



MASTER 1

Observation de la Terre et Géomatique

Travail d'Étude et de Recherche

2024-2025

SERY Ange

Sujet du TER : Traitement et intégration de données réseaux multi-sources pour le calcul d'itinéraires.

Encadrant : Paul Salze
Laboratoire Image, Ville, Environnement
UMR7362 CNRS-Unistra
3 rue de l'Argonne F-67000 Strasbourg

XXX

Traitemet et intégration de données réseaux multi-sources pour le calcul d'itinéraires

Ange SERY

Université de Strasbourg Master 1

Observation de la Terre et Géomatique (OTG)

Mémoire de TER soutenu le 02 juillet 2025 devant le jury composé de : M. Paul Salze, encadrant pédagogique, M. Olivier Finance et Mme Nadège Blond.

Remerciement

Je tiens à exprimer ma profonde gratitude à mon encadrant, Monsieur Paul Salze, pour son accompagnement, sa disponibilité et la qualité de ses conseils tout au long de ce travail de recherche.

Tables des matières

Remerciement.....	3
Tables des matières.....	4
Introduction	6
Chapitre I – Contexte et problématique de l'étude	8
I.1. Le modèle ABSUM.....	9
I.2. Représentation et qualités des réseaux routiers pour les modèles de simulation de la mobilité.....	10
I.2.1. Structure et exigences des modèles de simulation	10
I.2.2. Qualités des données et compromis entre précision et performance	10
I.3. Méthodes de fusion de données vectorielles multi-sources	11
I.4. Conclusion	13
Chapitre II – Identification et automatisation de la correction des erreurs (topologiques et attributaires) de la BD Carto	14
II.1. Démarche Méthodologique	15
II.2. Résultats.....	18
II.2.1. Identification des erreurs topologiques et attributaires.....	18
II.2.2. Correction des erreurs topologiques et attributaires	29
II.2.2.1. Vitesses de circulations non renseignées : nulle	29
II.2.2.2. Connexité du réseau.....	30
II.2.3. Conclusion	31
Chapitre III – Intégration de la BD Topo aux données de la BD Carto et évaluation du réseau intégré.....	32
III.1. Intégration de la BD Topo	33
III.1.1. Identification des tronçons de routes à intégrer	33
III.1.2. Intégration des données et nettoyage du réseau fusionné.....	34
III.1.3. Remodelage de l'ensemble de données fusionnées.....	35

III.2. Calcul des itinéraires	37
III.3. Evaluation du réseau fusionné.....	39
III.3.1. Comparaison du nombre d'entités et du temps de calcul des itinéraires entre BD Carto et le réseau fusionné	39
III.3.2. Comparaison des coûts (temps et distances) de trajets entre la BD Carto et le réseau fusionné	39
III.4. Conclusion.....	44
Conclusion générale	45
Bibliographie.....	47
Annexe	49

Introduction

Aujourd’hui, la mobilité urbaine est un enjeu majeur pour l’aménagement du territoire, la gestion des infrastructures de transport. Avec l’augmentation des flux, la diversification des modes de déplacement et les exigences croissantes en matière de planification, les outils de simulation spatiale et de calcul d’itinéraires sont essentiels. Les modèles de simulation à base d’agents sont particulièrement utiles pour représenter de manière détaillée et individuelle les comportements de déplacement dans une zone donnée.

C’est dans ce contexte que le modèle ABSUM (Agent-Based Simulation of Urban Mobility) a été développé par le laboratoire Image, Ville, Environnement (LIVE) dans le cadre du projet Interreg SuMo-Rhine (2018-2021). Ce modèle a pour objectif de simuler les déplacements domicile-travail de la population active en tenant compte des différents modes de transport disponibles. Pour fonctionner, un réseau routier structuré, connecté et enrichi d’informations attributaires (vitesse, type de voie, direction de circulation) est indispensable.

Deux évaluations du modèle ont été réalisées, en 2022 et 2024, dans le cadre d’une application sur l’Eurométropole de Strasbourg (Tignon, 2022 ; Dieffenbacher H., 2024). La première (2022) a révélé des écarts entre les simulations de trafic routier et les données observées sur le territoire étudié, nécessitant un approfondissement du modèle, réalisé en 2024. La seconde évaluation (2024) a mis en évidence plusieurs erreurs dans la génération d’itinéraire, notamment des trajets discontinus dans le réseau original (BD Topo, IGN) et des trajets non calculés dans le réseau simplifié par l’un des modules du modèle (module RNWM). De plus, l’intégration de la BD Carto dans le modèle n’a pas présenté d’itinéraires discontinus, mais des problèmes liés à la simplification du réseau.

Le travail que nous allons mener dans le cadre de ce TER, s’inscrit en partie dans ce cadre et a pour but de mettre en place un méthodologie reproductible et automatisé, qui permettra d’identifier et corriger les erreurs (topologiques et attributaires) et d’intégrer les deux bases de données réseaux (BD Topo et BD Carto) afin de construire un jeu de données adapté aux besoins du modèle ABSUM.

La première partie de ce travail est consacrée à une présentation du modèle ABSUM et à une revue de la littérature sur la qualité des réseaux routier pour les modèles de simulation de la mobilité, ainsi que les méthodes d’intégration ou de fusion des données réseaux.

La deuxième partie va se concentrer sur l'identification et la correction des erreurs attributaires et topologiques dans les données réseaux, qui empêchent le calcul d'itinéraires à une résolution spatiale fine nous porterons une analyse particulière sur le réseau routier de la BD Carto.

La troisième partie traitera de l'intégration des données Topo aux données de la BD Carto.

Chapitre I – Contexte et problématique de l'étude

Chapitre I – Contexte et problématique de l'étude	8
I.1. Le modèle ABSUM.....	9
I.2. Représentation et qualités des réseaux routiers pour les modèles de simulation de la mobilité.....	10
I.2.1. Structure et exigences des modèles de simulation	10
I.2.2. Qualités des données et compromis entre précision et performance	10
I.3. Méthodes de fusion de données vectorielles multi-sources	11
I.4. Conclusion	13

I.1. Le modèle ABSUM

Le modèle ABSUM (Agent-Based Simulation of Urban Mobility) est un modèle de simulation de la mobilité urbaines, développé au laboratoire Image, Ville, Environnement (Université de Strasbourg) dans le cadre du projet Interreg européen SuMo-Rhine (2018-2021). L'un des objectifs du projet était de concevoir un système d'indicateurs permettant de mesurer l'influence des transports transfrontaliers sur le développement durable dans la région du Rhin Supérieur, et en l'associant à un outil de simulation de la mobilité quotidienne (Blond *et al.*, 2022).

Développé sur la plateforme GAMA, un environnement de modélisation et de simulation open source à base d'agent (Taillandier *et al.*, 2019), le modèle ABSUM s'appuie sur des données ouvertes, telles que celles de l'IGN pour les réseaux et l'INSEE pour la population. Il a été conçu dans l'optique d'être transférable à d'autres territoires français (Dieffenbacher, 2024).

Le modèle ABSUM permet de simuler les déplacements (itinéraires) domicile – travail des populations active d'un territoire donnée selon différents modes de transports (marche, vélo, voiture et transport en commun). Cependant, dans sa version actuelle, le modèle ne prend pas en compte les individus en transit (i.e. qui transitent par le territoire), ni d'échanges (i.e. qui ont pour origine ou destination une commune hors du territoire).

Le modèle ABSUM se structure en 7 modules fonctionnels (Dieffenbacher, 2024) :

- RNWM (Road NetWork Management) : il génère un réseau routier nettoyé et simplifié à partir de la base de données TOPO de l'IGN.
- POGEN (Population Generator) : il permet de générer une population spatialisée à partir des données de l'INSEE, désagrégées sur une maille de 200 mètres de côté.
- ROCCGEN (Route Choise Set Generator) : il permet de générer l'ensemble des itinéraires possibles entre lieux de résidence et lieux de travail, pour chaque mode de transport.
- HATTRISIM (Habit-Trip Simulator) : ce module génère des habitudes de déplacement et simule les trajets domicile-travail des individus. C'est au sein de ce module qu'est implémenté l'algorithme d'affectation dynamique du trafic routier.
- MOFOSIM (Mobility Forecast Simulator) : il a pour objectif de simuler les changements de comportements qui pourraient être associés à des scénarios mis en place sur le territoire.

- EMM (Energy and Emission Module) : c'est le dernier module, il permet calculer les émissions (dioxydes d'azote, les particules fines, le dioxyde de carbone, le monoxyde de carbone, les composés organiques volatils et les particules de suies de carbone) et consommation d'énergie liées au trafic routier. Les calculs d'émissions sont réalisés sur les tronçons du réseau à partir des flux générés par le module HATTRISIM et les facteurs d'émissions des véhicules puis réagrégés si besoin sur les mailles utilisées par le module POPGEN.

I.2. Représentation et qualités des réseaux routiers pour les modèles de simulation de la mobilité

I.2.1. Structure et exigences des modèles de simulation

Les modèles à base d'agents reposent sur la simulation du comportement individuel d'entités autonomes – ici, les usagers de la route – qui interagissent avec leur environnement selon des règles. Leurs déplacements sont conditionnés par la structure spatiale du réseau de transport, modélisé sous forme de graphe orienté, où les nœuds représentent les intersections et les arcs, les tronçons de route (Gräbener, 2010). Pour être exploité correctement, ce graphe doit être topologiquement valide, structurellement cohérent (connecté, orienté) et enrichi de données attributaires (vitesse de circulation, sens de circulation, accessibilité, etc.). Des modèles comme MATSim ou METROPOLIS2 démontrent que la qualité du graphe routier conditionne directement les résultats des simulations : temps de trajets, choix modaux, ou le choix des itinéraires (Javaudin *et al.*, 2024). Dès lors, la construction du réseau constitue une étape fondamentale de la chaîne de modélisation.

I.2.2. Qualités des données et compromis entre précision et performance

Le choix des données sources pour représenter les réseaux routiers dans des modèles de simulations fait l'objet d'un compromis constant entre finesse géométrique et efficacité algorithmique. D'un côté, des bases de données géographiques comme la BD Topo de l'Institut national de l'information géographique et forestières (IGN), à précision métrique, fournissent une représentation fine détaillé des infrastructures routières, incluant les voies secondaires, les

giratoires, etc. D'un autre côté, des bases généralisées comme la BD Carto (précision décimétrique) offre une représentation simplifiée, adéquats pour les traitements massifs, mais au prix d'une perte de détail potentiellement problématique pour certains usages.

Dans leur analyse, Mustière et Devogele (2008) montrent que les données à détail fin, bien que précises, peuvent ralentir considérablement les processus de calcul d'itinéraires lorsqu'elles ne sont pas simplifiées. A l'inverse, des données trop simplifiées peuvent masquer des alternatives d'itinéraires ou introduire des biais dans la modélisation du comportement spatial des agents (individus).

En effet, ces réseaux routiers pour être utilisées dans des modèles de simulation de mobilité, doivent satisfaire plusieurs critères de qualité, dont les principales identifiées dans la littérature sont (Devogele, 2002 ; Jiang *et al.*, 2004) :

- La validité topologique : connectivité du graphe, absence de discontinuités ;
- La précision sémantique : cohérence et exhaustivité des attributs ;
- La précision géométrique : position, longueur et configurations des segments.

I.3. Méthodes de fusion de données vectorielles multi-sources

L'intégration de données provenant de sources hétérogènes est une problématique récurrente dans les sciences de l'information géographique (SIG), en particulier lorsqu'il s'agit de construire un réseau routier cohérent à partir de jeux de données présentant des niveaux de détail, des informations sémantiques et des structures topologiques divergents. L'intégration doit être entendu ici dans le sens très large de rendre explicite les relations existantes entre les données, permettant aux utilisateurs de données d'effectuer des analyses géographiques efficace en tenant compte de plusieurs points de vue (Mustière *et al.*, 2008).

Dans ce contexte, la fusion ou l'intégration de données géospatiales est définie comme le processus visant à identifier, appairer, puis fusionner les entités géométriques et attributaires issues de différentes sources (Daneshgar *et al.*, 2018 ; Zhang *et al.*, 2023). Zhang *et al.* (2016) définissent la fusion de données dans un environnement SIG, comme la combinaison de deux ensembles de données géospatiales pour en produire un troisième considéré comme meilleur ou optimisé que ceux initiaux. Pour les données réseaux, l'enjeu est d'établir une correspondance précise entre les tronçons de routes, en s'appuyant sur des critères de proximité

géométrique, orientation, forme, longueur et topologie locale. Plusieurs méthodes ont été développées pour automatiser ce processus.

Parmi les approches, Zhang *et al.* (2016) ont proposé une méthode automatisée basée sur l'algorithme DSO (Delimited Stroke Oriented), permettant d'identifier les correspondances entre segments de deux réseaux. Leur procédure inclut :

- L'appariement des segments de routes par similarité géométrique (orientation, longueur, proximité) ;
- La détection des segments à fusionner basé sur les objets non appariés ;
- La transformation des objets à appariés pour éliminer les incohérences géométriques ;
- La création de nœuds d'intersection et le transfert des attributs ;
- La détection et correction manuelle des erreurs résiduelles.

Cette méthode a été testée en Allemagne sur une superficie de 360 000 km², et a permis d'intégrer les voies piétonnes d'un réseau routier vers un autre, avec un taux de précision supérieur à 99 %. La méthodologie a également été testé avec les données de plusieurs pays européens.

De leur côté, Daneshgar *et al.* (2018) ont développé une approche basée sur un algorithme itératif (ModelBuilder) au sein de l'environnement ArcGIS, pour appairer les segments d'un réseau cible avec ceux d'un réseau sources selon des critères géométrique et attributaires (nom de rue, direction, emplacement). Leur méthode repose notamment sur le calcul du taux de recouvrement linéaire, et permet une pondération des données transférées. Testée dans le Maryland (Etats-Unis) sur un total de 586 segments de routes, elle a produit des résultats avec plus de 85 % des données appariés de manière fiable.

Une autre approche différente est celle de Smarzaro *et al.* (2021), qui ont cherché à intégrer spatialement des données issues de sources hétérogènes pour la création d'un réseau de transport multimodale urbain. La méthodologie mise en place repose sur un schéma conceptuel structuré autour d'un graphe orienté.

I.4. Conclusion

Cette revue de la littérature montre qu'il existe de nombreuses méthodes automatisées d'intégration ou de fusion de réseaux routiers de sources différentes. L'enjeu de ces méthodes réside dans la capacité à constituer un réseau unifié, cohérent, complet et exploitable pour la navigation multimodale selon les contextes d'utilisations de ces réseaux.

Chapitre II – Identification et automatisation de la correction des erreurs (topologiques et attributaires) de la BD Carto

Chapitre II – Identification et automatisation de la correction des erreurs (topologiques et attributaires) de la BD Carto	14
II.1. Démarche Méthodologique	15
II.2. Résultats.....	18
II.2.1. Identification des erreurs topologiques et attributaires.....	18
II.2.2. Correction des erreurs topologiques et attributaires	29
II.2.3. Conclusion	31

II.1. Démarche Méthodologique

Pour identifier et corriger automatiquement les erreurs topologiques et attributaires du réseau routier de la BD Carto, nous avons mis en place une méthodologie consistant à calculer des itinéraires entre des paires de points Origine – Destination (OD) représentatives des mailles de 200 mètres de côté de l'INSEE. Tous les traitements ont été réalisés à l'aide du logiciel QGIS (Version 3.40).

Pour obtenir les échantillons de points OD utilisés pour les calculs d'itinéraires, nous avons sélectionné les mailles contenant plus de 4 bâtiments et généré des centroïdes à l'intérieurs de celle-ci. Cela nous a permis d'obtenir 3 480 points OD sur 8 101, couvrant l'étendu de l'EMS (Figure 2). Chaque point est considéré comme une origine et une destination.

Nous avons choisi de sélectionner les mailles avec plus de 4 bâtis pour constituer notre échantillon de points OD, afin de représenter les zones à forte activité humaine, vu qu'il s'agit de modéliser des déplacements domicile-travail, tout en évitant aussi un échantillon beaucoup plus grand qui rallongerait les temps de calcul.

