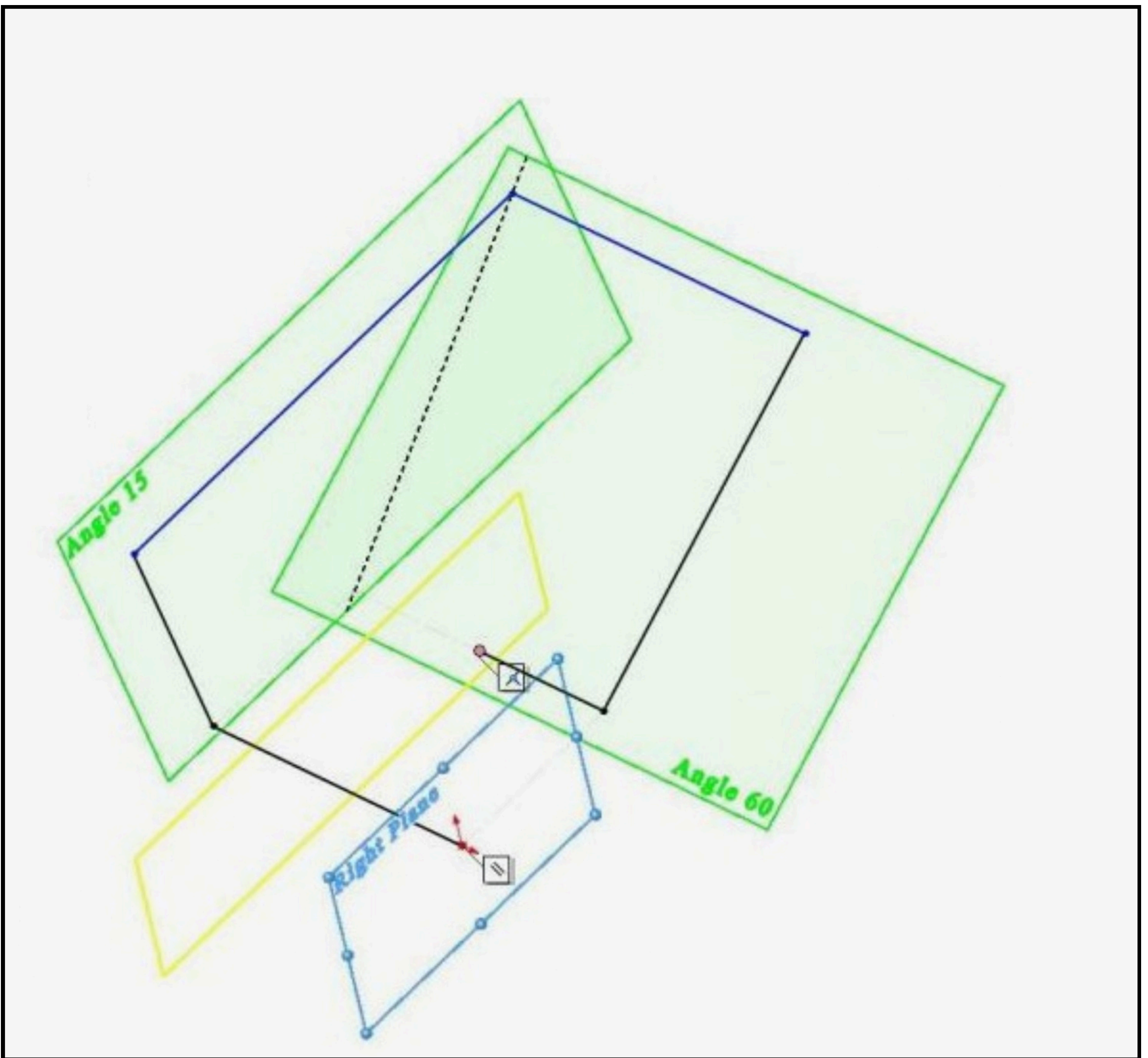


TD8: Topologie : Compréhension et réparation des défauts



Introduction

La topologie sur les Systèmes d'Information Géographique joue un rôle essentiel dans la gestion et l'analyse des données spatiales. Elle permet de garantir la cohérence géométrique des entités vectorielles en évitant les erreurs telles que les superpositions, les discontinuités ou les intersections incorrectes. Dans ce TD, nous explorerons les méthodes de compréhension et de correction des défauts topologiques à travers divers logiciels et outils.

Nous aborderons tout d'abord l'utilisation des outils de topologie sur les logiciels QGIS et ArcGIS, en mettant en évidence leurs capacités à identifier et réparer les erreurs courantes. Ensuite, nous nous intéresserons à des solutions plus avancées en exploitant les bases de données spatiales via PostGIS , ce qui permet d'automatiser la détection et la correction des erreurs par des requêtes SQL directement en base de données. Enfin, nous verrons comment FME peut être utilisé pour traiter et corriger efficacement des géométries défectueuses à grandes échelles.

L'objectif de ce TD est donc d'acquérir une compréhension approfondie des erreurs topologiques et de maîtriser les différentes techniques et outils disponibles pour assurer une qualité optimale des données géospatiales.

