

---

## **Apport de la géomatique dans la modélisation d'un éco-urbain en 2D à l'aide de logiciels de SIG**

**Auteur :** Goffin, Céline

**Promoteur(s) :** Donnay, Jean-Paul; Dewals, Benjamin

**Faculté :** Faculté des Sciences

**Diplôme :** Master en sciences géographiques, orientation géomatique et géométrologie, à finalité spécialisée

**Année académique :** 2015-2016

**URI/URL :** <http://hdl.handle.net/2268.2/1594>

---

### *Avertissement à l'attention des usagers :*

*Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.*

*Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.*

---



Université de Liège  
Faculté des sciences  
Département de géographie

# Apport de la géomatique dans la modélisation d'un écoulement en région urbaine en 2D à l'aide de logiciels de SIG

---

*Mémoire présenté par*  
**Céline Goffin**  
*En vue de l'obtention du titre de*  
**Master en sciences géographiques**  
**Orientation géomatique et géométrie**

*Année académique 2015-2016*

*Membres du jury:*  
*Jean-Paul DONNAY (Promoteur)*  
*Benjamin DEWALS (Co-Promoteur)*  
*Pierre ARCHAMBEAU*  
*Pierre HALLOT*

## *Remerciements*

Je tiens tout d'abord à remercier mon promoteur, Jean-Paul Donnay, pour sa patience et sa disponibilité tout au long de ce mémoire. Ses précieux conseils ont sans aucun doute contribué à l'aboutissement de ce travail.

Je remercie également mon co-promoteur, Benjamin Dewals, le doctorant Martin Bruwier et Pierre Archambeau pour leurs suggestions avisées dans la réalisation de ce travail.

Merci également à Corentin, pour son soutien tout au long de mon parcours universitaire ainsi que pour la relecture de ce mémoire.

Enfin, je n'oublierai pas de remercier ma famille et mes amis pour leurs encouragements, et plus particulièrement Maman qui m'a permis d'aller au bout de ces études.

# Résumé

---

L'équipe du HECE (*Hydraulics in environmental and civil engineering*) de l'Université de Liège conçoit un modèle hydraulique d'inondation en région urbaine. Une approche à l'aide de techniques de géomatique a été sollicitée dans sa conception. Le présent mémoire propose des méthodologies construites à partir de logiciels de SIG (Système d'Information Géographique) en vue de les intégrer au modèle des ingénieurs. En outre, ce travail décrit une méthode basée sur les concepts de l'*antialiasing* pour représenter les éléments de voirie et de bâti de la région urbaine, sans généraliser l'information d'affectation à l'échelle du pixel. Il fournit également une méthode pour simuler un écoulement dans le modèle urbain, grâce à l'utilisation des algorithmes de « distance-coût ». Celle-ci s'avère toutefois trop générale pour représenter efficacement un écoulement. Ensuite, deux méthodes ont été élaborées pour travailler les valeurs de propagation à différentes résolutions. La première, basée sur l'agrégation des pixels en conservant la moyenne des valeurs issues de l'algorithme de « distance-coût », ne répond pas aux attentes des hydrauliciens. La seconde méthode, plus adéquate, calcule des coefficients de propagation en sortie de mailles grossières en considérant la moyenne des pixels en bordure de maille. Enfin, une technique pour extraire les axes centraux des voiries a été conçue. Il s'agit cependant d'une approximation, et les intersections des rues ne sont pas toujours placées correctement au centre des carrefours.

# Abstract

---

HECE's (*Hydraulics in environmental and civil engineering*) team from the University of Liège is designing an hydraulic model of flooding in urban area. An approach using geomatic methods has been requested for its design. This master thesis presents some GIS (Geographic Information System) methods that will be integrated to the engineer's model. Furthermore, this work describes a method based on antialiasing to represent the roads elements and built zones in the urban area, without generalising the assignment to the pixel scale. This work also produces a method to simulate a runoff in the urban model thanks to « cost distance » algorithms. But this last method is too general to represent the runoff with efficiency. Then, two methods are created to manipulate propagation values at different resolutions. The first is based on pixels aggregation which assigns the pixels mean value coming from the « cost distance » algorithm to the new one, but it does not match with the engineers desires. The second one, more suitable, calculates propagation coefficients in coarse pixel output by considering the pixels' mean in edges. Finally, a method to extract the roads central axes has been designed. Nevertheless, this last method is an approximation, the roads intersections are not always placed in the crossroads center with enough precision.

# Contenu

Résumé.....	3
Abstract.....	4
Liste des acronymes.....	7
1. Introduction .....	8
2. Etat de l'art .....	10
2.1. Etudes des inondations.....	10
2.1.1. Méthodes courantes.....	10
2.1.2. Méthodes en géomatique .....	13
3. Objectifs .....	15
4. Hypothèses .....	17
5. Données.....	18
5.1. Les données de base .....	18
5.2. Les données résultantes.....	20
5.3. Les logiciels de SIG .....	21
6. Traitements .....	22
6.1. Représentation des obstacles.....	22
6.1.1. Conservation de l'information .....	22
6.1.2. Méthode de calcul de la porosité.....	26
6.1.3. Discussion .....	35
6.2. Simulation de propagation d'un écoulement.....	37
6.2.1. Théorie et tests.....	37
6.2.2. Application au modèle.....	42
6.2.3. Application au modèle avec des valeurs de porosité .....	50
6.2.4. Discussion .....	61
6.3. Adaptation à différentes résolutions.....	65
6.3.1. Par agrégation des pixels .....	65
6.3.2. Par moyenne des valeurs en sortie .....	73
6.3.3. Discussion .....	81
6.4. Extraction des axes de voirie.....	82

6.4.1. Méthodologie .....	82
6.4.2. Discussion .....	99
7. Conclusion et perspectives.....	100
Références .....	103
Annexes.....	107

# Liste des acronymes

---

DEM	Digital Elevation Model – Modèle numérique de terrain
GPS	Global Positioning System – Système de positionnement mondial
SIG	Système d’Information Géographique
LIDAR	Light/Laser Detection And Ranging
HECE	Hydraulics in Environmental and Civil Engineering
CSV	Comma-Separated Values
ID	Identifiant
AVL	Attribute Values File
VB	Visual Basics



# 1. Introduction

---

Les écoulements en milieu urbain sont des phénomènes complexes qui présentent de grands enjeux pour la société actuelle. De tout temps, de nombreuses villes ont été construites à proximité des fleuves pour diverses raisons (consommation, transport, etc.). Par après, l'urbanisation a entraîné la prolifération des villes dans des zones plus vulnérables ainsi que l'augmentation de la part de population urbaine. Ainsi, certaines villes ont même été construites sous le niveau de la mer et entourées de digues pour se protéger des aléas des marées. Malgré les précautions prises au cours du temps, les inondations sont toujours aussi présentes dans les régions maritimes ou fluviales. Que ce soit sur le plan économique, social ou autre, les écoulements engendrent des impacts importants dans les zones densément peuplées. Il est dès lors nécessaire de les étudier, afin de prévoir et limiter les dégâts qu'ils engendrent.

Les données relatives aux inondations peuvent être obtenues de différentes façons. De nos jours, outre l'obtention de données en temps réel sur le terrain comme les mesures directes du niveau d'eau à l'aide de marqueurs, de nombreux outils permettent d'obtenir des données relatives aux inondations et à leurs risques avant même qu'elles ne se produisent. La géomorphologie, l'étude de la sédimentation, la modélisation hydraulique, l'emploi de modèles digitaux d'élévation (DEM) dans les systèmes d'information géographique (SIG) et la recherche des crues historiques sont des exemples très répandus de méthodes d'analyse des inondations. L'intérêt de la plupart de ces techniques est de prévenir du risque de d'inondation avant même d'aménager le territoire. Ainsi, des décisions et des stratégies adéquates peuvent être prises afin de limiter les constructions dans les zones où les risques sont trop élevés. Par ailleurs, il est parfois difficile et dangereux de récolter des données sur le terrain, c'est pourquoi les simulations à l'aide de modèles sont souvent utilisées.

Le présent travail a été initialement réalisé sur la demande des ingénieurs en hydraulique faisant partie de l'équipe du HECE (*Hydraulics in Environmental and Civil Engineering*) de l'Université de Liège. Ils recherchent en effet à développer un modèle de simulation d'inondation en région urbaine. Pour ce faire, ils ont dans un premier temps considéré l'écoulement expérimenté en laboratoire à partir d'un modèle physique réduit par Araud à Strasbourg, en 2012. Dans un second temps, il fut question de numériser ce modèle afin de l'utiliser dans la simulation numérique d'inondation. La finalité du projet est l'écriture d'un script permettant de calculer et de représenter le plus fidèlement l'écoulement. A cette fin, une validation du modèle numérique à l'aide de la géomatique a été sollicitée. La demande du doctorant M. Bruwier en charge du projet fut donc, grâce à la géomatique, d'assurer un schéma spatial numérique cohérent employé comme base, et de fournir une méthode de calcul permettant d'obtenir des valeurs multiscalaires susceptibles de les aider dans la détermination de leur écoulement.

Nous nous sommes dès lors penchés sur la question de la représentation des obstacles dans le modèle numérique afin d'en tenir compte dans l'écoulement des inondations. Nous nous sommes également demandé si la géomatique permettait de simuler des écoulements dans le modèle, et dans quelles mesures l'exploitation des données est possible à différentes résolutions. Enfin, ce mémoire propose une méthodologie pour extraire les axes centraux des voiries de modèles urbains.

Ce mémoire est organisé comme suit. Un bref aperçu de l'état actuel des études sur les inondations et des apports de la géomatique est repris dans la section 2. La partie 3 fait part des objectifs du travail et des notions à prendre en compte pour sa compréhension. La section 4 formule l'hypothèse autour de laquelle s'est basé ce travail. La partie 5 décrit les données fournies. Des traitements sont ensuite appliqués à ces dernières et sont exposés dans la section 6. Finalement, les résultats de ce travail seront discutés et conclus dans la section 7. Quelques idées de travaux futurs y seront également énumérées.

## 2. Etat de l'art

---

### 2.1. Etudes des inondations

Durant les dernières années, le nombre d'études se rapportant aux inondations n'a cessé de croître et ce notamment à cause des changements climatiques dénoncés par le GIEC (Groupe d'Experts Intergouvernemental sur l'Evolution du Climat)(2013) qui pourraient augmenter la fréquence et l'importance des crues (Smith *et al.*, 2000) et à l'urbanisation. Beaucoup de ces études tendent à se rapporter aux régions urbaines car ces villes se sont développées dans des régions à risque, à proximité des côtes et des fleuves (Gralepois & Guevara, 2016). Ces travaux prennent toute leur importance dans le cadre de l'aménagement du territoire afin d'étudier les risques éventuels liés aux aléas d'inondation.

Il est bien évidemment possible de récolter des données sur les inondations en temps réel à l'aide de repères de crues, d'images satellitaires,... Mais de nombreux articles traitant d'inondation ont pour objectif d'analyser les risques qu'elles se produisent à l'avenir dans une région donnée, et les conséquences que cela entrainerait. Il en ressort que des stratégies urbaines spécifiques doivent être mises en place, permettant de s'adapter aux aléas d'inondation (Gralepois & Guevara, 2016). Différentes techniques d'analyse sont utilisées afin de prévoir les risques d'inondation et d'apporter des résultats précieux destinés à l'aménagement du territoire.

#### 2.1.1. Méthodes courantes

Des simulations hydrauliques et hydrologiques ont été développées afin d'étudier les inondations. Elles se basent sur des modèles, qui sont des abstractions de la réalité (Izquierdo *et al.*, 2004). Ces modèles consistent à fournir des données et des valeurs en input, et de leur faire subir des traitements –généralement mathématiques- pour les transformer en outputs (Izquierdo *et al.*, 2004). Divers modèles hydrauliques ont été développés en se basant sur des principes physiques tels que les conservations du moment, de la masse, du volume,... (Cea & Rodriguez, 2016; Dottori & Todini, 2013) ou sur des équations hydrauliques. Certains modèles emploient des algorithmes plus complexes tels que la discrétisation par Quadrees (Liang *et al.*, 2004; Liang *et al.*, 2008) ou la triangulation de Delaunay et les diagrammes de Voronoï (Karimipour *et al.*, 2013). De manière générale, si les modèles hydrauliques représentent de simulations d'écoulements canalisés, les modèles hydrologiques sont, eux, plus proches de l'étude des problèmes liés aux pluies et aux conditions climatiques, et des impacts qui en résultent (Guinot, 2003). Ils se basent généralement sur des équations hydrologiques et

font appel à des données de précipitation, de capacité de rétention du sol,... (Cabral *et al.*, 2014).

Ces simulations sont très répandues, mais elles font appel à des connaissances spécifiques à leur domaine respectif. Dans ce travail, le domaine d'étude qui nous intéresse est la géographie. En plus des notions d'aménagement du territoire qu'elle fournit, d'autres aspects de cette discipline permettent de mettre en lumière la problématique des inondations.

La géomorphologie, l'hydromorphologie et d'hydrogéomorphologie sont ainsi fréquemment utilisées. L'hydromorphologie se définit comme « l'étude de la morphologie des cours d'eau, fleuves et rivières, et notamment l'évolution des profils en long et en travers ainsi que du tracé planimétrique » (Encyclo-ecolo, 2011). Dans certaines définitions, le concept d'hydrodynamisme est couplé à celui de la morphologie (Ifremer, 2010). Quant à la géomorphologie, il s'agit de « la science qui décrit et explique les formes du relief terrestre » (SSGm, 2009) et s'avère donc plus générale. L'hydrogéomorphologie, elle, recoupe les deux concepts et est en plus conditionnée par la problématique du risque d'inondation. Elle se caractérise comme « une méthode basée sur une démarche naturaliste qui met en évidence les différents lits des cours d'eau, les divers aménagements susceptibles de perturber les écoulements, en les accélérant ou en les ralentissant, et à en déduire les zones inondables ainsi que les valeurs approchées des paramètres physiques des inondations (hauteur d'eau et vitesse d'écoulement) » (Ballais *et al.*, 2011). La notion d'hydrogéomorphologie fut seulement adoptée à partir des années 90 par Masson, Garry et Ballais (Ballais *et al.*, 2011). Les termes de géomorphologie et d'hydromorphologie reviennent régulièrement dans les travaux sur les crues, mais la méthode l'hydrogéomorphologie a été conçue exclusivement à cet effet. Quoi qu'il en soit, ces méthodes d'analyse permettent de déterminer l'espace concerné par une inondation. Par exemple, en étudiant la granulométrie du lit ou en utilisant des photos aériennes, il est possible de retrouver des tracés et des formes géomorphologiques dues aux crues (Ballais *et al.*, 2005). Si les crues influencent la géomorphologie de la région, il est vrai aussi que la géomorphologie des cours d'eau joue un rôle essentiel dans les crues. Par exemple, un lit qui s'est incisé augmentera sa capacité de transport et engendrera des débordements moindres (Guan *et al.*, 2016). Par ailleurs, lors d'une inondation extrême, la dynamique de sédimentation et la géométrie du cours peuvent être modifiés. De tels changements influencent les crues, mais les résultats varient énormément d'un cas à l'autre et selon l'échelle considérée (Guan *et al.*, 2016). En tout état de cause, l'étude des inondations à l'aide de la géomorphologie, de l'hydromorphologie ou de l'hydrogéomorphologie est prônée par de nombreux auteurs. Certains utilisent cette dernière en vue de l'obtention d'une cartographie hydrogéomorphologique qui consiste à représenter les délimitations des zones inondables (Ballais *et al.*, 2011; Demers *et al.*, 2014; Montané *et al.*, 2015). En France, l'emploi de ce type de cartographie dans des plans de prévention de risques d'inondation est en hausse, surtout dans les régions ayant subi des crues extrêmes (Montané *et al.*, 2015).

Une autre méthode pour déterminer les risques d'inondation est, par exemple, de récupérer et d'analyser des données sur les crues extrêmes précédentes issues du même cours d'eau. Il convient dès lors de se renseigner sur ces crues dites historiques. Les témoignages écrits, oraux, photographiques et autres des habitants des régions concernées peuvent être employés à cet effet. Un autre moyen est l'étude et la datation des dépôts dans le lit majeur (Ballais *et al.*, 2004). L'intérêt est notamment de déterminer les périodes de retour des crues. Cependant, l'étude des risques basée sur les crues historiques présentent des lacunes (Ballais *et al.*, 2004). En effet, de nombreux facteurs influençant les crues varient au cours du temps. A titre d'exemple, la topographie de la région a pu évoluer de façon significative en quelques décennies. En l'absence de données topographiques anciennes, il est difficile d'apporter des corrections pertinentes aux résultats. Les calculs effectués se basent souvent sur l'hypothèse que les conditions de crue sont restées constantes, et c'est la topographie actuelle qui est utilisée (Ballais *et al.*, 2004).

Les modèles digitaux d'élévation (DEM) sont des représentations digitales ou numériques de la surface d'un terrain tenant compte de son élévation (O'Callaghan & Mark, 1984). Ils sont notamment utilisés dans les simulations de hausse du niveau de la mer, des éruptions volcaniques, et autres phénomènes continus durant lesquels un flux quelconque se propage à la surface de la terre. Ils peuvent servir dans la représentation de réseaux de drainage et souvent employés dans des modèles hydrauliques ou hydrologiques pour la modélisation d'inondations. Il est possible de les représenter en 3D, mais leurs données sont parfois simplement structurées selon une grille de carrés réguliers (Kim *et al.*, 2011; O'Callaghan & Mark, 1984). Ils peuvent s'obtenir de différentes façons telles que la photogrammétrie, le GPS, le scanner laser, la cartographie, le LiDAR,... et à différentes résolutions (Coveney & Fotheringham, 2011; J.P. Leitão *et al.*, 2008). Il est aussi possible de créer un DEM en interpolant simplement les courbes de niveau (O'Callaghan & Mark, 1984). La résolution des instruments influence considérablement la précision des données et par conséquent l'efficacité de l'utilisation des DEM (Leitão *et al.*, 2008). Par ailleurs, les DEM présentent parfois des erreurs d'élévation susceptibles d'engendrer des résultats faussés. Ces erreurs peuvent par exemple être dues à une résolution trop basse des appareils de mesure utilisés, à la végétation présente lors des mesures ou à une trop grande variabilité du terrain (Coveney & Fotheringham, 2011). Pour limiter et diminuer ces erreurs, Coveney & Fotheringham (2011) proposent de fusionner plusieurs techniques de mesure, et plus particulièrement le GPS et le scanner laser terrestre, pour un même DEM. Les DEM sont régulièrement employés avec des SIG ou d'autres méthodes pour simuler des inondations. Les réseaux d'écoulement simulés à partir de DEM sont basés sur la gravité. En général, les flux suivent les pentes, allant du niveau d'élévation le plus élevé au plus faible en négligeant les phénomènes d'interception, d'évapotranspiration et d'infiltration (Ozdemir & Bird, 2008). Cependant, les traitements deviennent vite lourds lorsqu'ils sont appliqués à haute résolution et sur de larges surfaces (Shen *et al.*, 2015).

### 2.1.2. Méthodes en géomatique

Plus précisément, nous nous sommes posé la question de l'apport de la géomatique dans les modélisations d'inondation. Elle fait notamment appel à des notions de cartographie numérique, d'analyse spatiale et d'infographie. Le domaine de la géomatique s'est fortement développé dans les dernières décennies (Zazo *et al.*, 2015). Ce développement récent en fait une méthode d'étude des inondations encore incomplète.

La géomatique propose, entre autres, différentes techniques d'acquisition des données géospatiales (Zazo *et al.*, 2015). Les nouvelles technologies permettent d'obtenir des données toujours plus précises. Après avoir été collectées, ces données peuvent être stockées, traitées et diffusées. La géomatique permet d'utiliser des données issues de différentes sources et représentées dans un format digital (Gomarasca, 2004).

Les traitements proposés par la géomatique font appel à la cartographie numérique. La cartographie décrit la forme et les dimensions de la Terre par la représentation numérique de régions selon des règles. Elle permet de détailler des phénomènes réels ou abstraits et de les localiser à la surface de la Terre (Gomarasca, 2004). Les jeux de données géospatiales sont généralement représentés selon une surface de référence et un système de référence grâce à la cartographie. L'avènement de la cartographie numérique a permis le traitement des données par des techniques de géomatique.

En ce qui concerne le traitement des données, des Systèmes d'Information Géographique (SIG) ont été conçus. Selon Swanson (2001), « les SIG utilisent la technologie de l'information et des données pour saisir, structurer, manipuler, intégrer, analyser et afficher l'information contenant un aspect géospatial ». Ces SIG permettent entre autres de géoréférencer et mettre en relation des données environnementales issues de différentes sources et de différentes résolutions (Stuart & Stocks, 1993).

Outre la position géographique des données en étroite rapport avec la cartographie numérique, les SIG peuvent gérer les attributs (c'est-à-dire les propriétés) de ces données, les relations entre elles, et même le facteur temps de phénomènes dynamiques. Les relations entre entités peuvent être de différents types : topologiques, directionnel, qualitatif ou quantitatif (Gomarasca, 2004). Plusieurs de ces types seront considérés dans ce mémoire. Les SIG savent traiter les données sous forme raster ou sous forme vectorielle.

Stuart et Stocks (1993) avaient déjà relevé dans les années 90 les apports de l'utilisation des outils de SIG dans la modélisation hydrologique. Chang et Huang (2015) proposent d'utiliser des SIG pour calculer la distribution spatiale de données d'exposition, de sensibilité et de capacité adaptative d'une région urbaine. Ils modélisent ainsi l'hétérogénéité de la vulnérabilité de la région aux inondations. Cabral *et al.* (2014) utilisent les SIG pour superposer des niveaux d'eau, calculés à partir de modélisation

hydrologique et hydraulique, à des données topographiques. Le résultat escompté est la modélisation de la hauteur des inondations. Les SIG peuvent également être utilisés pour déterminer la valeur de facteurs susceptibles d'influencer les inondations (Diakakis *et al.*, 2016). En soi, l'utilisation des modèles numériques de terrain ou d'élévation 2D et 3D nécessite également l'emploi de techniques de SIG. Le champ d'application des outils de SIG dans le domaine des inondations est large.

Des logiciels spécifiques ont été créés, utilisant ces systèmes. Ils portent également la dénomination de SIG. Ils proposent des outils pour réaliser des analyses complexes sur de grandes quantités de données géospatiales (Gomasasca, 2004). Un premier logiciel de SIG apparaît sur le marché dans les années 60 (Gomasasca, 2004). Depuis lors, différents organismes en proposent. Ils peuvent être en accès libre ou nécessiter des licences d'utilisation. Certains de ces logiciels incluent des outils spécifiques pour la gestion des données hydrologiques (Crayfish, VectorFieldCalc, Floodrisk...).

Parmi les différentes techniques présentées pour étudier des inondations, nous nous basons uniquement sur l'approche de la géomatique. La géomatique est un domaine récent et en constante progression, parallèlement aux nouvelles technologies qui fournissent des méthodes d'acquisition et de traitement de données géospatiales toujours plus efficaces. Plus particulièrement, nous allons essayer de traiter des données relatives à un modèle d'inondation en région urbaine à l'aide de logiciels de SIG. Toutefois, les fonctions spécifiques aux traitements de données hydrologiques ne seront pas utilisées car elles demandent généralement des paramètres déjà complexes en entrée tels que les débits, les vitesses, etc.

### 3. Objectifs

---

L'intérêt principal de ce mémoire est de présenter des techniques de géomatique susceptibles d'avoir un réel apport pour la modélisation du franchissement d'une inondation dans une surface bidimensionnelle composée d'obstacles. En d'autres mots, il est question de représenter le comportement du flux par son passage à travers une image maillée constituée de rues et d'îlots de bâti, en vue d'intégrer les résultats dans un modèle de simulation hydraulique.

Le premier objectif consiste à représenter fidèlement les blocs de bâti malgré l'intégration des données sous forme de mailles. En effet, dans leur modèle, les ingénieurs en hydraulique représentent les données sous forme d'une grille régulière. Puisque les résultats de ce travail leurs sont destinés, une unité carrée a également été sollicitée ici. Le problème rencontré fut de gérer les mailles où une part de celle-ci est de type voirie, et l'autre partie de type obstacle. Comment conserver la précision fournie par les données, et ne pas généraliser chaque maille à une seule affectation? Est-ce qu'il en ressort un réel avantage par rapport à un modèle qui généraliserait l'affectation à l'échelle de la maille?

Le second objectif concerne la simulation du comportement d'un écoulement dans le modèle. Nous allons tenter de représenter la propagation d'un écoulement dans une image parsemée d'obstacles à l'aide d'outils de SIG.

Enfin, pour répondre au dernier objectif, il est question de trouver une méthodologie permettant de représenter l'information issue de la propagation à différentes échelles. En effet, nous travaillons ici sur un modèle aux dimensions restreintes. Cependant, dans un cas réel, la quantité de données risque d'être trop importante pour les traiter à haute résolution. Les coûts en termes de temps et de technologie seraient trop importants. Nous souhaitons construire un modèle permettant de représenter l'information à des mailles de résolution basse, c'est-à-dire généralisant les données à des mailles plus larges.

Maintenant que les objectifs sont définis et avant d'aller plus loin, il est nécessaire de préciser quelques concepts qui sont utilisés régulièrement dans ce mémoire.

Premièrement, il existe différents types d'**inondations** au sens large. En effet, nous pouvons parler d'inondations maritimes ou fluviales, de crues subites dites « *flash floods* » dans la littérature anglophone, d'inondation urbaine, etc. Les crues subites sont des événements particulièrement rapides qui laissent peu de temps aux habitants pour s'y préparer (Sene, 2013). Elles peuvent être subdivisées en différents types en fonction de l'élément qui les a engendrées. Ainsi, Sene (2013) différencie les débordements des rivières à la suite de fortes pluies ou à la fonte de neige ; les inondations conséquentes aux embâcles de glace ; les laves torrentielles ; les inondations urbaines lorsque le



réseau de drainage ne parvient pas à récupérer toutes les eaux de pluie ; les ruptures de barrage ; les inondations brutales dues par exemple à la vidange d'un lac glaciaire ainsi que les inondations dues aux brèches de digues. Dans le présent travail, il est question d'étudier le comportement des inondations dans le cas le plus simple, peu important leurs origines. Nous verrons également plus loin dans ce travail que les simulations demandent une entrée d'écoulement à partir d'un point ou d'une zone précise. L'inondation ne se déclenche pas en tout point de la région considérée comme cela pourrait être le cas lors d'une pluie diluvienne. Le concept d'inondation choisi ici est donc très général. Il s'agit plutôt d'un écoulement dont la cause et la composition important peu.

Bien que nous parlions de **région urbaine**, aucun système de drainage anthropique n'est pris en compte. Le terme urbain s'intègre plutôt dans la définition des obstacles qui présentent une forme polygonale et dans celle des axes d'écoulement qui ont été choisis rectilignes à l'image d'une ville très généralisée.

Par ailleurs, le terme « **obstacle** », désigne ici un îlot c'est-à-dire un ensemble de bâtiments formant un bloc uni. Cet îlot obstruera le passage de l'écoulement. A contrario, les rues (ou voiries) sont des voies libres de circulation pour le flux. Notons que l'**écoulement** considéré dans ce travail est à surface libre, c'est-à-dire que la pression à la surface de l'eau égale la pression atmosphérique. L'écoulement dépend donc essentiellement de la pente, s'il y en a une. Ensuite, aucune charge sédimentaire et aucun effet de frottement contre les obstacles ne seront considérés.

Enfin, le concept de « **porosité** » sera également utilisé. Il est développé pour le modèle numérique maillé et se définit comme la part de voirie dans une maille unitaire. Lorsque la porosité est de 1, cela signifie que l'entièreté de la maille est composée de voirie. Par opposition, une maille couverte d'un obstacle présentera une porosité de 0, c'est-à-dire que le pourcentage de voirie la composant est nul.

## 4. Hypothèses

---

L'objectif premier de ce mémoire est de représenter les obstacles de façon adéquate dans un modèle maillé sans perdre en précision. Dès lors, nous pensons à utiliser les principes de l'algorithme d'*antialiasing* pour répondre à cet objectif. Ce dernier sera expliqué en détails lors de son utilisation (voir Section 6.1.1).

En ce qui concerne la représentation du comportement d'un écoulement dans un modèle constitué de voiries et de zones de bâti, nous avons envisagé d'étudier cet écoulement comme le franchissement d'une surface jonchée d'obstacles. Le terme de franchissement est important, car il induit une notion d'effort. De plus, aucun paramètre initial de débit ou de direction ne sera défini. Il s'agit de tenter de représenter l'effort d'un écoulement pour parcourir les rues depuis une source jusqu'à un point du modèle et de déduire une valeur de propagation à ce point. Les « surfaces de coûts » seraient une solution raster à la modélisation du franchissement bidimensionnel et permettraient donc de répondre à la problématique posée par les hydrauliciens. En effet, différents logiciels de SIG proposent des algorithmes de « distance-coûts » basés sur la distance à parcourir et sur le coût nécessaire pour aller d'un certain point à un autre. Le coût serait dès lors représenté par la distance ou par la surface de friction à traverser dans un modèle rasterisé.

Enfin, nous avons émis l'hypothèse que l'agrégation des pixels permettrait de représenter l'information de propagation d'un écoulement à des mailles de résolution plus basse que celle des données initiales. L'agrégation des pixels consiste ici à calculer la moyenne de ces pixels et de l'assigner à la maille formée par leur assemblage.

# 5. Données

---

## 5.1. Les données de base

Les données originales utilisées sont issues des travaux d'Araud (2012) et de Debaucheron (2014). Elles ont été retraitées dans le cadre d'un doctorat en cours et fournies par le doctorant en charge M. Bruwier.

A l'origine, Araud (2012) a développé un pilote d'inondation au cours de sa thèse de doctorat à Strasbourg. Il s'agit d'un dispositif expérimental en plexiglas (voir Figure 1). Ce modèle réduit a une superficie de  $5\text{m} \times 5\text{m}$ . Il ne représente pas une ville réelle mais s'approche d'un quartier européen de type « moyen » de  $1\text{km}$  de côté à l'échelle  $1/200$  (Araud, 2012; Debaucheron, 2014). Il est agencé en 64 îlots imperméables représentant des blocs de bâti et est parcouru par 7 rues selon l'axe Nord-Sud et 7 rues selon l'axe Ouest-Est. Ces rues présentent une largeur soit de  $5\text{cm}$ , soit de  $12,5\text{cm}$ , et s'intersectent selon différents angles.



Figure 1 - Vue d'ensemble du pilote d'inondation (Araud, 2012)

Araud (2012) a utilisé le modèle réduit afin d'expérimenter les écoulements en laboratoire. Il a en effet injecté des débits à l'entrée des rues des faces Nord et Ouest du pilote. De cette façon, il a pu étudier le comportement hydraulique du flux. Les résultats qui en découlèrent furent utilisés par Debaucheron (2014) dans le cadre de son mémoire afin d'analyser dans quelles mesures la simulation hydraulique produite par le modèle WOLF reproduit les écoulements observés expérimentalement. WOLF2D est un logiciel de simulation hydraulique développé à l'ULg par l'équipe du HECE. Le modèle réduit a donc dû être numérisé pour être implémenté dans le logiciel.

Ce sont ces données numériques que nous emploierons dans ce travail. En fait, pour chaque sommet des polygones représentant les îlots, les coordonnées  $x$  et  $y$  sont

fournies. A la base, les sommets ont été numérisés en situant l'origine du système au Sud-Est (Debaucheron, 2014). Cependant, les données ont été manipulées par M. Bruwier afin qu'elles correspondent à un système d'axe de même échelle mais dont l'origine se situe à l'intersection des faces Sud et Ouest. Ce sont ces dernières qui servent de base à nos traitements.

Les coordonnées sont fournies en mètres, avec une précision millimétrique. Au total, 255 points en 2D sont ainsi mis à disposition. En plus des sommets, les identifiants des îlots auxquels ils se rapportent sont précisés. Ces données ont été compilées dans un tableur *Excel* pour plus de lisibilité (voir Annexe 1). Les îlots, ou polygones, sont nommés par une lettre suivie d'un chiffre. La lettre, qui démarre à A et qui est incrémentée après chaque axe de rue, correspond à l'ordre de l'îlot par rapport à l'axe Ouest-Est. Le chiffre démarre à 1 et s'incrémente également après chaque rue, mais par rapport à l'axe Nord-Sud. Les polygones vont ainsi de A1 à H8 et sont au nombre de 64 (voir Figure 2).

Avant de démarrer les traitements, les données ont été enregistrées au format CSV (*Comma-Separated Values*). Elles peuvent ainsi être directement importées dans le logiciel QGIS sous forme d'une couche *Shapefile* vectorielle (voir Annexe 2) avec une table d'attributs associée comprenant les colonnes « x », « y » et l'identifiant du « polygone ». QGIS est un logiciel qualifié de système d'information géographique. Il a été choisi comme outil de travail car il est en libre accès et bien adapté à la gestion des couches vectorielles. Le format vectoriel est employé dans un premier temps afin de ne pas généraliser l'information à la résolution des pixels comme ce serait le cas sous un format maillé (raster). Etant donné que les dimensions du modèle ne dépassent pas quelques mètres, le choix d'un système de projection plan tel que, par exemple, le Lambert Belge 72 importe peu. En effet, le modèle ne devrait pas être affecté par des erreurs de projection étant donné sa faible étendue. Les traitements présentés ci-après ont cependant été réalisés en définissant le système local de coordonnées planes en mètres.

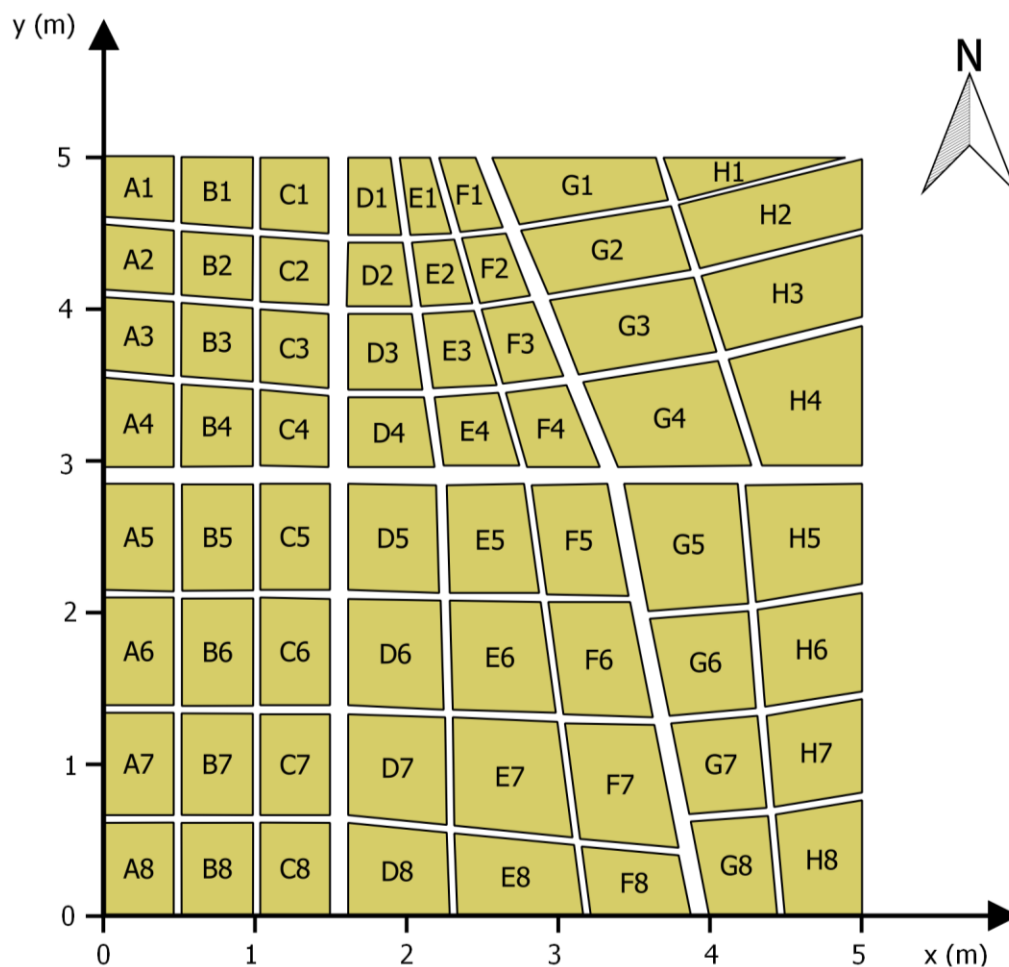


Figure 2 - Représentation schématique des données de base

## 5.2. Les données résultantes

La demande des ingénieurs hydrauliciens consiste à leur fournir une méthode de calcul de valeurs à implémenter dans leur modèle. L'intérêt d'une méthode est de pouvoir l'utiliser à plusieurs reprises, avec différents jeux de données. Ils pourront intégrer la ou les méthodes proposées dans le modèle hydraulique qu'ils conçoivent. Les résultats seront donc accompagnés de modules de traitements présentés sous forme de « modèles ». En effet, les différents logiciels employés dans ce travail proposent des modeleurs permettant de décrire les différents traitements à enchaîner pour obtenir les résultats voulus. Ces modèles peuvent être enregistrés et réutilisés pour différentes simulations en changeant uniquement les données et les paramètres en entrée. Dans la mesure du possible, certains modèles seront convertis en programmes indépendants (« scripts ») de modélisation.

### 5.3. Les logiciels de SIG

Pour effectuer les différents traitements, trois logiciels de SIG sont utilisés : QGIS Desktop (2.12.3), IDRISI Selva (17.0) et ArcGIS (10.3.1). Si certaines opérations sont spécifiques ou plus aisées sur l'un ou l'autre logiciel en particulier, elles peuvent généralement être accomplies sur plusieurs de ces logiciels.

# 6. Traitements

---

## 6.1. Représentation des obstacles

### 6.1.1. Conservation de l'information

Tant qu'à présent, les sommets des îlots bâtis ont simplement été importés dans QGIS dans une couche *Shapefile*. Ils sont représentés en mode ponctuel. Cependant, les îlots en tant que tels sont des obstacles imperméables. Il est plus adéquat de les représenter par des entités polygonales. Pour ce faire, le plugin *Points2One* a été utilisé. Il permet de créer des polygones à partir de la couche des données ponctuelles. Les entités sont groupées selon l'attribut « polygone » correspondant au nom des îlots et les polygones ainsi construits sont enregistrés dans une nouvelle couche *Shapefile* nommée *Donnees\_poly* (voir Figure 3).

