WHERE azimut_inv = azimut.az_value;
```

Entrée : arcs (depuis PostGIS)
Sorties : azimut, champs de direction dans la couche arcs (dans PostGIS)

v. Phase 5 : calcul de pondération induite par le vent

L'objectif de cette phase est de déterminer l'allure d'un drone en fonction de sa position par rapport au vent. Cette opération permet de connaître la contrainte de déplacement qui s'applique sur le drone pour les deux sens de parcours de chaque arc de la zone d'étude.

Cette contrainte de déplacement vient alors s'ajouter à la pondération pré-existante des deux sens d'arcs calculée lors de la *Phase 3 : calcul de pondération induite par la bathymétrie*.

Module d'analyse multicritères - phase 5 : définition des allures des drones

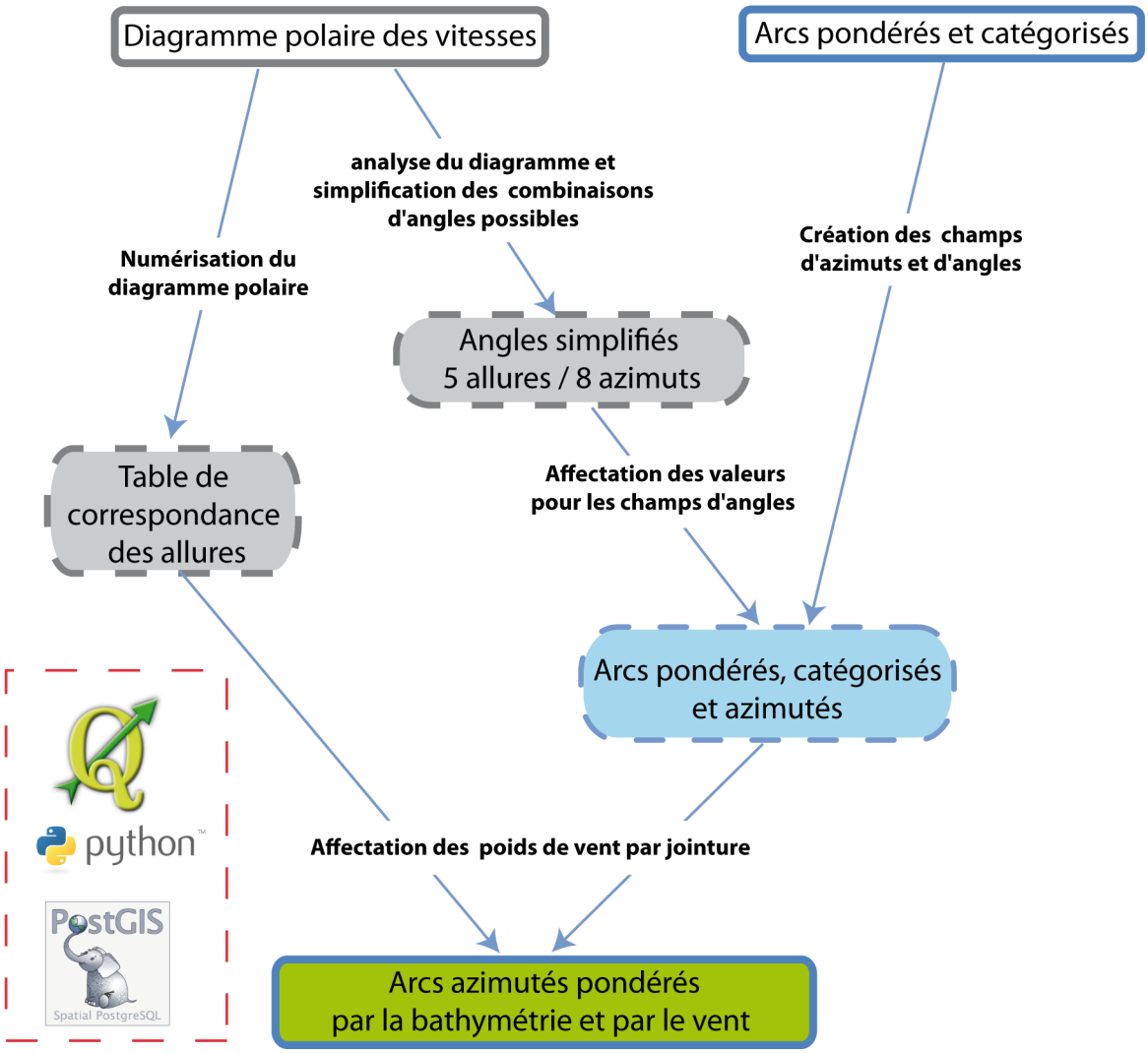


Illustration 19: Préparation à la navigation - chaîne de traitements phase 5

Pour connaître l'allure d'un drone en fonction du vent, nous nous appuyons sur son diagramme polaire des vitesses. Celui-ci permet de connaître l'allure d'un bateau en fonction de sa position par rapport à la direction, le sens et la vitesse du vent.

Chaque bateau a généralement un diagramme polaire qui lui correspond. N'ayant pas encore de données sur le comportement d'un drone Protei, nous nous basons ici sur la polaire d'un voilier de 7m.

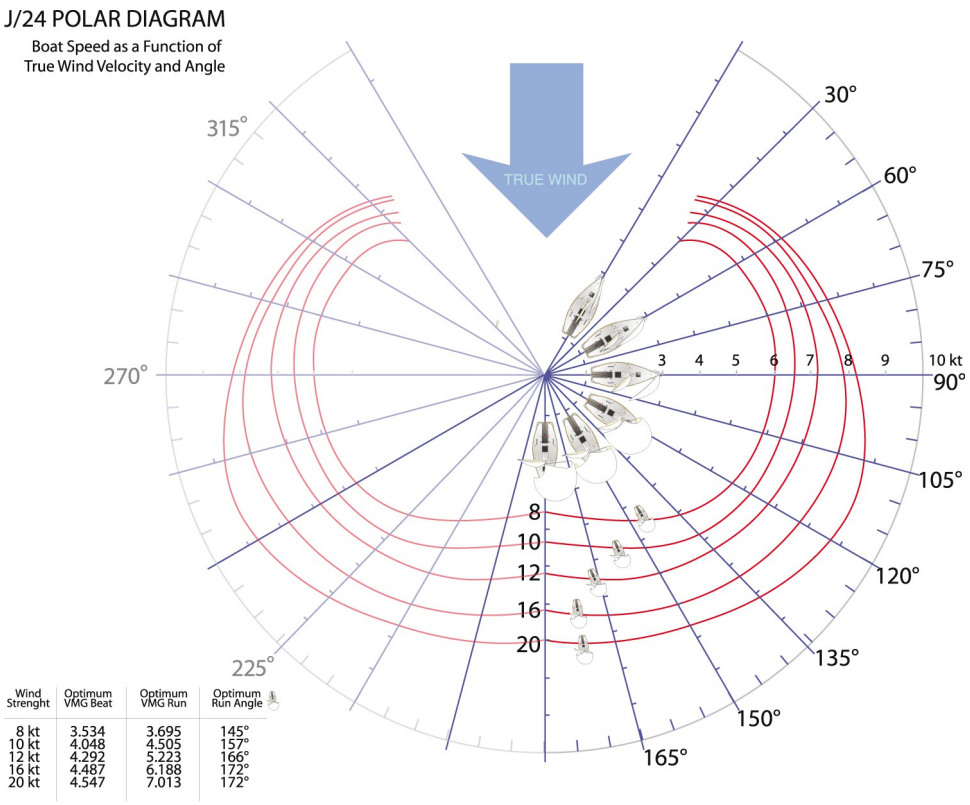


Illustration 20: Préparation à la navigation - diagramme polaire d'un voilier de 7m

Paramètres d'entrée :

- un azimut et un azimut inverse pour chaque entité de la couche "arcs" générés lors de la *Phase 3 : calcul de pondération induite par la bathymétrie*
- un azimut pour le vent
- une vitesse de vent

L'objectif est donc de définir, à partir du diagramme polaire, la vitesse du drone en fonction de ces trois paramètres d'entrée et donc un poids de parcours d'arc. Ainsi, plus la vitesse est élevée, plus le poids est faible car le drone a moins de contraintes à se déplacer sur l'arc. Au contraire, si la vitesse du drone est faible, cela signifie qu'il est plus difficile de se déplacer : le coût augmente alors.

Cette phase se déroule ainsi selon les trois étapes suivantes :

1. Mise en place de l'algorithme de calcul d'angle du diagramme polaire. Celui-ci traduit l'angle formé par l'azimut de vent et celui du drone afin de correspondre avec un des angles présents dans le diagramme polaire. Il est alors possible de traduire l'ensemble des combinaisons d'azimuts de vent ou du drone par seulement quelques angles
2. Création d'une table d'association "allure" qui consiste en une "numérisation" du diagramme polaire représenté plus haut
3. Affectation des allures sur la couche "arcs" en fonction des champs 'azimut' et 'azimut_inv'

L'idée est donc de connaître, pour chaque combinaison d'azimut drone/vent, l'angle qui lui correspond. A partir des données d'entrée de vitesse et d'azimut de vent, il est alors possible de connaître l'allure du drone sur chaque arc.

a) Étape 1 : PostGIS - algorithme de calcul d'angle entre un drone et le vent

Ce diagramme décrit le comportement du drone selon différentes vitesses de vent pour une direction Nord-Sud (N-S), soit pour un azimut de 180°.

Pour mieux comprendre ce diagramme, nous allons donner quelques exemple d'allure en se basant

sur cet azimut de vent :

Azimut vent (°)	Azimut drone (°)	Vitesse vent (noeuds)	Vitesse drone (noeuds)
180	90	8	6
180	135	12	6.5
180	180	10	4.3

Le modèle d'analyse multicritères ne peut prendre en compte que 8 azimuts de 0 à 315° avec un pas de 45° conformément à la couche d'arcs issue de la *Phase 3 : calcul de pondération induite par la bathymétrie*, ce qui limite le nombre de combinaisons.

- **Symétrie axiale de la polaire**

Pour simplifier le nombre de combinaisons possibles d'azimuts du drone, il faut noter la symétrie verticale du diagramme polaire qui divise environ par 2 le nombre d'allures possibles pour l'azimut de vent considéré.

Ainsi, en prenant en compte le demi axe vertical [O,y) comme position de départ (soit un azimut de vent 'N-S' à 180°), chaque azimut du drone va former un angle qui se traduit par une allure selon la vitesse de vent. Cela évite aussi d'afficher la partie semi-transparente de gauche du diagramme polaire qui représente finalement la même information que la partie de droite.

La table suivante résume ainsi cette simplification des résultats :

Azimut vent (°)	Azimut drone (°)	angle demi axe [0,y) - azimut drone
180	0	**0**
180	45	**45**
180	90	**90**
180	135	**135**
180	180	**180**
180	225	**135**
180	270	**90**
180	315	**45**

Pour appliquer cette simplification de calcul d'allure sur la couche des arcs, on passe par un calcul trigonométrique qui transforme l'azimut en cosinus puis en arccosinus (sa fonction inverse) pour obtenir l'angle exprimé dans la colonne de droite du tableau. C'est ainsi un moyen de limiter le nombre d'entrées dans la table "angle_drone_vent".

$$angle<[O,y)-azimut\_drone> = \cos^{-1}(\cos((azimut\_drone)*PI/180))*180/PI$$

Cette simplification est le moyen de faire correspondre l'orientation d'un drone avec son allure pour un azimut et une vitesse de vent définis, soit 5 vitesses possibles pour 8 azimuts.

- **Application de l'algorithme à l'ensemble des azimut du vent**

Pour généraliser le calcul des allures du drone, il faut également faire varier l'azimut du vent. Ce calcul revient à effectuer à chaque fois une rotation du demi axe [0-y) de 45°.

On remarque finalement que l'angle formé entre l'azimut du vent et du drone est le même que l'écart Δ = 180° - azimut du vent.

On peut ainsi définir la formule trigonométrique finale de calcul d'angle entre l'azimut du vent et celui du drone, quels que soient leur valeur respective sur les huit combinaisons possibles :

$$angle<azimut\_vent,azimut\_drone> = cos^{-1}(cos((azimut\_drone + 180° - azimut\_vent)*PI/180))*180/PI$$

- **Intégration de l'algorithme sous PostGIS**

A partir de l'algorithme défini plus haut, deux champs 'angle_dir' et 'angle_ind' sont créés pour déterminer l'angle formé par l'azimut de vent - saisi en paramètre d'entrée - avec chaque sens d'arc, et ce pour tous les arcs du graphe de parcours.

Le script pyQGIS suivant permet de prendre en compte l'azimut de vent saisi pour le calcul d'angle :

```
#Creation des champs d'angle entre l'azimut de vent et l'azimut de chaque arc
dans les deux sens de parcours
outputs_0=processing.runalg("gspg:postgisexecutesql", 'Protei', "DO $$ \
    BEGIN \
        BEGIN \
            ALTER TABLE arcs ADD COLUMN angle_dir int; \
        EXCEPTION \
            WHEN duplicate_column THEN RAISE NOTICE 'column <angle_dir> already
exists in <arcs>.'; \
       END; \
        BEGIN \
           ALTER TABLE arcs ADD COLUMN angle_ind int; \
       EXCEPTION \
           WHEN duplicate_column THEN RAISE NOTICE 'column <angle_ind> already
exists in <arcs>.'; \
       END; \
   END; \
$$; \
/* Calcul des angles dans les deux sens de parcours de l'arc par la prise \
en compte de l'azimut de vent en entree */ \
UPDATE arcs \
SET \
    angle_dir = acos(cos((azimut+180-" + azimut_vent + ")*Pi()/180))*180/Pi(), \
    angle_ind = acos(cos((azimut_inv+180-" + azimut_vent +
")*Pi()/180))*180/Pi();")
```

Remarque : la valeur %azimut_vent% est récupérée comme paramètre d'entrée du modèle global de la chaine de traitements.

Entrée : arcs (depuis PostGIS)
Sortie : champs d'angle de la couche arcs (dans PostGIS)

b) Étape 2 : PostGIS - création de la table "allure"

Cette table d'association, sous sa forme "numérisée", reprend les allures décrites dans le diagramme polaire. D'après l'algorithme de simplification défini lors de la première étape, ce sont 5 angles qui sont reportés dans cette table : 0°, 45°, 90°, 135° et 180°.

Le nombre d'allures, selon cette configuration, est donc le résultat combinatoire des 5 angles avec les 5 vitesses de vent (8, 10, 12, 16 et 20 noeuds), soit 25 allures possibles.

La requête SQL suivante permet de créer la table :

```
DROP TABLE IF EXISTS allure;
CREATE TABLE allure
(
  id integer,
  speed_vent integer,
  angle integer,
  speed_drone float,
  poids float)
WITH (
 OIDS = FALSE
);
INSERT INTO allure (id, speed_vent, angle, speed_drone) VALUES
(1, 8, 0, 0.001), (2, 8, 45, 5.5), (3, 8, 90, 6), (4, 8, 135, 5.1), (5, 8, 180,
3.6),
(6, 10, 0, 0.001), (7, 10, 45, 6), (8, 10, 90, 6.5), (9, 10, 135, 5.8), (10, 10,
180, 4.3),
(11, 12, 0, 0.001), (12, 12, 45, 6.1), (13, 12, 90, 7.1), (14, 12, 135, 6.4),
(15, 12, 180, 5.2),
(16, 16, 0, 0.001), (17, 16, 45, 6.8), (18, 16, 90, 7.9), (19, 16, 135, 7.5),
(20, 16, 180, 6.1),
(21, 20, 0, 0.001), (22, 20, 45, 7), (23, 20, 90, 8.1), (24, 20, 135, 8.1), (25,
20, 180, 7);
UPDATE allure SET poids = 1/speed_drone;
```

Le poids est ici défini par la fonction inverse de l'allure du drone. En effet, le poids est un facteur de contrainte : plus le poids est élevé, plus le drone avance lentement.

Entrée : néant
Sortie : table de correspondance allure (dans PostGIS)

c) Étape 3 : PostGIS - affectation des poids d'allure sur chaque entité de la couche "arcs"

Cette dernière étape permet d'affecter le poids de chaque arc de la zone d'étude pour chaque sens de circulation. Il s'agit donc, à partir des deux paramètres de vent en entrée (azimut et vitesse), d'associer :
l'angle défini par l' azimut de vent et le cap du drone qui a été calculé par le biais de l'algorithme précédent
la valeur de poids inscrite dans la table d'association "allure".

Ainsi, les poids relatifs à l'allure sur les deux sens ('pds_vent_d' et 'pds_vent_i') de chaque arc seront ajoutés à la couche "arcs".

Cette étape est réalisée à partir du script pyQGIS suivant :

