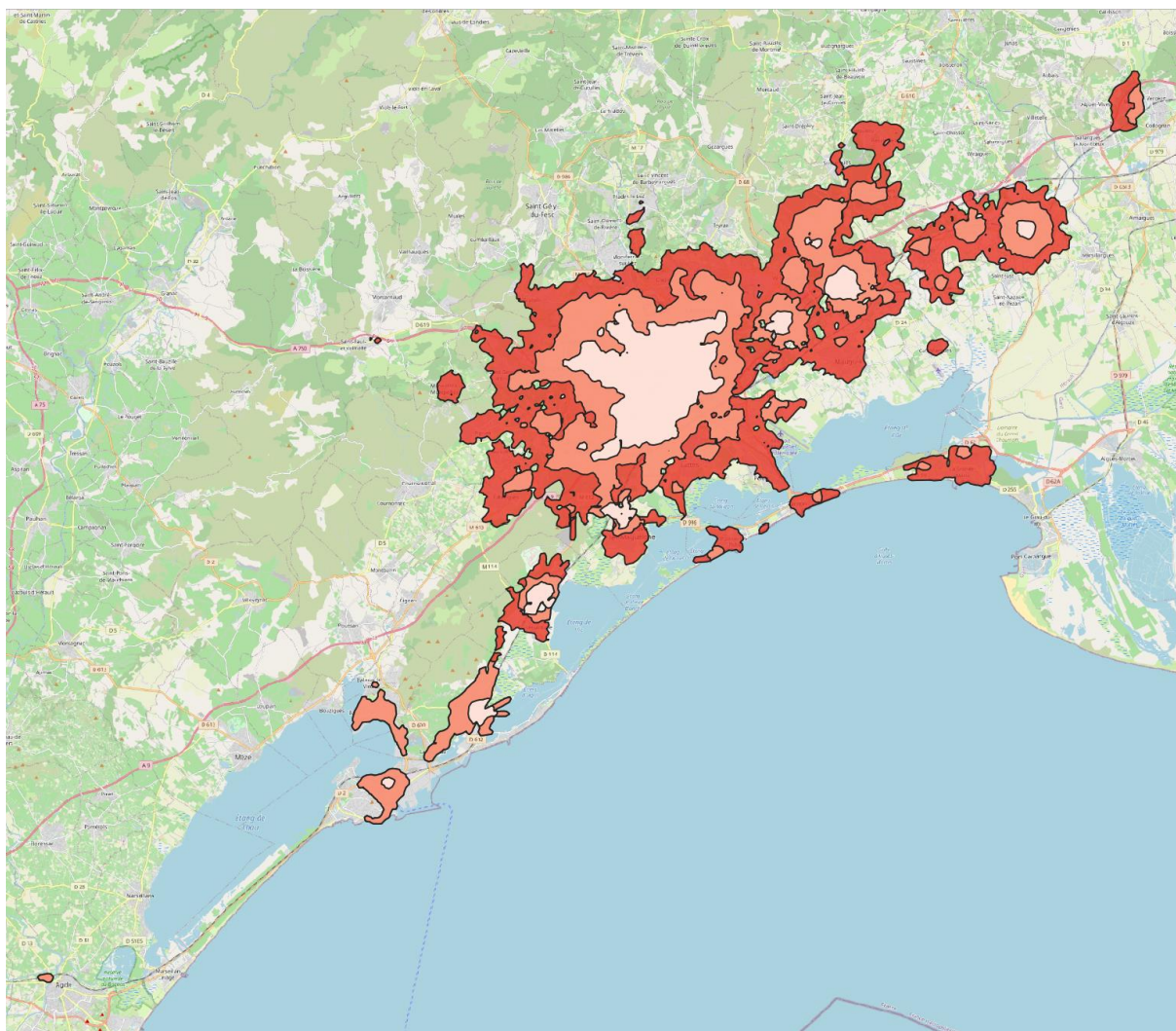


Créer des isochrones avec OpenTripPlanner 1.5 et QGIS



Jean-Clément Ullès

Version 1.0 du document

Avril 2023

Conditions d'utilisation du manuel : CC BY-NC-ND 4.0



<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.fr>

Vous êtes autorisé à :

Partager — copier, distribuer et communiquer le matériel par tous moyens et sous tous formats

Selon les conditions suivantes :

Attribution — Vous devez créditer l'Œuvre, intégrer un lien vers la licence et indiquer si des modifications ont été effectuées à l'Œuvre. Vous devez indiquer ces informations par tous les moyens raisonnables, sans toutefois suggérer que l'Offrant vous soutient ou soutient la façon dont vous avez utilisé son Œuvre.

