Simulation Netlogo

Simulation Netlogo

MASTER 2

Spécialité : MIASHS

Parcours : IMA

*Sous la direction de Pierre Chauvet*

Session : Janvier 2023

CHENON Théo et LAMBERT Clara

Faculté Sciences

Institut des mathématiques appliqués

Année universitaire : 2022 -2023



**CHARTE DE NON-PLAGIAT**

**Protection de la propriété intellectuelle**

Tout travail universitaire doit être réalisé dans le respect intégral de la propriété intellectuelle d’autrui. Pour tout travail personnel, ou collectif, pour lequel le candidat est autorisé à utiliser des documents (textes, images, musiques, films etc.), celui-ci devra très précisément signaler le crédit (référence complète du texte cité, de l’image ou de la bande-son utilisés, sources internet incluses) à la fois dans le corps du texte et dans la bibliographie. Il est précisé que l’UCO dispose d’un logiciel anti-plagiat dans lms.uco.fr, aussi est-il demandé à tout étudiant de remettre à ses enseignants un double de ses travaux lourds sur support informatique.

*Cf. « Règlement intérieur UCO-Angers »*

Nous soussignons, Théo CHENON et Clara LAMBERT, étudiants en Master 1 MIASHS s’engageons à respecter cette charte.

Fait à Angers le 29/01/2023

Signature (pour la version imprimée) :





Table des matières

[Introduction 1](#_Toc124754024)

[I-Fonctionnement du modèle SIR 2](#_Toc124754025)

[1. Modèle SI 2](#_Toc124754026)

[2. Modèle SIR 4](#_Toc124754027)

[3. Notre Modèle 7](#_Toc124754028)

[II-Incrémentation/Application du modèle sous python 9](#_Toc124754029)

[III-Visualisation du modèle sous Netlogo 12](#_Toc124754030)

[1. Présentation Netlogo 12](#_Toc124754031)

[2. Fonctionnement du code 13](#_Toc124754032)

[3. Analyse 16](#_Toc124754033)

[IV-Conclusion 20](#_Toc124754034)

Introduction

Les épidémies de virus ont été connues comme l’un des principaux problèmes de santé causant un taux de mortalité élevé tout au long de l’histoire. Notamment la dernière épidémie la plus connue est la COVID 19, avec un taux élevé de transmissibilité, à mener à une catastrophe mondiale qui a paralysé les systèmes de santé.

Dans le cadre de notre Master 2 nous devions réaliser une étude sur la crise du COVID 19, avec différentes approches de modélisation mathématique, afin de pouvoir simuler le cours de la maladie. Au cours de la dernière année divers modèles de SIR simples et modifiés ont été élaborés pour prédire l’évolution de la COVID 19, que nous avons utilisé pour notre étude, et que nous détaillerons dans ce rapport.

Dans un premier temps, nous verrons l’aspect mathématiques du modèle SIR puis on montrera 2 modèles (un SIR et un basé sur le SIR) sur 2 graphiques afin de voir les courbes de chaque variable (sains, infectés, rétablis…). Enfin nous finirons par simuler ces modèles via NetLogo.

# I-Fonctionnement du modèle SIR

Le modèle mathématique peut être utilisé pour prédire le nombre prévu de cas à la fois. Nous ne pouvons pas prédire directement le nombres de cas de coronavirus en le considérant simplement comme une courbe exponentielle et en utilisant la régression pour donner la prédiction. En effet comme la Covid est une maladie épidémique, nous pouvons utiliser une meilleure alternative, le modèle « SIR » pour l’analyse de la transmission de la maladie. Ainsi ce modèle est utile dans le cas de toute prédiction de propagation de maladies infectieuses selon 3 critères : « Le contact des personnes », « la durée de l’infection » et « les mesures de santé (vaccin) ».

## Modèle SI

Soit N la population totale étudiée, S le nombre de personnes susceptibles d’être contaminées à l’instant t, I le nombre de personnes infectées à l’instant t.

Les états du modèle ains que leur dépendance, incarnée par le facteur B, peuvent être schématisés comme suit :

β

**I**

**S**

Sont introduits S(t) qui représente le nombre de personnes susceptibles d’être infectées à l’instant t, et I(t) le nombre de personnes infectées à l’instant t.

