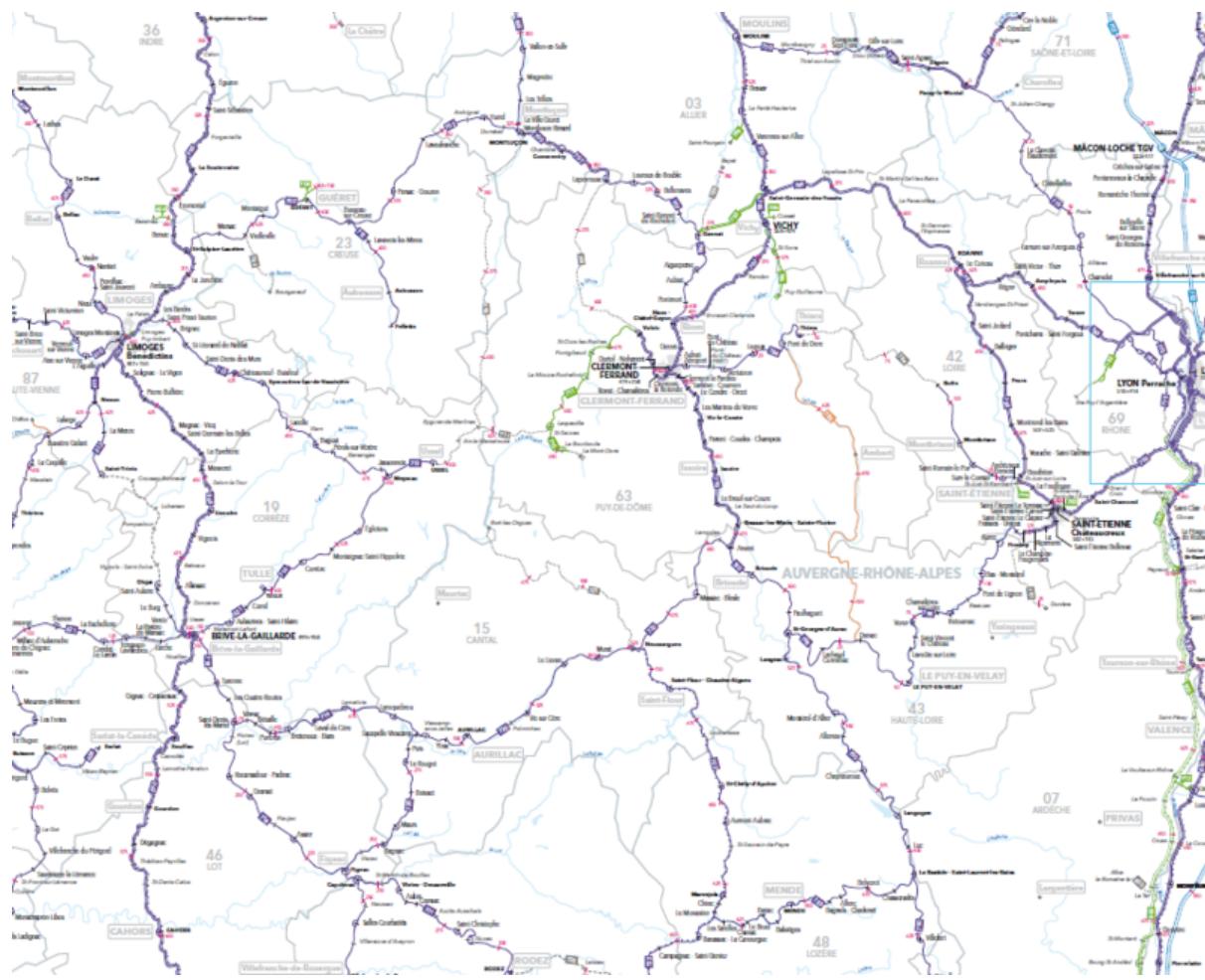


Barbier Lucas
Bayle Sarah
Dutartre Julien

Groupe A14
9 juin 2023

Rapport de projet modélisation

Sujet 8 - Concevoir un plan de transport



<http://raildusud.canalblog.com/archives/2022/01/23/39304318.html>

Référent entreprise : Clément RAOUX - SNCF Direction Générale de l'Exploitation
Référent ENTPE : Christel DIONET - ENTPE Responsable du Cursus Ingénieur

Introduction

La Société Nationale des Chemins de fer Français, ou SNCF, est une entreprise publique française créée par l'Etat en 1937. Bien que le réseau ferré en France appartienne à l'Etat, la SNCF est l'entreprise qui l'exploite le plus, et gère le transport de personnes et de marchandises. Tous les ans, la SNCF revoit sa grille horaire afin de mieux s'adapter à la demande de ses voyageurs, au niveau national comme régional. A ce niveau, la région Auvergne-Rhône-Alpes (AURA) est une des régions les plus importantes de France, que cela soit pour le nombre de voyageurs ou le nombre de kilomètres de voies ferrées. En ce sens, il faut modéliser le réseau ferré de cette région ainsi que le déplacement des trains afin de produire une grille horaire cohérente et satisfaisante pour les usagers.

Au cours du projet de modélisation, nous devons produire une grille horaire sur le réseau AURA, ainsi qu'un graphique espace-temps (GET). Le réseau à modéliser comporte toutefois quelques simplifications afin de faciliter la tâche qui nous est confiée. Ces différentes missions passent tout d'abord par la mise en place d'une modélisation simplifiée, mais fiable du réseau ferré en AURA, ainsi que des équations qui régissent le déplacement des trains en eux-mêmes. Ce sont donc ces étapes, puis les résultats, que nous allons présenter ici.

Sommaire

Introduction	2
I- Hypothèses du projet	4
II- Présentation de notre travail	5
1- Organisation générale	5
2- Détails des programmes informatiques	6
a) Les données	6
b) Explication des fonctions	7
c) Plan de transport	8
d) Graphique espace-temps	8
e) Grille horaire	8
3- Le résumé de la structure du projet	9
III- Analyse critique des résultats	10
Annexes	11

I- Hypothèses du projet

Dans ce projet de conception de plan de transport de la région AURA, l'objectif est de réaliser un plan de transport réalisable pour cette zone. Pour cela, plusieurs contraintes sont à prendre en compte. En effet, nous devons prendre en compte les temps d'arrêt des trains, les vitesses maximales sur les tronçons, l'électrification ou non des lignes... En tout, le nombre de lignes à prendre en compte était de 13 pour le train et de 4 pour le TGV.

Pour ce projet, de nombreuses hypothèses ont déjà été données dans l'énoncé afin de simplifier la réalisation de ce dernier :

- Nombre de trains et personnel illimités
- 3 types de trains (2N, Regiolis, Autorail) et un type de TGV
- Aucune pente sur le parcours
- Beaucoup de petites gares ont été retirées du projet
- Les lignes de FRET sont retirées
- Les gares de Lyon (Perrache et Part-Dieu) sont regroupées en une seule gare
- Temps d'arrêt en gare de départs et de terminus imposé (10 minutes)

Enfin, pour effectuer nos programmes, nous avons dû prendre des décisions de modélisation. Nous avons donc ajouté des hypothèses :

- Nous avons arrondi les valeurs des distances pour les tronçons
- Nous avons assimilé des lieux de bifurcation avec des gares lorsqu'elles étaient très proches (moins de 2km)

D'autres hypothèses sont expliquées dans la partie où nous détaillons les calculs (partie II-2).

II- Présentation de notre travail

1- Organisation générale

Nous avons commencé le projet en récoltant des données sur les documents fournis dans la commande et sur le site data.sncf. Grâce à ces données, nous avons pu réaliser 2 cartes modélisant l'infrastructure du réseau. Une carte représente les différentes lignes (grâce à un code couleur), les distances et le nombre de voies pour chaque ligne [Annexe 1]. La deuxième carte indique les vitesses sur les tronçons et si les lignes sont électrifiées ou non [Annexe 2].

Nous avons ensuite créé plusieurs fonctions utiles pour réaliser un essai sous forme de tube. Nous avons notamment conçu des fonctions pour calculer :

- L'accélération et le freinage
- La distance minimale et durée pour atteindre la vitesse maximale sur un tronçon
- La vitesse de parcours d'un tronçon puis d'une ligne
- ...

En parallèle, nous avons créé des programmes regroupant toutes les données des lignes et des trains pour pouvoir les exploiter informatiquement sur Python. Ce fichier a été modifié tout au long du projet, lorsque nous nous rendions compte de l'utilité de certaines données et pour rendre la mise en forme la plus pratique possible, pour manier les données dans les fonctions.

Grâce aux premières fonctions et données, nous avons pu réaliser un test sur un tube. Puis, nous avons créé le programme pour un plan de transport sur 1h. A l'aide de ce programme, nous avons construit le programme du graphique espace-temps et la grille horaire.

Voici un tableau récapitulatif des actions de chacun :

Lucas	<ul style="list-style-type: none"> - Majorité des fonctions du programme acceleration_train - Plan de transport 1h - GET - Création du plan de transport final - Création des résultats finaux
Julien	<ul style="list-style-type: none"> - Récolte de données - Une fonction du programme acceleration_train - Grille horaire - Aide pour l'affichage du GET
Sarah	<ul style="list-style-type: none"> - Cartes du réseau - Une fonction du programme acceleration_train - Informatisation des données - Aide pour l'affichage du GET - Création des résultats finaux - Rapport

2- Détails des programmes informatiques

Dans cette partie, nous allons expliciter les formules et méthodes utilisées et les calculs réalisés ainsi que le fonctionnement général des nos programmes.

a) Les données

Tout d'abord, nous avons créé un programme regroupant toutes les données des lignes sous formes informatiques.

Pour commencer, nous avons représenté chaque tronçon de cette manière :

```
'''lien1 = l1 = [Point de départ, point d'arrivée, vitesse (km/h), distance (km),
électricité (1) ou non (0), nombres de voies]'''
```

Puis, pour chaque ligne nous avons indiqué les tronçons la composant, le sens de parcours des tronçons et nous avons vérifié si toute la ligne était électrifiée ou non. Pour cela, nous avons choisi de présenter les lignes de cette manière :

```
Ligne_Lyon_Macon=[[19,18,17,16,120,121], [False, False, False, False, True, True],
all([19[4],18[4],17[4],16[4],120[4],121[4]])]
```

Voici l'exemple de la ligne Lyon - Mâcon. Les tronçons la constituant sont réunis dans une liste. Puis, dans une seconde liste, nous indiquons pour chaque tronçon s'il est orienté dans le sens de parcours, c'est-à-dire si le point de départ du tronçon est bien parcouru avant le lieu de fin du tronçon pour la ligne étudiée : si le tronçon est dans le bon sens, nous marquons 'True' sinon nous mettons 'False' dans la liste. Enfin, pour le dernier terme de la liste, représentant la ligne, nous avons utilisé la fonction python 'all' sur le terme qui indique l'électrification (ou non) des lignes de chaque tronçon. Cette fonction renvoie 'True' si tous les éléments en entrée sont vrais et 'False' sinon.

