

# IL DIT QU'IL VOIT PAS LE RAPPORT

**Sexuel le rapport, sexuel.**

22 avril 2015

Vincent DUFOUR-DÉCIEUX  
Odilon FORMERY  
Ahmed KRARTI  
Louis LEGRAND  
Augustin LENORMAND  
Camille MASSET



# 1

## MODÉLISATION NUMÉRIQUE

---

### 1.1 MOTIVATIONS ET OBJECTIFS

---

#### 1.1.1 • MOTIVATIONS

En parallèle de notre étude de la diffusion du véhicule électrique, nous avons travaillé sur le modèle permettant d'évaluer l'impact futur des véhicules électrique sur la consommation globale d'électricité. L'usage de l'outil informatique nous semblant essentiel pour une telle modélisation, étant donné le nombre important de paramètres et la taille de nos échantillons de calculs. Nos efforts se sont donc tournés vers l'élaboration d'un algorithme de calcul capable de produire une courbe représentant la puissance nécessaire à la recharge d'un véhicule électrique au cours du temps.

En répétant cet algorithme pour un nombre donné de véhicules, et en sommant les puissances calculées, on obtient ainsi une courbe de charge globale pour cette flotte.

Cette production visuelle nous a semblé la plus à même d'être exploitée dans le cadre de notre étude. Un excès de précision n'aurait pas eu de sens, étant donné la nature statistique de nos données, et les différentes approximations et extrapolations effectuées.

#### 1.1.2 • OBJECTIFS

Les objectifs principaux de notre travail de modélisation sont donc :

1. Produire un résultat visuel et facilement exploitable de la charge imposée au réseau par une flotte de véhicules électrique, flotte la plus représentative possible de la réalité (Modèle de véhicule, utilisation, ...)
2. Permettre de modifier aisément certains paramètres du modèle comme la taille de l'échantillon ou les hypothèses de flexibilité (SmartGrid, Vehicule to grid).
3. Présenter un algorithme et un cheminement logique clair et détaillé, pour permettre une réutilisation ou une adaptation ultérieure aisée.

Nous nous sommes donc rapidement attachés à développer une structure et un organigramme du code assez précis. Ainsi même si des contingences techniques apparaissaient et retardaient notre production d'un programme fonctionnel, et donc de courbes résultats, notre travail pourrait tout de même être une base suffisamment claire et modulaire pour une exploitation future.

### 1.2 CHEMINEMENT DE PENSÉE

---

Après un premier algorithme naïf et peu exploitable, présenté dans le rapport intermédiaire, nous avons choisi d'aborder le problème différemment. Pour notre modèle final nous avons décidé de travailler avec une approche plus atomique. Nous avons en effet choisi de considérer les véhicules individuellement, et définis de manière indépendante — au lieu d'un véhicule moyen — et de faire évoluer chaque véhicule pendant la durée de la simulation selon ses caractéristiques propres.

Cela nous a permis de définir nos véhicules avec une cohérence physique plus grande que lors de l'étude d'un véhicule moyen ; par exemple l'utilisation de distributions probabilistes dans l'affectation de certains paramètres permet de simuler le comportement individuel aléatoire d'un véhicule réel.

Le suivi d'un grand nombre de ces véhicules, permet ensuite d'obtenir une courbe moyenne de consommation pour une flotte.

Cette méthode probabiliste de définition des véhicules s'apparente à une approche de type Monte-Carlo, ou le loi des grands nombres justifie l'utilisation de paramètres aléatoires de loi adaptées pour modéliser des systèmes dont la résolution déterministe est impossible.

Une fois notre approche globale définie nous avons été amenés à préciser les paramètres à étudier et à prendre en compte dans notre modèle. En effet la définition de ces paramètres et leurs dépendances allaient fortement conditionner la manière dont l'algorithme et le code seraient structurés.

Les paramètres retenus sont donc issus d'un travail en amont, et leur définition a été progressivement affinée au fur et à mesure de l'élaboration du modèle.