Pour les calculs d'itinéraires, nous nous sommes concentrés uniquement sur trajets les plus rapides. Etant donné que la BD Carto présente un niveau de détail beaucoup plus grossier que la BD Topo, qui décrit le paysage avec une précision métrique, il n'est pas pertinent d'utiliser ces données pour les itinéraires en mode actifs (marche, vélo), mais plutôt pour les trajets en mode motorisé.

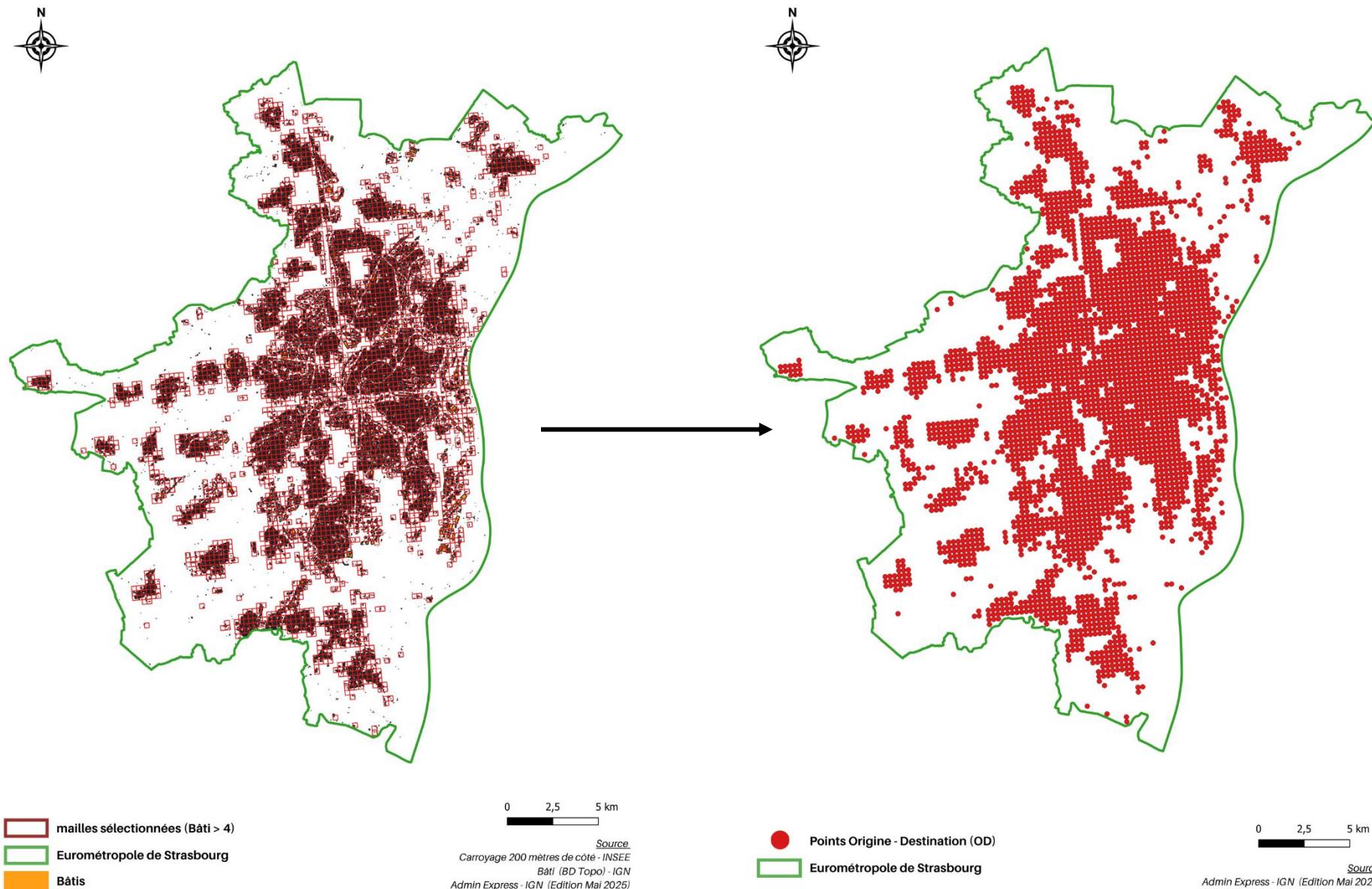


Figure 1 : Les mailles sélectionnées ($bâti > 4$) parmi toutes les mailles couvrantes l'étendue de l'EMS (à gauche) et les points OD générés à partir des mailles sélectionnées (à droite) - Ange SERY, 2025.

Un pré-traitement a été effectué sur la donnée pour définir l'accessibilité des routes en véhicule automobile. Nous avons exclu les sentiers, les tronçons qui ont un accès « physiquement impossible » aux véhicule léger ; et ceux sans sens de circulation. Ces choix ont été basés sur la documentation de la BD Carto.

Le code SQL appliqué dans la table attributaire pour filtrer des tronçons de routes accessibles en véhicule automobile est : `NOT ("NATURE" = 'Sentier' OR "ACCES_VL" = 'Physiquement impossible' OR "SENS" = 'Sans objet')`.

Définition des termes selon la documentation de la BD Carto Révision Avril 2024

Nature = « Sentier »

Voie étroite ne permettant pas le passage de véhicule automobiles et de largeur inférieure à 2 mètres.

Accès véhicule léger = « Physiquement impossible »

Cette valeur traduit l'impossibilité physique d'accéder aux tronçons, même pour les véhicules prioritaires (police, pompiers...).

Sens de circulation = « sans objet »

Valeur réservée aux tronçons de route de Nature = “Escalier”, “Sentier” et “Bac ou liaison maritime”.

Le premier calcul d'itinéraires, a été réalisé sur le réseau orienté, prenant en compte les sens de circulation. Les premiers résultats obtenus (cf. II.2) ont révélé que certains itinéraires n'étaient calculés. Nous avons d'abord supposé que cela était lié au sens de circulation, et avons testé un réseau non orienté, où le sens de circulation n'est pas pris en compte.

Cependant, les résultats obtenus avec le graphe non orienté (itinéraires non calculés), nous ont amené à inspecter manuellement le réseau pour identifier les sources du problème. Nous avons ensuite apporté des solutions et recalculé les itinéraires, en conservant les mêmes paramètres que précédemment, pour vérifier la résolution du non calcul d'itinéraires.

En suivant cette démarche, nous avons réalisé 3 essais de calcul d’itinéraires sur le réseau. En effet, chaque fois qu’un itinéraire n’est pas calculé, nous déterminons la ou les raisons puis apportons des corrections nécessaires avant de relancer le calcul d’itinéraires à l’essai suivant.

Le dernier essai (essai 3) nous a permis d’obtenir un taux de succès de 99,22 % pour les itinéraires calculés sur le réseau orienté et de 99,99 % sur le réseau non orienté.

Les calculs d’itinéraires ont été effectué à l’aide de l’outil OD Matrix du plugin QNEAT3 (Short for Qgis Network Analysis Toolbox 3 – Version 1.0.6), développé par Clemens Rafle.

II.2. Résultats

Les résultats portent d’une part sur les erreurs (topologiques et attributaires) identifiées dans le réseau routier de la BD Carto, d’autre part sur les corrections apportées.

II.2.1. Identification des erreurs topologiques et attributaires

Essai 1 : premier calcul d’itinéraires

Le premier calcul d’itinéraires s’est fait sur la base du réseau accessible en voiture. Les résultats ont révélé des itinéraires non calculés tant sur le réseau orienté qu’orienté.

- **Réseau orienté**

Avec le réseau orienté, nous avons enregistré 9 869 088 itinéraires calculés sur un total de 12 106 920¹, soit un pourcentage de 81,52 %. Les itinéraires non calculés sont de 2 237 832, soit un pourcentage de 18,48 %.

- **Réseau non orienté**

Avec le réseau non orienté, nous avons enregistré 10 007 742 itinéraires calculés, soit un pourcentage de 82,66 %. Les itinéraires non calculés s’élèvent au nombre de 2 099 178, soit un pourcentage de 17,34 %.

¹12 106 920 est le total des itinéraires possibles sur la base de notre échantillon de points OD : $(3\ 480)^2 = 12\ 110\ 400 - 3\ 480$ (correspondant aux itinéraires ayant une origine et une destination identique) = 12 106 920 itinéraires possibles.

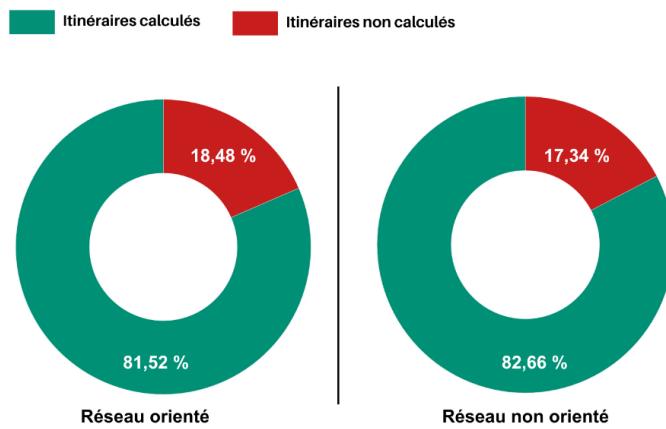


Figure 2 : Pourcentage des itinéraires calculés et non calculés sur le réseau orienté et non orienté (essai 1).

Comme le montre la figure 2, le pourcentage des itinéraires calculés et non calculés sur le graphe orienté ainsi que sur le réseau non orienté sont pratiquement identiques. Nos recherches, effectuées manuellement sur QGIS dans ce cas-ci, ont permis d'identifier deux types d'erreurs. Le premier type d'erreur est lié à la non-connexité du réseau, provoqué par son extraction à l'échelle du territoire étudié (Figure 3).

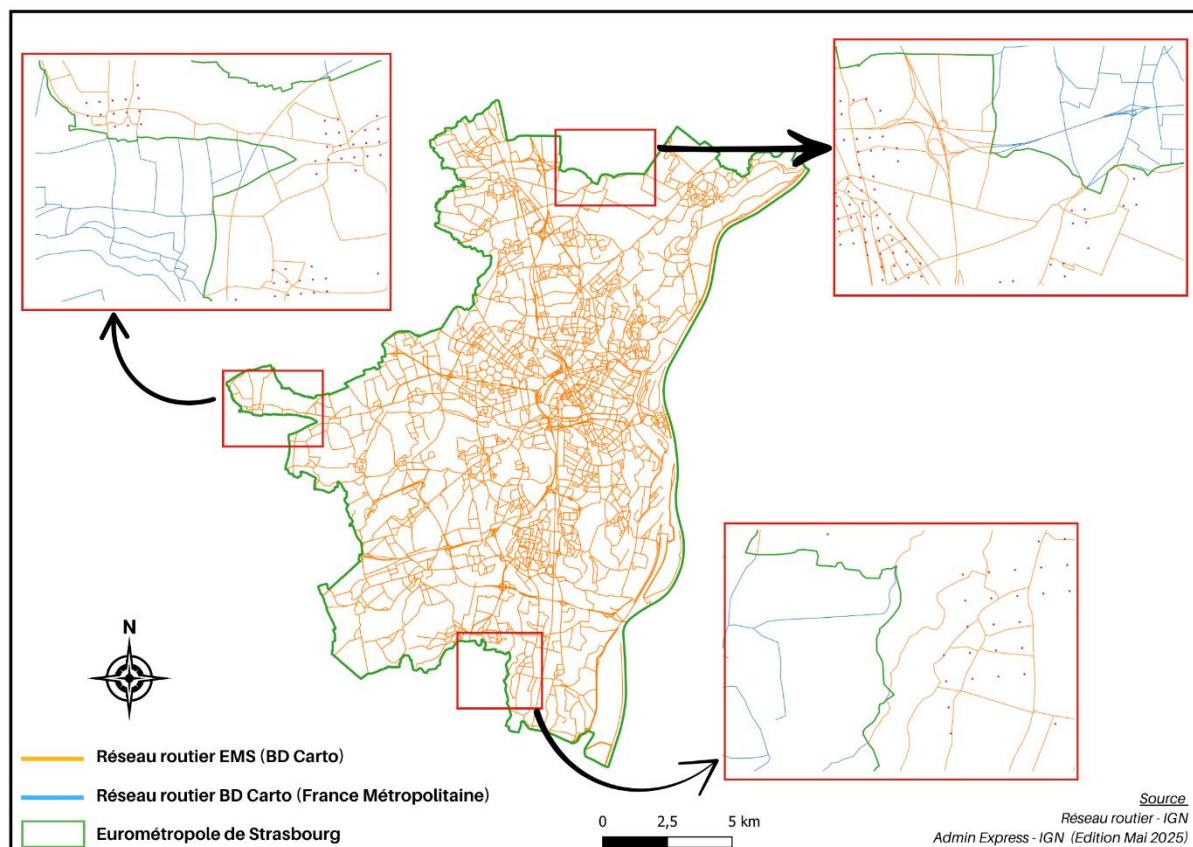


Figure 3 : Erreurs topologiques (discontinuités dans le réseau de route) suite au découpage de celui à l'échelle de l'EMS – SERY Ange, 2025.

Comme le montre la figure 3, l'extraction du réseau routier simplifié (BD Carto) à l'échelle de l'EMS, a conduit à la déconnexion de certains segments de route, rendant inaccessible certains endroits du territoire, en particulier les zones en périphériques.

Le deuxième type d'erreur identifié, est de nature attributaire. Nous avons constaté que certains tronçons de route ou chemins accessibles aux véhicules ou réservés à une certaine catégorie de personnes (pompiers, police, employé...) présentent des vitesses de circulation nulles (pas renseignées) (Figure 4). De plus, la majorité de ces tronçons problématiques se situent entre de segments de routes avec des vitesses non nulles (Figure 5). Cette erreur attributaire rend impossible le calcul d'itinéraires pour les points OD utilisant ces tronçons.

ERT	SENS	ACCES_VL	FERMETURE	RESTR_H	RESTR_P	FICTIF	VIT_MOY_VL	ID_RN	NUMERO	NUM_EUROP	CL_A
25	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
26	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
27	Double sens	Libre	NULL	4,40	NULL	Non	0	NULL	NULL	NULL	NULL
28	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
29	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
30	Double sens	Libre	NULL	NULL	NULL	Non	0	MEROUNUM00...	NULL	NULL	NULL
31	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
32	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
33	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
34	Sens direct	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
35	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
36	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
37	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
38	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
39	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
40	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
41	Double sens	Libre	NULL	NULL	NULL	Non	0	MEROUNUM00...	NULL	NULL	NULL
42	Double sens	Libre	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
43	Double sens	Libre	NULL	4,40	NULL	Non	0	NULL	NULL	NULL	NULL

ERT	SENS	ACCES_VL	FERMETURE	RESTR_H	RESTR_P	FICTIF	VIT_MOY_VL	ID_RN	NUMERO	NUM_EUROP	CL_A
1	Double sens	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
2	Double sens	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
3	Double sens	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
4	Double sens	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
5	Double sens	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	MEROUNUM00...	NULL	NULL	NULL
6	Sens direct	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	MEROUNUM00...	M468	NULL	Route ir
7	Double sens	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
8	Double sens	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
9	Double sens	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	MEROUNUM00...	NULL	NULL	NULL
10	Double sens	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	MEROUNUM00...	NULL	NULL	NULL
11	Double sens	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	NULL	NULL	NULL	NULL
12	Double sens	Restreint aux ayants droit	NULL	NULL	NULL	Non	0	MEROUNUM00...	NULL	NULL	NULL

Figure 4 : Extrait de quelques erreurs attributaires du réseau de route de la BD Carto : vitesse de circulation (VIT_MOY_VL) = 0.