```
#Affectation de la ponderation du vent pour les deux sens d'arc en fonction de
l'azimut de vent, la vitesse de vent, et l'azimut de l'arc
outputs_2=processing.runalg("gspg:postgisexecutesql", 'Protei', "DO $$ \
    BEGIN \
        BEGIN \
            ALTER TABLE arcs ADD COLUMN pds_vent_d float; \
        EXCEPTION \
            WHEN duplicate_column THEN RAISE NOTICE 'column <pds_vent_d> already
exists in <arcs>.'; \
        END; \
        BEGIN \
            ALTER TABLE arcs ADD COLUMN pds_vent_i float; \
        EXCEPTION \
            WHEN duplicate_column THEN RAISE NOTICE 'column <pds_vent_i> already
exists in <arcs>.'; \
```

```
        END; \
    END; \
$$; \
UPDATE arcs \
SET pds_vent_d = allure.poids \
FROM allure \
WHERE speed_vent = 8 \
AND angle = angle_dir; \
UPDATE arcs \
SET pds_vent_i = allure.poids \
FROM allure \
WHERE speed_vent = " + vitesse_vent + " \
AND angle = angle_ind;")
```

Remarque : la valeur %vitesse_vent% est récupérée comme paramètre d'entrée du modèle global de la chaine de traitements

Entrée : arcs, allure (depuis PostGIS)
Sortie : champs de pondération du vent de la couche arcs (dans PostGIS)

vi. Phase 6 : calcul de l'itinéraire optimal de la zone à couvrir

L'objectif de cette phase est de déterminer l'itinéraire optimal d'un drone en fonction des conditions de navigation qui incluent la bathymétrie et les contraintes dues au vent.

Le modèle de cette phase est donc constitué des étapes suivantes :

1. Calcul du coût total par arc en fonction de la bathymétrie, du vent et de la longueur de l'arc
2. Calcul du chemin le plus court entre chaque point de passage permettant de déterminer l'itinéraire le moins coûteux
3. Export de l'itinéraire optimal

Module d'analyse multicritères - phase 6 : détermination du parcours idéal

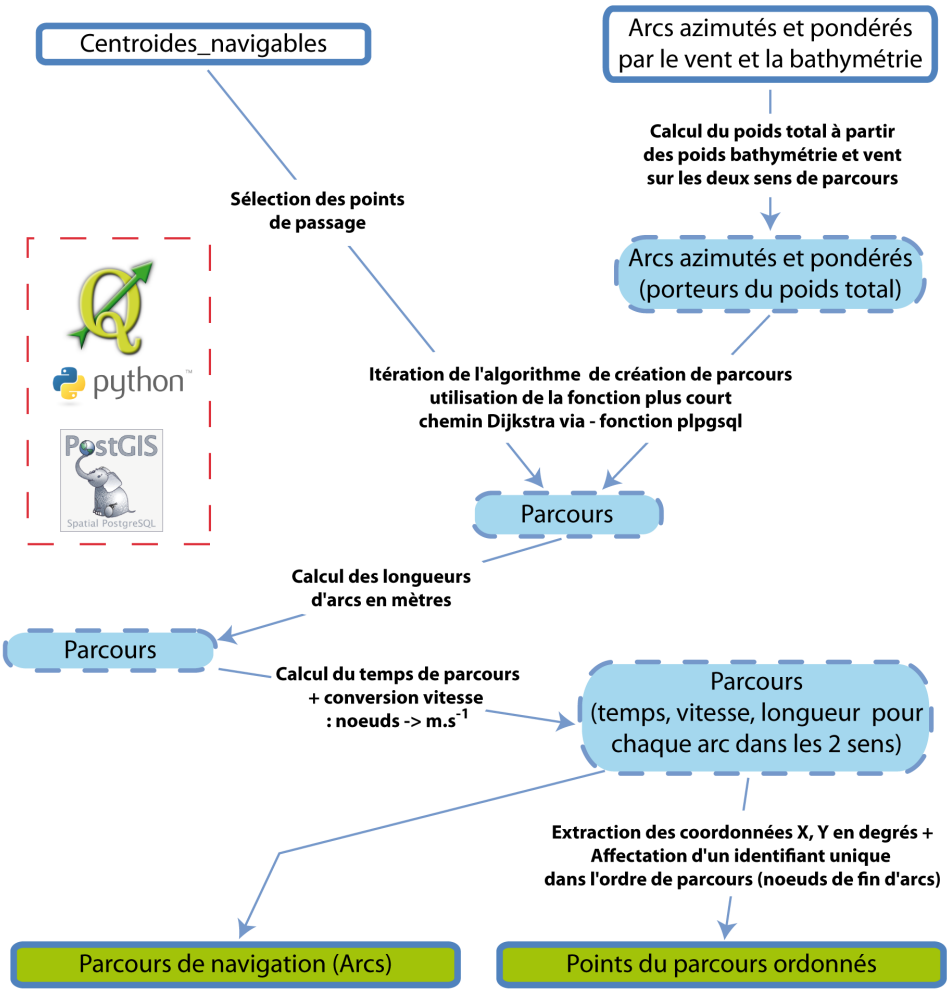


Illustration 21: Préparation à la navigation - chaîne de traitements phase 6

a) Étape 1 : Calcul du coût total pour chaque sens d'arc

Cette étape finalise le calcul des coûts de chaque arc afin d'exécuter l'algorithme du plus court chemin.

Ce calcul prend en compte pour chaque sens d'arc :

- la longueur de l'arc exprimée en degrés décimaux
- le poids dû aux contraintes de vent exprimé par l'inverse de la vitesse du drone en noeuds
- le poids de la bathymétrie pour les arcs proches des obstacles. Ce facteur n'a pas vraiment de sens physique, mais joue surtout le rôle de répulseur aux abords des obstacles

Le coût total définit donc un temps de parcours pour chaque sens d'arc. Cependant, ce temps de parcours est un calcul intermédiaire qui ne prend pas en compte la distance métrique des arcs. Le calcul du coût total se fera donc en deux étapes qui s'intercaleront avant et après l'algorithme du plus court chemin. Ce choix découle du système de coordonnées sur lequel le modèle est basé et est expliqué en détail dans la partie .

En projetant la couche de bathymétrie dans un référentiel type Mercator, il apparaît que chaque cellule de la grille de navigation n'est en réalité pas carrée, mais rectangulaire. En d'autres termes, la largeur et la hauteur ont la même longueur d'angle, mais pas la même longueur métrique.

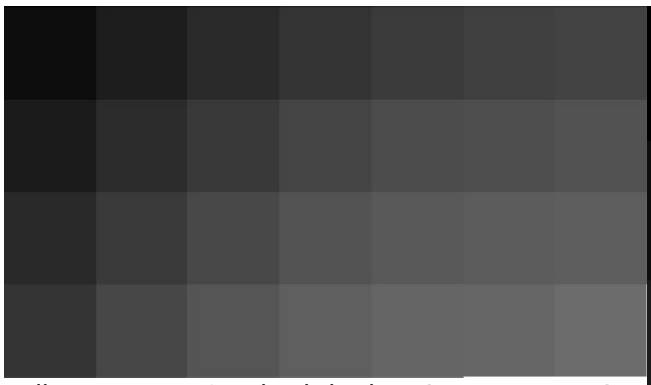


Illustration 22: Couche de bathymétrie non projetée (WGS84)

Ici, le pixel a les dimensions 0.009° x 0.009°

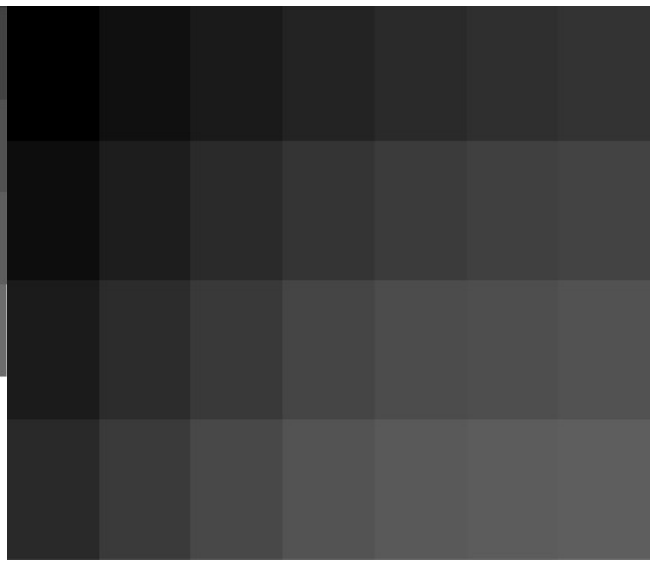


Illustration 23: Couche de bathymétrie projetée (Web Mercator)

Ici, le pixel a les dimensions 670m x 1000m

Cela induit donc un biais significatif pour le déplacement du drone sur les arcs. En effet, cela signifie qu'il lui faut parcourir une distance beaucoup plus importante sur les arcs orientés verticalement qu'horizontalement. Dans le cas des projections Mercator, plus la zone d'observation est éloignée de l'équateur, plus le ratio entre la hauteur et la largeur s'accentue.

Par conséquent, le coût engendré par la différence de longueur d'arc incitera le drone à se déplacer horizontalement, même dans le cas d'un vent défavorable. L'idée consiste donc à isoler la longueur d'arc dans le calcul du plus court chemin, puis de la prendre en compte dans le calcul du temps de parcours total.

- **Calcul du coût total de parcours par sens d'arc**

La formule suivante est appliquée :

```
poids total = poids bathy + poids vent
```

Il est ainsi nécessaire de créer deux champs de coût par arc pour prendre en compte les deux sens de parcours du graphe. Cela se traduit donc en SQL par les requêtes suivante :

```
-- Création des champs 'cout' et 'cout_inv';
DO $$
    BEGIN
        BEGIN
            ALTER TABLE arcs ADD COLUMN cout float;
      EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <cout> already
exists in <arcs>.';
        END;
        BEGIN
            ALTER TABLE arcs ADD COLUMN cout_inv float;
      EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <cout_inv> already
exists in <arcs>.';
        END;
    END;
$$;
-- Calcul des coûts;
UPDATE arcs
```

```
  SET
cout = (pds_direct + pds_vent_d),
cout_inv = (pds_indir + pds_vent_i);
```

Entrée : arcs (depuis PostGIS)
Sortie : champs de poids total de la couche arcs (dans PostGIS)

b) Étape 2 : Calcul du plus court chemin

Cette étape permet de calculer l'itinéraire le plus court en fonction des conditions de navigation (bathymétrie + vent) à partir des différents points de passage successifs du pattern de navigation théorique.

Ce calcul de parcours optimal se base donc sur les poids affectés à chaque arc et dans chaque sens entre deux points de passage successifs, et s'appuie sur l'algorithme de Dijkstra.

- **Principe de l'algorithme Dijkstra**

L'algorithme de Dijkstra , dans la théorie des graphes, sert à résoudre le problème du plus court chemin entre un point d'origine et un point de destination. Dans le cas présent, le drone se déplace depuis le premier point de passage vers le second en utilisant les arcs, l'objectif étant d'emprunter les arcs qui sont le moins coûteux en temps.

Pour plus d'informations sur cet algorithme, une documentation est disponible :
http://fr.wikipedia.org/wiki/Algorithme_de_Dijkstra

- **pgr_Dijkstra : algorithme de plus court chemin dans pgrouting**

L'algorithme de plus court chemin est opéré par la fonction *pgr_dijkstra* comprise dans l'extension pgrouting 2.0 de PostGIS.

Pour plus d'informations sur cette fonctionnalité :
http://docs.pgrouting.org/dev/src/dijkstra/doc/index.html

c) Calcul sur l'ensemble des points de passage

Pour effectuer le calcul du plus court chemin sur l'ensemble des points de passage, le modèle embarque une fonction plpgsql qui s'appuie sur un langage procédural au sein de requêtes SQL sous PostgreSQL. (*http://www.postgresql.org/docs/9.3/static/plpgsql.html*)

Cette fonction effectue une itération sur l'ensemble des points passage répertoriés dans la table "centroïdes_navigables" et se découpe selon les étapes suivantes :

1 si c'est le premier point de passage, alors il est considéré comme point de départ
2 Sinon :
  2.1 le point de passage est considéré comme point de destination
  2.2 l'ensemble des arcs qui constituent le chemin le plus court entre les points de départ et d'arrivée courants sont enregistrés dans la table "parcours". Pour illustrer le sens de parcours du drone, il est nécessaire d'enregistrer chaque géométrie d'arc dans le même sens :
    2.2.1 si le drone parcourt l'arc dans le même sens que la géométrie de l'arc généré lors de la *Phase 2 : création des arcs de navigation*, alors cette géométrie est conservée
    2.2.2 sinon, le sens de l'arc est inversé
  2.3 Un identifiant unique est généré pour chaque arc
  2.4 Le premier arc sélectionné pour la prochaine itération aura l'identifiant suivant du dernier arc sélectionné lors de cette étape
  2.5 le point d'arrivée courant devient le point de départ pour l'itération suivante
3 Nouvelle itération qui reprend l'étape 2

L'ensemble de ces étapes est réalisé par la requête plpgsql suivante :

```
DROP TABLE IF EXISTS parcours;CREATE TABLE parcours(gid integer, noeud_dep
integer, id_noeud_dep_arc integer, cout double precision, geom geometry)WITH
(OIDS=FALSE);
CREATE OR REPLACE FUNCTION parcours_drone() RETURNS VOID AS $$
DECLARE
    wp centroïdes_navigables.id%TYPE;
    iterator integer = 1;
    id_dep integer;
    id_dest integer;
    affected_rows integer = 0;
    nb_arcs integer = 0;
BEGIN
    FOR wp IN
        SELECT id FROM centroïdes_navigables WHERE pt_passage IS NOT NULL ORDER
BY  pt_passage
    /*Iteration sur l'ensemble des points de passage */
LOOP
        IF iterator = 1 THEN
            id_dep = wp;
        ELSE
            id_dest = wp;
           /* Calcul du plus court chemin entre deux points de passage -
assimilee a une etape */
           WITH etape AS (INSERT INTO parcours SELECT seq+nb_arcs AS gid, node
AS noeud_dep, arcs.id_dep AS id_noeud_dep_arc, cout,
       /* si le drone parcourt l'arc dans le meme sens que l'arc trace en etape
2, on garde la même geometrie. Sinon on inverse le sens */
           CASE WHEN arcs.id_dep=node
        THEN geom
            ELSE
                st_reverse(geom)
            END as geom
            FROM arcs,
        /*Calcul du plus court chemin entre deux points de passage */
                (SELECT seq, id1 AS node, id2 AS edge, cost
                FROM pgr_dijkstra('SELECT gid AS id, id_dep::int4 AS source,
                id_dest::int4 AS target, cout::double precision AS cost,
cout_inv::double precision
                AS reverse_cost FROM arcs',id_dep,id_dest, true, true)) AS
dijkstra
            WHERE arcs.gid = dijkstra.edge RETURNING 1)

/* calcul du nombre d'arcs empruntes par le drone lors de cette etape */
            SELECT COUNT(*) INTO affected_rows FROM etape;
        END IF;
        /* Reaffectation du nb d'arcs necessaires pour parcourir l'étape afin de
calculer les ids des arcs de la prochaine etape*/
        nb_arcs = nb_arcs + affected_rows;
    /* incremention de l'iterateur */
        iterator = iterator + 1;
        id_dep = wp;
    END LOOP;
 END;
 $$ LANGUAGE plpgsql;
SELECT parcours_drone();
```

Entrées : centroïdes_navigables, arcs (depuis PostGIS)
Sortie : parcours (dans PostGIS)

d) Étape 4 : Calcul du temps total de parcours

Le parcours optimal déterminé par l'algorithme de Dijkstra est basé sur des distances exprimées en degrés, et non en mètres.

En effet, le modèle du module d'analyse multicritères est établi sur la couche de bathymétrie dont les coordonnées sont exprimées dans le référentiel WGS84. Les distances sont donc exprimées en degrés décimaux. L'objectif est donc de passer les unités de longueur des arcs en mètres pour être en mesure de donner des coûts de déplacements en secondes.

- **Conversion des unités**

Le diagramme polaire indique les vitesses de navires en nœuds. Pour obtenir les temps de parcours, il faut donc convertir cette vitesse en ms^{-1} :

$1\ kt = 0.51444444444\ ms^{-1} \rightarrow alpha$

- **Calcul des longueurs d'arcs (en mètres)**