Sommaire

Partie 1: Topologie de ligne

A / Sur Qgis

B / Sur Arcgis

Partie 2: Réparation et vérification des mauvaises topologies

A / Sur Qgis

B / Réparation sur Arcgis via la fonction Intégrer

Partie 3 : Utilisation de PostGis

A / Identification des erreurs

B / Correction des erreurs

Partie 4 : Utilisation des outils FME

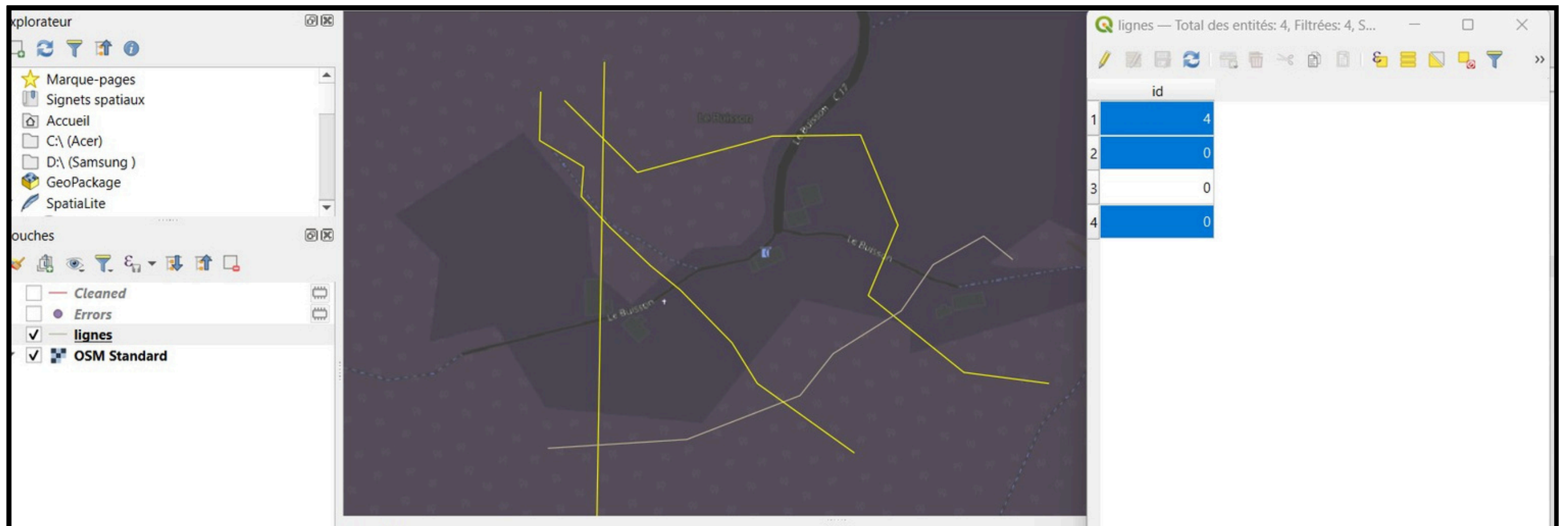
Conclusion

Partie 1: topologie de ligne

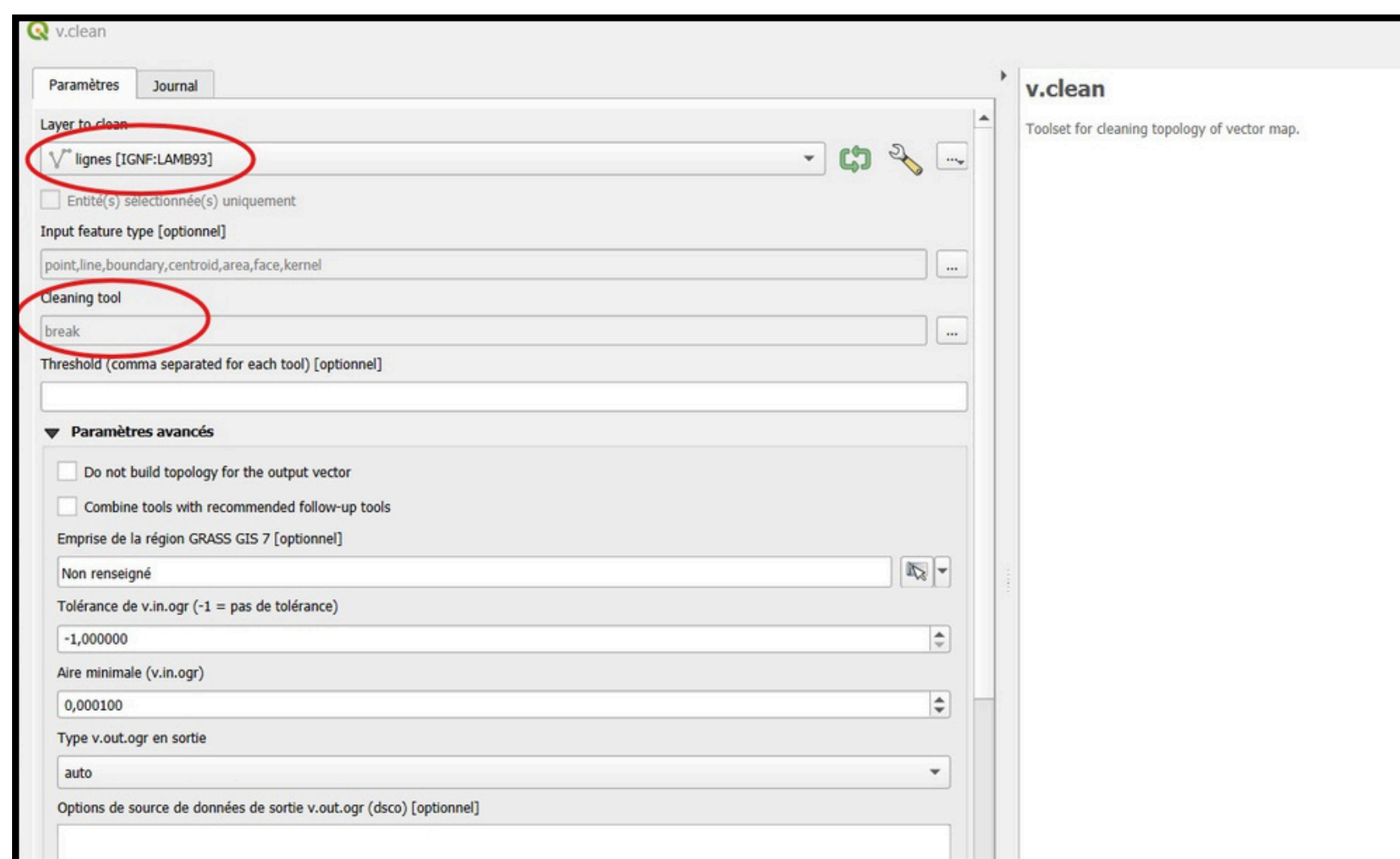
L'objectif de cette partie est d'obtenir un graphe planaire c'est-à-dire une couche où toutes les lignes se croisent avec un noeud à leur intersection

A / Sur Qgis

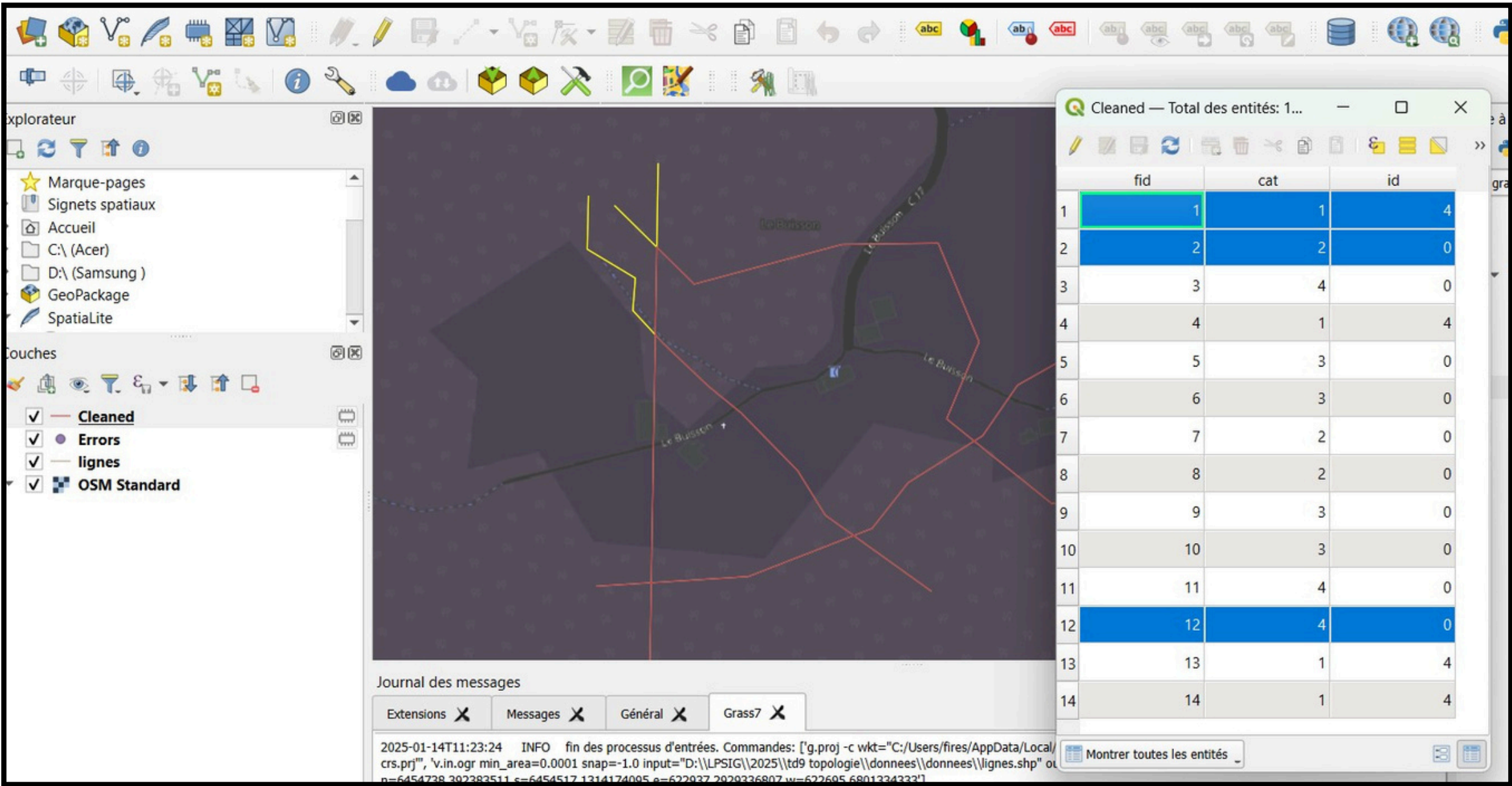
Pour cela nous avons la donnée suivante sur Qgis où on voit bien que seulement 4 entités sont disponibles