Pas d'Utilisation Commerciale — Vous n'êtes pas autorisé à faire un usage commercial de cette Œuvre, tout ou partie du matériel la composant.

Pas de modifications — Dans le cas où vous effectuez un remix, que vous transformez, ou créez à partir du matériel composant l'Œuvre originale, vous n'êtes pas autorisé à distribuer ou mettre à disposition l'Œuvre modifiée.

Table des matières

Introduction	4
1) Prérequis : installation de Java	4
2) Téléchargement du répertoire d'exemple	4
3) Démarrage du logiciel.....	6
4) Création des isochrones	7
5) Intégration des isochrones directement dans QGIS.....	9
Annexes	12
Annexe 1 : contrôle des requêtes des isochrones dans OTP	12
Annexe 2 : vigilance pour les GTFS de la SNCF	13
Annexe 3 : illustration du déploiement des isochrones et de la prise en compte des horaires.....	14
Annexe 4 : localisation du jeu de données d'exemple (département de l'Hérault)	15

Introduction

La création des isochrones peut s'opérer dans la dernière version d'OTP (2.2.0) mais peu d'options sont disponibles. En effet, la majeure partie des développements informatiques sur les isochrones est réalisée dans la version 1.5 et n'a pas complètement été mis à jour dans les versions suivantes. L'installation de cette version se trouve ici : <https://docs.opentripplanner.org/en/v1.5.0/Basic-Tutorial/>.

Un manuel complet d'installation et d'utilisation d'OpenTripPlanner 2.2.0 est également disponible et fournira des informations complémentaires sur le fonctionnement général d'OTP.

1) Prérequis : installation de Java

OpenTripPlanner fonctionne avec Java (voir le manuel d'installation et d'utilisation d'OpenTripPlanner 2.2.0). Il est nécessaire d'avoir la version 8 minimum de Java. Par exemple, on peut télécharger la version 17 pour Windows ici : <https://www.oracle.com/java/technologies/downloads/#java17>. Le x64 Installer convient pour un utilisateur classique. Java 17 est une version de long terme stabilisée et compatible avec OTP.

2) Téléchargement du répertoire d'exemple

Un répertoire d'exemple est disponible ici :

<https://www.dropbox.com/scl/fo/90cigyfu012uc0xyq1b86/h?dl=0&rlkey=tcn6df4nqxy87bltw09q5onb8>. La validité des données GTFS correspond au début du mois de mars 2023.

otp_1.5	→	graphs	→	current	→	34_gtfs	GTFS
	→	otp-1.5.0-shaded			→	beziers_gtfs	GTFS
					→	capbus_gtfs	GTFS
					→	intercite_gtfs	GTFS
					→	or_gtfs	GTFS
					→	sete_gtfs	GTFS
					→	TAM_MMM_gtfs	GTFS
					→	ter_gtfs	GTFS
					→	tgvt_gtfs	GTFS
					→	Herault.osm.pbf	OpenSteeetMap

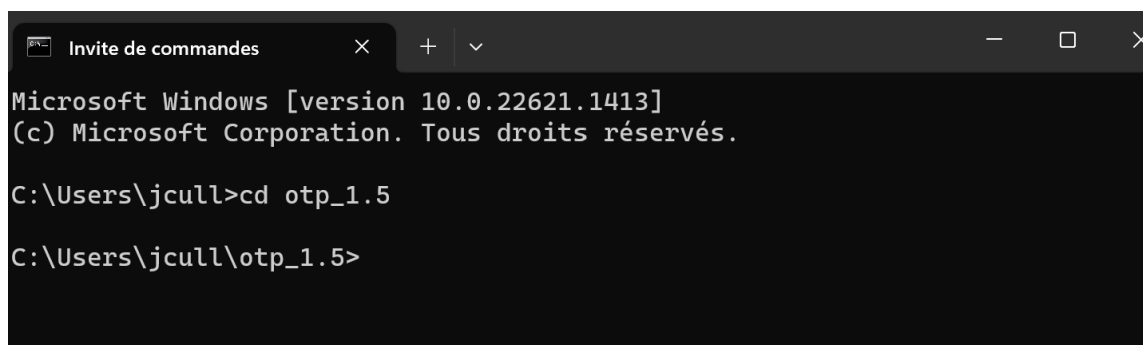
Le répertoire otp_1.5 est à glisser directement dans l'ordinateur, idéalement juste après le nom d'utilisateur dans Windows afin de faciliter le lancement d'OTP à partir de l'invite de commande (voir le manuel d'OTP 2.2.0).