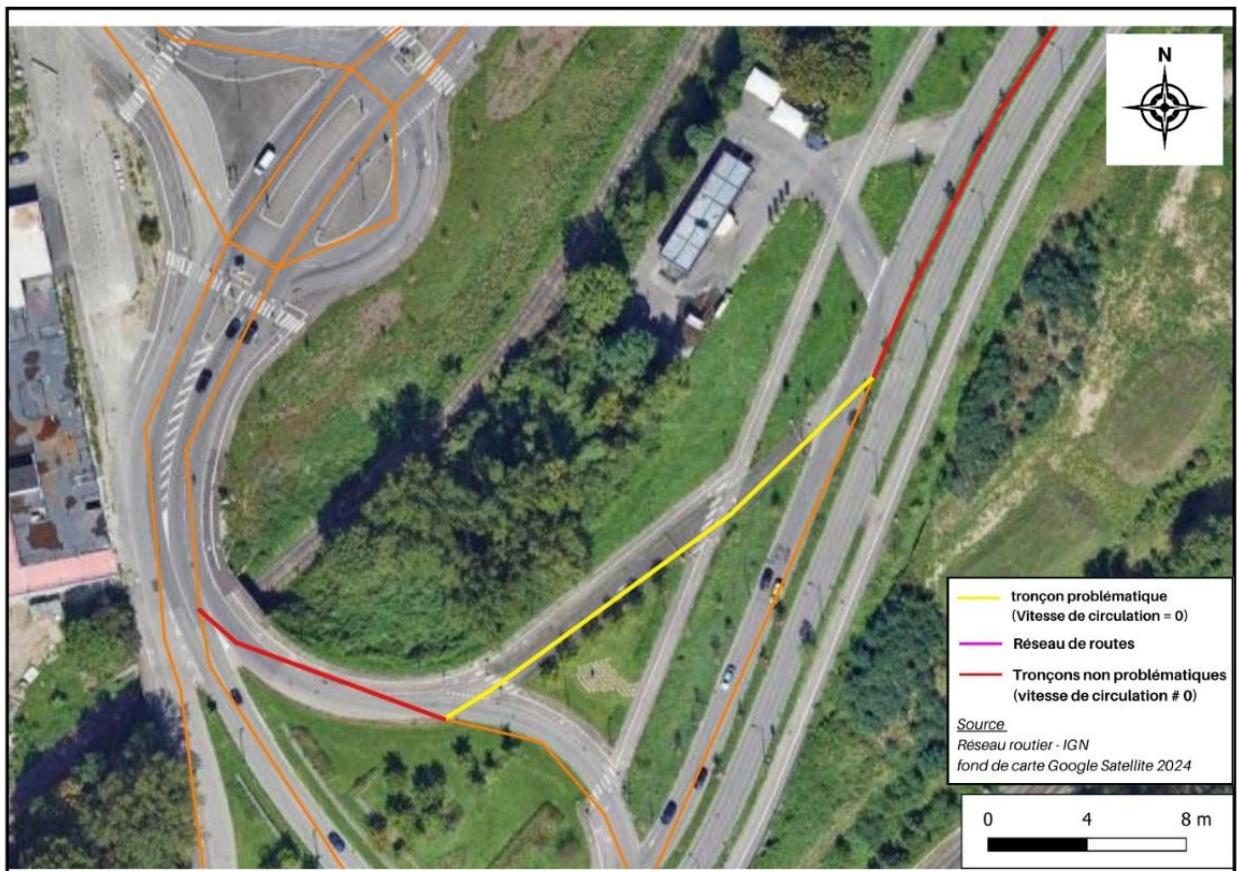


Figure 5 : Erreur attributaire dans le réseau de route de la BD Carto de 2024 – SERY Ange, 2025.

Pour résoudre le problème de discontinuité du réseau en périphérie du territoire, nous avons décidé de créer une zone tampon d'un kilomètre autour de la limite de l'EMS, permettant de couvrir une grande partie du réseau en périphéries, rendant ainsi accessible les points OD situés à ces endroits.

Pour corriger les erreurs sémantiques (vitesse de circulation nulle), nous avons développé un script Python, exécutable dans la console python de QGIS, qui de manière itérative, attribue à ces segments de routes la vitesse la plus petite parmi leur voisin directs (cf. II.2.2).

Essai 2 : après correction des erreurs identifiées au premier essai

Après avoir corrigé les erreurs découvertes lors de l'essai 1, nous avons cependant remarqué des itinéraires non calculés à la fois sur le réseau orienté et non orienté (Figure 6). Cela suggère que d'autres raisons, en plus des erreurs identifiées précédemment, sont à l'origine du non calcul de certains itinéraires.

- **Réseau orienté**

Nous avons enregistré 11 992 396 itinéraires calculés, ce qui représente 99,05 % de l'ensemble des itinéraires attendus. Les itinéraires non calculés sont de 114 524, soit 0,95 %.

- **Réseau non orienté**

Nous avons enregistré 12 086 920 itinéraires calculés, soit un pourcentage de 99,83 %. Les itinéraires non calculés s'élèvent au nombre de 20 890, soit un pourcentage de 0,17 %.

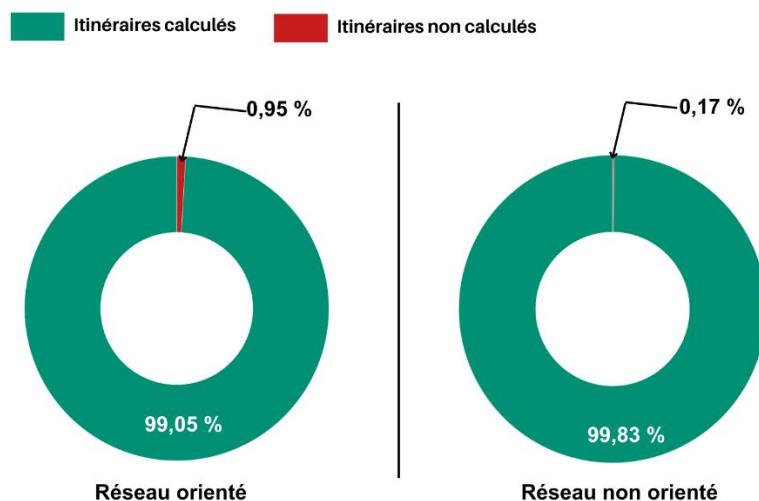


Figure 6 : Pourcentage des itinéraires calculés et non calculés sur le réseau orienté et non orienté (essai 2).

A partir d'un processus automatique de détection des erreurs mis en place à partir du ModelBuilder de QGIS (Annexe 1), accompagné de quelques vérifications manuelles, nous avons identifié deux autres erreurs qui pouvaient être de 2 types.

Le premier type d'erreurs est lié au degré de simplification des données de la BD Carto, avec trois cas de figures différentes.

Dans le premier cas [1], certains segments de route à sens unique (en vert) (Figure 7 – B), qui n'autorise la circulation que dans un seul sens, limite l'accessibilité à certains points considérés comme des destinations ou des origines car il n'existe pas d'autres alternatives de routes.

Par exemple, sur la Figure 7 – A, l'itinéraire entre les points OD 3248 et 3249 sur le réseau orienté, ne peut être calculé que dans une seule direction, de 3248 vers 3249 car il n'existe pas d'alternatives de routes pouvant être empruntées pour le trajet 3249 → 3248. Par contre, sur le réseau non orienté, l'itinéraire est calculé dans les deux directions (de 3248 vers 3249 et de 3249 vers 3248).

Au total, sur la base des itinéraires non calculé avec le réseau orienté (114 524), 93 403 ont été identifiés à ce problème, c'est-à-dire des itinéraires calculés que dans une seule direction.

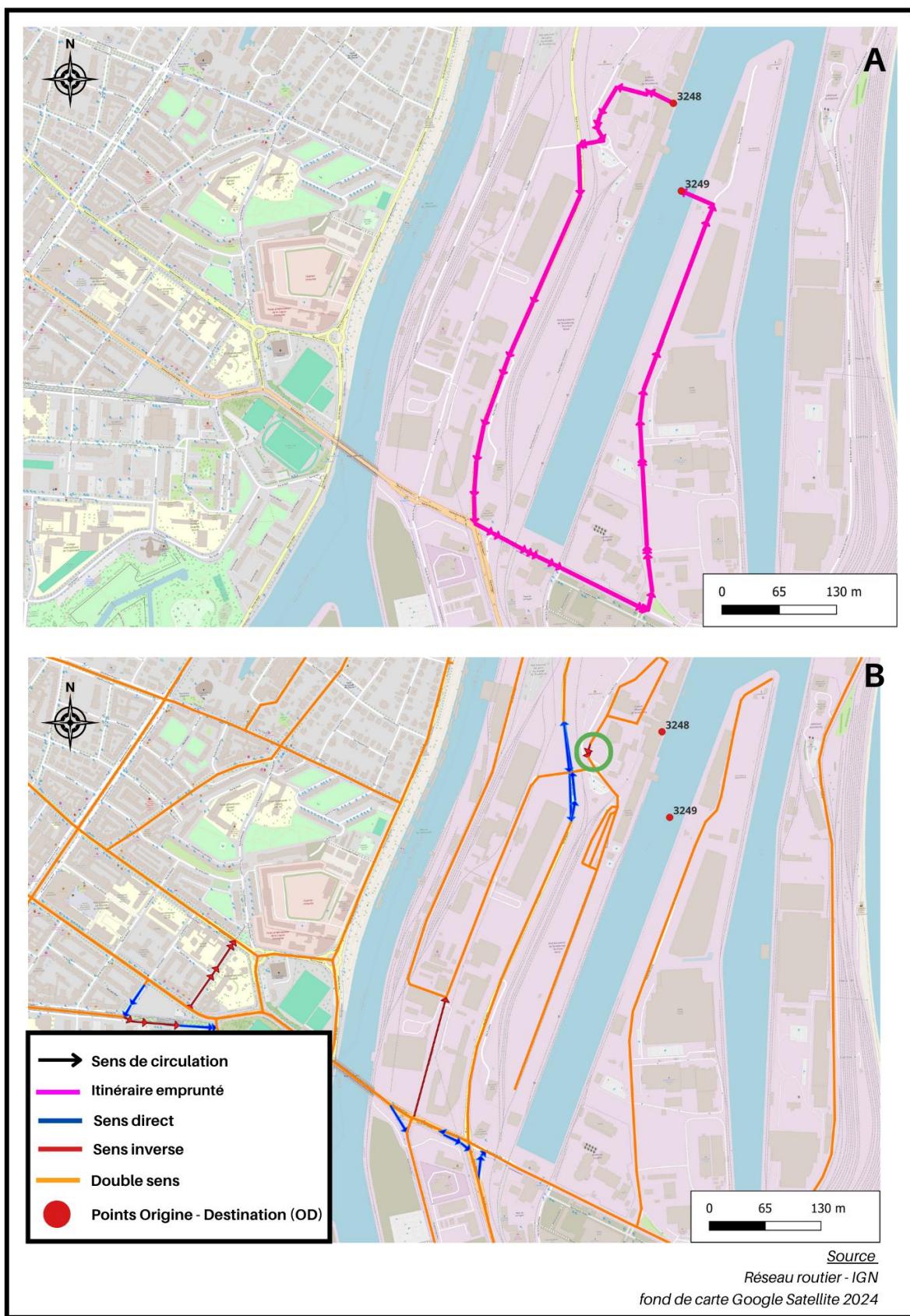


Figure 7 : Itinéraire calculé (chemin plus rapide, réseau orienté - A) et réseau routier d'après la BD Carto (B) - SERY Ange, 2025.

Dans le deuxième cas [2], nous avons des points OD situés à des endroits du réseau où le trajet est impossible entre des paires de points OD, c'est-à-dire d'origine à destination et de destination à origine, sur le réseau orienté. La raison pour laquelle il n'y a pas d'itinéraires calculés entre ces paires de points est que les segments de routes les plus proche d'eux sont des routes à sens uniques, ce qui les rend impossible à franchir lorsque l'un des points est considéré comme destination et l'autre origine, et vice versa. (Figure 8).

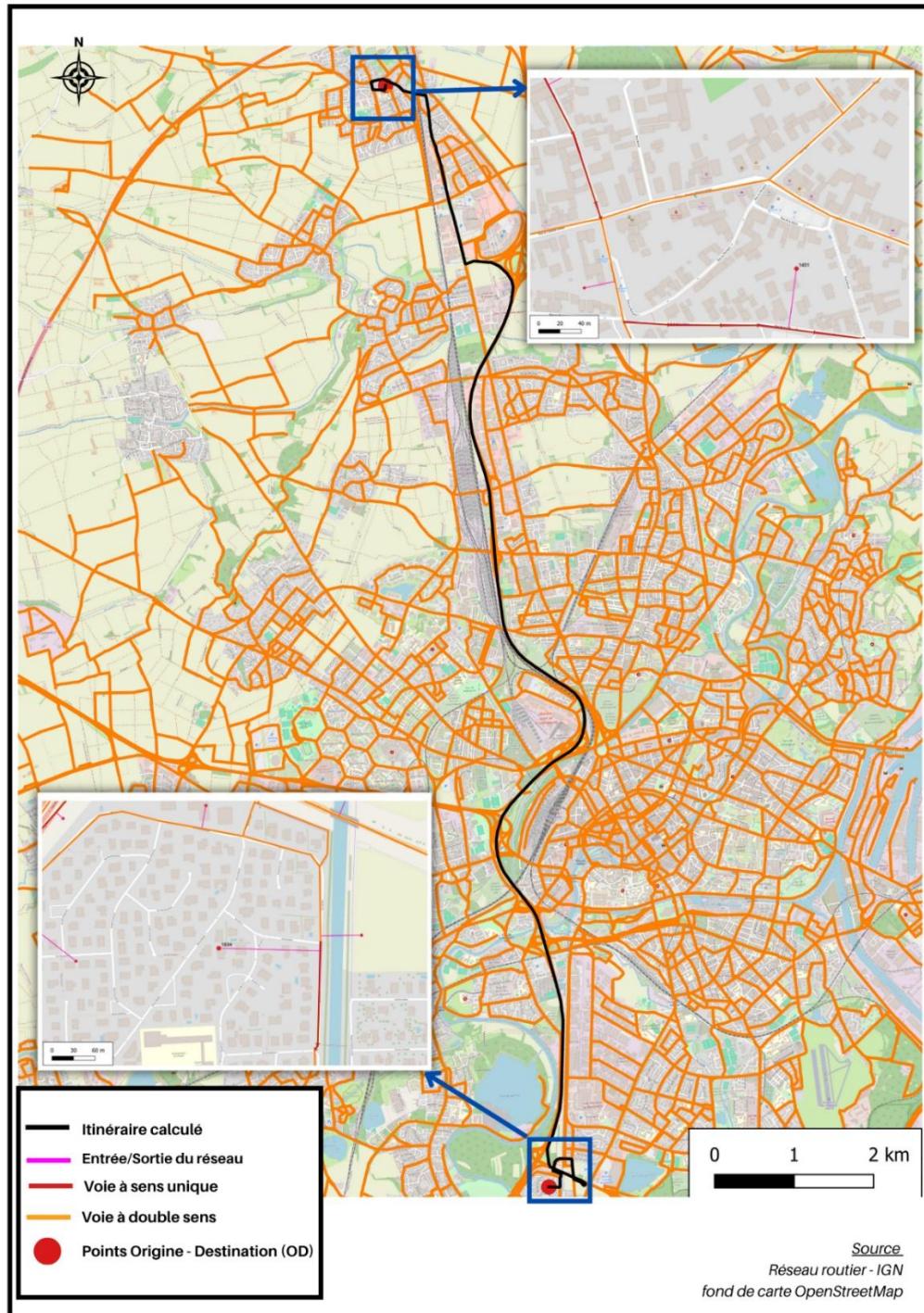


Figure 8 : Exemple d'itinéraires calculés sur le réseau non orienté et non calculé sur le réseau orienté entre des points OD.

La Figure 8 illustre le cas des paires de points OD 1451 et 1834. Sur le réseau non orienté, les itinéraires sont calculés entre ces paires de points, et le trajet emprunté reste le même dans les deux directions (en noire). En revanche, sur le réseau orienté, la navigation entre ces points devient impossible car certains tronçons de routes qui étaient praticables (sur le réseau non orienté) sont devenus infranchissable dans une direction donnée. De plus, il n'existe pas de routes alternatives pouvant être empruntées car elles ne sont pas toutes numérisées dans la BD Carto.

Sur un total de 12 106 920 itinéraires attendus, nous avons eu peu de cas de ce genre : 312 itinéraires non calculés.

Le troisième cas de figure [3], est des itinéraires non calculés entre certaines mailles adjacentes (maille en noir – Figure 9). En effet, pour ces mailles adjacentes le point d'entrée dans le réseau ainsi que le point de sortie sont identiques et se situent à l'extrémité d'un tronçon de routes. Cela empêche le calcul d'itinéraires entre ces mailles vu que les points d'entrée dans le réseau et de sortie sont confondus. Au total, sur la base des itinéraires non calculé (réseau orienté et non orienté), 24 ont été identifiés à ce problème.