```
-- Création du champ 'longueur';
DO $$
    BEGIN
    ALTER TABLE parcours ADD COLUMN longueur float;
EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <longueur> already
exists in <parcours>.';
        END;
$$;
-- Calcul des longueurs des arcs (en metres);
UPDATE parcours
  SET longueur = ST_Length(geom, TRUE);
```

Entrée : parcours (depuis PostGIS)
Sortie : champ de longueur de la couche parcours (dans PostGIS)

- **Calcul du temps de parcours par arc**

Cette étape permet de calculer le temps de parcours par arc en prenant en compte :

- le coût calculé par l'algorithme du plus court chemin
- la longueur d'arc exprimée en mètres
- le facteur de conversion permettant de convertir les noeuds marins en $m.s^{-1}$

La formule est donc : `duree = cout * longueur / alpha`

Ce calcul s'effectue donc par le biais de la requête SQL suivante :

```
-- Création du champ 'duree';
DO $$
    BEGIN
    ALTER TABLE parcours ADD COLUMN duree float;
EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <duree> already
exists in <parcours>.';
        END;
$$;
-- Calcul de la duree de parcours (en m.s-1);
UPDATE parcours
  SET duree = cost * longueur / 0.51444444444;
```

Entrée : parcours (depuis PostGIS)
Sortie : champ de durée de la couche parcours (dans PostGIS)

- **Calcul du temps total de parcours**

Cette opération découle directement du calcul précédent en faisant la somme de l'ensemble des temps de parcours par arc. Cette durée de parcours est exprimée en secondes.

Cette nouvelle couche de parcours est effectuée par le biais de la requête suivante :

```
/* Enregistrement de l'itineraire dans une entite geographique unique comprenant
la duree totale de parcours en secondes */
DROP TABLE IF EXISTS trace_parcours;
CREATE TABLE trace_parcours AS
SELECT ST_Union(geom) AS geom, SUM(duree) as duree
FROM parcours;
```

Entrée : parcours (depuis PostGIS)
Sortie : trace_parcours (dans PostGIS)

e) Étape 5 : extraction du parcours le plus court en coordonnées GPS

Cette étape consiste à extraire les coordonnées GPS de chaque limite d'arc parcouru par le drone le long de l'itinéraire déterminé par l'algorithme de plus court chemin.

La requête suivante récupère l'ensemble des nœuds de fin d'arc, calcule pour chacun d'eux les coordonnées X et Y en degrés décimaux, et leur affecte un identifiant unique permettant de déterminer un ordre de passage :

