

# État de l'art

## Modèles SIS et SIR stochastiques

23 mai 2021

---

### Élèves :

Arthur BOIVERT boivert@etud.insa-toulouse.fr  
Philippe CONESA conesa@etud.insa-toulouse.fr

### Tuteur :

M. Lorick HUANG

**Mots-clés :** SIR, modèles stochastiques, graphes, processus de Markov

### Abstract :

De nos jours, la compréhension d'une épidémie et l'anticipation de l'usage ou non de mesures restrictives est essentielle. Des modèles déterministes simples permettent de simuler la propagation d'une épidémie au cours du temps, mais omettent l'aspect aléatoire d'une épidémie. Dans le cadre de notre projet de modélisation, nous nous sommes penchés sur les modèles SIS et SIR stochastiques. A l'aide du simulateur stochastique GEMFSim, nous avons simulé la propagation d'une épidémie dans diverses situations. Nous avons analysé l'impact de différentes mesures restrictives, et, après avoir analysé le comportement du  $R_0$ , avons finalement complexifié le modèle SIR en implémentant notre propre modèle.

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Les modèles SIS et SIR</b>	<b>2</b>
1.1 Différences entre stochastique et déterminisme . . . . .	2
1.1.1 Définition des termes . . . . .	2
1.1.2 Des modèles stochastiques et des modèles déterministes . . . . .	2
1.2 Les modèles SIS et SIR . . . . .	2
1.2.1 Des modèles compartimentaux en épidémiologie . . . . .	3
1.2.2 Le modèle SIS déterministe . . . . .	3
1.2.3 Le modèle SIR déterministe . . . . .	3
1.3 Réflexion autour du $R_0$ pour les modèles SIS et SIR déterministes . . . . .	4
<b>2 GEMFSim, un simulateur stochastique de propagation d'épidémie</b>	<b>6</b>
2.1 Choix du simulateur GEMFSim . . . . .	6
2.1.1 L'histoire de GEMFSIM . . . . .	6
2.1.2 Principe de GEMFSim . . . . .	6
2.1.3 Avantages et inconvénients de GEMFSIM . . . . .	6
2.2 Concepts mathématiques sur lesquels GEMFSim est basé . . . . .	6
2.2.1 Théorie des graphes . . . . .	7
2.2.2 Processus de Markov . . . . .	7
2.2.3 Algorithme de Gillespie . . . . .	7
2.3 Théorie mathématique de GEMFSim . . . . .	8
2.3.1 Graphe et processus de Markov à valeurs discrètes . . . . .	8
2.3.2 Déterminer l'ensemble des états $X$ au cours du temps . . . . .	8
2.3.3 Élection de l'événement . . . . .	8
2.3.4 Élection du processus . . . . .	9
2.3.5 Comptage déterministe, principe de Monte-Carlo . . . . .	10
2.4 Application informatique du modèle mathématique . . . . .	10
2.4.1 Le code utilisé . . . . .	10
2.4.2 Création du graphe . . . . .	11
2.4.3 Retour sur le modèle SIR : transitions edge-based et node-based . . . . .	11

2.4.4	Déclaration de la liste Para . . . . .	11
2.4.5	Implémentation d'une version modifiée de l'algorithme de Gillespie . . . . .	12
2.4.6	Simulation de la propagation du virus . . . . .	12
2.5	Modification du code pour ajouter de nouvelles options . . . . .	13
2.5.1	Implémentation du confinement . . . . .	14
2.5.2	Implémentation du couvre-feu . . . . .	14
2.5.3	Implémentation du $R_0$ . . . . .	14
<b>3</b>	<b>Simulations</b>	<b>16</b>
3.1	À propos du modèle SIS . . . . .	16
3.2	Influence des paramètres du modèle . . . . .	16
3.2.1	Taux d'infection . . . . .	16
3.2.2	Taux de rémission . . . . .	17
3.2.3	Taille de la population . . . . .	18
3.2.4	Pourcentage de population initialement infectée . . . . .	20
3.3	Mise en place des mesures restrictives . . . . .	20
3.3.1	Paramètres de simulation . . . . .	20
3.3.2	Graphes des situations . . . . .	21
3.3.3	Seuils de mise en place des mesures restrictives . . . . .	22
3.4	Impact des mesures restrictives . . . . .	22
3.4.1	Méthode d'analyse . . . . .	23
3.4.2	L'épidémie initiale . . . . .	23
3.4.3	Stratégie de confinement long et strict . . . . .	23
3.4.4	Stratégie de Stop&Go . . . . .	24
3.4.5	Stratégie de couvre-feu . . . . .	26
3.4.6	Stratégie de confinement léger et de couvre-feu simultanés . . . . .	26
3.4.7	Conclusion sur les mesures restrictives . . . . .	27
3.4.8	À propos du $R_0$ . . . . .	27
3.5	Implémentation du modèle déterministe . . . . .	28
3.5.1	Comparaison avec notre modèle stochastique . . . . .	28
3.5.2	Correction du modèle stochastique . . . . .	29
3.6	Implémentation de notre propre modèle . . . . .	29

3.6.1	Limites du modèle SIR . . . . .	30
3.6.2	Notre modèle . . . . .	30
3.6.3	Quelques résultats obtenus . . . . .	32
3.6.4	Améliorations possibles . . . . .	34
<b>Conclusion</b>		<b>36</b>
<b>Références</b>		<b>37</b>
<b>4</b>	<b>Annexes</b>	<b>A1</b>
4.1	Minimum d'un échantillon de loi exponentielles indépendantes - 2.3.3 . . . . .	A1
4.2	Détermination de $\Delta t$ - 2.4.5 . . . . .	A1
4.3	Compléments sur les modifications apportées au code . . . . .	A2
4.3.1	Implémentation du confinement - 2.5.1 . . . . .	A2
4.3.2	Implémentation du couvre-feu - 2.5.2 . . . . .	A2
4.3.3	Implémentation du $R_0$ - 2.5.3 . . . . .	A3
4.3.4	Implémentation du modèle SIRCM - 3.6.2 . . . . .	A4
4.4	Implémentation du modèle déterministe - 3.5 . . . . .	A4
4.4.1	Réflexion mathématique . . . . .	A4
4.4.2	Implémentation informatique . . . . .	A5

## Introduction

Aujourd'hui, les modèles épidémiques sont plus que jamais d'actualité. Tous les jours, nous pouvons lire dans les journaux ou dans les articles scientifiques, ou encore voir à la télévision de nouvelles prévisions et modélisations quant à l'évolution de la pandémie de covid-19. Nous vivons une pandémie qui a renvoyé au premier plan la nécessité de prévoir et de modéliser la circulation d'un virus.

Actuellement, la prédiction de l'évolution de cette pandémie est indispensable. On a pu ainsi voir l'importance de l'anticipation de l'augmentation de personnes infectées pour préserver la société. Une mauvaise anticipation et donc une mauvaise prévision de la circulation d'un virus peut amener à la prise de mesures de protections restrictives telles que le confinement, le couvre-feu...

Les modèles épidémiques apparaissent donc comme centraux dans une volonté de contrôler une épidémie, contrôler sa propagation pour la juguler. Face à l'importance qu'a pris dans nos vies le coronavirus, nous avons voulu comprendre comment fonctionnaient ces modèles. Pour cela, la simplification permise par les modèles compartimentaux SIS et SIR rend intéressante et abordable l'analyse de la propagation d'une épidémie au sein d'une population au cours du temps. Devant la profusion des travaux réalisés sur les modèles SIS et SIR, nous avons décidé de nous concentrer sur les modèles stochastiques plutôt que déterministes. En effet, prendre en compte l'aspect aléatoire d'une épidémie nous a semblé beaucoup plus proche du réel et plus intéressant à modéliser.

En travaillant sur ce sujet passionnant, nous avons tenté de répondre aux interrogations qui ont surgi au fur et à mesure de notre étude.

Comment ces modèles épidémiologiques sont-ils définis ? Quels paramètres sont utilisés dans ces modèles afin de donner des estimations très précises et poussées de la propagation du virus ? Comment peut-on simuler la propagation d'une épidémie de manière stochastique ? Quel est l'impact des mesures restrictives sur la propagation d'un virus ? Et finalement, pouvons-nous vérifier la pertinence du modèle stochastique en comparaison avec le modèle déterministe ?

Pour répondre à ces questions, tout d'abord nous définirons les modèles SIS et SIR et présenterons les différences entre l'approche stochastique et déterministe. Dans un second temps, il s'agira de présenter le simulateur stochastique de la propagation d'épidémie que nous avons utilisé. Nous vous expliquerons la théorie mathématique associée à ce simulateur ainsi que l'application informatique que nous pouvons en faire. Enfin, nous évoquerons les simulations réalisées pour comprendre l'influence des paramètres du modèle SIR stochastique et voir l'importance de mesures restrictives sur la circulation d'un virus au sein d'une population.

# 1 Les modèles SIS et SIR

Dans cette partie, nous définirons dans un premier temps les différences entre un modèle stochastique et un modèle déterministe. Ensuite, nous introduirons le concept de modèle compartimental, afin de présenter les modèles SIS et SIR déterministes, et finalement nous tenterons de déterminer le nombre de reproduction de base, ou  $R_0$ , de ces deux modèles déterministes.

## 1.1 Différences entre stochastique et déterminisme

Dans le cadre de nos recherches concernant les modèles SIS et SIR, une première interrogation nous est apparue : quelle est la différence entre un modèle déterministe et stochastique ? Quelles sont leurs caractéristiques ? Ont-ils des avantages ?

### 1.1.1 Définition des termes

Tout d'abord, commençons par bien définir les termes qui nous intéressent.

D'après le dictionnaire *Larousse*, le terme stochastique se réfère à des

« phénomènes qui, partiellement, relèvent du hasard et qui font l'objet d'une analyse statistique. »

Quant au mot déterminisme, le *Larousse* le définit comme suit :

« Enchaînement de cause à effet entre deux ou plusieurs phénomènes. »

### 1.1.2 Des modèles stochastiques et des modèles déterministes

Maintenant que nous avons défini les termes, nous allons tenter d'expliquer les différences entre un modèle stochastique et un modèle déterministe.

Un modèle déterministe est un système qui obéit à des lois d'évolution non-probabilistes. Il est généralement défini par un système d'équations qui décrira l'évolution de ce modèle au cours du temps. On part du principe que l'on sait ce qu'il se passe dans le système de façon déterministe, et qu'aucune évolution probabiliste n'est possible.

Un modèle stochastique représente l'évolution, en temps discret ou continu, d'une variable aléatoire. On va alors affecter des probabilités à des événements se produisant dans une période de temps donnée : par exemple la probabilité qu'une personne soit infectée par un virus sur un temps  $t$  donné.