De plus, nous avons formé de nombreux dictionnaires pour y mettre les données :

- Pour rassembler toutes les lignes (dans les 2 sens)
- Pour indiquer les gares desservables pour chaque ligne
- Pour indiquer la capacité de chaque noeud
- Pour attribuer une couleur à chaque ligne sur le GET afin de le rendre plus clair

Par ailleurs, nous avons informatisé les données des différents trains pouvant circuler pour notre projet. Nous avons créé pour chaque train une liste avec la masse du train, les trois coefficients de résistance à l'avancement, la matrice des efforts, la capacité d'accueil de passagers et électrification ou non. Ces listes ressemblent à cela :

```
# TER 2N
M_v_e_2n = np.array([[0 , 370400] , [10 , 370400] , [20 , 370400] , [30 , 352000] , [40 ,
306000] , [50 , 244800] , [60 , 204000] , [70 , 174860] , [80 , 153290] , [90 , 126960] ,
[100 , 107260] , [110 , 92090] , [120 , 80120] , [130 , 70490] , [140 , 62610] , [150 ,
56070] , [160 , 50570]])
Ter_2n = [357.5 , 8.028 , 0.1 , 0.001854 , 160 , M_v_e_2n , 570, True]
```

La matrice des efforts rassemble des listes de 2 éléments : une vitesse et l'effort correspondant. Pour l'électrification, nous avons fait le choix d'utiliser un booléen : 'True' si le train est électrique et 'False' sinon.

Enfin, nous avons regroupé les divers trains dans un dictionnaire, en les associant à leurs noms.

b) Explication des fonctions

Dans un des programmes nous avons regroupé toutes les fonctions nécessaires à la détermination du mouvement d'un train. Voici le détail et les équations utilisées des principales fonctions.

- *Accélération*

Tout d'abord, pour calculer l'accélération, nous appliquons le principe fondamental de la dynamique. Pour cela, nous avons calculé la résistance à l'avancement (RAV) à partir des trois données dans les documents sur les trains et en utilisant la vitesse actuelle du train. La formule à laquelle nous avons eu recours est la suivante : $RAV = masse \times (RAV_A + RAV_B \times vitesse + RAV_C \times vitesse^2)$. Ensuite, nous avons

appliqué la formule du PFD pour obtenir l'accélération : $accélération = \frac{\sum Forces extérieures}{masse}$

avec $\sum Forces extérieures = effort - RAV$

- *temps_distance_acc*

De plus, pour calculer les distances d'accélération, nous avons utilisé cette formule dans le cas où le train doit freiner :

$$d = \frac{v_2^2 - v_1^2}{2 a_{max}}$$

avec v_1 la vitesse de la consigne (la vitesse maximale du tronçon), v_2 la vitesse actuelle (vitesse à laquelle roule le train avant d'entrer sur ce tronçon) et a_{max} la valeur absolue du freinage du train.

Dans le cas inverse, nous avons eu recours à la méthode des rectangles pour approximer le calcul des intégrales de vitesses et de distances, à partir des valeurs d'effort données dans le fichier *Caractéristiques Techniques*.

- *transition_troncon*

Par ailleurs, nous avons réalisé un programme qui renvoie le temps pour parcourir un tronçon donné et la vitesse en sortie de ce dernier. Pour cela, il est nécessaire de distinguer plusieurs cas. Il faut déjà savoir si le train doit accélérer, freiner, voire même les deux pour passer au tronçon suivant. Ensuite, suivant les cas, nous calculons les distances et temps d'accélération et/ou de freinage.

- *temps_parcours_ligne*

Cette fonction renvoie le temps nécessaire pour parcourir une ligne donnée du réseau. La sortie se présente sous forme de liste avec les temps d'arrivée (et de départ si

une desserte est réalisée) à chaque gare. Pour cela, les temps sont accumulés au fur et à mesure de l'avancée dans la ligne.

Nous avons fait des cas à part pour les lignes de TGV : nous avons défini des points de "lâché de TGV" : à ces endroits, le TGV disparaît de notre modélisation étant donné qu'il rentre sur la LGV. Pour le reste des tronçons constituant la ligne étudiée, nous avons ajouté la distance et la durée de parcours à la distance cumulée et la durée cumulée du programme. De plus, nous avons compté les temps d'arrêts en gares desservies (2 min sauf pour Lyon où l'arrêt est de 6 min) sur la ligne.

Pour le tronçon terminant la ligne, nous ne procédons pas tout à fait pareil. Nous calculons le temps de sortie et la vitesse de sortie de ligne, avec des cas particuliers pour les endroits où les TGV passent d'une ligne normale à une LGV. Puis, nous ajoutons le temps et la distance de sortie à la durée cumulée et à la distance cumulée.

c) Plan de transport

Ce programme est celui qui définit le plan de transport. Il est composé de 7 fonctions-clés. Une fonction récursive *Ajout_train* qui a pour but d'ajouter un train de plus dans le plan de transport, 4 fonctions de vérification pour les tronçons en fonction de leur nombre de voies (de 1 à 4), une fonction de vérification de la surcharge d'un noeud et enfin une fonction qui permet de vérifier un plan de transport entier (*verif_plan*). Les 4 fonctions de vérification des tronçons vont extraire les dates d'entrée et de sortie des trains dans ces tronçons, et regarder pour chaque minute s'il y a des accidents ferroviaires sur ce tronçon. La fonction de vérification d'un noeud va, en fonction de la capacité de ce noeud, regarder s'il n'y a pas plus de trains qu'il ne peut en accepter à chaque minute. La fonction qui vérifie le plan de transport en entier va appeler les fonctions précédentes pour savoir si tous les tronçons et tous les noeuds ne sont pas surchargés.

La fonction *Ajout_train* teste tous les horaires pour toutes les lignes qui ne sont pas présentes dans le plan de transport, et retient seulement ceux qui sont valides, en vertu de la fonction *verif_plan*. Après avoir fait ces tâches, le train avec le plus de contraintes est choisi, et nous le faisons démarrer le plus tôt possible. Nous rappelons ensuite la fonction *Ajout_train*, en mettant comme argument le plan de transport précédent auquel nous avons ajouté le train. Dans le cas où un train n'a aucun horaire de départ disponible, nous récupérons l'heure dans laquelle nous tentons de faire partir ce train, et nous supprimons un train déjà planifié au hasard, jusqu'à ce que la contrainte soit relâchée. Nous rappelons ensuite *Ajout_train* avec le plan contenant le(s) train(s) supprimé(s) en argument.

d) Graphique espace-temps

La fonction *g_e_t* permet de tracer le graphique espace-temps sur une ligne comprise dans la desserte. Elle prend en compte les trains qui parcourent la ligne dans chaque sens, mais aussi les trains qui passent sur chaque tronçon constituant la ligne.

e) Grille horaire

Pour créer la grille horaire, nous avons écrit une fonction qui retourne les gares desservies sur la ligne et les horaires de passages à ces gares. Pour cela, pour chaque tronçon d'une ligne donnée nous devons regarder si les lieux de début et de fin du tronçon sont des gares où le train s'arrête ou non. Il faut faire attention au sens de parcours du tronçon pour placer les gares dans le bon ordre.

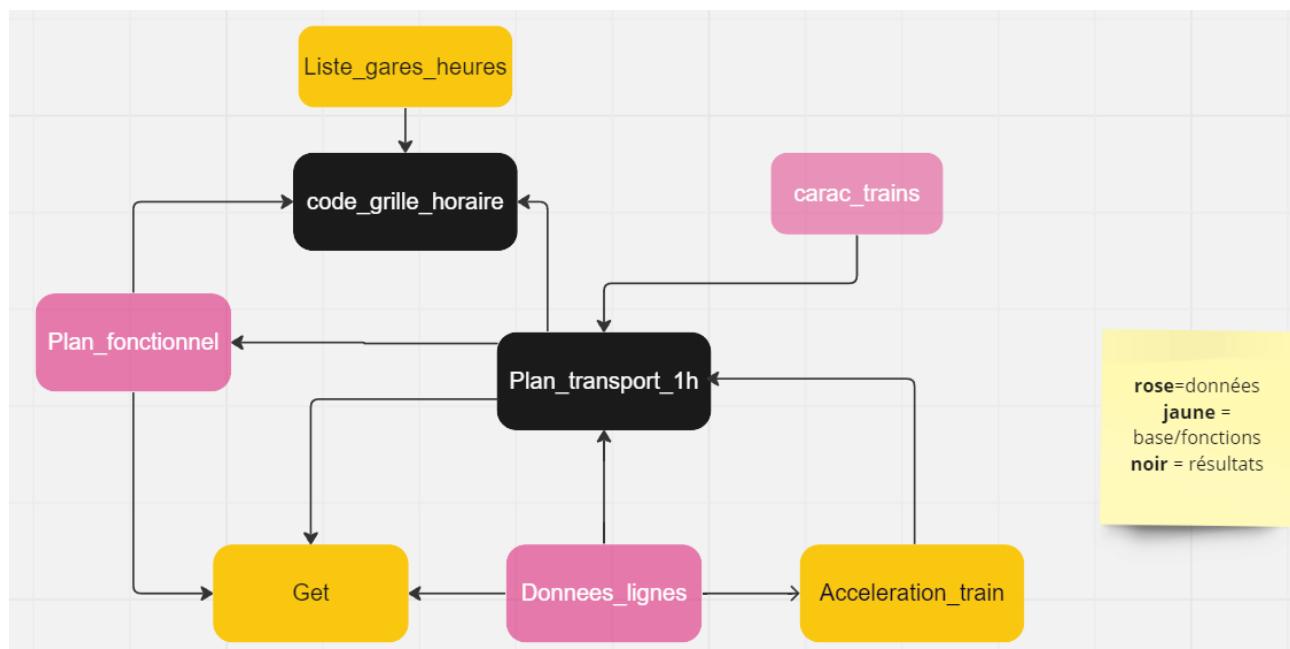
Ensuite, pour les tronçons qui ne finissent pas par une gare, nous avons retiré les temps de fin de tronçon de la liste des horaires à afficher.

Enfin, nous avons utilisé matplotlib.pyplot pour afficher les gares et les horaires sous forme de tableau en faisant attention à convertir les horaires sous forme : 00h00. Pour cela, nous avons dû convertir les résultats du programme Liste_gares_heures grâce à plusieurs fonctions python. Nous avons eu recours à la fonction ‘zip’ qui permet de regrouper les éléments de plusieurs séquences (listes, tuples, etc.) en une seule séquence d'éléments regroupés et la fonction ‘map’ qui permet d'appliquer une fonction à chaque élément d'une ou plusieurs séquences.