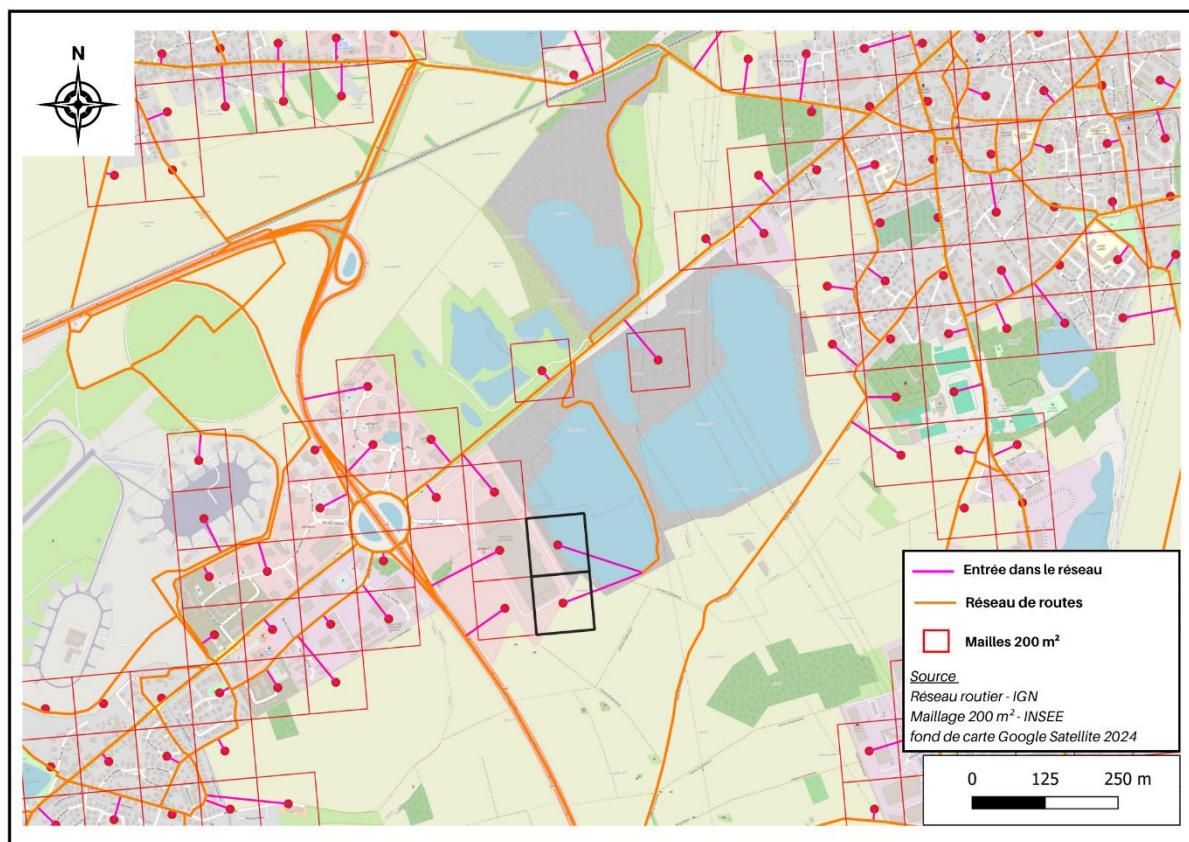


Figure 9 : Exemple de mailles adjacentes (en noir) avec pour conséquence des itinéraires non calculés entre elles sur le réseau orienté et non orienté - SERY Ange, 2025.

Le deuxième type d'erreurs identifiées est la non connexité du réseau causé par le filtrage des segments de routes accessibles en véhicule automobile. Ces erreurs topologiques ont été repérées grâce à l'outil “*v.net.components*” de GRASS sur QGIS, puis corrigées par la mise en place d'un ModelBuilder (cf. II.2.2).

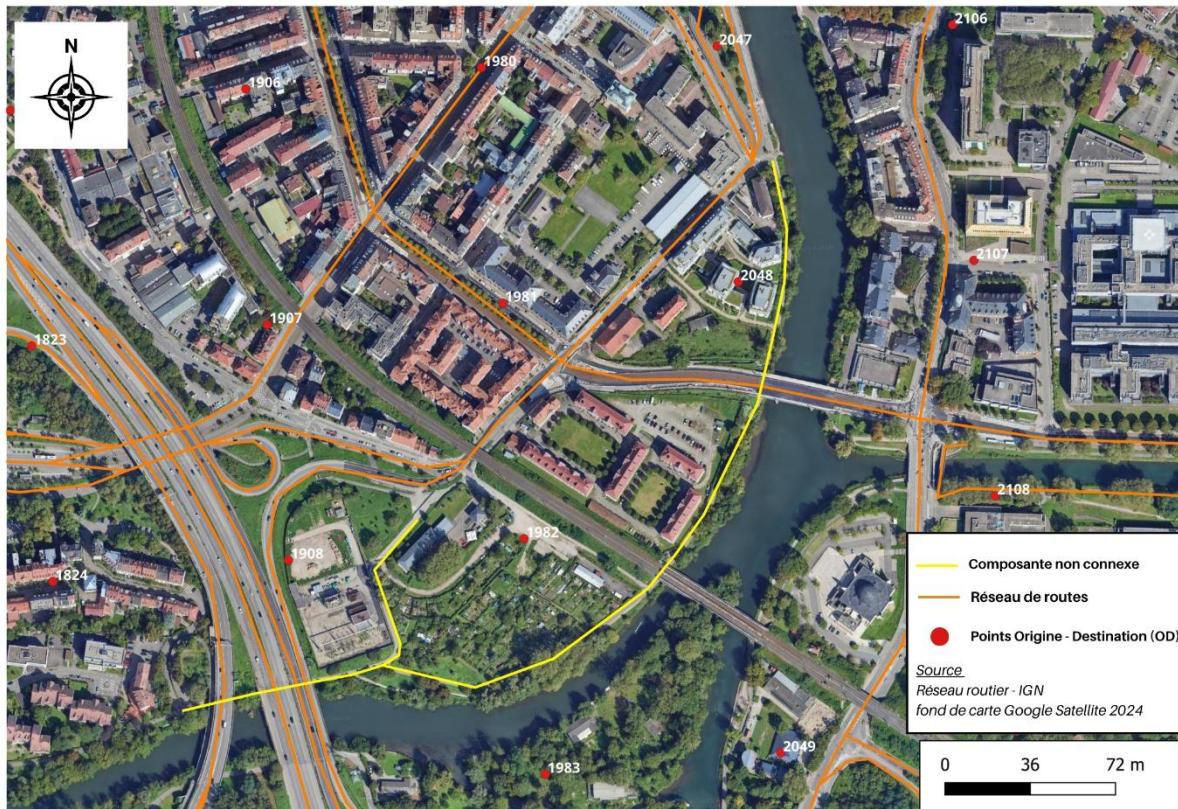


Figure 10 : Erreur topologique de réseau de routes de la BD Carto de 2024 - SERY Ange, 2025.

Comme le montre la figure 10, on voit que la composante (en jaune) n'est pas connectée au réseau principale (en orange). Cela a pour conséquence l'inaccessibilité des points OD se trouvant proche de ces segments de route car pour un point de navigation donnée, l'entrée dans le réseau se fait du côté du segment de route le plus proche de celui-ci (Bocher *et al.*, 2014).

Essai 3 : après correction des erreurs identifiées au deuxième essai

Après la correction des erreurs identifiées lors de l'essai 2, notamment celles liées à la non-connexité du réseau des suite du filtrage des routes accessibles en véhicule automobile, des itinéraires non calculés ont encore été enregistrés sur le réseau orienté et non orienté.

- **Réseau orienté**

Avec le réseau orienté, nous avons enregistré 12 013 180 itinéraires calculés, ce qui représente 99,22 % de l'ensemble des itinéraires attendus. Les itinéraires non calculés sont de 93 739, soit un pourcentage de 0,78 %.

- **Réseau non orienté**

Avec le réseau non orienté, nous avons enregistré 12 106 896 itinéraires calculés, soit un pourcentage de 99,99 %. Les itinéraires non calculés s'élèvent au nombre de 24 (0,01 %).

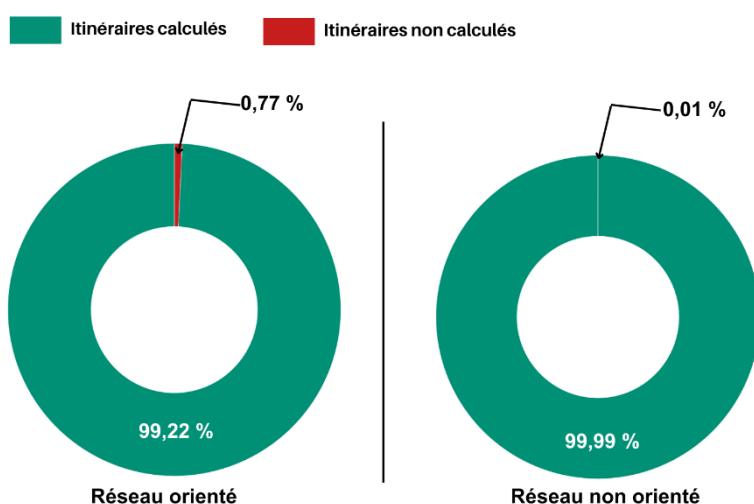


Figure 11 : Pourcentage des itinéraires calculés et non calculés sur le réseau orienté et non orienté (essai 3).

Le processus de détection automatique des erreurs mis en place nous a révélé que les itinéraires non calculés, ont pour causes les mêmes que celles présentées précédemment dans les 3 cas de figures à l'essai 2.

Cependant, pour pallier au problème du non calcul des itinéraires liés à ces causes, nous avons décidé d'intégrer la BD Topo aux données de la BD Carto. Cela permettrait d'ajouter les tronçons de routes qui ne sont pas numérisés dans la BD Carto.

II.2.2. Correction des erreurs topologiques et attributaires

II.2.2.1. Vitesses de circulations non renseignées : nulle

Pour corriger les vitesses de circulations nulles, nous avons mis en place un script Python (PyGis), exécutable dans la console python de QGIS (Annexe 2). Ce script corrige de manière itérative les vitesses nulles en attribuant aux tronçons de routes concernés, la vitesse minimale parmi leurs voisins directement connectés (c'est-à-dire ayant au moins un sommet en commun). Ce processus est répété jusqu'à ce que tous les tronçons initialement nuls soient corrigés.

Le choix d'affecter les vitesses minimales plutôt que maximales, permet de garder une certaine prudence, afin d'éviter de surestimer les vitesses de circulation et d'être plus en phase avec la réalité dans les zones urbaines où les vitesses varient considérablement.

```
from qgis.PyQt.QtCore import QVariant
from qgis.core import *

# Paramètres

nom_couche = "bd_carto_routes"    # <- nom du réseau à corriger
champ_vitesse = "VIT_MOY_VL"      # <- champ vitesse BD Carto
champ_correction = "VIT_CORR"     # <- nouveau champ vitesse

..... (suite code)

# Résultat final

reste = [f.id() for f in layer.getFeatures() if
f[champ_vitesse] == 0 and f[champ_correction] == 0]
print("Correction terminée." if not reste else f"{len(reste)} "
tronçon(s) non corrigé(s) après {iteration} itération(s).")
```

Figure 12 : Extrait du code de correction de vitesses de circulations nulles dans la BD Carto.

II.2.2.2. Connexité du réseau

Pour corriger l'erreur topologique liée à la non connexité du réseau, nous avons mis en place un processus automatique de détection et de suppression automatique des composantes non connectées au réseau. Ce processus a été implémenté à partir du ModelBuilder de QGIS (Figure 13).

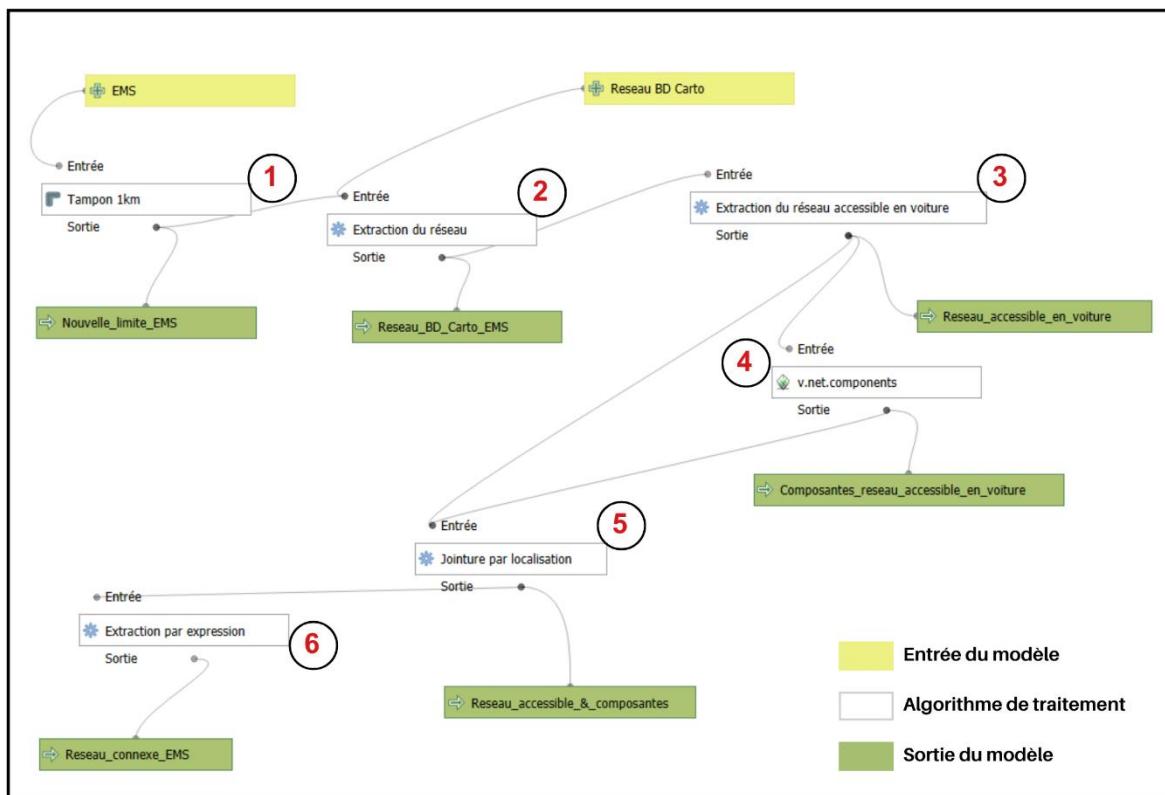


Figure 13 : Processus de détection et correction d'erreurs topologiques liées à la non connexité du réseau via le ModelBuilder de QGIS.

Les deux entrées du modèle sont la couche administrative de l'EMS et le réseau routier de la BD Carto. A partir de la couche de l'EMS, une zone tampon de 1km est appliquée afin rendre accessible les points OD situés en périphérie du territoire [1]. S'ensuit alors une extraction du réseau de la BD Carto à l'échelle de la zone tampon [2] et des tronçons de route accessible en voiture [3]. A l'aide de l'outil “V.net.components” (issus) du module GRASS intégré à QGIS), toutes les composantes du réseau (connexes et non connexes) sont déterminées [4] puis identifiées dans le réseau à l'aide d'une jointure par localisation [5]. Enfin, une extraction par expression permet de filtrer les composantes connexes (valeur de composante = 1) [6].

II.2.3. Conclusion

Dans ce chapitre, notre objectif était tout d'abord d'identifier les erreurs attributaires et topologiques qui empêchent le calcul d'itinéraires à une résolution spatiale fine (mailles de 200 mètres de côtés), puis de corriger ces erreurs.

Pour ce faire, nous avons constitué un échantillon de 3 480 points origine-destination (OD) basés sur les mailles de 200 mètres de côtés, et pouvant être considérés chacun comme origine et destination pour nos simulations de calcul d'itinéraires sur le réseau routier de la BD Carto à l'échelle de l'EMS.