Le modèle déterministe nous permet donc de faire une modélisation générale : cela nous donne une idée globale de l'évolution de notre modèle mathématique. Cependant il reste très dépendant du modèle mathématique choisi. D'un autre côté, le modèle stochastique semble plus polyvalent, car il permet de modéliser notre système d'une manière plus fidèle à la réalité.

Si on prend le cas des modèles décrivant les épidémies, un modèle stochastique semble très intéressant car il permet de corriger les approximations faites par le modèle déterministe. En effet, une épidémie comporte une grande part d'aléatoire dans sa propagation. Ainsi, un modèle déterministe pourra donner une estimation de l'évolution de l'épidémie mais ne prendra pas en compte l'aspect aléatoire de la propagation d'un virus.

## 1.2 Les modèles SIS et SIR

Maintenant que nous avons défini et vu plus en détail les modèles stochastiques et déterministes, il convient de présenter les modèles SIS et SIR décrivant l'évolution d'une épidémie au sein d'une population.

### 1.2.1 Des modèles compartimentaux en épidémiologie

Les modèles SIS et SIR sont des modèles compartimentaux. Ce genre de modèle se base sur deux concepts :

- les compartiments, autrement dit les différents états possibles pour la population par rapport à la maladie
- les règles, qui spécifient la proportion des individus passant d'un compartiment à un autre.

En épidémiologie, plusieurs compartiments sont généralement utilisés :

- $S$ , pour les personnes susceptibles
- $I$ , pour les personnes infectées
- $R$ , pour les personnes immunisées
- $D$ , pour les personnes décédées
- d'autres états selon la maladie modélisée,  $Q$  pour Quarantine,  $A$  pour Alerted (équivalent du cas-contact),  $C$  pour Critical,...

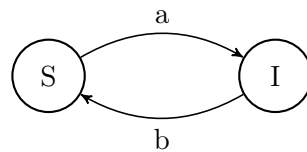
Ces modèles compartimentaux, dans le cas déterministe, sont régis par des lois, principalement des équations différentielles, qui modélisent au cours du temps l'évolution de la proportion de population dans chacun des états.

### 1.2.2 Le modèle SIS déterministe

Le modèle SIS (*Susceptible – Infected – Susceptible*) comporte deux états pour les individus :

- *Susceptible*, à savoir un individu sain pouvant être contaminé
- *Infected*, à savoir un individu infecté

Dans ce modèle, après avoir été contaminé, on devient à nouveau Susceptible : on considère que l'individu n'a pas développé d'immunité.



Ici, sur le schéma, on peut observer que les personnes susceptibles ont un taux  $a$  de devenir infectées. D'un autre côté, les personnes infectées ont un taux  $b$  de redevenir susceptibles. On retrouve ces paramètres dans le modèle mathématique que l'on peut voir ci-dessous.

$$\begin{aligned}\frac{dS}{dt} &= bI - aSI \\ \frac{dI}{dt} &= aSI - bI\end{aligned}$$

Dans la deuxième équation, intéressons-nous à comment a été définie la possibilité d'être infectée. Elle dépend du contact entre les individus susceptibles  $S$  et les individus infectés  $I$  (terme  $aSI$ ) et dépend bien sûr du nombre de personnes infectées devenues saines entre temps  $-bI$ .

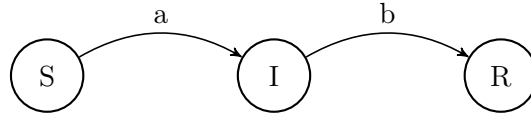
La population  $S$  est quant à elle définie par le nombre de personnes guéries redevenues susceptibles (terme  $bI$ ). On enlève également les personnes en contact avec des individus infectés et ayant été contaminés (terme  $-aSI$ ).

### 1.2.3 Le modèle SIR déterministe

Le modèle SIR a été inventé en 1927 par 2 chercheurs : Kermack et McKendrick. Ils ont réutilisé le travail du chercheur britannique Ronald Ross, qui a reçu le Prix Nobel de Médecine pour ses

travaux sur le paludisme. A l'époque, ces scientifiques ont tenté d'expliquer pourquoi la pandémie de grippe espagnole de 1918 n'avait pas infecté toute la population.

Le modèle SIR (*Susceptible - Infected - Recovered*) comporte donc trois états. Dans ce modèle-ci, après avoir été infecté, on considère que la personne a développé une immunité et elle n'est donc plus considérée comme susceptible.



Le schéma précédent s'interprète comme celui du modèle SIS.

$$\begin{aligned}\frac{dS}{dt} &= -aSI \\ \frac{dI}{dt} &= aSI - bI \\ \frac{dR}{dt} &= bI\end{aligned}$$

L'équation pour la population infectée ne change pas.

Pour la population susceptible, on retrouve le terme  $bI$  qui correspond au nombre de personnes infectées guéries, puisqu'on considère à présent que les personnes infectées guéries sont immunisées. La population susceptible ne fera donc que baisser tout au long de l'épidémie, comme on peut le voir avec le terme  $-aSI$ . Cela exprime le fait que la population saine diminue proportionnellement au nombre de personnes susceptibles et au nombre de personnes infectées.

Enfin la population  $R$  (guérie et immunisée) est définie par le nombre de personnes infectées guéries (terme  $bI$ ).

### 1.3 Réflexion autour du $R_0$ pour les modèles SIS et SIR déterministes

Nous allons ici tenter de définir le  $R_0$ , le taux de reproduction de base. Pour cela, prenons par exemple le modèle SIR déterministe.

$$\begin{aligned}\frac{dS}{dt} &= -aSI \\ \frac{dI}{dt} &= aSI - bI \\ \frac{dR}{dt} &= bI\end{aligned}$$

Quand une population est victime d'une épidémie, elle met tout en oeuvre pour tenter de l'éliminer. Le but est donc de réduire le plus possible le nombre d'infectés. Au niveau mathématique, cela se traduit par la nécessité d'avoir un  $I(t)$  décroissant, et donc que la quantité  $\frac{dI}{dt}$  soit négative. Essayons d'atteindre cet objectif.

$$\begin{aligned}\forall t > 0, \frac{dI}{dt} &\leq 0 \\ \implies aSI &\leq bI \\ \implies \frac{aS}{b} &\leq 1\end{aligned}$$



Dès lors, nous pouvons pour le modèle SIR déterministe définir le  $R_0$  comme suit

$$R_0(t) = \frac{aS(t)}{b}$$

Les équations déterminant le nombre d'infectés au cours du temps étant les mêmes pour les modèles SIS et SIR déterministes, on peut donc affirmer que le  $R_0$  est défini de la même manière pour le modèle SIS déterministe.

## 2 GEMFSim, un simulateur stochastique de propagation d'épidémie

Afin de simuler la propagation d'un virus de manière stochastique, nous avons décidé d'utiliser le simulateur GEMFSim. Tout au long de cette partie, après avoir justifié notre choix d'utiliser ce modèle, nous expliquerons la théorie mathématique sur laquelle GEMFSim est basée, nous présenterons rapidement comment l'application informatique du modèle mathématique est implémentée, et nous parlerons de quelques modifications que nous avons apporté au simulateur afin de le compléter.

### 2.1 Choix du simulateur GEMFSim

Plutôt que d'écrire nous-même notre code afin de simuler de manière stochastique la propagation d'une épidémie, nous avons décidé d'utiliser un simulateur déjà existant. Après l'avoir présenté rapidement, nous expliquerons pourquoi avoir fait ce choix.

#### 2.1.1 L'histoire de GEMFSIM

Ce modèle est à l'initiative de Faryad Darabi Sahneh, professeur assistant de recherche à la Kansas State University. Après avoir rédigé un article[1] expliquant les fondements du modèle, il a tout d'abord, avec l'aide de Caterina Scoglio, professeure en Génie Informatique à la même université, implémenté son modèle sous Matlab. Par la suite, 4 autres de ses collaborateurs ont implémenté son modèle dans d'autres langages de programmation, dont Heman Shakeri, professeur assistant, qui a écrit la version Python, que nous utiliserons.

#### 2.1.2 Principe de GEMFSim

Ce modèle a attiré notre attention notamment car il permet de modéliser de manière stochastique la propagation d'une épidémie suivant les modèles SIS et SIR au travers d'un graphe.

Le principe est très simple. On définit un graphe  $\mathcal{G}$  et des taux d'infection et de rémission. Ensuite, le simulateur se base sur un algorithme stochastique qui permet, à partir d'une condition initiale, de faire se propager l'épidémie dans le graphe  $\mathcal{G}$ . Après simulation, nous obtenons des courbes, qui représentent la proportion de population (i.e. le pourcentage de noeuds de  $\mathcal{G}$ ) dans chacun des états au cours du temps. Finalement, le simulateur permet aussi de sauvegarder une animation GIF représentant l'évolution de l'épidémie dans le graphe  $\mathcal{G}$ .

#### 2.1.3 Avantages et inconvénients de GEMFSIM

GEMFSim présente de nombreux avantages. Tout d'abord, il permet une simulation très visuelle de la propagation d'une épidémie, grâce notamment à l'utilisation d'un graphe et le rendu d'une animation visuelle. Il est aussi extrêmement simple d'utilisation, il suffit en effet de définir les différents taux d'infection et de rémission et la taille de notre population. Il s'avère aussi très facilement modifiable et permet d'ajouter son propre modèle, comme nous le verrons plus tard. Finalement, il est très complet, et permet de faire la propagation d'une épidémie dans plusieurs graphes simultanément (aspect *multi layers*)<sup>1</sup>.

L'un des principaux inconvénients reste le temps de simulation, qui peut s'avérer très long dans le cadre d'une population plutôt grande (le problème se ressent dès 1000 noeuds).

## 2.2 Concepts mathématiques sur lesquels GEMFSim est basé

Le simulateur GEMFSim est une implémentation d'une adaptation de l'algorithme de Gillespie sur un graphe. Cependant, avant de décrire la mathématique du code, il nous semblait important

---

1. Nous avons cependant décidé de propager notre épidémie sur un seul graphe (1 *layer*) car c'est notamment plus simple pour la compréhension et l'analyse.

de faire quelques rappels sur les principaux concepts mathématiques utilisés.

### 2.2.1 Théorie des graphes

La théorie des graphes[2] est une discipline mathématique et informatique qui étudie des graphes, représentant généralement des réseaux reliant des objets. De nombreux types de graphes existent, les graphes orientés par exemple, mais nous étudions dans notre cas les graphes les plus simples, que nous allons définir comme suit.