### 1.3 HYPOTHÈSES ET PARAMÈTRES

**Hypothèses de base.** Tout d'abord, il est important de noter que nous n'avons travaillé que dans le cadre d'une journée de semaine non chômée, avec des véhicules effectuant pour l'essentiel des trajets domicile-lieu de travail. Ce choix de situation, comme nombre des autres hypothèses présentées plus bas, s'est imposé à nous pour des questions de simplicité de modélisation. De plus il nous a semblé le plus représentatif d'une situation réelle.

Afin de représenter les déplacements du véhicule, nous avons retenu trois lieux de stationnement où l'utilisateur est susceptible d'utiliser une borne de recharge :

- Sa maison ;
- Son lieu de travail ;
- La voie publique, qui regroupe toutes les autres destinations possibles (centre commercial, centre ville, école des enfants, lieu de loisir, etc.).

Nous avons aussi considéré quatre états de mouvements possible pour un véhicule :

- En train de rouler : le véhicule se déplace ;
- Garé non branché : le véhicule est à l'arrêt et n'est pas relié à une borne de recharge ;
- Branché en charge : le véhicule est à l'arrêt, relié à une borne, et se recharge ;
- Branché par en charge : le véhicule est à l'arrêt, relié à une borne, mais ne se recharge pas (par utilisation d'une recharge différée, ou parce que le véhicule est pleinement chargé).

**Paramètres physiques.** Nous avons retenu cette liste de paramètres physiques caractérisants nos véhicules :

- Le type de véhicule, à savoir : Véhicule particulier (VEP), Véhicule d'entreprise (VEE), ou Véhicule d'auto partage (VAP) ;
- Le modèle de véhicule : nous en avons retenus 5 différents (Renault Zoé, Nissan Leaf, Bolloré Blue Car, Smart ForTwo et Tesla Model S) ;
- La capacité de la batterie du véhicule (en  $\text{kW}\cdot\text{h}$ ) et la consommation (en  $\text{kW}\cdot\text{h}\cdot\text{km}^{-1}$ ), dépendant directement du modèle de véhicule ;
- La vitesse : supposée constante pour chaque véhicule, elle est initialisée pour chaque véhicule à partir d'une distribution gaussienne centrée à  $38,75\text{ km}\cdot\text{h}^{-1}$ , valeur moyenne pour les déplacements des véhicules en France, et un écart type de 10.
- La longueur d'un trajet : de même que pour la vitesse, nous effectuons l'hypothèse que pour un véhicule donné, tous ses trajets font la même longueur. Cette longueur est tirée selon la répartition statistique des distances parcourues par les véhicules en France ;
- Le nombre de trajets effectués en une journée : de même que pour les deux paramètres précédents, chaque véhicule effectue un nombre de trajet fixe, initialisé selon une distribution gaussienne, dont les paramètres sont issus des données statistiques à notre disposition ;
- L'accès aux bornes : chaque véhicule a éventuellement à sa disposition une borne aux différentes positions possibles (ex : le VEP n° 1 possède une borne chez lui, mais ni au travail, ni dans les lieux publics qu'il fréquente, tandis que le VEP n° 2 possède une borne chez lui, et également sur son lieu de travail ou son employeur a mis en place des bornes à disposition des employés). Cette répartition est issue de données statistiques a priori, mais peut être modulée pour générer des projections dans divers cas futurs.

- L'état de charge (State of Charge ou SOC) : évoluant au cours du temps, évalué en pourcentage de la capacité totale de batterie ;
- L'état de mouvement du véhicule et sa position, tels que définis ci-dessus ;
- Les horaires de trajets du véhicule : à partir de données statistiques sur les horaires de trajets réels, et du nombre de trajets qu'il va effectuer, chaque véhicule se voit affecter une liste d'horaires à laquelle il est censé partir de sa position actuelle (s'il est insuffisamment chargé il attendra le temps nécessaire à sa recharge).

Dans cette liste de paramètres apparaissent déjà un certain nombre d'hypothèses, qui peuvent sembler assez fortes, sur la définition d'un véhicule : vitesse et longueur des trajets fixes ou encore horaires des trajets définis à l'avance et pour toute la durée de la simulation.