3- Le résumé de la structure du projet

Pour le projet, nous avons utilisé plusieurs supports. Nous avons fait les cartes du réseaux sur papier, nous avons résumé des informations dans des documents Excel mais le principal de notre travail a été effectué sur Python. Voici un résumé des programmes écrits et leurs liens :

- Informatisation de toutes les données des lignes (Donnees_lignes)
- Informatisation des caractéristiques des trains (carac_trains)
- Fonctions utiles à la création d'un get (acceleration_trains)
- Premier essai de plan de transport sur 1h qui sert de base à la construction du vrai plan de transport (plan_transport_1h)
- Plan de transport final (Plan_fonctionnel)
- Programme du graphique espace temps (get)
- Création de la grille horaire (Liste_gares_heures et code_grille_horaire)



III- Analyse critique des résultats

Tout d'abord, nous avons modélisé la carte de manière beaucoup plus simplifiée qu'elle ne l'est en réalité. En effet, tous les phénomènes de changement de voie, de placement sur une voie plutôt qu'une autre ne sont pas pris en compte. Plus précisément, nos trains parcourent des bouts de ligne, mais sur des portions de ligne à plus de 2 voies, savoir sur quelle ligne placer tel train n'a pas été pris en compte. C'est encore plus vrai au niveau des bifurcations : notre modèle bloque le point de passage pendant toute la durée de passage du train, ce qui s'avère être plus contraignant sur les horaires des différents trains. Dans le même domaine, notre modélisation de la carte est simplifiée : nous avons fait fi de plusieurs bifurcations, en choisissant de faire des sortes de "ponts" lorsque certaines lignes se croisent.

Du point de vue du déplacement des trains en eux-mêmes, nous avons fait le choix de les faire circuler le plus vite possible, ce qui n'est pas représentatif de la réalité. En effet, en réalité, les sillons sont plus souples et conditionnent une conduite moins énergivore et moins rapide que celle que nous avons modélisée.

Le programme de plan de transport est aussi très lent, nous aurions pu optimiser grandement son temps de calcul afin d'être plus efficaces dans la création de ce plan. Pour créer le plan de transport, ajouter des trains jusqu'à saturation aurait peut-être été une meilleure idée que d'imposer une desserte par heure.

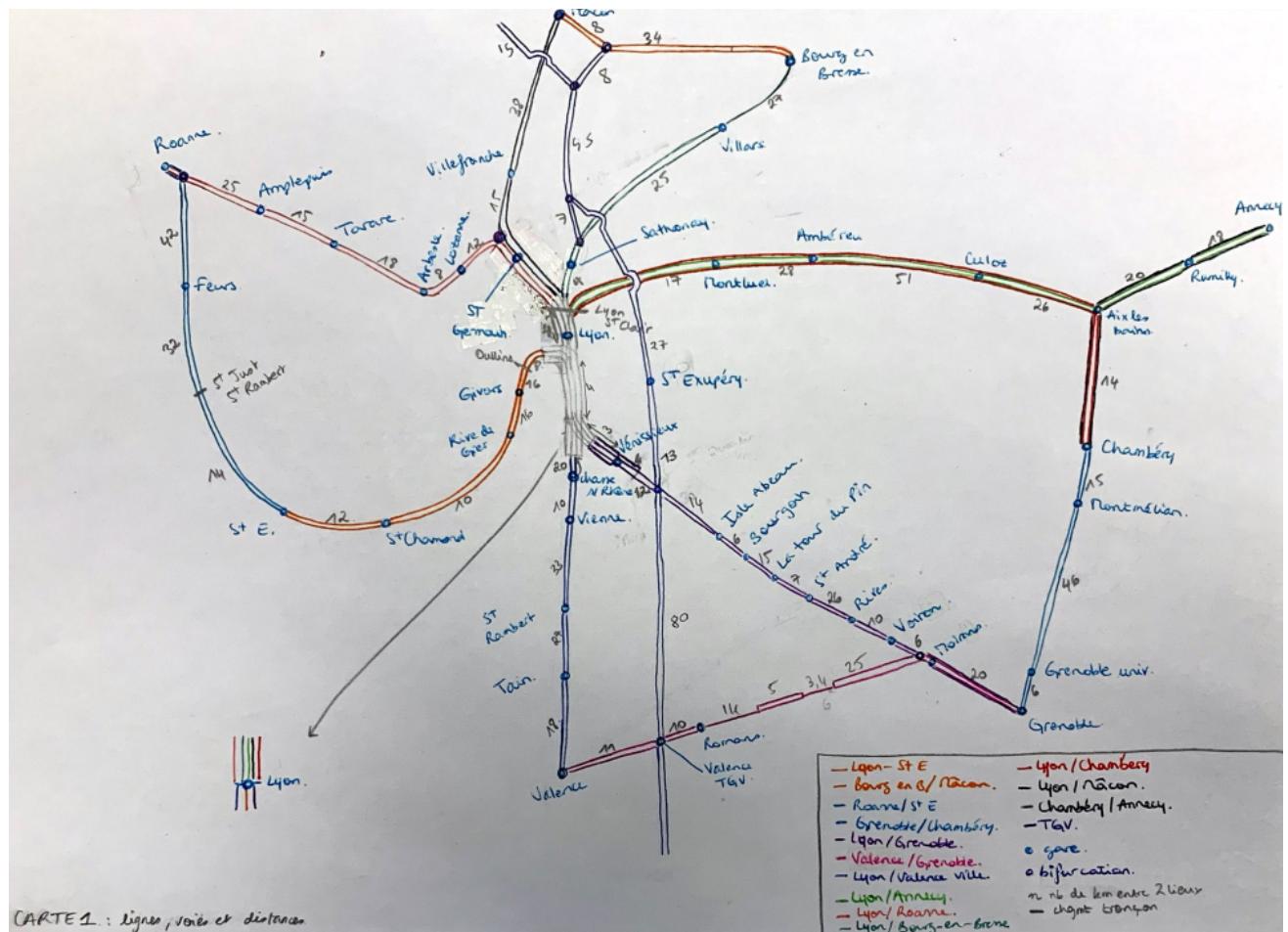
Dans un premier temps, nous n'avons réussi à réaliser qu'un plan de transport sur les créneaux de 6h-13h. Si nous transposons les résultats sur l'horaire 16h-23h, le plan de transport reste valable. Les résultats sous forme de grille horaire et de graphique espace-temps sont en annexes [3 et 4]. Toutefois, des messages d'erreur nous empêchaient de faire circuler des trains sur le créneau 14h-16h. L'erreur provenait des fonctions qui vérifient un tronçon (dans le programme "plan_transport_1h"). Après débogage dudit programme, nous avons pu constater que le problème se situait au Nord, avant la séparation en direction de Mâcon ou Roanne.

De plus, la grille horaire de l'annexe 3 n'était pas complète. Puisqu'aucune différence entre les horaires d'arrêts et les horaires de passages n'a été faite, la grille n'est pas très visible. Grâce aux modifications, nous avons réussi à différencier ces horaires avec des couleurs : des exemples de grilles horaires améliorées sont en annexes [9 à 10]

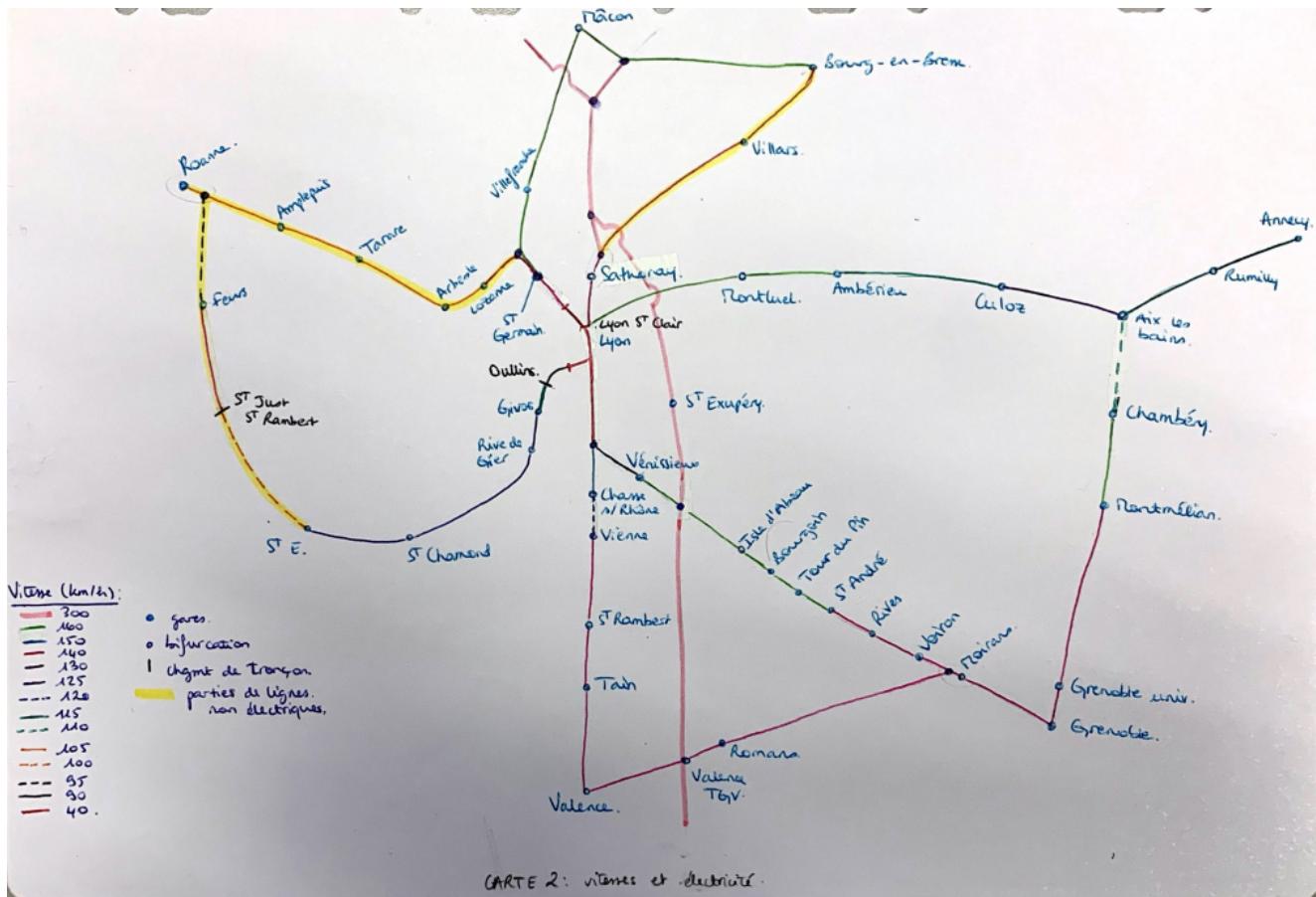
Enfin après l'identification précise du problème dans la fonction, nous avons réussi à obtenir un plan de transport allant de 6h à 21h. Les graphiques espace-temps et les grilles horaires sont en annexes [5 à 8].