Les premiers résultats obtenus, nous ont indiqué sur un total de 12 106 920 paires d'itinéraires possibles, un taux de 81,52 % des itinéraires calculés sur le réseau orienté et 82,48 % sur le réseau non orienté. La procédure automatique de détection des erreurs (Annexe 1) que nous avons mis en place, nous a permis de constater que le non calcul des itinéraires pouvait être de 3 types : la non-connexité du réseau due au découpage à l'échelle de l'EMS et au filtrage des routes accessibles en véhicule [1], des vitesses de circulation nulles (égale à 0) pour des tronçons accessibles en véhicule [2], et le degré de simplification du réseau accentuée par la prise en compte des sens de circulation [3].

La correction du premier type d'erreur [1], via une zone tampon de 1 kilomètre autour de la limite de l'EMS et une procédure automatique de détection et de suppression des tronçons de routes non connectés au réseau, ainsi que celle du second [2], par un script PyGIS, ont permis d'atteindre un taux de 99,22 % d'itinéraires calculés sur le réseau orienté et 99,99 % sur le réseau non orienté. En revanche, le troisième type d'erreur [3], responsable de 0,78 % d'échecs de calcul d'itinéraires sur le réseau orienté et 0,01 % sur le réseau non orienté, n'a pas pu être corrigé dans cette partie, car nous avons supposé que l'intégration de la BD Topo aux données de la BD Carto pourrait résoudre ce problème.

Chapitre III – Intégration de la BD Topo aux données de la BD Carto et évaluation du réseau intégré

Chapitre III – Intégration de la BD Topo aux données de la BD Carto et évaluation du réseau intégré.....	32
III.1. Intégration de la BD Topo	33
III.2. Calcul des itinéraires	37
III.3. Evaluation du réseau fusionné.....	39
III.4. Conclusion.....	44

III.1. Intégration de la BD Topo

La méthodologie que nous avons mise en œuvre pour intégrer le réseau routier de la BD Topo au réseau de la BD Carto, se décline de la façon suivante :

- Identification des tronçons de routes à intégrer dans la BD Topo [1] ;
- Fusion des tronçons identifiés dans la BD Topo, au réseau de la BD Carto sur la base des nœuds correspondants aux deux bases de données [2] ;
- Nettoyage du réseau routier fusionné [3] ;
- Remodelage de l'ensemble de données fusionné [4].

III.1.1. Identification des tronçons de routes à intégrer

L'objectif de cette étape est d'identifier tous les tronçons de routes accessibles en véhicule et n'ayant pas été saisis dans la BD Carto. Pour cela, nous avons effectué plusieurs filtrages dans la base de données Topo, selon la définition des attributs dans la documentation de l'IGN. Le premier filtrage a consisté à éliminer toutes les routes non accessibles en voiture, comme nous avons fait avec les données de la BD Carto au chapitre précédent. Le second filtrage a consisté à supprimer ensuite les tronçons de routes déjà saisis dans la BD Carto afin de conserver la nature simplifiée du réseau de la BD Carto et d'éviter les écarts géométriques entre les deux réseaux liés à leur superposition (Figure 14). Pour faire cela, nous nous sommes appuyés sur un attribut commun aux deux bases de données (“*ID_BDTOPO*²” dans le BD Carto et “*ID*” dans la BD Topo), et un script Python (PyGIS) a été utilisé pour automatiser le processus (Annexe 3).

² **ID_BDTOPO** est les ou l'identifiant (s) du ou des tronçon (s) BD_TOPO correspondant (s). Pour les correspondances 1 – 1 entre objets de la BD Carto et ceux de la BD Topo, l'*ID_BDTOPO* prend une valeur de la forme “TRONROUT000000001479949”. Cependant, pour les correspondances 1 – 2, l'*ID_BDTOPO* est de la forme “TRONROUT0000002223135928/TRONROUT0000002223135928”.

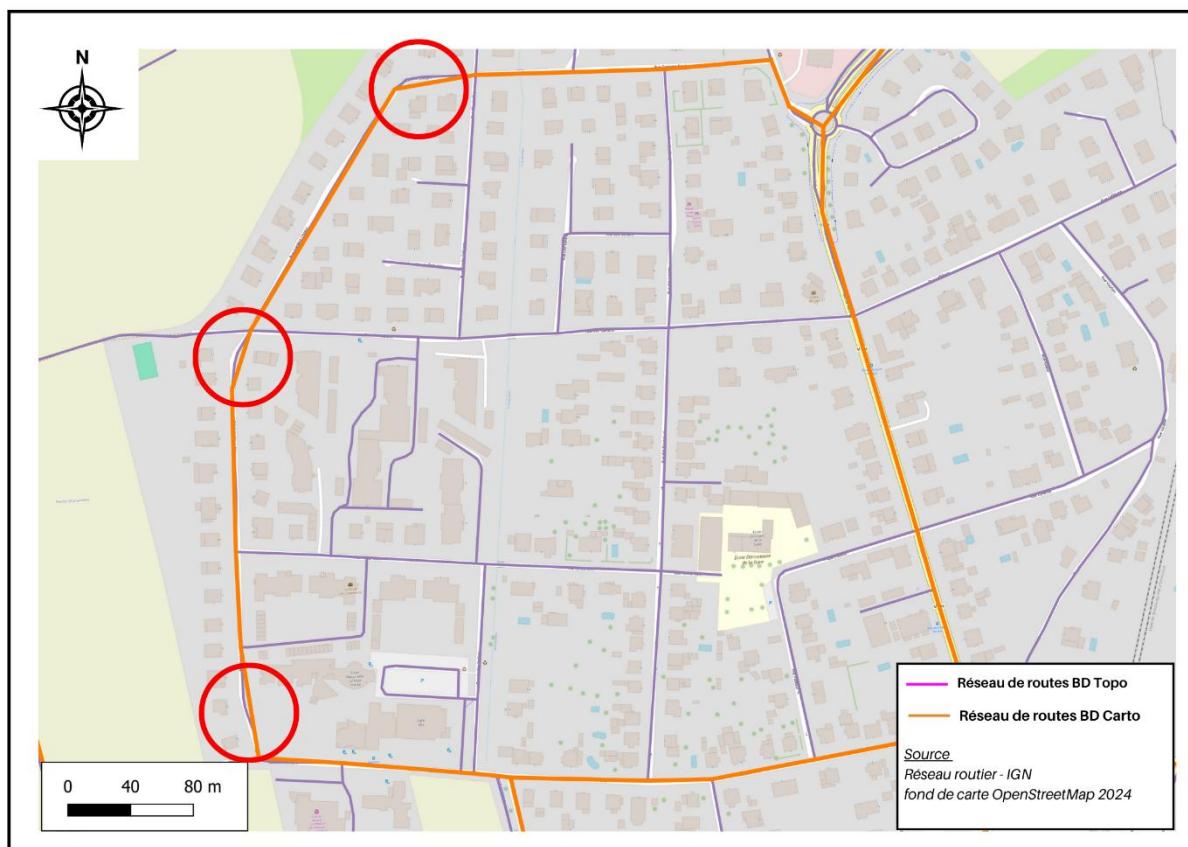


Figure 14 : Exemple d'écart géométrique entre des tronçons de routes identiques mais saisis dans des bases de données différentes.

III.1.2. Intégration des données et nettoyage du réseau fusionné

L'intégration des données de la BD Topo au réseau de routes de la BD Carto a été fait sur QGIS grâce à l'algorithme “*Fusionner des couches vecteurs*”. Cette fusion a été possible car les deux bases de données sont définies dans le même système de coordonnées de référence et ont des informations sémantiques en commun (ID, NATURE, IMPORTANCE, SENS, VIT_MOY_VL....) (Daneshgar *et al.*, 2018). La correspondance sémantique entre les deux bases de données est très importante car cela permet de mettre à jour automatiquement les attributs dans le réseau fusionné et de ne pas créer des nouveaux attributs.

Après la fusion, nous avons procédé au nettoyage du réseau fusionné, afin de s'assurer de la bonne connexité du réseau. Cela a été fait grâce à l'outil “*v.net.components*” de GRASS, qui a permis d'identifier les tronçons de routes non connecté au réseau et de les supprimer. Enfin, nous avons mis à jour les vitesses de circulation des tronçons intégrés afin d'attribuer des vitesses aux tronçons qui n'en possèdent pas (vitesse nulle).

III.1.3. Remodelage de l'ensemble de données fusionnées

Cette étape a consisté à bien organisé d'un point de vue topologique le réseau fusionné. Après le nettoyage du réseau fusionné, nous avons remarqué que certains tronçons de routes provenant de la BD Topo ne se sont pas connectés automatiquement au réseau de la BD Carto (Figure 15). Ce problème a été observée principalement au niveau des routes simplifiées à double sens de la BD Carto. Afin de connecter ces tronçons au réseau de la BD Carto, nous avons utilisé l'outil "accrocher les géométries" de QGIS avec une distance maximale d'accrochage de 20 mètres³.

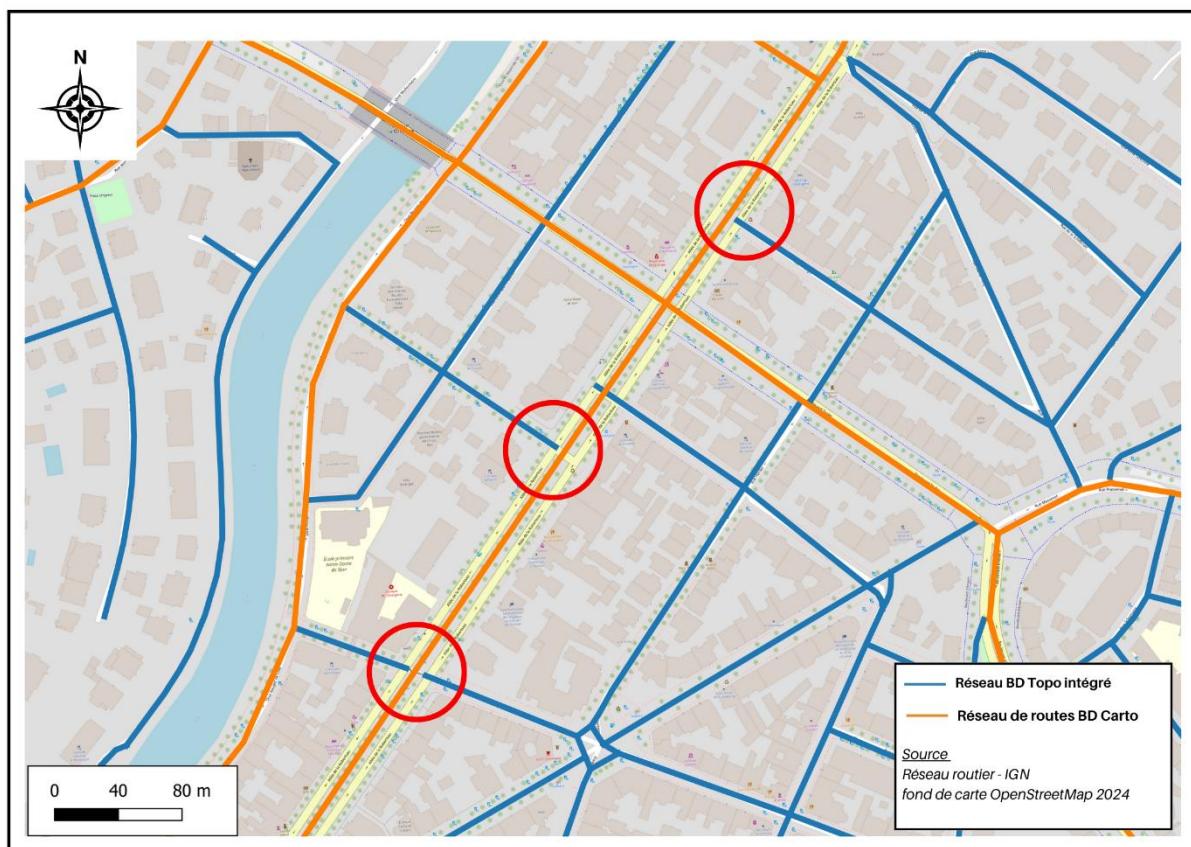


Figure 15 : Exemples de tronçons de routes BD Topo non connecté au réseau de la BD Carto (en rouge) - SERY Ange, 2025.

La démarche méthodologique été automatisée à l'aide du ModelBuilder de QGIS (Figure 16), à l'exception de deux étapes : La suppression des tronçons de routes déjà numérisé dans la BD Carto, et la mise à jour des vitesses de circulation, qui elles ont été automatisées grâce à des scripts PyGIS.

³ Paramètre retenu à la suite de plusieurs mesures de distances euclidiennes des tronçons concernés sur QGIS.

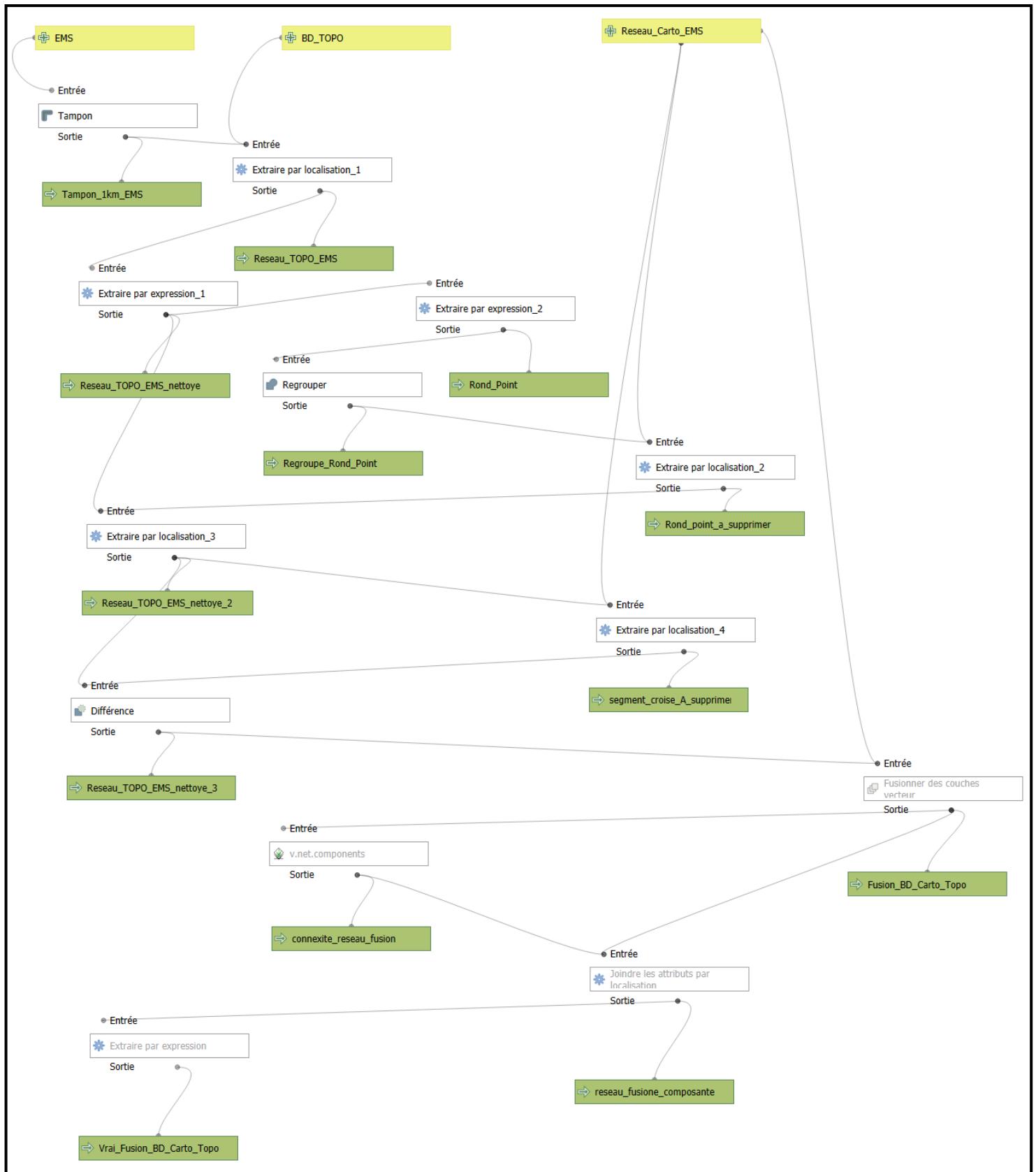


Figure 16 : Processus d'intégration de la BD Topo à la BD Carto via le modeleur de QGIS.