Ces hypothèses nous ont semblé nécessaires pour pouvoir créer un algorithme d'une complexité raisonnable. De plus ces écarts à la réalité physique d'un véhicule sont compensés par l'utilisation d'un grand nombre de véhicules de paramètres aléatoires lors des modélisations. Comme les variables aléatoires sont choisies en adéquation avec les données statistiques réelles, la loi des grands nombres assure la convergence de nos simulations avec des véhicules virtuels vers les valeurs attendues dans la réalité.

**Hypothèses supplémentaires.** A ce qui a été présenté précédemment nous avons ajoutés un certain nombre de variables et d'hypothèses qui ont un sens physique, mais dont les définitions relèvent plus de l'arbitraire et du *qualified guess* que de données statistiques. Tous ces paramètres font partie de la définition d'un véhicule et même si certains évoluent au cours de la simulation :

- Les destinations des véhicules sont choisies a priori.
  - Pour un VEP nous avons considéré divers cas horaires :
    - de 7h à 11h tous les trajets d'un particulier le conduisent sur son lieu de travail, et dans le cas où il est déjà sur son lieu de travail un trajet l'amène dans un lieu public (déplacement professionnel) ;
    - de 11h à 14h un trajet depuis un lieu public (ou il a pris son repas par exemple) l'amène sur son lieu de travail, et un trajet depuis sa maison ou son lieu de travail l'amène aléatoirement à l'un des deux autres lieux possibles (trajet travail->maison pour le repas par exemple, ou maison->lieu extérieur pour aller chercher les enfants à l'école) ;
    - de 0h à 7h et de 14h à 24h un trajet depuis la maison l'amène sur un lieu public (école des enfants, sortie le soir, courses, etc.), et tout autre trajet a pour direction la maison.
  - Pour un VEE il y a aussi une distinction horaire, mais plus simple :
    - s'il est plus de 18h le véhicule rentre sur son lieu de travail, qui dans le cas d'un véhicule d'entreprise est sa base de départ ;
    - sinon tous les déplacements sont des déplacements professionnels qui vont vers un lieu public.
  - Enfin pour un véhicule d'auto-partage le trajet est aléatoirement vers un lieu public, ou un lieu de travail.

Ces destinations sont déterminées lors de l'initialisation d'un véhicule, et un parcours de ce tableau vérifie que chaque véhicule accède à une borne au cours de ses trajets, quitte à recommencer la création du véhicule si celui-ci ne trouve pas de bornes sur son trajet.

- Nous avons imposés certaines contraintes sur la répartition des bornes des VEE et VAP :
  1. Un VEE aura toujours une borne sur son lieu de travail ;
  2. Un VAP aura toujours une borne disponible dans un lieu public (nous faisons l'assomption que l'utilisateur le ramène toujours près d'une borne après utilisation).
- Pour prévenir la "panne sèche" d'un véhicule au cours d'un trajet, nous avons introduit une variable `socMin` qui contient l'état de charge minimum requis pour voyager jusqu'à la prochaine borne. Le calcul de cette variable est simplifié par la connaissance au préalable des destinations, de l'accès aux bornes et de la distance des trajets. Cela peut paraître un peu artificiel dans le cadre de notre modèle, mais dans la réalité un utilisateur a la plupart du temps connaissance de ses capacités de recharge au cours du trajet, et les prends en compte lorsqu'il débranche ou non son véhicule. Nous supposons que le véhicule ne peut redémarrer s'il n'a pas atteint un état de charge supérieur ou égal à `socMin`.

## 1.4 DESCRIPTION DE L'ALGORITHME

L'algorithme se décompose donc en deux parties distinctes :