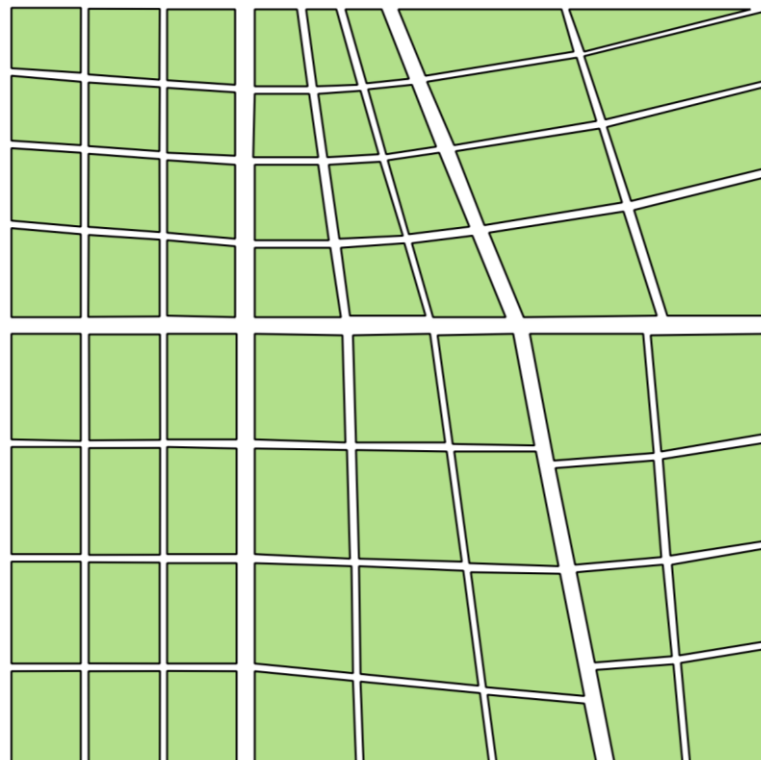


Figure 3 - Couche *Shapefile Donnees\_poly*

Comme cité précédemment, les données sont sous format vectoriel pour ne pas perdre de l'information due à la rasterisation. La rasterisation adapte les données numérisées à une grille régulière de pixels. Le pixel est la primitive graphique du mode raster et se définit comme une maille rectangulaire, généralement carrée, en 2D

(Donnay, 2012). Pour comprendre où se perd l'information lors de la rasterisation, référons nous à l'algorithme de Bresenham (voir Figure 4).

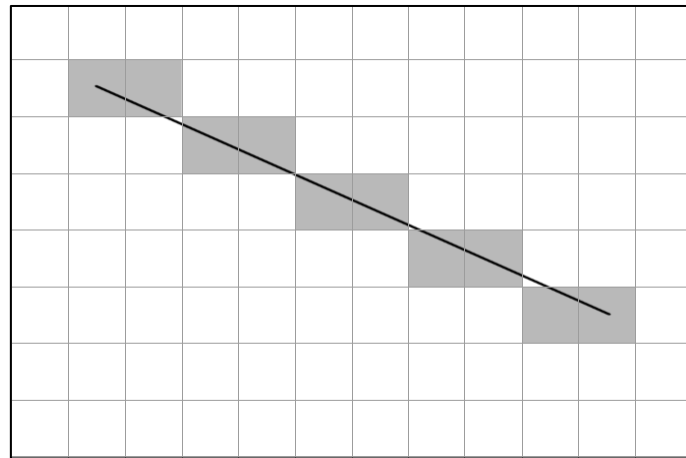


Figure 4 - Représentation de l'algorithme de Bresenham

Cet algorithme vise à repérer les points du maillage les plus adéquats à tracer pour dessiner une droite (Agoston, 2005). A l'origine, Bresenham développe cet algorithme pour l'utilisation des traceurs, ou *plotters*, dans l'impression graphique de vecteurs (Bresenham, 1965; Salomon, 2011). Il se base sur les huit mouvements linéaires qui peuvent être effectués par le traceur (voir Figure 5).

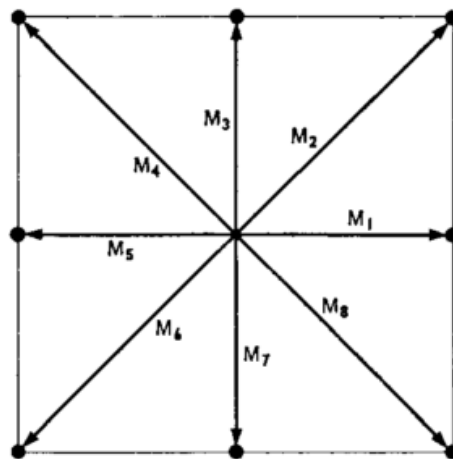


Figure 5 - Mouvements du traceur (Bresenham, 1965)



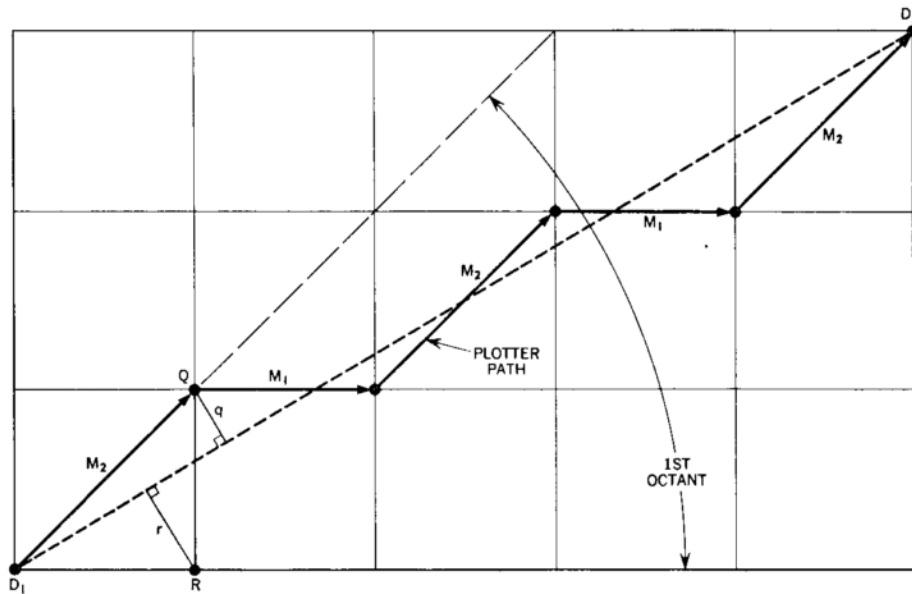


Figure 6 - Séquence de mouvements du traceur dans le premier octant (Bresenham, 1965)

Par exemple, pour une droite située dans le premier octant (c'est-à-dire dont la pente est entre 0 et 45°), Bresenham (1965) se base sur le principe que des tracés  $M_1$  et  $M_2$  suffisent à la représenter (voir Figure 6). Des variantes de cet algorithme ont été développées, comme pour décrire plus efficacement les courbes (Agoston, 2005; Salomon, 2011). Quoiqu'il en soit, lorsque les vecteurs sont transposés en mode maillé, ils sont approximés par une série de pixels en suivant le principe de base de l'algorithme de Bresenham. Cette approximation donne lieu à une perte de précision puisque les éléments représentant les droites sont désormais des surfaces. Pour la représentation des îlots de notre modèle, l'utilisation du mode vecteur permet de s'affranchir de cette généralisation. La rastérisation entrainerait la considération de pixels entiers qui, pourtant, ne seraient couverts qu'en partie par l'obstacle de bâti. De la même façon, il y aurait des pixels non-considérés alors qu'une partie de leur surface devrait l'être. En plus, les éléments linéaires rastérisés présentent un effet d'escalier, dit *aliasing* en anglais (voir Figure 7).

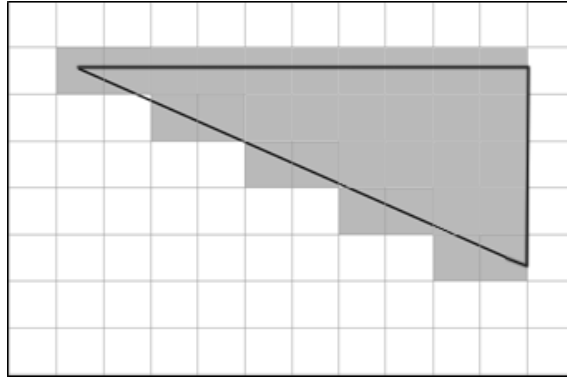


Figure 7 - Représentation de la rasterisation d'un flot

Nonobstant, dans le modèle hydraulique en cours de mise au point par le HECE, l'information utilisée est rasterisée. Il est donc impératif que les modèles réalisés ici le soient également. Dès lors, comment représenter les obstacles en mode raster sans pour autant perdre de l'information ?

Directement, c'est le principe d'*antialiasing* qui nous vient à l'esprit pour résoudre cette question. En effet, cet algorithme peut utiliser les aires des pixels et découler ainsi directement de l'algorithme de Bresenham (Agoston, 2005). En fait, bien qu'il existe plusieurs méthodes d'*antialiasing* (Herubel *et al.*, 2011), toutes se basent sur la considération des pixels non pas comme des points mais bien comme des entités bidimensionnelles occupant une superficie (Salomon, 2011).

Des méthodes simples et intuitives d'*antialiasing* existent, telles que la coloration en ton de gris inversement proportionnelle à la distance entre le pixel et la ligne (Salomon, 2011), ou la coloration en fonction de l'aire du pixel à l'intérieur de la ligne. D'après Salomon (2011), ces méthodes simples sont parfois insatisfaisantes visuellement (voir Figure 8), c'est pourquoi des techniques d'*antialiasing* plus complexes ont été développées. Généralement, les algorithmes d'*antialiasing* sont utilisés pour résoudre les effets d'escaliers et donner un meilleur rendu visuel, que cela soit pour des polices d'écriture, des images ou même en animation (Agoston, 2005; Herbuel *et al.*, 2011; Salomon, 2011).

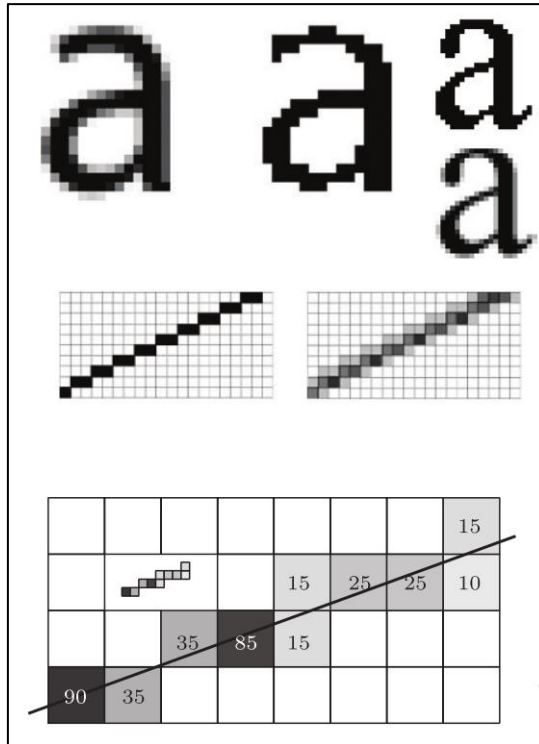


Figure 8 - Exemples d'antialiasing avec des tons de gris (Salomon, 2011)

En ce qui nous concerne, le problème n'est pas l'aspect visuel des lignes mais la conservation de l'information malgré la résolution des pixels choisie. Une idée très simple serait d'augmenter la résolution de la grille et d'utiliser des mailles millimétriques. Cependant, deux raisons s'y opposent. Premièrement, la génération de la grille serait un traitement lourd à mettre en place à une telle précision. Deuxièmement, le modèle utilisé dans ce travail doit pouvoir être utilisé dans celui des hydrauliciens du HECE, or ceux-ci désirent y appliquer des données spatiales centimétriques. Travailler à une précision plus grande serait donc inutile. Ainsi, tout comme l'algorithme d'antialiasing, nous allons utiliser les aires des mailles et étudier l'intersection vectorielle des polygones obstacles dans chacune d'entre elles. Ces principes devraient permettre la représentation exacte des obstacles malgré la taille de maille choisie.

Il est question, pour chaque maille, de définir la part de sa superficie couverte par la voirie et donc accessible aux écoulements. Cette part est la porosité de la maille (voir Section 3). Par après, lors de la rasterisation des données, il suffira d'assigner à chaque maille sa valeur de porosité et de la représenter.

### 6.1.2. Méthode de calcul de la porosité

Le modèle suivant (voir Figure 9) permet d'obtenir la porosité sous le logiciel QGIS. Nous allons le détailler étape par étape.

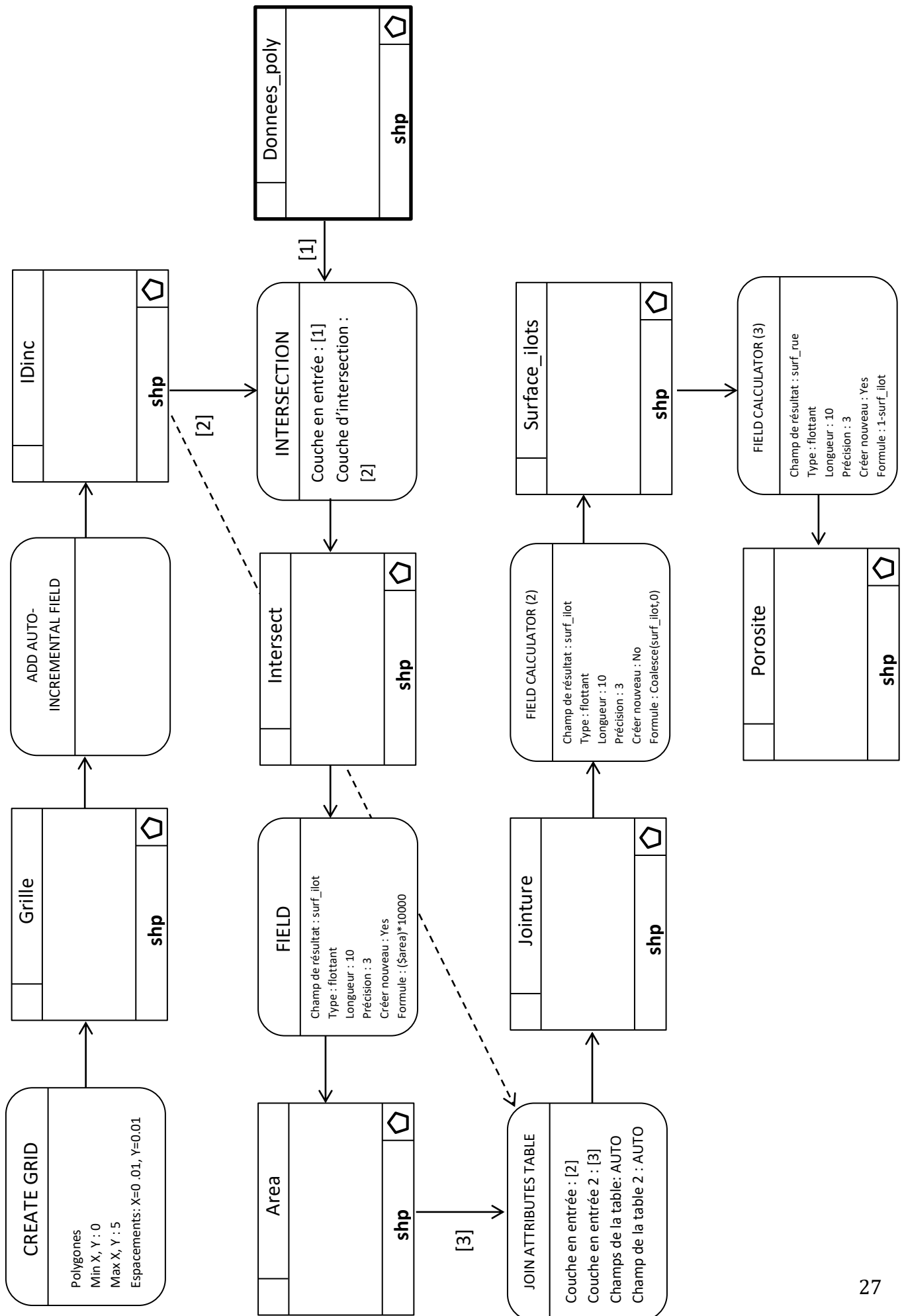


Figure 9 – Organigramme de porosité sous QGIS

- 1) GRILLE VECTEUR: cette opération consiste à créer une grille vectorielle de mailles carrées. Les dimensions de la grille doivent correspondre à celles du modèle de base composé des rues et des blocs de bâti (*Données\_poly*). Dans ce cas-ci, la grille créée est carrée et mesure 5 m de côté, le système de projection choisi étant le plan et l'unité le mètre. L'espacement horizontal et vertical est de 0,01, ainsi nous obtenons 250 000 mailles carrées de 1 cm de côté. Les dimensions des maillages ont été choisies pour répondre à la demande des ingénieurs en hydraulique. En effet le modèle qu'ils mettent en place utilise cette résolution. La couche *Shapefile* en sortie de ce traitement est nommée *Grille*.

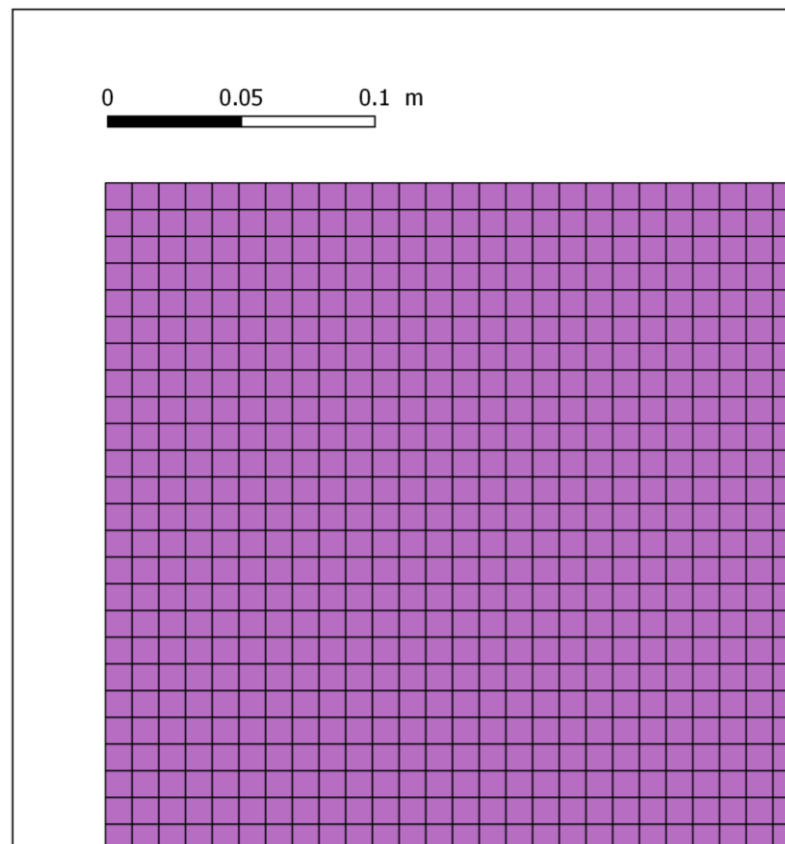


Figure 10 - Coin Nord-Ouest de la couche *Shapefile Grille*

- 2) ADD AUTOINCREMENTAL FIELD: ce plugin permet la création d'un nouveau champ attributaire auto-incrémenté commençant à 1. Il fournit de cette façon un identifiant (*ID*) propre à chaque maille de la grille. Il est nécessaire pour identifier les mailles et permettra, par la suite, de joindre les tables attributaires. Le champ attributaire est nommé *AUTO* par le logiciel. La couche en sortie, *IDinc*, présente les mêmes entités et les mêmes propriétés que la couche *Grille*, à la différence qu'elle dispose du champ supplémentaire *AUTO* dans sa table d'attributs (voir Figure 11).

	left	top	right	bottom	AUTO
0	0.0000000000...	0.0000000000...	0.0100000000...	0.0100000000...	1
1	0.0000000000...	0.0100000000...	0.0100000000...	0.0200000000...	2
2	0.0000000000...	0.0200000000...	0.0100000000...	0.0300000000...	3
3	0.0000000000...	0.0300000000...	0.0100000000...	0.0400000000...	4
4	0.0000000000...	0.0400000000...	0.0100000000...	0.0500000000...	5
5	0.0000000000...	0.0500000000...	0.0100000000...	0.0600000000...	6
6	0.0000000000...	0.0600000000...	0.0100000000...	0.0700000000...	7
7	0.0000000000...	0.0700000000...	0.0100000000...	0.0800000000...	8
8	0.0000000000...	0.0800000000...	0.0100000000...	0.0900000000...	9
9	0.0000000000...	0.0900000000...	0.0100000000...	0.1000000000...	10
10	0.0000000000...	0.1000000000...	0.0100000000...	0.1100000000...	11
11	0.0000000000...	0.1100000000...	0.0100000000...	0.1200000000...	12
12	0.0000000000...	0.1200000000...	0.0100000000...	0.1300000000...	13
13	0.0000000000...	0.1300000000...	0.0100000000...	0.1400000000...	14
14	0.0000000000...	0.1400000000...	0.0100000000...	0.1500000000...	15

Figure 11 - Aperçu de la table d'attributs de la couche *Shapefile IDinc*

- 3) INTERSECTION: l'intersection de la couche *Donnees\_poly* (voir Figure 3) avec la couche *IDinc* permet d'obtenir en sortie une couche *Shapefile Intersect* (voir Figure 12) reprenant les entités de la couche d'entrée, ici *IDinc*, qui sont couvertes par les entités de la couche d'intersection *Donnees\_poly*. En d'autres termes, la couche *IDinc* a été découpée par la couche *Donnees\_poly* comme si cette dernière constituait un masque. Seules les entités superposées par les polygones de *Donnees\_poly* sont conservées. Pour les entités partiellement couvertes par les polygones, elles sont découpées le long de la frontière de façon à ce que seule la partie comprise à l'intérieur des polygones soit conservée. La table attributaire de la couche en sortie reprend les champs des deux couches intersectées (voir Figure 13).

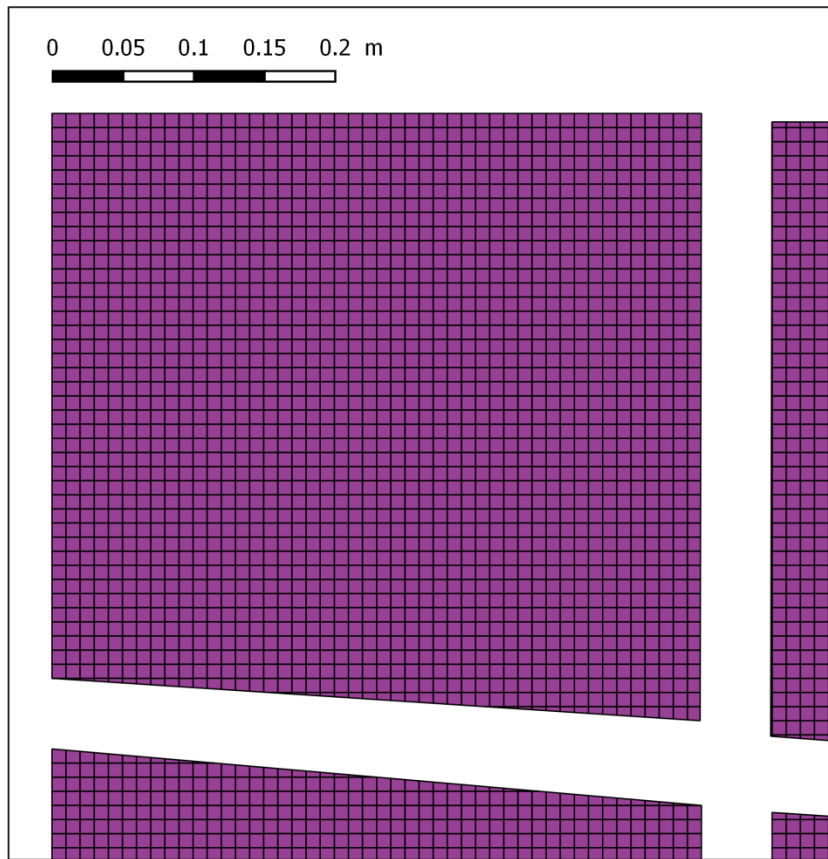


Figure 12 - Coin Nord-Ouest de la couche *Shapefile Intersect*

Table attributaire - intersect :: Total des entités: 210940, filtrés: 210940, sélectionnés: 0

	x	y	polygone	ID-pt	left	top	right	bottom	AUTO
20374	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.0000000000...	0.0100000000...	0.0100000000...	1
20373	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.0100000000...	0.0100000000...	0.0200000000...	2
20378	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.0200000000...	0.0100000000...	0.0300000000...	3
20379	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.0300000000...	0.0100000000...	0.0400000000...	4
20377	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.0400000000...	0.0100000000...	0.0500000000...	5
20326	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.0500000000...	0.0100000000...	0.0600000000...	6
20325	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.0600000000...	0.0100000000...	0.0700000000...	7
20317	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.0700000000...	0.0100000000...	0.0800000000...	8
20318	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.0800000000...	0.0100000000...	0.0900000000...	9
20361	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.0900000000...	0.0100000000...	0.1000000000...	10
20354	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.1000000000...	0.0100000000...	0.1100000000...	11
20356	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.1100000000...	0.0100000000...	0.1200000000...	12
20347	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.1200000000...	0.0100000000...	0.1300000000...	13
20349	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.1300000000...	0.0100000000...	0.1400000000...	14
20339	0.0001000000...	-0.003000000...	A8	32	0.0000000000...	0.1400000000...	0.0100000000...	0.1500000000...	15

Montrer toutes les entités

Figure 13 - Aperçu de la table d'attributs de la couche *Shapefile Intersect*

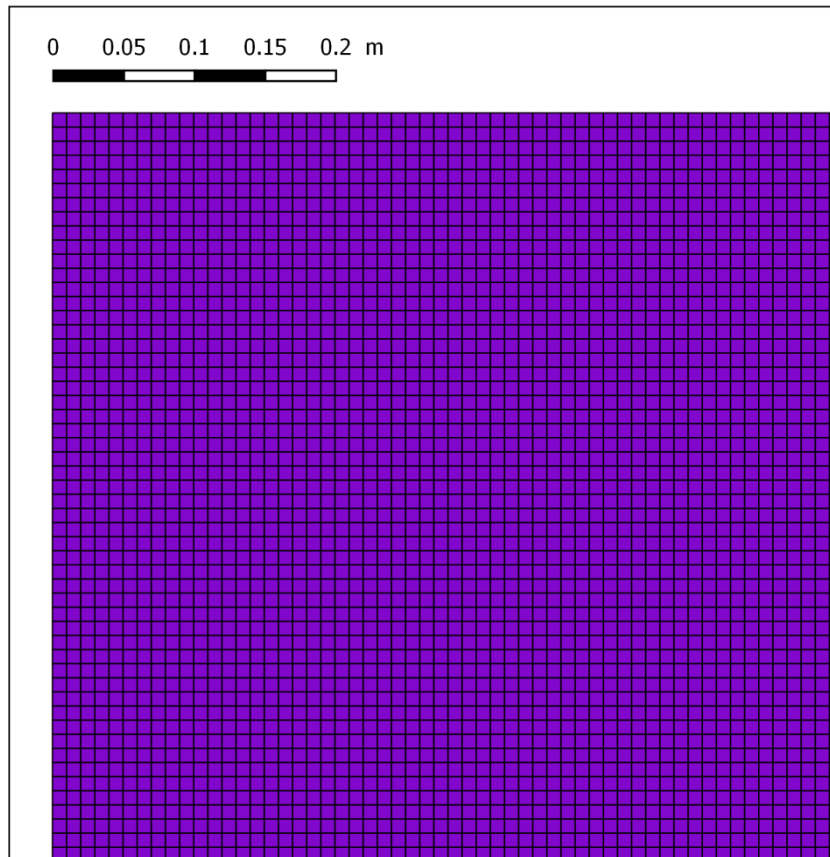
- 4) FIELD CALCULATOR: au cours de cette étape, un champ attributaire *surf\_ilot* est créé et ajouté à la table (voir Figure 14) de la couche *Intersect*. Il est destiné à recevoir la valeur de superficie des entités découpées lors de l'étape précédente. La formule calculant la valeur pour chaque entité est :  $(\$area) * 10\ 000$ . La fonction *area* est disponible dans la calculatrice du logiciel. Quant au facteur 10 000, il permet de fournir les valeurs en  $cm^2$  étant donné que le système original est métrique. Puisque les mailles font  $1\ cm^2$ , obtenir des valeurs dans la même unité permet d'avoir un rapport unitaire. Ainsi, une entité dont la superficie est de 1 correspond à une maille entièrement couverte par un polygone ou un obstacle. Il résulte de ce traitement la couche nommée *Area*.

	gone	ID-pt	left	top	right	bottom	AUTO	surf_ilot
73140		116	2.1899999999...	2.4199999999...	2.2000000000...	2.4300000000...	109743	1.000000
73141		116	2.1899999999...	2.4300000000...	2.2000000000...	2.4399999999...	109744	1.000000
73142		116	2.1899999999...	2.4399999999...	2.2000000000...	2.4500000000...	109745	1.000000
73135		116	2.1899999999...	2.4500000000...	2.2000000000...	2.4600000000...	109746	1.000000
73136		116	2.1899999999...	2.4600000000...	2.2000000000...	2.4700000000...	109747	1.000000
73137		116	2.1899999999...	2.4700000000...	2.2000000000...	2.4800000000...	109748	0.996479
73138		116	2.1899999999...	2.4800000000...	2.2000000000...	2.4900000000...	109749	0.971831
73131		116	2.1899999999...	2.4900000000...	2.2000000000...	2.5000000000...	109750	0.943662
73132		116	2.1899999999...	2.5000000000...	2.2000000000...	2.5100000000...	109751	0.915493
73133		116	2.1899999999...	2.5100000000...	2.2000000000...	2.5200000000...	109752	0.887324
73134		116	2.1899999999...	2.5200000000...	2.2000000000...	2.5300000000...	109753	0.859155
73127		116	2.1899999999...	2.5300000000...	2.2000000000...	2.5400000000...	109754	0.830986
73128		116	2.1899999999...	2.5400000000...	2.2000000000...	2.5500000000...	109755	0.802817
73129		116	2.1899999999...	2.5500000000...	2.2000000000...	2.5600000000...	109756	0.774648

Figure 14 - Aperçu de la table d'attributs de la couche *Shapefile Area*

- 5) JOIN ATTRIBUTES TABLE: cette option joint à la table d'attributs de la couche en entrée *IDinc* les attributs de la table de la couche *Area* (voir Figure 16). Les deux couches disposent d'un attribut commun, à savoir *AUTO* qui correspond à l'ID des mailles. C'est cet attribut commun qui permet de joindre les deux tables tel que les entités reprises dans les deux soient fusionnées. La couche en sortie est intitulée *Jointure* (voir Figure 15). Elle contient les 250 000 entités et tous les attributs de *IDinc*, avec en plus les attributs de *Area*.





**Figure 15 - Coin Nord-Ouest de la couche *Shapefile Jointure***

Table attributive - jointure : Total des entités: 250000, filtrés: 250000, sélectionnés: 0

	left	top	right	bottom	AUTO	x	y	polygone	ID-pt	left_2	top_2	right_2	bottom_2	auto_2	surf_ilot
10338	0.2000000000...	3.3799999999...	0.2100000000...	3.3900000000...	10339	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.3800000000...	0.2100000000...	3.3900000000...	10339	1.000000
10339	0.2000000000...	3.3900000000...	0.2100000000...	3.3999999999...	10340	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.3900000000...	0.2100000000...	3.4000000000...	10340	1.000000
10340	0.2000000000...	3.3999999999...	0.2100000000...	3.4100000000...	10341	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.4000000000...	0.2100000000...	3.4100000000...	10341	1.000000
10341	0.2000000000...	3.4100000000...	0.2100000000...	3.4199999999...	10342	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.4100000000...	0.2100000000...	3.4200000000...	10342	1.000000
10342	0.2000000000...	3.4199999999...	0.2100000000...	3.4300000000...	10343	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.4200000000...	0.2100000000...	3.4300000000...	10343	1.000000
10343	0.2000000000...	3.4300000000...	0.2100000000...	3.4399999999...	10344	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.4300000000...	0.2100000000...	3.4400000000...	10344	1.000000
10344	0.2000000000...	3.4399999999...	0.2100000000...	3.4500000000...	10345	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.4400000000...	0.2100000000...	3.4500000000...	10345	1.000000
10345	0.2000000000...	3.4500000000...	0.2100000000...	3.4600000000...	10346	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.4500000000...	0.2100000000...	3.4600000000...	10346	1.000000
10346	0.2000000000...	3.4600000000...	0.2100000000...	3.4700000000...	10347	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.4600000000...	0.2100000000...	3.4700000000...	10347	1.000000
10347	0.2000000000...	3.4700000000...	0.2100000000...	3.4800000000...	10348	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.4700000000...	0.2100000000...	3.4800000000...	10348	1.000000
10348	0.2000000000...	3.4800000000...	0.2100000000...	3.4900000000...	10349	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.4800000000...	0.2100000000...	3.4900000000...	10349	1.000000
10349	0.2000000000...	3.4900000000...	0.2100000000...	3.5000000000...	10350	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.4900000000...	0.2100000000...	3.5000000000...	10350	1.000000
10350	0.2000000000...	3.5000000000...	0.2100000000...	3.5100000000...	10351	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.5000000000...	0.2100000000...	3.5100000000...	10351	1.000000
10351	0.2000000000...	3.5100000000...	0.2100000000...	3.5200000000...	10352	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.5100000000...	0.2100000000...	3.5200000000...	10352	1.000000
10352	0.2000000000...	3.5200000000...	0.2100000000...	3.5300000000...	10353	0.0001000000...	2.9500000000...	A4	16	0.2000000000...	3.5200000000...	0.2100000000...	3.5300000000...	10353	0.213990
10353	0.2000000000...	3.5300000000...	0.2100000000...	3.5400000000...	10354	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10354	0.2000000000...	3.5400000000...	0.2100000000...	3.5500000000...	10355	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10355	0.2000000000...	3.5500000000...	0.2100000000...	3.5600000000...	10356	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10356	0.2000000000...	3.5600000000...	0.2100000000...	3.5700000000...	10357	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10357	0.2000000000...	3.5700000000...	0.2100000000...	3.5800000000...	10358	0.0001000000...	3.5900000000...	A3	12	0.2000000000...	3.5700000000...	0.2100000000...	3.5800000000...	10358	0.786010
10358	0.2000000000...	3.5800000000...	0.2100000000...	3.5899999999...	10359	0.0001000000...	3.5900000000...	A3	12	0.2000000000...	3.5800000000...	0.2100000000...	3.5900000000...	10359	1.000000
10359	0.2000000000...	3.5899999999...	0.2100000000...	3.6000000000...	10360	0.0001000000...	3.5900000000...	A3	12	0.2000000000...	3.5900000000...	0.2100000000...	3.6000000000...	10360	1.000000

Montrer toutes les entités

Figure 16 - Aperçu de la table d'attributs de la couche Shapefile Jointure

- 6) FIELD CALCULATOR (2): la fonction  $Coalesce(surf\_ilot, 0)$  utilisée permet de remplacer par 0 les valeurs *NULL* (c'est-à-dire sans valeur) du champ attributaire *surf\_ilot* (voir Figure 16) de la couche *Jointure*. En effet certaines entités ne disposaient pas de valeur pour la porosité car cet attribut est issu de la couche *Area*, or celle-ci n'est composée que des entités intersectée à l'étape 3. Les entités non-intersectées n'avaient donc pas de valeur pour l'attribut de superficie. Une nouvelle couche, *Surface\_ilot*, a été créée en sortie.
- 7) FIELD CALCULATOR (3): la dernière étape de ce traitement consiste à utiliser la surface d'îlot pour obtenir la surface de voirie pour chaque maille. Cela correspond à la porosité recherchée. L'équation  $1 - surf\_ilot$  donne ce résultat puisque le rapport se fait pour une maille unitaire, et que la part non-occupée par les îlots de bâti l'est forcément par de la voirie. Les valeurs sont enregistrées dans un nouveau champ attributaire, *surf\_rue* (voir Figure 17) et la couche résultante est nommée *Porosite*.

	right_2	bottom_2	auto_2	surf_ilot	surf_rue	
0	000...	0.0100000000...	0.0100000000...	1	0.990000	0.010
1	000...	0.0100000000...	0.0200000000...	2	0.990000	0.010
2	000...	0.0100000000...	0.0300000000...	3	0.990000	0.010
3	000...	0.0100000000...	0.0400000000...	4	0.990000	0.010
4	000...	0.0100000000...	0.0500000000...	5	0.990000	0.010
5	000...	0.0100000000...	0.0600000000...	6	0.990000	0.010
6	000...	0.0100000000...	0.0700000000...	7	0.990000	0.010
7	000...	0.0100000000...	0.0800000000...	8	0.990000	0.010
8	000...	0.0100000000...	0.0900000000...	9	0.990000	0.010
9	000...	0.0100000000...	0.1000000000...	10	0.990000	0.010
10	000...	0.0100000000...	0.1100000000...	11	0.990000	0.010
11	000...	0.0100000000...	0.1200000000...	12	0.990000	0.010
12	000...	0.0100000000...	0.1300000000...	13	0.990000	0.010
13	000...	0.0100000000...	0.1400000000...	14	0.990000	0.010
14	000...	0.0100000000...	0.1500000000...	15	0.990000	0.010
15	000...	0.0100000000...	0.1600000000...	16	0.990000	0.010
16	000...	0.0100000000...	0.1700000000...	17	0.990000	0.010
17	000...	0.0100000000...	0.1800000000...	18	0.990000	0.010
18	000...	0.0100000000...	0.1900000000...	19	0.990000	0.010
19	000...	0.0100000000...	0.2000000000...	20	0.990000	0.010
20	000...	0.0100000000...	0.2100000000...	21	0.990000	0.010

Figure 17 – Aperçu de la table d'attributs de la couche *Shapefile Porosite*

### 6.1.3. Discussion

Le principe de l'*antialiasing* a effectivement permis d'obtenir des valeurs de porosité. L'algorithme faisant notamment appel à des notions de topologie, l'utilisation d'un logiciel de SIG était appropriée. Ce dernier nous a également permis de traiter nos données sous forme d'une grille régulière comme demandé par les ingénieurs, et ce malgré le format vectoriel et non raster.

Finalement, le modèle conçu pour extraire la porosité a été reproduit dans le modeleur graphique de QGIS (voir Figure 18). Il pourra ainsi être utilisé comme un algorithme unique autant de fois que souhaité. QGIS permet également d'exporter les modèles en script python (voir Annexe 3).