```
/* Creation d'une table permettant de generer l'ensemble des points GPS par
lequel le drone devra passer */
DROP TABLE IF EXISTS pts_parcours;
CREATE TABLE pts_parcours AS
SELECT ST_EndPoint(geom) AS geom
FROM parcours;
ALTER TABLE pts_parcours ADD gid serial NOT NULL primary key ;
DO $$
    BEGIN
            ALTER TABLE pts_parcours ADD COLUMN x float;
        EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <x> already exists in
<pts_parcours>.';
        END;
        BEGIN
            ALTER TABLE pts_parcours ADD COLUMN y float;
        EXCEPTION
            WHEN duplicate_column THEN RAISE NOTICE 'column <y> already exists in
<pts_parcours>.';
        END;
    END;
$$;
/* Calcul des coordonnees X,Y de chaque point de parcours */
UPDATE pts_parcours
   SET x = ST_X(geom), y = ST_Y(geom)
```

Entrée : parcours (depuis PostGIS)
Sortie : pts_parcours (dans PostGIS)

E. *Résultats d'analyse et limites*

i. Un outil de simulation plus complet que la première version

Ce modèle de simulation de parcours optimal présente, contrairement à la première version, l'avantage notable de prendre en compte un paramètre de vent selon huit directions et cinq vitesses différentes, soit 25 valeurs de pondération possibles pour ce seul facteur. Couplé au *Module « Stratégies de couverture de zone »*, il permet d'effectuer de nombreux tests de simulation afin de définir le parcours le moins coûteux pour le drone.

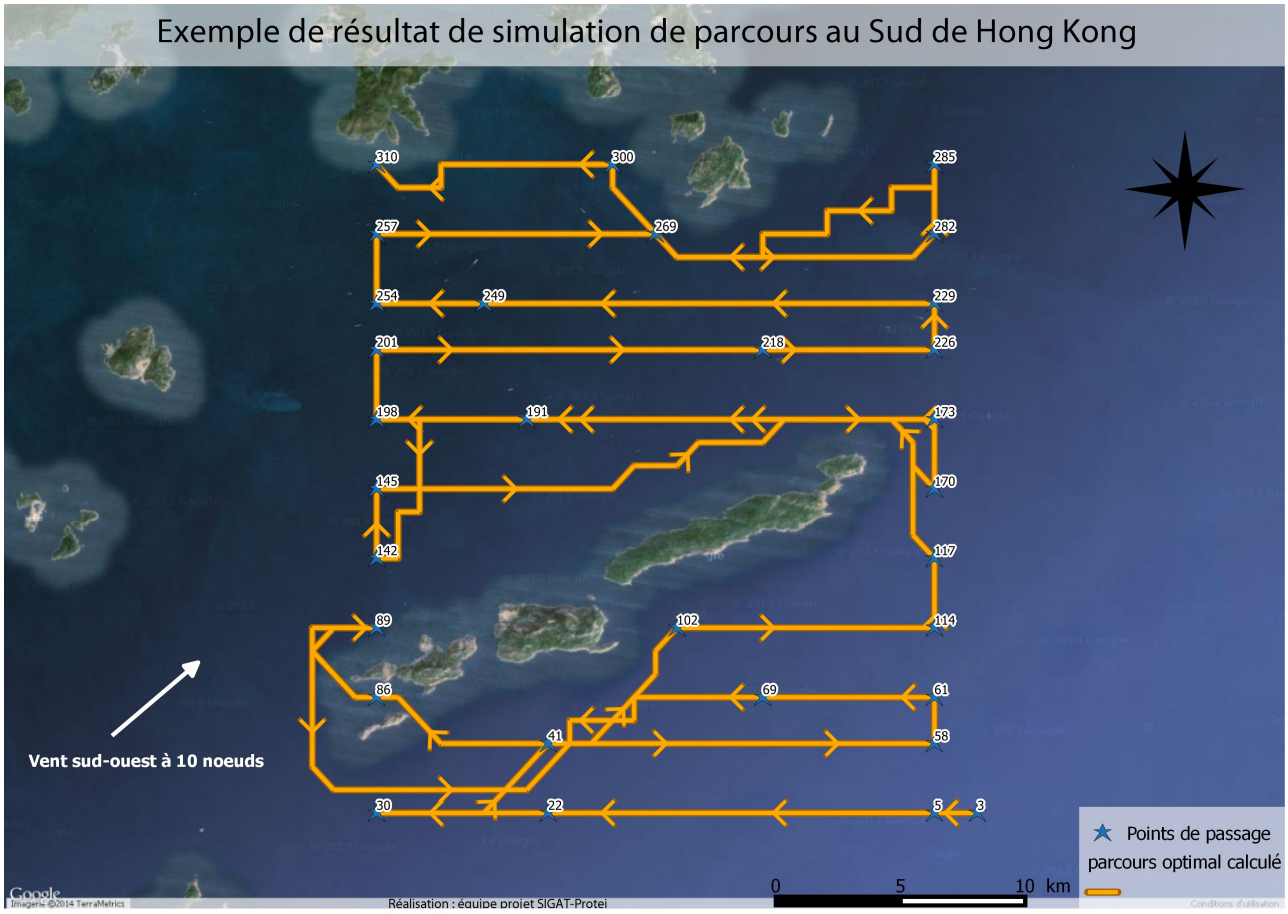


Illustration 24: Préparation à la navigation - exemple de résultats de simulation à Hong Kong

Comme le montre ce résultat de simulation, le modèle respecte également les données de bathymétrie en entrée pour éviter qu'un drone ne passe au travers d'une île. La logique de sécurité est donc ici respectée.

Globalement, ce module est un outil de pilotage visant à définir le parcours d'un drone de manière automatisée. La couche de points de parcours fournie en sortie du modèle définit l'ensemble des coordonnées GPS par lesquels le drone pourra passer pour couvrir la zone. C'est finalement une ouverture pour répondre aux enjeux de la phase d'évolution n°4 du projet Protei qui prévoit de déployer plusieurs drones en même temps à grande échelle, chose plus difficile à entreprendre si le guidage d'un drone ne peut se faire que manuellement.

ii. Des données de bathymétrie de basse résolution

Toutefois, un certain nombre de points restent à aborder pour que le module de préparation à la navigation soit plus conforme aux conditions de navigation réelles.

Premièrement, la donnée bathymétrique disponible en ligne a une résolution très faible - souvent de

l'ordre de 600 voire 800m par pixel - qui va mécaniquement impacter les résultats de simulation.

Dans le cadre du projet, nous avons récupéré une donnée de bathymétrie de l'IFREMER sur le sondage du relief sous marin de la Manche dont la résolution est à environ 600m en longitude et 1000m en latitude. Aussi, le site de l'Institut anglais des données océanographiques (*http://www.bodc.ac.uk/*) nous a permis de récupérer une donnée bathymétrique autour de Hong Kong dont la résolution est d'environ 850m en longitude et 900m en latitude.

Dans l'hypothèse d'une simulation effectuée autour d'un archipel composé de nombreux îlots de petite taille, la donnée bathymétrique en entrée considérera que les îlots de dimension inférieure à sa résolution font partie de la zone de navigation. Dans ce cas de figure, la logique de sécurité des drones n'est alors pas pris en compte dans le modèle.

cependant, l'amélioration des techniques d'acquisition de données bathymétriques (LiDAR, Radar, satellite, sonar, etc.) permettront de réduire les coûts de production et d'accéder à des données de meilleure qualité. A titre d'exemple, Marine Explore (*http://marinexplore.com/*) est une entreprise positionnée sur les problématiques du Big Data qui propose de faciliter les échanges de données océanographiques aux formats souvent hétérogènes. A ce titre, elle développe deux projets, *marineos.com* et *marineplore.org*, qui visent respectivement à organiser et diffuser une large quantité de données à l'échelle du globe. C'est sur ce type d'initiative que le projet Protei peut s'appuyer pour obtenir des données de meilleure qualité.

iii. Une logique de définition des points de passage à revoir

Un autre facteur limitant la qualité des résultats réside dans le choix de points de passage obligatoires pour le drone. En effet, ceux-ci sont définis par le module de stratégies de couverture de zone qui ne prend pas en compte la bathymétrie en entrée du module. Ainsi, les points de passage sont placés arbitrairement sur la zone à couvrir et certains d'entre eux n'ont pas vraiment d'intérêt à être présents. A titre d'exemple, l'illustration du dessus montre que le point de passage situé au sud-ouest oblige le drone à passer au nord de l'ile de Zhiwan avant de revenir au sud pour se rendre à l'est de la zone vers le point de passage suivant.

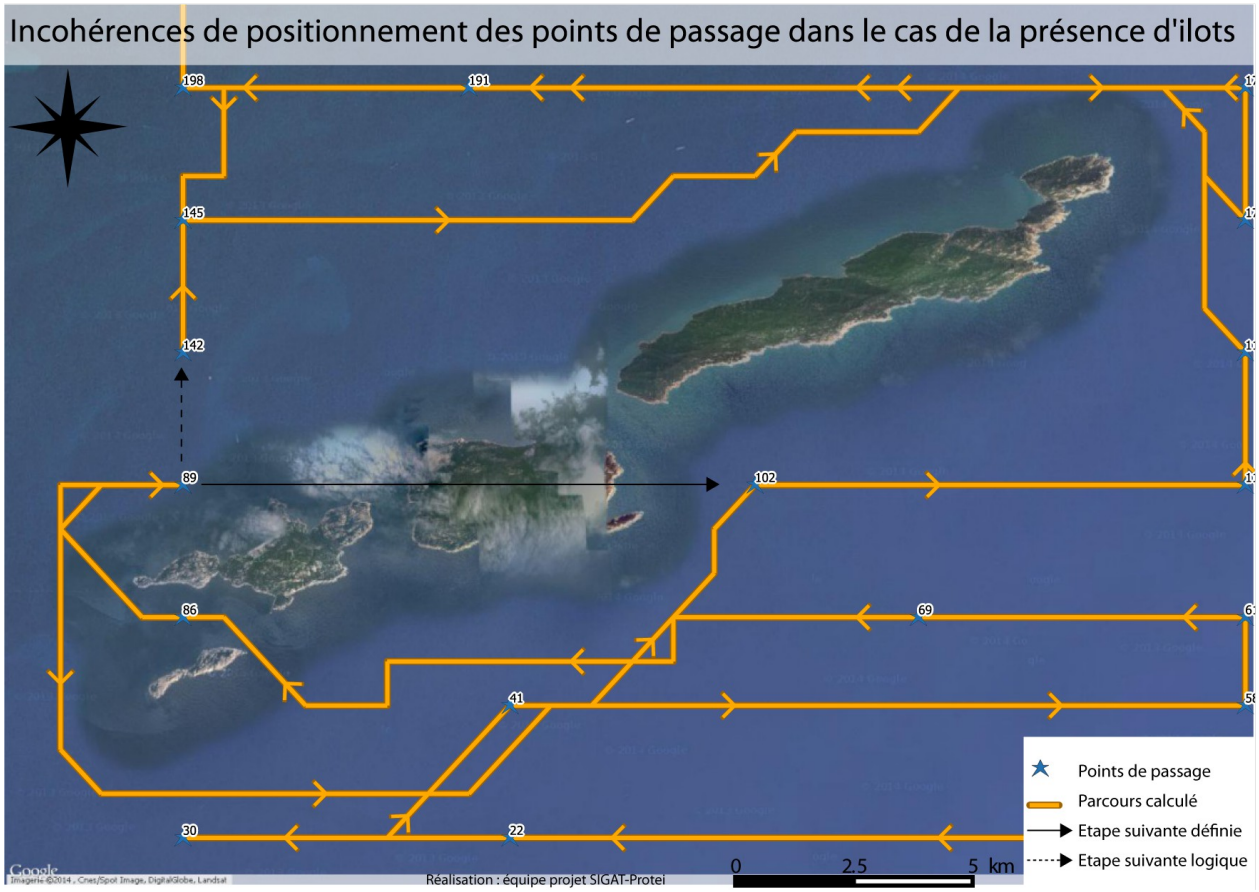


Illustration 25: Préparation à la navigation - exemple d'incohérence de positionnement de points de passage

Dans cet exemple, le drone passe d'abord par le point n°89 avant de se rendre, comme prévu, vers le points de passage n°102. Or, il eut été logique qu'il passe directement au point de passage n°142 situé un peu plus au nord puis repartir vers l'est pour continuer de balayer la zone.

iv. Les limites d'efficacité du drone selon l'orientation du vent

Les résultats de simulation sont très souvent peu satisfaisants dès lors que la stratégie de couverture sélectionnée pour la simulation a une orientation défavorable par rapport à l'orientation du vent. Dans l'exemple ci-dessous, le modèle effectue une simulation avec un vent venant de l'ouest :

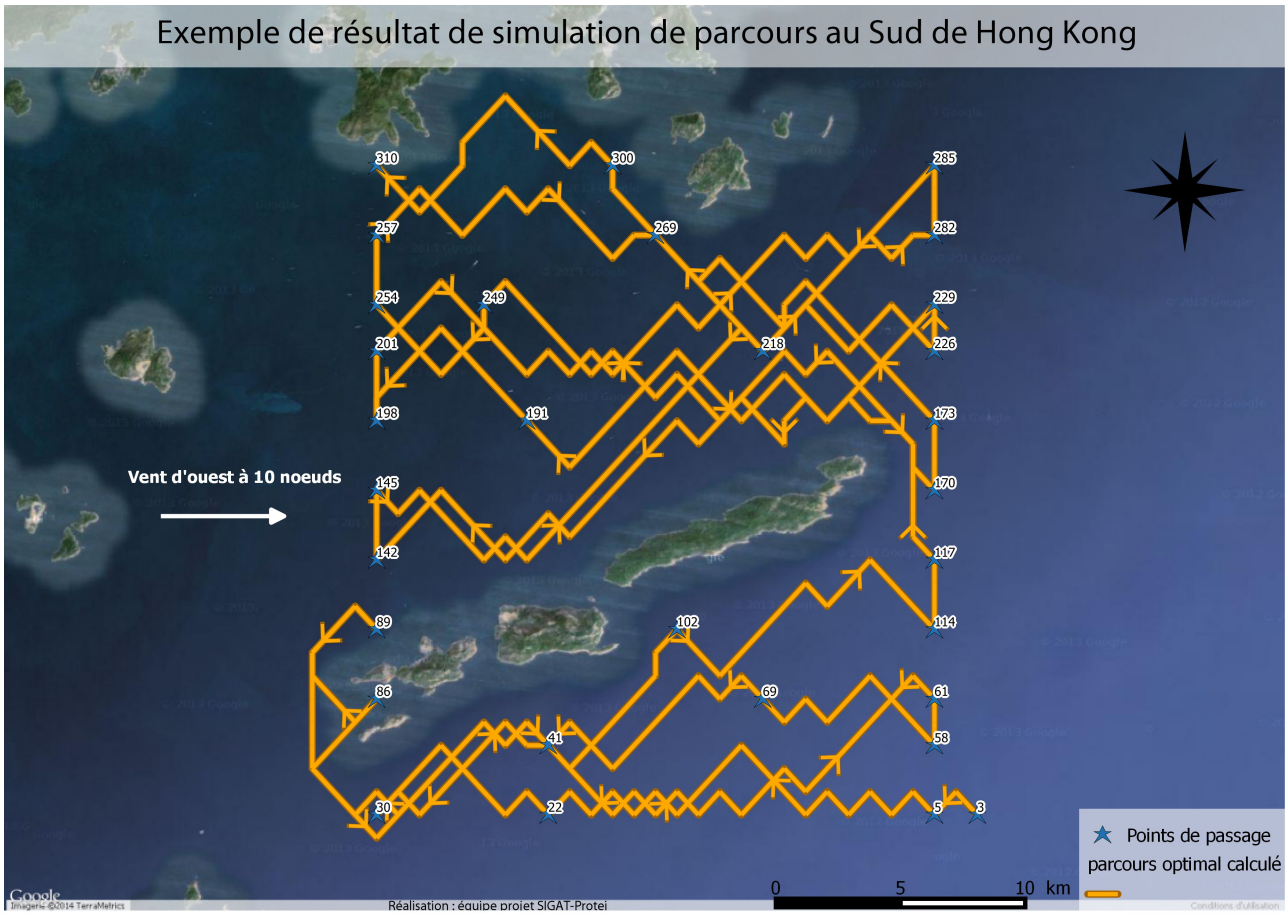


Illustration 26: Préparation à la navigation - simulation avec un vent globalement contraignant

Contrairement au premier résultat affiché, le vent vient ici de l'ouest et contraint le drone à louvoyer lorsqu'il navigue d'est en ouest pour atteindre le point de passage suivant.

Ce manque d'efficacité peut être pallié par rotation du chemin théorique pour éviter que le drone ne se retrouve trop face au vent.

v. La non prise en compte de l'historique des résultats de tests

L'architecture du modèle, en l'état actuel des choses, ne permet pas de gérer l'historique des données générées. Par conséquent, le modèle écrase les résultats de simulation à chaque nouvelle exécution de la chaîne de la traitements.

Pour éviter de perdre les résultats de simulation, il est possible d'exporter la couche de parcours optimal au format Shape. cependant, cette méthode a des limites dans le sens où cela demande de créer un système d'archivage des couches sur sa machine.

F. Perspectives

i. Amélioration du fonctionnement du modèle actuel

a) *Dissociation de la couche bathymétrique de la couche définissant l'emprise de la zone d'étude*

A terme, la bathymétrie pourrait être chargée directement en fonction de l'emprise sélectionnée dans l'interface graphique. Le découpage des zones de bathymétrie serait géré via un système de tuilage.

Pour préparer cette évolution, il convient de définir l'emprise de la zone d'étude dans une couche différente de la bathymétrie.

La *Phase 1 : vectorisation de la zone d'étude* sera donc modifiée pour recevoir en entrée une couche d'emprise (polygone) par laquelle sera découpée la couche bathymétrique (raster).

Préparation à la navigation : optimisation de la phase 1

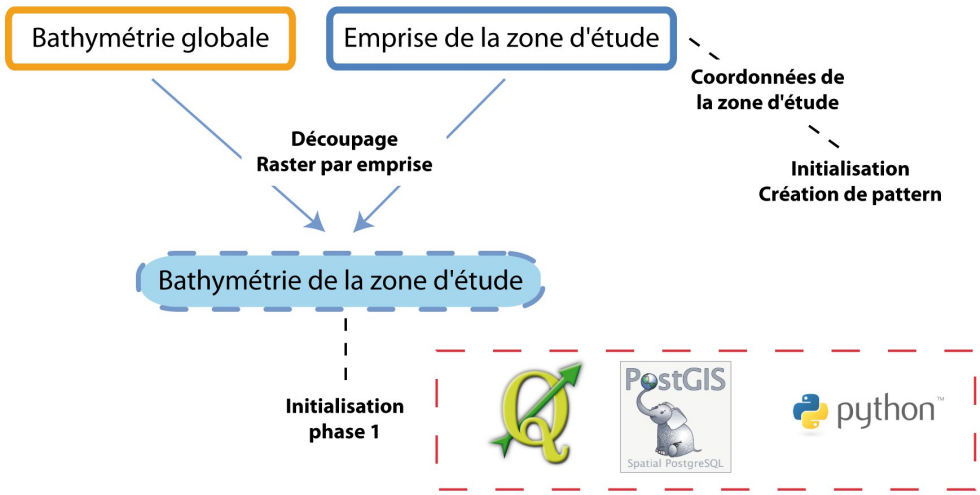


Illustration 27: Préparation à la navigation - Optimisation de la phase 1

b) *Gestion de la temporalité et prise en compte du module déploiement de flotte :*

Pour déployer les drones en flotte, il est nécessaire de définir un itinéraire pour chacun d'entre eux en répétant l'algorithme de parcours de moindre coût, et ce autant de fois qu'il y a de drones. Cela implique donc de conserver et de référencer les chemins définis par le module en fonction :

- de l'identifiant du drone pour lequel il a été calculé
- et du chemin théorique choisi.

Il semble également nécessaire d'introduire un système de version (historisation) du parcours calculé s'il en existe plusieurs pour un même drone.

A terme, ces évolutions pourraient se traduire par l'intégration d'un gestionnaire de parcours dans l'interface graphique de navigation.

c) *Définition de zones de vent*

Dans la *Modèle v2 : analyse de graphe selon la bathymétrie et le vent*, les variables environnementales sont intégrées de manière figée et appliquées à l'ensemble de la zone de navigation. Selon la surface à couvrir et la précision des données météorologiques utilisées, l'intensité et la direction du vent peuvent être différentes pour une même zone d'étude.

Dans le but d'intégrer ce facteur de vent, il est nécessaire de sectoriser la surface étudiée. L'étape de

sectorisation pourrait s'ajouter à l'étape de paramétrage des données de vent et s'opérer de la manière suivante :

- Création d'une couche de polygones représentant le découpage des différentes zones de vent (chaque zone intégrant une orientation et une valeur de vitesse de vent)
- Affectation des directions et vitesses de vent aux arcs en fonction de ces zones.

On cherche à déterminer la zone de vent à laquelle appartient chaque arc par le biais d'une requête spatiale. Une fois la sectorisation opérée, la chaîne de traitements peut être poursuivie, les valeurs de vent étant prises en compte en *Phase 5 : calcul de pondération induite par le vent* par jointure sur la table de correspondance azimut / direction de vent.

d) Enrichissement de la polaire des vents

Dans sa version actuelle, le modèle n'intègre qu'une version simplifiée de la polaire des vents. En effet, seules 8 directions et 5 vitesses de vent peuvent être passées en paramètres . L'augmentation du nombre de directions possibles(en passant par exemple de 8 à 16 directions possibles pour chaque nœud) permet d'augmenter le nombre de routes pouvant être prises depuis chaque nœud et ainsi affiner la précision de l'algorithme de parcours de graphe.

e) Optimisation du positionnement des points de passage

Dans le modèle tel qu'il est conçu à présent, le positionnement des points de passage est conditionné uniquement au traçé du parcours idéal défini dans le *Module « Stratégies de couverture de zone »* et contraint par la bathymétrie.

Ainsi, dans le cas où un point de passage est situé sur une zone non navigable ou dangereuse, il est automatiquement supprimé lors de la *Phase 1 : vectorisation de la zone d'étude*. Pour optimiser le positionnement des points et palier à ce problème de suppression, il conviendrait de s'appuyer sur un algorithme qui viserait à déplacer latéralement le point de passage plaçé sur un obstacle pour limiter les zones non couvertes :

Pour tout point de passage qui intersecte la couche d'obstacles :
il faudrait sélectionner le point, récupérer ses coordonnées x, y, récupérer le rayon (r) du buffer de sécurité autour de l'obstacle concerné (calculé en *Phase 3 : calcul de pondération induite par la bathymétrie*) et effectuer une translation horizontale d'un vecteur égal au rayon (r) (r,0). De là, il est possible de déterminer les nouvelles coordonnées du point et de mettre à jour l'entité sélectionnée.

Pour aller plus loin, il serait intéressant de faire une analyse spatiale de la zone de navigation, de déterminer la surface à couvrir, et d'estimer quelle serait la distance maximum tolérée entre deux points de passage. Il s'agirait ensuite de générer des points de passage en s'appuyant sur le *Calcul d'efficacité*. La distance entre les points de passage pourrait être déterminée selon un seuil de tolérance dépendant de l'impact du vent :

- si le vent risque de dérouter le navire de sa trajectoire, les points de passage visant à contraindre le parcours devraient être plus rapprochés
- A l'inverse, si les conditions de navigation sont bonnes et que la mer n'est pas trop forte, les points de passage pourraient être plus espacés

f) Optimisation du modèle : privilégier les traitements côté python

L'architecture du modèle s'appuie sur des phases qui se composent principalement de requêtes SQL vers la base de données 'Protei'.

Néanmoins, de nombreux traitements peuvent être implémentés en python (création, mise à jour d'un champ de table, géo-traitements). Cela permettrait ainsi de limiter le nombre d'aller-retours entre le module et la base de données et d'ainsi d'optimiser les temps de calcul.
De plus, il serait intéressant de gérer côté python la session de connexion avec la base de données

'Protei'. Dès lors, il serait possible de mettre en place un système transactionnel. L'ensemble des modifications d'une phase seraient alors validées dans la base de données seulement si toutes les étapes se sont déroulées correctement.

ii. Amélioration des algorithmes

a) Mise en place de l'algorithme du voyageur de commerce - ATSP

A la différence de l'algorithme de Dijkstra, l'algorithme de voyageur de commerce est en mesure de calculer le plus court chemin sur l'ensemble des points de passage sans avoir besoin d'en définir l'ordre au préalable. Au regard des critiques formulées lors de l'analyse des résultats, le modèle serait alors grandement optimisé et le drone serait alors moins contraint à suivre le chemin théorique de couverture.

A l'heure actuelle, cet algorithme est embarqué dans différents logiciels, mais il est systématiquement proposé dans sa version symétrique. En d'autres termes, cela signifie qu'il ne permet pas de prendre en compte une pondération par sens d'arc. Or, cet aspect est fondamental pour le modèle car un drone n'aura pas la même vitesse selon son allure (sens de déplacement - ex : vent debout vs vent arrière).

Néanmoins, des recherches sont effectuées par des développeurs contributeurs pour être en mesure d'intégrer cette option à l'extension *pgrouting* de PostGIS. Dans l'hypothèse que cette nouvelle fonctionnalité soit disponible, il serait pertinent de l'intégrer à la chaîne de traitements pour optimiser le calcul de parcours optimal.

b) Choix d'une fonction sur laquelle s'appuie le coût du vent

A l'heure actuelle, la pondération affectée au facteur de vent est attribuée selon une fonction inverse qui suppose que plus les conditions de vent sont fortes, plus le poids affecté au facteur de vent est important. Une autre fonction permettrait peut être de mieux approcher les effets de vent mais cela suppose la réalisation d'un travail bibliographique plus avancé.

iii. Ajout de paramètres

a) Intégration des courants marins

Dans la version du modèle présentée ici, la prise en compte du vent sert de test à l'intégration de paramètres environnementaux dynamiques. Ainsi, les courants n'y sont pas pris en compte.

Dans une version ultérieure, il conviendrait d'intégrer ce paramètre qui a une influence massive sur la navigation d'un navire et particulièrement d'un voilier. Cela supposerait de passer par des partenariats avec les organismes producteurs des cartes de navigation marine pour chaque zone où les drones seraient positionnés.

b) Intégration d'une carte des routes maritimes

Dans une version statique du modèle il est possible de prendre en compte les zones de fort traffic maritime en discriminant les zones de navigation situées dans des couloirs ou croisant des routes de navigation comme le rail d'Ouessant par exemple.

ces données peuvent être retrouvées sur : https://www.marinetraffic.com/fr/

c) Intégration d'une analyse d'images satellites pour estimer la position de la nappe

Sur certains sites de déploiement ou dans le cas de certaines catastrophes, l'événement dure suffisamment longtemps pour que des images satellites couvrant la zone soient disponibles.

A la condition où les images aient une résolution suffisante et couvrent bien la zone d'étude, il peut être intéressant d'intégrer une routine de traitement et de classification d'images satellitaires au sein du module de préparation à la navigation. Cette classification permettrait de cibler en priorité les zones où la pollution a été détectée.

d) Intégration d'un facteur de probabilité de présence de nappe dans le choix des points de passage

En fonction des informations recueillies sur la zone d'étude (courants, vent, mer totale) et de modèles de simulation de dispersion de polluants, il est possible de définir des seuils de probabilité a priori en amont du déploiement des drones. Ces informations pourraient être intégrées au modèle de préparation à la navigation pour favoriser l'exploration des zones où la probabilité est la plus forte.

e) Intégration des zones à risque de manière à prépositionner des flottes de drone à proximité des zones à enjeux

Pour intégrer ces paramètres au modèle, il est possible de s'appuyer sur les recensements qui ont été effectués comme par exemple par l'ISSEMAR *http://www.isemar.asso.fr/fr/pdf/carte-isemar-56.pdf*

Il est également possible d'envisager que dans le cas où Protei intervient après le déploiement de moyens plus conventionnels, les pilotes aient besoin d'avoir à leur disposition des données de contexte comme les périmètres de sécurité établis sur la zone, le positionnement des navires destinés à pomper le polluant dans l'épave.

iv. Dynamisation

a) Appui sur un flux de données de météo marine dynamique - via une API

A l'heure actuelle, les données météorologiques sont modélisées à partir d'informations extraites depuis divers sites météorologiques selon la zone d'étude.(Météo France par exemple).

A terme, il serait intéressant de moissonner le contenu proposé par certains sites de données météorologiques marines. Ces sites proposent des API plus ou moins complètes, gratuites ou non. Parmi eux, il est peut être cité le site *wunderground.com* qui propose de récupérer via son API un flux de données XML : *http://www.wunderground.com/weather/api/d/docs.*
D'autres API existent. Le site suivant en liste quelques unes :
http://www.programmableweb.com/apis/directory/1?apicat=Weather

Il serait ainsi envisageable d'obtenir de manière dynamique et en temps réel les informations concernant les paramètres suivants :

- Vents
- Courants
- Pression atmosphérique (isobares)
- Hauteur et direction des trains de houle
- Visibilité (brume, brouillard)
- Mer totale : vagues liées au vent et houle
- Marées (coefficients, hauteur, horaires)

b) Intégration des données de trafic maritime via API

A l'image de l'ajout d'une donnée de routes maritimes fréquentées, il pourrait être envisageable de connaître la position de l'ensemble des navires sur la zone de déploiement ainsi que leur type, vitesse et route. C'est ce que propose le site marinetraffic.com : https://www.marinetraffic.com/fr/p/services

c) Adaptation dynamique du chemin en fonction des relevés du sondeur, des instruments météo embarqués et des remontées de capteur (module détection)

En plus des flux de données météo, le traitement du parcours optimal pourrait être optimisé par la prise en compte des informations renvoyées par le(s) sondeur(s) et les capteurs météo embarqués : anémomètre, baromètre…, mais aussi par les capteur environnementaux destinés à détecter la pollution ou tout autre objet d'étude.

VI.Module « Détection de pollution »

A. Contexte et présentation générale

Ce module géostatistique est destiné à offrir une vision synthétique et précise des données de pollution extraites sur le terrain. Il entre donc en service lorsque les drones sont sur la zone d'étude et qu'ils ont commencé à collecter des données de pollution.

Il définit la localisation de la source de polluants, déduite de l'analyse des données, ainsi qu'une idée de l'ampleur de la pollution au niveau de sa source. Il prend également en compte l'incertitude des données en entrée ainsi que leur exhaustivité afin d'en déduire un résultat nuancé. Celle-ci s'exprime par un ensemble de positions possibles de la source de pollution représentée en fonction de leur probabilité.

Dans le cas du déploiement de plusieurs drones, les informations recueillies peuvent être croisées. De cette manière, il est possible d'améliorer la connaissance de l'environnement de déploiement. Cela permet également de mutualiser ces informations et de les diffuser auprès de chaque pilote. Le module peut être scindé entre une première partie nommée "direction" qui évalue un certain nombre de directions probables de la source de pollution (voir *S'extraire du périmètre du drone*) à partir des données d'un drone et une seconde partie qui croise les données de plusieurs drones pour en déduire une zone de présence probable de la source de pollution ainsi que sa valeur estimée (voir *Modéliser un phénomène pour comprendre son importance*).

Ce module prend en compte l'emprise de la zone d'étude qui aura été définie au préalable. Des paramètres environnementaux connus a priori sont également pris en compte, notamment la bathymétrie de la zone dont les espaces émergés où la présence de pollutions ne sont pas pris en compte dans l'analyse.

B. Axes méthodologiques

L'axe méthodologique principal du module repose sur trois phases :

1. l'interpolation et l'extrapolation des données de pollutions recueillies par le drone
2. la traduction probabiliste des résultats
3. l'interprétation des résultats grâce à des outils dédiés

Les exemples de sorties du module présentés ici correspondent aux données de radiations extraites au Japon après la catastrophe de Fukushima, dont celles récoltées par César Harada. Ces données sont terrestres mais à ce niveau de l'analyse cela n'a pas d'importance. Nous avons également simulé l'emplacement de deux drones qui auraient pu récolter ces données

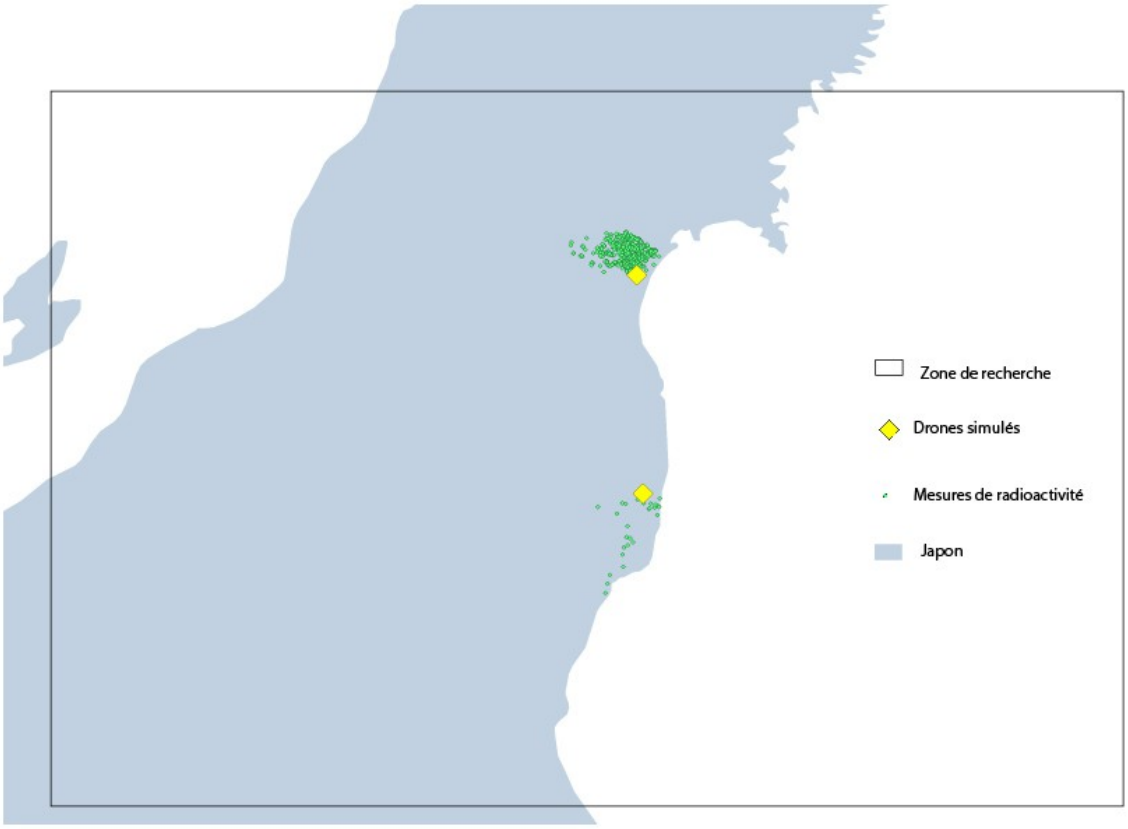


Illustration 28: Détection : exemples d'emplacements des mesures de radioactivité

i. S'extraire du périmètre du drone

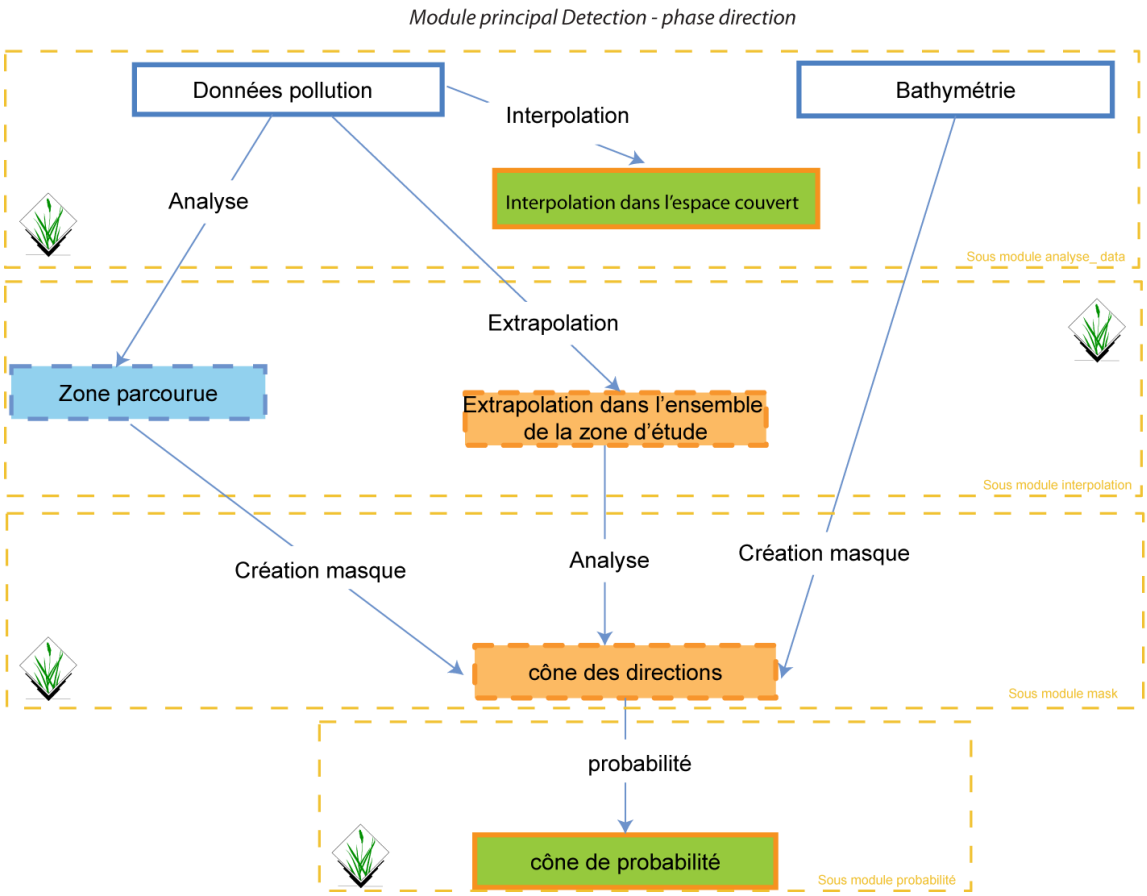


Illustration 29: Détection - chaîne de traitements du calcul de direction

La première étape consiste à passer de l'information ponctuelle des mesures prises par le drone à une information surfacique par interpolation. Pour ce faire dans GRASS on utilise la fonction d'interpolation par l'inverse de la distance. Cette méthode est limitée car elle consiste en une simple affectation de la valeur de chaque pixel à la donnée de pollution la plus proche avec une pondération par la distance. Cette méthode comporte des limites car elle ne prend pas en compte les tendances globales qui se dégagent de l'ensemble des données. Une interpolation par krigeage serait préférable pour prendre en compte les tendances globales mais est absente des modules de GRASS. De plus, cette méthode nécessite de nombreux paramétrages qui rendent difficile son utilisation dans un algorithme automatisé.

Fonction GRASS : La ligne de commande Shell utilisée est la suivante :