Annexes

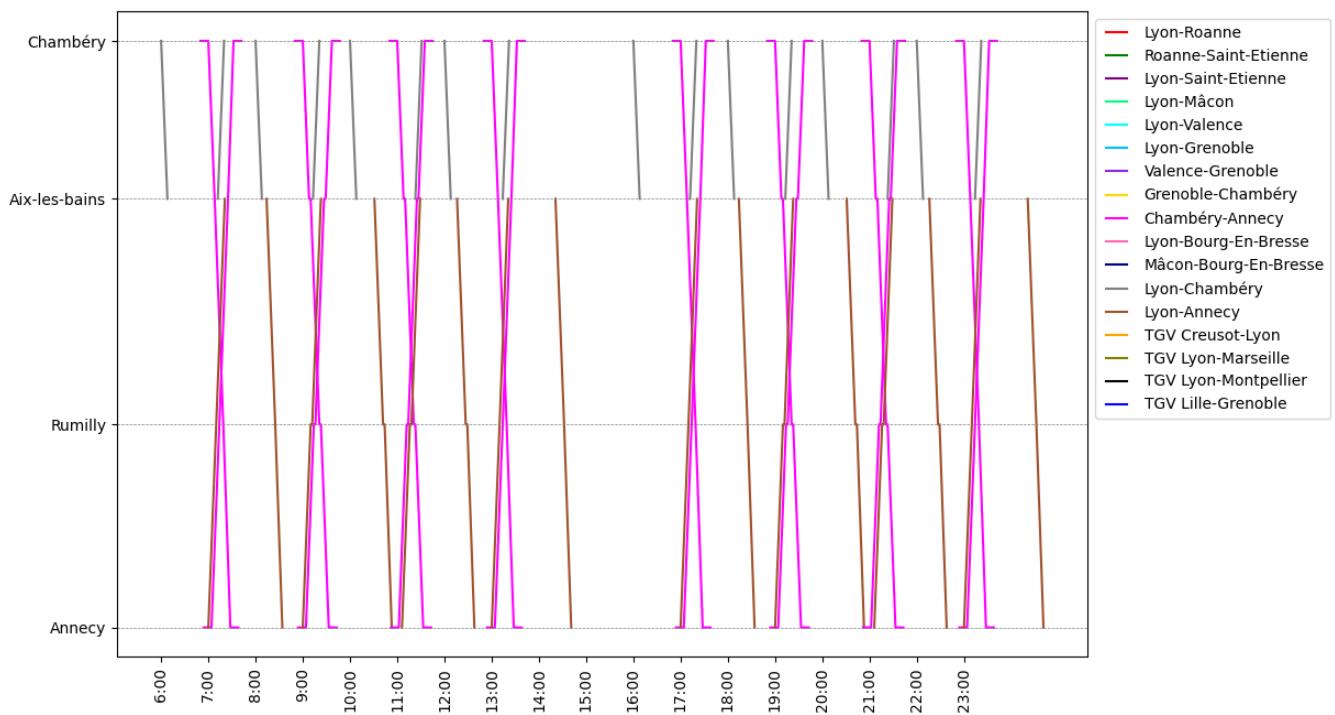
Annexe 1 : Carte des lignes, des distances et du nombres de voies



Annexe 2 : Carte des vitesses et de l'électrification



Annexe 3 : GET incomplet de la ligne Annecy-Chambéry

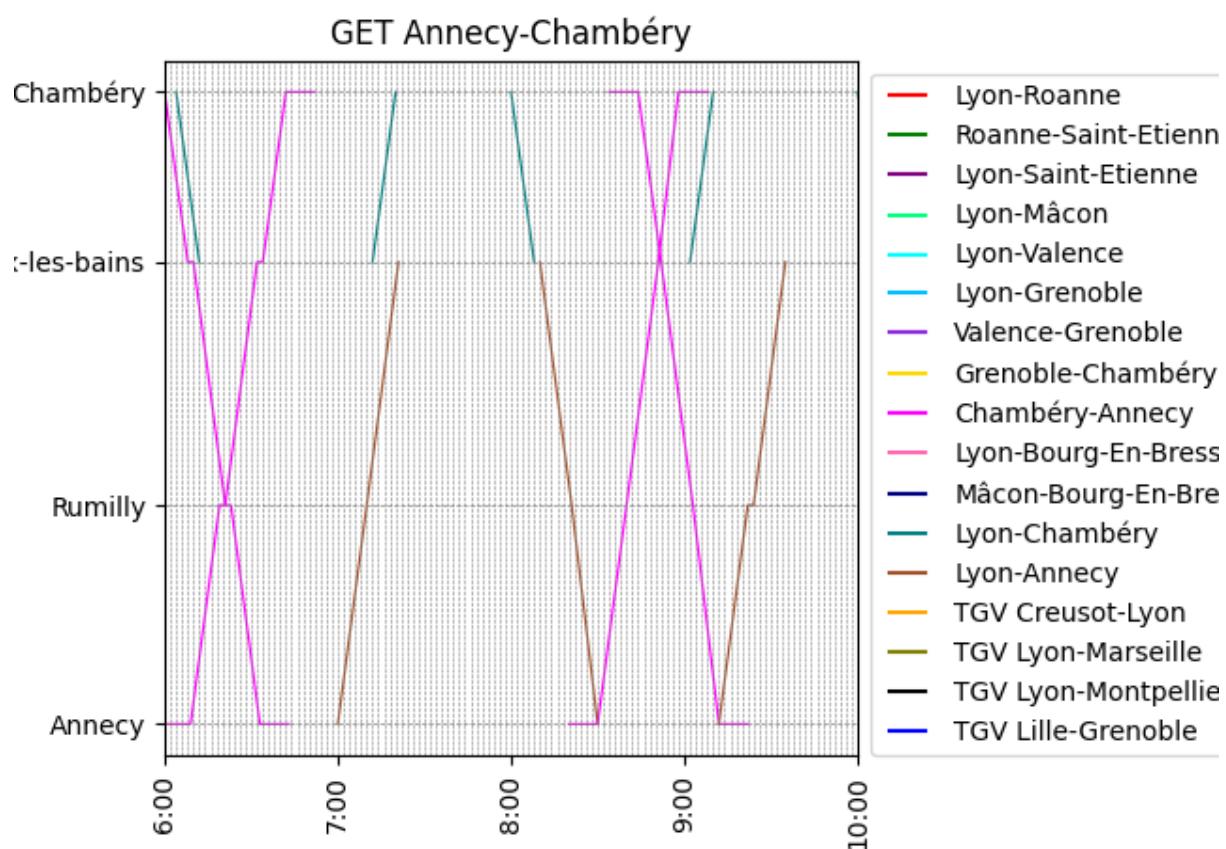


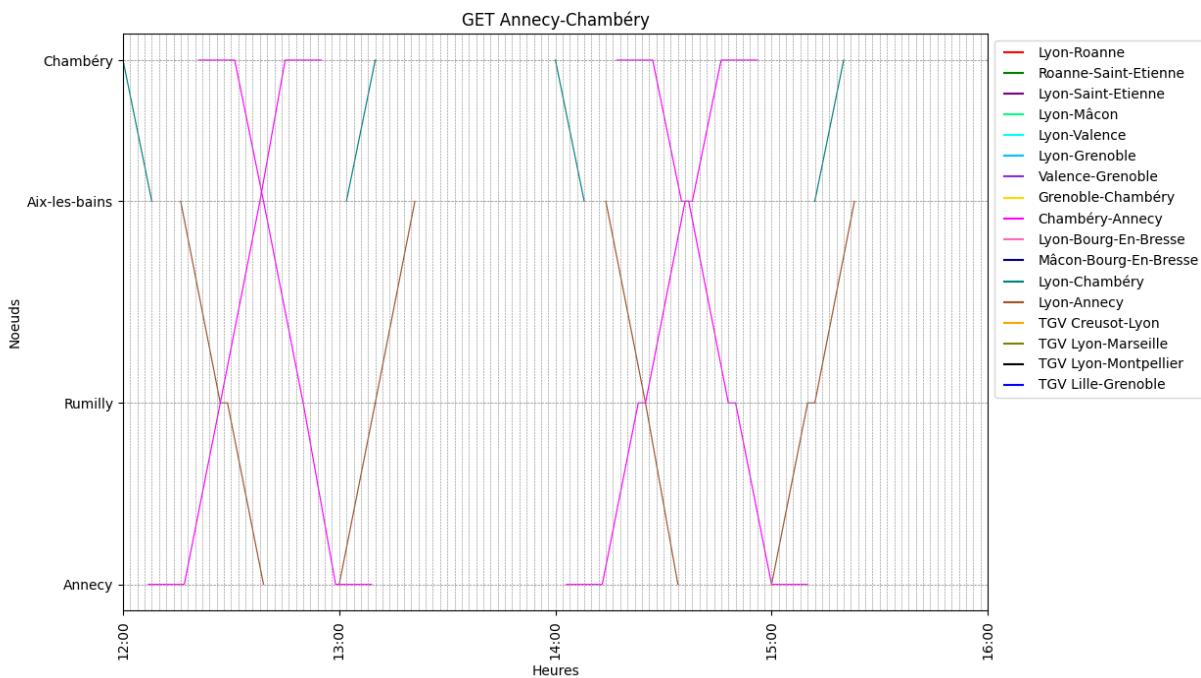
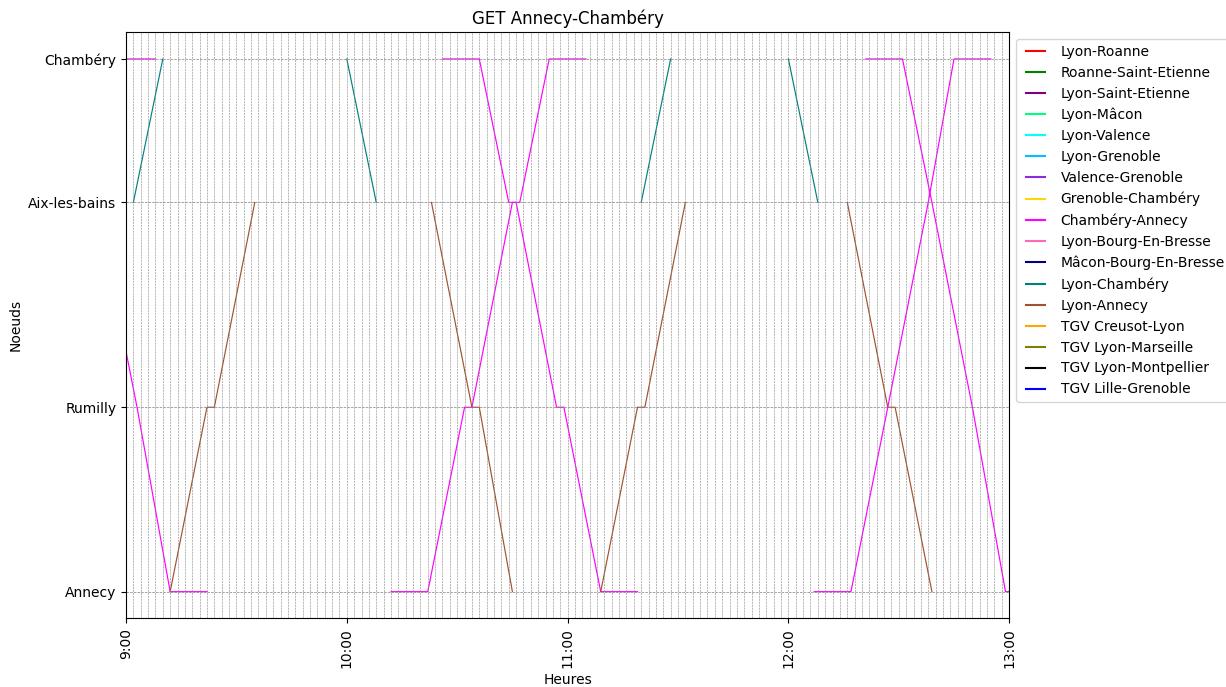
Annexe 4 : Grille horaire incomplète de la ligne Annecy-Chambéry