Ces deux états S et I sont liés par une relation de passage, qui traduit la transition unidirectionnelle des personnes nouvellement infectées de S à I.

Cette relation de passage, également appelée dynamique de ces états est mathématiquement exprimée, dans le cadre d’une modélisation continue, sous la forme du système d’équations différentielles suivant :

Dans ces équations, I(t) / N est la probabilité de rencontrer un individu infecté, donc S(t)× I(t) N représente le nombre de contact entre l’ensemble des susceptibles et l’ensemble des infectés. Enfin, β représente le nombre moyen de personne qu’un individu infecté contamine pendant une durée dt.

Ce modèle fait l’hypothèse que la population est fermée (pas de naissance, décès, immigration etc.). Ceci implique :

Donc si on pose :

Alors :

Nous avons présenté le modèle SI qui est un modèle compartimental à deux états, S & I, dans la partie suivante nous allons présenter le modèle SIR, un modèle bien différent du SI et avec 3 états S, I et R.

## Modèle SIR

Petit rappel, soit N la population totale étudiée, S le nombre de personnes susceptibles d’être contaminées à l’instant t, I le nombre de personnes infectées à l’instant t et le nouveau paramètres :

R qui est le nombre de personnes guéries à l’instant t. Nous supposons aussi qu’une personne à l´état R ne peut passer à l’état S et qu’une personne ne peut pas rester infectée indéfiniment.

Nous avons les fractions suivantes : S(t), I(t) & R(t)

Quelques premières analyses peuvent d’ores et déjà être effectuées. Par hypothèse, la population est censée être fermée. Ainsi, les seules possibilités d’un individu dans l’état S sont soit de rester, soit de quitter ce compartiment en passant dans le compartiment I. Le compartiment S ne pouvant pas être alimenté par des transitions de l’état R à l’état S. On pourra s’attendre à voir « S » décroitre en fonction du temps, pour « I » on peut s’attendre à un comportement croissant puis décroissant formant une bosse, et enfin pour R on peut s’attendre à un mouvement croissant.

Le schéma du modèle SIR, de la même forme que celui du modèle SI, est présenté de la façon suivante :

**S**

**I**

β

α

**R**

Dès lors qu’un individu susceptible est infecté, il passe de l’état S à l’état I, et y reste jusqu’à être guéri. Ainsi dès qu’il guérit, il quitte l’état I et rejoint l’état R.

Il est dès lors possible d’établir des relations de passage qui lient les 3 états de ce modèle, qui s’expriment mathématiquement par le système de trois équations différentielles suivant :

Avec :

— N le nombre total d’individus considérés

— S(t) le nombre d’individus sains à l’instant t

— I(t) le nombre d’individus infectés à l’instant t

— R(t) le nombre d’individus rétablis à l’instant t

— β et α deux constantes, détaillées ci-après

Le paramètre de passage entre l’état S et l’état I, β, à la même interprétation que dans le cadre du modèle SI : il s’agit du nombre moyen de personnes qu’un individu infecté contamine pendant une unité de temps.

α, quant à lui, est plus facilement interprétable par son inverse : α −1 représente la durée moyenne d’infection d’un individu infecté, α est donc la proportion d’infectés se rétablissant par unité de temps.

Il l’a été évoqué les paramètres β et α sont, dans le modèle SIR, des constantes. Cette hypothèse très forte, remise en question par la suite, implique que les dynamiques restent somme toutes assez rigides et donc que ce modèle ne permet pas, en l’état, de prendre en compte certains cas de figure pourtant fréquents qui apparaissent communément dans certaines épidémies. Il est en effet tout à fait envisageable, et ce cas de figure se vérifie sur certaines épidémies, que la proportion de personnes quittant l’état S pour l’état I varie au cours de l’épidémie, la même réflexion s’appliquant à la proportion de personnes quittant I pour R.