Un graphe est un couple  $\mathcal{G}(V, E)$  où

- $V$  est un ensemble de sommets
- $E$  est un ensemble d'arêtes, ou liens, qui sont des paires de sommets : un lien est associé à deux sommets distincts

Par exemple, considérons le graphe  $\mathcal{G}$  suivant.

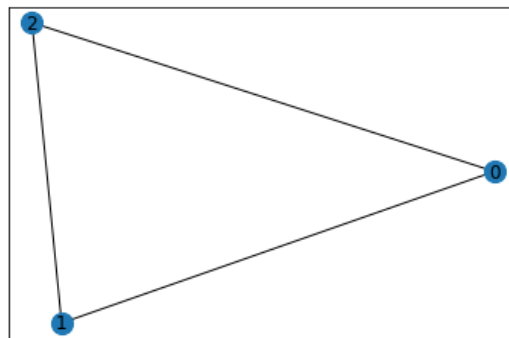


FIGURE 1 – Graphe  $\mathcal{G}_1$  généré à l'aide de la librairie *networkx* sur Python

Ainsi, nous avons un graphe  $\mathcal{G}_1(V_1, E_1)$  tel que

$$V_1 = \{0, 1, 2\}$$

$$E_1 = \{\{0, 1\}, \{0, 2\}, \{1, 2\}\}$$

### 2.2.2 Processus de Markov

Un processus de Markov est un processus stochastique qui possède la propriété de Markov. En effet, dans ce genre de processus, la prédiction de l'état futur prend seulement en compte l'état présent, et cette prédiction n'est pas rendue plus précise par les états du passé.

Un processus de Markov est représenté par une séquence  $(X_1, \dots, X_N)$  de variables aléatoires telles que  $X_t$  représente l'état de notre système au temps  $t$ . Dans un tel cas, la propriété de Markov s'écrit comme suit :

$$\mathcal{P}(X_{n+1} = x | X_1, X_2, \dots, X_n) = \mathcal{P}(X_{n+1} = x | X_n)$$

avec  $x$  un état quelconque du système.

### 2.2.3 Algorithme de Gillespie

L'algorithme de Gillespie a été créé par Joseph L. Doob aux alentours des années 1945[3, 4], mais popularisé par Dan Gillespie en 1977, dans un article[5, 6] où il l'utilise pour simuler des systèmes de réactions chimiques et biochimiques de manière efficace et précise. De manière générale, cet algorithme génère une trajectoire statistiquement correcte d'un système d'équations stochastiques pour lequel les taux de transition sont connus.

L'algorithme de Gillespie s'articule en 11 étapes :

1. initialisation du temps à  $t = 0$

2. choix d'un état initial pour le système
3. lister les taux de transition  $r_{k,i}$  des  $N_k$  transitions possibles d'un état  $k$  à un état  $i$
4. calcul de  $R_{k,i} = \sum_{j=1}^i r_{k,j}$  pour  $i \in \llbracket 1, N_k \rrbracket$
5. calcul du taux total  $Q_k = R_{k,N_k}$
6. tirage aléatoire d'un nombre  $u$  compris entre 0 et 1
7. recherche de l'événement  $i$  tel que  $R_{k,i-1} < uQ_k \leq R_{k,i}$
8. réalisation de l'événement  $i$  : on passe de l'état  $k$  à l'état  $i$
9. tirage aléatoire d'un nombre  $u'$  compris entre 0 et 1
10. mise à jour du temps  $t$  au temps  $t + \Delta t$ , avec  $\Delta t = \frac{-\ln(u')}{Q_k}$
11. retour à l'étape 3, ou fin de l'algorithme selon des conditions exprimées

## 2.3 Théorie mathématique de GEMFSim

Après avoir rappelé quelques concepts mathématiques primordiaux à la compréhension de GEMFSim, nous allons désormais nous atteler à expliquer précisément le fonctionnement du modèle GEMFSim[1, 7].

### 2.3.1 Graphe et processus de Markov à valeurs discrètes

Dans un premier temps, nous établissons un graphe  $\mathcal{G}(V, E)$  où  $V$  représente un ensemble constitué de  $N$  sommets, et  $E$  l'ensemble des arêtes du graphe  $\mathcal{G}$ .

À un instant  $t$ , le graphe  $\mathcal{G}$  est représenté par son état  $X_t$  et tel que

$$X_t = [x_1(t), x_2(t), \dots, x_N(t)]$$

avec,  $\forall i \in \llbracket 1, N \rrbracket$ ,  $x_i(t)$  représentant l'état du noeud  $i$  à l'instant  $t$ .

Ainsi, il apparaît que  $\mathcal{G}$  est un processus de Markov à valeurs discrètes dans un ensemble  $S$  de taille  $M^N$  (avec  $M$  le nombre d'états différents que peut prendre un noeud<sup>2</sup>) le nombre d'états différents possibles pour  $\mathcal{G}$ .

### 2.3.2 Déterminer l'ensemble des états $X$ au cours du temps

Désormais, nous souhaitons faire circuler notre épidémie dans  $\mathcal{G}$ , autrement dit déterminer  $(X_t)_{0 \leq t \leq t_{max}}$ , la suite représentant l'ensemble des états que  $\mathcal{G}$  prendra sur l'intervalle de temps  $[0, t_{max}]$ .

Cette nouvelle étape consiste finalement à faire évoluer notre processus de Markov. Ceci s'effectue en deux étapes :

- l'élection de l'événement (i.e. l'instant auquel aura lieu le changement d'état)
- l'élection du processus (i.e. le noeud qui va changer d'état)

### 2.3.3 Élection de l'événement

Ensuite, il convient de définir l'instant  $\Delta t$  auquel le graphe va changer d'état.

Considérons un noeud  $i$  dans le cadre du modèle SIR. Ce qui est énoncé est aussi valable pour le modèle SIS, en adaptant les transitions possibles. À un instant  $t + \Delta t$ , ce noeud  $i$  a quatre possibilités :

- passer à un état  $I$  s'il est lié à des noeuds infectés
- passer dans un état  $R$  s'il est infecté
- rester dans un état  $R$  s'il a déjà été infecté

---

2. soit 2 pour le modèle SIS (Susceptible ou Infected) et 3 pour le modèle SIR (Susceptible, Infected ou Recovered)

— rester dans un état  $S$  s'il n'est pas lié à des noeuds infectés

Les deux derniers cas ne présentent pas d'intérêt. Nous considérons donc deux transitions possibles :

- $m = S \longrightarrow n = I$
- $m = I \longrightarrow n = R$

Nous notons que deux processus (ou transitions) ne peuvent pas avoir lieu simultanément, et qu'ils se déroulent aux taux de transition liés à la transition en question.

Ainsi,

$$\mathcal{P}(x_i(t + \Delta t) = n | x_i(t) = m) = r_i \Delta t$$

avec  $r_i$  tel que

- pour une transition  $S \longrightarrow I$ ,  $r_i = n_{vi} \times \beta$  avec  $n_{vi}$  le nombre de voisins infectés liés au noeud  $i$  à l'instant  $t$
- pour une transition  $I \longrightarrow R$ ,  $r_i = \delta$

Considérons désormais l'ensemble du graphe  $\mathcal{G}$ .  $\mathcal{G}$  étant soumis à un processus de Markov à temps continu, seulement un noeud change d'état à un instant  $t + \Delta t$ .

Ainsi, toutes les transitions que peuvent effectuer chacun des noeuds (1 seule transition possible par noeud) sont en compétition, et ces transitions sont mutuellement indépendantes.

Aussi, la transition induite par le noeud  $i$  a un temps de transition  $T_i$  tel que

$$T_i \sim \mathcal{E}(r_i)$$

Supposons  $k$  noeuds ( $k \leq N$ ) pouvant changer d'état à l'instant  $t$ . Le noeud  $i$  qui changera d'état à l'instant  $t + \Delta t$  sera celui qui a son temps de transition  $T_i$  le plus petit.

On a donc

$$T_{t+\Delta t} = \min\{T_1, T_2, \dots, T_k\}$$

Dès lors, tous ces processus étant mutuellement indépendants et en compétition, et sachant que le minimum d'un échantillon de variables aléatoires mutuellement indépendantes de distribution exponentielle suit une distribution exponentielle de paramètre la somme des paramètres de distribution de chacune des variables aléatoires ([démonstration en annexe](#)),

$$T_{t+\Delta t} \sim \mathcal{E}(R)$$

avec  $R = \sum_{j=1}^N r_j$

### 2.3.4 Élection du processus

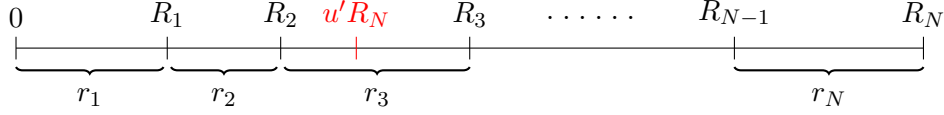
Après avoir déterminé le temps auquel le processus, autrement dit le changement d'état de notre graphe, va avoir lieu, il faut désormais déterminer quel noeud va changer d'état. Cela correspond aux [étapes 3 à 7](#) de l'algorithme de Gillespie.

Dans un premier temps, on liste pour chacun des noeuds son taux de transition  $r_i$  associé. On calcule ensuite, comme décrit dans l'étape 4, le  $R_i$  de chaque noeud, tel que  $R_i = \sum_{j=1}^i r_j$ . On tire un nombre  $u'$  aléatoire, tel que  $u' \in [0, 1]$ , et on cherche  $R_a$  tel que  $u' \times R_N \in [R_{a-1}, R_a]$ . Finalement, le noeud  $a$  sera celui qui changera d'état.

On peut assez visuellement représenter cette méthode. Elle consiste à représenter le segment des  $R_i$  comme suit, découpé en  $N$  morceaux de taille  $r_i$ <sup>3</sup>.

---

3. En réalité, il n'est pas rare que le  $r$  de certains noeuds soit nul. Il serait plus juste de définir ce segment comme un segment découpé en  $k$  morceaux (les  $k$  noeuds pouvant changer d'état à l'instant  $t$ ), où sont représentés seulement les  $R_i$  tels que  $R_i \neq 0$ .



Dans l'exemple précédent, le noeud 3 est choisi pour changer d'état.

### 2.3.5 Comptage déterministe, principe de Monte-Carlo

Maintenant, expliquons le processus de comptage déterministe. Pour cela, il faut recontextualiser. Une simulation renvoyée par l'algorithme de Gillespie nous retourne 3 grandeurs,  $S_t, I_t$  et  $R_t$ , où  $t < t_{max}$ . Cependant, l'intérêt d'un processus stochastique est de répéter un grand nombre de fois notre simulation, et d'en réaliser la moyenne, afin d'obtenir des résultats s'approchant le plus possible de la réalité.