III.2. Calcul des itinéraires

L'objectif ici était de vérifier l'état des itinéraires non calculés sur le réseau de la BD Carto après l'intégration des tronçons de route de la BD Topo.

Le calcul d'itinéraires s'est fait d'une part sur le réseau orienté et d'autres part sur le réseau non orienté. Nous avons conservé le même échantillon pour nos points OD c'est-à-dire 3 480 points pouvant être des origines et des destinations.

- **Réseau orienté**

Avec le réseau orienté, nous avons enregistré 11 978 466 itinéraires calculés, ce qui représente 98,93 % de l'ensemble des itinéraires attendus (12 106 920). Les itinéraires non calculés sont de 128 454, soit un pourcentage de 1,06 %.

- **Réseau non orienté**

Avec le réseau non orienté, nous avons enregistré 12 106 900 itinéraires calculés, soit un pourcentage de 99,99 %. Les itinéraires non calculés s'élèvent au nombre de 20 (0,01 %).

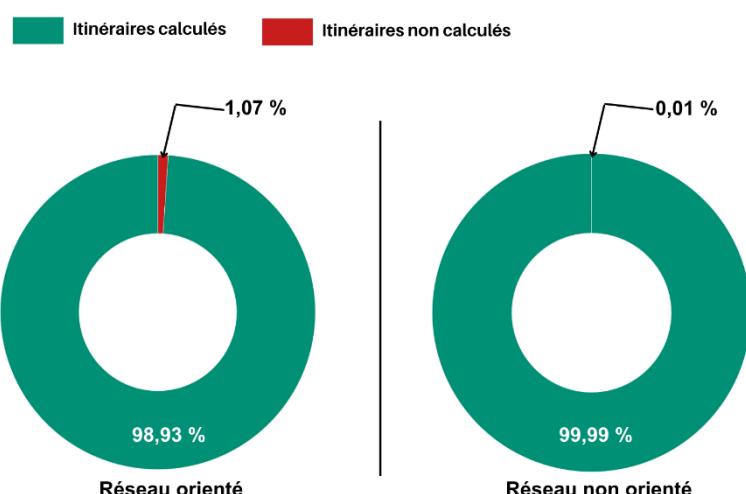


Figure 17 : Pourcentage des itinéraires calculés et non calculés sur le réseau orienté et non orienté.

Après vérifications, nous avons constaté que parmi les itinéraires non calculés (93 739) sur le réseau de la BD Carto, 86 376 ont pu être calculé sur le réseau orienté du réseau fusionné, soit un pourcentage de 92,14 %.

Cependant, nous avons recensé à nouveau des itinéraires non calculés sur le réseau fusionné (Figure 17). Parmi ces itinéraires non calculés, 93 134 (72,50 %) correspondaient à des trajets

calculés uniquement dans une seule direction c'est-à-dire du point d'origine vers le point de destination, et 35 320 (27,49 %) dans aucune des directions.

Après avoir utilisé la même procédure automatique que celle utilisée pour les données de la BD Carto (Annexe 1) et effectué quelques vérifications manuelles, nous avons constaté que les raisons pour lesquelles certains itinéraires n'étaient pas calculés sur le réseau fusionné étaient les mêmes que celles observées avec les données de la base de données Carto. En d'autres termes, il s'agissait de la présence de sens de circulation opposés (sens direct et indirect) sur certains trajets empruntés, ou de la présence de sens de circulation unique (sens direct ou sens indirect), contraignant la navigation dans une seule direction ou dans aucune direction entre des paires de points OD (cf. II.1. – Essai 2). De plus, les itinéraires générés avec le réseau non orienté nous ont permis de confirmer cette observation car aucune erreur de calcul d'itinéraire n'a été enregistrée pour les paires de points OD concernées par ce problème.

La seconde cause, identifiée aussi avec les données de la BD Carto, concerne les mailles adjacentes dont les points d'entrée et de sorties dans le réseau sont identiques et situés à l'extrémité d'un segment de routes. Cela empêche le calcul d'itinéraires entre ces mailles adjacentes vu que les points d'entrée et de sortie dans le réseau sont les mêmes. C'est ce problème qui est à l'origine des 20 itinéraires non calculés sur le réseau non orienté.

Nous avons pensé que l'intégration de la BD Topo à la BD Carto aurait permis d'accéder à certains points OD auparavant inaccessibles lors du calcul d'itinéraires avec le réseau routier de la BD Carto. Cela a été le cas pour une majorité, puisque sur 93 739 itinéraires non calculés, 83 376 ont pu l'être sur le réseau fusionné. Toutefois, d'autres points sont devenus inaccessibles, comme avant l'intégration de la BD Topo et ce pour les mêmes raisons qu'avant.

A ce stade de l'étude, nous avons donc préféré évaluer le réseau fusionné en le comparant avec celui de la BD Carto sur la base du nombre d'entités (tronçons de routes), de temps de calcul des itinéraires et du coût de trajets (temps et distances).

III.3. Evaluation du réseau fusionné

III.3.1. Comparaison du nombre d'entités et du temps de calcul des itinéraires entre BD Carto et le réseau fusionné

La comparaison du nombre d'entités (tronçons de route) entre les deux bases de données a été réalisée en se basant principalement sur les réseaux de routes utilisés lors des calculs d'itinéraires. Pour effectuer cette comparaison, nous avons compté le nombre total de tronçons de routes présents dans chacune des bases de données. Ainsi, dans la BD Carto, nous avons recensé un total de 18 212 entités. En parallèle, dans le réseau fusionné, le nombre de tronçons de routes s'élève à 30 372 dont 12 160 entités de plus que la BD Carto, soit une augmentation de 40 %. Cependant, par rapport à la BD Topo, ce chiffre représente une diminution de 49 % par rapport au nombre total de tronçons de routes dans la BD Topo.

Concernant les temps de calcul des itinéraires générés par l'outil QNEAT3 sur QGIS, nous avons noté qu'il a fallu 1 heure et 10 minutes pour le réseau fusionné et 55 minutes pour la BD Carto pour. Cela indique un temps de calcul de 15 minutes de plus sur le réseau fusionné, qui semble plutôt logique car le nombre d'entités du réseau fusionné est supérieur à celui de la BD Carto.

III.3.2. Comparaison des coûts (temps et distances) de trajets entre la BD Carto et le réseau fusionné

L'objectif ici était de vérifier l'intérêt de l'intégration de la BD Topo pour la suite des simulations du modèle ABSUM. Cette vérification s'est basée sur la comparaison des itinéraires calculés sur le réseau de la BD Carto avec ceux calculés sur le réseau fusionné, en termes de temps de trajets (chemin le plus rapide) sur le graphe orienté et distances de trajets (chemin le plus court) sur le graphe non orienté. Notre analyse portera que sur 60 000 paires OD choisi aléatoirement, représentatif de l'ensemble des 12 106 920 paires OD possible à l'échelle de l'EMS.

- **Réseau orienté**

Les itinéraires sur le réseau orienté, ont été calculés sur la base des vitesses moyennes maximales de circulation des tronçons du réseau routier, telles que définies par l'IGN.

La moyenne des temps de trajets sur le réseau de la BD Carto est de 978 secondes, soit 16 minutes et 18 secondes. Celle sur le réseau fusionné est de 964 secondes, soit 16 minutes et 4 secondes. On a donc une différence moyenne de 14 secondes en plus entre une même paire OD sur le réseau de la BD Carto. Cela correspond à des temps de trajets d'environ 1 % plus longs en termes de temps sur le réseau de la BD Carto comparé au réseau fusionné.

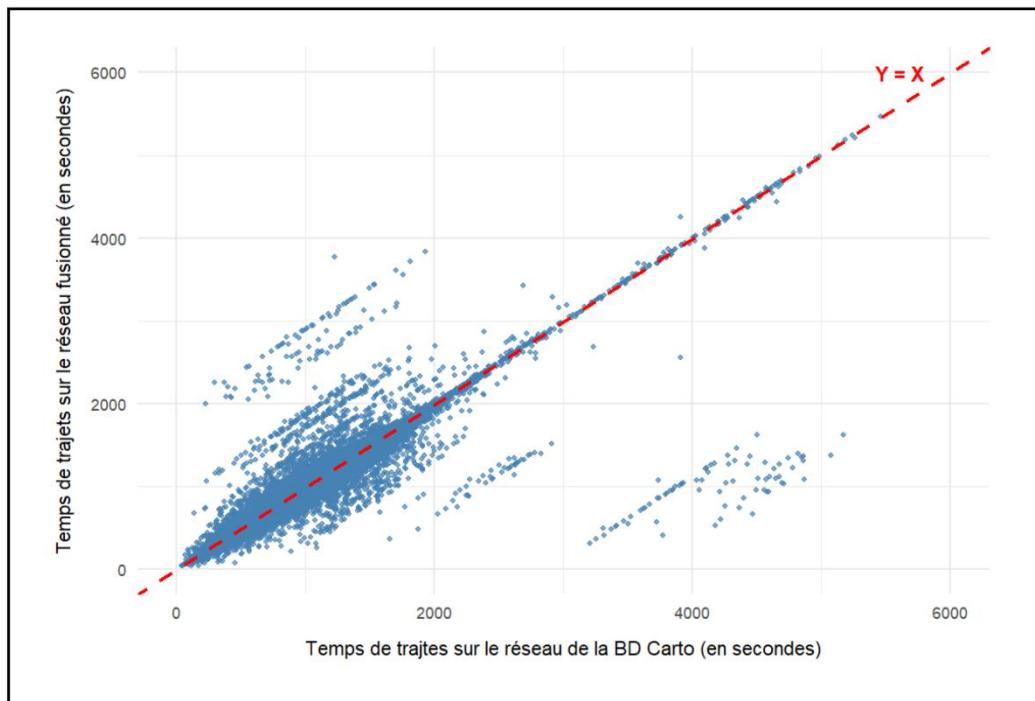


Figure 19 : Comparaison des temps de trajets entre le réseau de la BD Carto et le réseau fusionné. La droite (en rouge) correspond à la droite d'identité ($Y = X$).

Comme le montre la Figure 19, on note un alignement général autour de la droite d'identité ($Y=X$). Cela indique que les temps de trajets sur le réseau fusionné sont très proches de ceux de la BD Carto, même si on note quelques écarts.

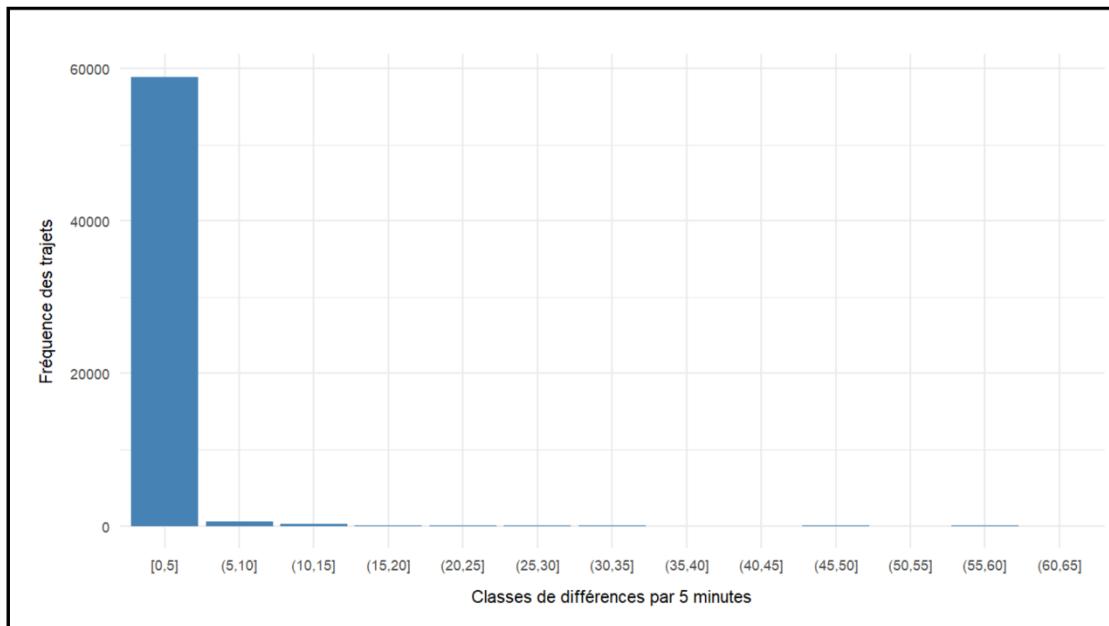


Figure 20 : Ecart absolu de temps de trajets entre le réseau de la BD Carto et le réseau fusionné – SERY Ange, 2025.

Comme on peut le voir sur la Figure 20, les 2/3 des trajets sur le réseau de la BD Carto sont en général plus long d'au plus 5 minutes que les trajets du réseau fusionné. On note des écarts de temps de plus de 5 minutes mais très minime.

- **Réseau non orienté**

Les itinéraires sur le réseau non orienté, ont été calculés sur la base d'une vitesse de déplacement constante de 9,15 km/h, qui correspond à la moyenne des vitesses de déplacements à pied (4km/h) et en vélo (15km/h) communément admises.

La moyenne des distances de trajets sur le réseau de la BD Carto est de 10 113 mètres, soit 10,11 km. Celle sur le réseau fusionné est de 10 066, soit 10,07 km. On a donc une différence moyenne de 40 mètres en plus entre une même paire OD sur le réseau de la BD Carto. Cela correspond à des temps de trajets d'environ 0,39 % plus longs en termes de distance sur le réseau de la BD Carto comparé au réseau fusionné.

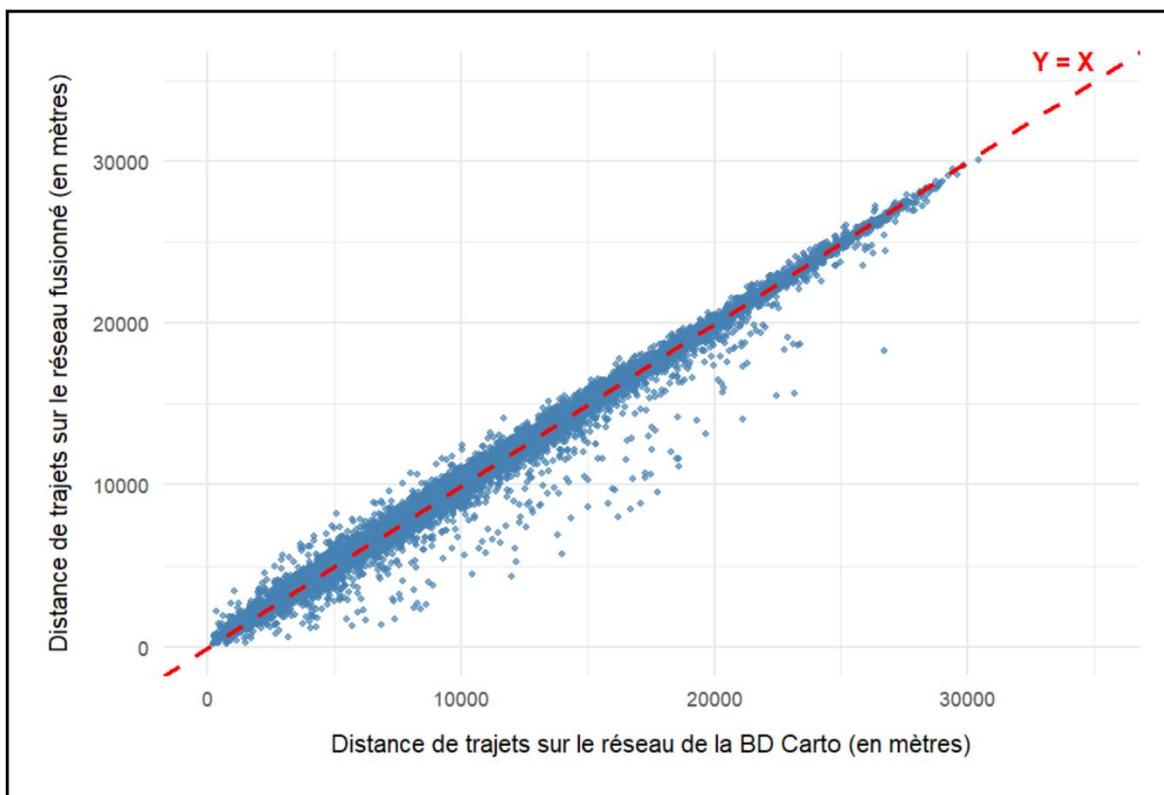


Figure 21 : Comparaison des distances de trajets entre le réseau de la BD Carto et le réseau fusionné. La droite (en rouge) correspond à la droite d'identité ($Y = X$).