Pour obtenir notre résultat il faut utiliser les outils Grass de Qgis ici on utilise l'algorithme v.clean qu'on paramètre sur break



Voici notre résultat on obtient bien une couche où à chaque intersection un nœud a été créé et divise l'entité

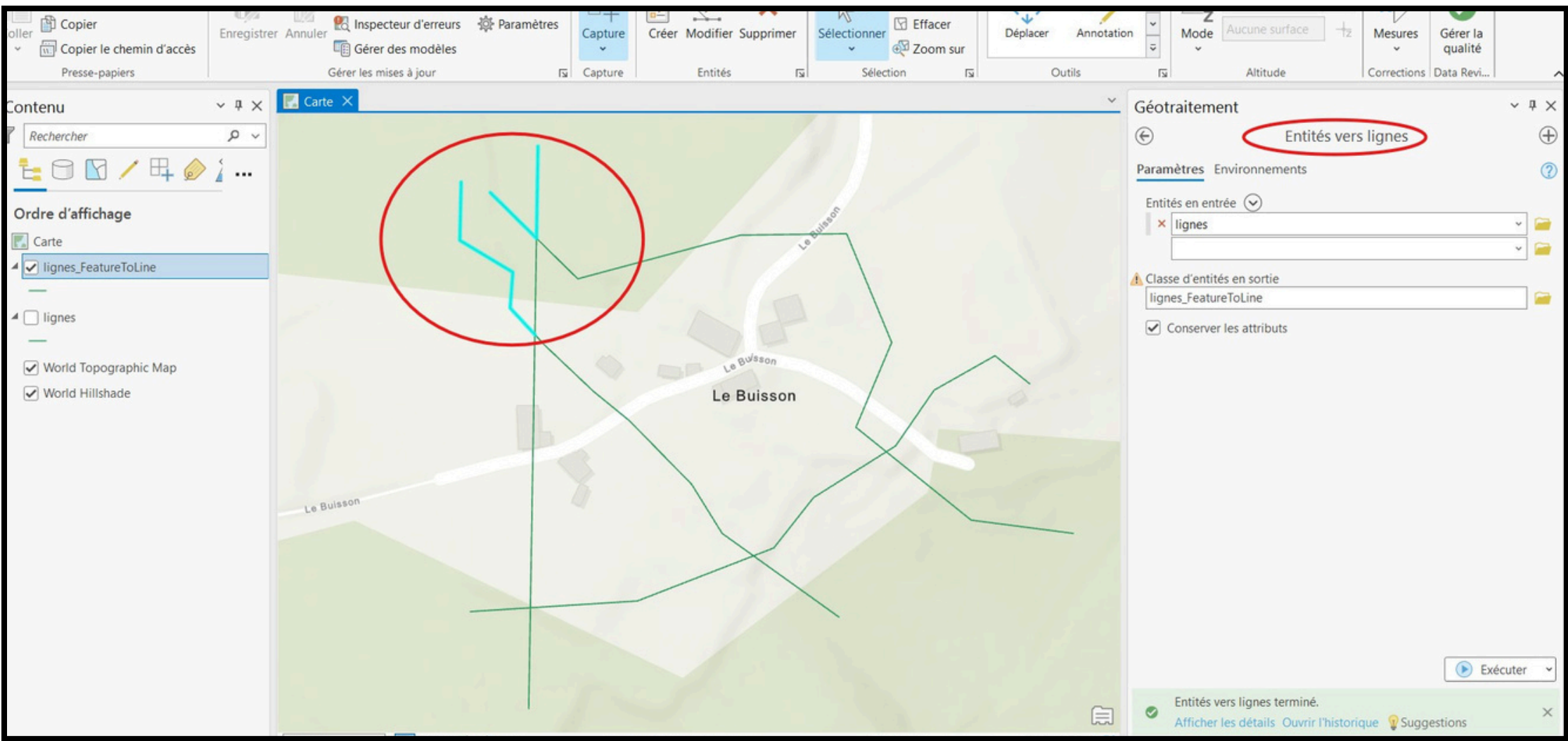


B/ Sur Arcgis

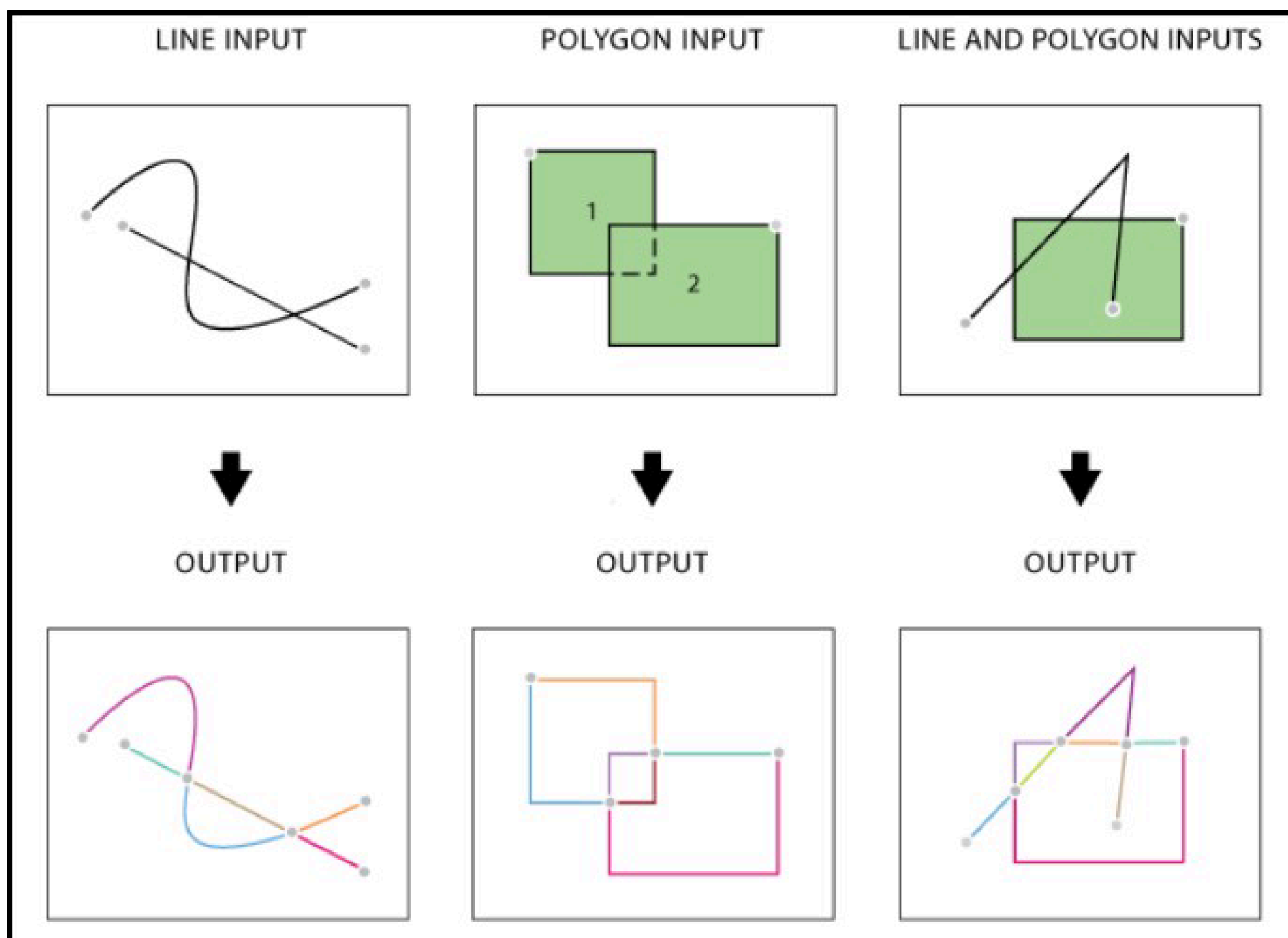
même principe on charge notre donnée, on voit bien qu'il y a seulement 3 entités