Ce PC > OS (C:) > Utilisateurs > jcul					Reche
Nom	Modifié le	Type	Taille		
.dbus-keyrings	13/04/2023 11:38	Dossier de fichiers			
.ms-ad	17/08/2022 13:21	Dossier de fichiers			
1.5otp	17/01/2023 09:50	Dossier de fichiers			
AppData	01/03/2023 21:24	Dossier de fichiers			
Bureau	14/04/2023 14:22	Dossier de fichiers			
Contacts	01/03/2023 21:27	Dossier de fichiers			
Documents	11/04/2023 09:26	Dossier de fichiers			
Favoris	01/03/2023 21:27	Dossier de fichiers			
Images	01/03/2023 21:27	Dossier de fichiers			
Liens	01/03/2023 21:27	Dossier de fichiers			
Musique	01/03/2023 21:27	Dossier de fichiers			
OneDrive	26/02/2023 10:17	Dossier de fichiers			
otp	23/02/2023 14:20	Dossier de fichiers			
otp_1.5	17/04/2023 21:12	Dossier de fichiers			

Figure 1 - localisation du répertoire dans Windows

3) Démarrage du logiciel

Comme pour OTP 2.2.0 (voir manuel), il faut connecter l'invite de commande Windows au répertoire otp_1.5 localisé dans l'ordinateur. Par défaut, l'invite de commande est connectée à mon nom d'utilisateur Windows. Pour lancer le logiciel, il faut impérativement connecter l'invite de commande au répertoire « otp_1.5 » qui contient tous les fichiers. Pour cela, il faut utiliser la commande « cd » accompagnée du nom du répertoire que vous voulez rejoindre. Dans mon cas, le répertoire « otp » est logé juste après mon dossier d'utilisateur « jcull », donc j'utilise la fonction « cd otp » pour connecter mon invite de commande à mon répertoire otp_1.5 :



```
Microsoft Windows [version 10.0.22621.1413]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\jcull>cd otp_1.5

C:\Users\jcull\otp_1.5>
```

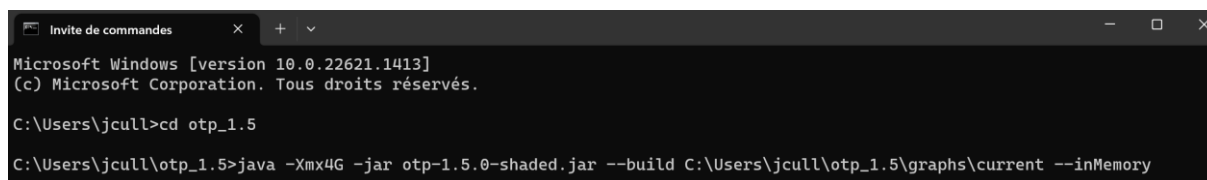
Figure 2 - Utilisation de la commande « cd » pour se connecter au répertoire otp_1.5 dans l'invite de commande

Une fois le répertoire connecté, nous pouvons démarrer le logiciel à partir d'une nouvelle commande. Cette commande est différente de celle pour OTP 2.2.0.

	Code
<pre>java -Xmx4G -jar otp-1.5.0-shaded.jar --build C:\CHEMIN_DU_REPERTOIRE\otp_1.5\graphs\current --inMemory</pre>	

Dans mon cas, le répertoire est localisé juste après mon nom d'utilisateur « jcull » donc le code prendra cette forme :

	Code
<pre>java -Xmx4G -jar otp-1.5.0-shaded.jar --build C:\Users\jcull\otp_1.5\graphs\current --inMemory</pre>	



```
Microsoft Windows [version 10.0.22621.1413]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\jcull>cd otp_1.5

C:\Users\jcull\otp_1.5>java -Xmx4G -jar otp-1.5.0-shaded.jar --build C:\Users\jcull\otp_1.5\graphs\current --inMemory
```