Comme on le voit sur la Figure 21, on observe une forte corrélation entre les distances calculées sur les deux réseaux. Pour certains points (sous la droite d'identité), la distance de trajets est plus grande sur le réseau de la BD Carto que sur le réseau fusionné.

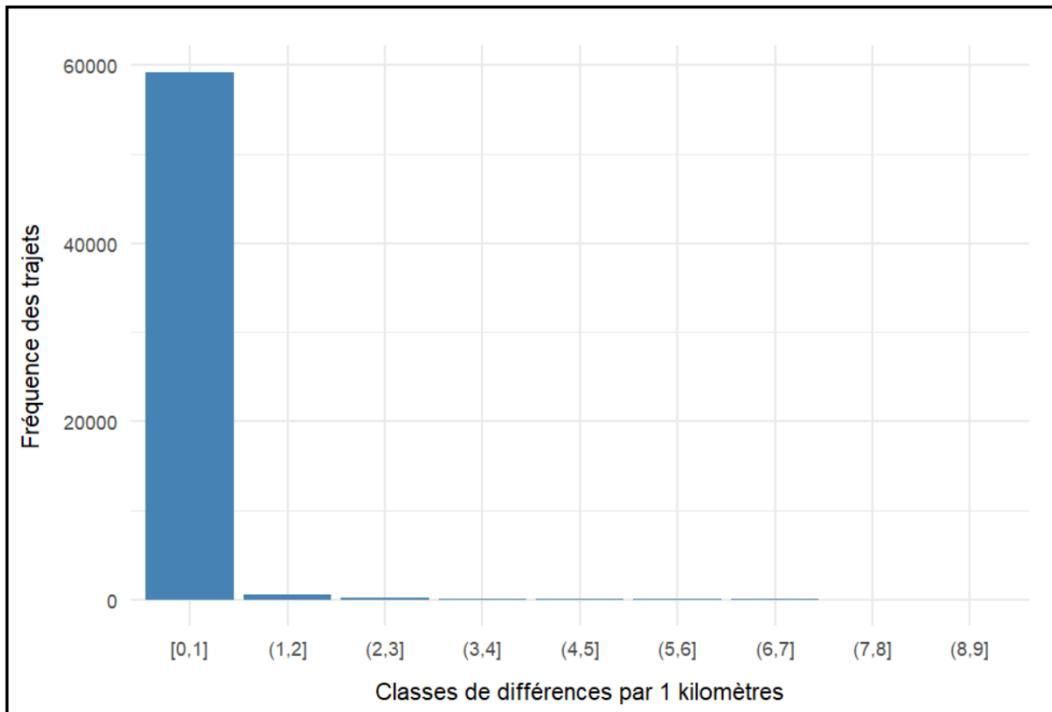


Figure 22 : Ecart absolu de distances de trajets entre le réseau de la BD Carto et le réseau fusionné – SERY Ange, 2025.

La figure 22, nous montre que les 2/3 des trajets sur le réseau de la BD Carto sont plus long d'au plus 1km que les trajets sur le réseau fusionné. Parmi notre échantillon de 60 000 itinéraires, on a que 824 trajets plus long de plus 1km, soit un pourcentage de 1,37 %.

III.4. Conclusion

Au cours de ce chapitre, notre objectif était double : d'abord, intégrer la BD Topo au réseau de la BD Carto, puis évaluer le réseau fusionné. L'intégration visait à rendre accessibles les points OD qui ne l'étaient pas avec les données de la BD Carto, notamment sur le réseau orienté.

Pour cela, nous n'avons sélectionné et intégré que les tronçons de routes de la BD Topo qui n'étaient pas déjà présents dans la BD Carto, afin de préserver la simplicité du réseau de référence (BD Carto). Grâce à cette intégration, 86 376 itinéraires sur 93 739 (soit 92,14%) qui n'avaient pas pu être calculés sur le réseau orienté de la BD Carto ont pu l'être sur le réseau fusionné en prenant en compte les sens de circulation. Cependant, de nouveaux itinéraires non calculés ont été identifiés, représentant un taux de 1,06% sur l'ensemble des itinéraires attendus (soit 12 106 920). Sur le réseau non orienté, seuls 20 itinéraires n'ont pas pu être calculés.

Les vérifications faites, basées sur une procédure automatisée suivi d'une vérification manuelle, nous ont permis de constaté que les causes des erreurs de calcul d'itinéraires sur le réseau orienté étaient les mêmes que celles détectées sur le réseau orienté de la BD Carto, c'est à dire des erreurs liées à la prise en compte des sens de circulation car sur le réseau non orienté, tous les itinéraires ont été calculés, sauf le cas des mailles adjacentes (des points d'entrée et de sortie dans le réseau identiques).

Enfin, nous avons comparé le réseau fusionné avec celui de la BD Carto pour évaluer sa pertinence pour la suite des simulations dans le modèle ABSUM. Les résultats indiquent que les temps de trajet sur la BD Carto sont toujours plus longs, d'au plus 5 minutes (réseau orienté) et plus long d'au plus 1 kilomètre sur le réseau non orienté.

Conclusion générale

L'étude et le travail réalisé avaient pour but de mettre en place un méthodologie reproductible et automatisé, qui permettra d'identifier et corriger les erreurs (topologiques et attributaires) et d'intégrer les deux bases de données réseaux (BD Topo et BD Carto) afin de construire un jeu de données adapté aux besoins du modèle ABSUM.

En effet, nous avons dans notre approche méthodologique, découvert dans un premier temps que les raisons des erreurs de calcul d'itinéraires avec les données de la BD Carto étaient la non-connexité du réseau au niveau des zones périphériques du territoire et des tronçons de routes, accessibles en voiture mais ayant des valeurs de vitesses nulles. Pour pallier ces deux premiers problèmes, nous avons décidé d'établir une zone tampon de 1 kilomètre autour du territoire, ce qui a permis de rendre accessible les points OD situés en périphérie du territoire. La deuxième erreur a été corrigée à l'aide d'un script python, qui attribue automatiquement des valeurs de vitesse aux tronçons dont la vitesse de circulation est nulle, en fonction des vitesses de leurs voisins directement connectés. Cela nous a permis d'avoir 99,23 % des itinéraires calculés sur le réseau orienté et 99,99 % sur le réseau non orienté.

Cependant, nous avons supposé que le pourcentage restant, c'est-à-dire 0,77 % des itinéraires non calculé étaient dus aux degrés de simplification de la BD Carto et l'intégration de la BD Topo pourrait résoudre ce problème. Bien vrai que cela a permis le calcul d'itinéraires entre des paires de points OD, qui ne l'était pas dans la BD Carto, elle a malheureusement engendré des itinéraires non calculé, liée aux mêmes causes qu'avec les données de la BD Carto avant intégration.

Nous n'avons malheureusement pas trouvé, bien que cela soit possible, de solution pour résoudre ce problème, lié au sens de circulation. Nous avons donc proposé de comparer le réseau fusionné avec celui de la BD Carto pour évaluer sa pertinence pour la suite des simulations dans le modèle ABSUM. Les résultats : 40 % d'entités de tronçons de routes de plus que la BD Carto, 15 minutes de temps de calcul d'itinéraires de plus que la BD Carto, et des temps de trajets et de distance pratiquement identiquement à ceux de la BD Carto, nous ont amené à se poser la question de la pertinence d'intégrer le réseau de la BD Topo aux données de la BD Carto. Face à ces résultats, nous pensons que cela en vaut la peine, car le niveau de détail de la BD Carto ne permettra pas de simuler les mobilités actives.

Il serait aussi possible dans la continuité de ce travail, d'étudier l'exhaustivité sémantique des réseaux de routes (BD Carto et BD Topo) afin de s'assurer qu'ils représentent fidèlement la réalité, avant de repenser à affiner si besoin notre méthodologie.

Bibliographie

Blond N., Salze P., Kohler F., Hirtzel J., Fruiquiere M., Othmane H., Hawkins J., Grigorovschi A., Skupinski G., Piombin A., Conesa A., Finance O., Enaux C., Prospeck E., Bassett M., Bindel S., Drouhin F., Ledy J. et Ould Abdeslam D., 2022, *Modelling to support the evaluation of the sustainability of the mobility systems*. SuMo-Rhine.

Bocher E., Petit G. et Lecoeuvre M., 2014, 31 octobre, *H2Network : un outil pour la modélisation et l'analyse de graphes dans le Système d'Information Géographique OrbisGIS* (report, IRSTV FR CNRS 2488 ; IFSTTAR). Repéré à <https://shs.hal.science/halshs-01133333>

Daneshgar F., Sadabadi K. F. et Haghani A., 2018, A Conflation Methodology for Two GIS Roadway Networks and Its Application in Performance Measurements, *Transportation Research Record*, 2672(45), p. 284-293. DOI : [10.1177/0361198118793000](https://doi.org/10.1177/0361198118793000)

Devogele T., 2002, A New Merging Process for Data Integration Based on the Discrete Fréchet Distance, in Richardson D. E. et Van Oosterom P. (dir.), *Advances in Spatial Data Handling*. Berlin, Heidelberg, Springer Berlin Heidelberg, p.167-181. DOI : [10.1007/978-3-642-56094-1_13](https://doi.org/10.1007/978-3-642-56094-1_13)

Dieffenbacher H., 2024, *Intégration de données de réseaux routiers dans le modèle ABSUM (Agent-Based Simulation of Urban Mobility)* ([Mémoire de fin d'étude]). Strasbourg : Université de Strasbourg.

Gräbener T., 2010, 15 novembre, *Calcul d'itinéraire multimodal et multiobjectif en milieu urbain* (phdthesis, Université des Sciences Sociales - Toulouse I). Repéré à <https://theses.hal.science/tel-00553335>

Javaudin L. et de Palma A., 2024, METROPOLIS2: Bridging Theory and Simulation in Agent-Based Transport Modeling. Repéré à <https://thema.u-cergy.fr/IMG/pdf/2024-03.pdf>

Jiang B. et Claramunt C., 2004, Topological Analysis of Urban Street Networks, *Environment and Planning B: Planning and Design*, 31(1), p. 151-162. DOI : [10.1068/b306](https://doi.org/10.1068/b306)

Mustière S. et Devogele T., 2008, Matching Networks with Different Levels of Detail, *GeoInformatica*, 12(4), p. 435-453. DOI : [10.1007/s10707-007-0040-1](https://doi.org/10.1007/s10707-007-0040-1)

Petit G. et Bocher E., 2020, OrbisGIS Manual Documentation. Repéré à <http://orbisgis.org/>

Viry M., Giraud T., Guérois M., Ysebaert R., Lambert N. et Feredj A., 2016, *Généralités liées à OpenStreetMap et la complétude des données* ([Research Report]). RIATE - Réseau interdisciplinaire pour l’Aménagement et la Cohésion des Territoires de l’Europe et de ses voisinages CNRS - CGET - Université Paris Diderot. Repéré à <https://hal.science/hal-03589030>

Zhang M., Yao W. et Meng L., 2016, Automatic and Accurate Conflation of Different Road-Network Vector Data towards Multi-Modal Navigation, *ISPRS International Journal of Geo-Information*, 5(5), p. 68. DOI : [10.3390/ijgi5050068](https://doi.org/10.3390/ijgi5050068)

Zhang X. et Chen M., 2023, Methodology for Conflating Large-Scale Roadway Networks, *Transportation Research Record*, 2677(3), p. 189-202. DOI : [10.1177/03611981221115085](https://doi.org/10.1177/03611981221115085)

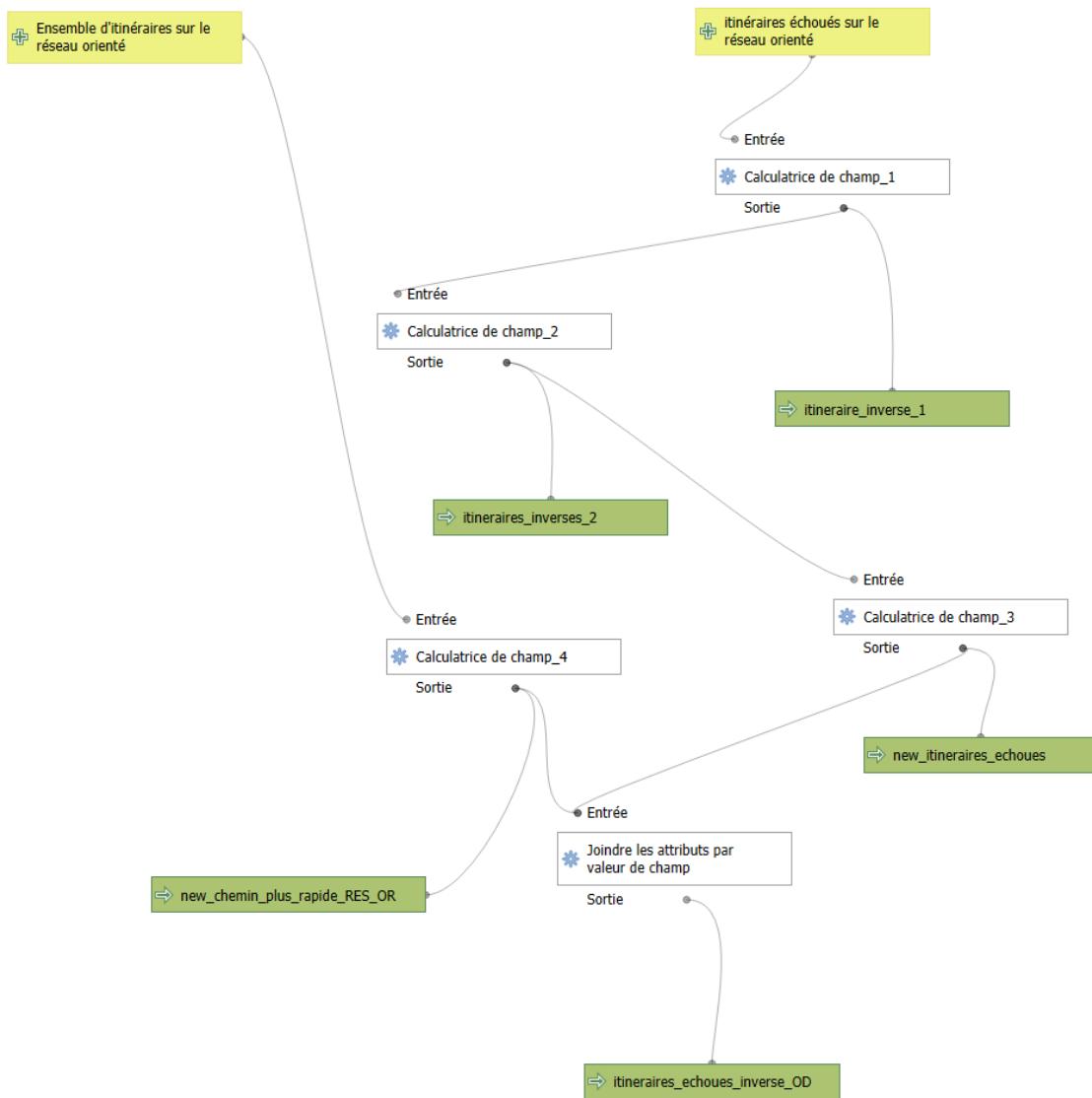
Autres sources :

BD CARTO® | Géoservices, (s. d.). Repéré à <https://geoservices.ign.fr/bdcarto>

BD TOPO® | Géoservices, (s. d.). Repéré à <https://geoservices.ign.fr/bdtopo>

Annexe

Annexe 1 : Processus de détection d'erreurs de non calcul d'itinéraires mis en place à partir du ModelBuilder de QGIS.