Ici on va utiliser l'algorithme entités vers lignes et voici notre résultat



Concrètement voici le traitement que
l'algorithme Arcgis effectue

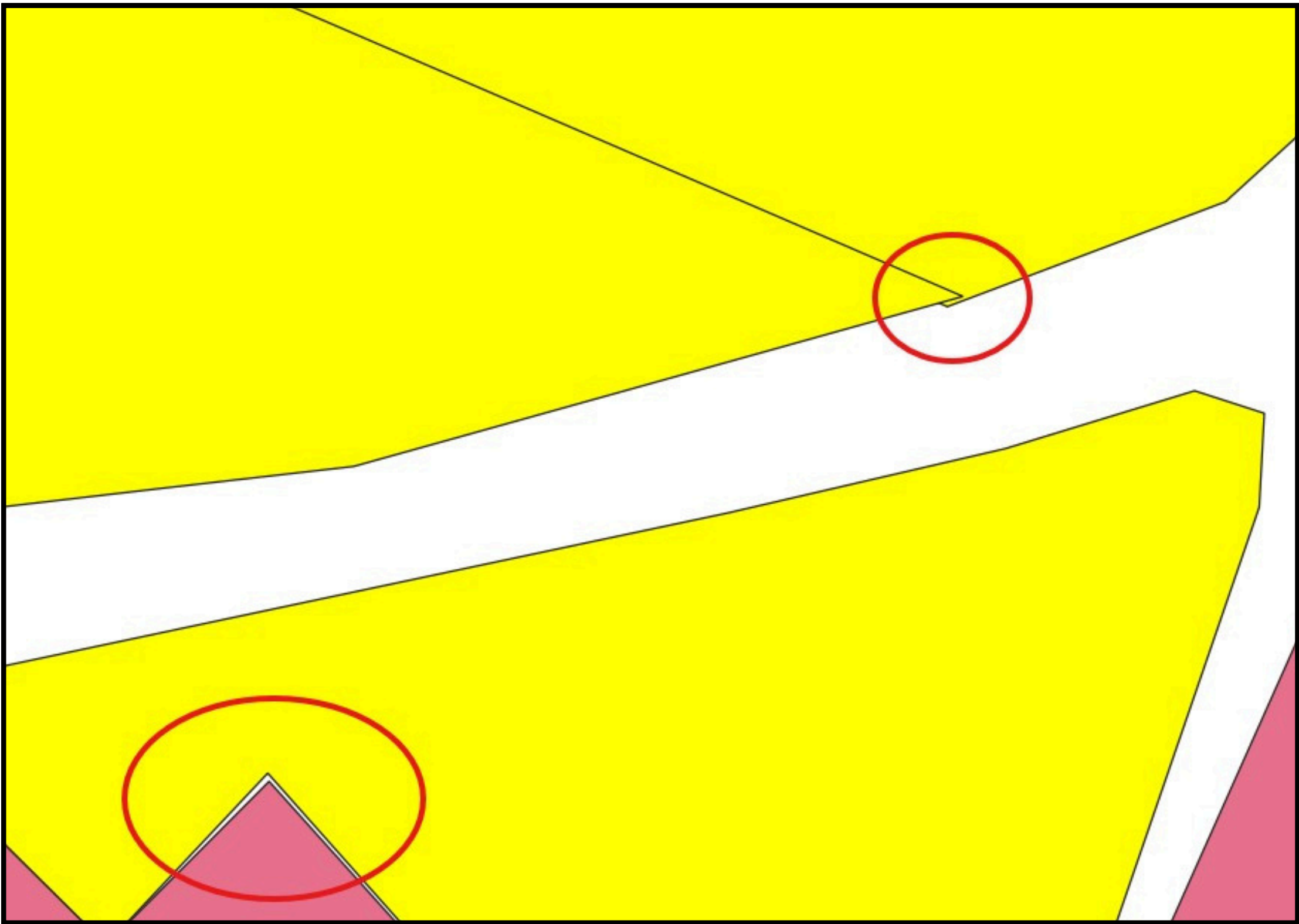


Partie 2: Réparation et vérification des mauvaises topologies

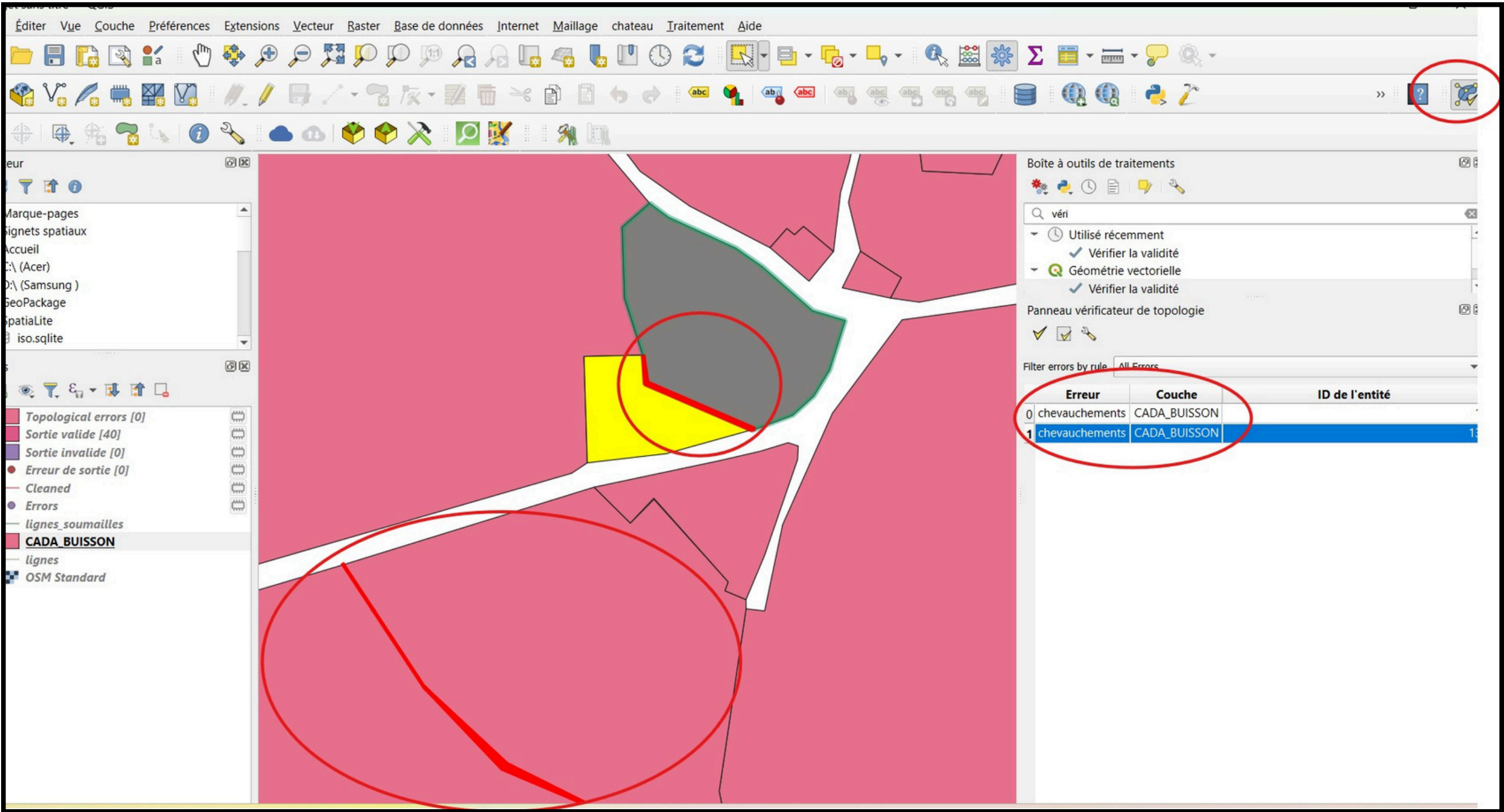
L'objectif de cette partie est d'obtenir une couche
vecteur avec aucun polygones défailants.
(chevauchement, polygone pas fermés)

A / Sur Qgis

Pour cela nous avons la données suivante sur Qgis ou on voit
bien que des entités se chevauchent mais ont également des
trous entres elles



Pour trouver nos problèmes il faut utiliser le vérificateur de topologie de Qgis qui est une extension et non un traitement.