Figure 3 - Démarrage du logiciel OTP 1.5 dans l'invite de commande

En validant le code, le logiciel démarre et la procédure peut prendre quelques minutes en fonction de la taille de la tuile OSM et des GTFS chargés. Lorsque le chargement est terminé, la console affiche « Grizzly server running. », de la même manière qu'OTP 2.2.0. Une fois le logiciel OTP démarré, il faut ouvrir le navigateur et entrer l'URL suivant : <http://localhost:8080/>. La visualisation du logiciel et la construction des requêtes est identique à OTP 2.2.0, on peut se référer au manuel pour toutes les explications du fonctionnement des requêtes.

4) Création des isochrones

La création des isochrones se construit dans l'URL ce qui renvoie une couche shapefile des isochrones demandés. Par exemple :

<http://localhost:8080/otp/routers/default/isochrone?fromPlace=43.697446,3.86475&mode=WALK,TRANSIT&date=03-07-2023&time=11:00am&maxWalkDistance=800&cutoffSec=1800&cutoffSec=2700&cutoffSec=3600&maxTransfers=2&minTransferTime=179>

Décomposons l'URL pour mettre en lumière les différents paramètres :

http://localhost:8080/otp/routers/default/isochrone? : base de l'URL, ne change jamais.

fromPlace=43.697446,3.86475 : coordonnées géographiques (WGS84) du lieu d'origine de l'isochrone.

mode=WALK,TRANSIT : déplacements paramétrés (ici transport en commun et marche, non modifiable).

date=03-07-2023 : date paramétrée (format mm-jj-aaaa).

time=11:00am : horaire paramétré (ici 11h00 du matin, en format anglais sur OTP).

maxWalkDistance=800 : Paramètre optionnel limitant la distance totale de marche de l'utilisateur à 800 mètres pour réaliser le déplacement (marche lors des correspondances comprise).

cutoffSec= : Paramètre de la durée totale, en seconde, pour laquelle l'utilisateur peut se déplacer. Lorsque la durée est atteinte, le déplacement s'arrête (limite de l'isochrone). On peut ajouter autant de « *cutoffSec=* » que l'on veut dans l'URL en utilisant le signe « & » pour les séparer. Dans notre URL d'exemple, 3 durées d'isochrone sont demandées : 30 minutes, 45 minutes et 60 minutes.

maxTransfers=2 : Paramètre optionnel du nombre maximal de correspondances que peut réaliser l'utilisateur. Dans notre exemple, l'utilisateur sera limité à 2 correspondances ce qui termine son déplacement même s'il lui reste du temps disponible pour une autre correspondance.

minTransferTime=179 : Paramètre optionnel de la durée d'attente minimale, en seconde, pour réaliser une correspondance. Cette option peut être utile pour éviter de prendre en compte les correspondances qui se jouent dans un temps très court : la correspondance est « théoriquement » possible mais dangereuse dans la réalité car l'utilisateur qui ne pourra pas la réaliser dans de bonnes conditions. Dans notre exemple, la durée minimale d'attente pour réaliser la correspondance est de 2 minutes et 59 secondes.

En retirant tous les paramètres optionnels, voici une construction possible de l'URL :

<http://localhost:8080/otp/routers/default/isochrone?fromPlace=43.697446,3.86475&mode=WALK,TRANSIT&date=03-07-2023&time=11:00am&cutoffSec=1800&cutoffSec=2700&cutoffSec=3600>

Un très grand nombre de paramètres sont disponibles afin de personnaliser ses isochrones :

http://dev.opentripplanner.org/apidoc/1.5.0/resource_Lisochrone.html

Les paramètres peuvent redéfinir largement la physionomie des isochrones, comme sur cet exemple :