1. Tout d'abord l'initialisation du véhicule, qui doit définir tous les paramètres décrits ci-dessus, et s'assurer que ceux ci caractérisent un véhicule fonctionnel, c'est à dire un véhicule qui a accès à des bornes, qui peut faire tous ses trajets, etc. Cette partie est organisée de manière très linéaire, avec une succession d'affectation et de vérifications qui aboutissent à la définition du véhicule. Elle est effectuée une fois par véhicule. Sa structure est détaillée au paragraphe 1.4.1
2. Ensuite la partie d'évolution. Elle consiste essentiellement en une série de tests sur l'état du véhicule (quel est son état de mouvement, son état de charge, le nombre de trajets à effectuer, etc) qui modifient ces mêmes paramètres d'état selon les résultats obtenus. C'est une boucle que chaque véhicule parcourt une fois par pas de temps. C'est dans cette partie que va être modifié le tableau résultat `puissanceReseau`, qui contient la puissance nécessaire à la recharge de la flotte considérée pour chaque pas de temps. La structure de cette boucle est illustrée par les diagrammes 1 et 2, et commentée dans le paragraphe 1.4.2.

### 1.4.1 • INITIALISATION DES VÉHICULES

Comme mentionné ci-dessus, l'initialisation des véhicule est un processus séquentiel, où l'ordre des étapes dépend de manière fondamentale des dépendances entre les paramètres. Une partie importante de sa conception a donc été de bien distinguer ces dépendances pour pouvoir écrire un programme ayant du sens physiquement, et avec des étapes les plus élémentaires possibles.

Voici donc les étapes de l'initialisation d'un véhicule :

1. Initialisation du SOC (`soc`) à 100 % : ab initio tous les véhicules sont chargés.
2. L'état de mouvement (`etatMouvActuel`) est défini comme branché en charge (`BRANCHE_EN_CHARGE`).
3. L'état de mouvement suivant, variable interne utilisée dans la boucle d'évolution, est défini de même : `etatMouvSuivant = BRANCHE_EN_CHARGE`.
4. Initialisation de la distance parcourue et du nombre de trajets effectués par le véhicule à 0 : `distanceParcourue = 0`; `nbTrajetsEffectues = 0`.
5. Initialisation du type de véhicule (`typeVehicule`) : VEE, VAP ou VEP.
6. Initialisation de la position (`position`) de départ du véhicule.
7. Définition du modèle du véhicule (`modele`).
8. Initialisation de la capacité et de la consommation du véhicule, qui dépendent du modèle de véhicule (`consommation, capacite`).
9. Initialisation de la vitesse de déplacement et de la longueur des trajets tels que présentés plus hauts (`vitesse, longueurTrajet`).
10. Initialisation du nombre de trajet par jour (`nbTrajets`).
11. Disponibilité des bornes : maison OUI/NON, lieu de travail OUI/NON, lieu public OUI/NON. (tableau `accesBornes`).
12. Test de la disponibilité d'au moins une borne pour le véhicule.
13. Calcul des horaires de départs (`horaireDepart`)
14. Vérification qu'il n'y a pas chevauchement entre un trajet et le suivant, c'est à dire que le véhicule à bien fini le trajet précédent au moment ou il doit effectuer le suivant.
15. Vérification que le véhicule passe bien par une borne au cours de la journée.
16. Initialisation du SOC minimal (`socMin`) à la valeur nécessaire pour atteindre la prochaine borne.

Cette initialisation correspond au constructeur `Vehicule::Vehicule(int deltaT)` de la classe `Vehicule.cpp`. Le paramètre `deltaT` correspond au pas de temps de la simulation. Une fois le véhicule correctement initialisé nous le faisons évoluer dans la boucle de simulation.

#### 1.4.2 • BOUCLE DE SIMULATION

L'organisation logique de la partie évolutive de la simulation est présentée dans le diagramme ci dessous (figure 1). Pour plus de lisibilité le bloc transition est présenté à part, dans la figure 2. Ces deux blocs correspondent respectivement aux méthodes `double Vehicule::simulation(int temps, int deltaT)` et `int Vehicule::transition(int temps, int deltaT)` de la classe `Vehicule.cpp`. L'opération de bouclage de la méthode `simulation` est effectué lors de son appel dans la fonction `main`.

La méthode `simulation` prend en paramètre le `temps` actuel — mesuré en nombre de pas de temps `deltaT` — et retourne la puissance demandée par le véhicule sur lequel on appelle cette méthode a l'instant `temps`. La méthode `transition` prend les mêmes paramètres que `simulation` et retourne un entier correspondant à l'état de mouvement dans lequel sera le véhicule au prochain pas de temps (`etatMouvSuivant`).