```
v.surf.idw input="...@zone" output="interpolation" layer=1 column="..."
npoints=... power=...
```

Paramétrage :

- column : Colonne contenant les valeurs de polluant
- npoints : Nombres de points pour l'interpolation
- power= : Pondération par rapport à la distance

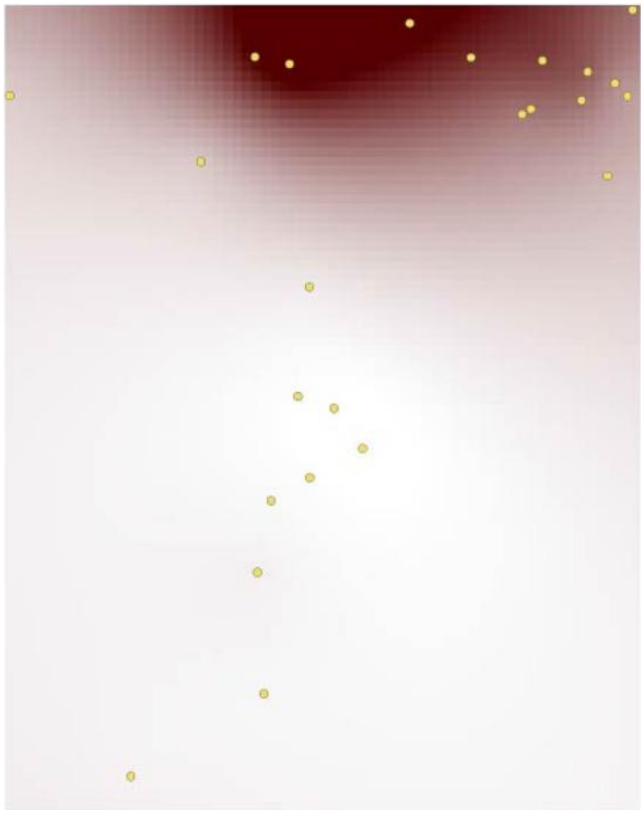


Illustration 30: Détection : résultat d'interpolation

Ici la valeur de radiation est globalement croissante vers le Nord mais le gradient n'est pas totalement uniforme (les points jaunes correspondent aux donnée source de César Harada)

Cette représentation permet de mieux se rendre compte de l'étendue spatiale d'un phénomène et d'en déduire des caractéristiques spatiales morphologiques. La question principale à se poser est : "où le phénomène polluant prend-t-il sa source ? En partant du postulat que plus on s'approche de la source plus la densité de polluant augmente, la caractéristique morphologique que nous voulons donc comprendre est dans quel sens et dans quelle direction le gradient de pollution augmente t'il ? Cette caractéristique peut être mise en lumière en étendant le résultat de l'interpolation locale à l'ensemble de la zone d'étude par extrapolation. Il est alors considéré que les zones avec de fortes valeurs extrapolées correspondent aux directions les plus probables vers le point source de polluant. L'extrapolation des données permet de s'abstraire des variations locales des données qui peuvent amener à de fausses interprétations. Néanmoins il faut qu'il y ait une cohérence qui se dégage à minima des données (un gradient de valeur dans une direction) et qu'il y ait un échantillon de donnée assez grand pour qu'une mesure incertaine ne brouille pas l'analyse.

Fonction GRASS : la ligne de commande Shell utilisée est la même que pour l'interpolation :

```
v.surf.idw input="...@zone" output="interpolation" layer=1 column="..."
npoints=... power=0
```

Paramétrage :

- power : le paramètre doit être ici de 0. Ainsi la valeur de polluant ne décroît pas avec la distance. En effet on ne cherche pas ici à connaître les valeurs de polluant de zone qui n'ont pas été sondées mais à connaître une direction.

Toutefois en amont de cette commande on définit l'emprise de l'interpolation à toute la zone d'étude

```
g.region vect="zone d'étude"@zone res=0.01
```

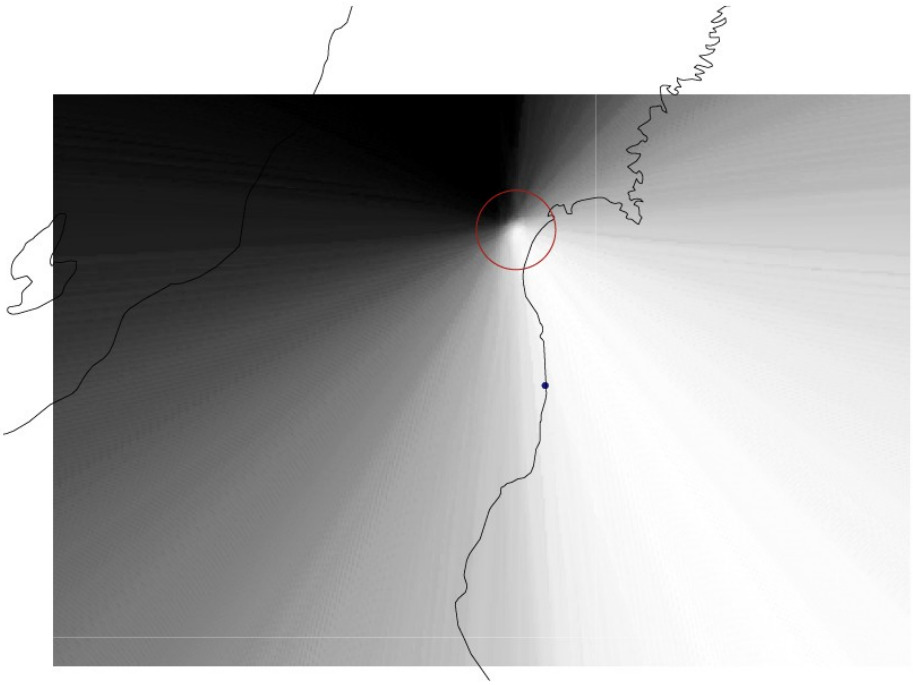


Illustration 31: Détection : résultats d'extrapolation

Ici la zone encerclée en rouge représente la zone qui a été sondée. Le reste de la zone en dégradé de gris est une extrapolation de l'interpolation de la zone sondée. En blanc les valeurs de polluant les plus fortes et en noir les plus faibles.

ii. Interpréter des données incertaines

Les résultats obtenus par extrapolation apportent un biais à l'analyse car ils sont fortement dépendants des données qui ont servi à leur réalisation. Ainsi les résultats sont difficilement interprétables en l'état tout en tenant compte de leur degrés d'incertitude. La question est donc de créer un indicateur de fiabilité et de l'intégrer dans la représentation.

L'indicateur de fiabilité doit prendre en compte l'étendue de la zone couverte par le drone par rapport à l'étendue totale. Plus un drone a couvert d'espace plus l'indice de confiance des résultats d'analyses est élevé. Il faut cependant prendre en compte aussi le nombre de points échantillonnés. Un drone ayant parcouru une surface importante mais n'ayant fait que peu de mesures aura des données peu fiables. Autrement dit, un drone ayant effectué un nombre important de mesures dans un espace restreint aura une bonne connaissance du local mais une connaissance globale faible. Inversement, un drone aura une bonne connaissance globale mais une connaissance locale faible s'il effectue un nombre faible de mesures sur une surface importante. Pour savoir si le nombre de points de mesure est acceptable, il faut définir une valeur de référence exprimée en nombre idéal de points de mesures par surface parcourue.

La fonction qui se dégage pour l'indicateur de fiabilité est :