La structure du modèle SIR est un choix judicieux pour modéliser des épidémies saisonnières, parmi le panel de modèles compartimentaux classiques. En effet, elles confèrent en général une immunité de longue durée en comparaison de la durée de l’épidémie. Ainsi, si le choix est fait d’une modélisation de chaque saison épidémique, l’horizon de modélisation est suffisamment court pour valider l’hypothèse que les individus rétablis ne peuvent retourner dans l’état S, étant immunisés au moins jusqu’à la saison épidémique suivante.

## Notre Modèle

Après avoir présenté les bases du modèle SIR, il est temps à présent de parler de notre modèle que nous avons réalisé.

Les fractions suivantes de notre modèle sont les suivantes :

S(t) : Indique le nombre de sujets sains

E(t) : Infectés non contagieux

I(t) : Infectés contagieux

Q(t) : Mis en quarantaine (Infecté confirmé)

R(t) : Cas guéris

D(t) : Morts

V(t) : Vaccinés

Les coefficients :

L : Nouvelles naissances et arrivants par unité de temps

β : Taux de transmission divisé par N

α : taux de vaccination

µ : taux de mort naturelle

g : période d’incubation

s : taux d’inefficacité du vaccin

d : temps d’infection

Λ : taux de létalité

l : temps de rétablissement

r = temps jusqu’à la mort

Une probabilité de décès liés à la maladie D(t) est appliquée pour scinder l’état I en deux groupes : les personnes effectivement guéries et les personnes décédées. Notons que cette probabilité de décès dépend du temps afin d’intégrer l’effet de la vaccination dans l’estimation du nombre de décès.

Le graphe ci-dessous représente le modèle utilisé :

**D**

Λ

**S**

**E -> I**

α

**R**

α

β

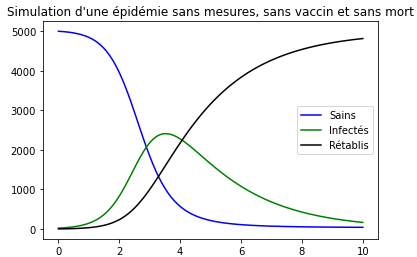
σ

Θ

µ

**V**

# II-Incrémentation/Application du modèle sous python

Nous avons réalisé 2 graphiques sous Python en utilisant la méthode RK4.  
Dans un premier temps, afin de bien comprendre le modèle, on a réalisé un premier graphique du modèle SIR :

On a une population de 5020 personnes, 5000 sont saines et 20 infectés au départ.

β = 0.0005, α = 0.5 et on effectue la simulation sur 10 jours.

Le code :

Une image contenant texte

Description générée automatiquement

Le code contient 2 fonctions. La principale, ici à droite, initialise nos variables et implémente la méthode RK4.

La méthode de Runge-Kutta 4 est donnée par l’équation :

* Avec , pente au début de l’intervalle
* , , pente au milieu de l’intervalle
* , , nouvelle pente au milieu de l’intervalle
* , pente à la fin de l’intervalle

L’idée de cette méthode est que la valeur suivante () est approchée par la somme de la valeur actuelle () et du produit de la taille de l’intervalle () par la pente estimée. La pente est obtenue par une moyenne pondérée des pentes.

Les équations S, I et R sont dans une autre fonction appelé au début de la fonction principale :

Une image contenant texte

Description générée automatiquement

Notre modèle : SEIRQDV (SIRQDV)

Petit rappel des lettres :