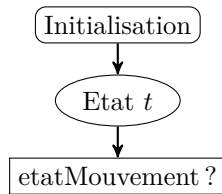


FIGURE 1 – Logigramme de la méthode `simulation`

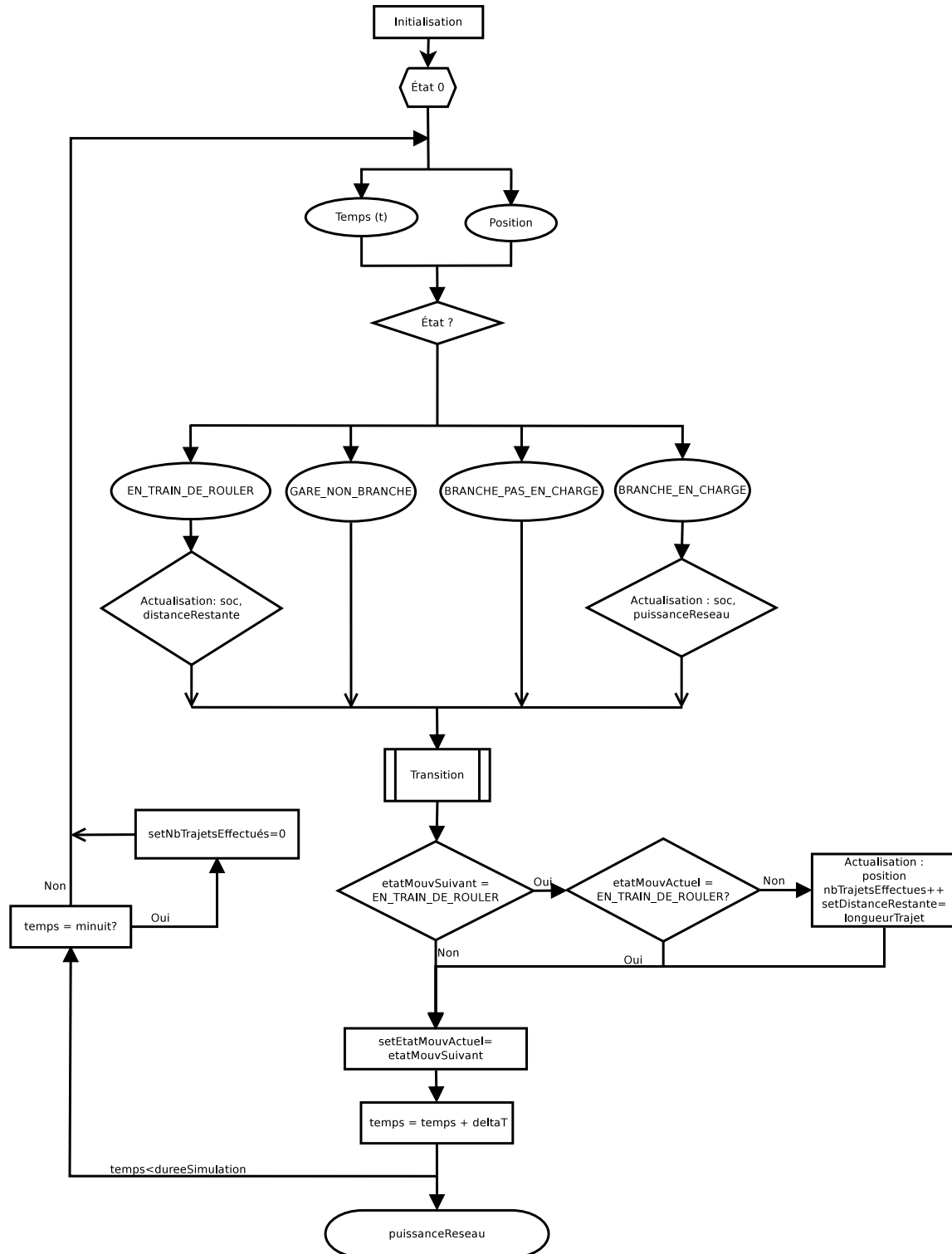