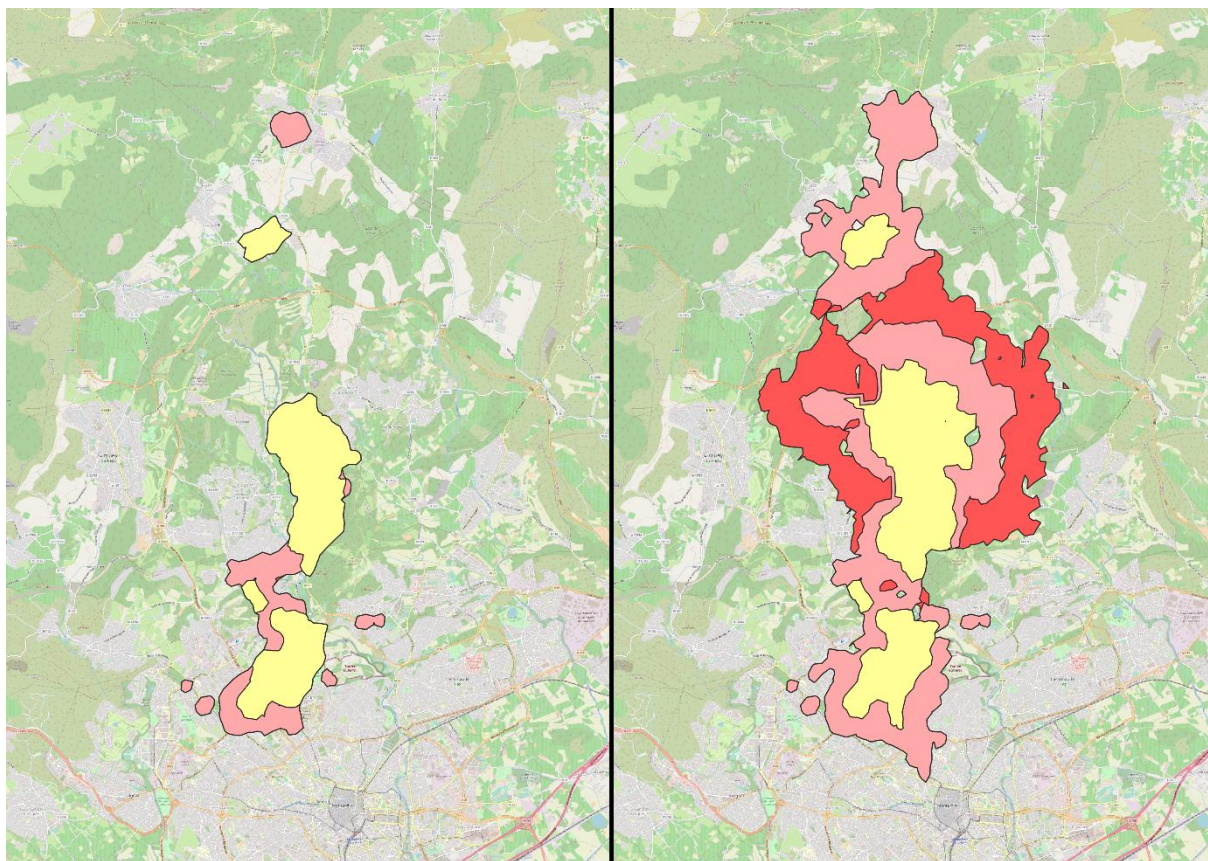


Figure 4 - Génération des isochrones (à gauche : tous les paramètres optionnels ; à droite : sans paramètres optionnels)

5) Intégration des isochrones directement dans QGIS

Il est possible d'intégrer directement la création des isochrones dans QGIS en utilisant la console de programmation Python du SIG.

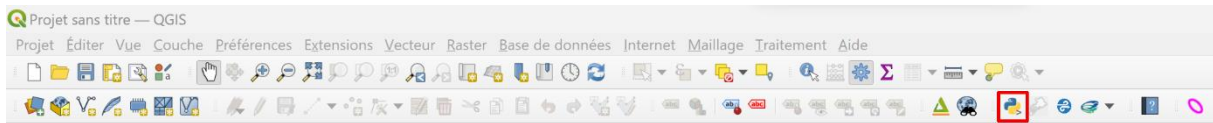


Figure 5 - La console Python de QGIS

Le code pour ajouter une nouvelle couche vecteur à partir de la requête au serveur OTP 1.5 par l'URL prend cette forme :

```
layer = iface.addVectorLayer('URL_DE_LA_REQUÊTE','NOM_DE_LA_COUCHE', 'ogr')
```

Code

- URL_DE_LA_REQUÊTE : URL de l'isochrone
- NOM_DE_LA_COUCHE : permet de donner un nom à la nouvelle couche qui s'ajoutera à QGIS