S(t) : Indique le nombre de sujets sains

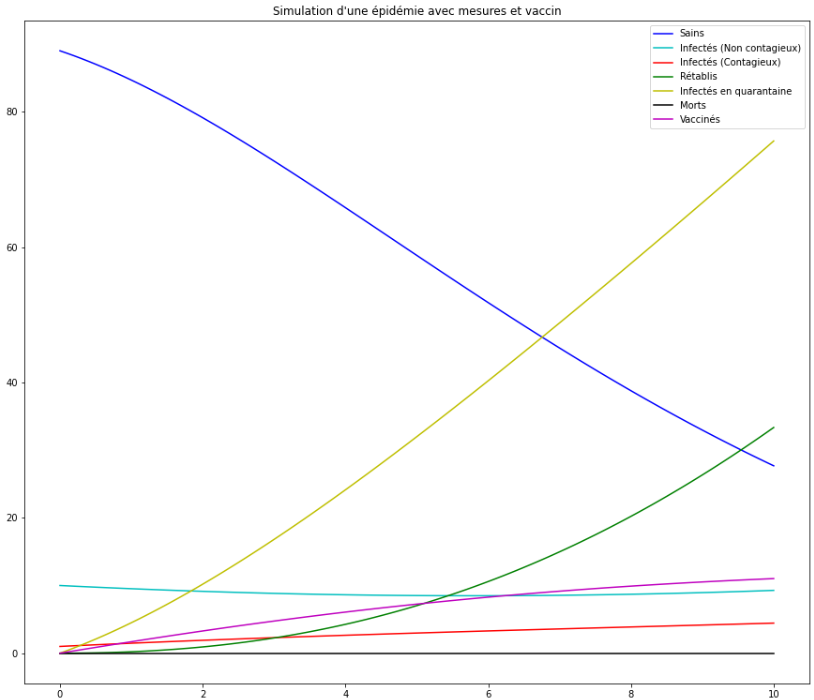
E(t) : Infectés non contagieux

I(t) : Infectés contagieux

Q(t) : Mis en quarantaine (Infectés confirmés)

R(t) : Rétablis

D(t) : Morts

V(t) : Vaccinés

On simule sur une population totale de 100 personnes, dont 10 infectées non contagieuses et 1 contagieuse au départ.

On a ici beaucoup plus de paramètres :

* L : 1/100
* β : 4/100
* α : 1.95/100
* µ : 0.000003
* g : 5.5/100
* s : 0.05
* d : 3.8
* Λ : 0.014
* l : 1/10
* r = 1/5

Le code est disponible avec ce fichier sur un jupyter notebook : « SIR\_RK4\_.ipynb »

# III-Visualisation du modèle sous Netlogo

## Présentation Netlogo

Cette année nous avons utilisé un nouveau logiciel de programmation qui se nomme « Netlogo », afin de présenter ce dernier nous avons listé les 4 principaux axes de ce logiciel :

* NetLogo est un environnement de programmation pour la modélisation/simulation de phénomènes collectifs naturels.
* Bien adapté à la modélisation de systèmes complexes composés de centaines, de milliers d’agents agissant en parallèle.
* Possibilité de « jouer » avec de nombreuses simulations en sociologie, biologie, médecine, physique, chimie, mathématiques, informatique, économie et psychologie sociale.
* Possibilité de créer ses propres modèles.

Le monde de Netlogo est constitué d’Agents, qui peuvent suivre des instructions. Les activités des différents agents s’exécutent simultanément. Il y a 3 types d’agents : tortues, patches et observateur.

* Les Tortues sont les agents qui se déplacent dans le monde. Ils correspondent aux agents vus en cours.
* Le Monde est en 2D ou 3D, divisé selon une grille (torique ou non) de patchs.
* Un Patch est une portion de sol sur laquelle les tortues peuvent se situer et se déplacer. Les patches correspondent au concept d'environnement vu en cours
* L'Observateur regarde de l’extérieur le monde des tortues et des patchs (n’est pas situé dans le monde).

## Fonctionnement du code

Dans cette partie nous parlerons uniquement de la fonction du code et ce qu’il en fait, le détail de chaque ligne est présent dans le code en question.

Dans un premier temps quand vous arrivez sur notre programme « Netlogo » vous tomberez sur ce qu’on appelle notre interface. Dans cette interface il y aura des informations que nous allons vous détaillez afin de faire fonctionner au mieux notre code, et les différents tests que vous pourrez réaliser.

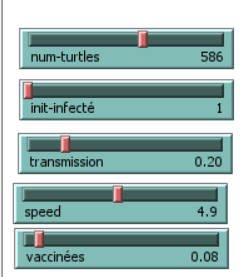
Ainsi sur cette interface nous allons dans un premier temps regarder les deux principaux boutons qui sont « SetUp » et « Go » :



Ces boutons permettent de pouvoir lancer le code en question, le bouton « Setup » permet de mettre en place tout les paramètres sélectionnés en amont, que nous détaillerons par la suite. Une fois le setup fait on peut lancer le code en appuyant sur le bouton « GO ». Nous avons mis une limite de temps afin de pouvoir voir nos courbes et éviter le programme se lance dans le vide.