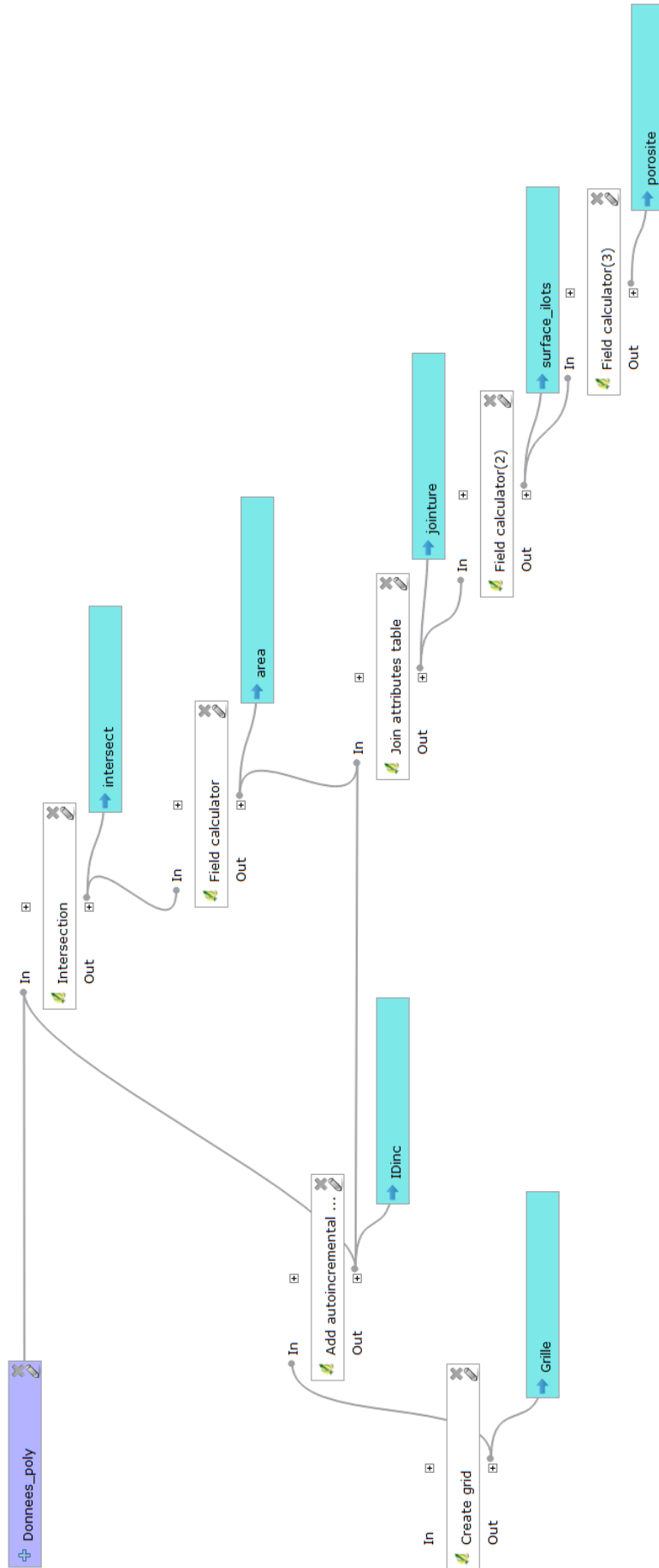


Figure 18 – Modèle de détermination de la porosité sous QGIS

## 6.2. Simulation de propagation d'un écoulement

Au cours de ce travail, il est également question d'étudier le comportement d'un écoulement par rapport aux obstacles placés sur son chemin. Les logiciels de SIG proposent de nombreux outils pour aider à modéliser les écoulements mais les données nécessaires en entrées sont souvent complexes, avec des valeurs de débits, de vitesses, etc. En ce qui nous concerne, nous souhaitons modéliser un écoulement de façon très simple, en se basant essentiellement sur la topologie. Les méthodologies proposées pourront ensuite, éventuellement, servir de base à un modèle hydraulique plus complexe.

Pour répondre à la demande, les données seront traitées en mode raster. Le logiciel utilisé dans un premier temps est IDRISI Selva car plus adapté aux images rasters que QGIS. Avant d'appliquer des traitements au modèle étudié, nous avons d'abord envisagé de les tester sur des prototypes très basiques. Nous avons en effet émis l'hypothèse que les algorithmes de « distance-coût » seraient efficaces pour représenter un écoulement sur une surface jonchée d'obstacles. En les appliquant sur des cas simples, nous déterminerons si ce choix est pertinent.

### 6.2.1. Théorie et tests

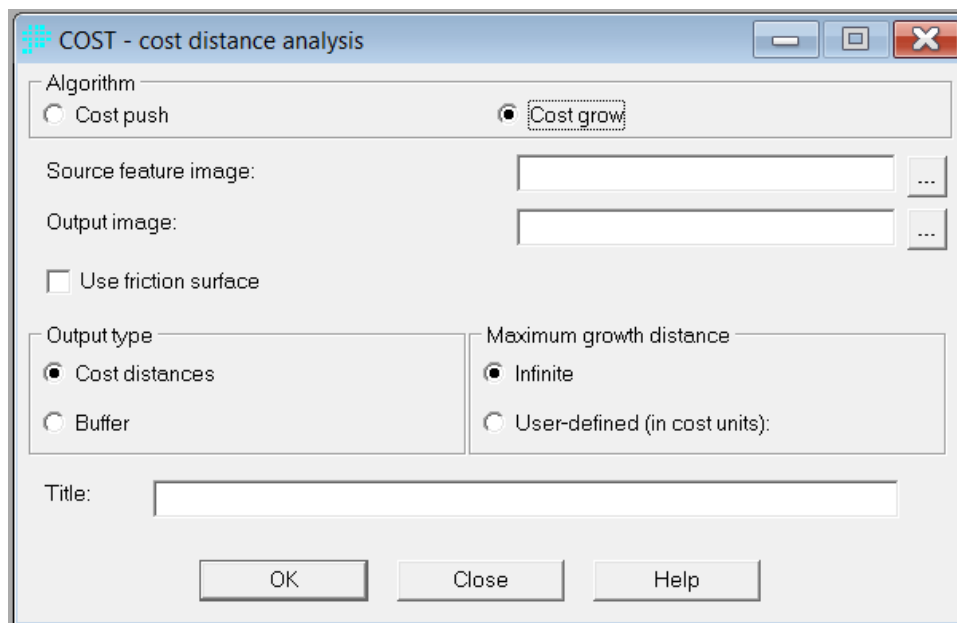
L'algorithme de coût nommé COST proposé par le logiciel IDRISI génère une surface de distance (ou de coûts). La distance mesurée correspond au moindre coût pour se déplacer sur une surface de friction<sup>1</sup>. Le coût de déplacement sur celle-ci dépend de la friction assignée aux cellules. Les distances sont calculées en chaque cellule par le montant minimum de friction cumulée depuis la source de propagation la plus proche, et les mouvements possibles se font dans huit directions (mouvement droit ou en diagonale) depuis une cellule. Le logiciel nous donne le choix entre deux algorithmes de calcul de coûts, « *Cost Push* » et « *Cost Grow* ». Nous utiliserons le second car il est mieux adapté à la représentation de surfaces complexes (présence d'obstacles infranchissables).

D'abord, nous allons effectuer des tests. Des images de 20x20 pixels unitaires sont générées grâce à la fonction INITIAL du logiciel. Nous obtenons ainsi des images rasters de 20 colonnes et de 20 rangées, numérotées de 0 à 19. Les tests sont tous réalisés sur des images de mêmes dimensions. Ces images sont en partie utilisées pour créer des surfaces de friction avec des obstacles imperméables. Une image est cependant employée comme source de propagation. Elle détermine à partir de quel(s) pixel(s) doit se propager l'algorithme.

---

<sup>1</sup> Les informations relatives à l'outil COST proviennent du système d'aide d'IDRISI Selva.

Outre l'image de source et la surface de friction, le lancement de l'algorithme COST demande également en entrée d'assigner un nom à l'image en sortie, de lui fournir éventuellement un titre, de définir la valeur de distance maximale et de cocher le type des valeurs en sortie (voir Figure 19). Pour cette dernière option, le choix s'effectue entre « *Cost distances* » qui fournit comme valeur le coût par pixel et « *Buffer* » qui est booléen et assigne la valeur 1 pour les pixels compris dans un corridor (*buffer*). C'est évidemment le type « *Cost distances* » qui nous intéresse. Par ailleurs, nous ne limitons pas la propagation à une distance définie.



**Figure 19 - Interface de l'outil COST d'IDRISI**

Notons que si aucune surface de friction n'est définie, une friction de valeur 1 est assignée automatiquement à chaque cellule. Au bout du compte, l'image de coût en sortie aura des valeurs correspondant au nombre de cellules franchies depuis la source de la propagation. Ajoutons que pour représenter des obstacles infranchissables, il faut créer une surface de friction en assignant aux obstacles la valeur de friction -1.

Pour donner des valeurs à nos images de départ créées avec la fonction INITIAL, nous utilisons l'option UPDATE. Il ne faut pas oublier que la numérotation des colonnes et des lignes commence à 0. Pour les images de friction, la valeur -1 est assignée aux pixels choisis comme obstacles, et 1 aux pixels où la propagation de la distance peut avoir lieu. Pour l'image servant de source, la valeur 1 est utilisée pour les pixels sources d'où se propagera l'algorithme, et 0 pour les autres (voir Figure 20).



Figure 20 - Image source des tests de distance coût sous IDRISI

L'algorithme COST est lancé une première fois à partir de l'image source mais sans surface de friction. La surface de distance résultante (voir Figure 21) correspond bien au nombre de pixels parcourus pour atteindre chaque pixel. En effet, puisqu'il y a 20 lignes et que la première sert de source de propagation, la valeur de coût sur la dernière doit être de 19. Nous obtenons bien ce résultat.

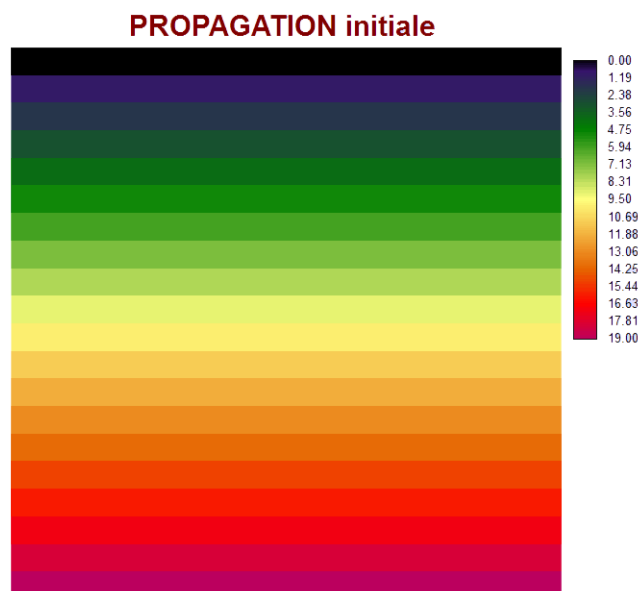


Figure 21 - Surface de coûts obtenue sans surface de friction sous IDRISI

Ensuite, le traitement est lancé à plusieurs reprises à partir de la même image source mais avec des surfaces de friction différentes (voir Figure 22 et Figure 23).



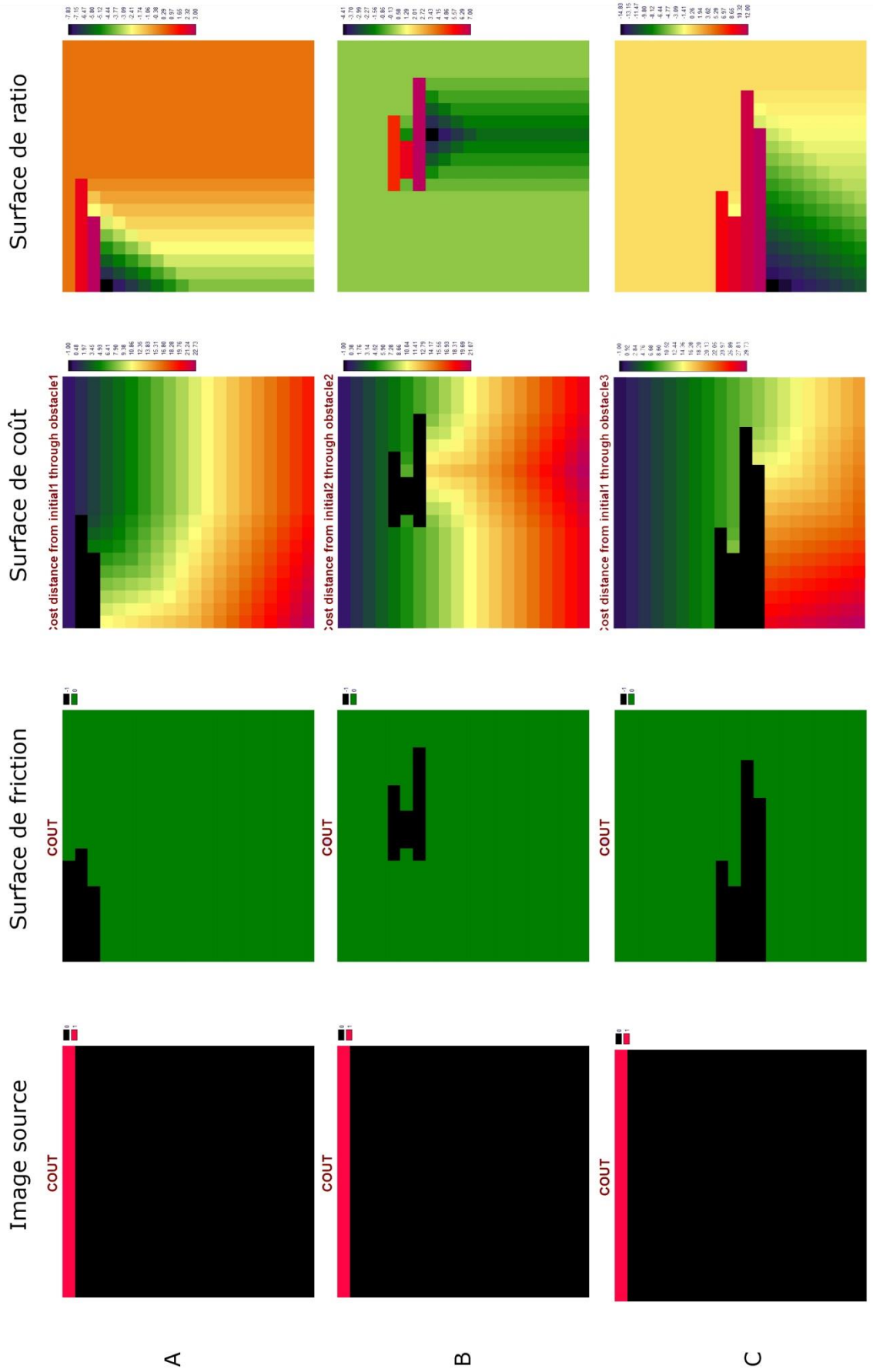


Figure 22 – Images de surfaces de coûts et de leur ratio pour différentes surfaces de friction sous IDRISI

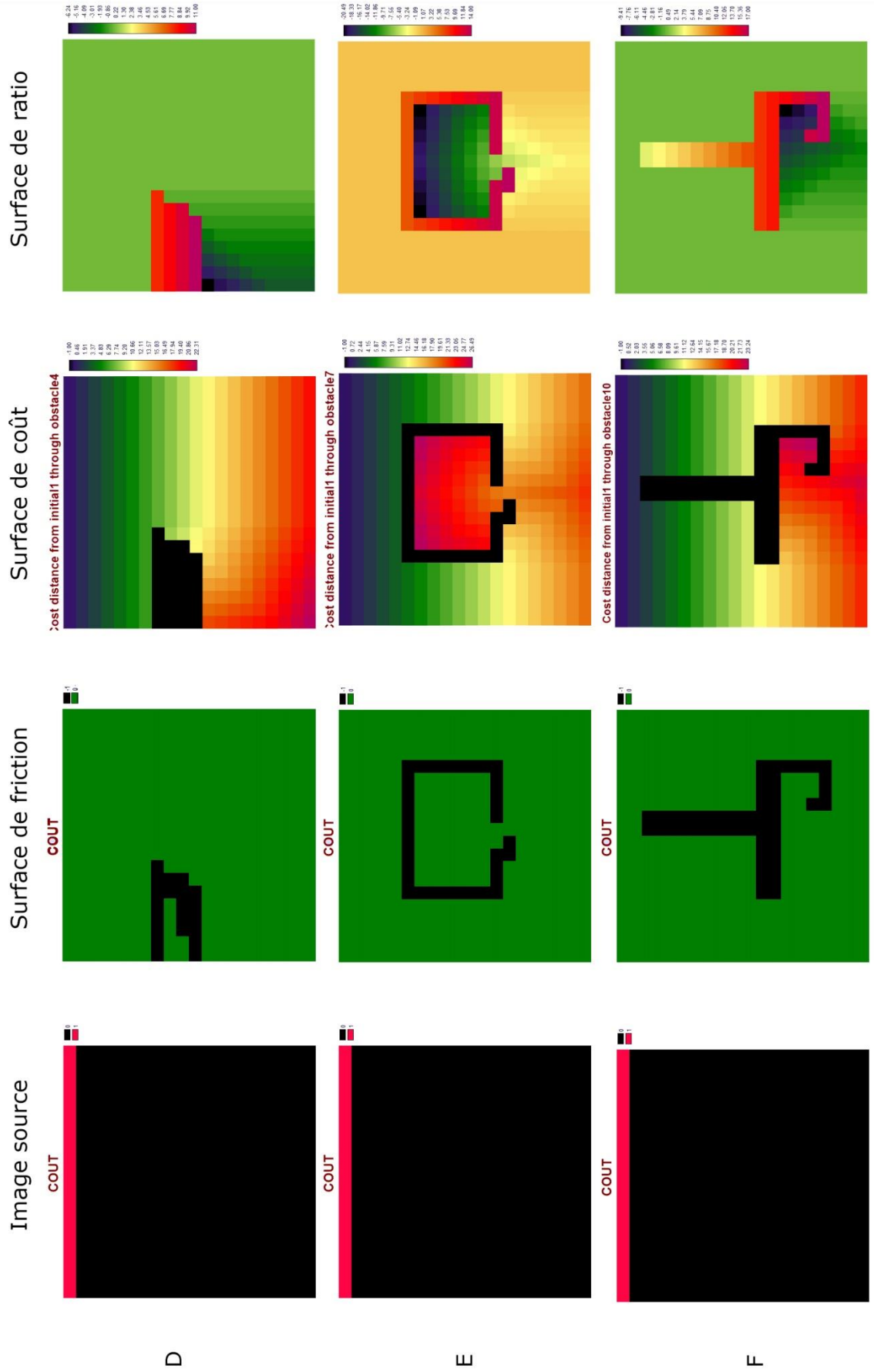


Figure 23 – Images de surfaces de coûts et de leur ratio pour différentes surfaces de friction sous IDRISI

En plus de la surface de coûts, une surface de ratio a été calculée. Elle a été créée grâce à l'outil IMAGE CALCULATOR et correspond à la différence entre la surface de coûts sans obstacle (voir Figure 21) et la surface de coûts dont la surface de friction comprend un obstacle. Le ratio permet de voir les écarts au coût sans surface de friction.

Les cas A, B et C sont assez simples. Intuitivement, la surface coût fait penser au déplacement d'un flux gêné par des obstacles. Si nous prenons maintenant le cas E, plus complexe, nous constatons que le coût maximal dépasse la valeur 26. En effet, le coût pour se déplacer jusqu'au centre de l'espace est élevé. Le ratio pour ces pixels atteint d'ailleurs les -20. Dans le cas d'un écoulement, nous pourrions aisément supposer que l'eau aurait plus de difficulté à atteindre les pixels au centre de l'obstacle que ceux à l'extérieur.

L'utilisation des algorithmes de « distance-coût » semble pertinente, du moins pour représenter des flux. Il peut aussi s'avérer utile pour la représentation des écoulements, mais en supposant que ces derniers soient à surface libre, sans turbulence, sans frottement, sans charge, etc.

## 6.2.2. Application au modèle

Pour analyser l'application de l'algorithme sur notre modèle, il est d'abord nécessaire de convertir la couche *Shapefile Donnees\_poly* comprenant les données en une image raster adaptée au logiciel IDRISI (voir Figure 24).

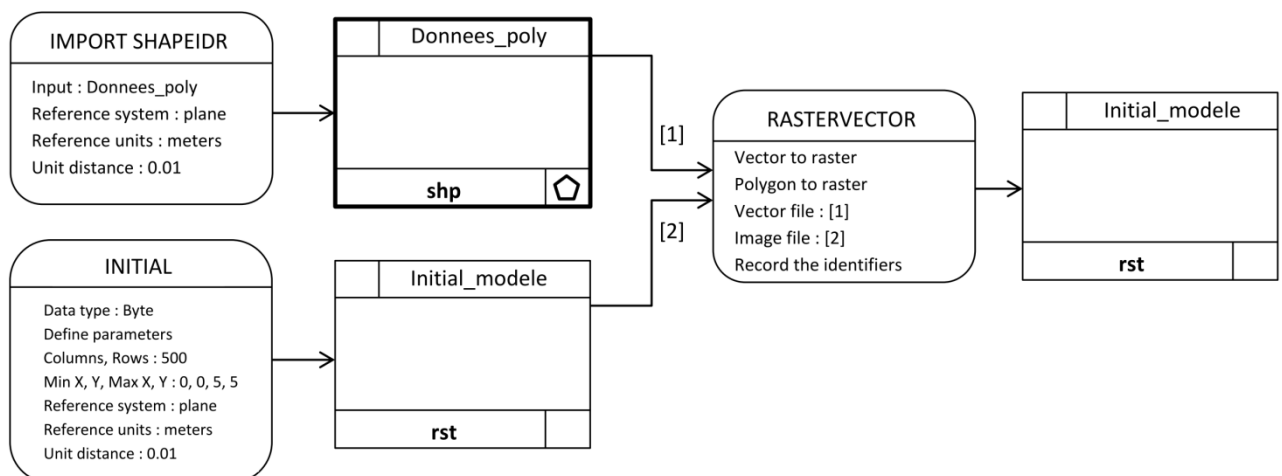


Figure 24 – Organigramme de la rasterisation de la couche *Donnees\_poly* sous IDRISI

Pour ce faire, il faut dans un premier temps importer la couche *Shapefile Donnees\_poly* à l'aide de l'outil d'importation spécifique IMPORT SHAPEIDR (voir Figure 25). Nous pouvons de suite constater que le système de référence choisi est plan. Comme expliqué précédemment, notre modèle ne mesurant que quelques mètres de côté, le système choisi importe peu. Toutefois, si les traitements sont appliqués à un modèle réel de plus grande envergure, il sera nécessaire de considérer le même référentiel (système géodésique et projection) d'un logiciel à l'autre.

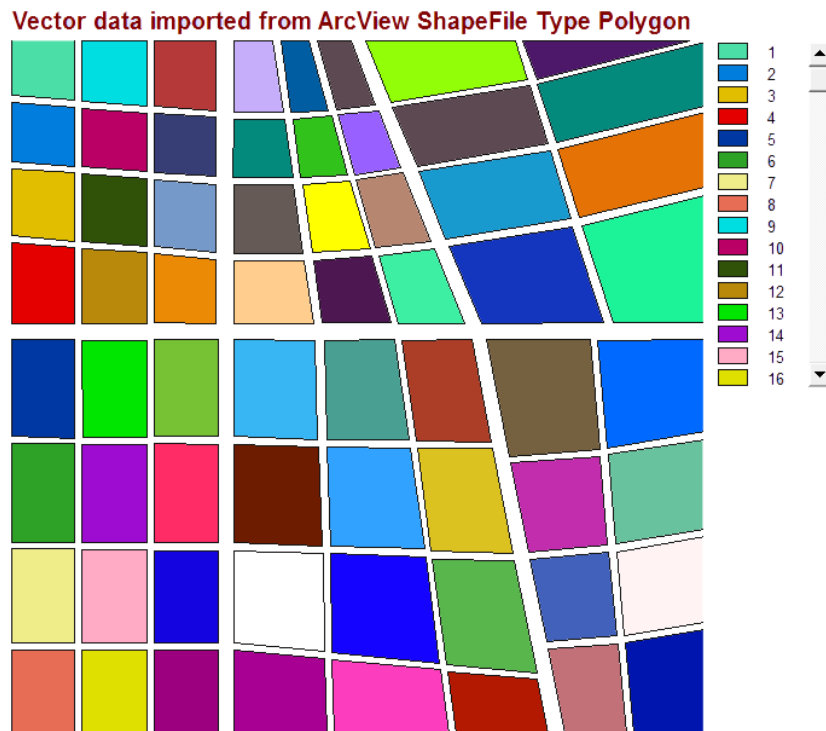


Figure 25 - Couche *Shapefile Donnees\_poly* sous IDRISI

Une fois la couche importée, elle doit être rastérisée. Cette opération n'est possible que si nous disposons d'une image raster avec les mêmes paramètres de référence. Une telle image est créée à l'aide de l'outil INITIAL. La valeur de fond n'a pas d'importance pour la suite.

La rastérisation s'effectue via l'option RASTERVECTOR, en précisant bien que nos données sont des polygones et qu'un identifiant doit être enregistré pour chacun. Il en ressort une image raster nommée *Initial\_modele* pour laquelle chacun des 64 polygones dispose d'une valeur servant d'identifiant, ainsi que la voirie à laquelle la valeur 0 est assignée par défaut (voir Figure 26).

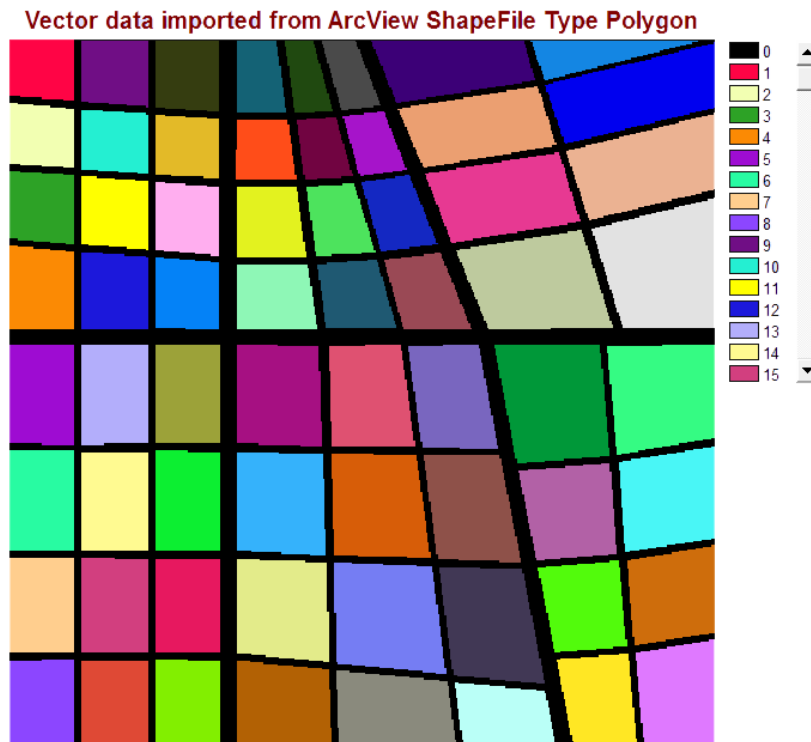


Figure 26 - Image raster *Initial\_modele*

Maintenant que nous disposons des données en mode raster, les traitements pour obtenir la surface de coûts peuvent être appliqués. L'organigramme suivant décrit la méthode à suivre (voir Figure 27).

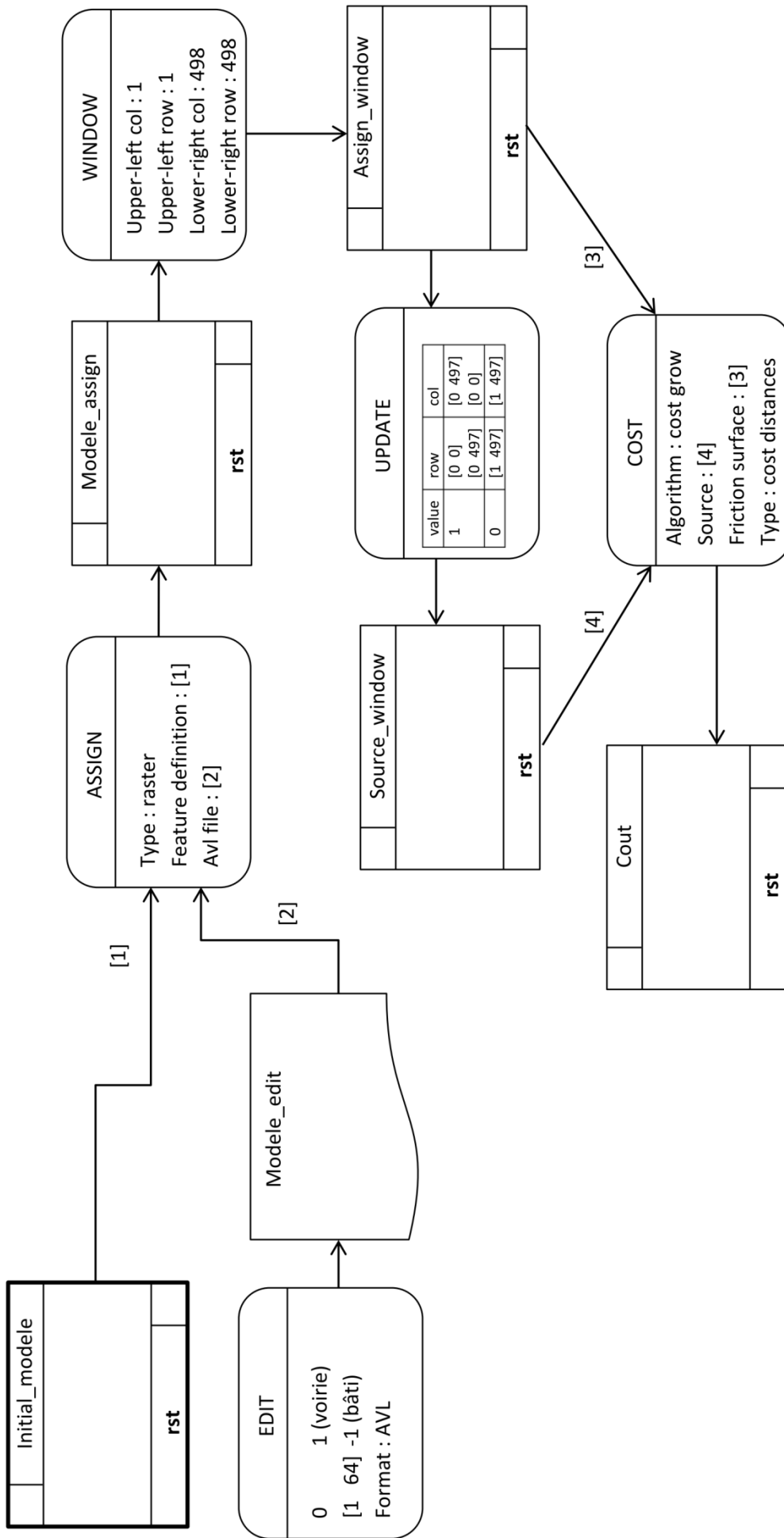


Figure 27 - Organigramme de l'extraction de la surface de coûts sous le logiciel IDRISI

- 1) ASSIGN: cet outil permet d'assigner des nouvelles valeurs aux valeurs actuelles de l'image. Un fichier au format AVL (*Attribute Values File*) a préalablement été construit grâce à la fonction EDIT. Dans ce document, les valeurs de bâti allant de 1 à 64 sont remplacées par la valeur -1, et la valeur de voirie 0 par 1. ASSIGN permet d'appliquer ces changements à l'image choisie, ici *Initial\_modele*. L'image en sortie est intitulée *Modele\_assign* (voir Figure 28). Grâce à ce traitement, l'image peut servir de surface de friction où les obstacles dont la valeur est -1 sont imperméables.

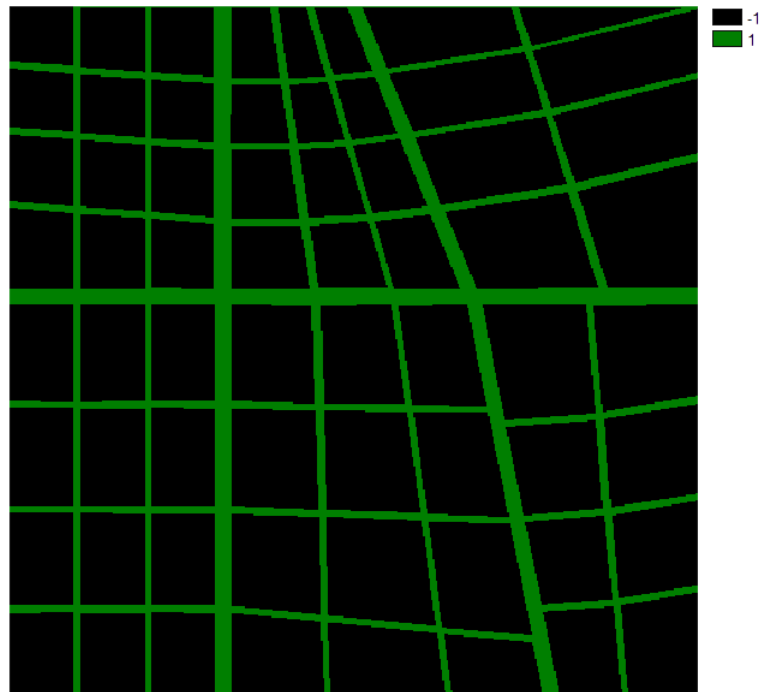


Figure 28 - Image raster *Modele\_assign*

- 2) WINDOW: il est nécessaire de recouper notre image pour éviter les effets de bord. En effet, les obstacles ne rejoignent pas les extrémités de notre image sur tout le contour de celle-ci. Les sommets de certains obstacles extérieurs ne sont pas exactement situés sur les bords à 0 et 5 m mais bien à 0,01 ou 4,99 m. Puisque la voirie est construite comme étant le reste de l'image une fois les blocs de bâtis mis à part, elle contient aussi les mailles situées entre les bords de l'image et les blocs les plus externes. Une fine voirie est donc visible aux bordures de l'image. En réalité, ces mailles ne devraient pas être utilisées dans l'algorithme de coût. Leur présence risque même d'engendrer des effets de bord. Effectivement, le flux théorique va aussi se propager sur les mailles en bordure (voir Annexe 4). La largeur d'une maille, soit 1 cm, est retirée tout autour de l'image. Ce recadrage fournit en sortie l'image *Assign\_window* (voir Figure 29).

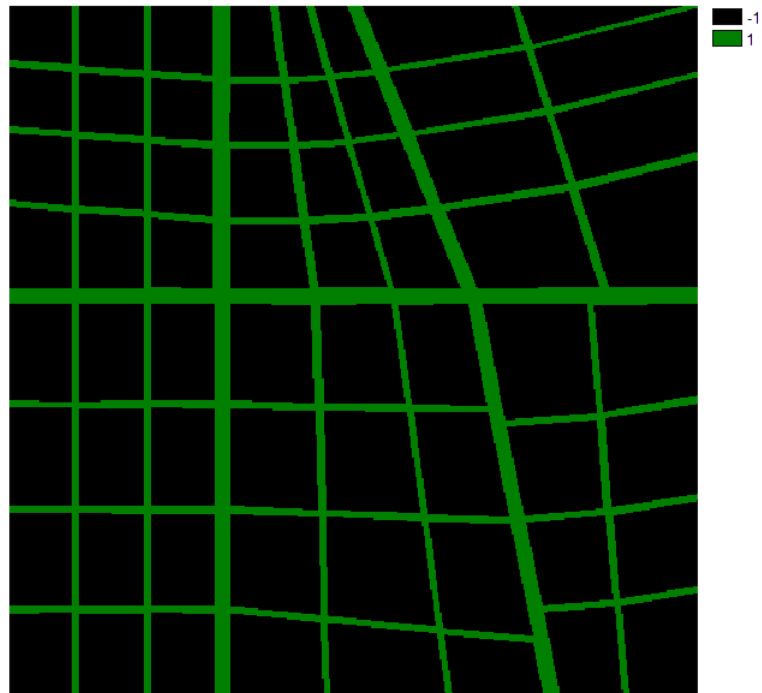


Figure 29 – Image raster *Assign\_window*

- 3) UPDATE: pour créer l'image qui fera office de source, partons de l'image recadrée *Assign\_window* à laquelle nous assignons de nouvelles valeurs. De cette façon, l'image en sortie aura exactement les mêmes propriétés que l'image *Assign\_window*. Pour changer complètement les valeurs, l'option UPDATE est utilisée. Celle-ci permet de choisir les pixels à changer grâce à la numérotation des lignes et des colonnes, et quelle valeur leur attribuer. La valeur 0 est assignée à tous les pixels de l'image sauf ceux de la première colonne et ceux de la première ligne qui, eux, se voient attribuer la valeur 1. Pour rappel, les pixels de valeur 1 sont ceux à partir desquels l'algorithme va se propager. Nous avons choisi ces pixels comme source car, dans le travail de Debaucheron (2014), il injecte des débits aux 14 rues situées sur les côtés Nord et Ouest du modèle pour simuler un écoulement. Nous avons donc opté pour la même approche. L'image raster qui résulte de cette opération est nommée *Source\_window* (voir Figure 30 – Image raster *Source\_window*).