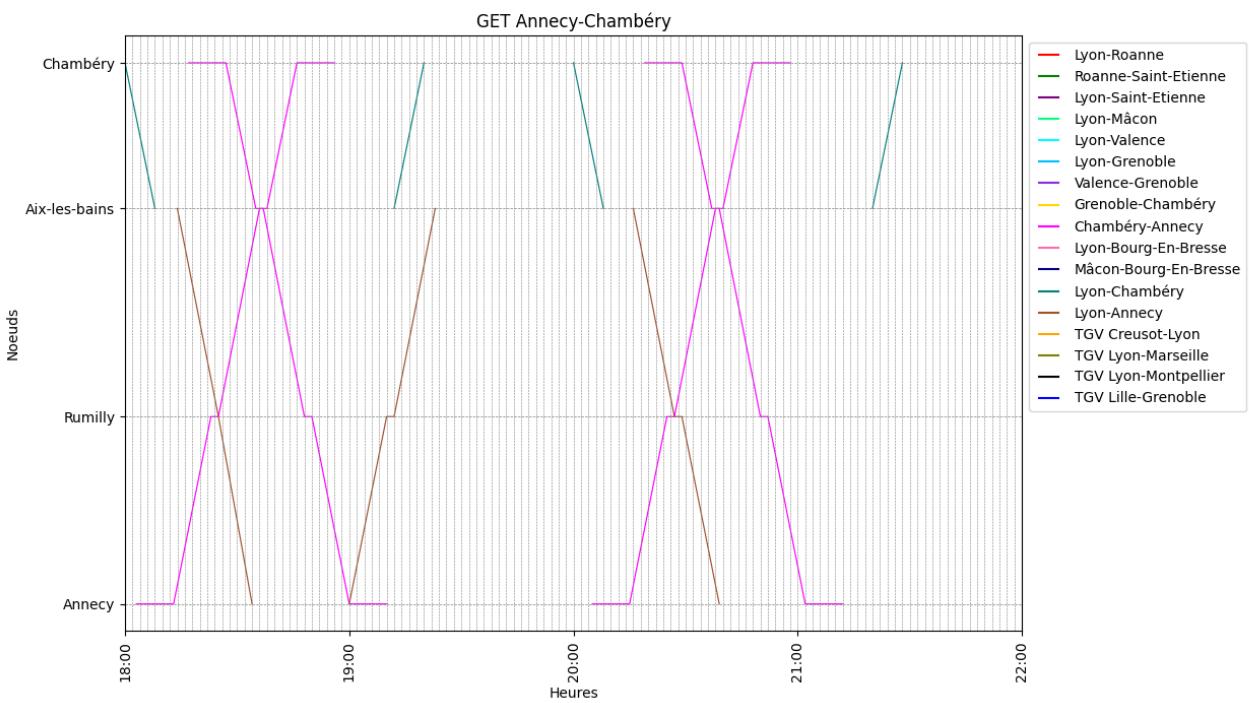
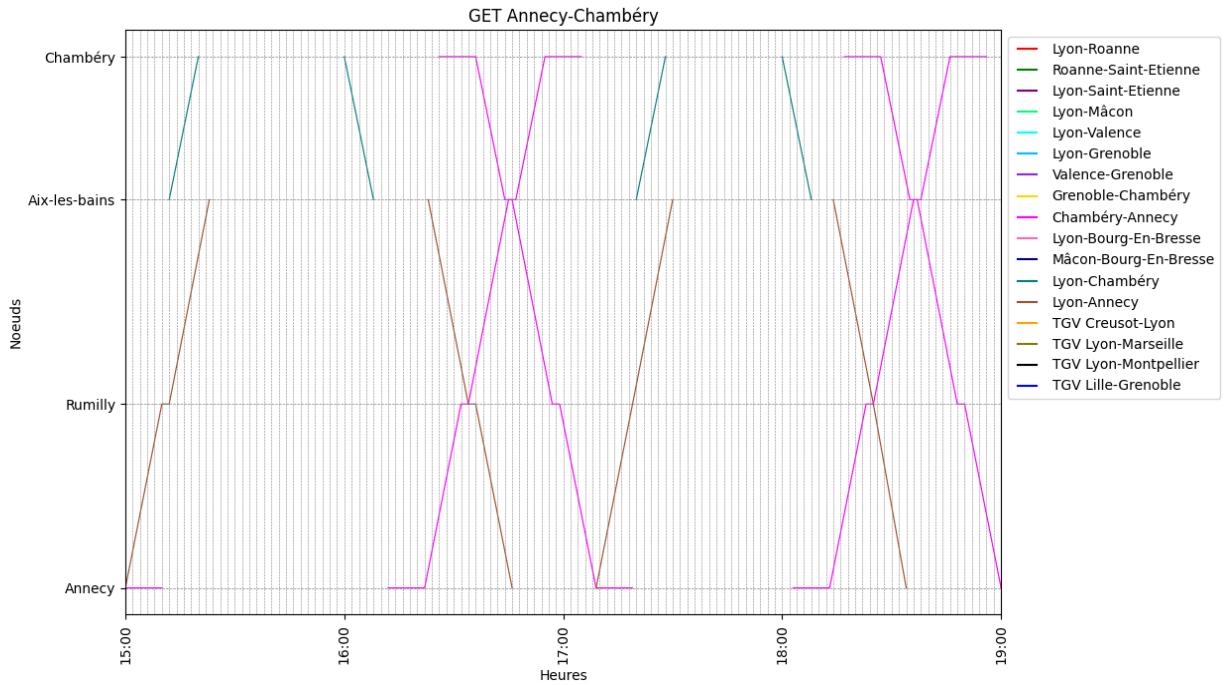
Gares	Horaire							
Annecy	7 h 4	9 h 4	11 h 2	13 h 4	17 h 4	19 h 4	21 h 2	23 h 4
Rumilly	7 h 14	9 h 14	11 h 14	13 h 14	17 h 14	19 h 14	21 h 14	23 h 14
Aix-les-bains	7 h 25	9 h 16	11 h 27	13 h 25	17 h 25	19 h 16	21 h 27	23 h 25
Chambéry	7 h 32	9 h 27	11 h 35	13 h 32	17 h 32	19 h 27	21 h 35	23 h 32

Présentation des GET (une graduation verticale représente 2 minutes)

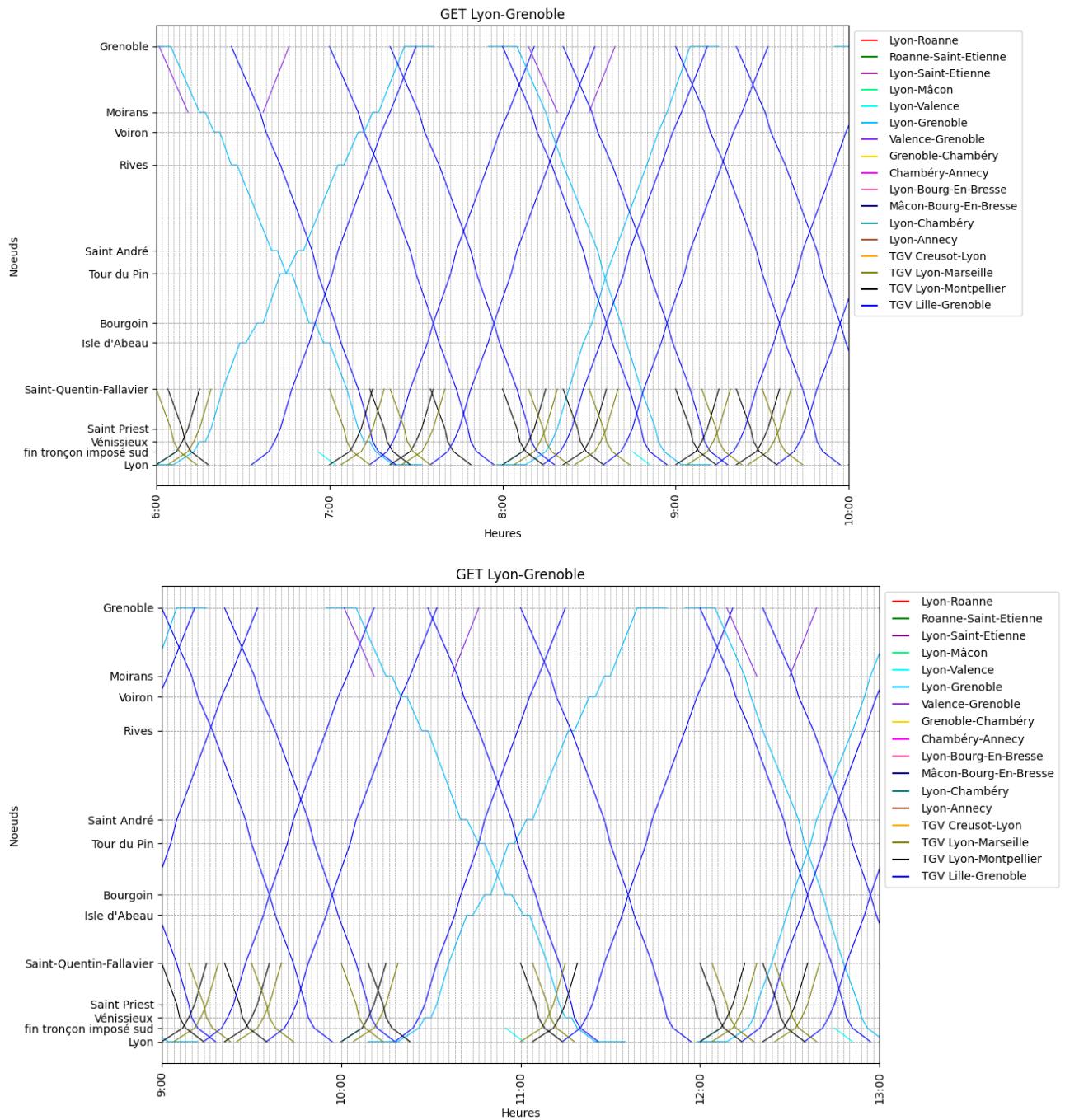
Annexe 5 : GET de la ligne Annecy-Chambéry (séparé pour plus de clarté)