Ensuite on peut retrouver différents boutons que nous appelons « Slider ». Ces Sliders permettent de choisir les différents paramètres de notre modèle.

Nous avons :



-Nous avons le nombre de personnes présentent dans le modèle

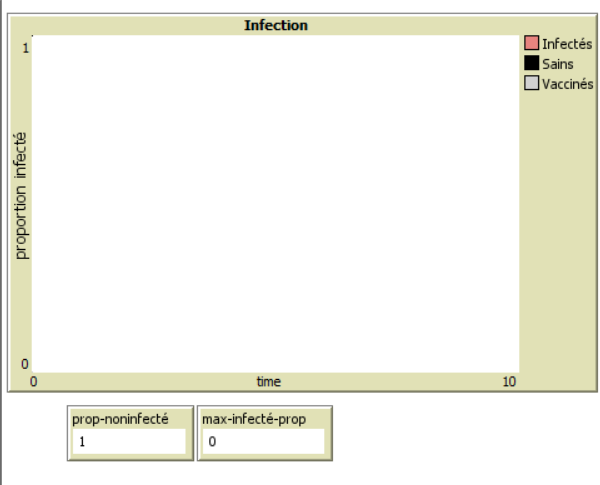
-Le nombre d’infectés au départ de l’infection

-La transmission comme son nom l’indique, définit l’infectiosité du virus (sa transmission)

-Speed définit le déplacement de nos personnes dans l’interface

-Vitesse du vaccin

Ici nous avons ce qu’on appelle le « Plot », c’est ici que les tracés vont être réalisés quand on lancera le modèle.



Dans ce plot nous avons notre population avec les personnes saines, les infectés et les personnes vaccinées. Pour cela nous devons faire du lien avec notre code pour pouvoir afficher nos commandes :

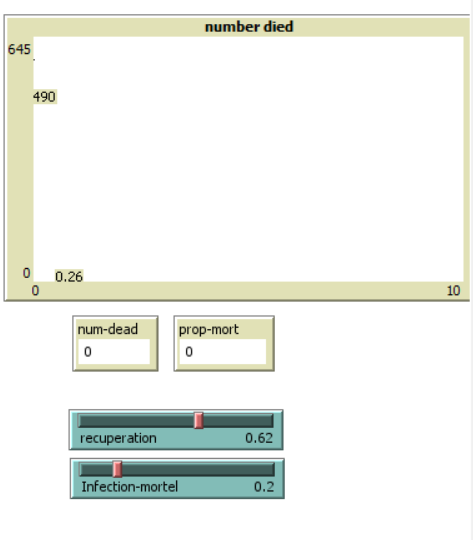
**plot (count turtles with [infecté?]) / num-turtles** ->Compte le nombre d’infectés

**plot (count turtles with [not infecté? and not immunisé?]) / num-turtles** -> Définit le nombre de personnes saines

**plot (count turtles with [immunisé?]) / num-turtles** -> Compte le nombre de personnes vaccinées

Nous avons aussi deux autres boutons que nous appelons « Monitor », qui nous permettent dans ce cas de compter la population saine au cours du temps « Prop noninfecté », ainsi que le pic max de notre population infecté « Max-infecté-prop »

Et enfin pour conclure nous avons un autre plot qui affiche le nombre de mort que nous pouvons avoir au cours de notre épidémie.



Dans ce plot on peut observer grâce aux moniteurs « Num-dead », et « Prop-mort », le nombre total de la population ainsi que le ratio moyen de mort au cours du temps.

En plus de cet affichage, nous avons mis deux Sliders, « Recuperation », et « Infection-Mortel ». Le slider « Recuperation », permet de choisir la resistance à l’infection, c’est-à-dire que si une personne est infectée elle a moins de chance de mourir. A l’inverse le slider « Infection-Mortel » rend l’infection plus violente et augmente le cas de morts.