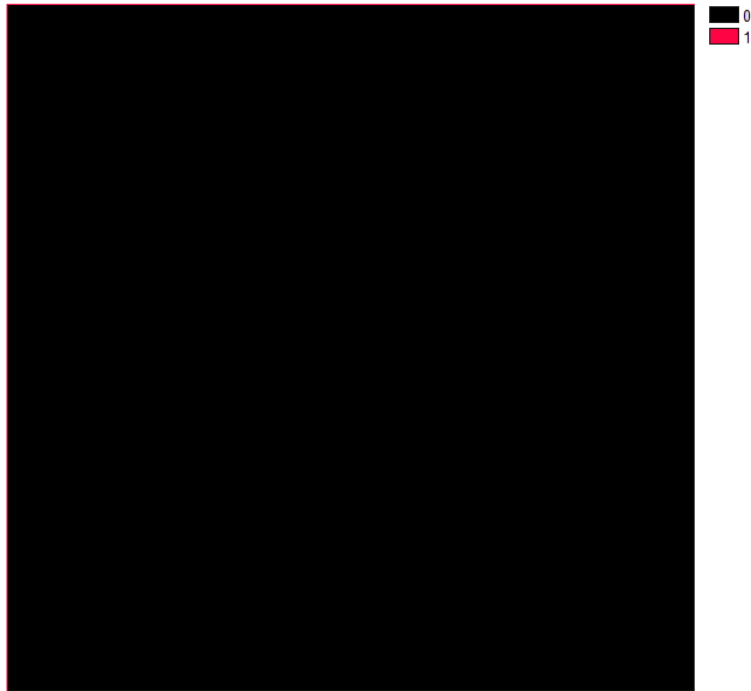
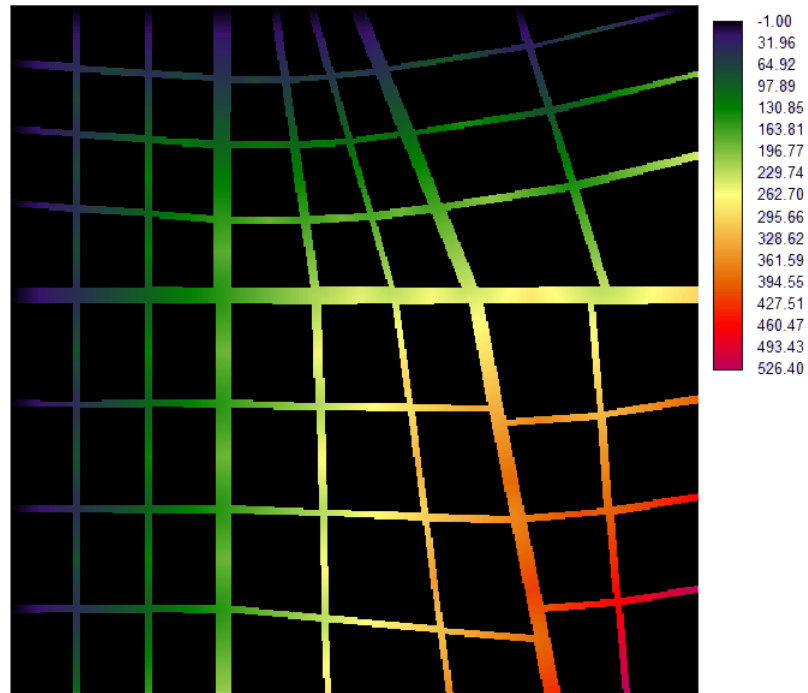


Figure 30 - Image raster *Source\_window*

- 4) COST: enfin, l'algorithme de calcul de la surface de coûts peut être lancé. En entrée, il nécessite la surface de friction *Assign\_window* et l'image source *Source\_Window*. L'image *Cout* en sortie est la suivante (voir Figure 31).



**Figure 31 – Image raster *Cout***

Nous observons sur cette image que la zone la plus difficile à atteindre pour le flux est le coin au Sud-Est du modèle. Ce résultat était pressenti puisqu’il s’agit de la zone à l’opposé des pixels sources. Dans le cas d’un écoulement, les carrefours sont des zones de « rencontre » des écoulements propagés depuis l’amont des rues au Nord et à l’Ouest. Nous observons ici un phénomène similaire puisque, à chaque carrefour, l’algorithme choisit entre plusieurs valeurs cumulées depuis différentes sources. Le coût atteint la valeur 526 dans la partie Sud-Est, ce qui est évidemment au-delà de la valeur de coût atteinte sans surface de friction (voir Figure 32 – Couche raster *Cout\_sansfriction* Figure 32). Dans ce dernier cas, nous sommes en effet sous la barre des 497, c’est-à-dire la longueur d’un côté de l’image sans le pixel source. Cette différence des maxima s’explique par la présence des obstacles qui obligent le flux à les contourner.

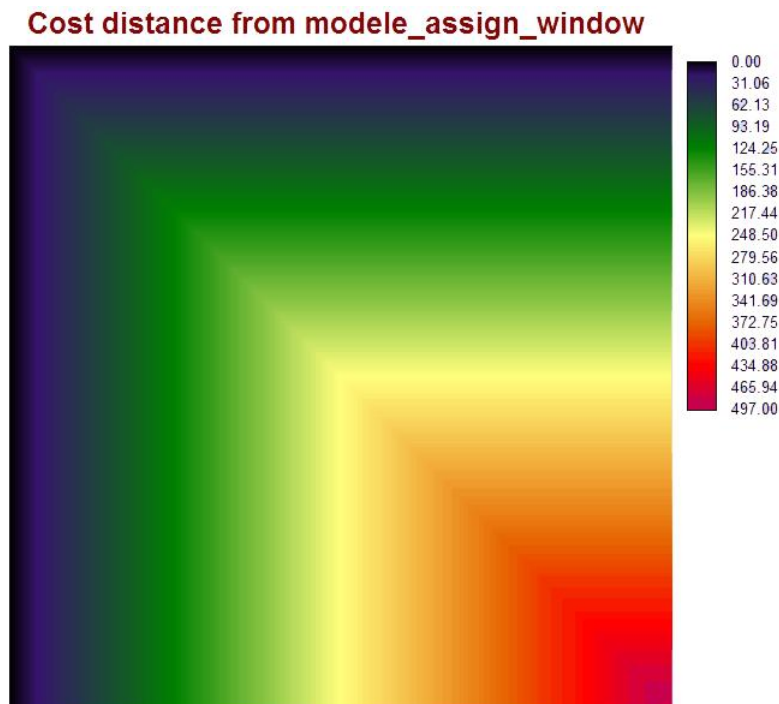


Figure 32 – Couche raster *Cout\_sansfriction*

L'algorithme de « distance-coût » semble être une bonne base pour représenter l'obstruction engendrée par les obstacles sur un écoulement. Le comportement d'une inondation est toutefois plus complexe qu'un simple flux au travers des obstacles de bâti.

### 6.2.3. Application au modèle avec des valeurs de porosité

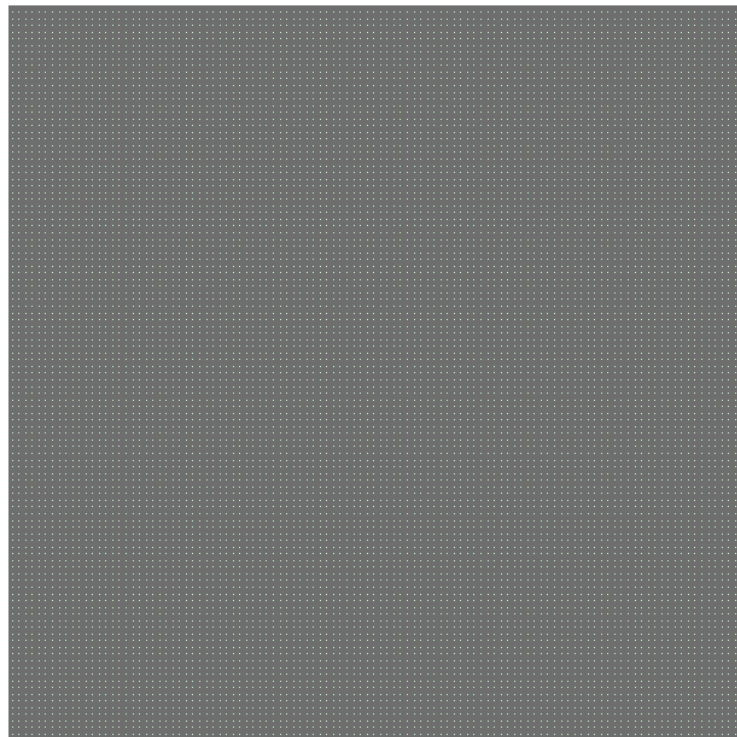
Il est maintenant envisageable de lancer les traitements en tenant compte de la porosité. C'est la couche *Shapefile Porosite* qui est utilisée en entrée. L'opération est cependant difficile à mettre en œuvre sur IDRISI car la surface de friction ne se contentera pas des obstacles à -1 et des voiries à 1. Les pixels composés à la fois d'une partie voirie et d'une partie obstacle doivent se voir attribuer une valeur différente. Ces pixels sont censés être plus difficile à traverser qu'un pixel entièrement en voirie puisque le flux ne peut passer que sur une partie de celui-ci. Pour représenter cette difficulté, il faudrait assigner aux pixels concernés des valeurs supérieures à 1. Mais quelle valeur leur donner? Il est difficile de concevoir une réponse à cette question dans la mesure où la valeur -1 représente l'imperméabilité. Il n'y a pas de valeur de friction maximale et donc pas d'intervalle défini où nous pourrions y calculer des valeurs de friction représentatives de la porosité.

L'utilisation d'un autre logiciel, ArcGIS, pourrait être la solution au problème. ArcGIS permet de rastériser des polygones en conservant un champ de valeurs. En

choisissant comme champ la porosité, nous pourrions en disposer pour créer une image avec des valeurs de friction.

La méthode à suivre pour obtenir une surface de coûts sous ArcGIS à partir de notre modèle est décrite ci-après.

D'abord, il faut y ouvrir la couche *Shapefile Porosite* (voir Figure 33). Il est conseillé de faire une copie de cette couche car elle sera utilisée à deux reprises, or certains traitements portant sur la table attributaire ne créent pas une nouvelle couche en sortie et enregistrent directement sur la couche de départ.



**Figure 33 - Couche *Shapefile Porosite* sous ArcGIS**

Ensuite, nous allons présenter les traitements à effectuer pour créer une surface de friction ou surface de coûts pour utiliser les mêmes termes que ceux utilisés dans le logiciel ArcGIS<sup>2</sup>. Les traitements à appliquer pour créer la surface de coûts s'organisent comme suit (voir Figure 34).

---

<sup>2</sup> Il faut être attentif au fait que la surface de friction sous IDRISI est analogue à la surface de coûts sous ArcGIS, et que la surface de coûts (ou de distances) chez le premier correspond en fait à la surface de distance coût dans le second ! Il faut donc bien différencier la surface de coûts du logiciel IDRISI qui correspond au résultat de l'algorithme de « distance-coût » de celle d'ArcGIS qui correspond à une surface de friction.

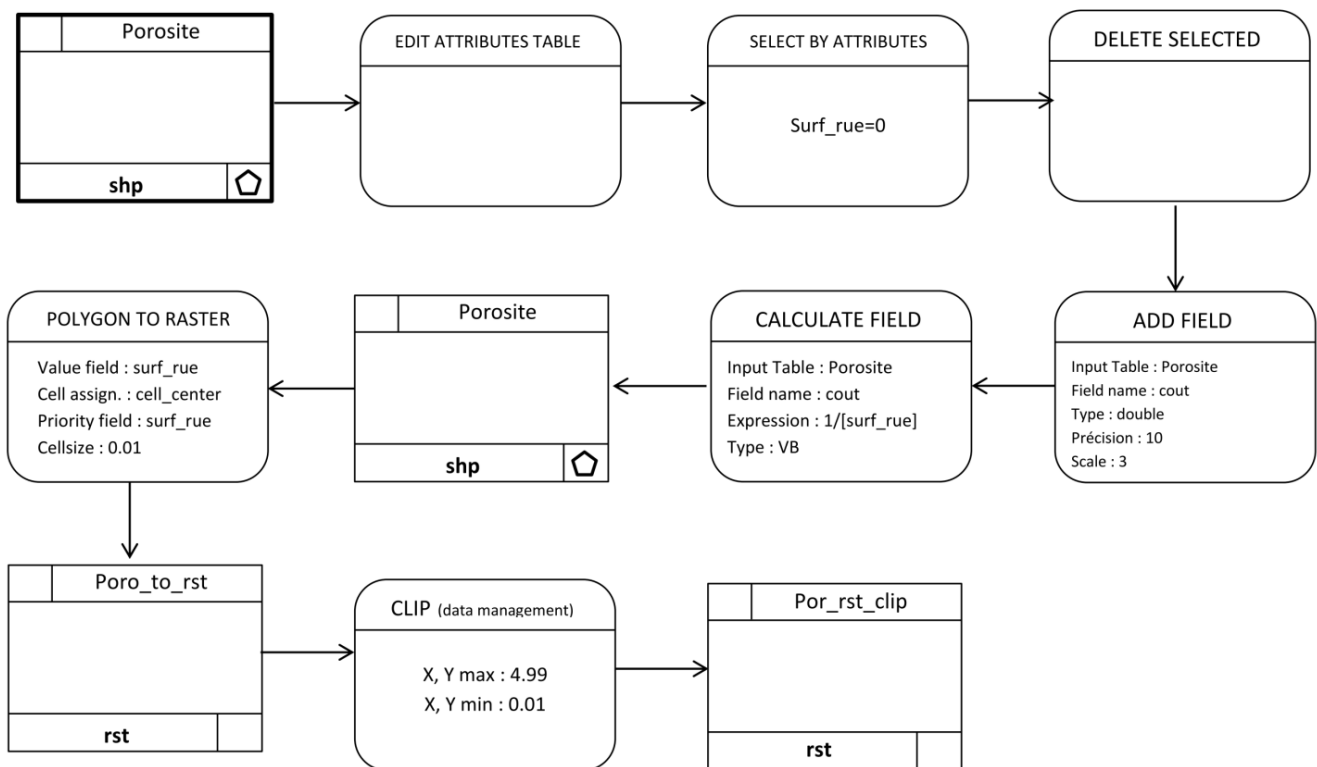


Figure 34 – Organigramme d'obtention d'une surface de coûts sous ArcGIS

- 1) EDIT ATTRIBUTES TABLE : la table d'attributs de *Porosite* est basculée en mode édition.
- 2) SELECT BY ATTRIBUTES : les obstacles vont être sélectionnés grâce à la formule  $surf\_ilot = 1$ .
- 3) DELETE SELECTED : les mailles sélectionnées sont supprimées. En supprimant ces polygones, nous délimitons ainsi la zone où l'algorithme pourra être appliqué. Il ne nous reste plus que les voiries et la bordure superficielle déjà mentionnée lors de l'extraction des rues sous IDRISI (voir Figure 35).

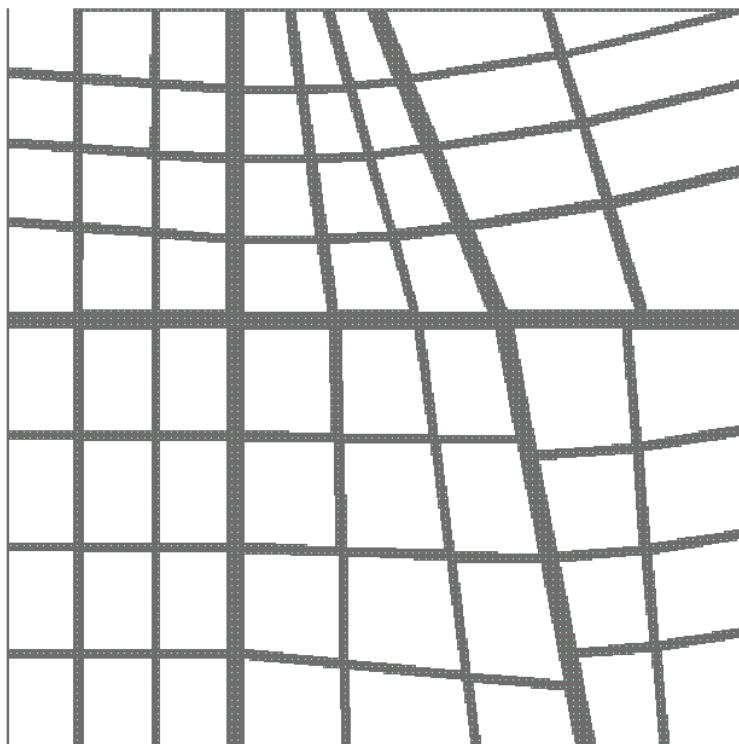


Figure 35 - Couche *Shapefile Porosite* après suppression des obstacles où *surf\_ilot=1*

- 4) ADD FIELD : il faut bien évidemment calculer les valeurs de coût. Pour le moment, nous disposons d'une porosité allant de 0 (pour les obstacles) à 1 (pour les voiries). Ces valeurs de porosité ne peuvent pas être utilisées comme telles pour former une surface de coûts. En effet, une valeur de porosité de 0,5 par exemple signifie qu'une seule moitié du pixel est de type voirie. Normalement, le coût sur ce pixel doit donc être supérieur au coût sur un pixel dénué d'obstacle. Pour commencer, il faut créer un nouveau champ attributaire. Ce que la fonction ADD FIELD nous permet de faire. Le nouveau champ attributaire est nommé *cout*, et est constitué de valeur double avec une longueur de 10 chiffres, dont 3 décimales possibles. Ces paramètres correspondent à ceux utilisés lors du calcul de la porosité. Il est inutile d'avoir une précision supplémentaire à celle de la porosité qui sera utilisée dans le calcul des valeurs de coût. Les autres paramètres de l'algorithme sont optionnels.
- 5) CALCULATE FIELD : pour obtenir des valeurs de coût valables à partir de la porosité, nous avons utilisé le rapport suivant :  $cout=1/[surf\_rue]$  (voir Figure 36). L'expression peut être exprimée en VB (*Visual Basics*) comme ici, ou en Python.

polygone	surf_ilot	surf_rue	cout
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.99	0.01	100
A8	0.398136	0.602	1.661
	0	1	1
	0	1	1
	0	1	1
	0	1	1
A7	0.495	0.505	1.98
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100
A7	0.99	0.01	100

Figure 36 - Aperçu de la table d'attributs de la couche *Shapefile Porosite* après calcul du champ *cout*

6) POLYGON TO RASTER : comme son nom l’indique, cet outil va permettre de rasteriser l’information. La couche *Porosite* est ainsi transformée en une couche raster nommée *Poro\_to\_rst* (voir Figure 37). Bien évidemment, la rasterisation se fait en conservant comme champ de valeur l’attribut *cout*. L’assignement des polygones aux cellules se fait ici via le centre de celles-ci. En fait, si le futur pixel contient plusieurs polygones, celui qui passera sur son centre sera celui dont la valeur est conservée<sup>3</sup>. Dans notre cas, les autres options devraient toutefois fournir les mêmes résultats puisque les polygones sont des mailles de même dimension que les pixels, à savoir 0,01 m. Sur l’image *Poro\_to\_rst*, les pixels dont *cout* vaut 1 sont noirs, et le ton de gris tend vers le blanc de façon proportionnelle lorsque la valeur de *cout* croît. Les pixels entièrement couverts par un obstacle et ayant été supprimés à l’étape précédente ont comme valeur *NoData*.

<sup>3</sup> Les informations relatives aux différents outils du logiciel ArcGIS sont issues de l’aide du logiciel ArcGIS 10.3.1.

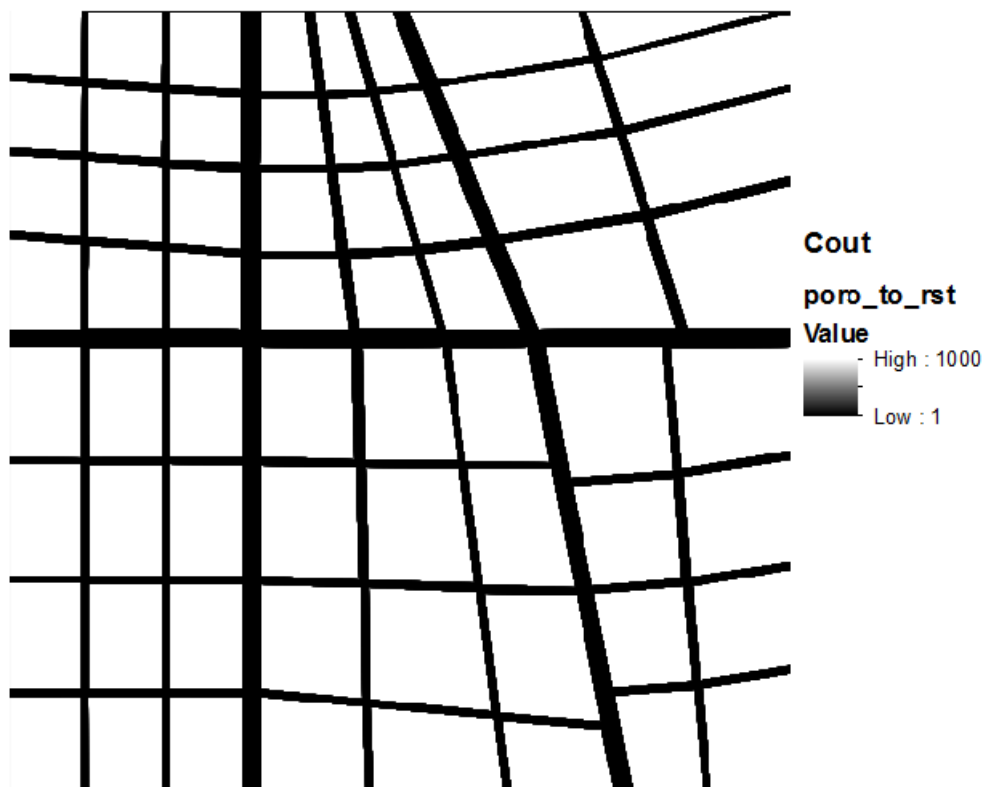


Figure 37 - Image raster *Poro\_to\_rst*

- 7) CLIP (*data management*) : à nouveau, l'image des voiries obtenue ici en supprimant les pixels de bâti présente des bords qui, en réalité, ne représentent pas des rues du prototype mis en place par Araud (2012). Pour éviter des effets non désirés, il est nécessaire de rogner ces bords. L'outil CLIP nous permet de rogner 1 cm de chaque côté du modèle. L'option « *Maintain Clipping Extent* » n'est pas sélectionnée puisque nous ne voulons pas ajuster le nombre de colonnes et de lignes ni ré-échantillonner les pixels, mais simplement ajuster l'extension en conservant l'alignement des pixels. L'image en sortie est intitulée *Por\_rst\_clip* (voir Figure 38). Elle correspond à la surface de coûts recherchée.



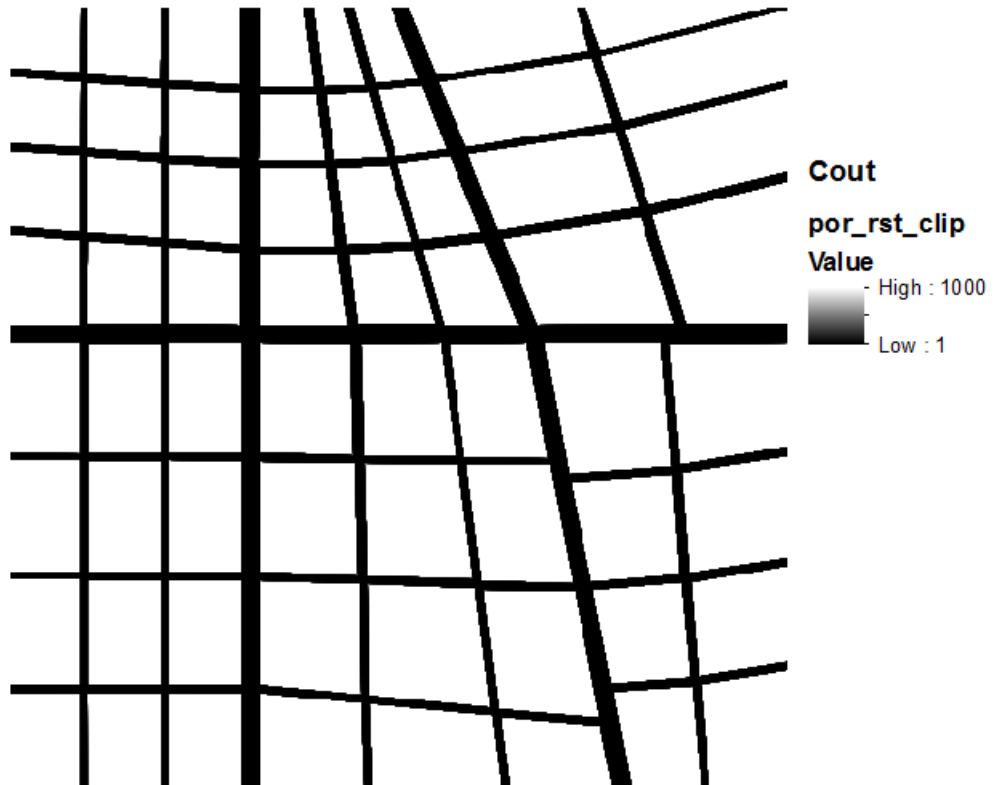


Figure 38 - Image raster *Por\_rst\_clip*

Il faut ensuite créer l'image source de propagation (voir Figure 39). Celle-ci est également obtenue à partir de la couche *Shapefile Porosite*.

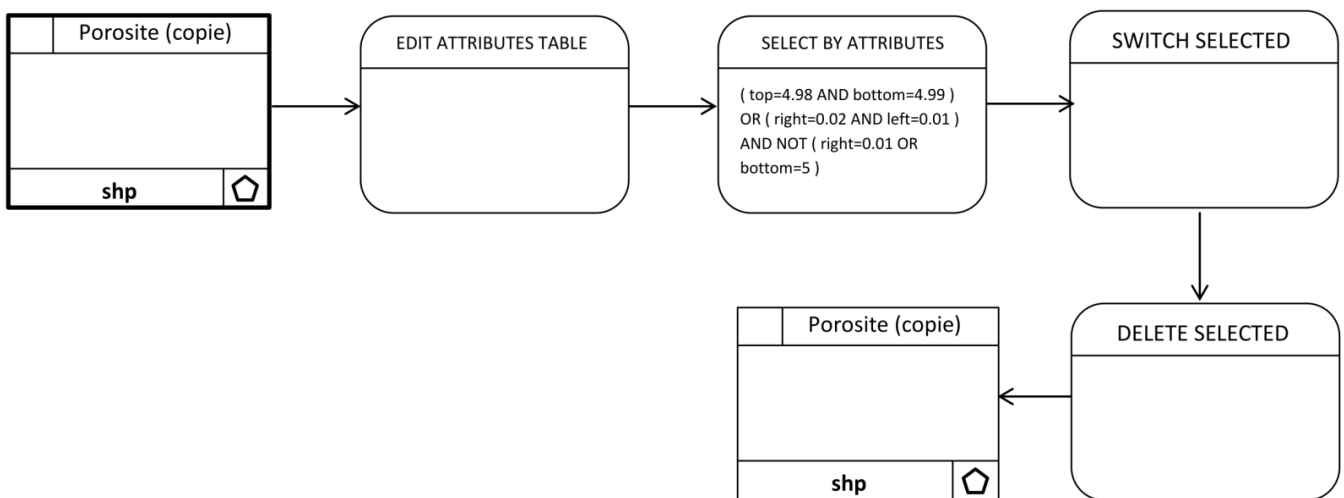
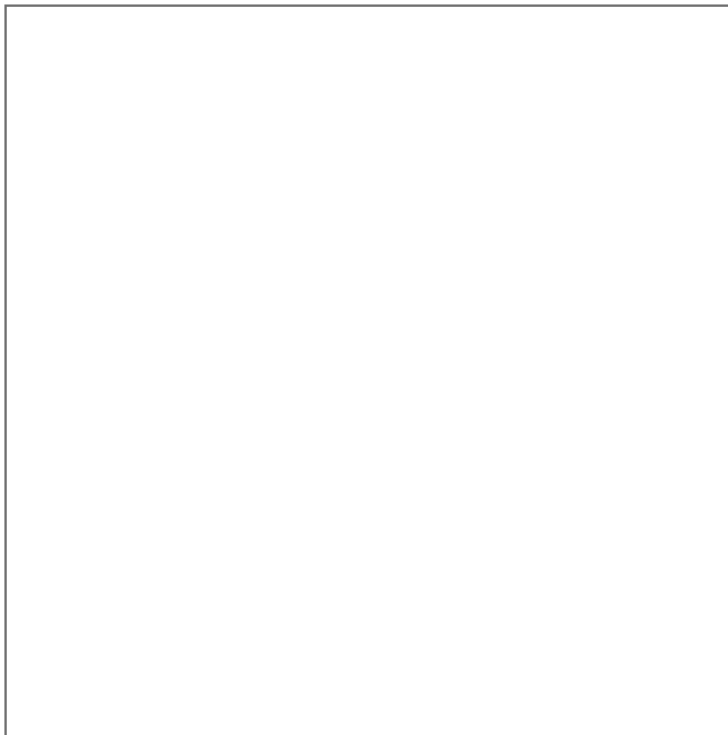


Figure 39 - Organigramme d'obtention d'une image source sous ArcGIS

- 8) EDIT ATTRIBUTES TABLE : à nouveau, la couche *Porosite* (une copie) est passée en mode édition.
- 9) SELECT BY ATTRIBUTES : les pixels qui vont servir de source de propagation sont sélectionnés grâce à la formule suivante :

```
SELECT * FROM Porosite WHERE (top = 4.98 AND bottom = 4.99)  
OR (right = 0.02 AND left = 0.01)  
AND NOT ("right" = 0.01 OR "bottom" = 5)
```

- 10) SWITCH SELECT : cette opération permet d'inverser la sélection. Notre sélection comprend maintenant toutes les entités qui ne feront pas office de source dans notre algorithme de coût.
- 11) DELETE SELECTED : ces entités sont supprimées, de cette façon il ne nous reste dans notre image que les pixels sources (voir Figure 40Figure ).



**Figure 40 - Couche *Shapefile* Porosite contenant uniquement les pixels sources**

Finalement, nous pouvons lancer l'algorithme de « distance-coût » nommé COST DISTANCE (voir Figure 41Figure ).

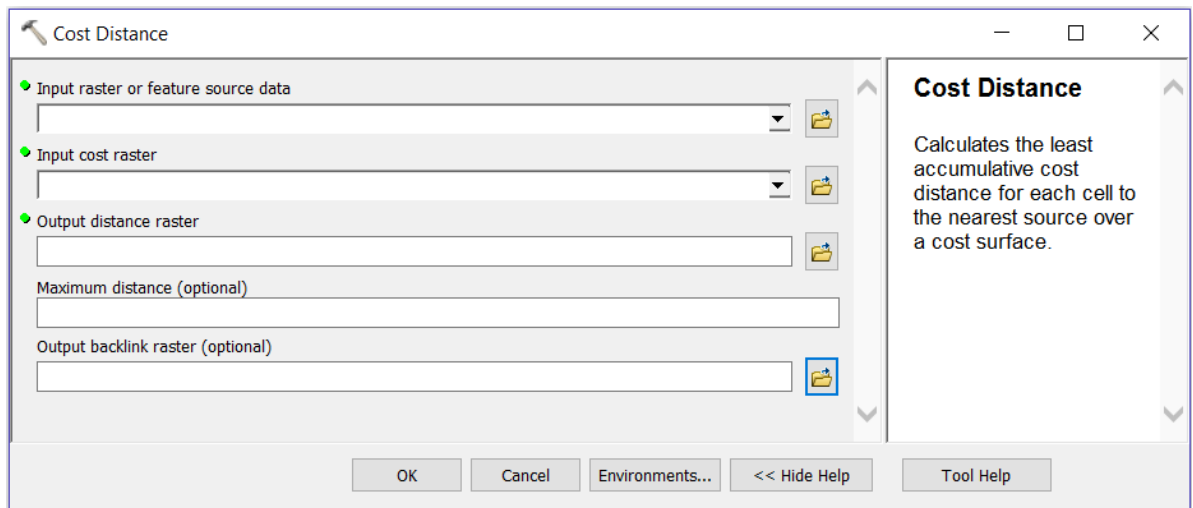


Figure 41 - Interface de l'outil COST DISTANCE sous ArcGIS

12) COST DISTANCE : le *Shapefile Porosite* (copie) est inséré en source de l'algorithme (voir Figure 42). L'image raster *Por\_rst\_clip* sert de raster de coûts. Il correspond à une surface de friction.

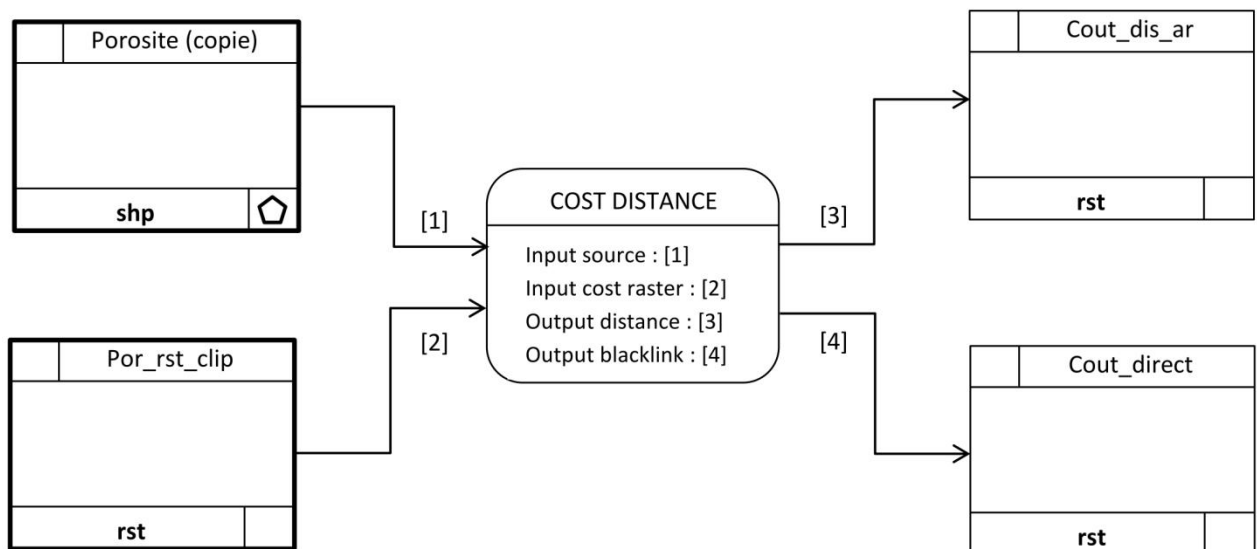


Figure 42 - Organigramme de l'algorithme COST DISTANCE sous ArcGIS

En sortie, nous obtenons la surface de distance *Cout\_dis\_ar* (voir Figure 44) ainsi que le raster *backlink Cout\_direct* (voir Figure 43). Ce dernier correspond à la direction empruntée par rapport à huit directions vers la prochaine cellule voisine. Cette information permet de reconstruire les chemins depuis n'importe quel pixel vers le pixel-source le plus proche de manière récursive.

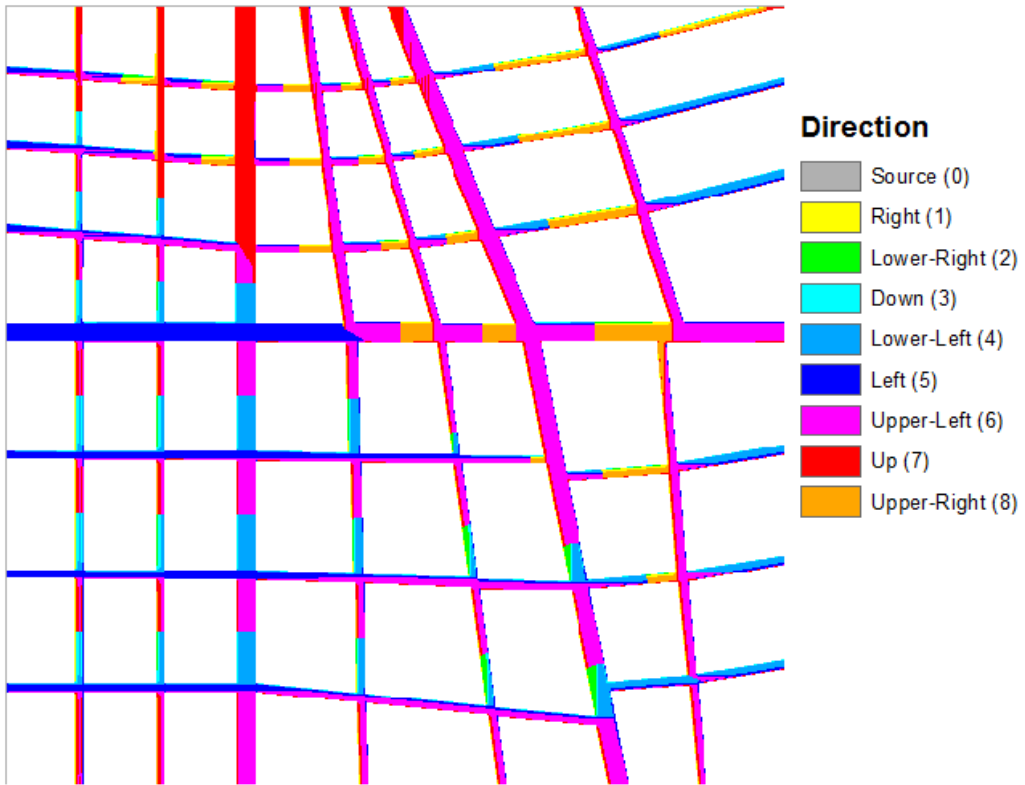


Figure 43 - Image raster *Cout\_direct*

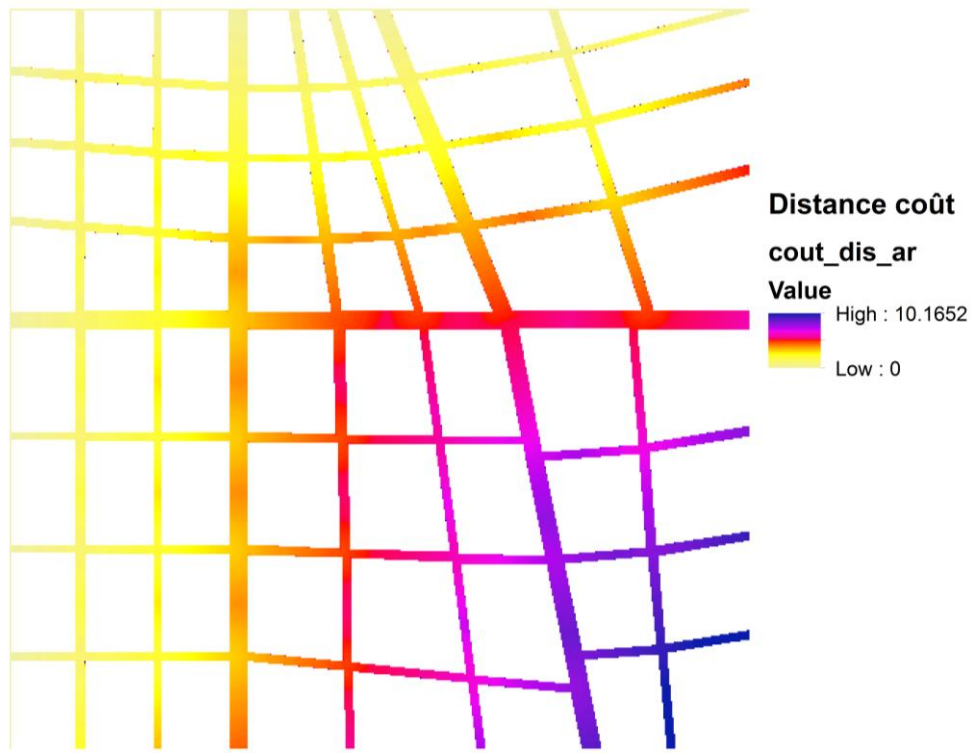


Figure 44 - Image raster *Cout\_dis\_ar*

Sur l'image des directions, nous constatons d'une part que le flux provient du Nord pour les rues verticales et de l'Ouest pour les rues horizontales, ce qui correspond aux origines des pixels sources. D'autre part, les tronçons de rues situés entre deux carrefours présentent souvent une séparation en leur milieu. Dans cette zone, deux flux opposés se rencontrent. Pour les rues qui s'interceptent à angle droit, la rencontre a toujours lieu exactement à leur centre, malgré les largeurs de rues parfois différentes. Pour les rues obliques, la séparation n'est pas au centre et, à nouveau, la différence de largeur des rues ne semble pas avoir d'influence. A priori, la position de la séparation dépend de l'inclinaison des rues. Cela peut s'expliquer par le fait que l'obliquité rallonge le trajet entre la source et le carrefour. Plus la rue est oblique, et plus l'algorithme est susceptible de prendre des directions diagonales et non droites. Or, les diagonales sont plus longues.