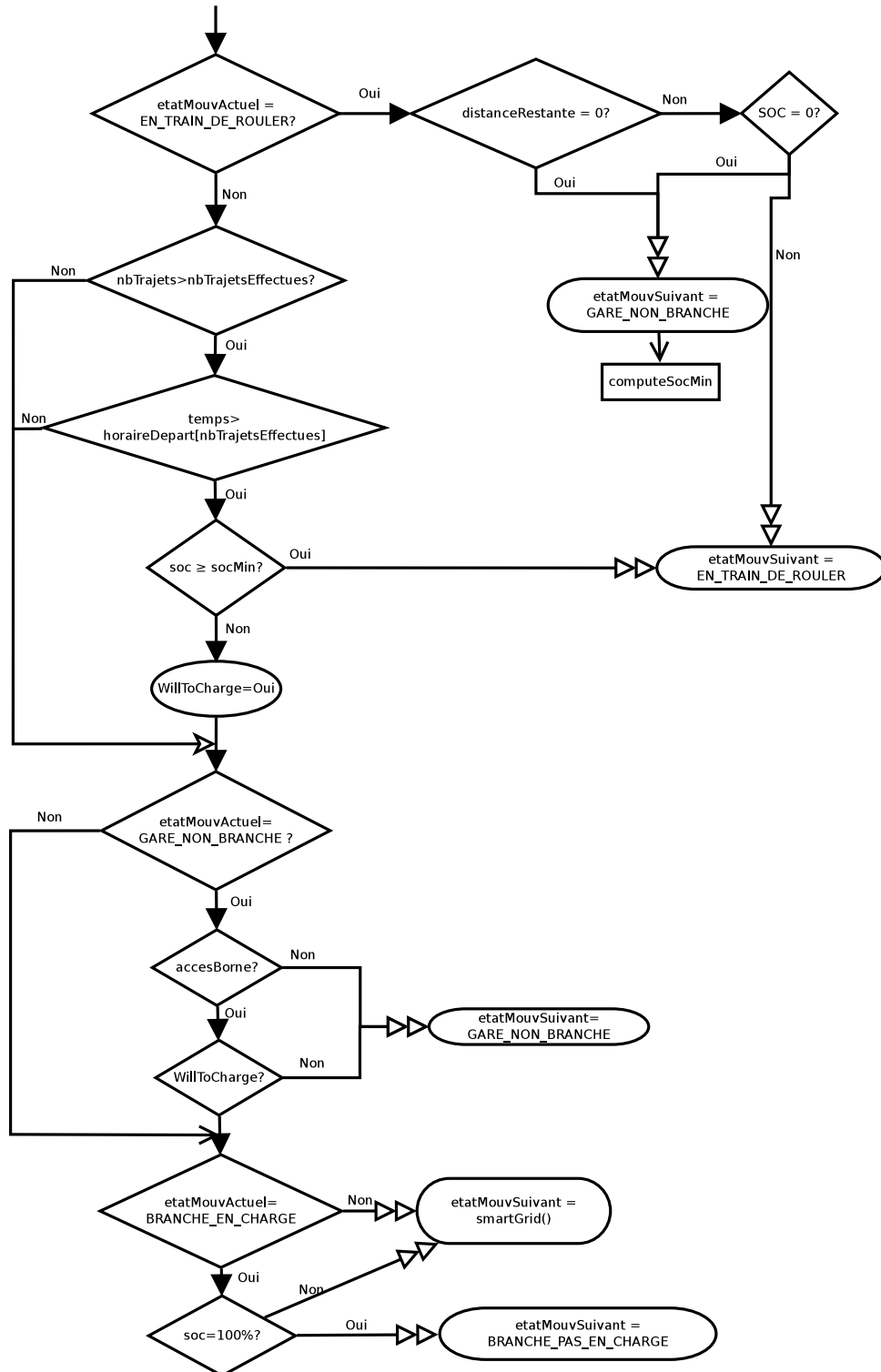


FIGURE 2 – Bloc Transition





## 1.5 RÉSULTAT

---

2 jours == 10 jours mentionner