En bonus on peut décider d’enlever le vaccin grâce au bouton « Switch », afin de voir comment peut varier l’épidémie sans aucun vaccin à disposition pendant toute l’épidémie.

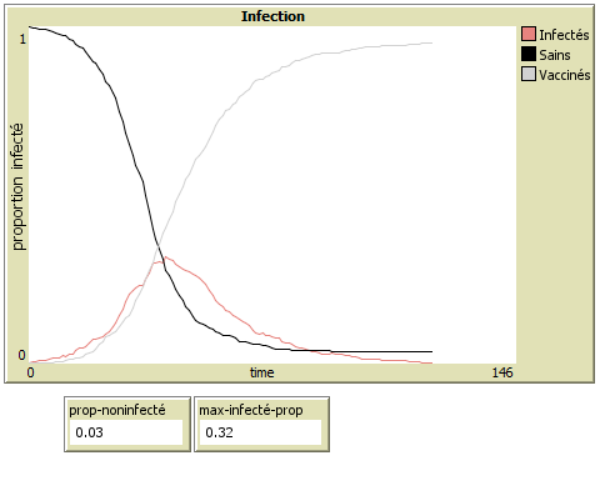


## Analyse

Maintenant que le fonctionnement de notre programme a été expliqué nous pouvons commencer à analyser les sorties que nous avons pu obtenir.

**Avec Vaccin**

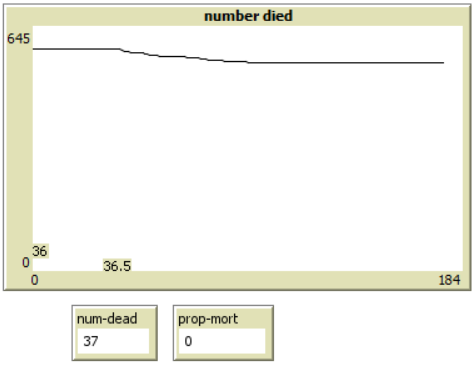
L’analyse de notre modèle SIR la plus cohérente est celle-ci :



Ici nous avons un modèle type d’épidémie, avec le nombre de personnes saines qui décroit rapidement par unité de temps, et les personnes infectées qui commencent à apparaitre. Au moment où le nombre de personnes infectées commencent à fortement croitre, un vaccin se met en place rapidement afin de stopper l’épidémie.

Une fois que toutes les personnes commencent à se vacciner la propagation de la maladie commence à diminuer, jusqu’à atteindre 0. Le moment où elle atteint 0 le modèle se stoppe et ne garde que les personnes vaccinées ainsi que les personnes saines.

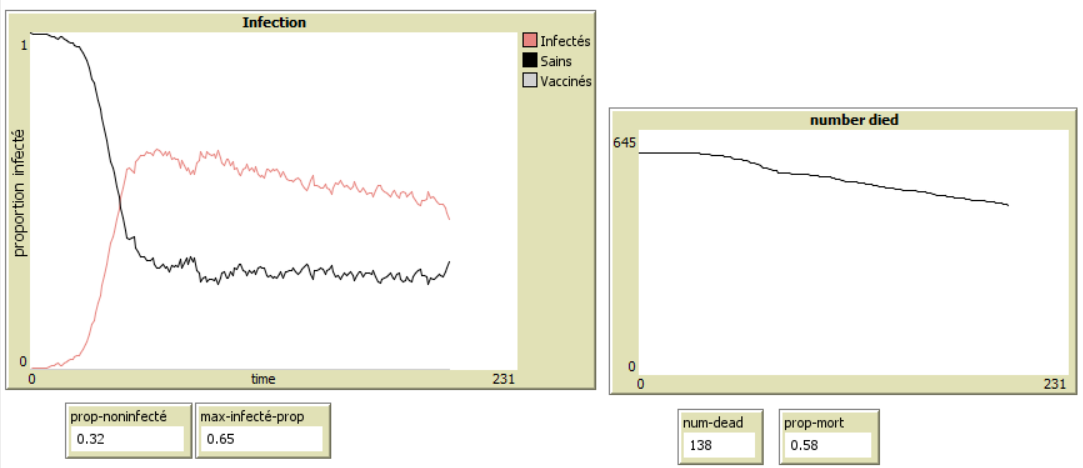
Nous remarquons que le maximum de notre population infectée sur l’ensemble de nos individus était de 32%, et que le nombre de personnes saines sans vaccination qui ont échappé à l’infection sont de 3%.