Les couches d'entrée du modèle sont l'ensemble des itinéraires générés par l'outil QNEAT3 sur le réseau orienté, ainsi que les itinéraires non calculés sur le même réseau (allant des points d'origine aux points de destination). Grâce aux étapes de traitement, nous obtenons en sortie une couche vectorielle des itinéraires calculés ou non entre les mêmes paires de points OD que ce en entrée mais cette fois ci du point de destination au point d'origine. Cela nous permet de savoir dans un premier temps quels sont les itinéraires calculés dans une seule direction ou dans aucune direction, avant d'effectuer quelques vérifications manuelles.

Annexe 2 : Script Python de la correction attributaire des valeurs de vitesse de circulation du réseau

```
from qgis.QtCore import QVariant
from qgis.core import *

# Paramètres

nom_couche = "bd_carto_routes" # nom du réseau à corriger
champ_vitesse = "VIT_MOY_VL" # champ vitesse BD Carto
champ_correction = "VIT_CORR" # nouveau champ vitesse

# Récupération de la couche du réseau routier (il faut que la couche du réseau routier soit chargée dans le projet QGIS)

layer = QgsProject.instance().mapLayersByName(nom_couche)[0]
if not layer:
    raise Exception(f"La couche '{nom_couche}' est introuvable.")

# Création de l'index spatial

index = QgsSpatialIndex(layer.getFeatures())

# Création du champ VIT_CORR si absent

if layer.fields().indexFromName(champ_correction) == -1:
    layer.dataProvider().addAttributes([QgsField(champ_correction, QVariant.Double)])
    layer.updateFields()

# Initialisation du champ VIT_CORR avec le champ VIT_MOY_VL

with edit(layer):
    for f in layer.getFeatures():
        f[champ_correction] = f[champ_vitesse]
        layer.updateFeature(f)
# Fonction : plus petite vitesse parmi les voisins directs

def min_vitesse_voisins(feature):
    geom = feature.geometry()
    voisins_ids = index.intersects(geom.boundingBox())
    vitesses_voisines = []
    for vid in voisins_ids:
        if vid != feature.id():
            voisin = layer.getFeature(vid)
            if voisin.geometry().touches(geom):
                v = voisin[champ_correction]
                if v and v > 0:
```

```

        vitesses_voisines.append(v)
    if vitesses_voisines:
        return min(vitesses_voisines)
    else:
        return None

# Liste des identifiants à corriger

troncons_a_corriger = [f.id() for f in layer.getFeatures() if
f[champ_vitesse] == 0]

# Boucle itérative jusqu'à stabilisation

iteration = 0
modification = True

while modification:
    iteration += 1
    modification = False

    with edit(layer):
        for fid in troncons_a_corriger:
            feat = layer.getFeature(fid)
            if feat[champ_correction] == 0:
                v_min = min_vitesse_voisins(feat)
                if v_min:
                    feat[champ_correction] = v_min
                    layer.updateFeature(feat)
                    modification = True

# Résultat final

reste = [f.id() for f in layer.getFeatures() if
f[champ_vitesse] == 0 and f[champ_correction] == 0]
print("Correction terminée." if not reste else f"{len(reste)}"
tronçon(s) non corrigé(s) après {iteration} itération(s).")

```

Annexe 3 : script permettant d'identifier et de supprimer les tronçons de routes déjà numérisé dans la BD Carto sur la base d'un attribut commun aux deux bases de données (ID pour la BD Topo et ID_BDTOPO pour la BD Carto).

```
from qgis.core import *
from qgis.utils import iface
from PyQt5.QtWidgets import QProgressDialog
from PyQt5.QtCore import Qt

# Paramètres

nom_couche_carto = "reseau_connexe_accesible_vehicule"      # Nom
de la couche BD CARTO
nom_couche_topo = "Reseau_TOPO_EMS_nettoye_3"          # Nom de la
couche BD TOPO
champ_carto = "ID_BDTOPO"                      # Champ dans des
identifiants des entités de la BD Topo dans la BD CARTO
(ID_BDTOPO)
champ_topo = "ID"                            # Champ des identifiants des
entités dans la BD TOPO

# Récupération des couches

couche_carto =
QgsProject.instance().mapLayersByName(nom_couche_carto)[0]
couche_topo =
QgsProject.instance().mapLayersByName(nom_couche_topo)[0]

if not couche_carto or not couche_topo:
    raise Exception("L'une des deux couches est introuvable.")

# Récupérer les valeurs du champ de BD CARTO (et éclatent les
identifiants de type AA/BB)

valeurs_carto = set()
for f in couche_carto.getFeatures():
    val = f[champ_carto]
    if val:
        sous_ids = [v.strip() for v in str(val).split("/")]
        valeurs_carto.update(sous_ids)

print(f"{len(valeurs_carto)} identifiants uniques extraits de
la BD CARTO (champ '{champ_carto}').")

# Initialisation de la barre de progression

nb_total = couche_topo.featureCount()
progress = QProgressDialog("Suppression des entités...",",
"Annuler", 0, nb_total)
progress.setWindowTitle("Progression")
```

```

progress.setWindowModality(Qt.WindowModal)
progress.show()

# Identifier les entités à supprimer dans la BD TOPO

ids_supprimer = []
for i, f in enumerate(couche_topo.getFeatures()):
    if f[champ_topo] in valeurs_carto:
        ids_supprimer.append(f.id())
    progress.setValue(i)
    if progress.wasCanceled():
        iface.messageBar().pushMessage("Annulé", "Suppression annulée par l'utilisateur.", level=Qgis.Warning)
        break

progress.setValue(nb_total)

# Suppression des entités dans la BD TOPO

if ids_supprimer:
    with edit(couche_topo):

        couche_topo.dataProvider().deleteFeatures(ids_supprimer)
        iface.messageBar().pushMessage(
            "Suppression terminée",
            f"{len(ids_supprimer)} entité(s) supprimée(s) de la BD TOPO.",
            level=Qgis.Success,
            duration=6
        )
else:
    iface.messageBar().pushMessage(
        "Aucune entité supprimée",
        "Aucune entité de la BD TOPO ne correspond aux identifiants extraits de la BD CARTO.",
        level=Qgis.Info,
        duration=5
    )

```

Annexe 4 : Processus d'intégration de la BD Topo à la BD Carto (format PyGIS) prêt à être intégré dans le ModelBuilder de QGIS.

```
"""
Model exported as python.
Name :
Group :
With QGIS : 34007
"""

from qgis.core import QgsProcessing
from qgis.core import QgsProcessingAlgorithm
from qgis.core import QgsProcessingMultiStepFeedback
from qgis.core import QgsProcessingParameterVectorLayer
from qgis.core import QgsProcessingParameterFeatureSink
import processing

class (QgsProcessingAlgorithm):

    def initAlgorithm(self, config=None):

        self.addParameter(QgsProcessingParameterVectorLayer('bd_topo',
        'BD_TOPO', types=[QgsProcessing.TypeVectorLine],
        defaultValue=None))

        self.addParameter(QgsProcessingParameterVectorLayer('ems',
        'EMS ', types=[QgsProcessing.TypeVectorPolygon],
        defaultValue=None))

        self.addParameter(QgsProcessingParameterVectorLayer('reseau_ca
rto_ems', 'Reseau_Carto_EMS', defaultValue=None))

        self.addParameter(QgsProcessingParameterFeatureSink('Tampon_1k
m_ems', 'Tampon_1km_EMSS',
        type=QgsProcessing.TypeVectorPolygon, createByDefault=True,
        supportsAppend=True, defaultValue=None))

        self.addParameter(QgsProcessingParameterFeatureSink('Reseau_to
po_ems_nettoye_3', 'Reseau_TOPO_EMSS_nettoye_3',
        type=QgsProcessing.TypeVectorAnyGeometry,
        createByDefault=True, defaultValue=None))

        self.addParameter(QgsProcessingParameterFeatureSink('Regroupe_
rond_point', 'Regroupe_Rond_Point',
        type=QgsProcessing.TypeVectorAnyGeometry,
        createByDefault=True, defaultValue='TEMPORARY_OUTPUT'))

        self.addParameter(QgsProcessingParameterFeatureSink('Reseau_to
po_ems_nettoye', 'Reseau_TOPO_EMSS_nettoye',
```

```

type=QgsProcessing.TypeVectorAnyGeometry,
createByDefault=True, defaultValue='TEMPORARY_OUTPUT')))

self.addParameter(QgsProcessingParameterFeatureSink('Rond_poin
t', 'Rond_Point', type=QgsProcessing.TypeVectorAnyGeometry,
createByDefault=True, defaultValue=None))

self.addParameter(QgsProcessingParameterFeatureSink('Vrai_fusi
on_bd_carto_topo', 'Vrai_Fusion_BD_Carto_Topo',
type=QgsProcessing.TypeVectorAnyGeometry,
createByDefault=True, defaultValue=None))

self.addParameter(QgsProcessingParameterFeatureSink('Reseau_to
po_ems', 'Reseau_TOPO_EMSS',
type=QgsProcessing.TypeVectorAnyGeometry,
createByDefault=True, defaultValue=None))

self.addParameter(QgsProcessingParameterFeatureSink('Rond_poin
t_a_supprimer', 'Rond_point_a_supprimer',
type=QgsProcessing.TypeVectorAnyGeometry,
createByDefault=True, defaultValue=None))

self.addParameter(QgsProcessingParameterFeatureSink('Reseau_to
po_ems_nettoye_2', 'Reseau_TOPO_EMSS_nettoye_2',
type=QgsProcessing.TypeVectorAnyGeometry,
createByDefault=True, defaultValue=None))

self.addParameter(QgsProcessingParameterFeatureSink('Segment_c
roise_a_supprimer', 'segment_croise_A_supprimer',
type=QgsProcessing.TypeVectorAnyGeometry,
createByDefault=True, defaultValue=None))

self.addParameter(QgsProcessingParameterFeatureSink('Reseau_fu
sione_composante', 'reseau_fusione_composante', optional=True,
type=QgsProcessing.TypeVectorAnyGeometry,
createByDefault=True, defaultValue=None))

    def processAlgorithm(self, parameters, context,
model_feedback):
        # Use a multi-step feedback, so that individual child
algorithm progress reports are adjusted for the
        # overall progress through the model
        feedback = QgsProcessingMultiStepFeedback(11,
model_feedback)
        results = {}
        outputs = {}

        # Tampon
        alg_params = {
            'DISSOLVE': False,
            'DISTANCE': 1000,

```

```

        'END_CAP_STYLE': 0, # Rond
        'INPUT': parameters['ems'],
        'JOIN_STYLE': 0, # Rond
        'MITER_LIMIT': 2,
        'SEGMENTS': 5,
        'SEPARATE_DISJOINT': False,
        'OUTPUT': parameters['Tampon_1km_ems']
    }
    outputs['Tampon'] = processing.run('native:buffer',
alg_params, context=context, feedback=feedback,
is_child_algorithm=True)
    results['Tampon_1km_ems'] =
outputs['Tampon']['OUTPUT']

    feedback.setCurrentStep(1)
    if feedback.isCanceled():
        return {}

    # Extraire par localisation_1
    alg_params = {
        'INPUT': parameters['bd_topo'],
        'INTERSECT': outputs['Tampon']['OUTPUT'],
        'PREDICATE': [0,6], # intersecte, est à
l'intérieur
        'OUTPUT': parameters['Reseau_topo_ems']
    }
    outputs['ExtraireParLocalisation_1'] =
processing.run('native:extractbylocation', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
    results['Reseau_topo_ems'] =
outputs['ExtraireParLocalisation_1']['OUTPUT']

    feedback.setCurrentStep(2)
    if feedback.isCanceled():
        return {}

    # Extraire par expression_1
    alg_params = {
        'EXPRESSION': 'NOT ("NATURE" = \'Sentier\' OR
"ACCES_VL" = \'Physiquement impossible\' OR "SENS" = \'Sans
objet\' OR "ETAT" = \'En projet\' OR "ETAT" = \'En
construction\' OR "ETAT" = \'En projet\' OR "ETAT" = \'En
construction\' OR "NUMERO" is not null OR "NOM_COLL_D" is null
OR "NOM_COLL_G" is null OR "IMPORTANCE" is not \'5\')',
        'INPUT':
outputs['ExtraireParLocalisation_1']['OUTPUT'],
        'OUTPUT': parameters['Reseau_topo_ems_nettoye']
    }
    outputs['ExtraireParExpression_1'] =
processing.run('native:extractbyexpression', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)

```

```

        results['Reseau_topo_ems_nettoye'] =
outputs['ExtraireParExpression_1']['OUTPUT']

        feedback.setCurrentStep(3)
        if feedback.isCanceled():
            return {}

# Extraire par expression_2
alg_params = {
    'EXPRESSION': '"NATURE" = \'Rond-point\'',
    'INPUT':
outputs['ExtraireParExpression_1']['OUTPUT'],
    'OUTPUT': parameters['Rond_point']
}
outputs['ExtraireParExpression_2'] =
processing.run('native:extractbyexpression', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
    results['Rond_point'] =
outputs['ExtraireParExpression_2']['OUTPUT']

        feedback.setCurrentStep(4)
        if feedback.isCanceled():
            return {}

# Regrouper
alg_params = {
    'FIELD': [''],
    'INPUT':
outputs['ExtraireParExpression_2']['OUTPUT'],
    'SEPARATE_DISJOINT': False,
    'OUTPUT': parameters['Regroupe_rond_point']
}
outputs['Regrouper'] =
processing.run('native:dissolve', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)
    results['Regroupe_rond_point'] =
outputs['Regrouper']['OUTPUT']

        feedback.setCurrentStep(5)
        if feedback.isCanceled():
            return {}

# Extraire par localisation_2
alg_params = {
    'INPUT': outputs['Regrouper']['OUTPUT'],
    'INTERSECT': parameters['reseau_carto_ems'],
    'PREDICATE': [0,4,5,7], #
intersecte,touche,chevauche,croise
    'OUTPUT': parameters['Rond_point_a_supprimer']
}

```

```

        outputs['ExtraireParLocalisation_2'] =
processing.run('native:extractbylocation', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
    results['Rond_point_a_supprimer'] =
outputs['ExtraireParLocalisation_2']['OUTPUT']

    feedback.setCurrentStep(6)
    if feedback.isCanceled():
        return {}

    # Extraire par localisation_3
    alg_params = {
        'INPUT':
outputs['ExtraireParExpression_1']['OUTPUT'],
        'INTERSECT':
outputs['ExtraireParLocalisation_2']['OUTPUT'],
        'PREDICATE': [2], # est disjoint
        'OUTPUT': parameters['Reseau_topo_ems_nettoye_2']
    }
    outputs['ExtraireParLocalisation_3'] =
processing.run('native:extractbylocation', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
    results['Reseau_topo_ems_nettoye_2'] =
outputs['ExtraireParLocalisation_3']['OUTPUT']

    feedback.setCurrentStep(7)
    if feedback.isCanceled():
        return {}

    # Extraire par localisation_4
    alg_params = {
        'INPUT':
outputs['ExtraireParLocalisation_3']['OUTPUT'],
        'INTERSECT': parameters['reseau_carto_ems'],
        'PREDICATE': [7], # croise
        'OUTPUT': parameters['Segment_croise_a_supprimer']
    }
    outputs['ExtraireParLocalisation_4'] =
processing.run('native:extractbylocation', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
    results['Segment_croise_a_supprimer'] =
outputs['ExtraireParLocalisation_4']['OUTPUT']

    feedback.setCurrentStep(8)
    if feedback.isCanceled():
        return {}

    # Différence
    alg_params = {
        'GRID_SIZE': None,

```

```

    'INPUT' :
outputs['ExtraireParLocalisation_3']['OUTPUT'],
    'OVERLAY' :
outputs['ExtraireParLocalisation_4']['OUTPUT'],
    'OUTPUT': parameters['Reseau_topo_ems_nettoye_3']
}
outputs['Diffrence'] =
processing.run('native:difference', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
results['Reseau_topo_ems_nettoye_3'] =
outputs['Diffrence']['OUTPUT']

feedback.setCurrentStep(9)
if feedback.isCanceled():
    return {}

return results

def name(self):
    return ''

def displayName(self):
    return ''

def group(self):
    return ''

def groupId(self):
    return ''

def createInstance(self):
    return ()

```