Pour cela, nous appliquons le principe de Monte-Carlo. Ce principe est directement basé sur la Loi des Grands Nombres, et consiste en la répétition d'expériences aléatoires, dans l'objectif d'évaluer une quantité, et d'approcher la résolution d'un système déterministe[8].

La loi (faible) des grands nombres nous permet d'affirmer que, pour une suite de variables  $(X_i)$  indépendantes et de même loi admettant une espérance,

$$\overline{X}_n = \frac{\sum_{i=1}^n X_i}{n} \xrightarrow{\mathcal{P}} \mathbb{E}(X)$$

Notre objectif est de déterminer  $t \longrightarrow s_t$ <sup>4</sup>, fonction déterministe qu'il nous est impossible de trouver de façon analytique.

Une simulation de l'algorithme de Gillespie nous renvoie  $t \longrightarrow S_t$ , une fonction du temps aléatoire, et telle que  $\mathbb{E}(S_t) = s_t$ .

Ainsi, le principe de Monte-Carlo nous permet de dire que

$$\overline{S}_t = \frac{\sum_{i=1}^{n_{sim}} S_t^i}{n_{sim}} \implies \overline{S}_t \xrightarrow{n_{sim} \rightarrow +\infty} s_t$$

En partant de cela, nous effectuons simplement un très grand nombre de fois une répétition de l'algorithme de Gillespie, et en moyennisant les suites  $(S_t^i)$ ,  $(I_t^i)$  et  $(R_t^i)$ <sup>5</sup>, nous obtenons des résultats qui permettent d'approcher les résultats renvoyées par le modèle déterministe.

## 2.4 Application informatique du modèle mathématique

Maintenant que nous savons comment fonctionne mathématiquement notre modèle, nous allons succinctement expliquer son fonctionnement au niveau informatique.

### 2.4.1 Le code utilisé

Nous avons décidé d'utiliser le langage Python pour travailler. Cependant, le simulateur est disponible dans quelques autres langages usuels, tels que R, Matlab, ou encore C. La version originale du code est disponible [ici](#), cependant nous avons modifié un peu cette version afin d'implémenter de nouvelles fonctionnalités. La version que nous avons utilisée est disponible sur ce [GitHub](#).

4. On doit faire de même pour les individus  $I$  et  $R$ , nous expliquons le processus seulement pour  $S$ . C'est exactement le même pour les deux autres cas.

5. Toutes nos expériences sont indépendantes, ce qui permet d'assurer l'indépendance de nos variables  $S, I$  et  $R$ .

### 2.4.2 Création du graphe

Afin de créer le graphe  $G$  sur lequel la propagation du virus se fera, nous utilisons la librairie *networkx*. Cette librairie permet notamment de créer un graphe aléatoire, mais aussi un graphe défini à partir d'une liste *pos*, qui contient les positions de chacun des  $N$  noeuds du graphe. L'argument *radius* permet de définir la distance maximale entre deux noeuds pour que ceux-ci soient liés.

```
1 G = networkx.random_geometric_graph(N, radius)
2 G = networkx.random_geometric_graph(N, radius, pos=pos)
```

Algorithme 1 – Déclaration du graphe  $G$  (aléatoire ou avec positions données)

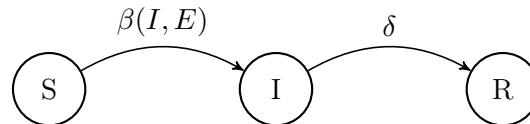
Nous récupérerons aussi des informations sur le graphe dans la variable *Net*, notamment sur les liens entre les noeuds, à l'aide des routines *MyNet* et *NetCmbn*, implémentées par GEMFSim.

```
1 Net = NetCmbn([MyNet(G)])
```

Algorithme 2 – Récupération des informations sur le graphe  $G$

### 2.4.3 Retour sur le modèle SIR : transitions edge-based et node-based

Reprenons le diagramme de transition du modèle SIR précédemment introduit.



Nous remarquons que la transition  $S \rightarrow I$  est un peu particulière : elle dépend en effet de  $(I, E)$ . Ceci signifie que la transition d'un état  $S$  à  $I$  pour un noeud  $i$  dépend du paramètre  $\beta$  ainsi que du voisinage du noeud  $i$  : cette transition est uniquement possible lorsqu'un ou plusieurs noeuds liés au noeud  $i$  sont dans un état Infecté  $I$ , état que nous appelons état-influteur. Nous qualifions ces transitions de edge-based, des transitions basées sur les liens entre les noeuds.

Elles sont à différencier des transitions node-based, la transition  $I \rightarrow R$ , qui ne dépend pas du voisinage du noeud  $i$ , seulement du paramètre  $\delta$ .

Les différences entre ces deux types de transitions se font à l'aide d'une matrice de transition edge-based, appelée  $A_\beta$  dans notre simulateur, et d'une matrice de transition node-based, notée  $A_\delta$ .

Notons que ces deux matrices sont pareillement définies pour le modèle SIS, la transition  $S \rightarrow I$  étant edge-based et la transition  $I \rightarrow S$  node-based.

### 2.4.4 Déclaration de la liste Para

Afin de simuler la propagation du virus, GEMFSim a besoin de la liste *Para*, définie à l'aide de la méthode *Para\_xxx*<sup>6</sup>. Cette liste permet notamment de définir le nombre d'états pour les noeuds, l'état-influteur (qui sera l'état  $I$  que ce soit pour le modèle SIS ou le modèle SIR), ainsi que les deux matrices  $A_\beta$  et  $A_\delta$  introduites précédemment.

```
1 Para = Para_SIR(delta, beta)
```

Algorithme 3 – Déclaration de la liste *Para* pour le modèle SIR

Dans le cadre du modèle SIR, la liste *Para* sera comme suit :

- $M = 3$  (nombre d'états possibles pour chaque noeud)
- $q = 1$  ( $q$  représente l'état-influteur, dans notre cas l'état  $I$ , qui est codé par 1 dans le simulateur)

---

6. *xxx* représente le modèle utilisé, SIS ou SIR dans notre cas

— les deux matrices  $Ab$  (transitions edge-based) et  $Ad$  (transitions node-based) telles que

$$Ab = \begin{pmatrix} 0 & \beta & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad Ad = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \delta \\ 0 & 0 & 0 \end{pmatrix}$$

#### 2.4.5 Implémentation d’une version modifiée de l’algorithme de Gillespie

Maintenant, il convient de décrire l’algorithme de simulation d’un processus de Markov du simulateur GEMFSim, qui permet de propager l’épidémie au sein de notre graphe.

Ce dernier implémente une adaptation de l’algorithme de Gillespie afin de simuler la propagation d’une épidémie au travers d’un graphe  $\mathcal{G}$ . Nous allons désormais expliquer chacune des étapes de l’algorithme.

1. initialisation du temps à  $t = 0$
2. choix d’un état initial pour  $\mathcal{G}$
3. modification de la matrice  $Nq$  qui contient pour chacun des noeuds le nombre de noeuds infectés auquel il est lié (*initialisation si première boucle*)
4. modification de la matrice  $Rn$  qui contient pour chacun des noeuds son taux de transition d’état  $r_i$  ( $r_i$  défini précédemment) (*initialisation si première boucle*)
5. modification de  $R$  ( $R$  défini précédemment) (*initialisation si première boucle*)
6. choix de  $\Delta t$  tel que  $\Delta t = \frac{-\ln(u)}{R}$  avec  $u$  compris entre 0 et 1 et tiré aléatoirement<sup>7</sup>
7. élection du noeud qui va changer d’état, de la manière précédemment définie
8. changement d’état du noeud choisi précédemment (transition  $S \rightarrow I$  ou  $I \rightarrow R$ )
9. mise à jour de  $T_f$ , tel que  $T_f = T_f + \Delta t$
10. retour à l’étape 3, ou arrêt de l’algorithme si  $T_f \geq t_{max}$

#### 2.4.6 Simulation de la propagation du virus

Finalement, nous pouvons effectuer la simulation de la propagation du virus. Nous avons tout d’abord besoin de définir  $t_{max}$ , contenu dans la variable *StopCond*, afin de déterminer la durée de la simulation. Ensuite, à partir de la ligne de code suivante, la simulation se passe en 3 étapes distinctes.

```
1 t_int,f = MonteCarlo(G, Net, Para, StopCond, inf_init, 3, step, nsim, N, x_init =
    np.zeros(N))
```

Algorithme 4 – Simulation de la propagation du virus

La routine *MonteCarlo* fait implicitement appel à deux autres fonctions essentielles *GEMF\_Sim* et *StateCount*.

De plus, elle nécessite quelques arguments :

- $G$  le graphe sur lequel on souhaite propager l’épidémie
- *inf\_init* le nombre d’infectés initiaux
- *nsim* le nombre de répétitions de l’algorithme de Gillespie
- $N$  la taille de notre population (nombre de noeuds du graphe)

---

<sup>7</sup>  $\Delta t$  est déterminé à l’aide de la manière précédemment définie et d’une inversion de la loi exponentielle, plus de détails donnés en annexe.



**Étape 1** Simulation de l'algorithme modifié de Gillespie sur le graphe  $G$  par la routine *GEMF\_Sim* : nous propageons notre virus dans  $G$  selon la version modifiée de l'algorithme de Gillespie, [définie précédemment](#).

Une simulation de l'algorithme de Gillespie nous renvoie des courbes de ce type-là. Les nombreuses oscillations sont dues au caractère aléatoire de l'algorithme.

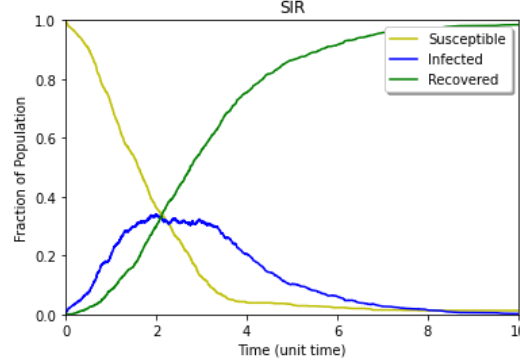


FIGURE 2 – Simulation de l'algorithme de Gillespie ( $\beta = 2, \delta = 0.8$ )

**Étape 2** Comptage des noeuds dans chacun des états au cours du temps par la routine *StateCount*

La routine *StateCount* permet de réaliser pour chacun des instants le comptage de noeuds dans chacun des états.