Sur ce plot nous remarquons que sur 580 personnes choisies au départ dans cette étude, 37 sont décédées de la maladie (covid), soit 6.38% de la population.

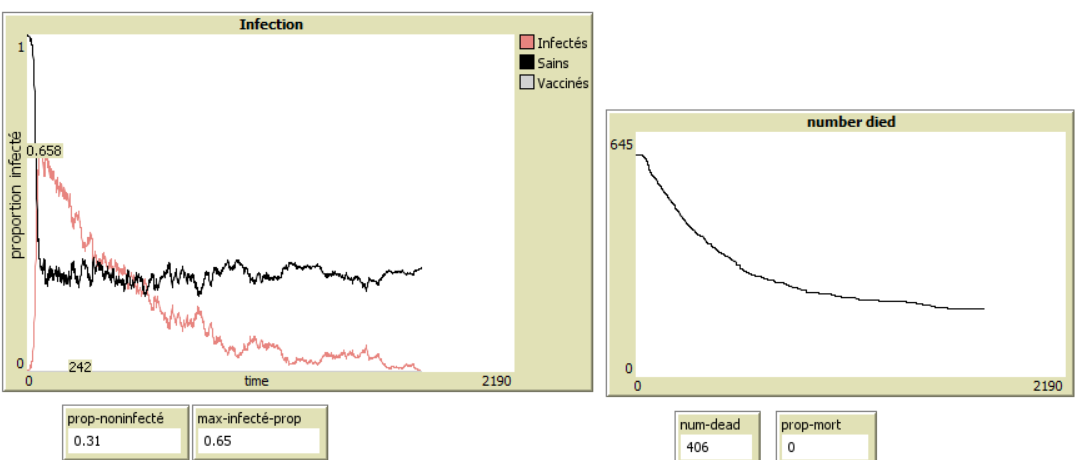
**Sans Vaccin :**

A présent nous pouvons réaliser le test mais sans mettre de vaccin :



Sur une unité de temps un peu plus élevée que notre modèle de vaccination, nous remarquons que le nombres de personnes infectées et saines sont équivalentes, malgré un nombre de personnes saines plus faible. Nous remarquons aussi que la courbe de mort décroit rapidement vers 0.

Mais si nous laissons tourner le programme sur des unités de temps encore plus élevées nous obtenons le modèle suivant :

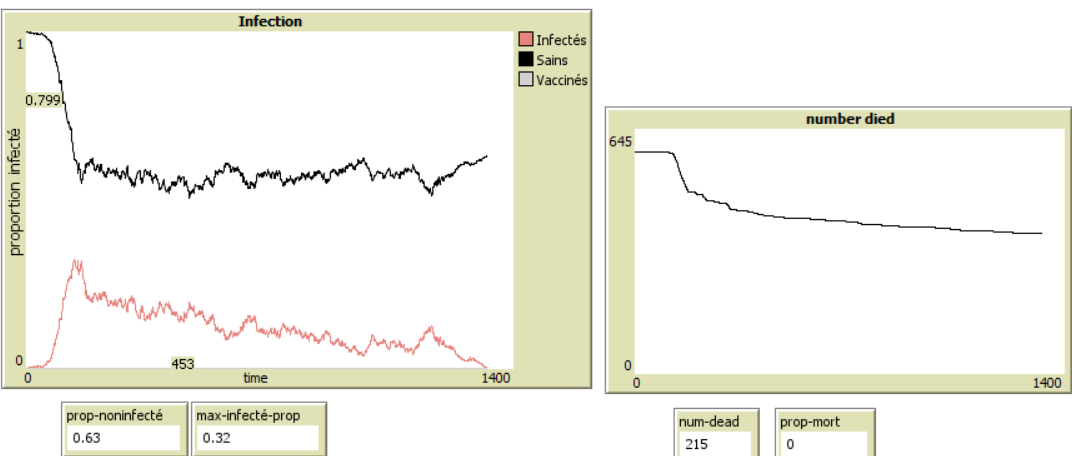


Nous remarquons à présent que l’épidémie s’est stoppée naturellement, avec la possibilité que nous avions faite, qui est que certaines personnes peuvent être rétablies en luttant contre la maladie. Ainsi ces personnes sont immunisées contre la maladie. Mais à quel prix ?