En ce qui concerne les valeurs de « distance-coût » calculées, elles vont à priori de 0 à 10. Contrairement à celui d'IDRISI, l'algorithme d'ArcGIS prend en compte la résolution des pixels qu'il multiplie au coût. Pour ne pas la considérer et avoir le même ordre de grandeur d'IDRISI, il suffit de multiplier les valeurs de distance résultantes par un facteur 100.

Directement, nous nous rendons compte que les valeurs montent à 1000 ce qui est à peu de choses près le double des valeurs obtenues avec IDRISI. En réalité, il y a des valeurs aberrantes dues aux indices de coût des pixels qui sont en partie obstacle et en partie voirie. En effet, ces pixels ont un coût élevé, qui augmente rapidement lorsque la proportion de voirie diminue. Certains coûts montent à 1000 lorsque la part de voirie est minimale, c'est-à-dire 0,001 étant donné la précision de cet attribut. Ce sont ces pixels qui atteignent une distance finale de plus de 1000. Ils sont facilement repérables visuellement grâce à l'échelle quantitative de couleurs (voir Figure 45). Ils se présentent comme des discontinuités dans le phénomène de flux.

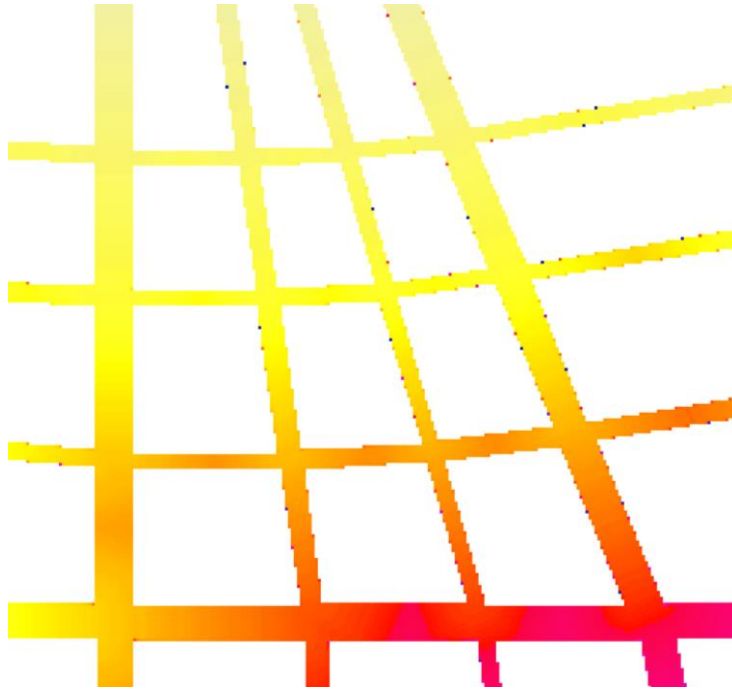


Figure 45 - Zoom sur l'image raster *Cout\_dis\_ar*

#### 6.2.4. Discussion

Si nous faisons abstraction des valeurs aberrantes, la valeur de « distance-coût » maximale atteinte sous ArcGIS est de 526,63 en considérant la porosité, contre 526,40 sous IDRISI sans la porosité. La différence entre les deux est faible. En fait, l'insertion des données de porosité dans le modèle réalisé sous ArcGIS a eu peu d'effet. A vrai dire, puisque les coûts des pixels contenant une partie de bâti sont supérieurs à 1, l'algorithme passera dans la mesure du possible par des pixels de type voirie. Or, comme les pixels de coûts élevés sont uniquement en bordure des voiries, il n'est pas nécessaire que l'algorithme les utilise pour traverser le modèle de part en part. La seule situation où le flux serait susceptible de se propager dans un pixel mixte plutôt que dans un pixel voirie serait lorsque le choix de voisinage se fait entre un mouvement droit passant par le pixel mixte et un mouvement en diagonal passant par le pixel voirie. Dans ce cas, il est possible que le coût de la traversée du pixel mixte soit inférieur à celle du pixel voirie.

L'analyse de l'image raster des directions entre pixels fournies par ArcGIS montrait que l'obliquité des rues du modèle jouait un plus grand rôle dans le choix de la direction que la largeur de ces rues. Par ailleurs, la variation entre les propagations avec et sans la porosité est presque négligeable à l'échelle de notre modèle. De plus, la friction générée en bordure par les valeurs de porosité engendre des artéfacts. Ces observations tendent à faire penser que l'algorithme de « distance-coût » est finalement peu adapté à la représentation d'un écoulement, du moins pour notre modèle. En réalité, ces résultats sont peu significatifs compte tenu de l'utilisation de données qui ne sont pas réelles. Il est possible qu'il soit plus efficace en insérant des valeurs plus proches de la réalité.

Nous pourrions par exemple tenir compte de la largeur des rues (voir Section 6.4), utiliser des valeurs de frottement plus adaptées aux affectations, considérer la présence de réseaux d'égouttage, etc. De même, la propagation pourrait être plus représentative d'un écoulement une fois couplée à des variables hydrauliques telles que le débit, et à des principes physiques comme la conservation du débit. Enfin, il est peut-être possible de trouver une formule de calcul du coût à partir de la porosité plus adéquate, afin de ne pas avoir de valeurs atteignant le millier.

L'organigramme permettant la réalisation de l'algorithme « distance-coût » sous ArcGIS en tenant compte de la porosité a tout de même été mis sous forme de modèle grâce au modéleur *ModelBuilder* du logiciel (voir Figure 46Figure 6.44). L'outil permet de l'exporter en script Python (voir Annexe 5 – Propagation de l'algorithme de « distance-coût » avec la porosité sous ArcGIS). Un modèle similaire a également été conçu sous IDRISI à l'aide de son modéleur mais, cette fois, sans tenir compte de la porosité (voir Figure 47).

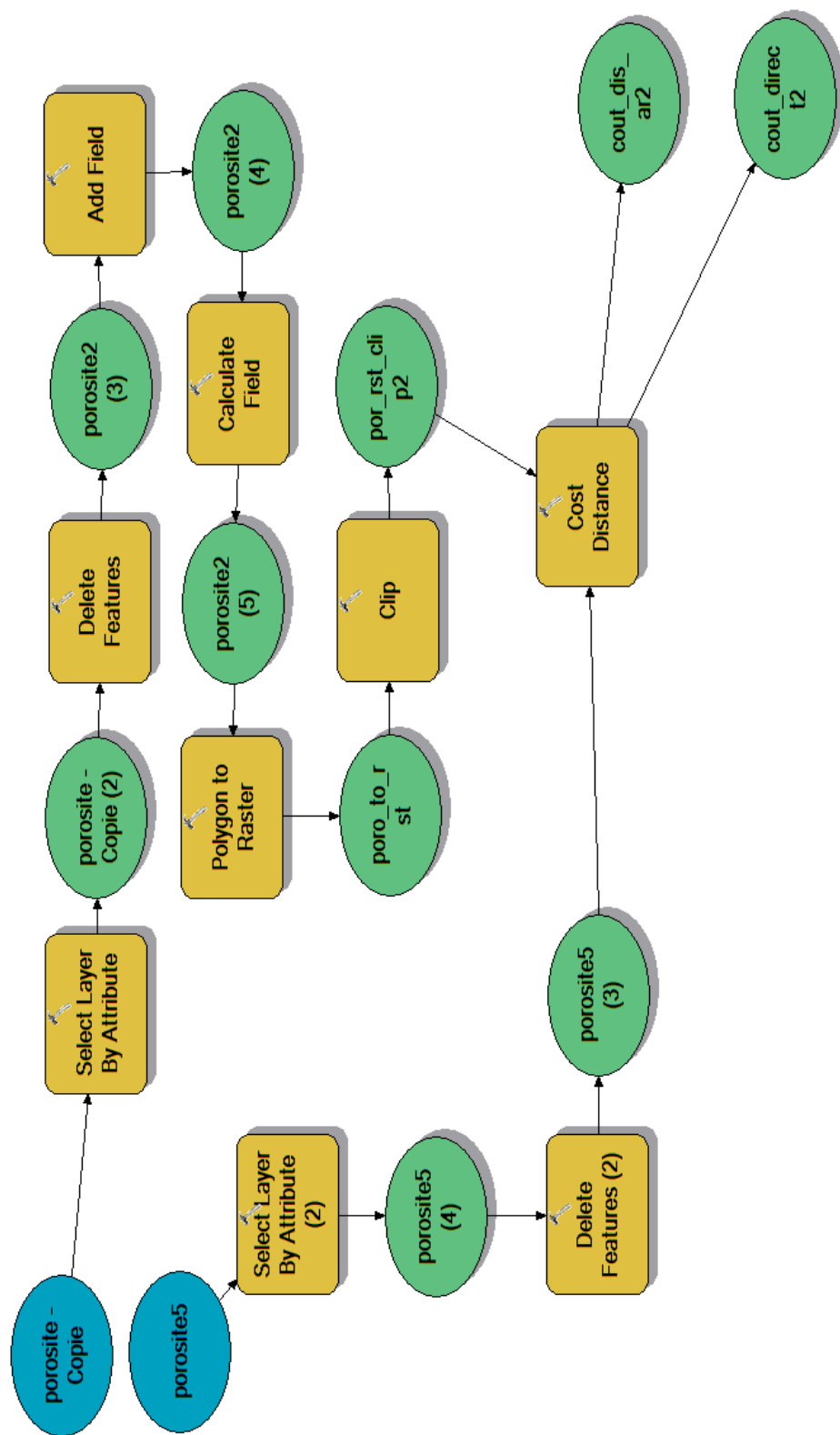


Figure 46- Modèle de propagation par « distance-coût » avec des valeurs de porosité sous ArcGIS



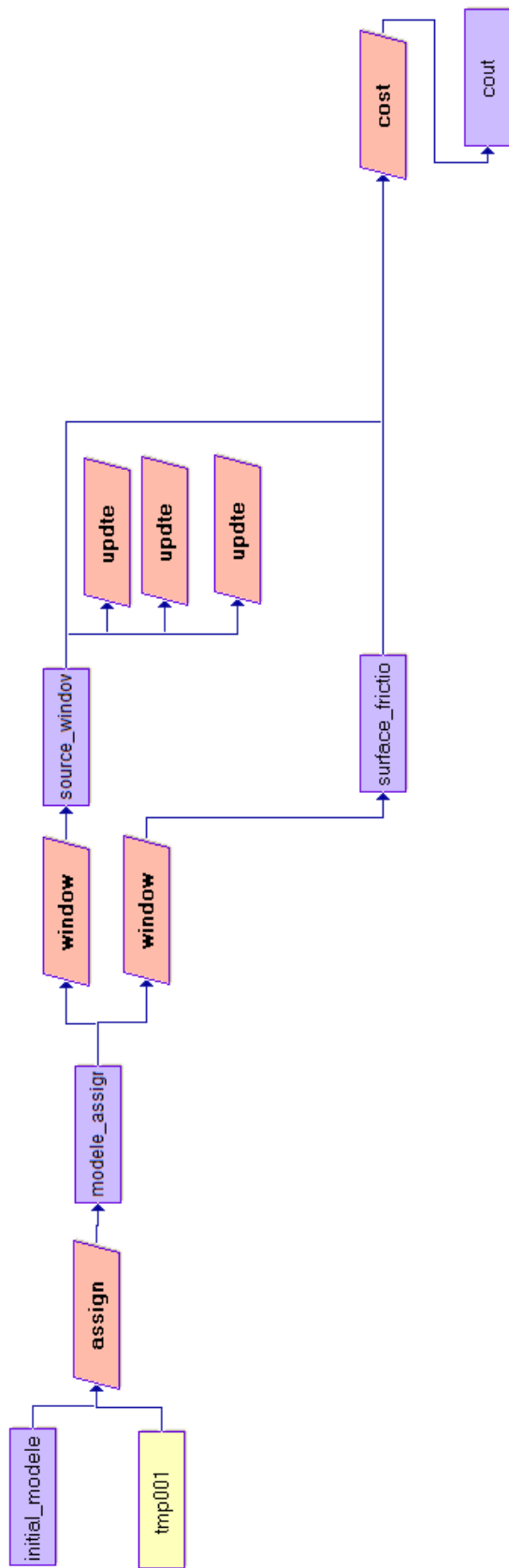


Figure 47- Modèle de propagation par « distance-coût » sous IDRISI

## 6.3. Adaptation à différentes résolutions

Un problème récurrent dans le traitement de données est le coût en termes de temps et de technologie qui peut s'avérer important. Pour y remédier, l'utilisation de données de plus faible résolution est parfois envisagée. La question est de savoir s'il est possible d'utiliser dans le modèle des mailles de plus grandes dimensions sans dégrader l'information issue des données. Pour rappel, les hydrauliciens intègrent leurs données dans des pixels de 1 cm de côté, ce qui correspond à la plus haute résolution disponible pour la grille de leur modèle.

Pour représenter la propagation à des résolutions différentes, nous avons dans un premier temps songé à agréger les pixels. Les manipulations ont été effectuées sous le logiciel IDRISI.

### 6.3.1. Par agrégation des pixels

La méthodologie pour agréger les pixels et former ainsi des mailles plus grandes est décrite ci-après. Les traitements sont appliqués directement sur l'image *Cost* (voir Figure 31 – Image raster *Cost*) correspondant à la surface de coûts générée sous IDRISI par propagation de l'algorithme de « distance-coût ». La méthode utilise un masque pour pallier les artéfacts dus à la prise en compte de pixels obstacles dans l'agrégation en bordure de voirie (voir Figure 48).

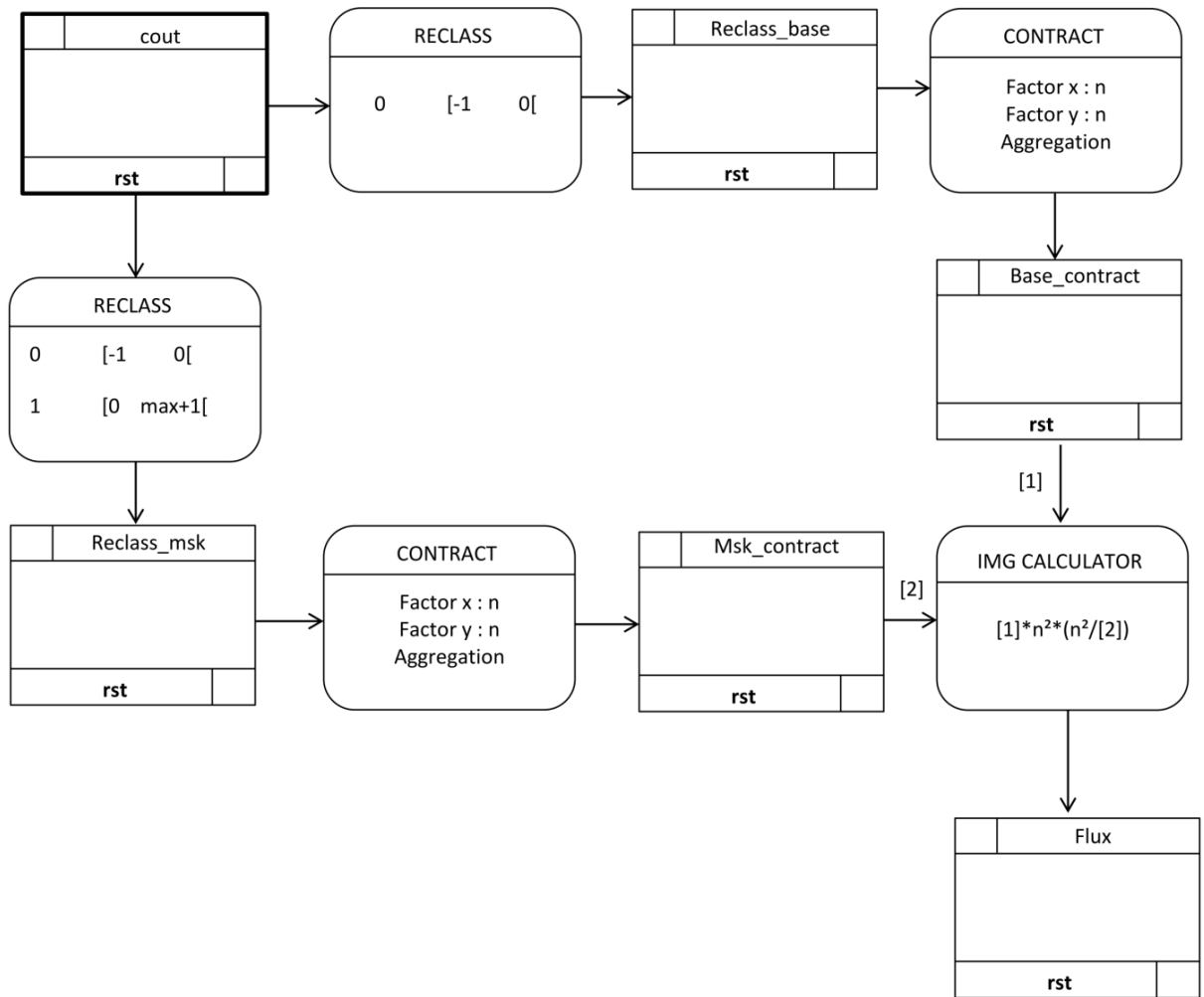


Figure 48 - Organigramme d'agrégation des pixels de coûts sous IDRISI

- 1) RECLASS : cette opération, appliquée à l'image *Cout*, permet de reclasser les valeurs voulues. Elle est utilisée à deux reprises.
- Dans un premier temps la valeur -1, correspondant aux obstacles, est remplacée par la valeur 0. La valeur -1 serait comptée comme une valeur de coût négative dans le calcul d'intégration. En vérité, il n'y a pas de coût sur les pixels obstacles, c'est pourquoi la valeur 0 leur est assignée. L'outil propose de transformer l'image résultante en entiers, mais cela ferait perdre en précision. L'image reclassée est appelée *Reclass\_base* (voir Figure 49).

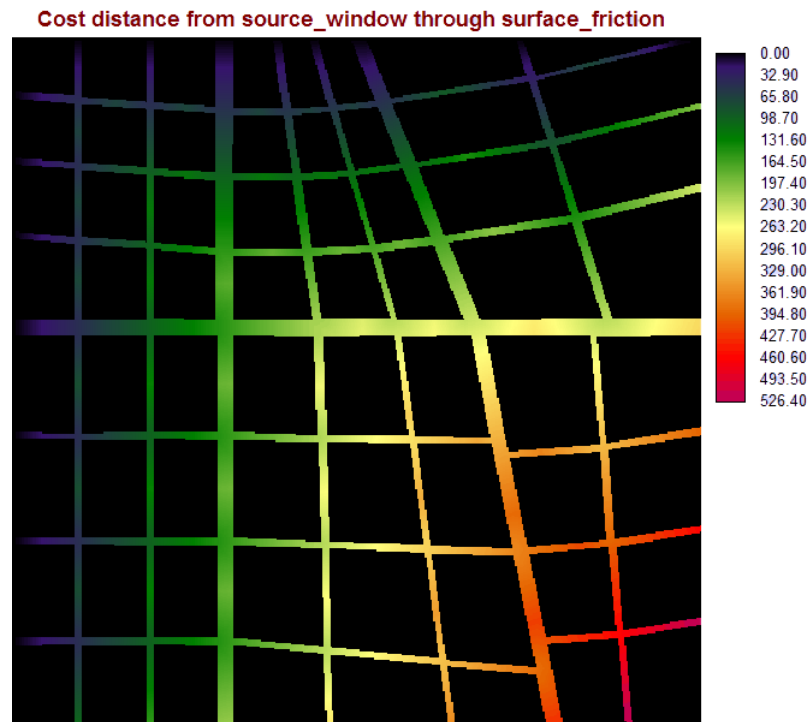


Figure 49 - Image raster *Reclass\_base*

L'image *Cout* est ensuite reclassée une seconde fois, toujours en remplaçant la valeur -1 par 0, mais en assignant en plus la valeur 1 aux pixels de coûts supérieurs à 0. Nous obtenons ainsi un masque, *Reclass\_msk* (voir Figure 50), avec une valeur 1 pour les pixels de voirie et 0 pour les pixels correspondant à des obstacles.

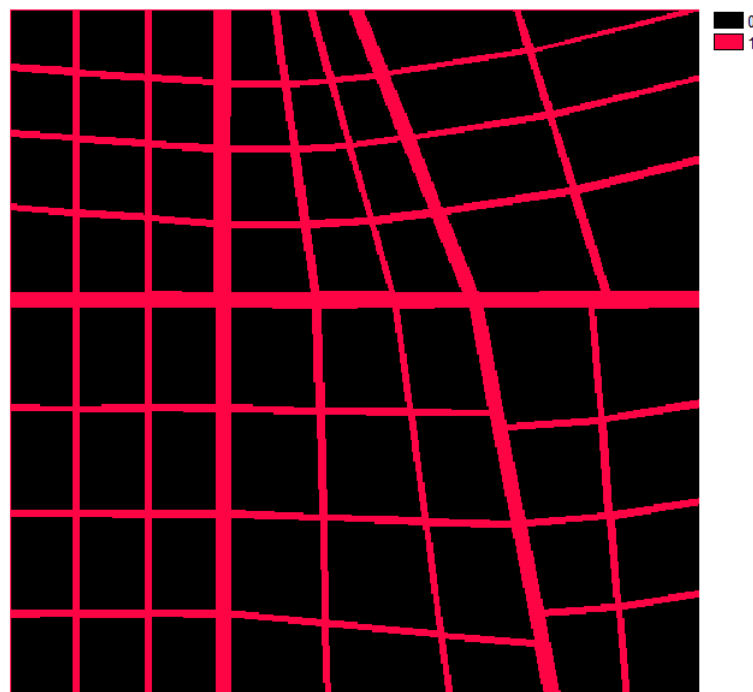


Figure 50 - Image raster *Reclass\_msk*

2) **CONTRACT** : cet outil réalise l'agrégation des pixels. Pour conserver le format carré des mailles aux autres résolutions, le facteur n doit être le même pour x et pour y. Pour l'exemple, nous avons choisi le facteur 2. De cette façon, 2 pixels en x et 2 pixels en y sont agrégés, soit 4 sous-pixels, pour former une nouvelle maille. Il en résulte que le nombre de colonnes et de lignes de l'image est divisé par 2. L'agrégation consiste à réaliser la moyenne des valeurs, à l'exception des valeurs *flag* définies dans les métadonnées. Tant qu'à présent, aucune valeur *flag* n'a été déterminée. L'opération est appliquée sur les deux images, *Reclass\_base* et *Reclass\_msk*. Les images en sortie sont respectivement *Base\_contract* (voir Figure 51) et *Msk\_contract* (voir Figure 52). Le premier reprend les moyennes des coûts et le second la part de sous-pixels de type voirie dans le nouveau maillage.

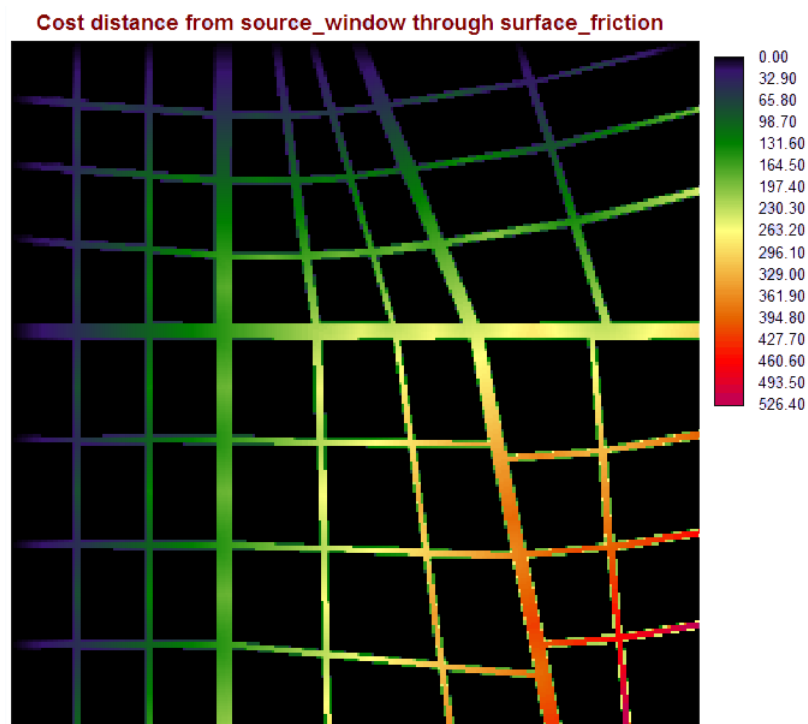


Figure 51 - Image raster *Base\_contract*

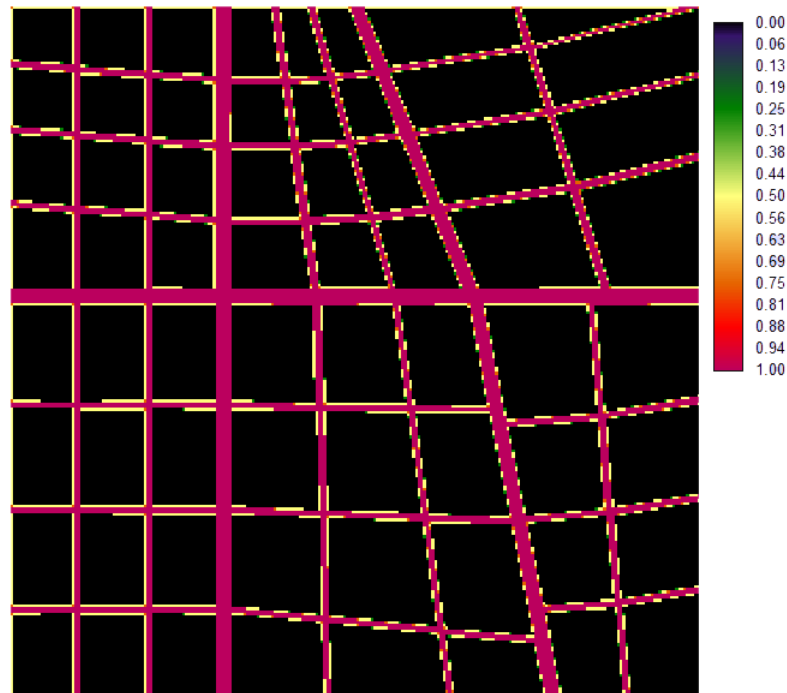


Figure 52 - Image raster *Msk\_contract*

- 3) IMAGE CALCULATOR : finalement, les pixels correspondants sur les deux images sont mis en relation tel que ;

$$Flux = Base\_contract * n^2 * [n^2 / (Msk\_contract * n^2)]$$

Dans notre exemple, le facteur  $n$  vaut toujours 2. *Base\_contract* est d'abord multipliée par  $n^2$  pour obtenir la somme des coûts des sous-pixels initiaux. Il en résulte le coût total de propagation dans la nouvelle maille. L'image est ensuite multipliée par un facteur correctif (situé entre crochet dans l'équation). Ce facteur est supérieur à 1 lorsqu'il y a des sous-pixels inefficaces dans la maille, et vaut 1 lorsque tous les sous-pixels sont de type voirie. Notons que *Msk\_contract* est multiplié par  $n^2$  pour obtenir le nombre de sous-pixels étant de type voirie dans chaque nouvelle maille, c'est-à-dire les pixels efficaces. L'image résultante est nommée *Flux* (voir Figure 53). La valeur maximale est quatre fois supérieure à celle de l'image *Cout*, ce qui est logique puisque le coût de propagation dans une maille correspond à la traversée de quatre sous-pixels.

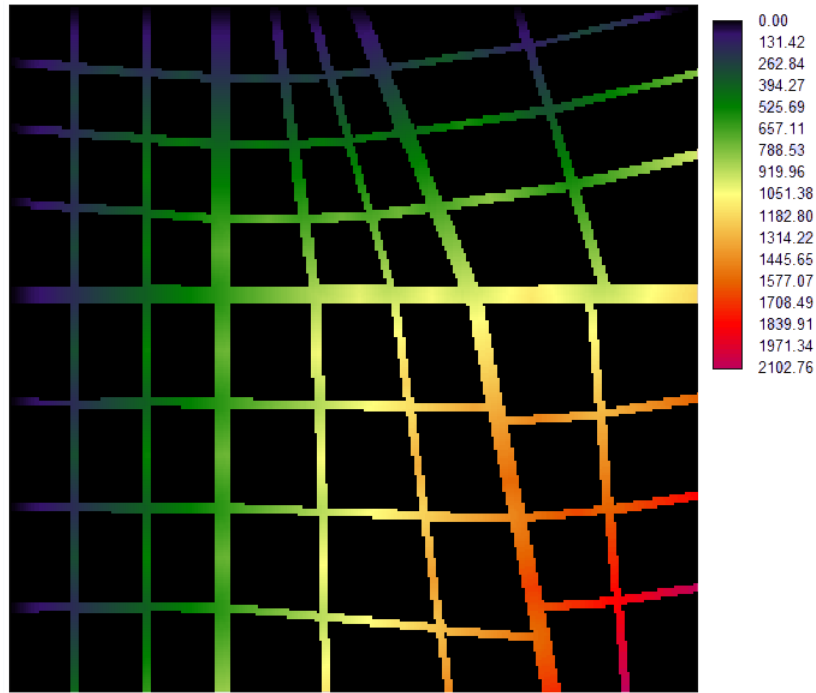


Figure 53 - Image raster *Flux*

Pour observer les différences entre l'image *Cout* initiale et l'image *Flux* issue de l'agrégation, nous les avons comparées. L'outil EXPAND est employé à cet esient. Il permet de créer une expansion de l'image *Flux*, séparant les mailles en sous-pixels selon un facteur. Dans notre cas, le facteur choisi est  $n$ , soit 2, afin d'obtenir le même nombre de pixels –et donc de lignes et de colonnes- qu'avant l'opération de contraction de l'image *Cout*. De cette façon, la nouvelle image *Flux\_expan* (voir Figure 54) dispose des mêmes paramètres que l'image *Cout*.

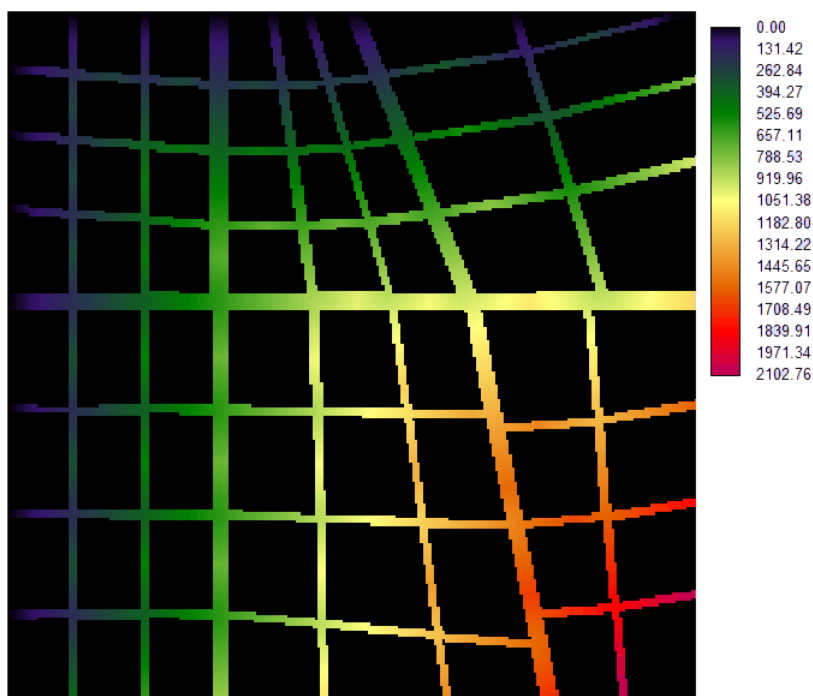


Figure 54 - Image raster *Flux\_expan*

La comparaison s'effectue grâce à l'application IMAGE CALCULATOR. L'image expansée, appelée *Flux\_expan*, divisée par  $n^2$  pour revenir à une valeur de coût par sous-pixel, est soustraite de l'image *Cout*. La formule est :

$$Comparaison\_Flux\_Cout = (Flux\_expan / 4) - Cout$$

Le résultat est une nouvelle image, *Comparaison\_Flux\_Cout* (voir Figure 55), qui permet de voir les effets de l'utilisation d'une résolution de maille plus basse d'un facteur 2.



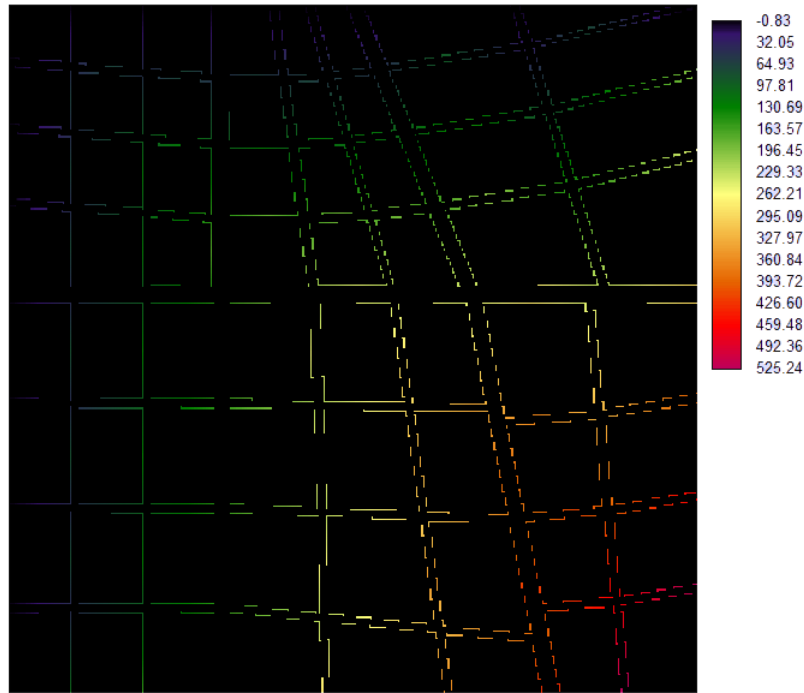


Figure 55 - Image raster *Comparaison\_flux\_cout*

Nous constatons évidemment des effets en bordure de voirie, dus aux sous-pixels de bâti présents dans des mailles dont le coût moyen correspond à la moyenne des sous-pixels de type voirie. Lors de l'expansion, ce coût moyen a été assigné à tous les sous-pixels de la maille, y compris ceux de bâti.

Après avoir reclassé l'image *Cout*, nous aurions pu utiliser le 0 comme valeur *flag* lors de l'option CONTRACT (voir Figure 56). A priori, la méthode serait plus simple puisque l'utilisation d'un masque ne serait pas nécessaire. Cependant, si dans une maille d'agrégation la majorité des sous-pixels sont de valeur *flag*, alors c'est cette valeur qui est assignée à la nouvelle maille et non la moyenne. Dans ce cas, des sous-pixels de type voirie risquent de ne pas être pris en compte. La comparaison (voir Figure 57), effectuée de la même façon que lors de l'agrégation sans valeur *flag*, montre effectivement que certains pixels de type voirie n'ont pas été considérés dans les calculs. De plus, cette méthode présente également des effets en bord de voirie dus à la résolution plus basse. L'utilisation de *flag* n'est donc pas une solution.