**Étape 3** Comptage déterministe des  $n_{sim}$  simulations à l'aide de la routine *MonteCarlo* et de la manière [définie précédemment](#)

La routine *MonteCarlo* nous renvoie des courbes de ce type-là. Ce sont en fait les moyennes des  $n_{sim}$  courbes retournées dans l'étape 1, les bruits ayant été "supprimés" du fait de la moyennisation et du grand nombre de répétitions de l'expérience.

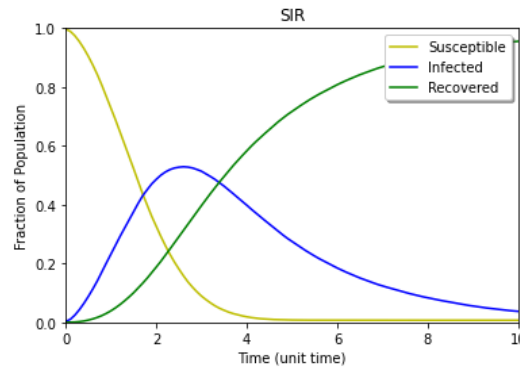


FIGURE 3 – Moyennisation de 100 répétitions de l'algorithme de Gillespie ( $\beta = 2, \delta = 0.8$ )

## 2.5 Modification du code pour ajouter de nouvelles options

De base très complet, nous avons tout de même ajouté quelques options au simulateur. En effet, nous voulons par exemple vérifier l'impact de mesures restrictives, telles que le confinement ou encore le couvre-feu. Aussi, il est intéressant d'analyser la courbe du  $R_0$  par rapport à celle des états.

Nous avons donc modifié la version originale du code pour implémenter nos nouveaux besoins. Nous détaillerons succinctement dans cette partie comment nous avons modifié le simulateur, et nous donnerons plus de détail, notamment au niveau du code, [en annexe](#).

### 2.5.1 Implémentation du confinement

Le principe du confinement est très simple, il consiste à contraindre une partie de la population à rester dans un lieu spécifique. Dans le cadre d'une épidémie, le confinement permet notamment, en obligeant la population à rester chez elle, de freiner sa propagation.

Finalement, confiner revient à fortement contraindre les relations entre les personnes.

Dès lors, il devient assez simple d'implémenter le confinement dans GEMFSim. Il suffit en effet de supprimer un certain nombre de liens entre les graphes du noeud, comme on peut le voir sur les deux graphes suivants.

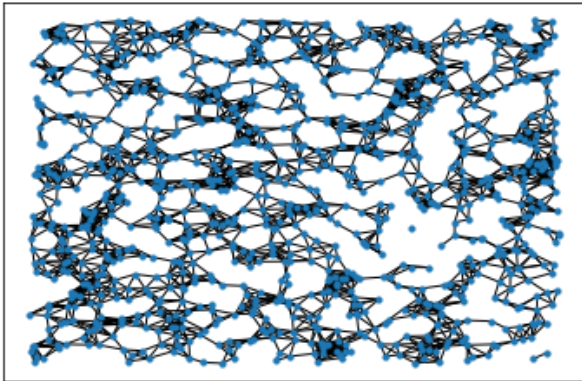


FIGURE 4 – Population sans restrictions

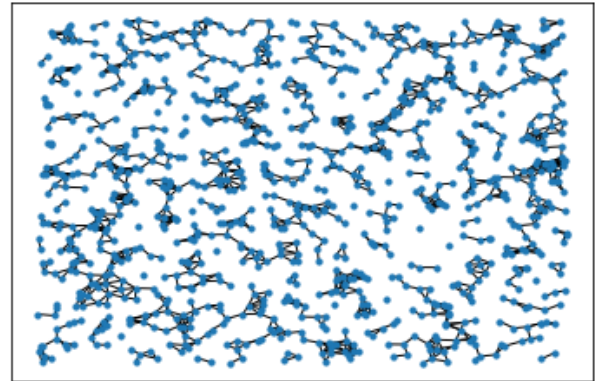


FIGURE 5 – Population confinée : de nombreux liens sont supprimés entre les noeuds

Généralement, le confinement est déclenché dès que l'épidémie explose, et qu'elle est donc amenée à devenir incontrôlable. Finalement, l'action de confiner est décidée, dans le cadre d'une épidémie, dès lors qu'un certain seuil d'infectés est atteint.

Ainsi, pour reproduire le confinement dans notre simulateur, il suffit de préciser que dès qu'un certain seuil d'infectés est atteint, un certain nombre de liens entre les noeuds de notre graphe sont supprimés.

Nous avons par la même occasion implémenté une situation de déconfinement, de manière très similaire : dès lors qu'un certain seuil bas d'infectés est dépassé, synonyme d'un contrôle de l'épidémie, les fameux liens supprimés sont rétablis entre les noeuds.

### 2.5.2 Implémentation du couvre-feu

On pourrait comparer la mesure du couvre-feu à un confinement de très courte durée. En effet, prenons l'exemple de la France, où depuis un certain temps, un couvre-feu est imposé en fin de journée. Pendant la durée de ce couvre-feu, il est strictement interdit de se réunir.

Une façon simple d'implémenter le couvre-feu est donc, sur une courte période de temps, de supprimer des liens entre les noeuds. Ces mêmes liens sont rétablis dès la fin de cette courte période de temps, et ce mécanisme est répété très régulièrement jusqu'à ce que les mesures de couvre-feu soient levées.

### 2.5.3 Implémentation du $R_0$

Suite à nos réflexions précédentes ([partie 1.1.3](#)) et à partir d'autres recherches[9], le  $R_0$  est finalement simplement défini comme le nombre attendu d'infectés causé par une personne infectée (état I) dans une population de personnes susceptibles (état S).

En suivant cette définition, il devient très simple d'implémenter le calcul du  $R_0$  dans GEMFSim. En effet, il suffit de compter, à chaque changement d'état du graphe  $\mathcal{G}$ , la proportion de noeuds

dans un état susceptible, et de multiplier cette proportion par le rapport  $\frac{\beta}{\delta}$ , représentant le rapport entre le taux d'infection et le taux de rémission.

### 3 Simulations

Après avoir compris le fonctionnement du modèle, nous avons décidé d'effectuer des simulations, dans l'objectif d'analyser l'impact des différents paramètres du modèle, ainsi que les mesures restrictives que nous avons mises en place. Ensuite, nous avons voulu comparer qualitativement notre modèle stochastique et le modèle déterministe. Finalement, nous avons souhaité utiliser un des principaux avantages de GEMFSim, qui permet d'implémenter notre propre modèle de propagation d'épidémie.

#### 3.1 À propos du modèle SIS

Nous avons rapidement décidé d'abandonner le modèle SIS. En effet, celui-ci est construit tel qu'un noeud peut-être infecté un nombre infini de fois.

On observe donc deux cas différents :

- soit l'épidémie s'éteint très rapidement, et dans ce cas-là, un équilibre est atteint tel que  $(pS, pI) = (1, 0)$ <sup>8</sup>
- soit l'épidémie se propage, et finit par se stabiliser tel qu'on peut le voir sur le graphique suivant

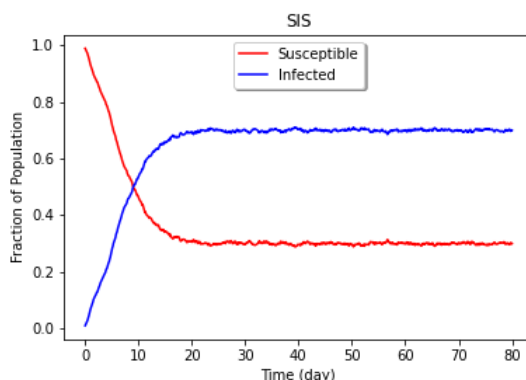


FIGURE 6 – Simulation stochastique du modèle SIS ( $\beta = 0.6, \delta = 0.3$ )

Ainsi, à l'image de son système déterministe, notre implémentation stochastique du modèle SIS présente un point d'équilibre. Malheureusement, nous n'avons pas réussi à déterminer cet équilibre pour notre modèle.

#### 3.2 Influence des paramètres du modèle

Dans cette partie, nous allons analyser l'influence des principaux paramètres du modèle que l'on peut modifier, à savoir le taux d'infection, le taux de rémission, la taille de la population, ainsi que le nombre d'infectés initial.

##### 3.2.1 Taux d'infection

On s'intéresse au paramètre  $\beta$ , qui définit le taux de transition d'un état S à un état I. Nous travaillons avec un  $\delta$  fixé à 1, un graphe de 1000 noeuds et une population initiale d'infectés de 0,3% de la population.

Nous obtenons les graphes suivants.

---

8.  $pS$  représente la proportion de population dans un état S,  $pI$  celle dans un état I

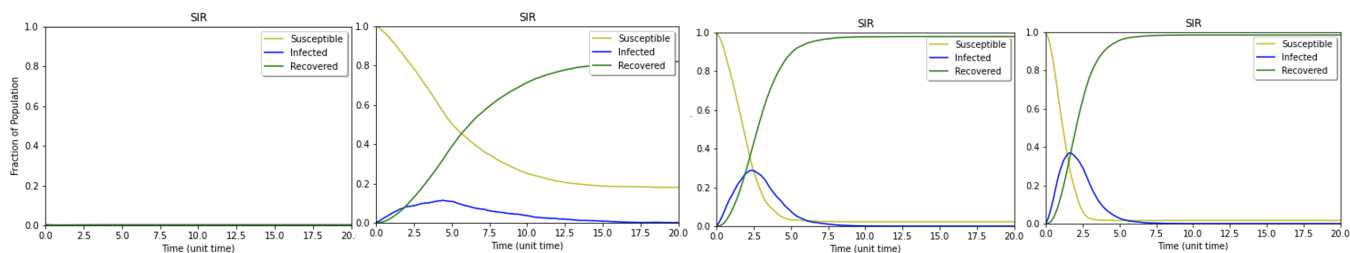


FIGURE 7 – Simulations pour  $\beta$  valant 0, 1, 2 et 3

Regardons tout d'abord le cas trivial, où  $\beta$  vaut 0. Dans ce cas-ci, le virus ne se transmet pas, c'est-à-dire qu'une personne susceptible ne peut pas devenir infectée. On obtient donc très logiquement 3 courbes :

- celle des susceptibles, qui reste à 0,997 (personne ne sort de l'état susceptible, occupé par 99,7% de la population initialement)
- celle des infectées, qui passe très rapidement à 0, car les infectés se remettent du virus
- celle des rémis et immunisés, qui croît de 0 à 0,03, car seuls les 3 noeuds initialement infectés peuvent se remettre de ce virus, qui ne circule pas

On remarque, grâce aux 3 autres courbes, que plus  $\beta$  grandit, plus le pic d'infectés est haut, et est atteint rapidement. En effet, plus  $\beta$  est grand, et plus le virus se transmet vite. Il est donc logique de voir un pic de plus en plus haut et atteint de plus en plus rapidement. On note aussi que la courbe des rémis a son maximum de plus en plus grand. C'est aussi logique. En effet, à  $\delta$  fixé, les infectés mettent autant de temps à guérir, mais plus le virus circule vite, plus il y aura de personnes qui seront infectées, et donc plus de personnes guériront du virus. Au passage, notons que cela n'est pas une bonne chose. Notre modèle est très simplifié et peu réel, donc certes tout le monde sera potentiellement immunisé contre le virus (si tant est que cette immunité soit effective au cours du temps), mais cela implique plus de victimes potentielles, ainsi qu'une plus grande surcharge des hôpitaux, les deux points qu'il faut réduire au maximum dans le cadre du contrôle de la propagation d'une épidémie.