```
fiabilite = 1/((Zp*(Npm/Npi))/Zt)
```

Avec :

- Npm : nombre de points échantillonnés
- Npi : nombre de points de mesure idéal (en point par km2)
- Zp : zone parcourue par le drone (en km2)
- Zt : zone totale de l'étude (en km2)

Le résultat varie entre zéro et l'infini. Plus le résultat est proche de zéro, plus la précision de la mesure est grande.

Il s'agit maintenant de représenter cette incertitude afin de croiser l'information venant de plusieurs drones ayant des niveaux de fiabilité variables. L'idée est que plus l'incertitude est élevée, plus le nombre de directions possibles vers la source est important car la probabilité d'avoir une direction exacte est faible. L'incertitude est représentée en gardant plus ou moins de directions probables. Ceci prend alors la forme d'un cône de directions probables. La taille de ce cône est calculée selon l'indice de fiabilité dont la valeur est traduite en pourcentage des plus fortes valeurs extrapolées affichées (lorsque l'indice est supérieur à 100 il est considéré que la fiabilité est trop faible pour pouvoir restreindre le nombre de directions probables). Donc plus la fiabilité de l'extrapolation est faible plus le cône sera large et inversement. Il faut ici faire remarquer que le pourcentage des plus fortes valeurs extrapolées représentées ne correspond pas mécaniquement au pourcentage de directions prises en compte (L'indice de fiabilité n'est pas directement traduisible en valeur d'angle). En effet, la variabilité des données extrapolées peut ne pas être linéaire que ce soit dans l'histogramme (loi normale imparfaite) ou dans l'espace. Tous ces traitements sont effectués dans le module "direction".

Un masque est créé sous GRASS en gardant le pourcentage x des valeurs extrapolées les plus fortes. x correspondant à l'indice de fiabilité.

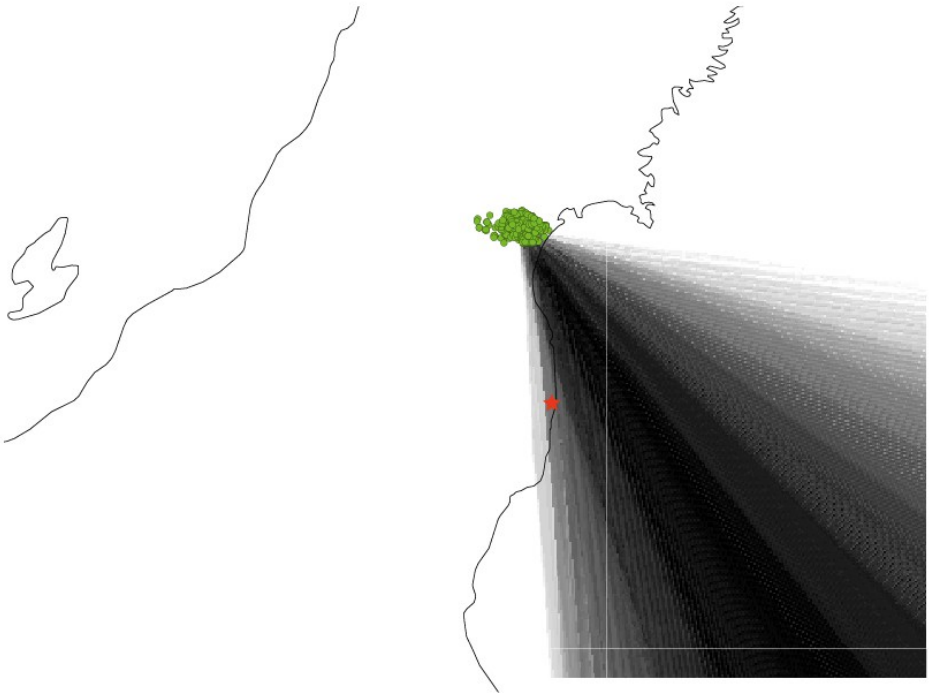


Illustration 32: Détection : création du cône de probabilité

Cette illustration montre le cône de probabilité de position de la source de pollution (ici la centrale de Fukushima) à partir de données de radioactivité mesurées dans une préfecture au Nord de celle-ci (points verts). Le nombre important de mesures et la zone couverte permettent d'avoir un indice de fiabilité de 25. Ainsi, 25% des valeurs les plus fortes extrapolées sont conservées. On obtient un cône d'environ 75 degrés. Plus le pixel est noir plus la probabilité que cette direction soit celle de la source est grande. On remarque que sur l'ensemble de ce cône les directions les plus probables sont vers le sud plus que vers le nord du cône. A titre de qualification du résultat, la centrale de Fukushima est représentée par une étoile rouge vers le centre de l'image.

On part du principe que la direction la plus probable exprimée par le traitement de l'échantillon correspond à la direction avec les plus fortes valeurs extrapolées. Pour représenter cette probabilité, on divise cette valeur maximale par l'ensemble des cellules du cône. On obtient ainsi des valeurs allant de 0 à 1, où 1 est la plus forte probabilité. Les zones où la présence de la source est impossible (exemple ici d'une centrale nucléaire qui ne peut se situer en pleine mer) sont retirées du cône.

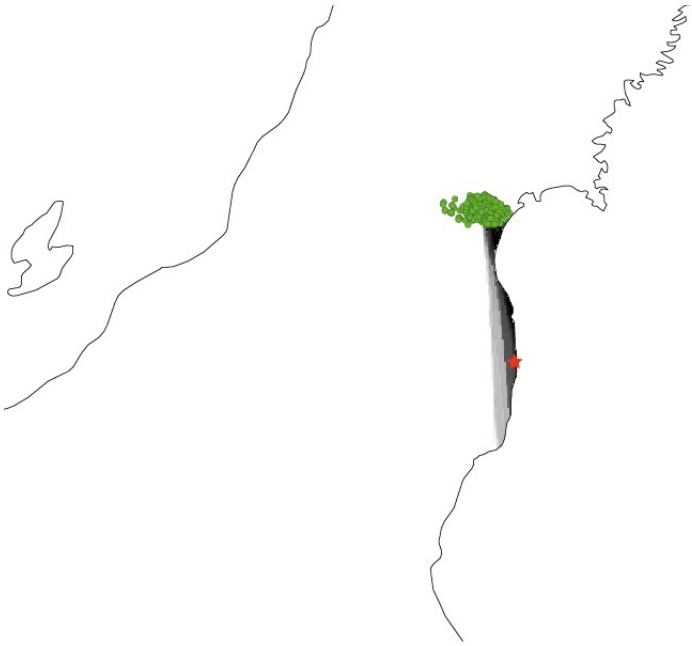


Illustration 33: Détection : application d'un masque pour les zones improbables

La zone de probabilité est obtenue par croisement des cônes de probabilité. Pour connaître la probabilité à l'intérieur de cette zone, les données issues de l'interpolation des données collectées par les différents drones sont ensuite extrapolées. Le résultat qui en découle est ensuite découpé par la zone de probabilité issue du croisement des cônes. Ce croisement de données est effectué dans le module "distance" (voir diagramme module "distance").

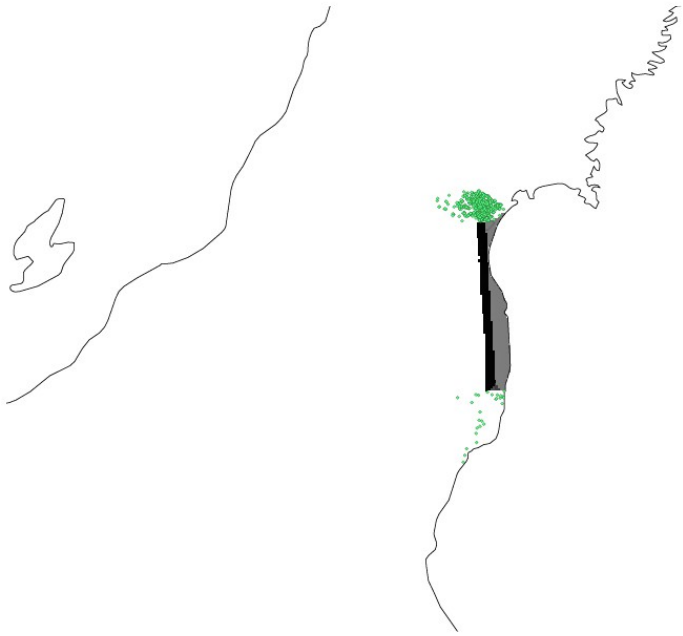


Illustration 34: Détection : zone de probabilité

iii. Se repérer par rapport à un nouveau référentiel

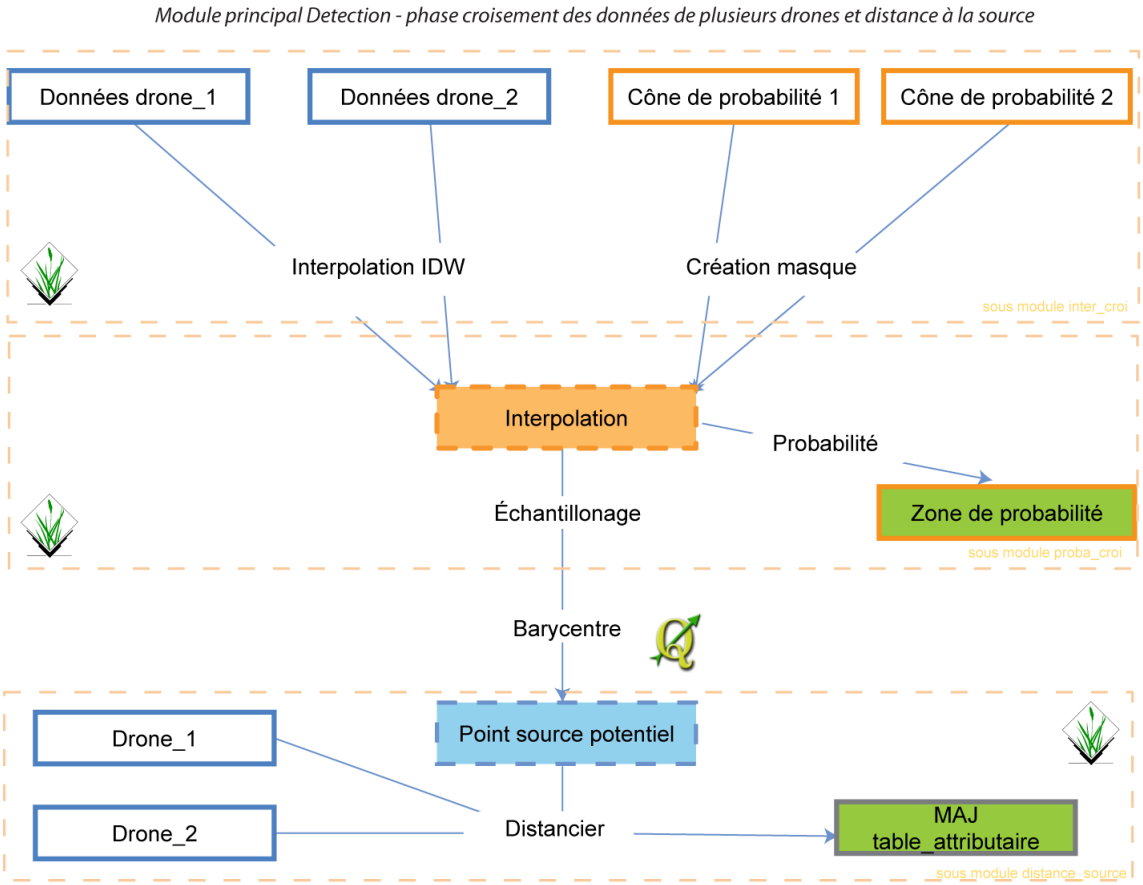


Illustration 35: Détection - Chaîne de traitements du calcul de distance

La zone de probabilité issue du croisement des données de plusieurs drones permet au pilotes d'avoir une idée de sa position par rapport au phénomène. Comment synthétiser cette information pour avoir une idée de la distance à la source ? Il faut alors créer un point source théorique à partir de la zone de probabilité, en créant un barycentre pondéré par la probabilité de chacune des cellules de la zone. Un distancier est alors créé pour chaque drone par rapport à la source théorique.

1. Sous GRASS, une couche de points échantillonnés est créée :

```
r.random input=...@zone n=... vector_output=echantillon
```

n : correspond au nombre de point de l'échantillonnage par défaut 200. La commande du barycentre est utilisée depuis QGIS car elle n'a pas été implémentée dans GRASS.

2. Sous GRASS, on calcule la distance entre le drone et la source théorique et on intègre le résultat dans la base attributaire du drone. La commande est :

```
v.distance from=drone_1@zone to=bary@zone to_type=point upload=dist column=DIST
```

Le point source théorique peut être affiché dans un visualiseur cartographique et les valeurs de distances mises en étiquette à chaque drone. Ce point source théorique donne juste une idée de la distance à la source car en donnant une valeur absolue, l'incertitude des résultats dans la zone de probabilité n'est pas prise en compte.

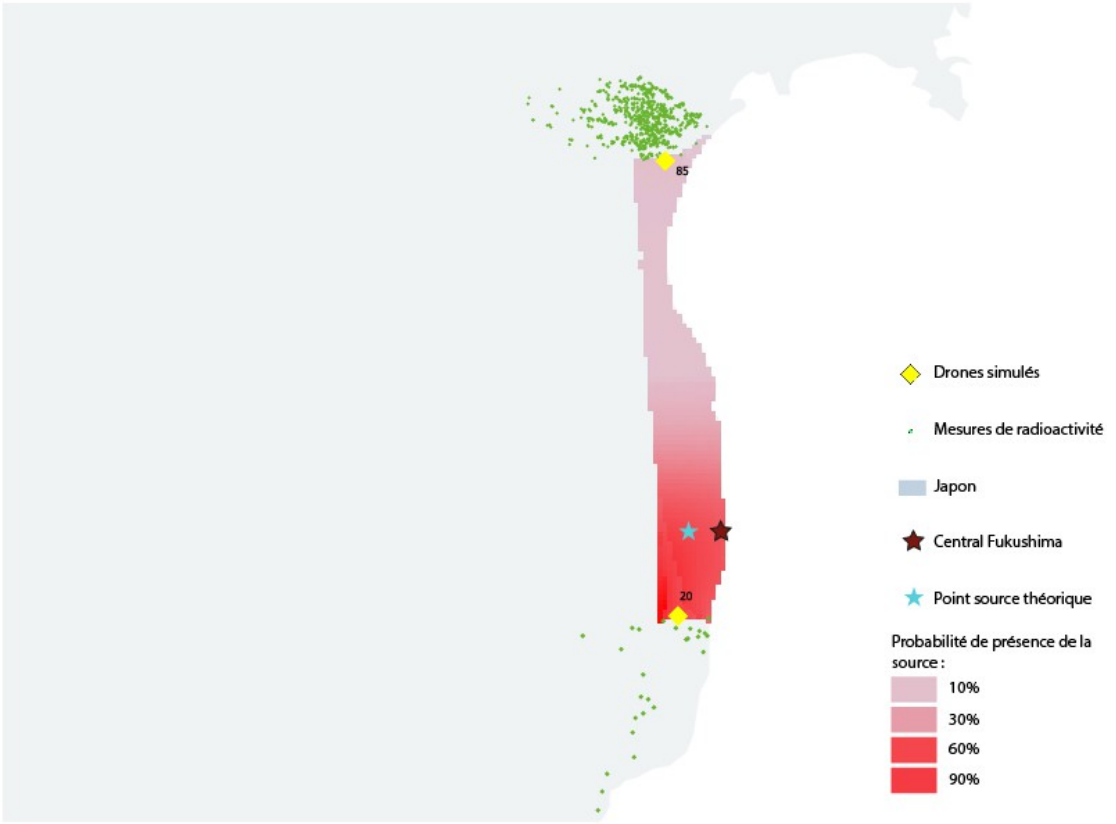


Illustration 36: Détection : résultat final

On voit ici que le point source théorique est situé un peu plus à l'ouest que Fukushima.

iv. Modéliser un phénomène pour comprendre son importance

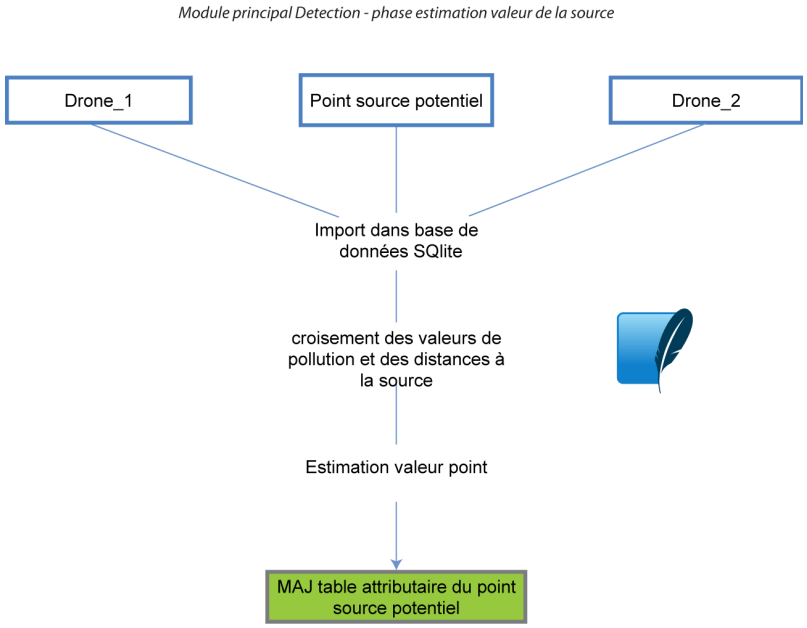


Illustration 37: Détetction - chaîne de traitements de l'estimation de la valeur source

En connaissant comment un phénomène se matérialise dans l'espace, il est possible de déterminer des prédictions sur son ampleur et sur son comportement dans le temps. Cette connaissance passe par l'établissement des lois physiques qui le caractérisent. Dans le cas d'une nappe de pollution, son comportement peut être prédit par connaissance suffisante des paramètres du système qui définissent ce comportement. Les drones Protei peut, en allant sur le terrain, valider, infirmer ou nuancer ces connaissances et s'en servir pour traiter au mieux la pollution. La première hypothèse posée, permettant de faire toutes les déductions qui ont suivi, est que la teneur en polluant diminue avec la distance à la source. Cette hypothèse semble juste dans notre cas d'étude lorsque la distance observée entre les points sources théoriques et réels est faible. La question est maintenant de connaître plus finement le paramètre qui définit la décroissance de la valeur de pollution avec la distance à la source. Pour ce faire, il est possible d'effectuer une régression entre la distance et la variable de pollution. Cependant, pour obtenir un résultat fiable, le nombre de points d'étude doit être important.

Dans l'outil que nous avons fourni, la valeur de pollution du point source est déduite par une loi linéaire entre la distance et les mesures de pollution. Un rapport linéaire entre distance et valeur de pollution peut être trop pauvre pour modéliser correctement le phénomène. Ce module n'est donc qu'une simple démonstration de ce qui pourrait être fait et n'est pas satisfaisant pour l'instant pour une étude fiable.

Sous GRASS, la colonne VAL_source du barycentre créée à l'étape précédente est mise à jour. La commande s'effectue dans une base SQLite qui contient les données attributaires et permet leur manipulation. En effet, la gestion de base de données au format DBF est impossible sous GRASS pour ce genre d'opération. GRASS permet néanmoins de lier les données avec une base SQLite pour effectuer ces opérations et ensuite mètre à jour les données au format DBF.