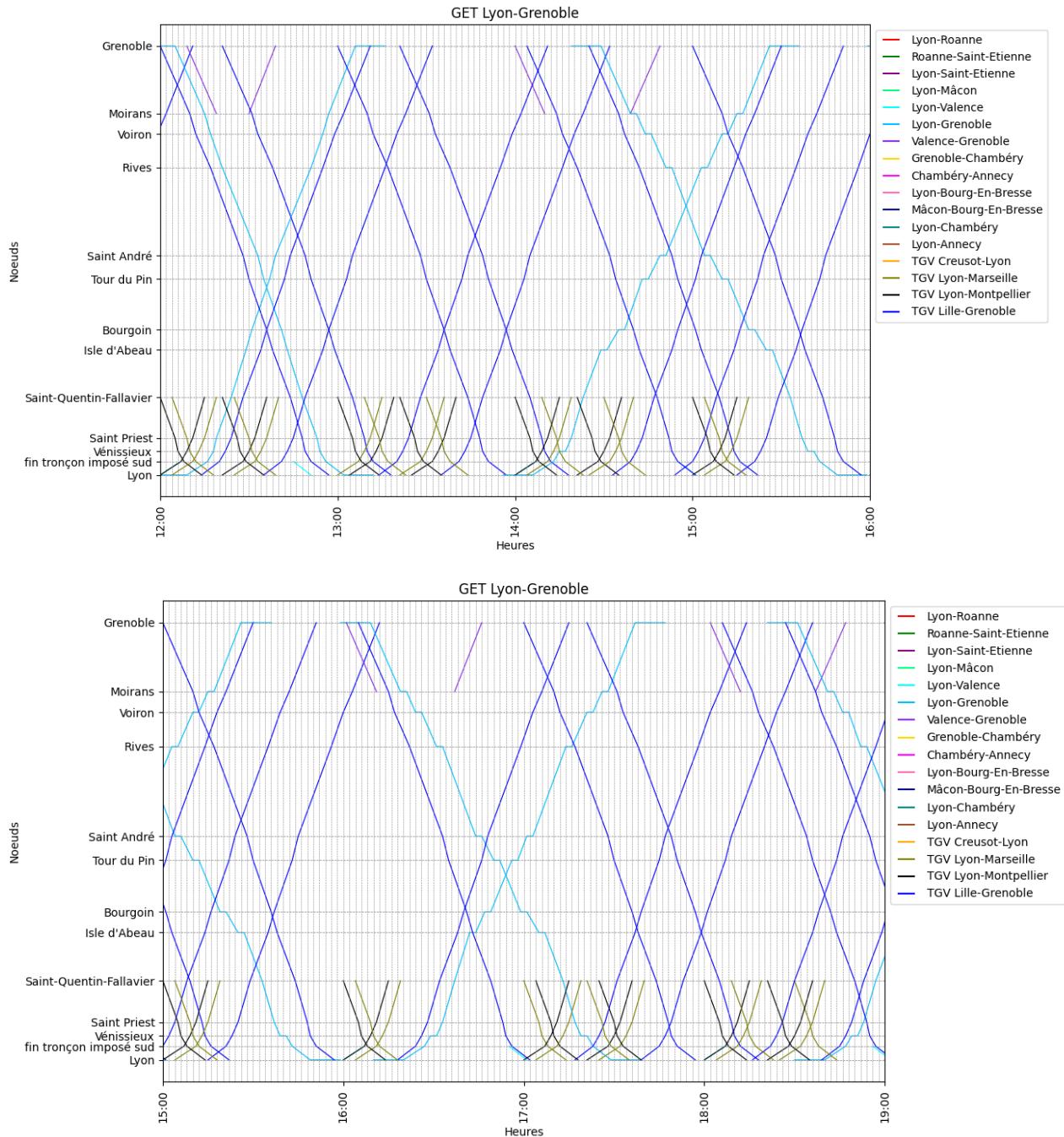


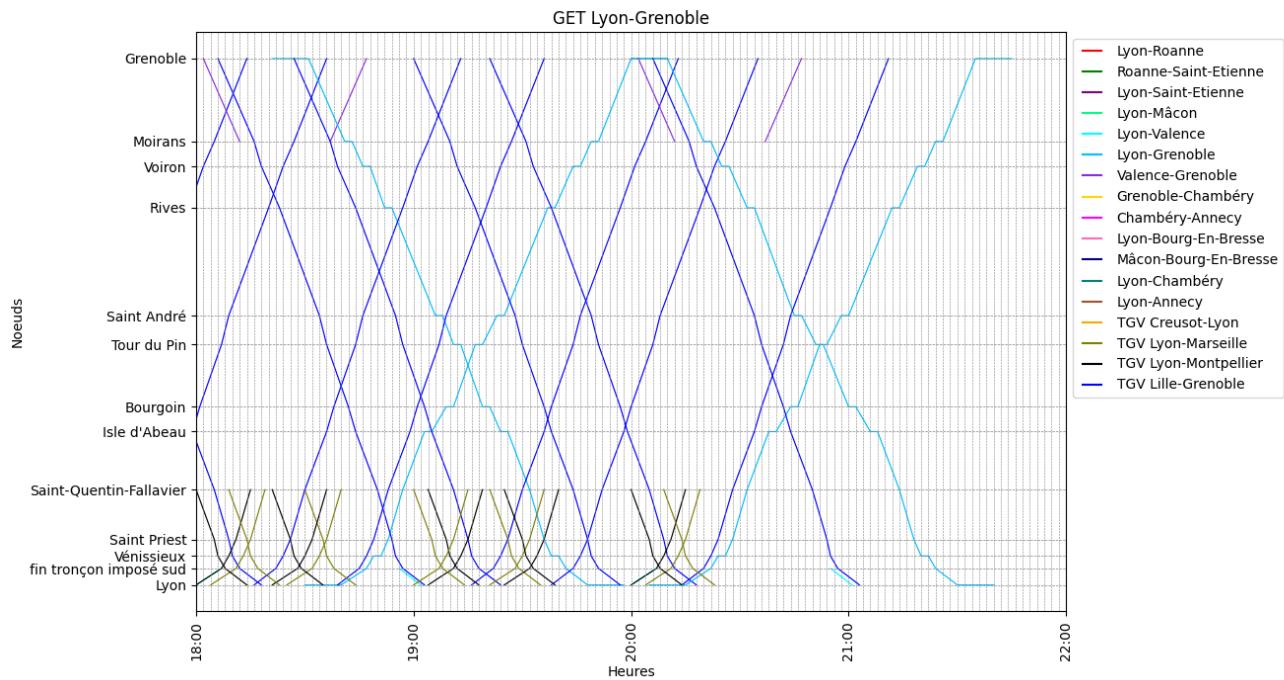




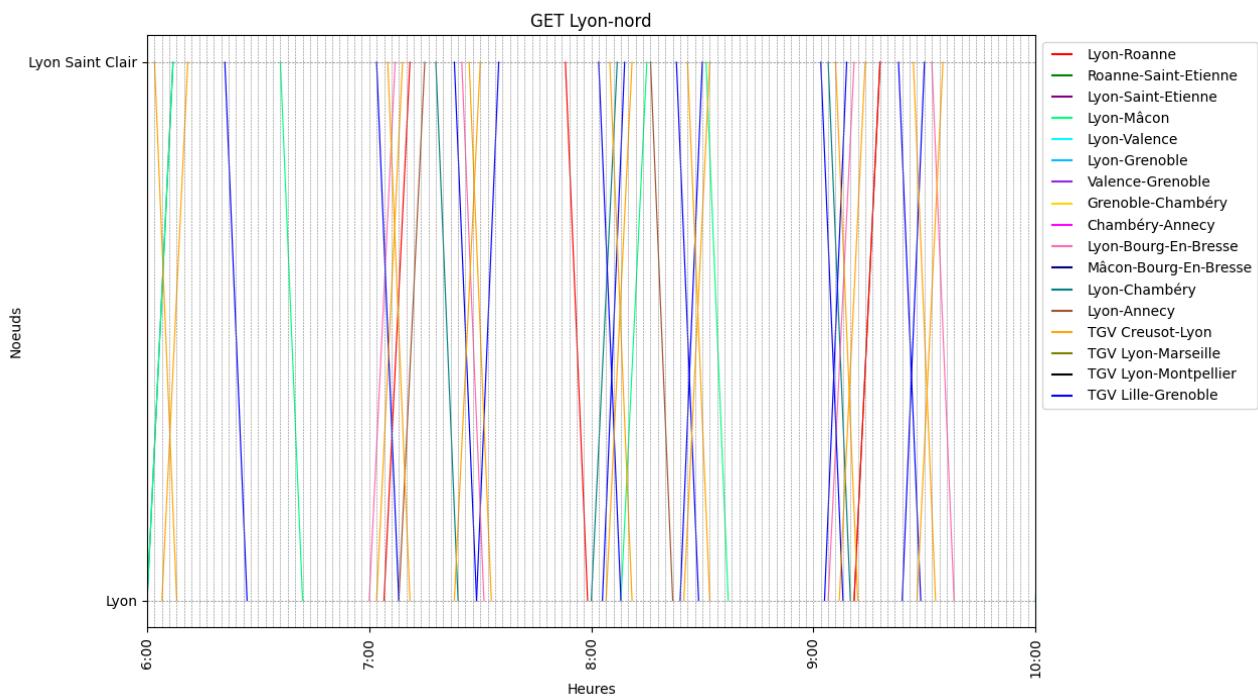
Annexe 6 : GET de la ligne Lyon-Grenoble (séparé pour plus de clarté)