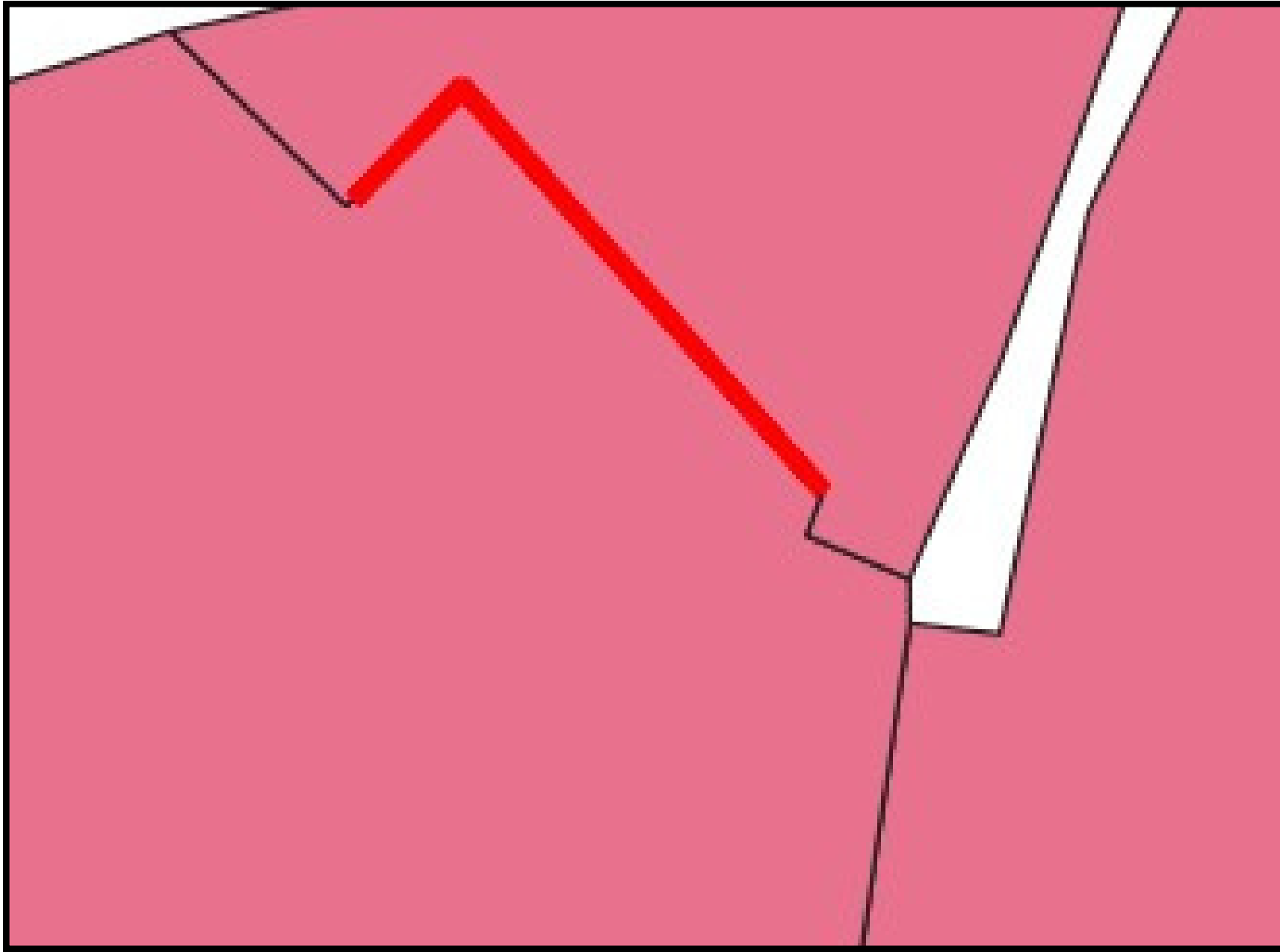
Erreur	Couche	ID de l'entité
0	chevauchements	CADA_BUISSON
1	chevauchements	CADA_BUISSON

On paramètre nos réglages et les règles que l'on souhaite appliquer



Règle	Couche #1	Couche #2
1 <input checked="" type="checkbox"/> ne doit pas avoir de trou	CADA_BUISSON	Pas de couche

et voici le résultat notre trou est a été identifié



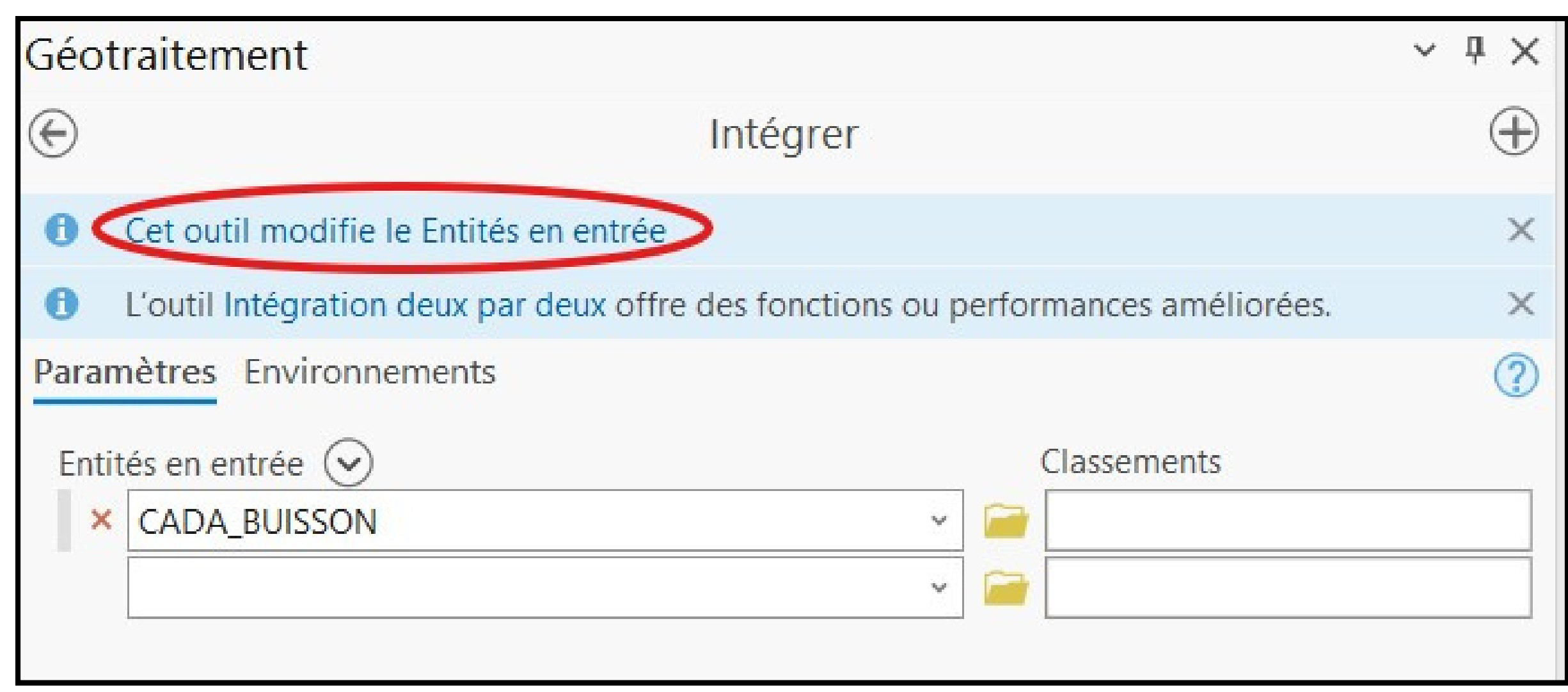
B / Réparation sur Arcgis via la fonction Intégrer