Ainsi, plus  $\beta$  est grand, plus le virus circule rapidement, et donc plus de personnes sont infectées de plus en plus rapidement.

### 3.2.2 Taux de rémission

Nous nous intéressons désormais au paramètre  $\delta$ , qui définit le taux de transition d'un état I à un état R. Comme précédemment, nous travaillons avec un  $\beta$  fixé, qui vaut 2, un graphe de 1000 noeuds et une proportion initiale d'infectés de 0,3% de la population.

Nous obtenons les graphes suivants.

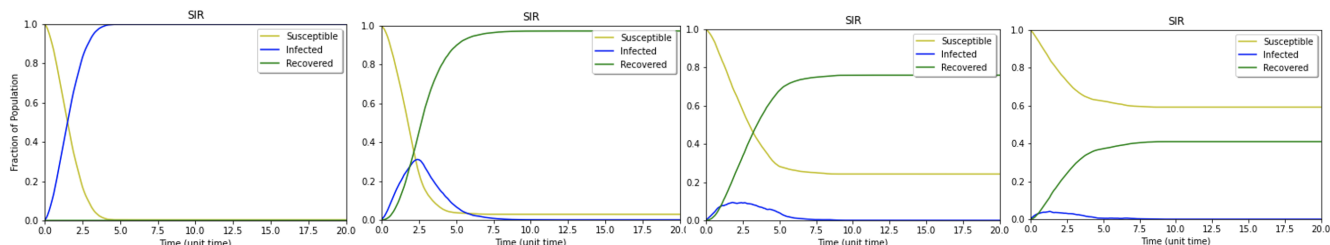


FIGURE 8 – Simulations pour  $\delta$  valant 0, 1, 2 et 3

Nous remarquons tout d'abord le cas trivial, où  $\delta$  vaut 0. Dans ce cas-ci, cela signifie qu'une

fois que l'on devient infecté, on le reste pour toujours. Assez logiquement, la courbe des infectés croît donc jusqu'à 1, à l'inverse de celle des susceptibles, qui décroît vers 0. De plus, la courbe des personnes rémises et immunisées reste à 0, car on ne peut pas guérir du virus dans ce cas-ci.

On remarque ensuite que plus  $\delta$  grandit, ce qui signifie qu'on se remet du virus de plus en plus vite, et plus le nombre d'infectés maximal diminue. La courbe représentant les personnes rémises atteint un maximum de plus en plus petit, étant donné que les infectés ont moins de temps pour propager le virus, car ils guérissent plus vite. Finalement, cette augmentation de  $\delta$  a aussi un impact sur la durée d'extinction de l'épidémie. Plus ce dernier est grand, et plus l'épidémie s'éteindra rapidement.

Finalement, plus les infectés guérissent vite, moins ils ont le temps de propager le virus, moins de personnes seront touchés, et donc l'épidémie s'éteindra plus rapidement, faute d'avoir eu le temps de se propager.

### 3.2.3 Taille de la population

Pour vérifier l'impact de la taille de la population, i.e. le nombre de noeuds de notre graphe, nous avons procédé différemment. Nous avons généré les deux graphes suivants.

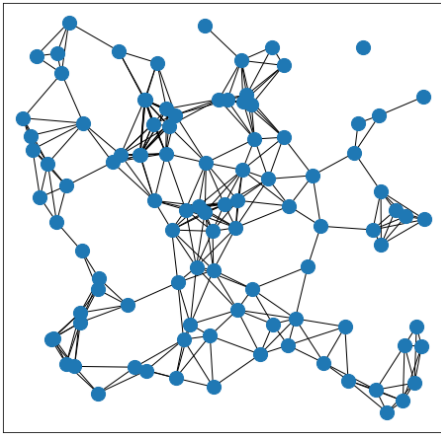


FIGURE 9 – Graphe composé de 100 noeuds

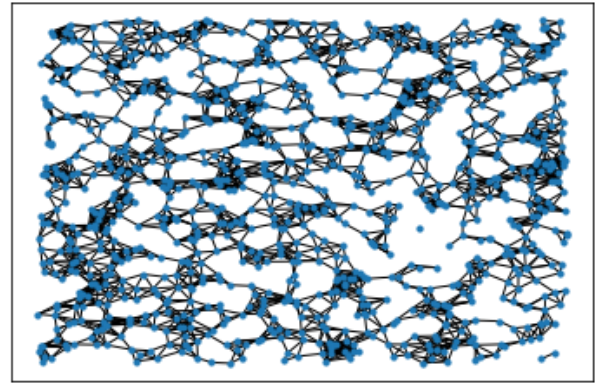


FIGURE 10 – Graphe composé de 1000 noeuds

Ensuite, nous avons simulé la propagation d'une épidémie en prenant initialement 3 noeuds infectés, et des valeurs de  $\beta$  et  $\delta$  dans l'intervalle  $\llbracket 0.1, 4.9 \rrbracket$ .

Nous avons tout d'abord comparé la courbe des infectés. Pour cela, nous avons regardé pour chacune des valeurs de  $\beta$  les valeurs de  $\delta$  correspondantes telles que le maximum d'infectés dépassait 15% de la population totale. Nous avons obtenu les résultats suivants<sup>9</sup>.

9. Courbes obtenues par régression linéaire. Pour le graphe de 100 noeuds, une épidémie provoque un pic de plus de 15% d'infections pour  $\delta \in [0, 1.29\beta]$ . Pour 1000 noeuds, on obtient  $\delta \in [0, 0.88\beta]$ .

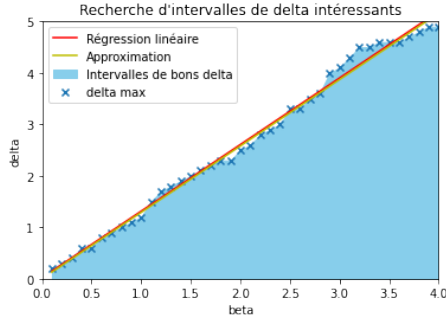


FIGURE 11 – Analyse pour le graphe de 100 noeuds

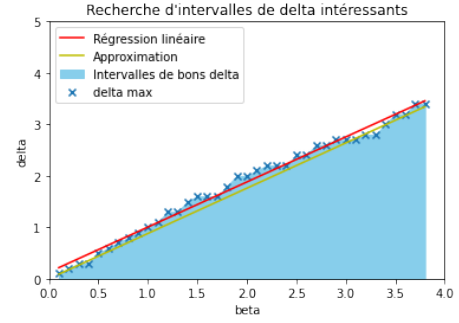


FIGURE 12 – Analyse pour le graphe de 1000 noeuds

On remarque que quand notre nombre de noeuds augmente, le nombre de situations où le pic d'infectés dépasse 15% diminue. Cela signifie que l'épidémie se propage moins quand la population augmente.

Au niveau de la vitesse de propagation, en prenant par exemple les deux graphes suivants, avec des valeurs de  $\beta$  et  $\delta$  provoquant un pic à plus de 15% pour les deux tailles de populations (3 infectés initiaux,  $\beta = 2$ ,  $\delta = 0.8$ ), on remarque qu'il n'y a aucune différence.

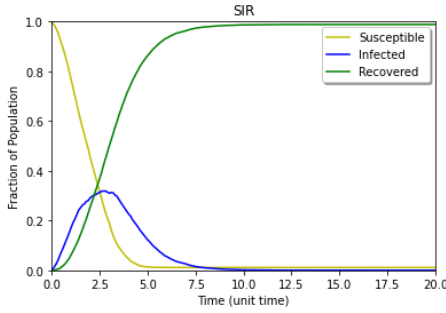


FIGURE 13 – Simulation pour le graphe de 100 noeuds

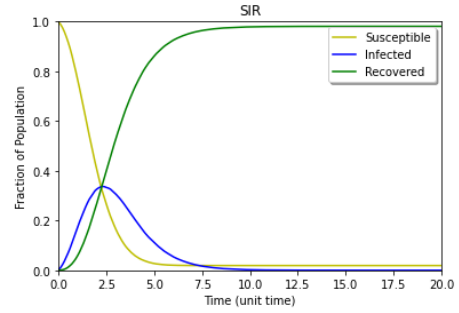


FIGURE 14 – Simulation pour le graphe de 1000 noeuds

Cependant, la différence se fait ressentir pour des couples  $(\beta, \delta)$  qui ne sont pas dans l'intervalle pour l'une ou les deux populations.

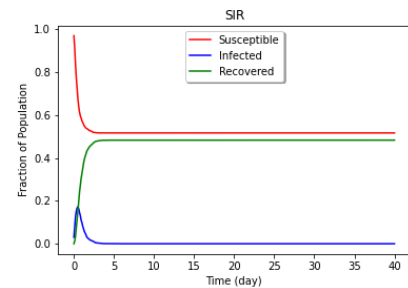


FIGURE 15 – Simulation pour le graphe de 100 noeuds

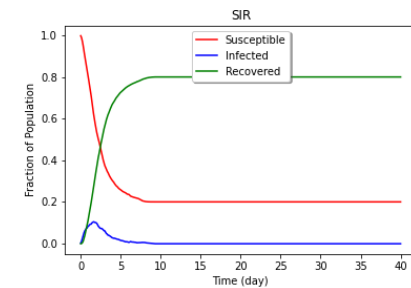


FIGURE 16 – Simulation pour le graphe de 1000 noeuds

Dans les graphes précédents, nous avons choisi  $\beta = 2.4$  et  $\delta = 2.5$ . Pour le graphe de 100 noeuds, on remarque une épidémie qui s'éteint plus rapidement certes, mais qui provoque un pic haut et important. C'est le cas dès que le couple dépasse les valeurs limites pour 1000 noeuds, ou pour 100 et 1000 noeuds. Dans ces cas-ci, on assiste à une propagation de plus en plus rapide au fur et

à mesure que la population diminue, tandis que l'épidémie mettra plus longtemps à s'éteindre, et contaminera plus de personnes, pour une plus grande population.