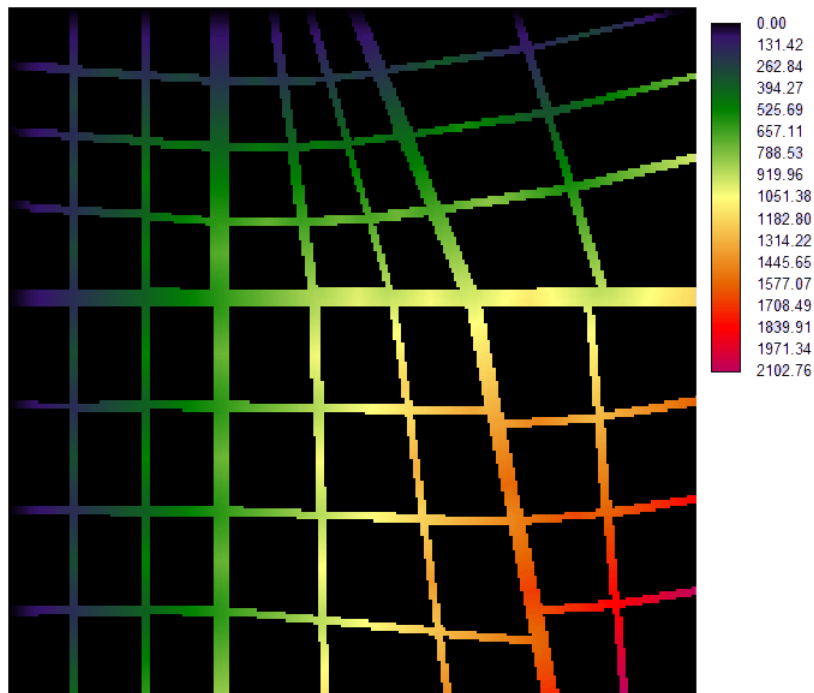


Figure 56 - Image raster *Flux\_flag*

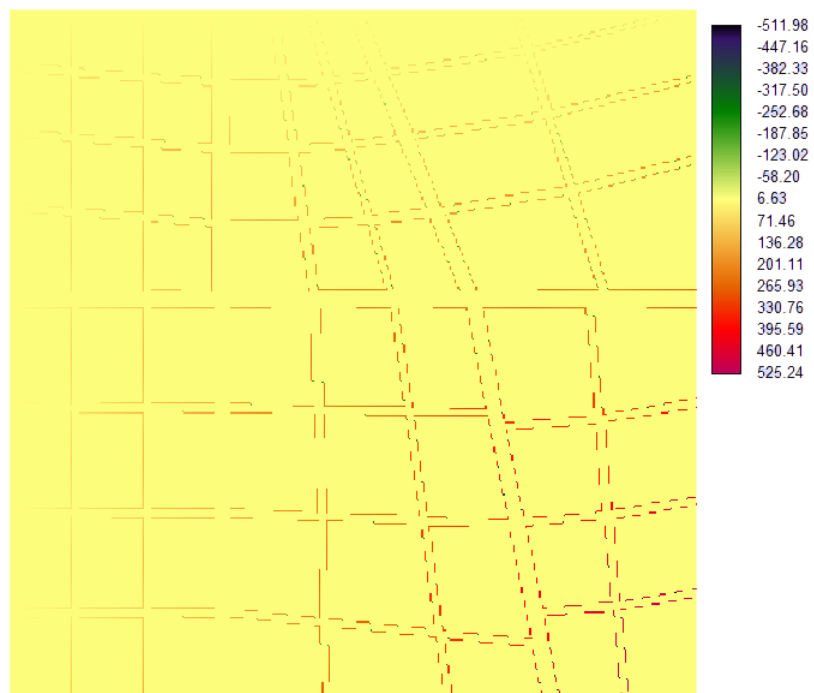


Figure 57 - Image raster *Comparaison\_flag\_cout*

### 6.3.2. Par moyenne des valeurs en sortie

Après discussion avec les ingénieurs du HECE, il s'avère que les résultats issus de la méthodologie précédente ne peuvent pas être implémentés dans leur modèle. Ce qu'ils recherchent est en fait un coefficient de propagation en sortie de chaque maille de

résolution grossière. Dans cet intérêt, nous avons envisagé une nouvelle approche. Celle-ci consiste à calculer pour chaque côté des mailles la valeur moyenne en sortie. Pour ce faire, nous allons diviser le modèle en quatre, formant ainsi quatre mailles de résolution grossière (voir Figure 58). Pour chacun des côtés de chaque maille, nous allons étudier la propagation dont la source est située le long du côté considéré. Ensuite, des valeurs moyennes seront calculées en sortie des autres côtés. La moyenne d'un côté est calculée à partir de tous les sous-pixels de la colonne ou de la ligne extrême correspondant à ce côté.

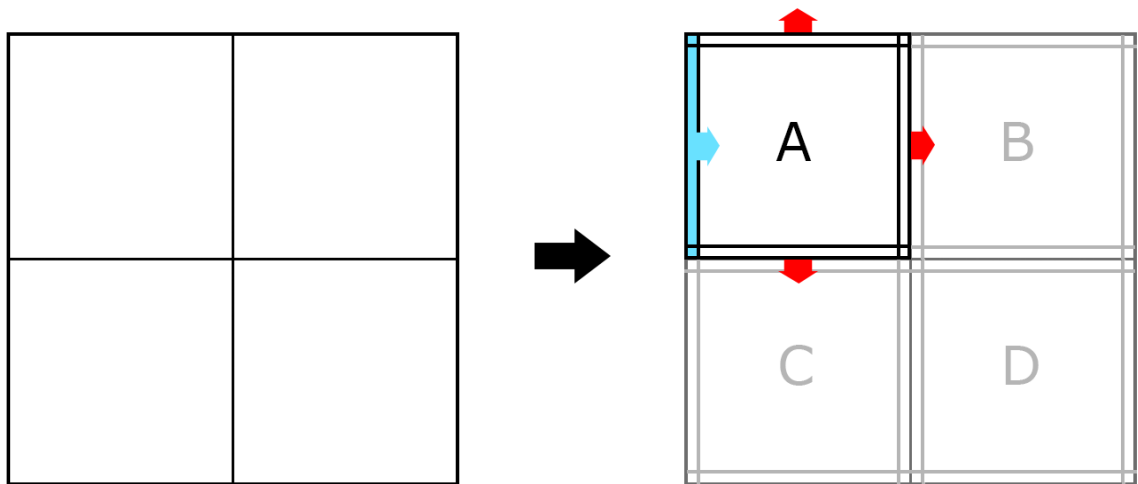


Figure 58 - Schéma du maillage grossier et des flux de propagation

La méthode utilise les outils d'IDRISI, mais des opérations similaires peuvent être envisagées sur un autre logiciel de SIG tel qu'ArcGIS.

Premièrement, les zones étudiées sont déterminées dans un masque. Ces zones sont les lignes et les colonnes situées aux extrêmes dont nous voudrions par après extraire la moyenne. Les zones se chevauchant dans les coins de la maille, deux masques devront être réalisés (voir Figure 59).

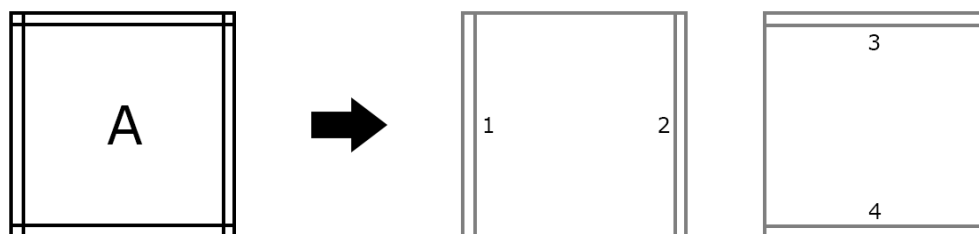


Figure 59 - Schéma des masques des zones d'étude

Pour obtenir ces masques sous IDRISI, différents outils doivent être utilisés (voir Figure 60).

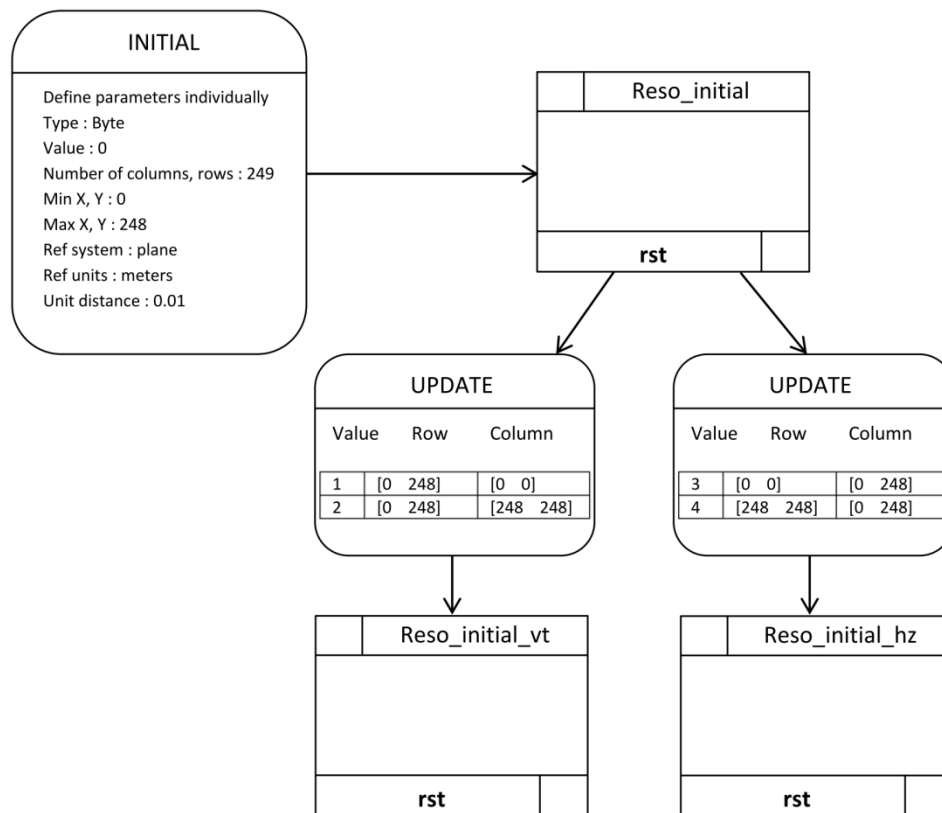


Figure 60 - Organigramme de création des masques des zones d'étude

- 1) **INITIAL** : une nouvelle image *raster* est créée, *Reso\_initial*. Les paramètres sont définis de façon à ce que les dimensions de cette image correspondent aux mailles grossières. Il suffit ici de diviser le nombre de colonnes et de lignes du modèle d'origine, c'est-à-dire 498, par deux. Les autres paramètres, comme le système de référence et les unités, sont similaires à ceux du modèle.
- 2) **UPDATE** : cette opération va permettre de différencier les lignes et les colonnes voulues en leur assignant une valeur propre. Elle est réalisée deux fois pour obtenir en sortie les images *Reso\_initial\_vt* (voir Figure 61) et *Reso\_initial\_hz* (voir Figure 62). La première comprend les colonnes extrêmes, numérotées 1 et 2, et la seconde les rangées extrêmes, numérotées 3 et 4.



Figure 61 - Image raster *Reso\_initial\_vt*

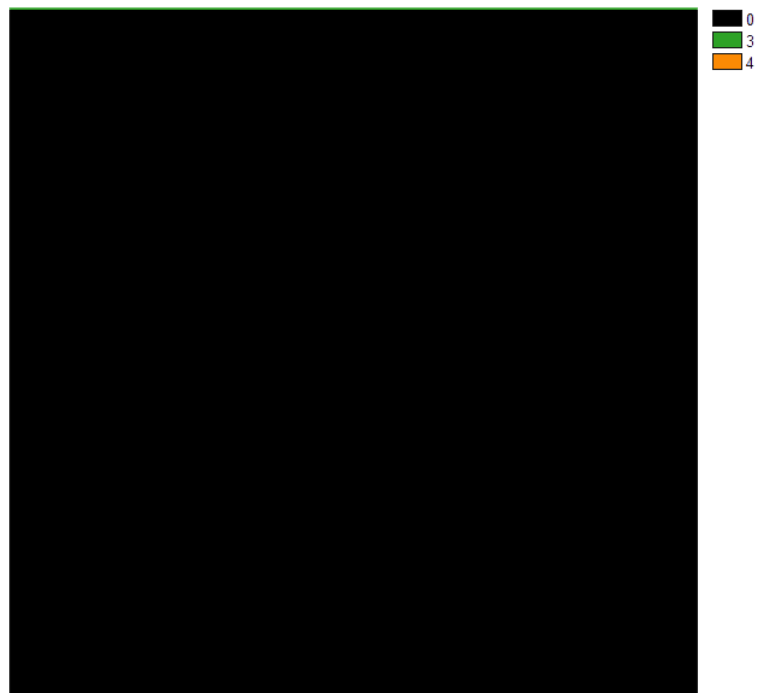
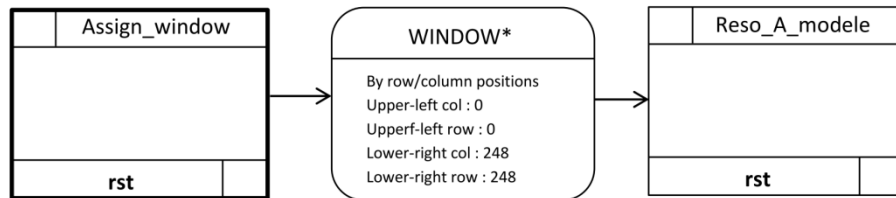


Figure 62 - Image raster *Reso\_initial\_hz*

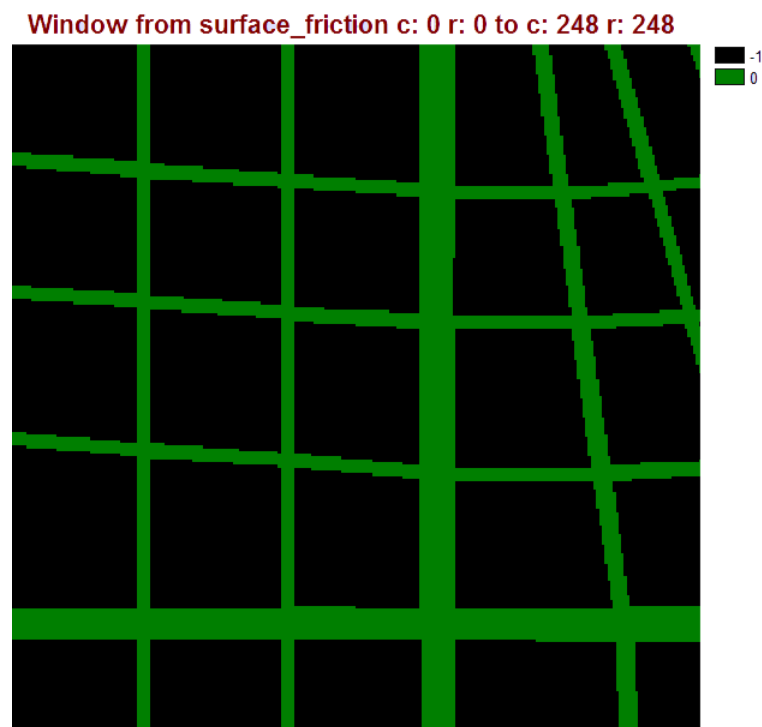
Deuxièmement, il faut adapter la surface de friction à la taille des mailles. La surface de friction étant le masque des voiries, il suffit de rogner l'image *Assign\_window* (voir Figure 29). L'outil WINDOW permet de recadrer l'image aux dimensions de la maille à l'aide des positions des lignes et des colonnes (voir Figure 63). Pour cet

exemple, nous avons décidé d'appliquer les traitements pour la maille A. Les lignes et les colonnes vont donc de 0 à 248. Il faudra bien évidemment adapter ces paramètres pour les autres mailles. Ici, l'image en sortie est nommée *Reso\_A\_modele* (voir Figure 64).



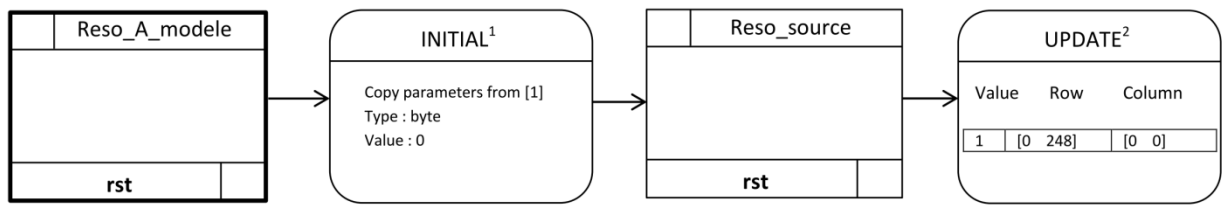
*\*Pour la maille A. Les paramètres sont à adapter pour les autres mailles.*

**Figure 63 - Organigramme d'obtention de la surface de friction pour la maille A**



**Figure 64 - Image raster *Reso\_A\_modele***

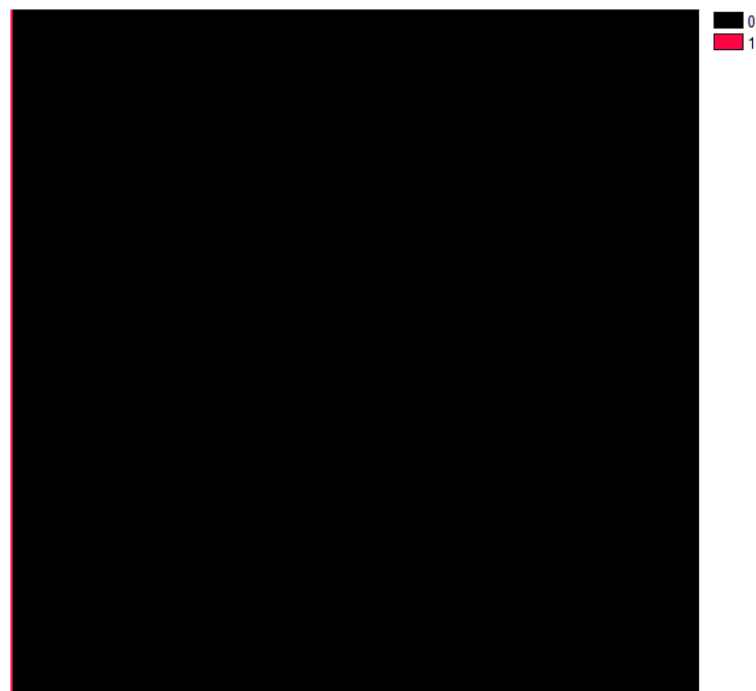
Ensuite, l'image source destinée à l'algorithme de « distance-coût » doit être définie (voir Figure 65). Elle est créée grâce au traitement INITIAL, avec les mêmes paramètres que *Reso\_initial* ou *Reso\_A\_modele*. Ensuite, l'option UPDATE permet d'assigner la valeur 1 au côté qui servira de source à l'algorithme. Ici, c'est la colonne 0 correspondant au côté Ouest qui est choisie comme source. Pour étudier la propagation depuis tous les côtés, l'opération doit être réalisée pour chacun de ces côtés. L'image résultante est *Reso\_source* (voir Figure 66).



<sup>1</sup>Pour la maille A. Les paramètres sont à adapter pour les autres mailles.

<sup>2</sup>Pour le côté Ouest. Les paramètres sont à adapter pour les autres côtés.

**Figure 65 - Organigramme d'obtention de l'image source pour le côté Ouest**



**Figure 66 - Image raster *Reso\_source***

L'algorithme de « distance-coût » COST peut enfin être lancé (voir Figure 67). *Reso\_A\_modele* est l'image de friction pour la maille A, et *Reso\_source* est l'image reprenant le côté Ouest comme source. L'algorithme choisi est *cost grow* et se base sur les coûts de distances. La surface de coûts générée en sortie est *Reso\_cout\_A\_Ouest* (voir Figure 68).

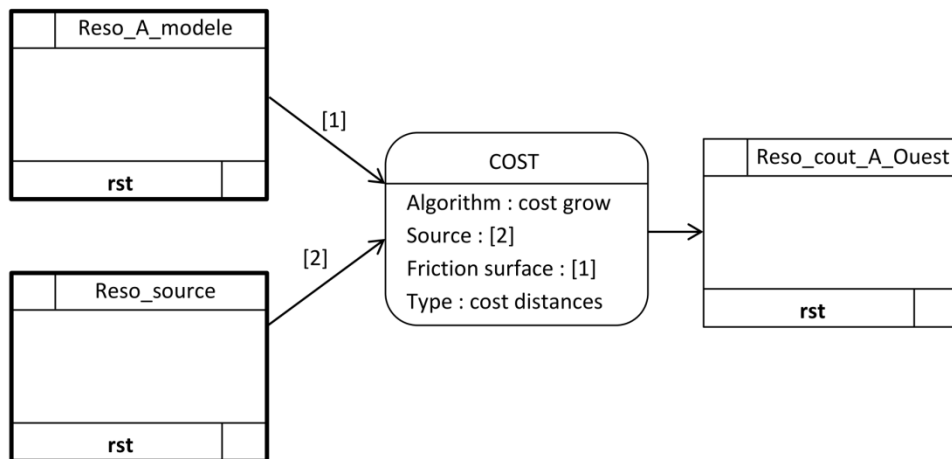


Figure 67 - Organigramme de l'algorithme COST

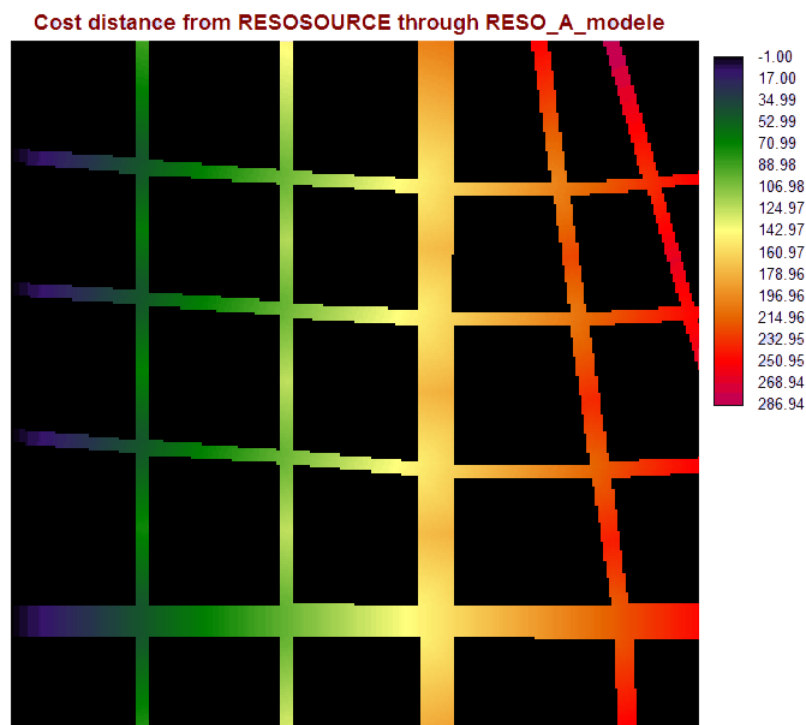
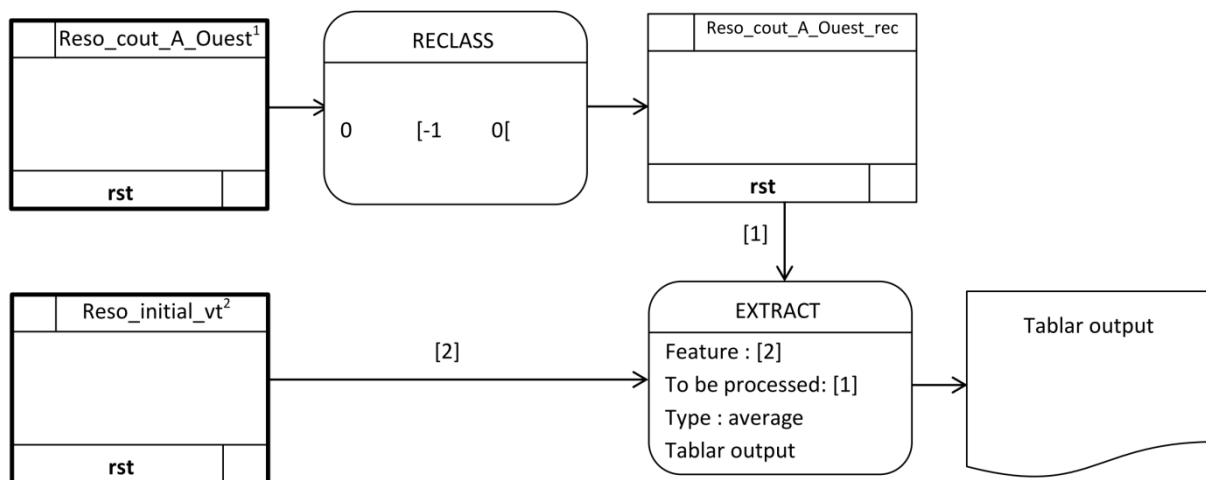


Figure 68 - Image raster *Reso\_cout\_A\_Ouest*

Enfin, il ne reste plus qu'à extraire les coûts moyens pour les zones définies dans les images *Reso\_initial\_vt* et *Reso\_initial\_hz*. L'outil EXTRACT est employé à cet effet (voir Figure 69). Ici, la manipulation est appliquée uniquement sur la surface de coûts générée suite à la propagation depuis le côté Ouest sur la maille A.





<sup>1</sup>Pour le côté Ouest de la maille A. La manipulation est aussi à réaliser pour les autres côtés de la maille et pour chaque maille.

<sup>2</sup>Pour les sorties des côtés verticaux. La manipulation est aussi à réaliser aussi pour les côtés horizontaux.

**Figure 69 - Organigramme d'extraction des moyennes**

L'image Reso\_cout\_A\_Ouest a préalablement été modifiée afin de remplacer la valeur des obstacles, tant qu'à présent de -1, par la valeur 0. La valeur -1 assignée aux obstacles va baisser le coefficient en sortie alors qu'en réalité le coût sur les zones bâties est nul. Pour faire cela, nous utilisons la fonction RECLASS. Cette opération terminée, nous obtenons l'image Reso\_cout\_A\_Ouest\_rec (voir...) avec nos valeurs de coûts initiales et les obstacles à 0.

L'extraction des moyennes via l'outil EXTRACT peut avoir lieu. En sortie, nous avons demandé un simple tableau (voir Tableau 1 et Tableau 2). Il est cependant possible d'enregistrer les résultats dans un fichier texte, ou encore dans une image.

```

Average values extracted from RESO_COUT_A_OUEST_FIN based on RESO_INITIAL_VT
Category      Average
0             32.107314
1             0.000000
2             45.784625
  
```

**Tableau 1 - Coefficient de coût moyen pour les zones 1 et 2 de la maille A à partir du côté Ouest**

Average values extracted from RESO\_COUT\_A\_OUEST\_FIN based on RESO\_INITIAL\_HZ

Category	Average
0	32.097041
3	28.144910
4	20.177238

**Tableau 2 - Coefficient de coût moyen pour les zones 3 et 4 de la maille A à partir du côté Ouest**

Dans ces tableaux, nous constatons que le côté numéroté 2 situé à l'opposé du côté de propagation a une valeur moyenne plus élevée que les autres.

Pour obtenir des coefficients en sortie pour tout le modèle, les traitements doivent être appliqués en considérant à tour de rôle comme source les différents côtés de la maille. Ensuite, la même méthodologie est appliquée aux autres mailles.

### 6.3.3. Discussion

L'objectif de ce traitement était de fournir une méthodologie pour traiter les données selon des résolutions de maille différentes. La première méthode proposée consistait à agréger les pixels entre eux pour former une maille plus grande. Nous observons cependant des artéfacts en bord de voirie. De plus, le résultat ne satisfaisait pas les ingénieurs en hydraulique. Ceux-ci demandèrent alors des coefficients de propagation aux sorties des mailles grossières. Comme coefficient, nous avons envisagé la valeur moyenne des sous-pixels en bordure de maille. Cette seconde méthode permet de choisir un côté de la maille comme source de propagation, et de sortir les valeurs moyennes de coûts en sortie des autres côtés. Ces coefficients semblent cohérents car la moyenne la plus élevée est celle du côté opposé à la source. Ce côté étant le plus éloigné de la source, il est normal que le coût de distance y soit plus élevé.

La seconde solution a été implémentée dans le modèleur graphique d'IDRISI (voir Figure 70). Ce modèle est réalisé pour la maille A, avec une propagation provenant du côté Ouest et uniquement pour les colonnes extrêmes. Les paramètres sont à adapter si d'autres conditions sont posées.

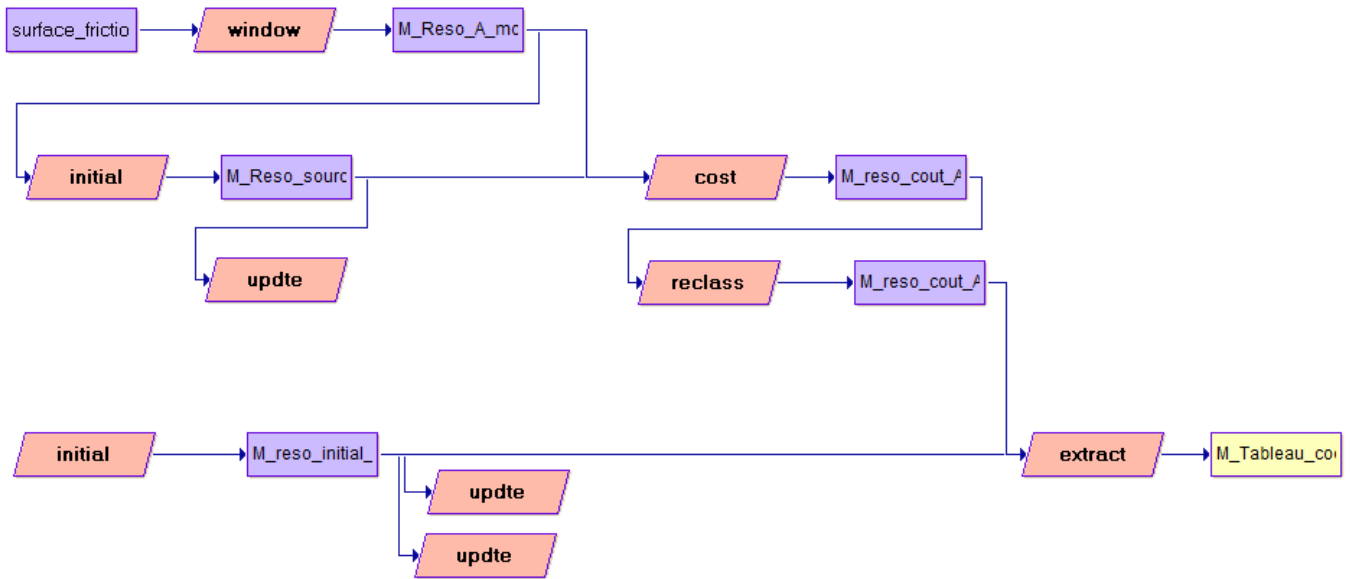


Figure 70 - Modèle d'extraction de coefficients de propagation à partir des coûts moyens des pixels extrêmes sous IDRISI

## 6.4. Extraction des axes de voirie

Au cours de ce travail, une méthode pour extraire les axes de voirie a également été conçue. Le but initial d'un tel modèle est d'obtenir des informations sur les rues telles que leur largeur ou leur longueur. Dans le cas présent, où nous nous basons sur un prototype réduit dont les dimensions ont été choisies, il est vrai que cela présente peu d'intérêt. Cependant, ces informations peuvent s'avérer intéressantes pour de futurs modèles hydrauliques partant de cas réels.

### 6.4.1. Méthodologie

La méthodologie suivante est réalisée sous QGIS. Pour commencer, nous allons créer une couche *Shapefile* intitulée *Voirie* qui représentera le masque des rues. Les traitements à appliquer sont les suivants (voir Figure 71).

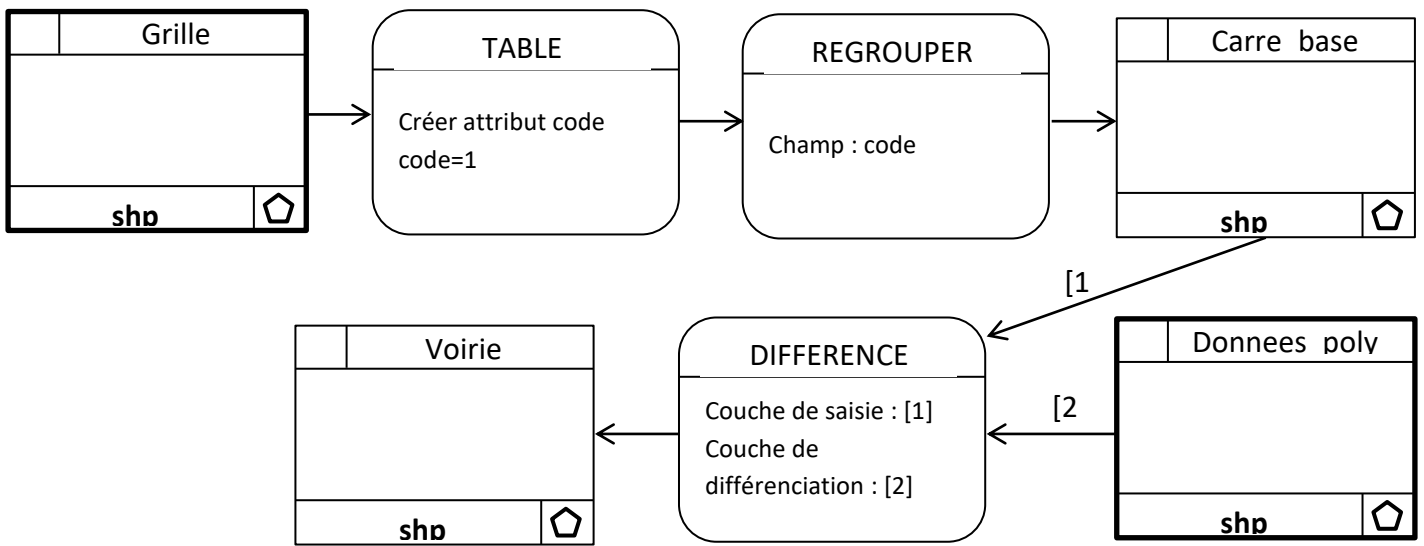


Figure 71 – Organigramme d'extraction de la voirie sous QGIS

- 1) TABLE MANAGER: une nouvelle colonne est ajoutée à la table d'attributs de la couche *Grille* précédemment créée (voir Figure 10). Ce nouvel attribut est nommé *code*, et la valeur 1 lui est assignée pour toutes les entités (voir Figure 72).

	left	top	right	bottom	code
0	0.0000000000...	0.0000000000...	0.0100000000...	0.0100000000...	1
1	0.0000000000...	0.0100000000...	0.0100000000...	0.0200000000...	1
2	0.0000000000...	0.0200000000...	0.0100000000...	0.0300000000...	1
3	0.0000000000...	0.0300000000...	0.0100000000...	0.0400000000...	1
4	0.0000000000...	0.0400000000...	0.0100000000...	0.0500000000...	1
5	0.0000000000...	0.0500000000...	0.0100000000...	0.0600000000...	1
6	0.0000000000...	0.0600000000...	0.0100000000...	0.0700000000...	1
7	0.0000000000...	0.0700000000...	0.0100000000...	0.0800000000...	1
8	0.0000000000...	0.0800000000...	0.0100000000...	0.0900000000...	1
9	0.0000000000...	0.0900000000...	0.0100000000...	0.1000000000...	1
10	0.0000000000...	0.1000000000...	0.0100000000...	0.1100000000...	1
11	0.0000000000...	0.1100000000...	0.0100000000...	0.1200000000...	1
12	0.0000000000...	0.1200000000...	0.0100000000...	0.1300000000...	1
13	0.0000000000...	0.1300000000...	0.0100000000...	0.1400000000...	1
14	0.0000000000...	0.1400000000...	0.0100000000...	0.1500000000...	1
15	0.0000000000...	0.1500000000...	0.0100000000...	0.1600000000...	1
16	0.0000000000...	0.1600000000...	0.0100000000...	0.1700000000...	1
17	0.0000000000...	0.1700000000...	0.0100000000...	0.1800000000...	1
18	0.0000000000...	0.1800000000...	0.0100000000...	0.1900000000...	1

Figure 72 – Aperçu de la table d'attributs de la couche *Grille*

- 2) REGROUPER: au cours de ce traitement, les entités de la couche *Grille* dont le champ *code* est similaire sont regroupées. Puisque la valeur de cet attribut est la même pour toutes les entités, l'entièreté de la grille sera regroupée en un carré unique dans la couche résultante *Carre\_base* (voir Figure 73Figure ).

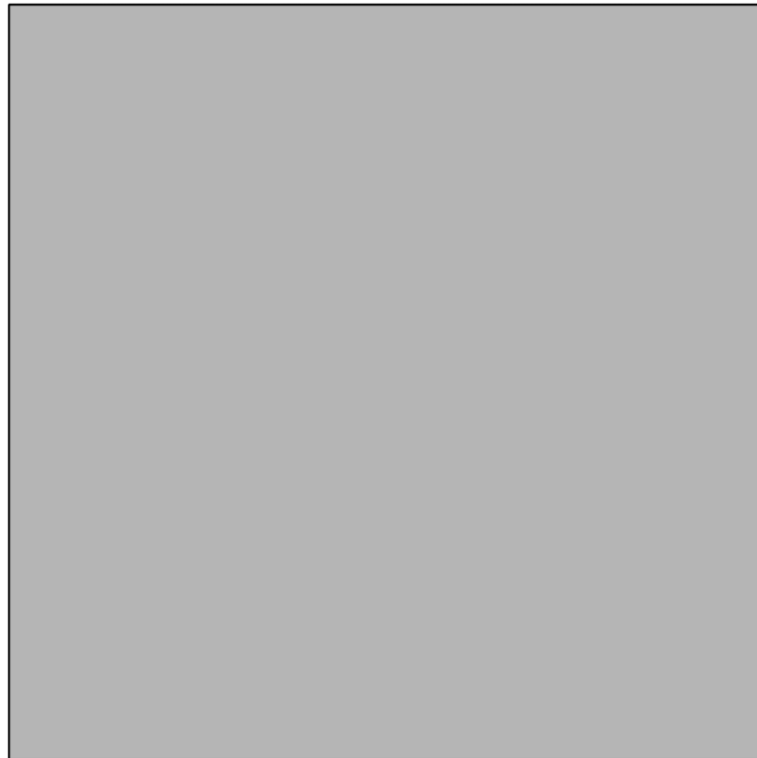


Figure 73 – Couche *Shapefile Carre\_base*

- 3) DIFFERENCE: la différence de ce carré unique avec la couche *Donnees\_poly* (voir Figure 3) reprenant les blocs de bâti fournit en sortie une couche *Shapefile Voirie*. Dans cette dernière, seules les rues (ainsi que la fine bordure d'erreur sur le contour de la couche) sont représentées. Elles sont reprises comme une seule entité polygonale (voir Figure 74).

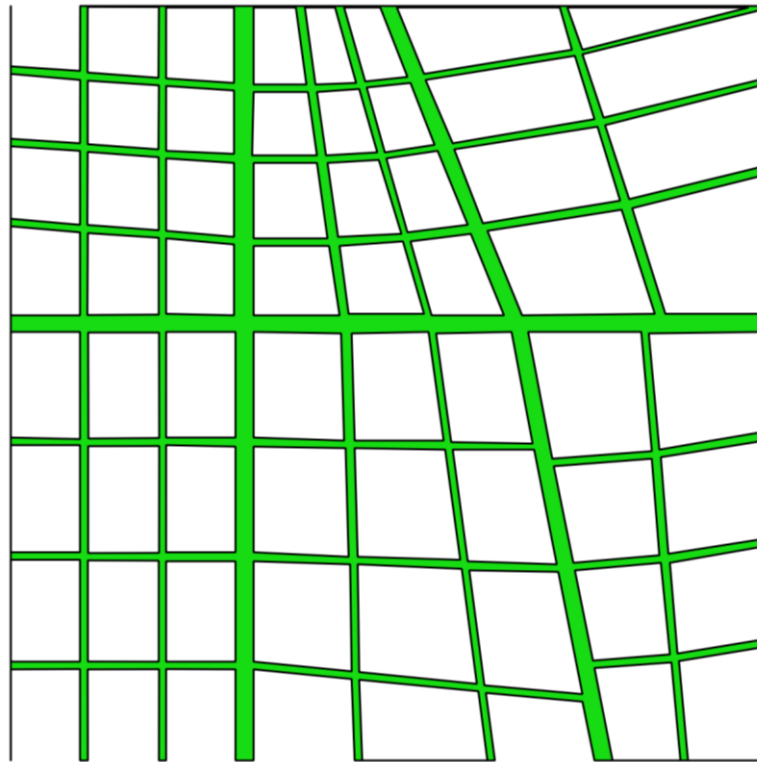


Figure 74 – Couche Shapefile Voirie

Une fois le masque des rues créé, il reste à déterminer les axes centraux de ces rues. Pour ce faire, nous allons utiliser le plugin « Polygones de Voronoï » du logiciel.