Malheureusement Qgis n'est pas idéal pour corriger ces erreurs. Ainsi on passe sur Arcgis qui a de nombreux outils plus adaptés

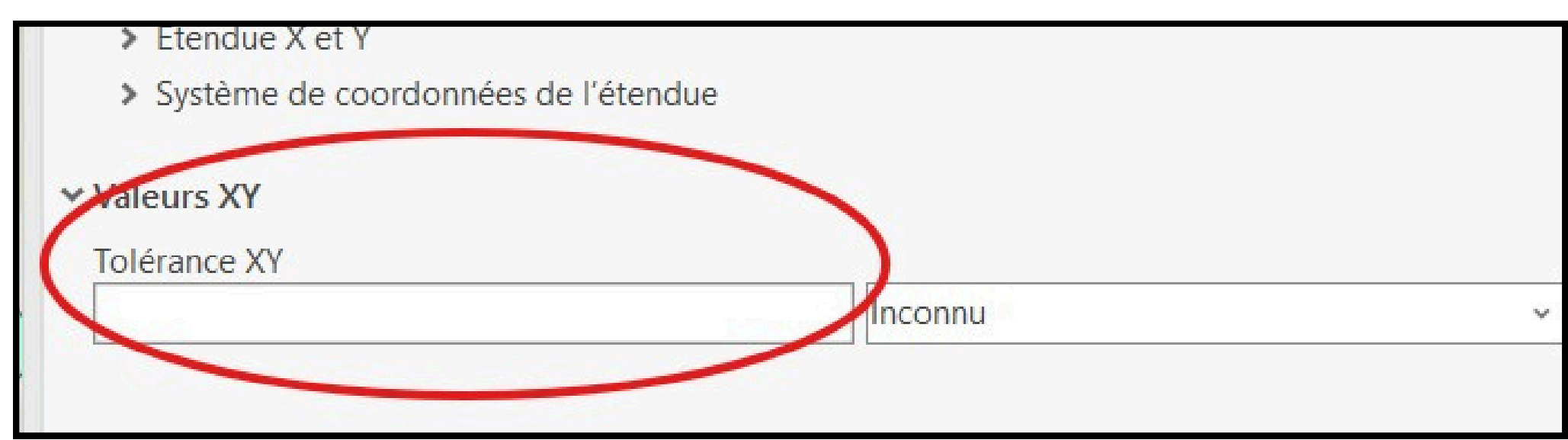
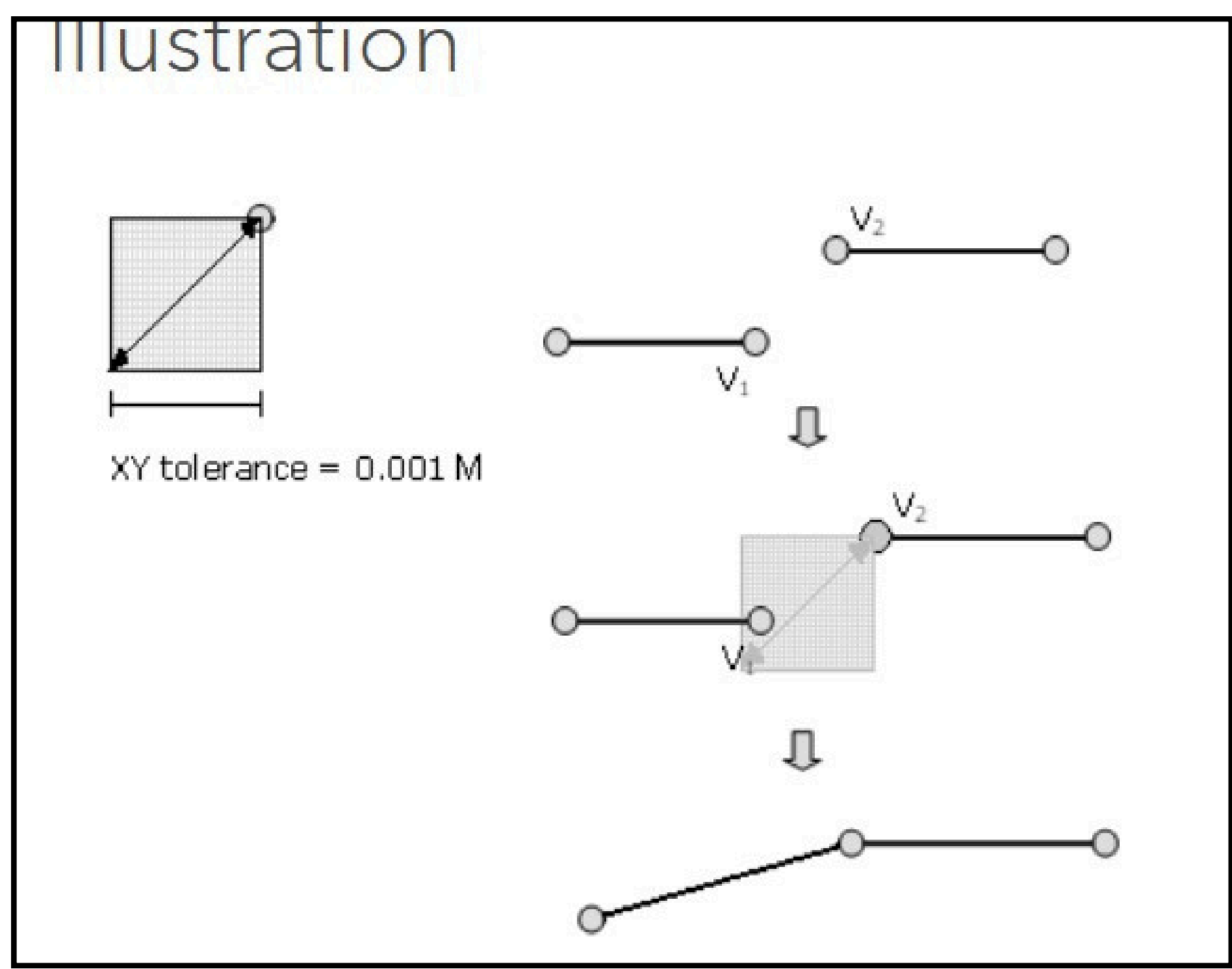
Notre donnée de base est identique