```
echo update bary set VAL_source=(-((DIST_drone_1*(VAL_drone_1-VAL_drone_2) /
(DIST_drone_1-DIST_drone_2)))+VAL_drone_1) | db.execute
```

Drone	Distance théorique	Valeur pollution		loi mathématique déduite par régression linéaire
1	85000	0,12		y = 5,84E-06x + 0,6169
2	20000	0,5		
Source	0	0,6169		

Illustration 38: Détection - résultat sous Excel de la valeur source

Dans notre cas d'étude il est peu probable que la décroissance de la valeur de pollution soit linéaire avec la distance. En effet la modélisation nous donne comme valeur de pollution à la source 0.61 Ms/h, ce qui parait bien faible. Le nombre d'échantillons est trop faible et nous n'avons utilisé que les données au niveau des drones simulés comme cela est fait dans l'outil.

En résumé le module de détection offre une méthode d'extrapolation des données issues des mesures faites par les drones et donne des pistes quant à l'exploitation prudente de ces résultats. Tout au long de sa présentation, de nombreuses limites et des propositions pour les dépasser ont été faites que le temps de l'étude ne nous permet pas de dépasser.

De nombreuses améliorations pourraient être ajoutées. Premièrement, il serait intéressant de pouvoir avoir un indice de fiabilité de la position du point source théorique. Il apparaît également important au vu du test sur les données de Fukushima que la valeur du point source théorique soit calculé à partir de l'ensemble des données échantillonnées et que son estimation se base sur des modèles mathématiques plus complexe.

Il serait également intéressant d'apporter une dimension temporelle pour être en mesure d'évaluer la dynamique de propagation du polluant.

Enfin, la principale amélioration serait la prise en compte de variables extérieures telles que le vent et les courants pour modéliser de manière théorique le comportement du polluant tout en enrichissant les modèles par les données terrain.

Dans le cadre du projet Protei en général, c'est tout un travail géostatistique qui pourrait permettre de fournir des aides à la décision rapides et synthétiques à partir de l'ensemble des données collectées par les drones. Ce travail permettrait de mieux comprendre le comportement des polluants en mer mais pourrait aussi s'appliquer à des études halieutiques ou géophysiques.

VII. Module « Déploiement d'une flotte »

A. Présentation générale

Lors d'une catastrophe environnementale, il est nécessaire de trouver des moyens pour intervenir le plus efficacement possible. Dans un cas de pollution, le temps de réaction et l'intervention des agents doivent être les plus efficaces possibles et ce afin de limiter au maximum les dégâts causés. L'enjeu est donc d'optimiser l'intervention en mobilisant plusieurs agents simultanément. L'objectif de ce module est donc de proposer des techniques de déploiement de plusieurs drones afin de répartir la surface à couvrir.

Cette méthode intervient en amont du déploiement des drones. En effet, le module "déploiement d'une flotte" tel qu'il a été pensé à ce stade du projet, permet de délimiter le périmètre d'intervention de chaque drone déployé, optimisant ainsi la couverture de zone et donc la recherche de la pollution. Il précède ainsi le module "Préparation à la navigation".

L'objectif est de découper l'emprise de recherche de manière équitable afin d'exploiter au maximum les potentialités de couverture des drones. L'efficacité de la couverture de zone croît donc en fonction du nombre d'agents mobilisés.

B. Méthode statistique "k-means" (MacQueen) :

La première méthode présentée s'appuie sur un découpage "statistique" de la zone reposant sur la méthode des "k-means" (algorithme de MacQueen).

Cette méthode nécessite de travailler à partir des centroïdes du maillage générés dans le *Module « Préparation à la navigation »* et formant ainsi un nuage de points qui est ensuite integré dans un repère orthonormé.

La classification par k-means fait partie des "algorithmes de partitionnement" dont l'objectif est de regrouper les individus dans des classes selon leur proximité. Dans le cas présent, les individus sont répartis de manière homogène (centroïdes d'une maille elle-même homogène). Ces classifications statistiques ne sont pas particulièrement destinées à un jeu de données "homogènes", toutefois l'attrait de cette méthode pour notre cas d'application réside dans sa nature "automatisée".

Le k-means consiste à définir autant d'individus de départ que de classes souhaitées en sortie. Ceux-ci sont d'abord positionnés de manière aléatoire. Il est ensuite nécessaire de fixer une "distance", c'est-à-dire la manière selon laquelle les points doivent être rassemblés (distance de Manhattan, euclidienne, etc). La distance euclidienne a été retenu car l'espace est considéré comme étant isotrope. Tous les individus les plus proches d'un point de départ aléatoire sont regroupés dans la même classe. Le barycentre de chaque classe est généré et la distance est ensuite recalculée entre chaque point et chaque barycentre. La particularité de l'algorithme MacQueen réside dans le fait que les réaffectations sont générés à chaque allocation d'individus. Ce processus est alors reproduit dix fois. Il a été démontré que les classes n'évoluent pas de manière significative au delà de ce seuil.

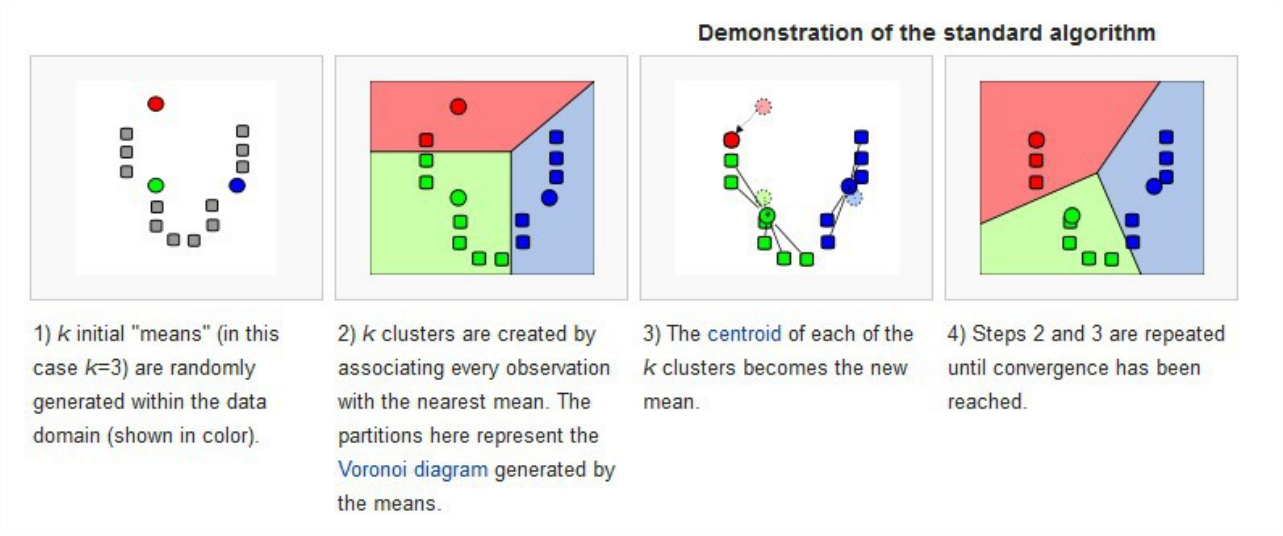


Illustration 39: Déploiement de flotte - méthode K-Means

Source : Wikipedia : http://en.wikipedia.org/wiki/K-means_clustering

Kmeans (Mc Queen): "les barycentres des classes ne sont pas recalculés à la fin des affectations , mais à la fin de chaque allocation d'un individu à une classe. L'algorithme est ainsi plus rapide, mais l'ordre d'apparition des individus dans le fichier n'est pas neutre." [1]

L'avantage de cet algorithme réside donc dans le fait qu'il soit plus rapide. C'est un atout dans ce cas d'application car le nombre d'individus peut être important (une grande emprise de recherche) et le fait que les points soient dispersés de manière "homogène" atténue l'impact sur leur ordre d'apparition dans le fichier. Il est en général contraignant de devoir déterminer le nombre de classes a priori, et une Analyse en Composante Principale précède alors souvent ces méthodes. Toutefois, dans ce cas d'application, c'est la "disponibilité" de drones qui va déterminer le nombre de classes.

L'idée avancée dans cette méthode est de "larguer" les drones sur la zone définie par le module "Déploiement d'une flotte".

1 Institut de Mathématiques de Marseille - CNRS UMR 7373 : *http://iml.univ-mrs.fr/~reboul/ADD4-MAB.pdf*

Module n° 4 « Déploiement d'une flotte » - Phase « Kmeans-McQueen »

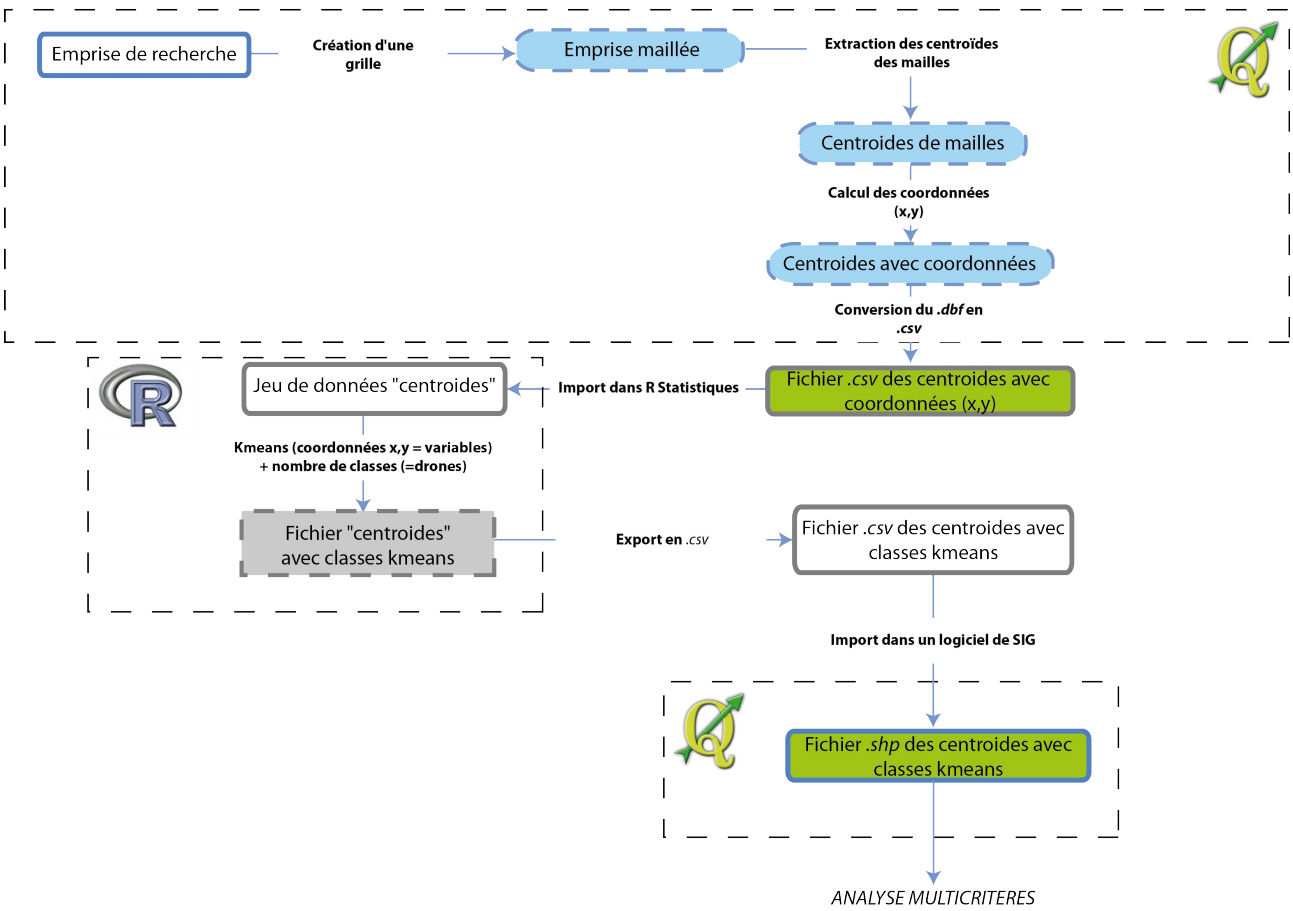


Illustration 40: Déploiement de flotte - chaîne de traitements globale

Entrée :
emprise de recherche maillée ou centroïdes, si disponibles (couche au format vectorielle)
Sortie :
centroïdes des mailles de la zone de recherche avec un champ de classification "Kmeans" (couche au format vectorielle)

i. Étape 1 : La zone de recherche

A ce stade, il y a plusieurs possibilités selon la donnée disponible. Dans le meilleur des cas, on part des centroïdes de mailles générés lors des phases précédentes (notamment le *Module « Préparation à la navigation »*). Dans le cas contraire, une grille est générée sur l'emprise de recherche déterminée, dont la taille de maille correspond au faisceau de recherche d'un drone. Les centroïdes sont ensuite générés. Le faisceau correspond au rayon de "balayage" par le capteur dans le cas d'une détection, ou l'emprise de captage de polluant pour un nettoyage.

ii. Étape 2 : Les coordonnées cartésiennes

Une fois les centroïdes disponibles, il faut extraire les coordonnées "X;Y" de chaque point et les placer dans deux champs distincts. Le fichier est alors exporté au format .csv.

iii. Étape 3 : Méthode Kmeans "MacQueen"

Le fichier généré lors de l'étape 2 doit être importé dans un logiciel de statistiques. Dans le cadre du projet, le logiciel libre R a été préféré (se référer au "manuel d'installation et d'utilisation" pour plus d'informations).

Le code R à exécuter pour l'import, l'exécution de la méthode statistique et l'export est présenté ci-dessous :

```
# Librairie à appeler : sert à intégrer la classification par kmeans dans le
# jeu de données (dans un champ du nom de la méthode en question)
library(Rcmdr)

# Import du jeu de données : centroïde_grid correspond à un fichier.csv
# les coordonnées (X;Y) doivent être en WGS84 :
Dataset <- read.table("D:/protei/data/centroïde_mailleCoord.csv",
   header=TRUE, sep=",", na.strings="NA", dec=",", strip.white=TRUE)

# Visualisation d'une partie du jeu de données pour vérifier l'importation
head(Dataset)

# Calcul de kmeans pour 3 points de départs (si mobilisation de 3 drones).
Effectif à modifier selon le besoin.
Kmean = kmeans(Dataset, centers=3, iter.max=10, algorithm="MacQueen")