Dans mon exemple, le code prendra cette forme :

```
layer =  
iface.addVectorLayer('http://localhost:8080/otp/routers/default/isochrone?fromPlace=43.697446,  
3.86475&mode=WALK,TRANSIT&date=03-07-  
2023&time=11:00am&maxWalkDistance=800&cutoffSec=1800&cutoffSec=2700&cutoffSec=3600&  
maxTransfers=2&minTransferTime=179','11:00', 'ogr')
```

Code

De cette manière, je réalise une requête au serveur OTP pour afficher les isochrones demandés directement dans QGIS en format vectoriel. Je lui donne le nom « 11:00 » car les isochrones sont calculés à cet horaire le 07/03/2023.

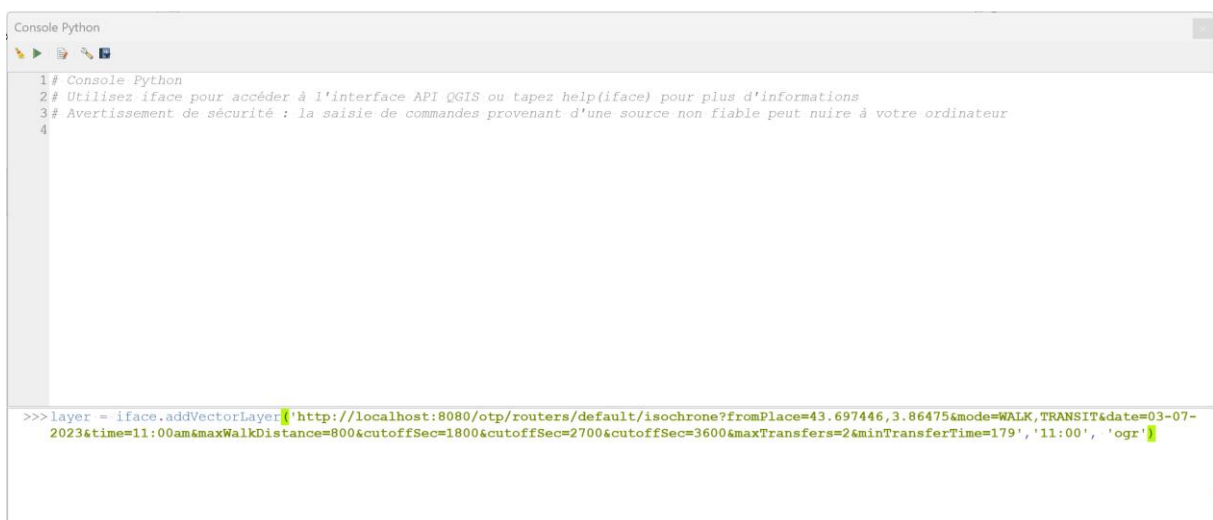


Figure 6 - Création des isochrones dans la console Python de QGIS

En validant, la couche s'affichera dans QGIS :