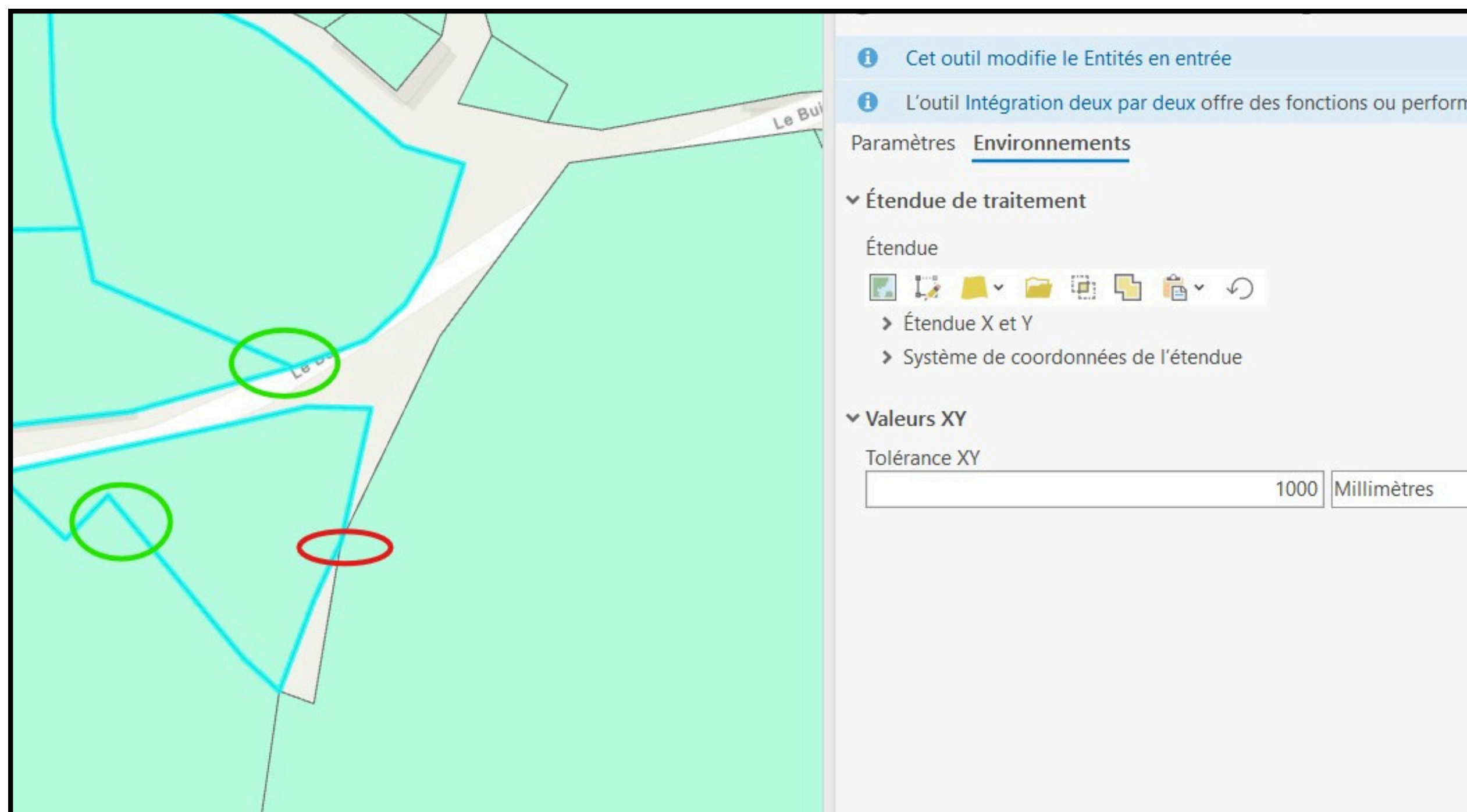
Pour corriger les erreurs de topologie Argis dispose de l'outil intégrer. Cependant il faut bien paramétrer la tolérance et faire attention car l'algorithme ne produit de nouvelle couche et modifie notre donnée.



La tolérance se comprend par rapport au schéma explicatif suivant



Ainsi, ici la tolérance est trop grande donc certes nos erreurs sont corrigés mais l'algorithme produit une autre erreur qui colle 2 entités.



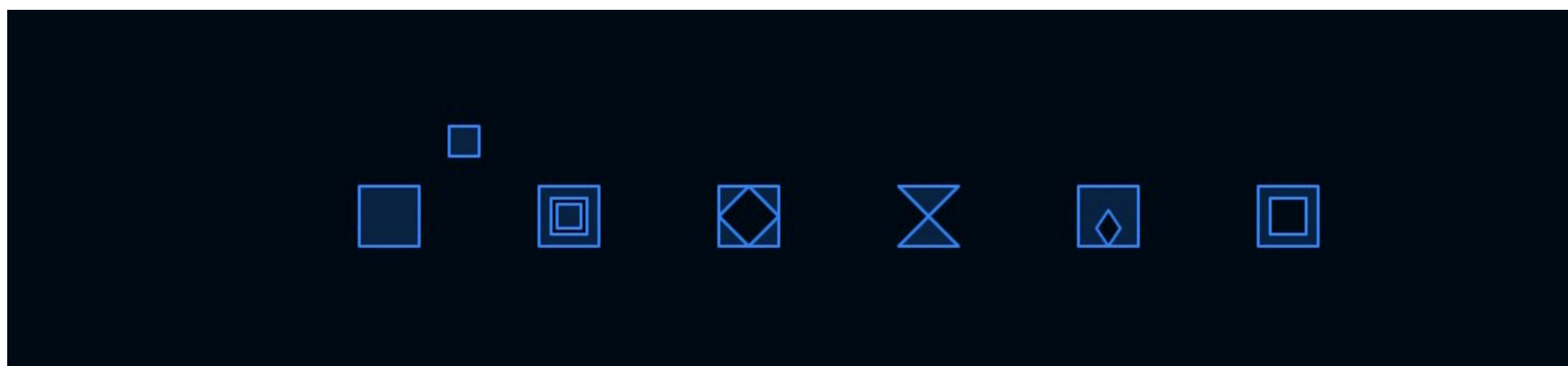
Partie 3 : Utilisation de PostGis

A / Identification des erreurs

D'autres outils que les SIG sont à notre disposition pour corriger les géométries défectueuses.

Une alternative aux solutions précédentes est de modifier nos données directement dans la base de données grâce à des requêtes SQL de Postgis.

Je vais illustrer les possibilités grâce aux données suivantes



Exemples

```
SELECT ST_IsValid(ST_GeomFromText('LINESTRING(0 0, 1 1)')) As good_line,
       ST_IsValid(ST_GeomFromText('POLYGON((0 0, 1 1, 1 2, 1 1, 0 0))')) As bad_poly
--results
NOTICE: Self-intersection at or near point 0 0
good_line | bad_poly
-----+-----
t         | f
```

La premier algorithmme que l'on
peux effectuer est St_isvalid dont la
syntaxe est expliqué au-dessus. Le
traitement nous renvoie un
booléen TRUE ou FALSE.
Ici dans notre exemple les 6 entités
ont des défauts

st_Isvalid	boolean
	false
	false
	false
	false
	false
	false

Un autre algorithmme disponible est st_isvalidreason qui
renvoie la raison de l'erreur en anglais malheureusement

st_isvalidreason(geom)
_invalidgeometry,

Visualisation cartogra... x Aperçu de graphiques x Notifications

SQL

	st_Isvalidreason
	text
	Hole lies outside shell[465015
00C0F98E59410000000034621C41000000C0F98E59410000000034621C41000000C0F88E59410000000024621C410000...	Holes are nested[465033 670
	Interior is disconnected[4650
	Self-intersection[465095 6700
	Ring Self-intersection[465125
	Nested shells[465152 670000

Cet algorithme est très utile et documenté avec ce lien
https://postgis.net/docs/ST_IsValidReason.html

B / Correction des erreurs

Pour effectuer les correction il existe l'outil St_Makevalid dont voici la syntaxe

```
1 UPDATE a000_temp.xx_invalidgeometry
2 SET geom = ST_MakeValid(geom)
3 WHERE NOT ST_IsValid(geom);
4
5
6
7
```

Data Output	Messages	Notifications
NOTICE: Hole lies outside shell at or near point 465015 6700015		
NOTICE: Holes are nested at or near point 465033 6700003		
NOTICE: Interior is disconnected at or near point 465065 6700000		
NOTICE: Self-intersection at or near point 465095 6700005		
NOTICE: Ring Self-intersection at or near point 465125 6700000		
NOTICE: Nested shells at or near point 465152 6700002		
UPDATE 6		
Requête exécutée avec succès en 46 msec.		