En effet si on regarder le graphe de droite nous pouvons voir que le nombre de mort est très élevé, 406 personnes sont décédées durant cette épidémie sur 580, soit 70% de la population.

Plus généralement, ceci permet de se rendre compte que sur un scénario de 2190 unités de temps, la population mondiale pourrait être amenée à être divisée par 3 si aucune mesure n’est mise en place.

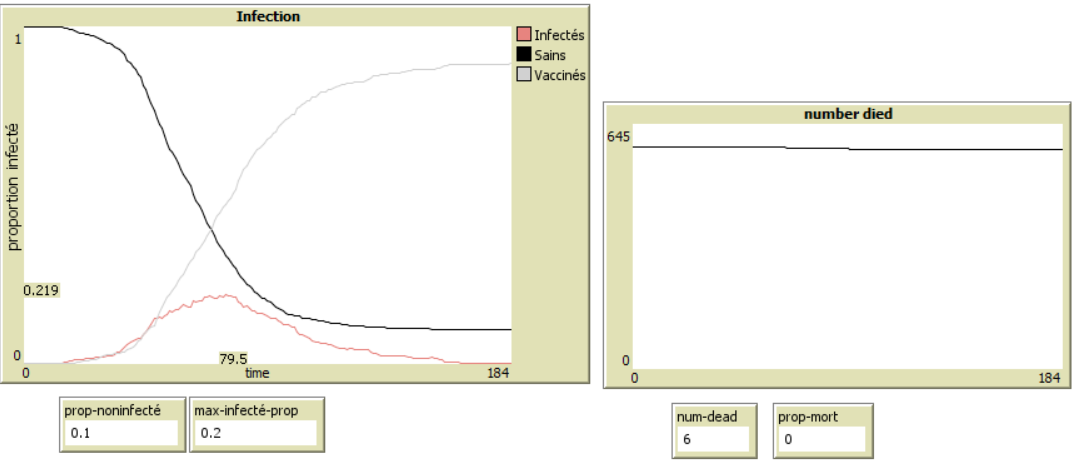
**Avec Vaccin et sans mesure :**

Ainsi si des mesures sont mises en place, comme le port de masque ou bien le confinement, mais sans vaccin, nous pouvons obtenir de meilleurs résultats comme celui-ci : 

Nous remarquons que sans vaccin mais avec des mesures mises en place la maladie se propage et ainsi évite que trop de personnes soient contaminées et risque de décéder.

**Avec Vaccin et avec mesures :**

Enfin notre dernière analyse sera avec le vaccin ainsi que les mesures mises en place pour lutter contre la maladie :



Nous pouvons conclure en analysant ces deux plots que c’est notre meilleur résultat. En effet on remarque que si la population respecte les mesures de confinement et se fait vacciner, la maladie ne peut plus se développer correctement et ainsi est amenée avec le temps à disparaitre !

A titre de comparaison :

Modèle avec Vaccination sans mesures : 6.38% de la population est décédé

Modèle sans Vaccination sans mesures : 70% de la population est décédé

Modèle sans Vaccination avec mesures : 37.07% de la population est décédé

Modèle avec Vaccination avec mesures : 1.03% de la population est décédé

# IV-Conclusion

Pour conclure ce rapport, nous avons donc réaliser une étude sur la crise du COVID 19, avec différentes approches de modélisation mathématique, afin de pouvoir simuler le cours de la maladie. Nous avons appris comment fonctionnait un modèle SIR que nous avons développé mathématiquement et appliqué à notre code. Durant ce projet nous avons pu découvrir un nouveau langage « Netlogo », qui nous à permis de modéliser notre programme.