Il s'agit de l'application de l'algorithme des polygones de Thiessen basé sur les diagrammes de Voronoï. Cet algorithme part d'un ensemble de  $n$  points  $a_1, a_2, \dots, a_n$  bien définis dans un espace. L'objectif est de séparer l'espace en différentes régions à partir de ces points. Il existe différentes méthodes pour y parvenir telles que la méthode de la triangulation ou la méthode basée sur une grille de pixels uniforme (Han & Bray, 2006). L'idée générale de l'algorithme de Thiessen est que, pour chacun des  $n$  points, nous associons un ensemble de localisations  $x$  qui, elles, ne sont pas échantillonnées. Chaque localisation  $x$  est associée au point  $a$  pour lequel la distance l'en séparant est minimale en comparaison aux distances avec les autres points  $a$  (Boots, 1980; Tatalovich *et al.*, 2016). L'ensemble des localisations  $x$  associées à leurs points  $a$  fournissent un set de polygones de Thiessen. Par exemple, le polygone de Thiessen  $A_1$  est formé à partir d'un point  $a_1$  défini et comprend toutes les localisations  $x$  dont  $a_1$  est le point le plus proche. Si  $x$  est à équidistance avec deux points  $a$ , il se situera sur la frontière des deux polygones adjacents (Boots, 1980).

De manière générale, nous allons prendre comme points de base des points situés sur les bords des blocs de bâtis. Certaines localisations, générées au cours de l'algorithme, se situent à équidistance entre les points des bords de deux blocs de bâti séparés uniquement par une rue. Ces localisations sont dès lors situées au centre de

cette rue. Nous pouvons en conclure que, lors du tracé des polygones de Thiessen, le centre de la rue sera représenté par la frontière entre les polygones de Voronoï des deux blocs de bâti voisins.

Le modèle suivant (voir Figure 75) décrit les traitements à réaliser pour obtenir les polygones de Thiessen, puis comment en isoler les axes centraux des rues.

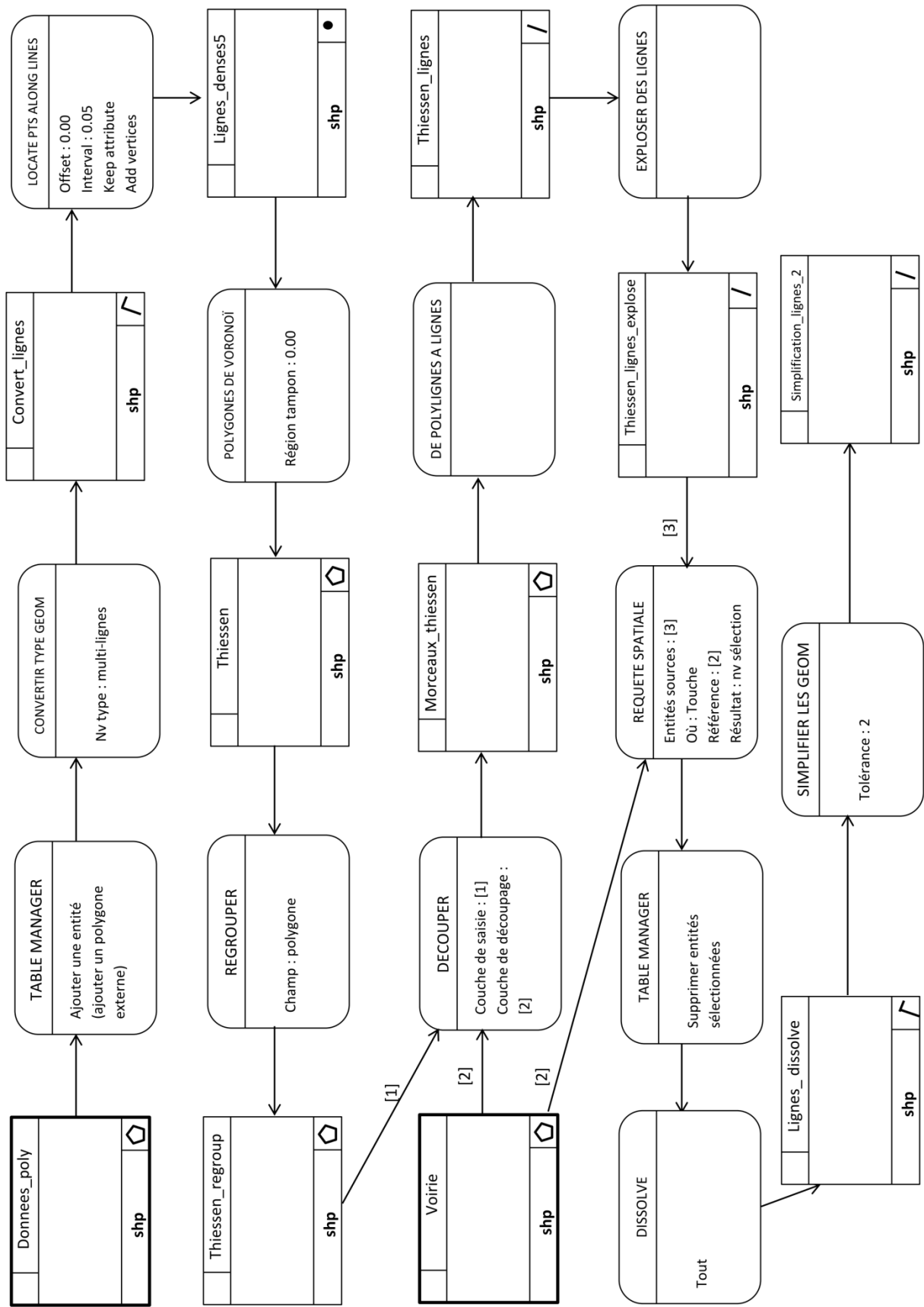


Figure 75 – Organigramme d'extraction des axes de voirie sous QGIS



- 1) TABLE MANAGER: en partant de la couche *Donnees\_poly* (voir Figure 3), il faut ajouter un polygone externe autour des données (voir Figure 76Figure ). Ici, l'opération a été réalisée après la conversion en polylignes, l'entité ajoutée le sera donc également. Cet ajout est nécessaire pour éviter des effets de bord. En effet, sans polygone externe, les axes des rues risquent de ne pas se prolonger de façon linéaire jusqu'aux extrémités de celles-ci.

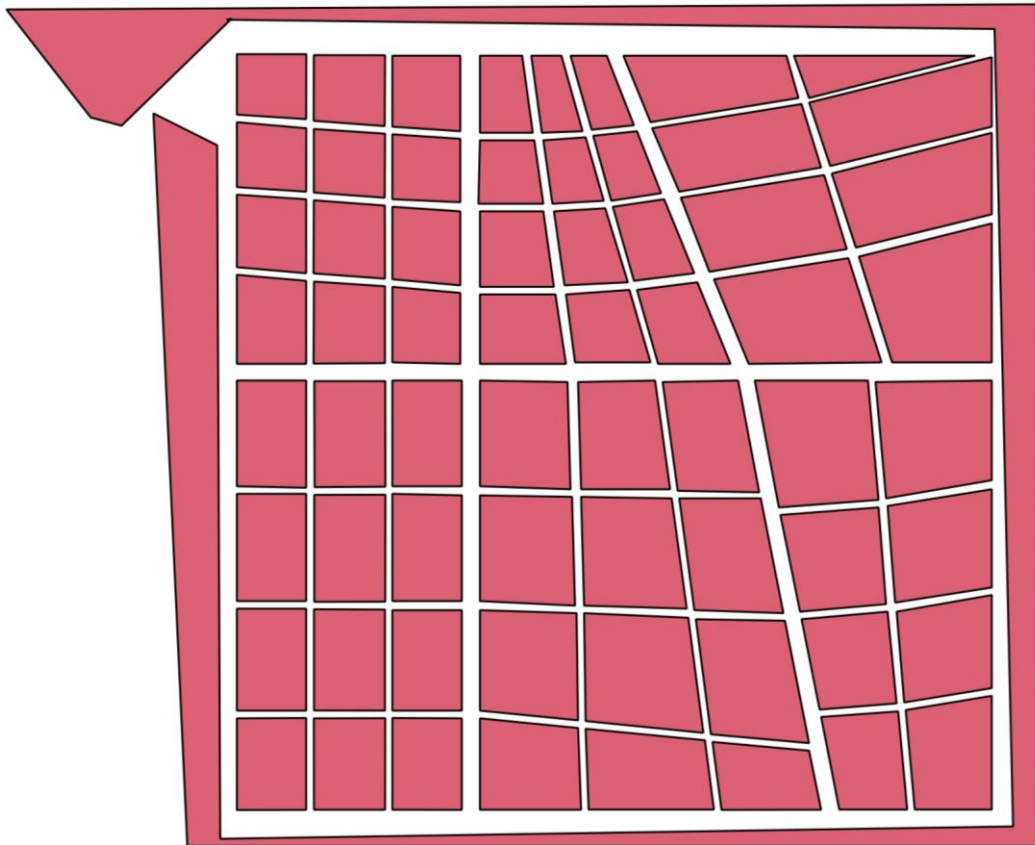
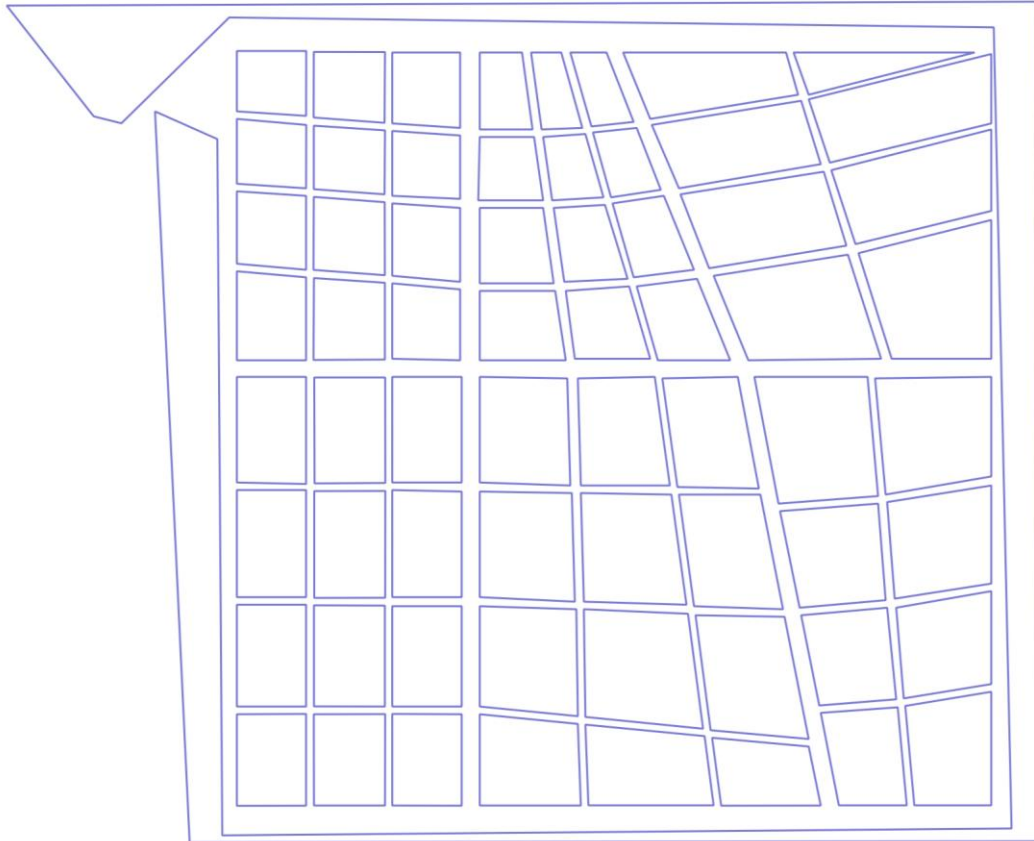


Figure 76 - Couche *Shapefile Donnees\_poly* après ajout d'un polygone externe

- 2) CONVERTIR LE TYPE DE GEOMETRIE: les obstacles représentés jusque-là par des polygones sont convertis en multi-lignes (voir Figure 77) dans la couche *Convert\_lignes* afin de permettre la réalisation de l'étape 3.



**Figure 77 - Couche Shapefile *Convert\_lignes***

- 3) LOCATE POINTS ALONG LINES: cette extension de QGIS permet de créer une couche de points positionnés le long des lignes (voir Figure 78Figure ). Cette étape est importante car l'algorithme des polygones de Voronoï s'applique sur des points. Il faut soigneusement choisir l'intervalle entre ces derniers. Un nombre insuffisant de points engendrerait des résultats trop grossiers et, au contraire, une quantité trop importante rendrait les traitements suivants trop coûteux. Ici, un intervalle de 0.05 a été choisi, et le résultat a été enregistré dans la couche *Lignes\_denses5*.

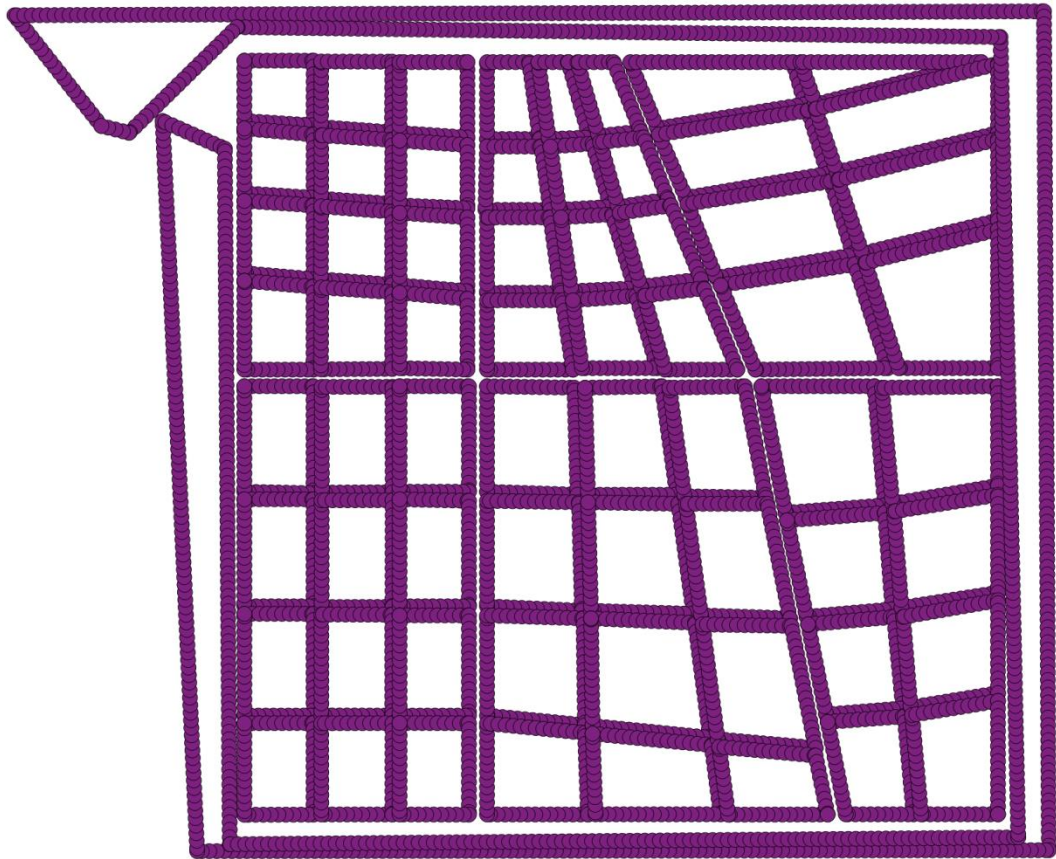


Figure 78 - Couche *Shapefile Lignes\_denses5*

- 4) POLYGONES DE VORONOÏ: enfin, les polygones de Thiessen peuvent être générés. Si la densité de points utilisés était suffisante, l'organisation initiale des obstacles et de la voirie devrait encore pouvoir se deviner dans la couche *Thiessen* en sortie (voir Figure 79).

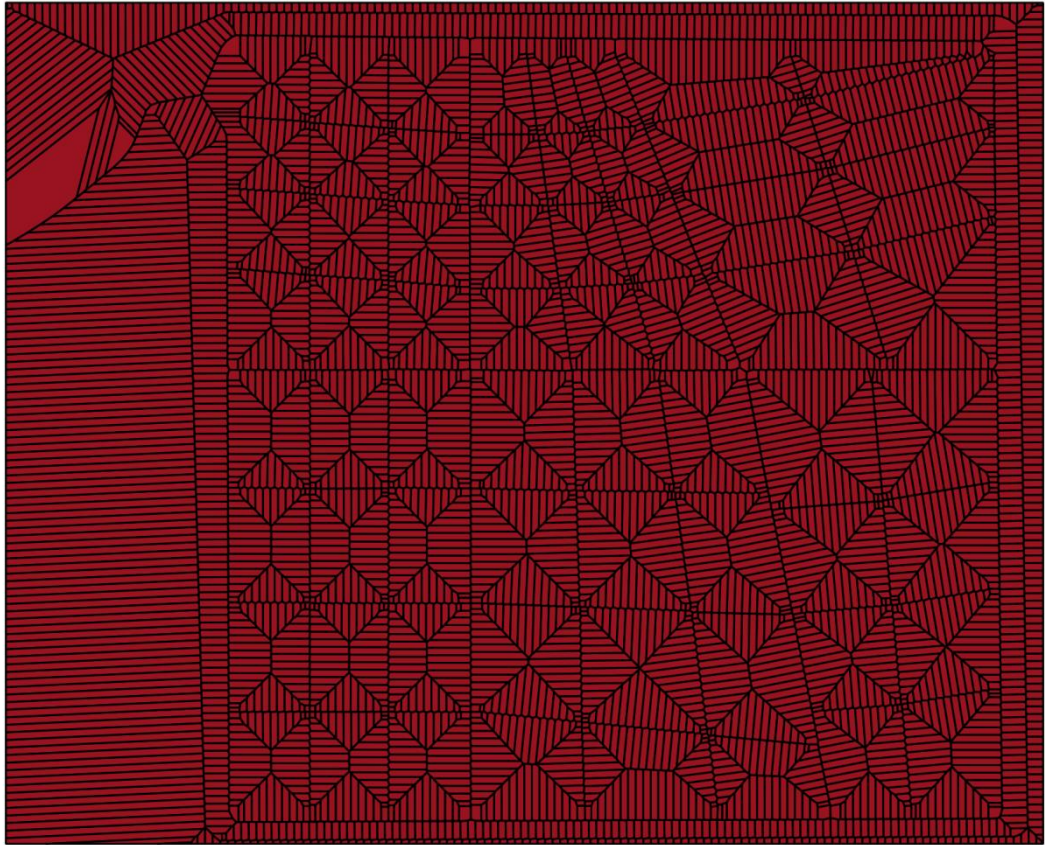
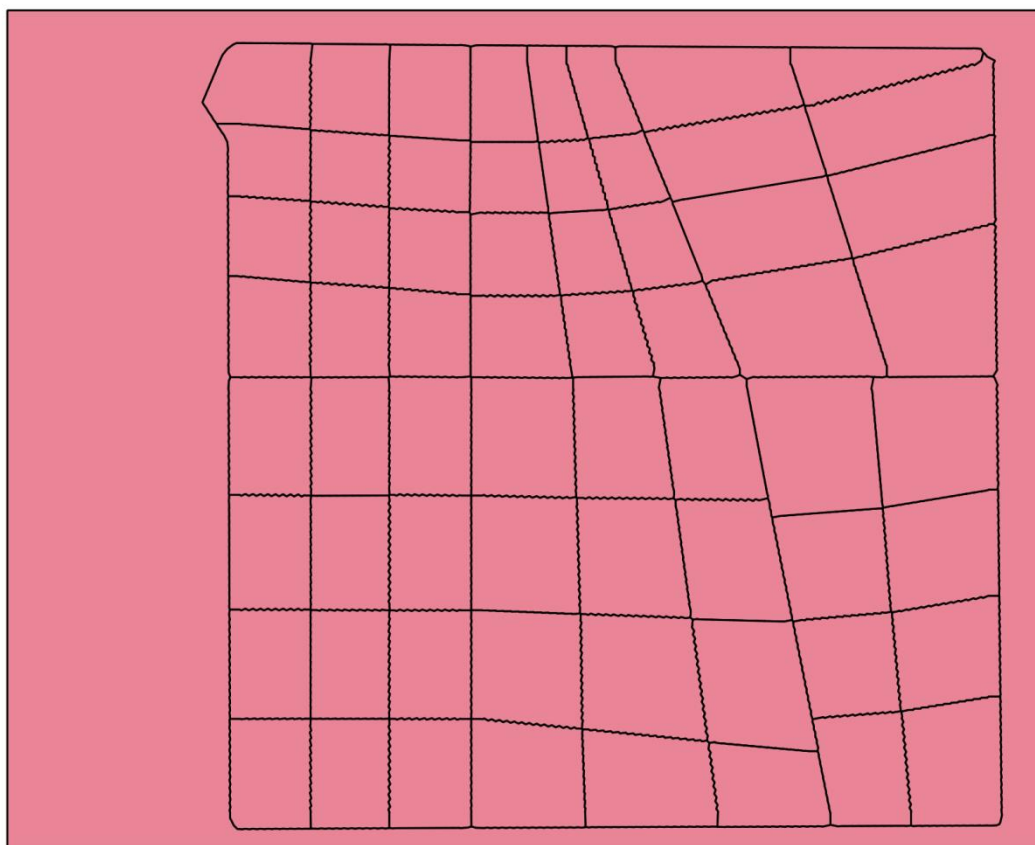


Figure 79 – Couche *Shapefile Thiessen*

- 5) REGROUPER: les polygones de Thiessen sont regroupés en fonction du nom (qui sert d'identifiant) des polygones obstacles. La couche en sortie est nommée *Thiessen\_regroup* (voir Figure 80).



**Figure 80 - Couche *Shapefile Thiessen\_regroup***

- 6) DECOUPER: La couche précédente est découpée avec le masque des voiries (voir Figure 74) pour obtenir la couche *Morceaux\_thiessen* (voir Figure 81).

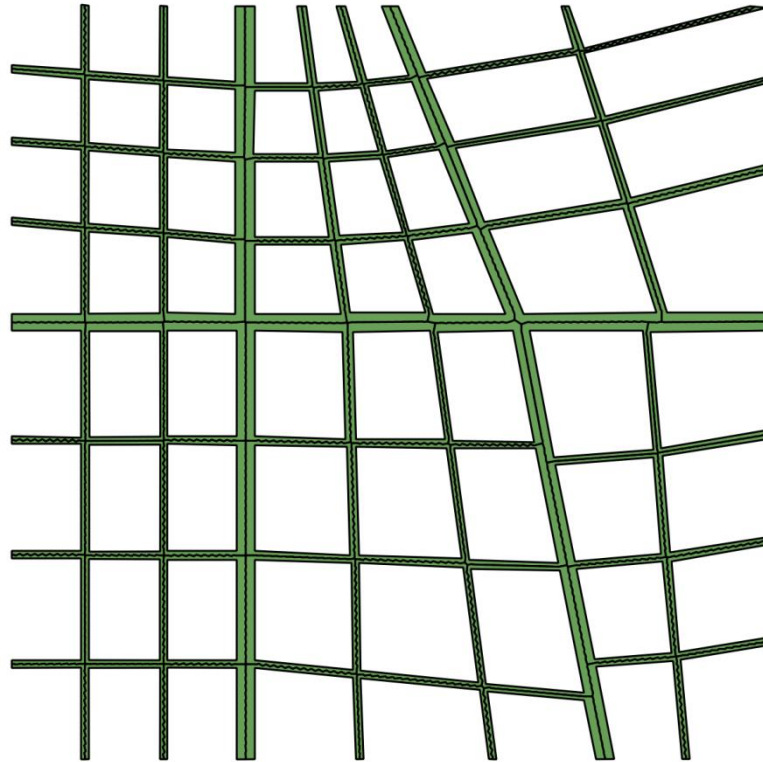
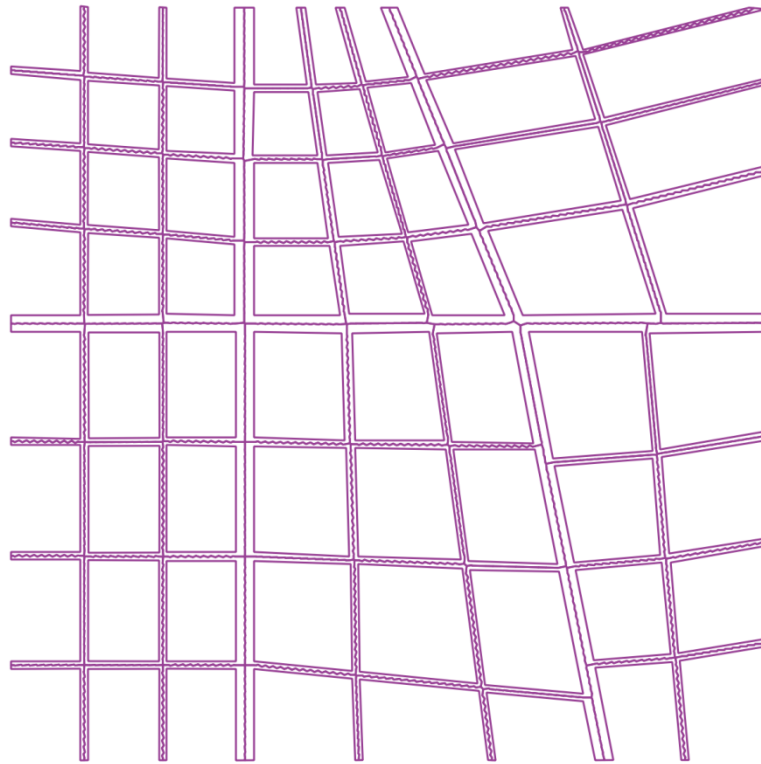


Figure 81 - Couche *Shapefile Morceaux\_Thiessen*

7) DE POLYLIGNES A LIGNES: Les polygones de la couche *Morceaux\_thiessen* sont converties en lignes pour fournir la couche *Thiessen\_lignes* (voir Figure 82).



**Figure 82 - Couche Shapefile *Thiessen\_lignes***

- 8) EXPLOSER DES LIGNES: comme le nom de l'outil l'indique, les lignes sont ensuite explosées, formant une nouvelle entité linéaire à chaque nœud. Les résultats sont repris dans une couche *Thiessen\_ligne\_explose*.
- 9) REQUETE SPATIALE: une requête spatiale est réalisée de façon à ce que toutes les entités de *Thiessen\_ligne\_explose* touchant celles de la couche *Voirie* soient sélectionnées.
- 10) TABLE MANAGER: la table d'attributs de la couche *Thiessen\_ligne\_explose* est éditée de façon à supprimer toutes les entités sélectionnées à l'étape précédente. Il ne reste ainsi que les entités qui font partie des axes centraux des rues (voir Figure 83).

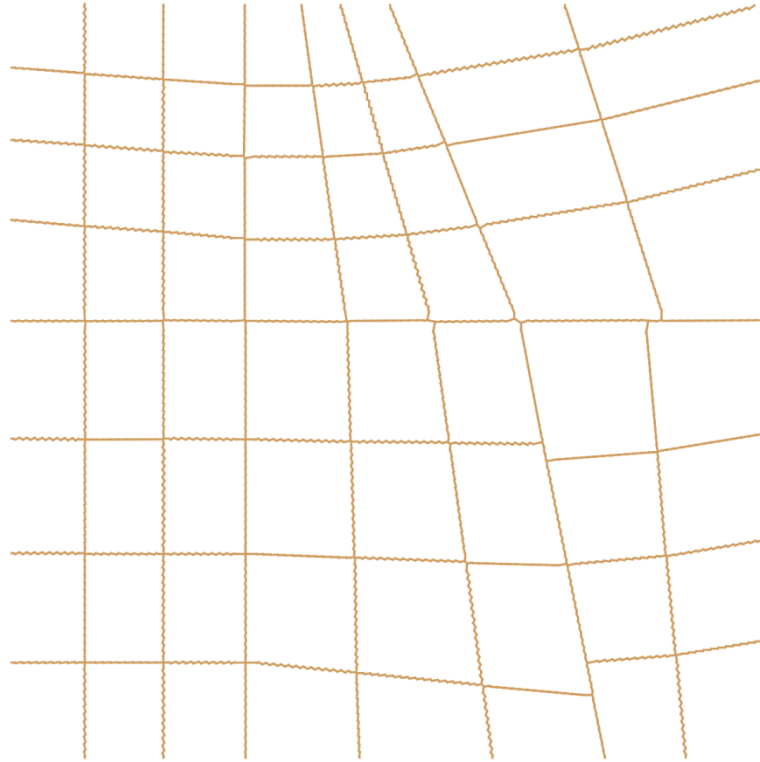


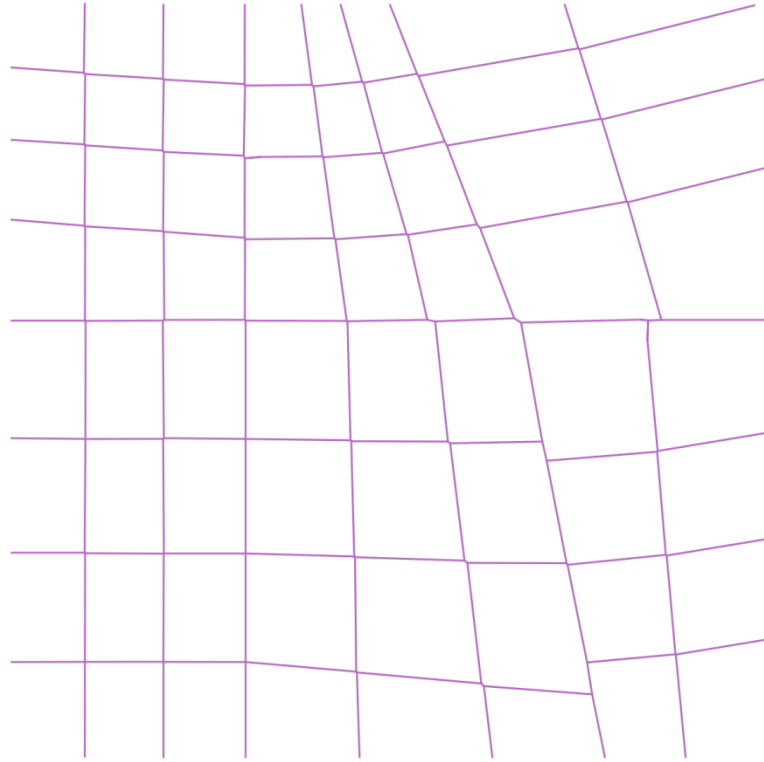
Figure 83 - Couche *Shapefile Thiessen\_ligne\_explose*

- 11) DISSOLVE<sup>4</sup>: cet outil permet de regrouper les entités une fois qu'elles ont été triées pour ne conserver que celles qui nous intéressent pour les axes de voirie. Elles sont fusionnées en une seule entité.
  
- 12) SIMPLIFIER LES GEOMETRIES: enfin, une simplification des géométries s'impose pour lisser les axes centraux des rues. En effet, certaines rues sont parcourues par de nombreuses lignes imbriquées en dents de scie. Plusieurs tolérances de simplification différentes ont été testées avant de trouver celle qui convenait le mieux. Dans un premier temps, une simplification avec la tolérance à 5 a engendré un réseau très lisse (voir Figure 85), mais les axes s'éloignent parfois significativement des centres des rues (voir Figure 86), et les intersections ne s'effectuent pas toujours correctement au centre des carrefours. Une simplification à 1 de tolérance a également été testée. Cette dernière est insuffisante car certaines lignes ont conservé un aspect en dents de scie (voir Figure 87). Finalement, dans notre cas, une simplification avec un facteur de tolérance de 2 semble un bon compromis (voir Figure 88). Il n'y a plus de dents de scie, et les axes suivent relativement bien les centres des rues (voir Figure 89).

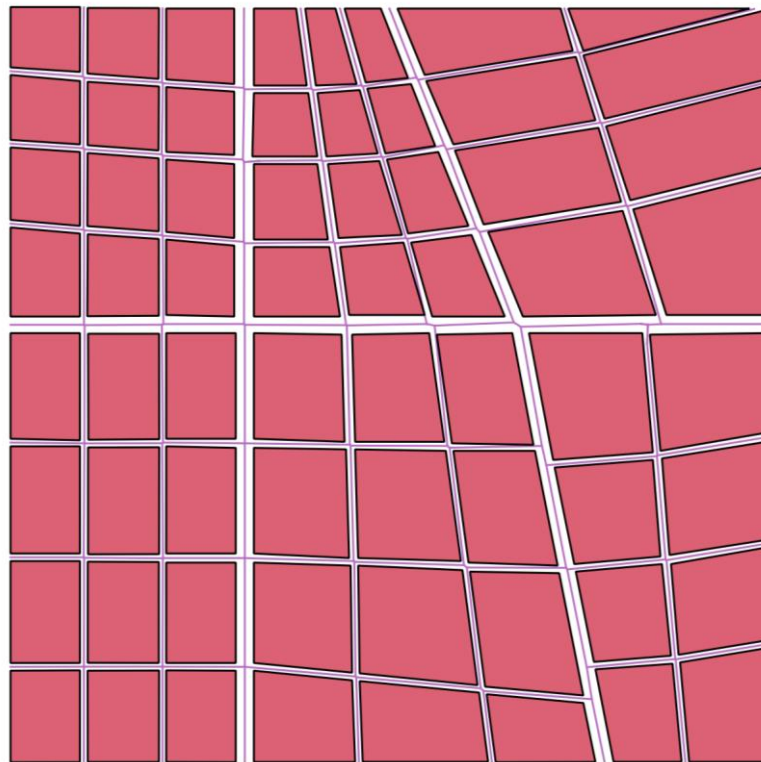
---

<sup>4</sup> DISSOLVE (EN) correspond à l'outil REGROUPER (FR), mais la traduction en français n'a pas été réalisée dans la boîte à outil de QGIS 2.14.

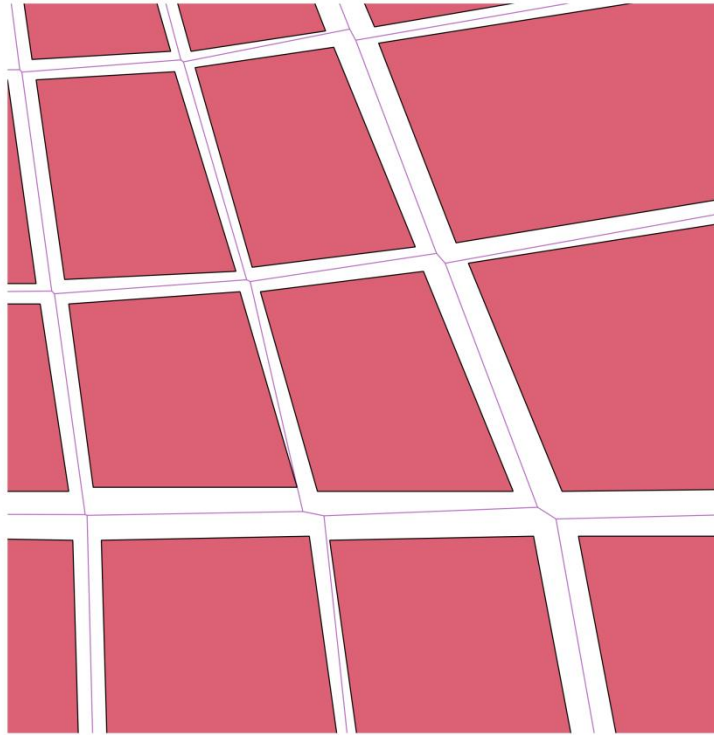




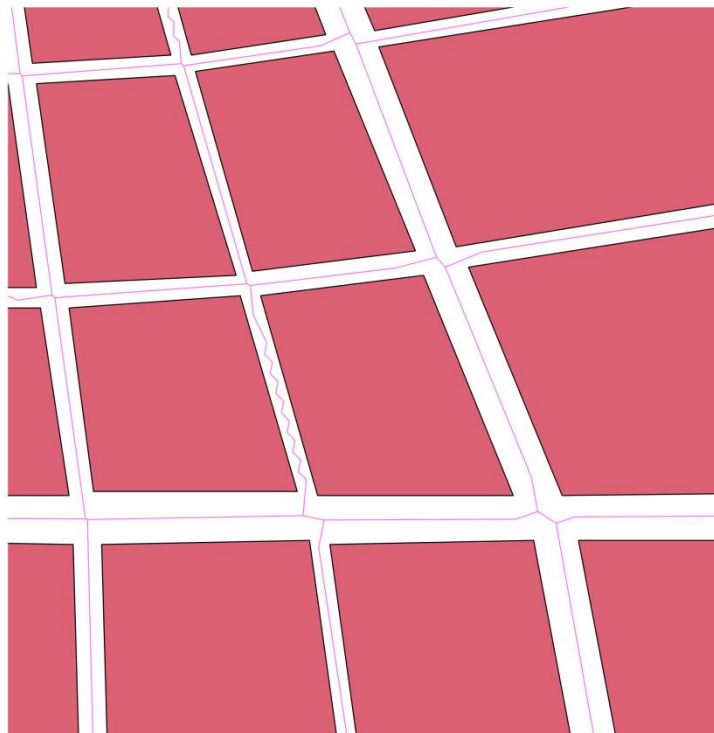
**Figure 84 - Couche Shapefile *Simplification\_lignes\_5***



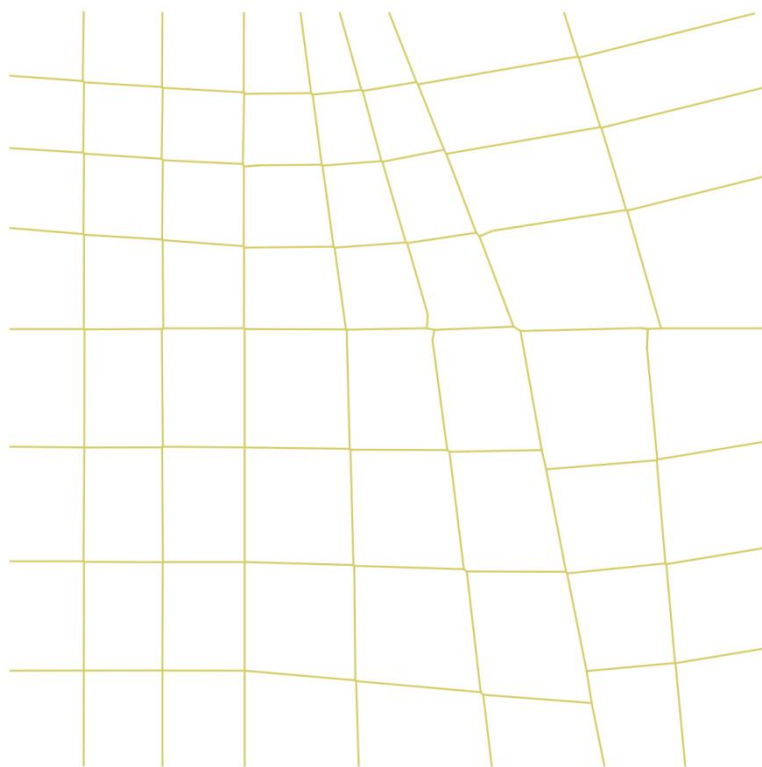
**Figure 85 - Couches Shapefiles *Simplification\_lignes\_5* et *Donnees\_poly***



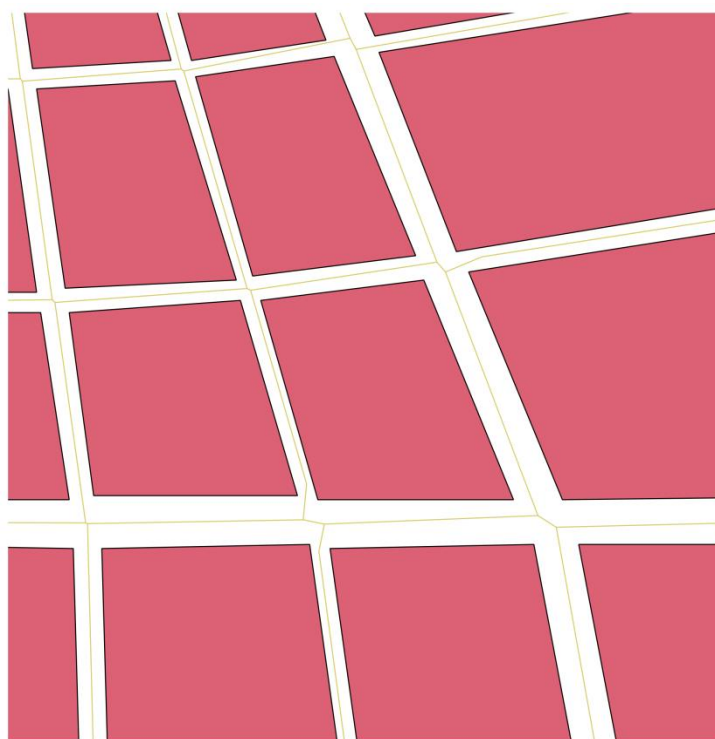
**Figure 86 - Zoom sur les couches *Shapefiles Simplification\_lignes\_5* et *Donnees\_poly***



**Figure 87 - Zoom sur les couches *Shapefiles Simplification\_lignes\_1* et *Donnees\_poly***



**Figure 88 - Couche Shapefile *Simplification\_lignes\_2***



**Figure 89 - Zoom sur les couches Shapefiles *Simplification\_lignes\_2* et *Donnees\_poly***

## 6.4.2. Discussion

Il est évident que certaines valeurs choisies au cours de cette méthode doivent être adaptées en fonction de la précision et de la quantité de données de départ. Par exemple, le traitement consistant à localiser des points le long des lignes risque d'être lourd si les données initiales sont nombreuses.