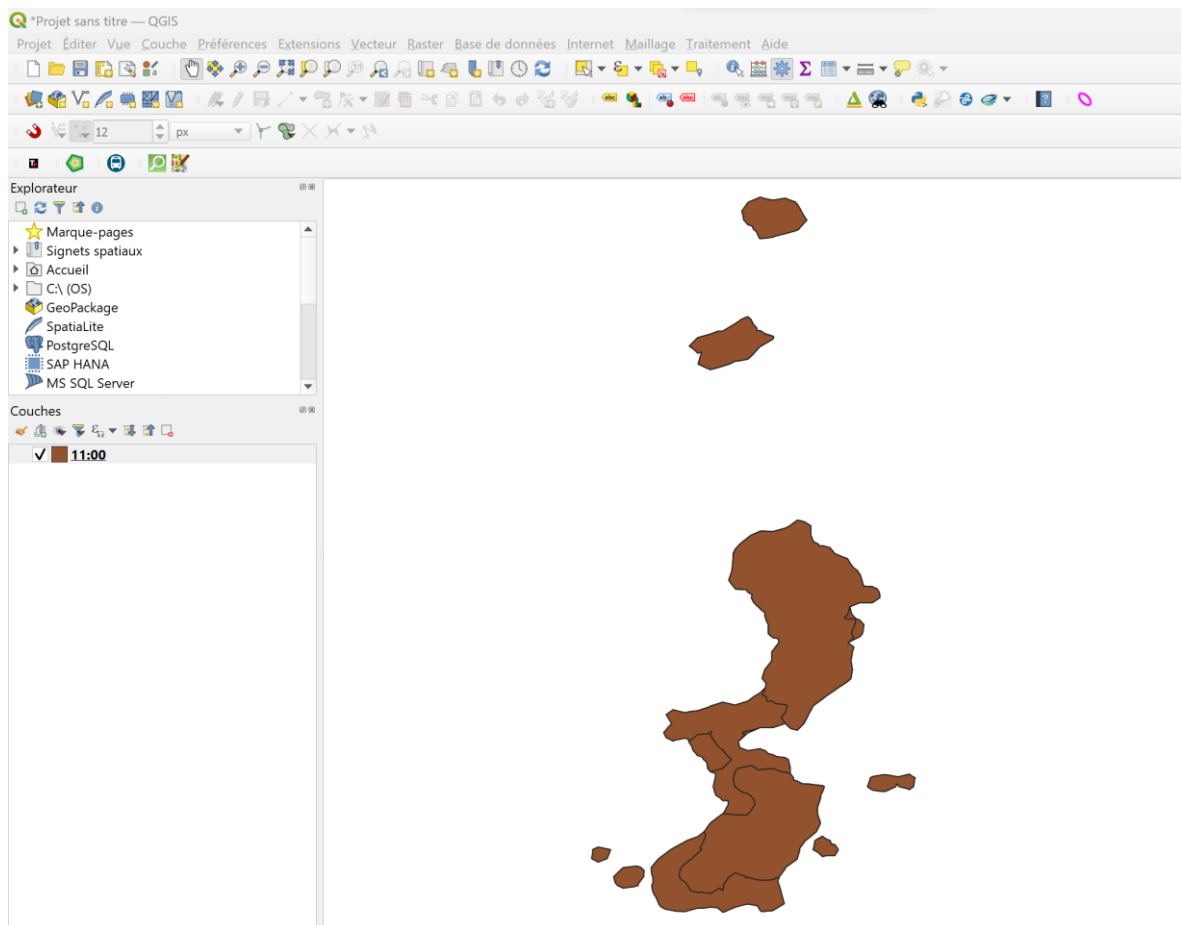


Figure 7 - Génération de la couche d'isochrones dans QGIS

Comme nous avons demandé la génération de 3 isochrones correspondant à 30, 45 et 60 minutes, la couche possède 3 entités distinctes et les durées sont affichées dans la table attributaire :

11:00 — Total des entités: 3, Filtrées: 3,		
	id	time
1	fid-4f6bd4f7_18...	3600
2	fid-4f6bd4f7_18...	2700
3	fid-4f6bd4f7_18...	1800

Figure 8 - Visualisation de la table attributaire de la couche d'isochrones

De ce fait, nous pouvons discrétiser les entités en fonction de la durée possible en transport public et marche :