# Visualisation de la donnée
Kmean

# Insérer le kmean dans un champ du jeu de données
Dataset$Kmean <- assignCluster(model.matrix(~-1 + X + Y, Dataset),
Dataset,Kmean$cluster)
# Export du jeu de données en .csv
write.table(Dataset,"D:/protei/data/centroïde_mailleKmeans.csv",",sep=";",col.na
mes=TRUE,row.names=FALSE, quote=TRUE, na="NA")
```

iv. Étape 4 : Import dans un logiciel de SIG

Il faut ensuite utiliser la fonction "charger un fichier de texte délimité" d'un logiciel de SIG (comme QGIS, se référer au manuel d'installation et d'utilisation pour plus d'informations).

En sortie du module, l'utilisateur peut alors repasser à la phase de préparation à la navigation afin de définir le parcours optimal pour chacune des sous-zones.

v. Perspectives

Les perspectives de ce bloc seraient de pousser la méthode statistique, notamment pour contraindre la classification à l'obtention de classes de même nombre d'individus (participant à l'optimisation de la couverture de zone). Il serait également intéressant de réussir à intégrer le script R dans le "modeler qgis" qui permettrait ainsi une automatisation quasi-complète de cette phase du module. De même, plusieurs manipulations manuelles sont nécessaires : modifications des entêtes de champs et sauvegarde de la couche de texte délimité. Elles pourraient être embarquées dans un script python intégré directement dans le modeler.

C. Méthode géographique

Outre la méthode statistique pour découper les zones, il est possible d'utiliser des méthodes relevant de la "géométrie algorithmique", qui sont déjà présentes en natif dans les logiciels de Systèmes d'Information Géographique. L'objectif reste de découper l'emprise de recherche en fonction du nombre d'agents déployés, dans un souci d'optimisation de couverture de zone.

Une de ces méthodes s'appelle "Polygone de Thiessen" (ou de Voronoi). Elle est basée sur une autre méthode appelée "Triangulation de Delaunay". Cette méthode mathématique dit que "un ensemble P de

points du plan est une triangulation DT(P) telle qu'aucun point de P n'est à l'intérieur du cercle circonscrit d'un des triangles de DT(P)." Ainsi, la condition de Delaunay affirme qu'un réseau de triangles est une triangulation de Delaunay si tous les cercles circonscrits des triangles du réseau sont vides[2].

Illustration 41: Déploiement de flotte - Triangulation de Delaunay avec cercles circonscrits (gris)

Source : Triangulation de Delaunay - Wikipedia : http://fr.wikipedia.org/wiki/Triangulation_de_Delaunay

Le "Polygone de Thiessen" découle de la "Triangulation de Delaunay" puisque "les sommets du diagramme de Voronoï sont les centres des cercles circonscrits des triangles de la triangulation de Delaunay. Les arêtes du diagramme de Voronoï sont sur les médiatrices des arêtes de la triangulation de Delaunay."

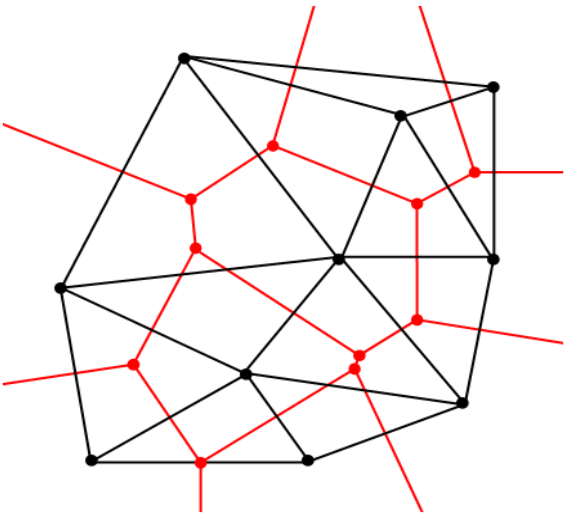


Illustration 42: Déploiement de flotte - Superposition de polygones de Thiessen (en rouge) et de sa triangulation de Delaunay (en noir)

Source : Polygone de Thiessen - Wikipedia : http://fr.wikipedia.org/wiki/Diagramme_de_Vorono%C3%AF

2 Source : Triangulation de Delaunay - *Wikipedia : http://fr.wikipedia.org/wiki/Triangulation_de_Delaunay*

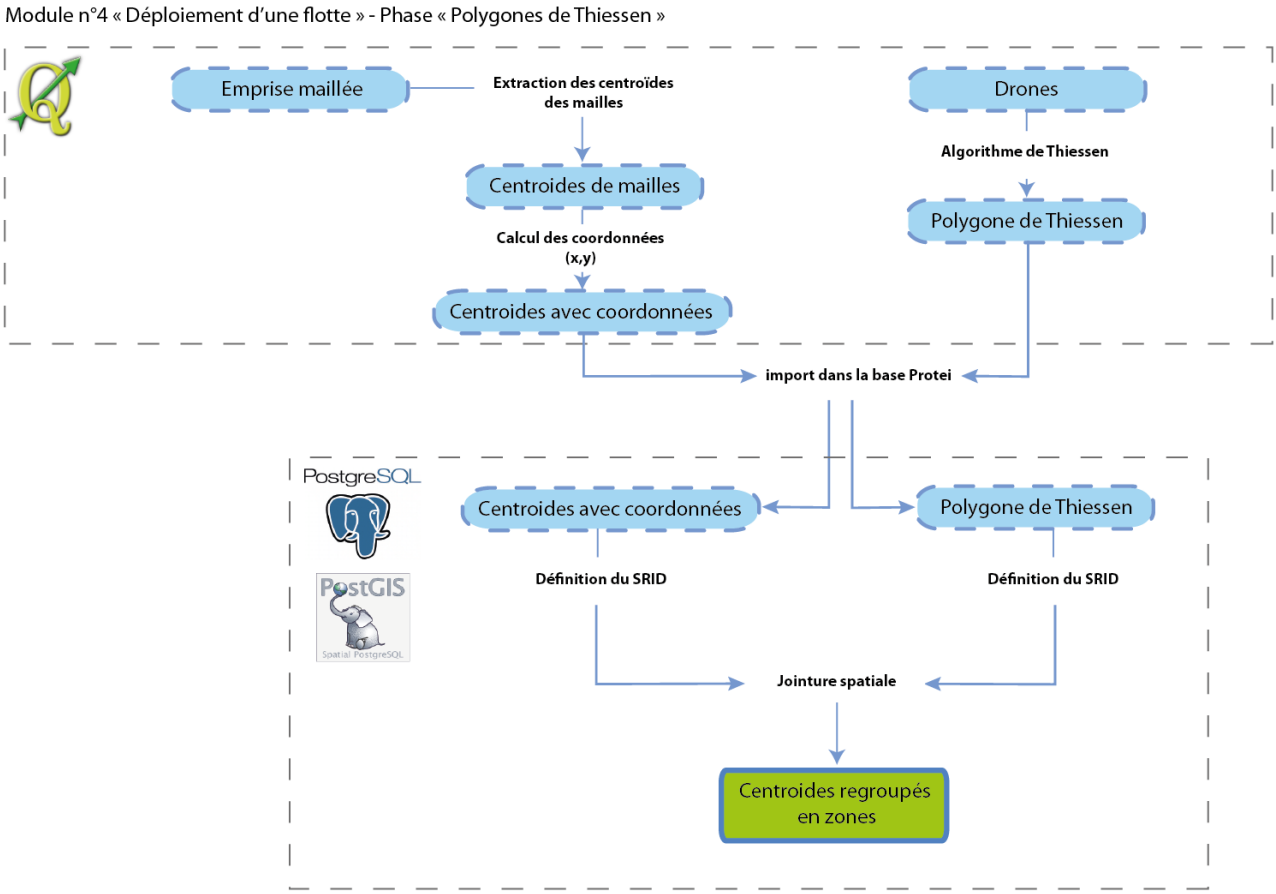


Illustration 43: Déploiement de flotte - Chaîne de traitements phase "Polygones de Thiessen"

Entrée :
emprise de recherche (couche vectorielle)
"agents" drones (couche vectorielle)
Sortie :
centroïdes de mailles classés (couche vectorielle)

i. Étape 1 : La zone de recherche

Comme pour la méthode par "k-means", il est possible d'effectuer cette chaîne de traitement à partir de l'emprise de recherche maillée, ou bien directement des centroïdes de mailles.

ii. Étape 2 : Les coordonnées cartésiennes

L'extraction des coordonnées "X;Y" de chaque centroïde n'est pas directement utile dans cette phase, toutefois il semble pertinent de les intégrer à la table en prévision des différents traitements qui suivront.

iii. Étape 3 : Algorithme de Thiessen

L'algorithme de Thiessen s'effectue à partir des drones positionnés de manière homogène autour de l'emprise de recherche. Cette emprise est alors divisée en autant de zones que de drones déployés et la grande majorité des centroïdes se trouvent dans l'une d'entre-elles.

iv. Étape 4 : Définition du système de coordonnées géographiques

Les centroïdes de mailles et les polygones de Thiessen sont alors importés dans une base PostgreSQL. Il est nécessaire de redéfinir le système de coordonnées géographiques (en WGS84, 4326) des deux

données via les lignes de requêtes SQL exposées ci-dessous afin d'éviter les messages d'erreur par la suite :

```
UPDATE thiessen SET the_geom = ST_SetSRID(the_geom, 4326);
UPDATE centroïdes_mailleCoord SET geom = ST_SetSRID(geom, 4326)
```

v. Étape 4 : Jointure spatiale

La dernière étape consiste à récupérer l'identifiant de chaque zone (=polygones de Thiessen) dans la table des centroïdes de mailles. La requête suivante permet d'effectuer la jointure, en créant une nouvelle table "centroïdes_classes" :

```
DROP TABLE IF EXISTS centroïdes_classes;
CREATE TABLE centroïdes_classes AS
    SELECT centroïdes_mailleCoord.gid, centroïdes_mailleCoord.id AS
    id_centroïde, centroïdes_mailleCoord.x, centroïdes_mailleCoord.y,
centroïdes_mailleCoord.xcoord, centroïdes_mailleCoord.ycoord,
centroïdes_mailleCoord.geom, A.ID
FROM thiessen A, centroïdes_mailleCoord
WHERE ST_WITHIN(centroïdes_mailleCoord.geom,A.the_geom)
```

La couche "centroïdes_classes" comprenant les centroïdes groupés en classes est ainsi créée dans la base "Protei".

vi. Perspectives

Cette méthodologie répond en partie au besoin de découpage de zone mais mériterait d'être étoffée. En effet, il serait possible d'imposer à l'utilisateur un positionnement homogène des agents sur le périmètre de l'emprise (via une contrainte d'intersection). Cela assurerait un découpage en parts égales de la zone, et donc permettrait d'optimiser au maximum les capacités de chaque drone.

D. Calcul d'efficacité

L'usage des drones peut se faire à des fins de « détection » ou de « nettoyage » de pollution. Dans ces deux cas, il est judicieux d'utiliser une simulation de couverture de zone afin de préparer au mieux le déploiement d'une flotte. En effet, l'utilisateur des drones ou bien la personne responsable de leur déploiement, se posera diverses questions en fonction du contexte auquel il fait face :

- Combien de drones doit-on mobiliser pour couvrir une zone de a km^2 en t heures ?
- En combien de temps les n drones pourront couvrir une zone de a km^2 ?
- Quelle zone serait couverte en déployant n drone pendant t heures ?

Quatre variables se distinguent alors selon le cas d'usage :

- n : le nombre de drones
- t : le temps (en heures)
- a : la surface (en km^2)
- v : la vitesse de déplacement du drone (en noeuds marins)

Dans le cadre du programme élaboré, une cinquième variable doit être renseignée. Il s'agit du champ de vision ou de la largeur de captation du drone, selon la nature de la mission : « captage » ou « détection ». Cette valeur pouvant varier selon le cas d'usage, elle n'a pas été prédéfinie pour que l'utilisateur la renseigne en fonction de l'objectif du déploiement.

i. Perspectives

Le programme a été écrit en langage Python parce qu'il est largement utilisé en application SIG. Toutefois, ce script n'étant pour l'instant relié à aucun SIG, un autre langage est peut-être plus pertinent. Ainsi, un informaticien confirmé voulant reprendre l'étude sera sûrement amené à optimiser le script, le développer et même le traduire dans un autre langage plus judicieux.

De plus, l'utilisateur doit rentrer l'ensemble des paramètres pour obtenir une simulation. Il serait possible à moyen-long terme de déterminer des paramètres fixes, éprouvés par des mises en situation des drones. Ainsi, un « contexte » pourrait être intégré au code, comprenant les vitesses moyennes correspondantes. Ainsi, l'utilisateur saurait qu'avec des conditions de vent et de courant particulières, le drone se déplace en moyenne à telle vitesse. De même pour les emprises de « captage » ou de « détection » qui pourraient être définies préalablement selon les outils équipés sur les drones.

Enfin, le script en lui-même n'est pas optimal :

- pas de gestion d'erreurs si l'utilisateur saisit un caractère au lieu d'une valeur,
- la boucle de répétition de la simulation n'est pas aboutie (fin de boucle prématurée et arrêt non-défini)

Ces éléments sont facilement corrigeables par un programmeur averti.

VIII. Conclusion et perspectives générales

L'information géographique n'avait pas encore été traitée avant notre participation au projet Protei. En effet, le développement du projet Protei étant encore à la phase de conception navale des drones, il est logique qu'aucun travail sur leur guidage ou la remontée d'information n'ait encore été effectué. Au début de notre atelier, aucun sujet n'avait été arrêté. Toutefois, les problématiques géographiques étaient évidemment présentes dans ce projet. Ainsi, un travail conséquent de réflexion avec le commanditaire et fondateur de Protei César Harada a été mené afin de définir notre périmètre d'étude, de manière a faire avancer le projet tout en exploitant les compétences délivrées dans notre formation en Géographie.

Finalement, ce rapport d'étude a démontré que la géomatique est omniprésente dans le projet Protei. En effet, elle a permis l'élaboration de stratégies de déploiement de drones : dans le guidage des agents et la couverture de zone ainsi que dans la détection et le nettoyage de polluants. Les axes qui ont été abordés doivent être approfondis mais nous pensons qu'ils peuvent constituer un premier socle méthodologique et technique suffisamment solide pour la poursuite de ces travaux. Les modules ont été développés uniquement en utilisant des outils et technologies open-source, en accord avec le paradigme du projet.

Au delà des aspects rappelés précédemment, et une fois que les drones Protei seront en capacité de récolter de l'information, celle-ci devra être remontée vers un module de contrôle pour être mise en base, stockée sur serveurs, traitée et diffusée. Les problématiques de déploiement étant éminemment géographiques, les données récoltées seront forcément géoréférencées et ainsi devront faire l'objet de manipulations où la compétence des géomaticiens semble nécessaire.

De plus, si le projet est amené à se développer, au point de voir l'usage des drones Protei se généraliser, une quantité importante d'information sera produite. Celle-ci pourra être centralisée afin d'être partagée à destination des acteurs du milieu marin comme au grand public, en accord avec la philosophie du projet : l'open-data.

Ainsi, il sera opportun de développer une Infrastructure de Données Géographiques (IDG) permettant la diffusion et la valorisation de celles-ci, à la fois par le biais de cartes, mais aussi par la mise à disposition des données en elles-mêmes. Cette plateforme permettra de fédérer une communauté d'utilisateurs pour enrichir la base de données. Pour cela, les données des drones devront être conformes aux standards de l'Open Geospatial Consortium (OGC) dans un souci d'harmonie des formats et assurer ainsi une interopérabilité.

Cette infrastructure sera le reflet de l'action de Protei à travers le monde, réunissant les contributions de l'ensemble des drones déployés dans toutes les mers et océans. Elle servira également de porte parole auprès du grand public, véritable vitrine des apports de ces drones sur la connaissance et la préservation du milieu marin. Au delà des aspects évoqués précédemment, diverses fonction pourront lui être attribuées :

- interface de simulation de déploiement
- visualisation/téléchargement des données produites
- espace d'échanges entre les contributeurs, les utilisateurs, le public
- journal d'information sur l'avancée du projet
- veille technologique et méthodologique liée aux capacités des drones

Cette notion de partage implique la valorisation de l'information géographique par un travail conséquent de représentation, notamment par le biais de la sémiologie graphique.

En ce sens, les géomaticiens auront toute leur place dans le développement du projet ainsi que dans son application opérationnelle en intégrant la communauté Protei.

Tables des figures