Nonobstant, la méthodologie présentée ici semble efficace mais doit être nuancée. Étant donné que la méthode fait appel à des algorithmes de généralisation tel que la simplification des géométries, le résultat obtenu n'est qu'une approximation des axes centraux des voiries. Nous avons d'ailleurs constaté que le centre des carrefours obtenu par cette méthode s'éloignait parfois significativement du centre réel.

## 7. Conclusion et perspectives

---

Suite à la demande de membres du HECE, nous avons utilisé la géomatique et les outils qui en découlent pour fournir des méthodologies susceptibles d'être intégrées dans un modèle hydraulique. Ce dernier modélise les inondations en région urbaine.

Il ne fait aucun doute que la géomatique peut apporter énormément aux études sur les inondations. D'ailleurs, ses apports en termes d'acquisition des données sont déjà très importants. En constante évolution depuis plusieurs dizaines d'années, son utilisation dans les traitements de données n'a pas encore été complètement explorée. Ce mémoire montre que les SIG proposent déjà de nombreux outils, et que ces derniers peuvent être utilisés de plusieurs façons dans des études sur les inondations.

Un jeu de données nous a été fourni pour tester les méthodologies imaginées. Il reprend les coordonnées de polygones faisant office de zones de bâti imperméables. Les espaces entre les polygones sont considérés comme un réseau de voiries où des écoulements peuvent se propager.

Les méthodologies présentées dans ce mémoire utilisent les outils de logiciels de SIG. Ces logiciels sont QGIS Desktop, IDRISI Selva, et ArcGIS. Pour chaque méthode, un logiciel en particulier est utilisé mais, à priori, les traitements pourraient également être effectués avec d'autres logiciels de SIG. Les données sont initialement importées sur QGIS en format vectoriel.

Premièrement, nous proposons une méthode sous QGIS pour représenter la voirie et les zones de bâti d'une ville fictive en mode raster, sans généraliser l'information d'affectation à la résolution du pixel. Pour faire cela, la méthode extrait comme valeur la part de voirie de chaque pixel. Elle correspond à la part du pixel où un écoulement peut avoir lieu et est nommée porosité. Les valeurs de porosité sont générées sous forme d'un attribut, et les entités polygonales auxquelles elles sont assignées ont les mêmes dimensions que des pixels carrés de 1 cm de côté.

Deuxièmement, nous avons envisagé de simuler la propagation d'un écoulement dans la ville modèle. Nous avons utilisé l'algorithme de « distance-coût » des logiciels à cet effet. L'algorithme a été lancé sous IDRISI en partant du modèle maillé et en généralisant chaque pixel à une seule affectation (voirie ou bâti). Le résultat est une surface de distance coût. La méthode semble cohérente pour représenter le flux général d'un écoulement à travers une région urbaine, mais n'est toutefois pas assez complexe. En effet, la largeur des rues n'influence pas le flux, contrairement à leur obliquité qui, lorsqu'elle est importante, oblige l'algorithme à utiliser des directions diagonales pour traverser les pixels. Une propagation de l'algorithme de « distance-coût » a également été testée en considérant les valeurs de porosité. Des valeurs de coûts sont calculées sous ArcGIS en prenant l'inverse de la porosité. Ces valeurs sont utilisées en tant que

surface de friction lors du lancement de l'algorithme. La surface de distance coût en sortie montrent des valeurs aberrantes. Elles sont dues au calcul des coûts qui n'est pas adéquat lorsque la porosité est faible. Par ailleurs, la différence avec la surface calculée sans tenir compte des porosités est faible, voire négligeable. Ces résultats ne sont pas significatifs compte tenu des généralisations du modèle. Il faudrait le tester sur des données réelles avec une surface de friction plus proche de la réalité pour voir ce qu'il en est.

Troisièmement, une adaptation du modèle à des résolutions différentes a été sollicitée. Elle permettrait de traiter d'importants jeux de données sans augmenter considérablement les coûts en termes de temps et de technologie. Une première méthodologie a été proposée sous IDRISI. Elle consiste à agréger les pixels entre eux selon un facteur commun en x et en y. La valeur moyenne des pixels est assignée à la nouvelle maille résultant de leur agrégation. L'agrégation a été effectuée directement sur l'image reprenant les valeurs de distance coût calculées précédemment. Les résultats correspondent à une généralisation des valeurs de distance coût à un maillage de plus faible résolution. Cependant, ils ne répondent pas à la demande des ingénieurs du HECE. Une seconde méthodologie est conçue, toujours sous IDRISI, pour déterminer des coefficients de propagation en sortie des mailles grossières. Pour ce faire, nous partons de la surface de friction. Celle-ci est découpée en mailles grossières. Pour chaque maille, les côtés sont considérés à tour de rôle comme source de propagation. L'algorithme de « distance-coût » est lancé pour chaque côté, générant une surface de distance coût par côté. Pour chaque surface de distance coût, les moyennes des côtés sont calculées et extraites en considérant la ligne ou la colonne extrême de pixels. Ces moyennes font office de coefficients de propagation. En observant ces coefficients, nous pouvons à priori déterminer le sens principal de l'écoulement.

Enfin, ce mémoire propose une méthode pour extraire les axes centraux des voiries. Elle se base notamment sur les polygones de Thiessen et est adaptée à partir de QGIS. Les axes résultants approximent bien les centres de voirie lorsque les paramètres choisis sont adéquats. Cependant, les intersections des axes ne sont pas toujours situées au milieu des carrefours.

Ces méthodes ne sont que des exemples d'utilisation des SIG dans le domaine des inondations. Elles pourraient être améliorées en considérant des données plus proches de la réalité. A titre d'exemple, les régions urbaines sont souvent pourvues d'un système de drainage complexe en surface (rigoles) et en sous-sol (égouts). Sa prise en compte semble essentielle pour simuler une inondation en ville plus réaliste.

Par ailleurs, nous pourrions enrichir les surfaces de coût en définissant une friction anisotrope en plus des obstacles. Par exemple, il est possible d'ajouter une pente au modèle. La friction pourrait aussi être adaptée en fonction de l'affectation du sol (pelouse, béton,...) et ne pas limiter celle-ci uniquement aux types « voirie » et « bâti ».

De même, le modèle présenté ici est en 2D mais nous pourrions adapter les méthodes à des modèles tridimensionnels. Dans cette optique, des obstacles de hauteurs  $Z$  différentes pourraient être considérés. Il serait alors question d'étudier la propagation à différentes altitudes. Pour ce faire, le modèle pourrait être simplement découpé en « tranches ». Ces tranches seraient être définies à un pas de  $Z$  constant, ou encore à des  $Z$  qui correspondent aux hauteurs préalablement triées des obstacles.

Une approche possible pour un futur travail serait l'utilisation des *quadtree*. H. Samet (1990) décrit le principe des *quadtree* comme une décomposition récursive de l'espace, permettant de structurer les données de façon hiérarchique. L'utilisation d'une telle structuration permettrait, par exemple, d'implémenter le modèle en considérant des obstacles de hauteurs différentes ou encore d'adapter la résolution spatiale, rendant ainsi l'implémentation plus efficace pour de larges régions (Bernardin *et al.*, 2006). Les discrétisations par *quadtree* sont déjà utilisées pour représenter des phénomènes continus comme les tsunamis (Popinet, 2012), les tremblements de terre et les fissures (Ooi *et al.*, 2015). Les *quadtree* ont d'ailleurs été utilisés pour étudier les inondations, notamment par Q. Liang. Il les a employés pour prédire les hauteurs d'eau (Liang *et al.*, 2008) et pour simuler des inondations faisant suite à une rupture de digue (Liang *et al.*, 2004) et à une pluie torrentielle (Liang & Borthwick, 2009).

# Références

---

- Agoston M. (2005). Computer graphics and geometric modeling (1<sup>st</sup> ed.). Springer-Verlag, London. <http://doi.org/10.1007/b138805>.
- Araud Q. (2012). Simulations des écoulements en milieu urbain lors d'un évènement pluvieux extrême. Thèse de doctorat. Université de Strasbourg.
- Ballais J., Bonté M. P., Masson M. M., Garry G., Kehia H., Eyraud C. & Ghram A. (2004). L'évolution du risque d'inondation par les cours d'eau méditerranéens français : le rôle de l'accumulation dans le lit majeur. *Bulletin de l'Association des géographes français : Proximités spatiales dans la ville. Risques naturels, risques de société, 1* ; pp. 64-74. <http://doi.org/10.3406/bagf.2004.2366>.
- Ballais J., Chave S., Dupont N., Masson E. & Penven M.-J. (2011). La méthode hydrogéomorphologique de détermination des zones inondables. *PHYSIO-GÉO Géographie Physique et Environnement (ISSN 1958-573X), Collection "Ouvrages"*.
- Ballais J., Garry G. & Masson M. (2005). Contribution de l'hydrogéomorphologie à l'évaluation du risque d'inondation : le cas du Midi méditerranéen français. *Géosciences de surface (Hydrologie-Hydrogéologie)*, 337, pp. 1120-1130. <http://doi.org/10.1016/j.crte.2005.06.010>.
- Bernardin T., Cowgill E., Gold R., Hamann B., Kreylos O. & Schmitt A. (2006). Interactive mapping on 3-D terrain models. *Geochemistry, Geophysics, Geosystems*, 7(10), pp. 1-12. <http://doi.org/10.1029/2006GC001335>.
- Boots B.N. (1980). Weighting Thiessen Polygons. *Economic Geography*, 56(3), pp. 248-259. URL : <http://www.jstor.org/stable/142716>.
- Bresenham J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1), pp. 25-30. <http://doi.org/10.1147/sj.41.0025>.
- Cabral S., Campos J., Silveira C. & Teixeira F. (2014). Hydrologic and Hydraulic Modelling Integrated With Gis: a Study of the Acarau River Basin. *Journal of Urban and Environmental Engineering*, 8(2), 167-174. <http://doi.org/10.4090/juee.2014.v8n2.167174>.
- Cea M. & Rodriguez M. (2016). Two-Dimensional Coupled Distributed Hydrologic-Hydraulic Model Simulation on Watershed. *Pure and Applied Geophysics*, 173(3), pp. 909-922. <http://doi.org/10.1007/s00024-015-1196-5>.
- Chang L. & Huang S. (2015). Assessing urban flooding vulnerability with an emergy approach. *Landscape and Urban Planning*, 143, pp. 11-24. <http://doi.org/10.1016/j.landurbplan.2015.06.004>.



- Coveney S. & Fotheringham A. S. (2011). The impact of DEM data source on prediction of flooding and erosion risk due to sea-level rise. *International Journal of Geographical Information Science*, 25(7), pp. 1191-1211. <http://doi.org/10.1080/13658816.2010.545064>.
- Debaucheron M. (2014). Analyse d'inondations et leurs conséquences en milieu urbain. Travail de fin d'études en Master Ingénieur Civil des Constructions. Université de Liège.
- Demers S., Olsen T., Buffin-bélangier T. & Biron P. M. (2014). L'hydrogéomorphologie appliquée à la gestion de l'aléa d'inondation en climat tempéré froid : l'exemple de la rivière Matane (Québec). *Physio-Géo*, 8. <http://doi.org/10.4000/physio-geo.3813>.
- Diakakis M., Deligiannakis G., Pallikarakis A. & Skordoulis M. (2016). Factors controlling the spatial distribution of flash flooding in the complex environment of a metropolitan urban area. The case of Athens 2013 flash flood event. *International Journal of Disaster Risk Reduction*, 18, pp. 171-180. <http://doi.org/10.1016/j.ijdrr.2016.06.010>.
- Donnay J.-P. (2012). Cartographie & SIG. Cours de 2<sup>ème</sup> bachelier en sciences géographiques, orientation générale à l'Université de Liège, 521 pp.
- Dottori F. & Todini E. (2013). Testing a simple 2D hydraulic model in an urban flood experiment. *Hydrological Processes*, 27(9), pp. 1301-1320. <http://doi.org/10.1002/hyp.9370>.
- Encyclo-ecolo (2011). Hydromorphologie. URL: <http://www.encyclo-ecolo.com/Hydromorphologie>.
- GIEC (2013). Résumé à l'intention des décideurs, Changements climatiques: Les éléments scientifiques. Contribution du Groupe de travail I au cinquième Rapport d'évaluation du Groupe d'experts intergouvernemental sur l'évolution du climat [sous la direction de Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S. K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex et P.M. Midgley]. Cambridge University Press, Cambridge, Royaume-Uni et New York (État de New York), États-Unis d'Amérique.
- Gomaresco M.A. (2015). Basics of Geomatics. *National Research Council of Italy, Institute for the Electromagnetic Sensing of the Environment*, Milano, Italy. <http://doi.org/10.1007/978-1-4020-9014-1>.
- Gralepois M. & Guevara S. (2016). L'adaptation aux risques d'inondation façonnée par les métiers de la ville. *Développement Durable et territoires, Réseau Développement durable et territoires fragiles*, 6 (3).
- Guinot V. (2003). Les modèles numériques en hydrologie et en hydraulique. Université Montpellier 2. URL : <http://perso.ens-lyon.fr/nathalie.revol/ED/VGuinot.pdf>.
- Han D. & Bray M. (2006). Automated Thiessen polygon generation. *Water Resource Research*, 42. <http://doi.org/10.1029/2005WR004365>.

- Herubel A., Biri V. & Deverly S. (2011). Morphological Antialiasing and Topological Reconstruction. *International Conference on Computer Graphics Theory and Applications (GRAPP'11)*, Juin 2011, Vilamoura, Portugal. pp. 187–192.
- Ifremer environnement (2010). Hydromorphologie. URL : <http://envlit.ifremer.fr/infos/glossaire/h/hydromorphologie>.
- Izquierdo J., Pérez R. & Iglesias L. (2004). Mathematical Models and Methods in the Water Industry. *Mathematical and Computer Modelling*, 39, pp. 1353–1374. <http://doi.org/10.1016/j.mcm.2004.06.012>.
- Karimipour F., Ghandehari M. & Ledoux H. (2013). Watershed delineation from the medial axis of river networks. *Computers and Geosciences*, 59, pp. 132–147. <http://doi.org/10.1016/j.cageo.2013.06.004>.
- Kim J., Kuwahara Y. & Kumar M. (2011). A DEM-based evaluation of potential flood risk to enhance decision support system for safe evacuation. *Natural Hazards and Earth System Science*, 59, pp. 1561–1572. <http://doi.org/10.1007/s11069-011-9852-2>.
- Leitão J. P., Boonia-aroonet S., Prodanovi D. & Maksimovi Č. (2008). Influence of DEM resolution on surface flow network for pluvial urban flooding and simulations of integrated system. *11th International Conference on Urban Drainage*, Edinburgh, Scotland, UK.
- Liang Q. & Borthwick A. G. L. (2009). Adaptive quadtree simulation of shallow flows with wet–dry fronts over complex topography. *Computers & Fluids*, 38(2), pp. 221–234. <http://doi.org/10.1016/j.compfluid.2008.02.008>.
- Liang Q., Borthwick A. G. L. & Stelling, G. (2004). Simulation of dam- and dyke-break hydrodynamics on dynamically adaptive quadtree grids. *International Journal for Numerical Methods in Fluids*, 46(2), pp. 127–162. <http://doi.org/10.1002/flid.748>.
- Liang Q., Du G., Hall J. W. & Borthwick A. G. (2008). Flood Inundation Modeling with an Adaptive Quadtree Grid Shallow Water Equation Solver. *Journal of Hydraulic Engineering*, 134(11), pp. 1603–1610. [http://doi.org/10.1061/\(ASCE\)0733-9429\(2008\)134:11\(1603\)](http://doi.org/10.1061/(ASCE)0733-9429(2008)134:11(1603)).
- Montané A., Vinet F., Buffin-béranger T., Vento O. & Masson, M. (2015). Cartographie hydrogéomorphologique : émergence d'utilisations réglementaires en France. *Physio-Géo*, 9. <http://doi.org/10.4000/physio-geo.4397>.
- O'Callaghan J. F. & Mark D. M. (1984). The extraction of drainage networks from digital elevation data. *Computer Vision, Graphics, and Image Processing*, 28, pp. 323–344. [http://doi.org/10.1016/S0734-189X\(84\)80047-X](http://doi.org/10.1016/S0734-189X(84)80047-X).
- Ooi E. T., Man H., Natarajan S. & Song C. (2015). Adaptation of Quadtree Meshes in the Scaled Boundary Finite Element Method for Crack Propagation Modelling. *Engineering Fracture Mechanics*, 144, pp. 101–117. <http://doi.org/10.1016/j.engfracmech.2015.06.083>.

- Ozdemir H. & Bird D. (2009). Evaluation of morphometric parameters of drainage networks derived from topographic maps and DEM in point of floods. *Environmental Geology*, 56, pp. 1405–1415. <http://doi.org/10.1007/s00254-008-1235-y>.
- Popinet S. (2012). Adaptive modelling of long-distance wave propagation and fine-scale flooding during the Tohoku tsunami. *Natural Hazards and Earth System Science*, 12, pp. 1213–1227. <http://doi.org/10.5194/nhess-12-1213-2012>.
- Salomon D. (2011). The computer Graphics Manual (Gries D. & Schneider F.B., Eds.). *Texts in Computer Science*, 2. Springer-Verlag, London. <http://doi.org/10.1007/978-0-85729-886-7>.
- Samet, H. (1990). The Design and Analysis of Spatial Data Structures (Harrison M.A., Ed.). Addison-Wesley Publishing Company. <http://doi.org/10.1080/02693799108927847>.
- Schreider S.Y., Smith, D. I. & Jakeman A. J. (2000). Climate change impacts on urban flooding. *Climate Change*, 47, pp. 91–115.
- Sene K. (2013). Flash Floods: Forecasting and warning. Springer-Verlag, London. <http://doi.org/10.1007/978-94-007-5164-4>.
- Shen D., Wang J., Cheng X., Rui Y. & Ye S. (2015). Integration of 2-D hydraulic model and high-resolution lidar-derived DEM for floodplain flow modeling. *Hydrol. Earth Syst. Sci.*, 19, pp. 3605–3616. <http://doi.org/10.5194/hess-19-3605-2015>.
- Société Suisse de Géomorphologie (SSGm) (2009). Qu'est-ce que la géomorphologie? Fiches pour l'enseignant - Chapitre 1 : Introduction. URL : <https://www.unifr.ch/geoscience/geographie/ssgmfiches/intro/1101.php>.
- Stuart N. & Stocks C. (1993). Hydrological modelling within GIS: an integrated approach. *HydroGIS 93: Application of GIS in hydrology and water resources*. Avril 1993, Vienne, Autriche. pp. 319-329.
- Swanson, S. E. (2001). GIS: Geographic Information Systems. *Journal of Hospital Librarianship*, 1(3), pp. 83–89. <http://doi.org/10.1300/J186v01n03>.
- Tatalovich Z., Wilson J. P. & Cockburn M. (2006). A Comparison of Thiessen Polygon, Kriging, and Spline Models of Potential UV Exposure. *Cartography and Geographic Information Science*, 33(3), pp. 217–231. <http://doi.org/10.1559/152304006779077318>.
- Zazo S., Molina J. & Rodríguez-gonzález P. (2015). Analysis of flood modeling through innovative geomatic methods. *Journal of Hydrology*, 524, pp. 522–537. <http://doi.org/10.1016/j.jhydrol.2015.03.011>.

# Annexes

---

Polygone	x	y
A1	0.000	4.997
A1	0.460	4.995
A1	0.459	4.570
A1	0.000	4.602
A2	0.000	4.550
A2	0.460	4.513
A2	0.459	4.089
A2	0.000	4.120
A3	0.000	4.071
A3	0.459	4.038
A3	0.459	3.550
A3	0.000	3.586
A4	0.000	3.536
A4	0.459	3.503
A4	0.459	2.954
A4	0.000	2.951
A5	0.000	2.836
A5	0.459	2.836
A5	0.459	2.134
A5	0.000	2.136
A6	0.000	2.086
A6	0.459	2.085
A6	0.459	1.376
A6	0.000	1.376
A7	0.000	1.326
A7	0.459	1.325
A7	0.459	0.655
A7	0.000	0.655
A8	0.000	0.604
A8	0.459	0.606
A8	0.459	-0.002
A8	0.000	-0.003
B1	0.979	4.524
B1	0.509	4.559
B1	0.510	4.994
B1	0.982	4.993
B2	0.979	4.050
B2	0.510	4.084
B2	0.510	4.505
B2	0.982	4.474
B3	0.981	3.514
B3	0.508	3.549
B3	0.510	4.033
B3	0.980	3.998

Polygone	x	y
B4	0.981	2.955
B4	0.508	2.952
B4	0.511	3.495
B4	0.981	3.464
B5	0.984	2.137
B5	0.515	2.134
B5	0.513	2.836
B5	0.983	2.836
B6	0.983	1.376
B6	0.513	1.375
B6	0.513	2.083
B6	0.984	2.084
B7	0.981	0.655
B7	0.511	0.656
B7	0.511	1.326
B7	0.981	1.324
B8	0.982	-0.001
B8	0.511	-0.001
B8	0.510	0.604
B8	0.983	0.605
C1	1.483	4.994
C1	1.483	4.485
C1	1.031	4.522
C1	1.031	4.994
C2	1.030	4.468
C2	1.481	4.441
C2	1.483	4.017
C2	1.031	4.044
C3	1.031	3.993
C3	1.484	3.959
C3	1.484	3.474
C3	1.032	3.508
C4	1.032	3.460
C4	1.482	3.421
C4	1.484	2.952
C4	1.031	2.956
C5	1.034	2.835
C5	1.486	2.835
C5	1.486	2.135
C5	1.034	2.135
C6	1.034	2.086
C6	1.485	2.084
C6	1.485	1.375
C6	1.032	1.376

Polygone	x	y
C7	1.034	1.324
C7	1.486	1.324
C7	1.486	0.655
C7	1.033	0.655
C8	1.031	0.606
C8	1.485	0.604
C8	1.487	-0.002
C8	1.033	-0.001
D1	1.959	4.479
D1	1.605	4.480
D1	1.607	4.993
D1	1.888	4.993
D2	2.025	4.009
D2	1.604	4.009
D2	1.606	4.432
D2	1.967	4.432
D3	2.104	3.464
D3	1.606	3.462
D3	1.606	3.957
D3	2.032	3.958
D4	2.175	2.953
D4	1.606	2.953
D4	1.606	3.409
D4	2.111	3.408
D5	2.211	2.120
D5	1.608	2.135
D5	1.609	2.836
D5	2.191	2.833
D6	2.237	1.350
D6	1.613	1.377
D6	1.611	2.080
D6	2.218	2.071
D7	2.261	0.591
D7	1.613	0.655
D7	1.614	1.324
D7	2.245	1.298
D8	2.277	-0.001
D8	1.609	-0.001
D8	1.609	0.604
D8	2.261	0.540
E1	2.150	4.990
E1	2.293	4.492
E1	2.021	4.478
E1	1.945	4.989

Polygone	x	y
E2	2.029	4.432

Polygone	x	y
F5	3.457	2.104

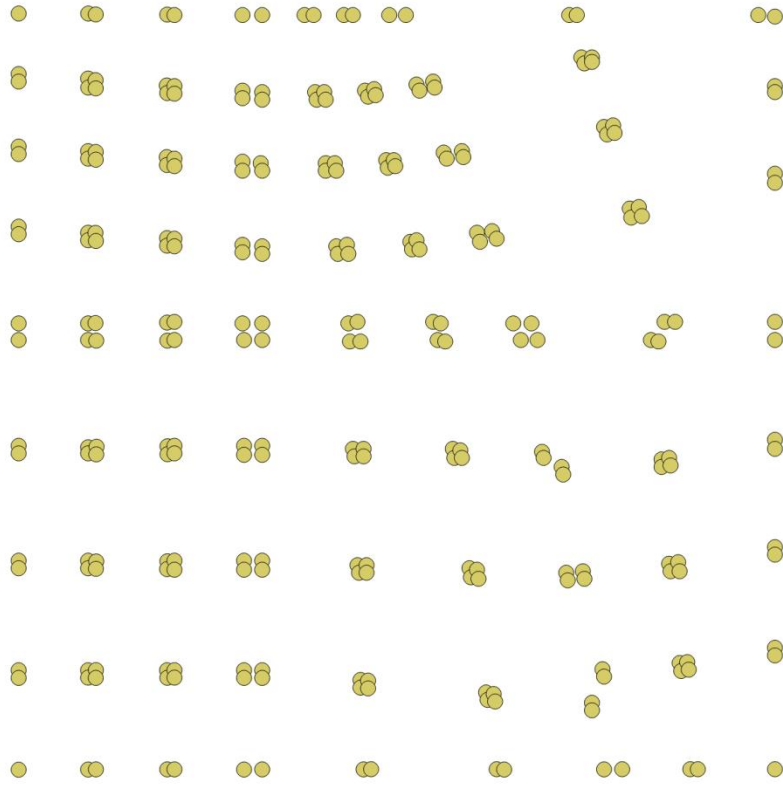
Polygone	x	y
G8	3.869	0.613

E2	2.305	4.448
E2	2.427	4.028
E2	2.085	4.009
E3	2.096	3.962
E3	2.442	3.982
E3	2.586	3.486
E3	2.166	3.466
E4	2.176	3.411
E4	2.602	3.437
E4	2.735	2.955
E4	2.236	2.955
E5	2.257	2.833
E5	2.769	2.837
E5	2.871	2.115
E5	2.276	2.122
E6	2.275	2.068
E6	2.876	2.064
E6	2.983	1.325
E6	2.295	1.348
E7	2.295	1.295
E7	2.987	1.269
E7	3.091	0.508
E7	2.313	0.586
E8	2.314	0.535
E8	3.099	0.459
E8	3.161	0.000
E8	2.328	0.000
F1	2.627	4.534
F1	2.351	4.496
F1	2.209	4.990
F1	2.445	4.990
F2	2.813	4.075
F2	2.477	4.034
F2	2.361	4.455
F2	2.646	4.493
F3	3.031	3.547
F3	2.632	3.496
F3	2.490	3.991
F3	2.833	4.039
F4	3.270	2.954
F4	2.785	2.953
F4	2.652	3.438
F4	3.052	3.490

F5	2.919	2.108
F5	2.819	2.834
F5	3.315	2.835
F6	3.617	1.302
F6	3.029	1.317
F6	2.927	2.060
F6	3.468	2.059
F7	3.786	0.440
F7	3.144	0.498
F7	3.038	1.262
F7	3.630	1.247
F8	3.871	-0.001
F8	3.210	-0.002
F8	3.148	0.448
F8	3.793	0.390
G1	3.638	4.993
G1	3.724	4.707
G1	2.737	4.551
G1	2.555	4.988
G2	3.735	4.670
G2	3.871	4.249
G2	2.925	4.091
G2	2.750	4.507
G3	3.886	4.195
G3	4.035	3.705
G3	3.133	3.564
G3	2.937	4.052
G4	3.158	3.506
G4	4.054	3.647
G4	4.267	2.957
G4	3.387	2.951
G5	3.430	2.836
G5	4.177	2.839
G5	4.245	2.050
G5	3.591	1.996
G6	3.604	1.950
G6	4.251	2.003
G6	4.304	1.358
G6	3.727	1.312
G7	3.742	1.264
G7	4.313	1.310
G7	4.371	0.703
G7	3.859	0.662

G8	4.376	0.650
G8	4.435	0.000
G8	3.989	0.000
H1	4.891	4.993
H1	3.786	4.711
H1	3.694	4.993
H2	5.000	4.516
H2	3.929	4.258
H2	3.792	4.679
H2	5.000	4.976
H3	5.000	3.938
H3	4.100	3.722
H3	3.943	4.211
H3	5.000	4.475
H4	5.000	2.957
H4	4.337	2.957
H4	4.117	3.658
H4	5.000	3.882
H5	5.000	2.177
H5	4.297	2.059
H5	4.227	2.834
H5	5.000	2.838
H6	5.000	1.469
H6	4.362	1.365
H6	4.307	2.010
H6	5.000	2.124
H7	5.000	0.804
H7	4.421	0.709
H7	4.367	1.313
H7	5.000	1.418
H8	5.002	0.000
H8	4.490	0.000
H8	4.428	0.659
H8	5.002	0.753

**Annexe 1 - Coordonnées des sommets des îlots**



**Annexe 2 - Sommets des polygones importés en couche *Shapefile* vectorielle**

```

##modele_final=name

##VECTORLAYER_ILOTS=vector

##intersect=output vector

##area=output vector

##surfaceilots=output vector

##porosite=output vector

##idinc=output vector

##jointure=output vector

##grille=output vector

outputs_QGISCREATEGRID_2=processing.runalg('qgis:creategrid',
1,['0,5,0,5'],0.01,0.01,'EPSG:31370',grille)

outputs_QGISADDAUTOINCREMENTALFIELD_1=processing.runalg('qgis:addautoincrementalfiel
d', outputs_QGISCREATEGRID_2['OUTPUT'],idinc)

outputs_QGISINTERSECTION_1=processing.runalg('qgis:intersection',
VECTORLAYER_ILOTS,outputs_QGISADDAUTOINCREMENTALFIELD_1['OUTPUT'],intersect)

outputs_QGISFIELDCALCULATOR_1=processing.runalg('qgis:fieldcalculator',
outputs_QGISINTERSECTION_1['OUTPUT'],'surf_ilot',0,10.0,6.0,True,'($area)*10000',area)

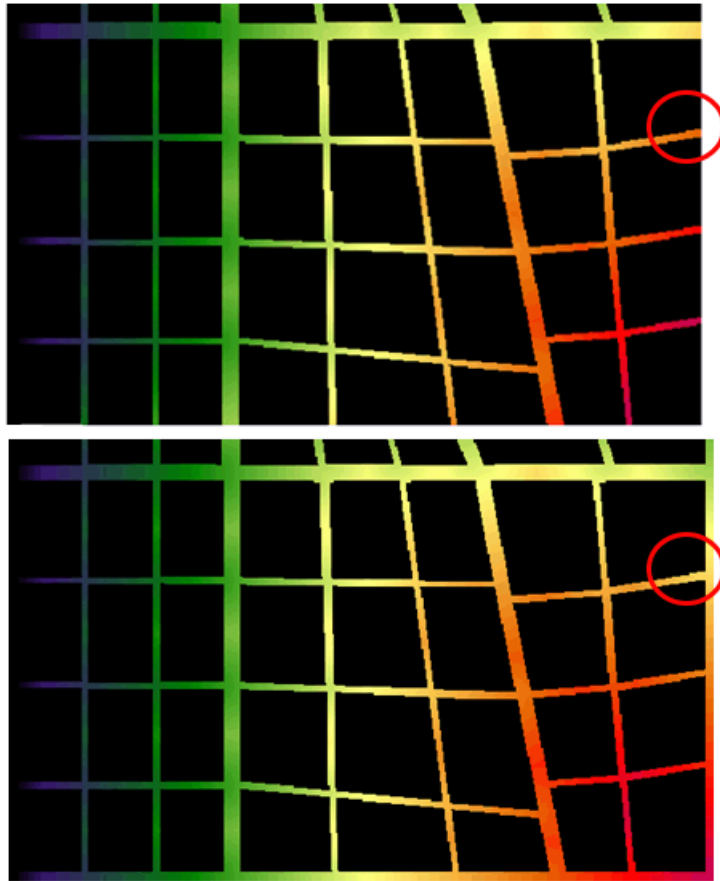
outputs_QGISJOINATTRIBUTESTABLE_1=processing.runalg('qgis:joinattributestable',
outputs_QGISADDAUTOINCREMENTALFIELD_1['OUTPUT'],outputs_QGISFIELDCALCULATOR_1[
'OUTPUT_LAYER'],'AUTO','AUTO',jointure)

outputs_QGISFIELDCALCULATOR_2=processing.runalg('qgis:fieldcalculator',
outputs_QGISJOINATTRIBUTESTABLE_1['OUTPUT_LAYER'],'surf_ilot',0,10.0,6.0,False,'coalesce(s
urf_ilot,0)',surfaceilots)

outputs_QGISFIELDCALCULATOR_3=processing.runalg('qgis:fieldcalculator',
outputs_QGISFIELDCALCULATOR_2['OUTPUT_LAYER'],'surf_rue',0,10.0,3.0,True,'1-
surf_ilot',porosite)

```

### Annexe 3 - Détermination de la porosité sous QGIS



Annexe 4 - Visualisation de l'effet d'un bord dans une surface de coûts sous IDRISI



```

# -*- coding: utf-8 -*-
# -----
# script_modele_dist.py
# Created on: 2016-08-22 10:36:01.00000
# (generated by ArcGIS/ModelBuilder)
# Description:
# -----

# Import arcpy module
import arcpy

# Local variables:
porosite5 = "porosite5"
porosite5_4_ = porosite5
porosite5_3_ = porosite5_4_
porosite_-_Copie = "porosite - Copie"
porosite_-_Copie_2_ = porosite_-_Copie
porosite2_3_ = porosite_-_Copie_2_
porosite2_4_ = porosite2_3_
porosite2_5_ = porosite2_4_
poro_to_rst = "C:\\Users\\Céline\\Documents\\memoire\\modeleur_arcgis_cout\\poro_to_rst"
por_rst_clip2 = "C:\\Users\\Céline\\Documents\\memoire\\modeleur_arcgis_cout\\por_rst_clip2"
cout_direct2 = "C:\\Users\\Céline\\Documents\\memoire\\modeleur_arcgis_cout\\cout_direct2"
cout_dis_ar2 = "C:\\Users\\Céline\\Documents\\memoire\\modeleur_arcgis_cout\\cout_dis_ar2"

# Process: Select Layer By Attribute (2)
arcpy.SelectLayerByAttribute_management(porosite5, "SWITCH_SELECTION", "( top=4.98 AND
bottom=4.99 ) OR ( right=0.02 AND left=0.01 ) AND NOT ( right=0.01 OR bottom=5 ) ")

# Process: Delete Features (2)
arcpy.DeleteFeatures_management(porosite5_4_)

# Process: Select Layer By Attribute
arcpy.SelectLayerByAttribute_management(porosite_-_Copie, "NEW_SELECTION", "\"surf_rue\"=0")

# Process: Delete Features
arcpy.DeleteFeatures_management(porosite_-_Copie_2_)

# Process: Add Field
arcpy.AddField_management(porosite2_3_, "cout", "DOUBLE", "10", "3", "", "", "NULLABLE",
"NON_REQUIRED", "")

# Process: Calculate Field
arcpy.CalculateField_management(porosite2_4_, "cout", "1/ [surf_rue]", "VB", "")

# Process: Polygon to Raster
arcpy.PolygonToRaster_conversion(porosite2_5_, "cout", poro_to_rst, "CELL_CENTER", "cout", "0.01")

# Process: Clip
arcpy.Clip_management(poro_to_rst, "0.01 0.01 4.99 4.99", por_rst_clip2, "", "", "NONE",
"NO_MAINTAIN_EXTENT")

# Process: Cost Distance
arcpy.gp.CostDistance_sa(porosite5_3_, por_rst_clip2, cout_dis_ar2, "", cout_direct2)

```