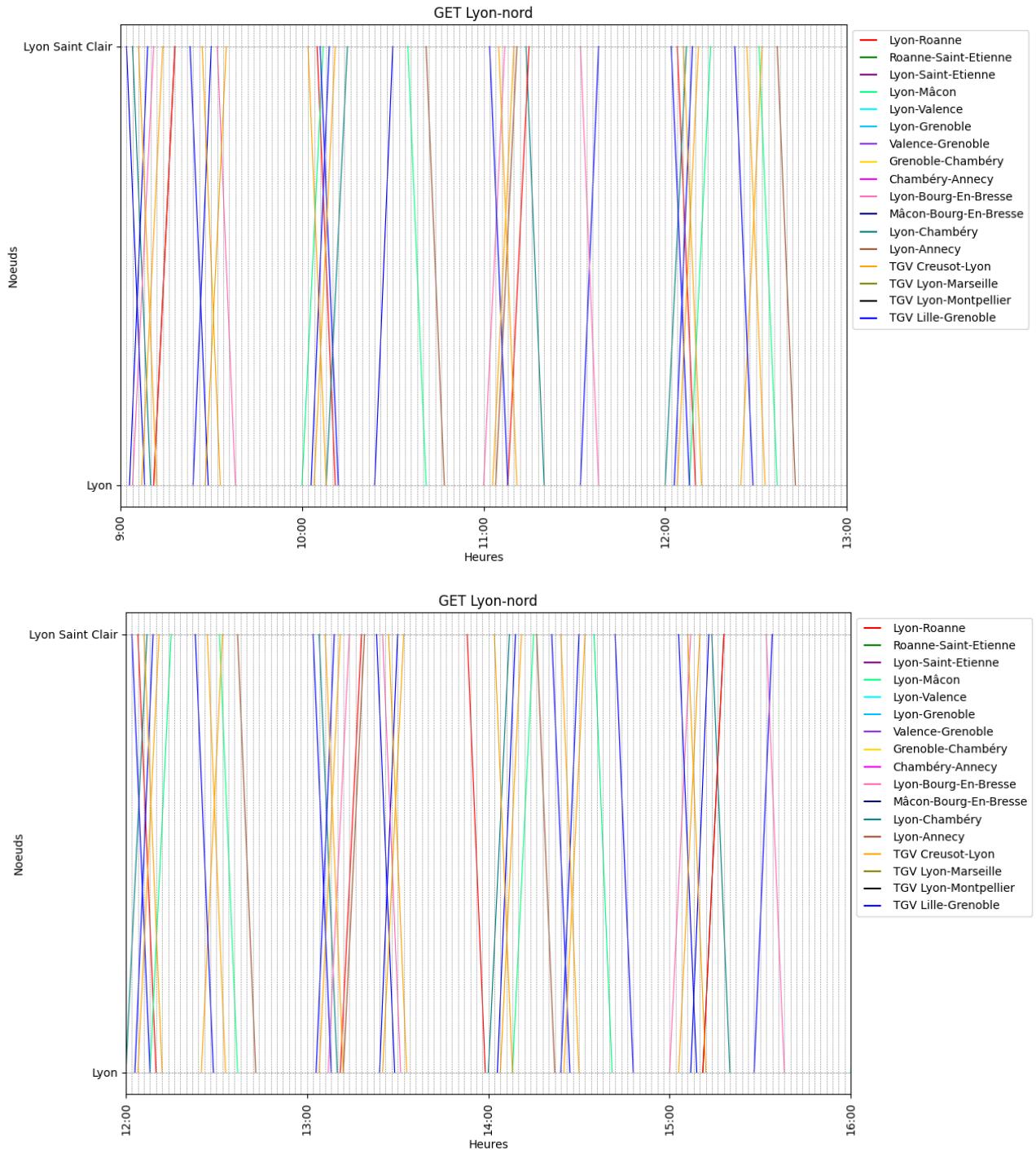


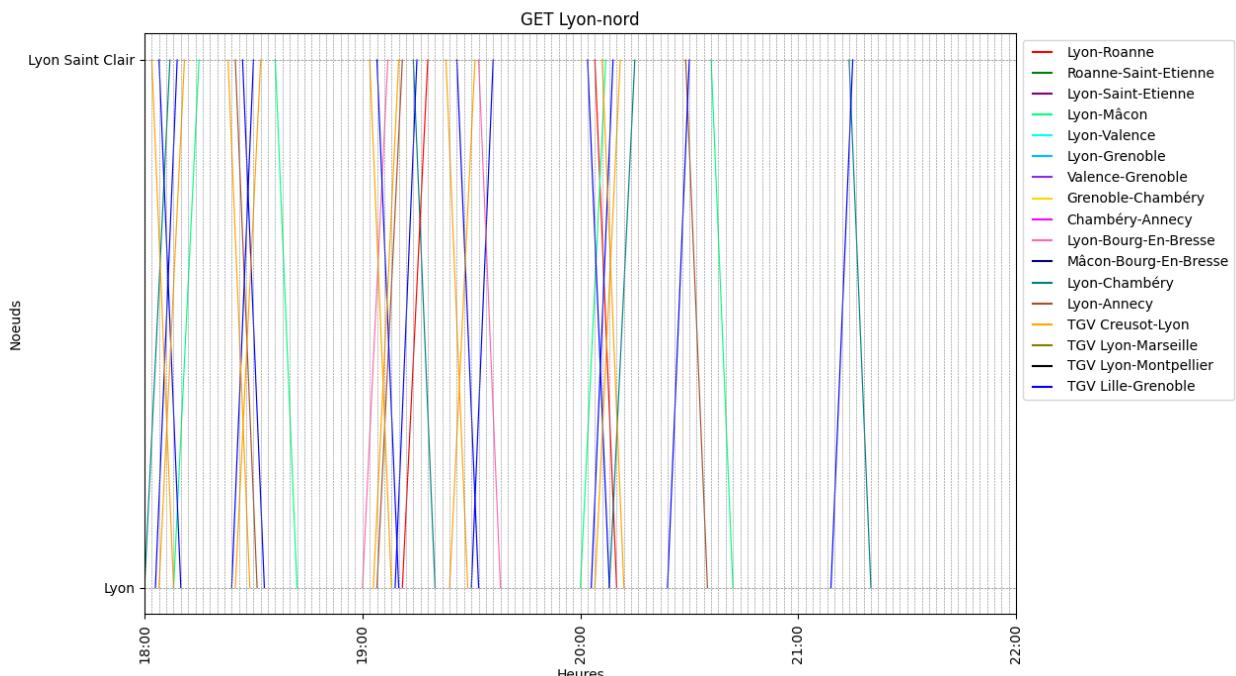
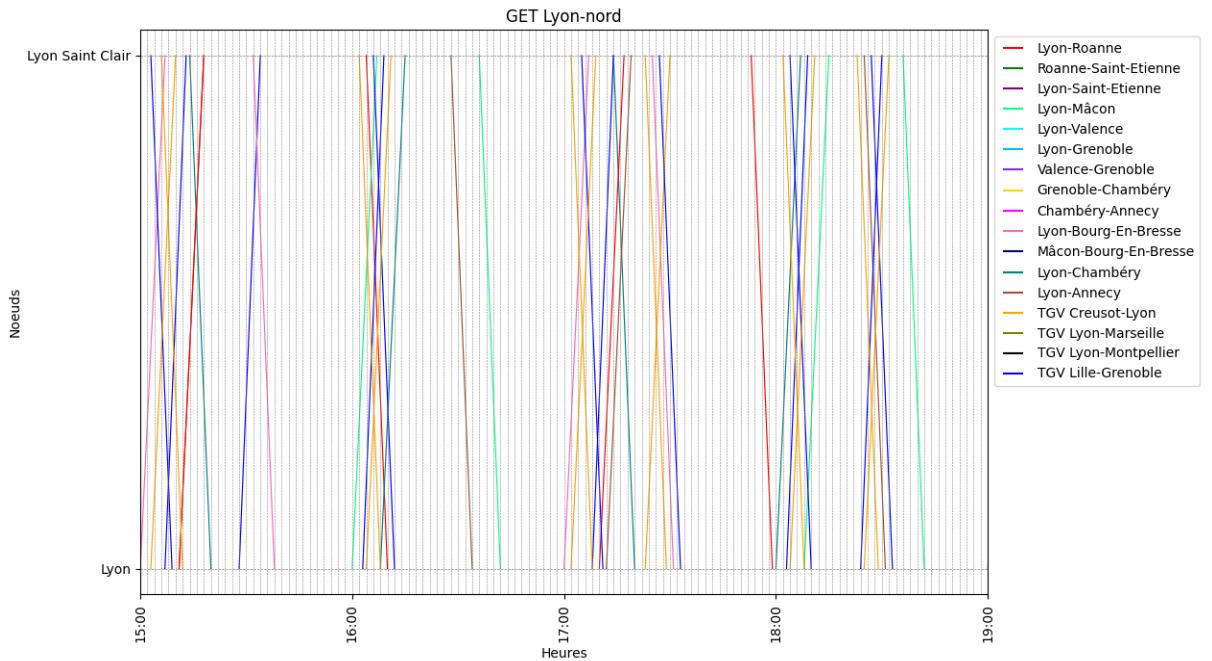




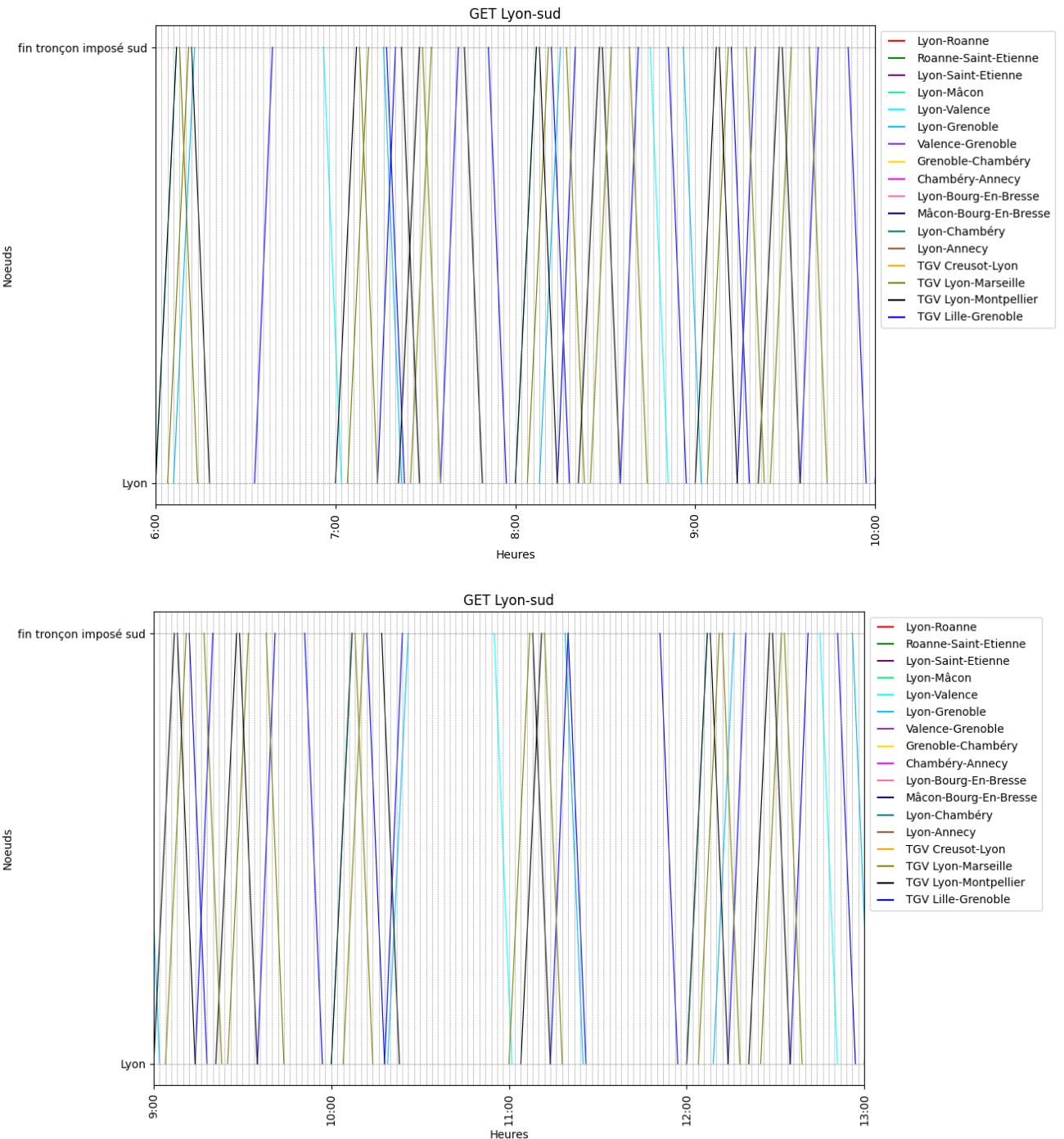
Annexe 7 : GET de la zone Lyon-Nord (séparé pour plus de clarté)