Figure 9 - Discrétisation des isochrones

Annexes

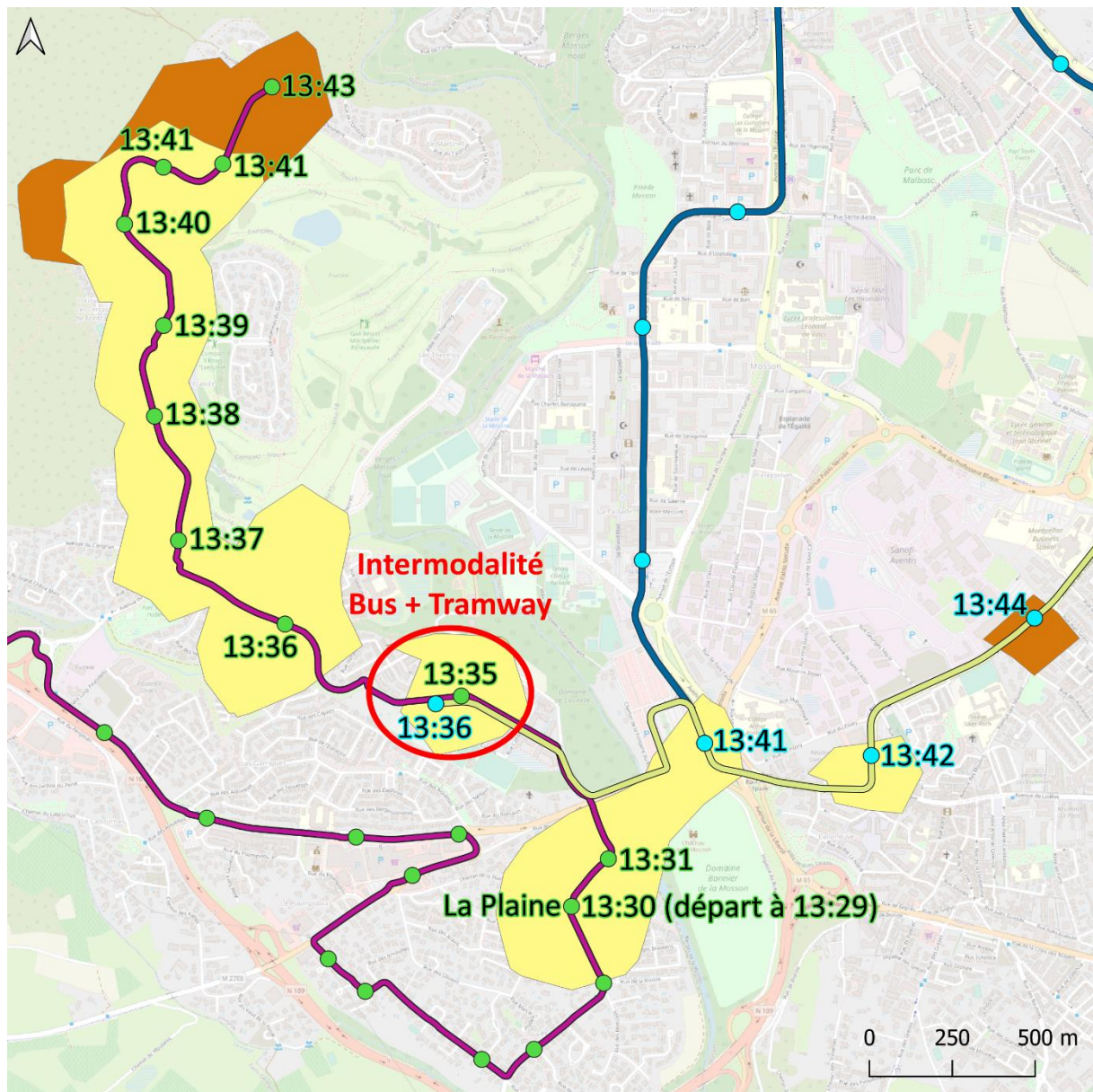
Annexe 1 : contrôle des requêtes des isochrones dans OTP

Lorsqu'une requête est envoyée à OTP, le logiciel affiche des informations dans l'invite de commande pouvant avoir un intérêt pour contrôler le bon fonctionnement de la génération :

```
15:45:37.292 INFO (SPTWalker.java:156) SPTWalker: Generated 60773 points (0 dup edges, 0 no geometry) from 11125 vertices / 24661 states.  
15:45:37.292 INFO (AccumulativeGridSampler.java:139) Added 272 closing samples to get a total of 772.  
15:45:37.292 INFO (SampleGridRenderer.java:89) Computed SPT in 32msec, 47msec for sampling (79 msec total)  
15:45:37.292 INFO (IsoChroneSPTRendererAccSampling.java:101) Computed 3 isochrones in 0msec
```

Figure 10 - Visualisation du logiciel OTP lors de la création d'isochrones

Annexe 3 : illustration du déploiement des isochrones et de la prise en compte des horaires



- Isochrone de 15 minutes (900 secondes)
- Isochrone de 16 minutes et 40 secondes (1 000 secondes)
- Bus ligne 25 (Juvignac)
- Tramway ligne 1
- Tramway ligne 3
- Arrêts de tramway
- Arrêts de bus

13:30 Horaire de passage du bus

13:41 Horaire de passage du tramway

Sources : OpenTripPlanner, Métropole de Montpellier,
OpenStreetMap, 2023
Réalisation : ULLES Jean-Clément, 2023

Figure 13 - Carte d'illustration du fonctionnement concret des isochrones

Annexe 4 : localisation du jeu de données d'exemple (département de l'Hérault)