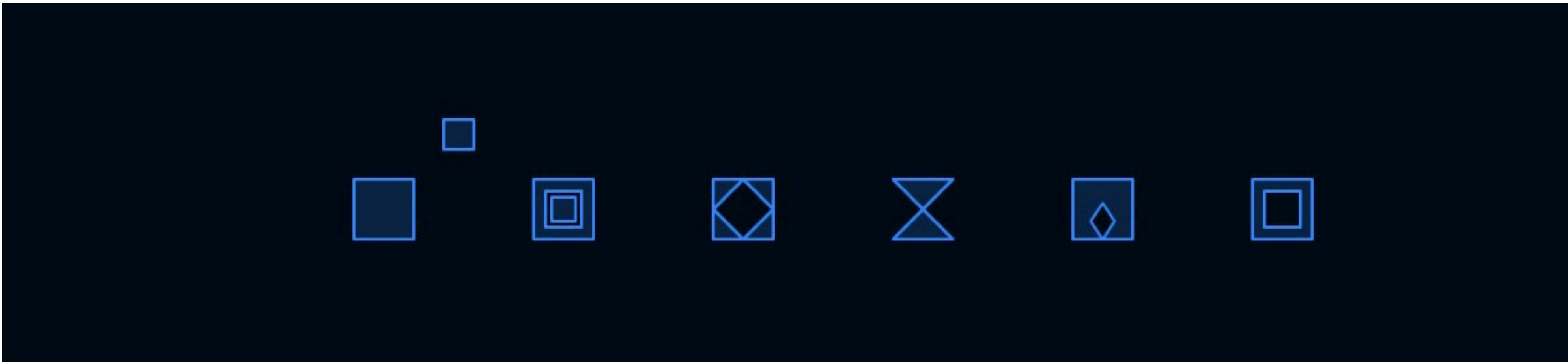
L'algorithme a modifier nos données puisqu'il a renvoyé 6 update en précisant les modifications effectués.

Partie 4 : Utilisation des outils FME

Un autre outil disponible pour corriger les géométries défectueuses est FME.

En effet modifier nos données via FME permet une d'intégrer le processus de vérification dans un traitement.

Pour illustrer cela je vais utiliser ces données invalides disponible dans ma base de données.



Je lance le traitement geometry validator

Visual Preview x Translation Log

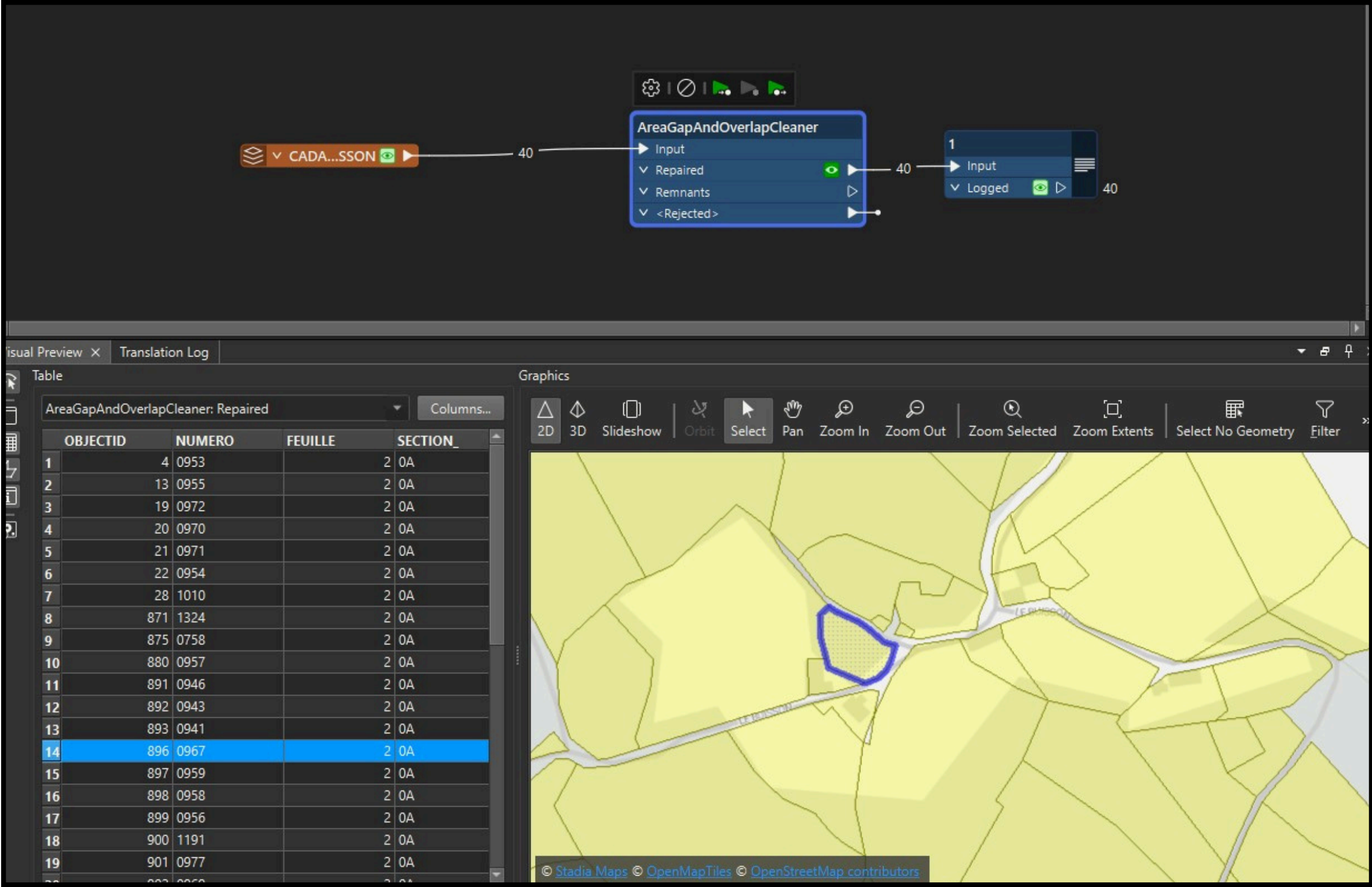
Table

Logger: Logged

id	type	count	details	issue_found
13	5 Ring Self Inters...	<missing>	<missing>	<missing>
14	6 Nested Shells	<missing>	<missing>	<missing>
15	1 Hole Outside S...	<missing>	Hole Outside S...	Fails OGC Comp...
16	1 Hole Outside S...	<missing>	Donut: Hole Ou...	Self-Intersectio...
17	2 Nested Holes	<missing>	Nested Holes	Fails OGC Comp...
18	2 Nested Holes	<missing>	Donut: Nested ...	Self-Intersectio...
19	3 Dis. Interior	<missing>	Miscellaneous	Incorrect Orient...
20	3 Dis. Interior	<missing>	Disconnected I...	Fails OGC Comp...
21	3 Dis. Interior	<missing>	Donut: Disjoint ...	Self-Intersectio...
22	4 Self Intersect.	<missing>	Self Intersection	Fails OGC Comp...
23	4 Self Intersect.	<missing>	Self Intersection	Fails OGC Simple
24	4 Self Intersect.	<missing>	Miscellaneous	Self-Intersectio...
25	5 Ring Self Inters...	<missing>	Self Intersection	Fails OGC Comp...
26	5 Ring Self Inters...	<missing>	Self Intersection	Fails OGC Simple
27	5 Ring Self Inters...	<missing>	Miscellaneous	Self-Intersectio...

J'obtiens un résultat qui filtre mes erreurs et me donne l'erreur ainsi qu'une solution au problème qu'il exécute lui même

Cependant pour les polygones qui se chevauche l'erreur n'est pas détecté car l'algorithme fonctionne entités par entités.



Mais il existe des traitements disponibles tels que Area_Gap_and_overlap_cleaner. Cet algorithme a pour effet de réparer les géométries qui se chevauchent ou s'éloignent

Conclusion

Comme nous l'avons vu la gestion de la topologie est un enjeu clé dans les SIG, garantissant la cohérence et la fiabilité des données géospatiales. Tout au long de ce TD, nous avons exploré différentes approches pour identifier et corriger les erreurs topologiques, en utilisant des outils tels que QGIS, ArcGIS, PostGIS et FME.

Les logiciels SIG offrent des solutions variées pour traiter ces problèmes : QGIS et ArcGIS permettent une correction manuelle et semi-automatisée, tandis que PostGIS offrent une approche plus avancée via des requêtes SQL. Enfin, FME se distingue par sa capacité à automatiser les traitements et à s'intégrer dans des workflows complexes. La maîtrise de ces outils est essentielle pour assurer des analyses précises et exploitables. En comprenant les principes topologiques et en utilisant les méthodes appropriées, il devient possible de produire des données propres et optimisées pour divers projets géospatiaux.