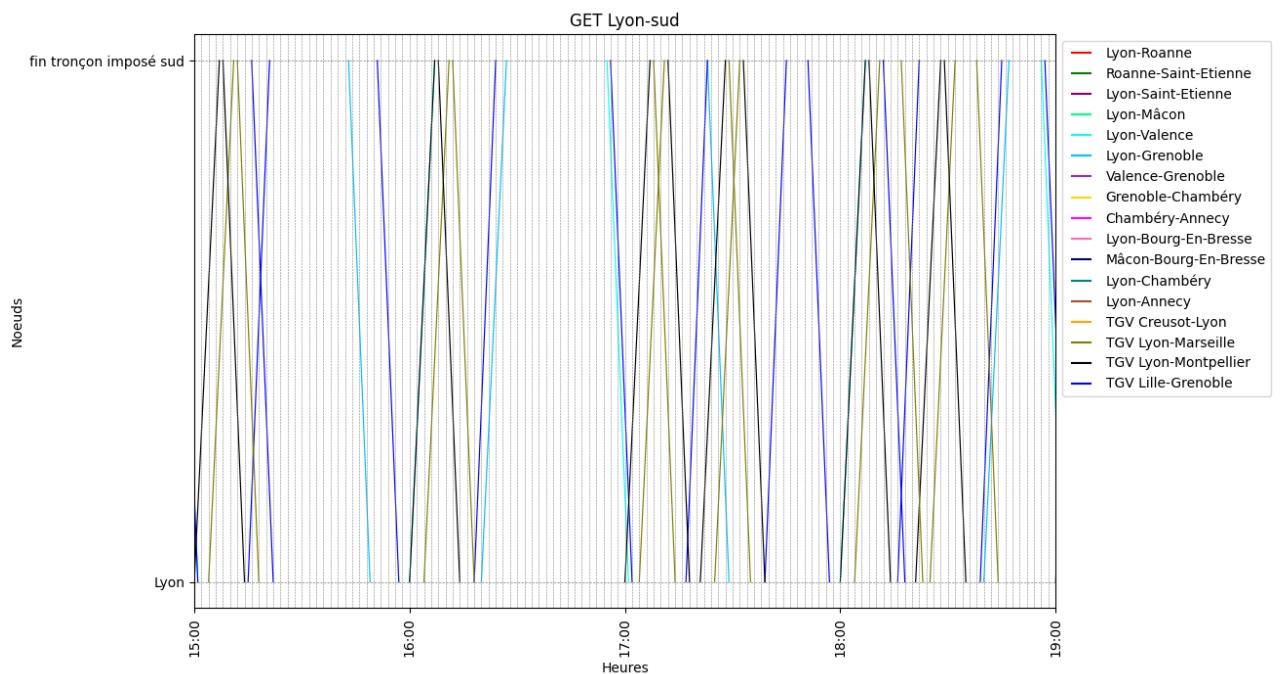
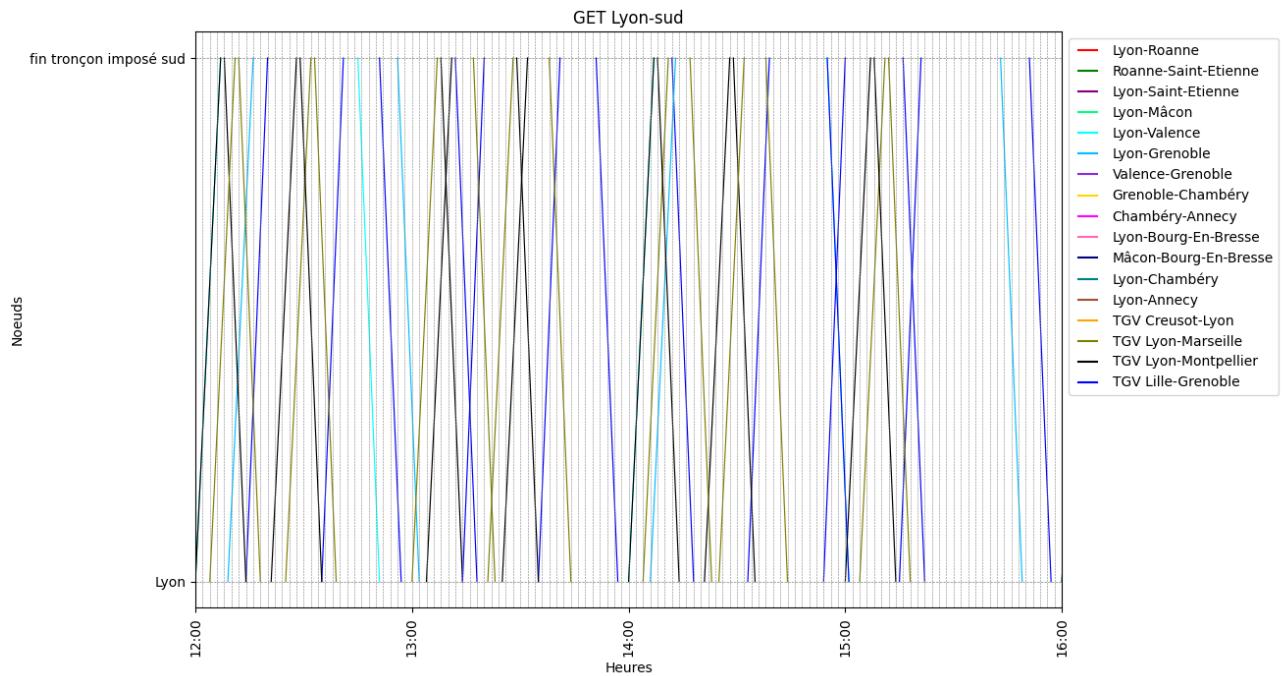


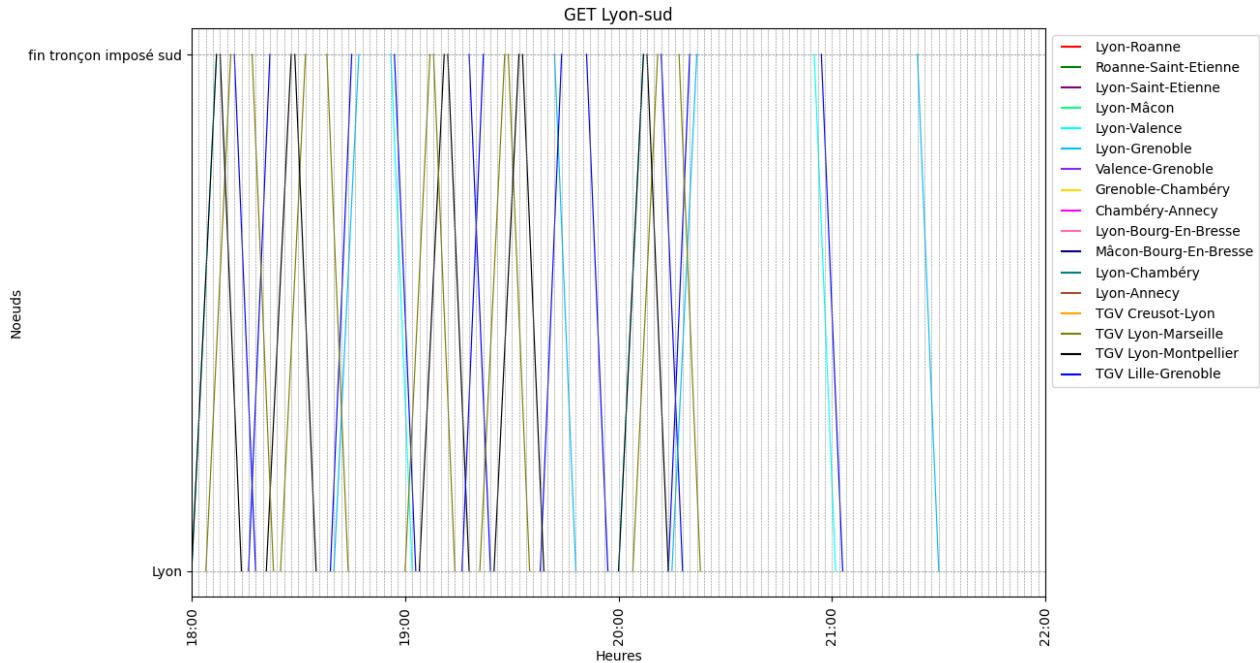




Annexe 8 : GET de la zone Lyon-Sud (séparé pour plus de clarté)







Annexe 9 : Grille horaire de la ligne Annecy-Chambéry

Gares	Horaire							
Annecy	6 h 9	8 h 30	10 h 22	12 h 17	14 h 13	16 h 22	18 h 13	20 h 15
Rumilly	6 h 21	8 h 40	10 h 34	12 h 27	14 h 25	16 h 34	18 h 25	20 h 27
Aix-les-bains	6 h 34	8 h 51	10 h 47	12 h 38	14 h 38	16 h 47	18 h 38	20 h 40
Chambéry	6 h 42	8 h 58	10 h 55	12 h 45	14 h 46	16 h 55	18 h 46	20 h 48

Annexe 10 : Grille horaire de la ligne Lyon-Valence

Gares	Horaire							
Lyon	6 h 0	8 h 0	10 h 0	12 h 0	14 h 0	16 h 0	18 h 0	20 h 0
Chasse sur Rhône	6 h 15	8 h 13	10 h 15	12 h 13	14 h 15	16 h 15	18 h 15	20 h 15
Vienne	6 h 23	8 h 18	10 h 23	12 h 18	14 h 23	16 h 23	18 h 23	20 h 23
Saint Rambert	6 h 39	8 h 32	10 h 40	12 h 32	14 h 40	16 h 39	18 h 39	20 h 40
Tain	6 h 53	8 h 44	10 h 54	12 h 44	14 h 54	16 h 53	18 h 53	20 h 54
Valence	7 h 1	8 h 52	11 h 2	12 h 52	15 h 2	17 h 1	19 h 1	21 h 2

Annexe 11 : Grille horaire de la ligne Lyon-Mâcon

Gares	Horaire							
Lyon	6 h 0	8 h 8	10 h 0	12 h 8	14 h 8	16 h 0	18 h 8	20 h 0
Saint Germain	6 h 18	8 h 24	10 h 18	12 h 24	14 h 26	16 h 18	18 h 26	20 h 18
Villefranche	6 h 27	8 h 30	10 h 27	12 h 31	14 h 35	16 h 27	18 h 35	20 h 27
Mâcon	6 h 42	8 h 45	10 h 42	12 h 45	14 h 50	16 h 42	18 h 50	20 h 42