Finalement, plus la population est grande, plus l'épidémie sera contrôlable. Elle touchera certes plus de personnes, mais sur un intervalle de temps bien plus long et réparti, ce qui permettra de pouvoir s'occuper de tout le monde plus facilement. Cependant, si la population est petite, l'épidémie sera très courte mais hors de contrôle.

### 3.2.4 Pourcentage de population initialement infectée

Nous nous intéressons ici à l'impact du pourcentage de population initialement infectée. Nous considérons un graphe de 1000 noeuds, avec des taux  $\beta$  (égal à 2) et  $\delta$  (égal à 0,8) fixes.

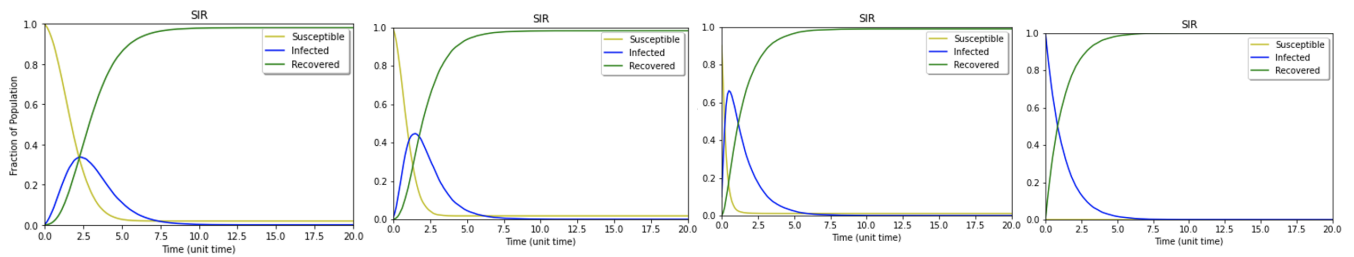


FIGURE 17 – Simulations pour un pourcentage de population initialement infectée de 0, 3%, 1%, 10% et 100%

On remarque tout d'abord 2 cas triviaux :

- 0% initialement infecté, l'épidémie ne se propage pas car aucun noeud ne peut initialement la transmettre (*pas sur la figure 17*)
- 100% initialement infectés, l'épidémie a infecté tous les noeuds dès le début, donc tout le monde se remet du virus et devient immunisé, l'épidémie ne "circule pas"

Ensuite, il est assez clair que plus il y a de noeuds initialement infectés, plus le pic d'infectés est important et est atteint rapidement. En effet, beaucoup plus de noeuds sont capables de propager l'épidémie dès le début, ce qui provoque l'accélération de l'épidémie. Il est donc aussi logique de voir la courbe des personnes rémises et immunisées atteindre plus rapidement son maximum à mesure que le nombre de noeuds initialement infectés augmente, car de plus en plus de personnes guérissent plus rapidement, étant infectées plus tôt. De ce fait, plus il y a de personnes initialement infectées, plus l'épidémie s'éteindra vite.

## 3.3 Mise en place des mesures restrictives

Par la suite, nous avons voulu vérifier l'impact qu'ont des mesures restrictives, telles que le confinement, ou le couvre-feu, sur la propagation d'une épidémie.

Nous avons pour cela voulu comparer 4 stratégies :

- mise en place d'un confinement strict et long
- mise en place d'une stratégie de Stop&Go, avec confinements et déconfinements successifs
- mise en place d'un couvre-feu
- mise en place d'un couvre-feu et d'un confinement léger (confinement où les contacts sont peu limités, avec les écoles ouvertes et pas de recours au télétravail par exemple)

### 3.3.1 Paramètres de simulation

Nous décidons d'effectuer nos simulations sur un graphe composé de 1000 noeuds, avec 3 noeuds infectés au temps  $t = 0$ . Nous commençons donc avec 0,3% de la population infectée.



Nous avons choisi d'effectuer nos simulations sur des intervalles de paramètres restreints.

Nous définissons donc des couples  $(\beta, \delta)$  tels que  $\delta \in \left[\frac{\beta}{5}, \beta\right]$ , avec  $\beta \in \{2, 2.5, 3\}$ . Dans ce cas ci, nous regardons des épidémies où le taux d'infection est de 1 à 5 fois le taux de rémission.

Aussi, nous décidons de définir une ligne de conduite afin de vérifier l'impact de nos mesures : les services hospitaliers sont en mesure de gérer l'épidémie s'il y a au maximum 10% d'infectés dans la population.

### 3.3.2 Graphes des situations

Nous avons ensuite besoin de définir des graphes pour les quatre situations.

Dans le cas où la population n'est soumise à aucune restriction, nous utilisons le [graphe défini précédemment](#). Au niveau informatique, cela revient à définir la variable *radius* telle que *radius* = 0.051.

Dans le cas où une stratégie de confinement est mise en place, nous réduisons drastiquement les liens entre les personnes, et utilisons ce graphe.

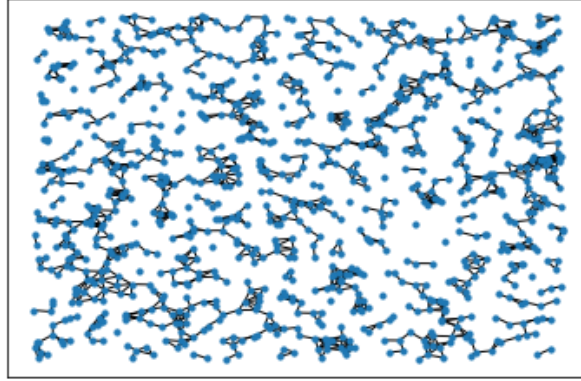


FIGURE 18 – Graphe d'une population de 1000 noeuds sous confinement strict (*radius* = 0.035)

Pour une stratégie de Stop&Go, nous estimons que la population, durant les périodes de déconfinement, renoue contact tout en étant un peu plus prudente qu'auparavant, par peur d'un retour à un confinement. Nous alternerons avec le graphe précédent sous régime de confinement, et le suivant lors du déconfinement.

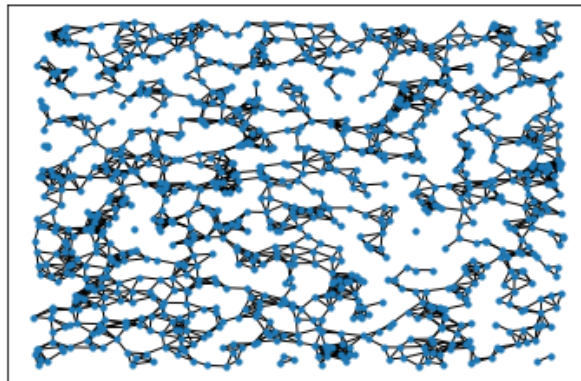


FIGURE 19 – Graphe d'une population de 1000 noeuds déconfinée (*radius* = 0.045)

Dans le cas du couvre-feu, nous estimons que la population respecte bien ce dernier, et très peu de contacts sont établis durant les horaires de couvre-feu.

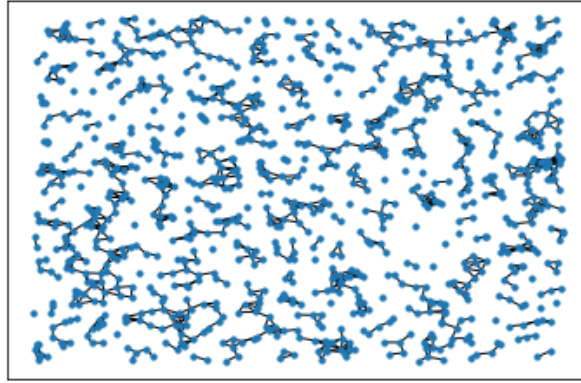


FIGURE 20 – Graphe d’une population de 1000 noeuds sous le couvre-feu ( $radius = 0.032$ )

Finalement, dans le cadre d’un couvre-feu associé à un confinement léger, nous effectuerons une alternance sur le graphe précédent et le suivant.

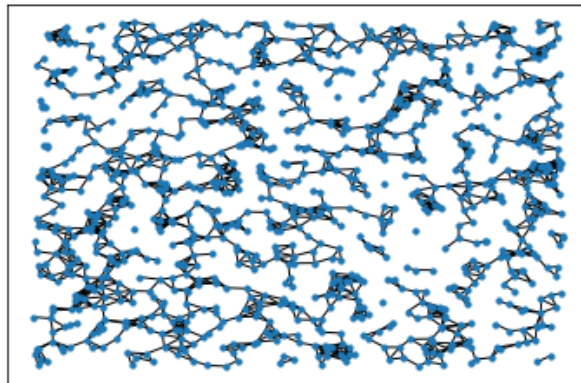


FIGURE 21 – Graphe d’une population de 1000 noeuds sous confinement léger ( $radius = 0.04$ )

Pour rappel, la variable *radius* est la variable qui définit la distance maximale pour laquelle deux noeuds peuvent être liés.

### 3.3.3 Seuils de mise en place des mesures restrictives

Afin de vérifier l’impact de nos mesures, nous établissons des seuils pour lesquels les mesures restrictives seront mises en place.

- Dans le cadre d’une stratégie de confinement long, nous ferons varier notre seuil de confinement entre 2% et 16% de population infectée (pas de 2%). Le confinement sera levé dès que moins d’1% de la population sera infectée.
- Dans le cadre d’une stratégie de Stop&Go, nous ferons varier notre seuil de confinement entre 2% et 16% de population infectée. Le seuil de déconfinement variera quant à lui entre 20% et 60% du seuil de confinement.

Pour le couvre-feu, il n’y aura pas de changement d’horaire de couvre-feu. Nous estimons que ceci aurait un impact très limité sur la propagation du virus. Pour cette stratégie, ainsi que celle consistant à mettre en vigueur un couvre-feu ainsi qu’un confinement léger, les mesures restrictives seront levées dès que moins d’1% de la population sera infectée.

## 3.4 Impact des mesures restrictives

Après avoir clairement défini l’étendue des situations que nous allons analyser, nous avons réalisé toutes les simulations. Nous allons désormais regarder l’impact que ces dernières ont eu sur la

propagation de notre épidémie.

### 3.4.1 Méthode d'analyse

Pour analyser l'impact des mesures restrictives, nous avons comparé 3 principaux indicateurs :

- la durée de l'épidémie
- le pic maximal d'infectés
- le pourcentage de population dans un état  $R$  à la fin de l'épidémie (plus ce pourcentage est important, plus au cours de l'épidémie il y a eu de noeuds dans un état  $I$ , et donc potentiellement beaucoup de victimes (aspect non pris en compte par le modèle SIR)

Finalement, notre objectif est de trouver les mesures qui permettent d'avoir une épidémie la plus rapide possible, tout en ayant le moins de noeuds dans un état  $R$  et en veillant à ce que le pic d'infectés ne dépasse pas les 10%.

### 3.4.2 L'épidémie initiale

Nous choisissons comme valeurs références les valeurs obtenues dans le cadre d'une simulation sans mesure restrictives.

Avec les paramètres [énoncés précédemment](#), nous obtenons ceci.

<i>Durée moyenne de l'épidémie</i>	<i>Moyenne du pic d'infectés</i>	<i>Moyenne des noeuds dans un état <math>R</math></i>
20,58	43%	99%

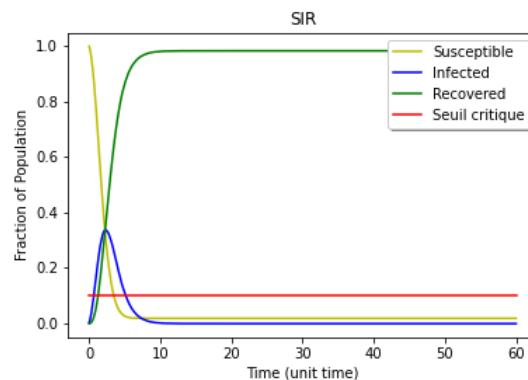


FIGURE 22 – Épidémie sans mesures restrictives ( $\beta = 2, \delta = 0.8$ )

Nous essayons donc d'agir sur des épidémies qui provoquent un pic d'infectés très important, et qui contamine la quasi-totalité de la population.

### 3.4.3 Stratégie de confinement long et strict

Dans un premier temps, analysons le confinement. Rappelons-nous qu'il est possible de le déclencher à partir d'un certain seuil d'infectés.

<i>Seuil de confinement</i>	2%	4%	6%	8%	10%	12%	14%	16%
<i>Durée moyenne de l'épidémie</i>	17,28	19,46	20,23	20,45	18,93	18,64	19,14	18,33
<i>Moyenne du pic d'infectés</i>	4,4%	6,1%	8%	9,9%	11,3%	13%	14,6%	16,4%
<i>Moyenne des noeuds <math>R</math></i>	13,1%	16,1%	19,5%	22,3%	25,5%	29,4%	32,1%	35,3%

On remarque assez facilement que le confinement remplit très bien sa mission, qui est de couper les chaînes de transmission dans l'objectif de diminuer fortement les contaminations et de stopper la propagation du virus.

Cette mission est remplie quel que soit le seuil de déclenchement de confinement, même s'il convient (logiquement) de le mettre en vigueur avant notre seuil de 10% d'infectés maximal pour que ce dernier ne soit pas dépassé.

Il est aussi très efficace au niveau du nombre de personnes immunisées. En effet, ce nombre est à minima divisé par 3, et même jusqu'à 9 selon le seuil de déclenchement. Ceci reste logique : de nombreux liens sont supprimés, et donc le virus se transmet seulement dans quelques petits clusters, permettant à une grande partie de la population de ne jamais être en contact avec le virus.

Finalement, au niveau de la durée de l'épidémie, il permet de la réduire un petit peu, mais pas significativement. Cette réduction est marquée si le confinement est déclenché très tôt ou assez tard, ce qui est logique :

- s'il est déclenché très tôt, l'épidémie ne circulera que dans des petits clusters, et s'éteindra assez vite
- s'il est déclenché plus tard, l'épidémie aura déjà contaminé un certain nombre de personnes, surtout les noeuds proches de ceux actuellement infectés, et donc ceux-ci auront beaucoup moins de possibilités de contaminer d'autres noeuds, et auront donc comme unique possibilité de guérir

Cependant, s'il est déclenché à un moment où l'épidémie commence à bien se développer, la période de contamination s'étendra un peu, avant que tous les noeuds guérissent totalement.

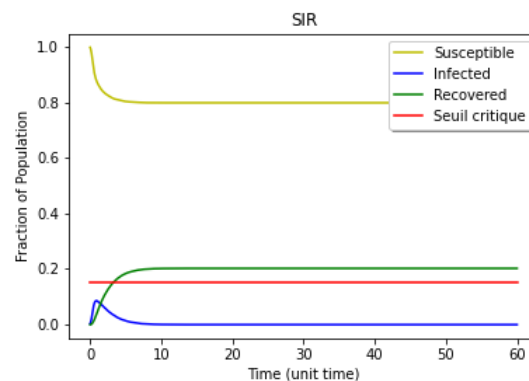


FIGURE 23 – Population soumise à un confinement (seuil à 8%,  $\beta = 2$ ,  $\delta = 0.8$ )

Par rapport au graphe précédent d'une épidémie sans restriction, on remarque très facilement l'impact du confinement, qui permet de diminuer très largement le nombre d'infectés, et de ne jamais dépasser le pic des 10%.

Finalement, la mise en place d'un confinement long et strict s'avère très efficace pour arrêter la propagation d'une épidémie, car il permet de couper les chaînes de transmission, et donc de diminuer le nombre de personnes touchées par le virus.

### 3.4.4 Stratégie de Stop&Go

La stratégie de Stop&Go consiste en une alternance de confinements et de déconfinements successifs dès que certains seuils sont atteints. Comme nous l'avons expliqué précédemment, nous avons réalisé des simulations pour de nombreuses valeurs de seuils de confinements et de déconfinements. Une tendance générale se remarque facilement, en témoigne le tableau suivant, pour un seuil de confinement de 8% d'infectés, et des seuils de déconfinements variables.

Seuil de déconfinement	2%	4%	5%	Situation normale	Confinement strict à 8%
Durée moyenne de l'épidémie	34,49	34,16	32,97	20,58	20,45
Moyenne du pic d'infectés	9,69%	9,69%	9,8%	43%	9,9%
Moyenne des noeuds $R$	48,2%	62%	65%	99%	22,3%
Nombre de confinements	2,11	3,2	3,87		

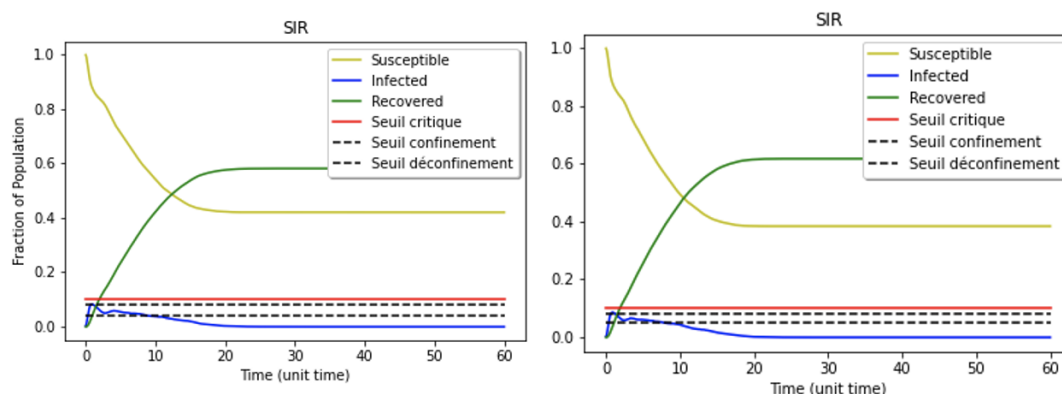


FIGURE 24 – Population soumises à une stratégie de Stop&Go (seuil de confinement à 8%, seuil de déconfinement à 4% à gauche, 5% à droite,  $\beta = 2, \delta = 0.8$ )

On remarque tout d'abord que l'épidémie dure plus longtemps sous une stratégie de Stop&Go. L'enchaînement de suppressions et de rajouts de liens rend difficile l'évolution de l'épidémie, et augmente donc le temps avant qu'elle ne s'éteigne.

On remarque qu'il y a très peu de variations au niveau du pic d'infectés entre une stratégie de confinement strict et une stratégie de Stop&Go. Finalement, plus le seuil de déconfinement est grand, et donc plus on a tendance à déconfiner rapidement, plus le nombre de personnes touchées par l'épidémie augmente. C'est assez logique : plus on déconfiner rapidement, moins les contacts entre les gens sont réduits, et donc l'épidémie circule plus.

Aussi, plus on déconfiner tôt, plus de confinements il y aura. En effet, le nombre d'infectés n'aura pas eu le temps de descendre suffisamment, et va donc rapidement remonter, obligeant un nouveau confinement.

Cette tendance pour un seuil de confinement à 8% s'étend aussi aux autres seuils de confinement et de déconfinement :

- plus on déconfiner tôt, moins l'épidémie dure
- l'impact d'une stratégie de Stop&Go sur le pic d'infectés est quasiment nul par rapport à un confinement strict
- plus on déconfiner tôt, plus l'épidémie circule, et donc le nombre moyen de noeuds dans un état  $R$  augmente
- plus on déconfiner tôt, plus il y aura de confinements

La stratégie de Stop&Go ne semble pas être à privilégier, étant donné qu'elle augmente la probabilité d'être touchée par l'épidémie, ainsi que sa durée. Cependant, elle permet à la population de ne pas être coupée entièrement pendant une longue période, à l'inverse d'un confinement strict, mais elle l'oblige à se confiner plus de fois...

### 3.4.5 Stratégie de couvre-feu

	<i>Durée moyenne de l'épidémie</i>	<i>Moyenne du pic d'infectés</i>	<i>Moyenne des noeuds R</i>
<i>Situation normale</i>	20,58	43%	99%
<i>Couvre-feu</i>	21,2	35,9%	97,8%

Au contraire de ce qui a pu être entendu récemment, le couvre-feu est efficace dans sa mission. En effet, on peut voir une certaine diminution dans le pic d'infectés, ainsi qu'une durée d'épidémie un peu supérieure. C'est assez logique, car on supprime les chaînes de transmission seulement momentanément. Pendant un certain intervalle de temps, les contacts sont très fortement limités, et donc les contaminations. Cependant, dès le retour de ces contacts, le processus de contamination reprend, et les noeuds n'ayant pas guéri durant le couvre-feu peuvent de nouveau contaminer. Comme attendu, le nombre de personnes touchées par le virus ne diminue quasiment pas : autant de personnes sont touchées, mais sur une période de temps un peu plus longue.

Finalement, le couvre-feu remplit sa mission, mais n'est pas assez efficace. Même s'il permet de diminuer le pic d'infectés, et donc de soulager la pression hospitalière, il ne stoppe pas la propagation du virus, il la diminue seulement.

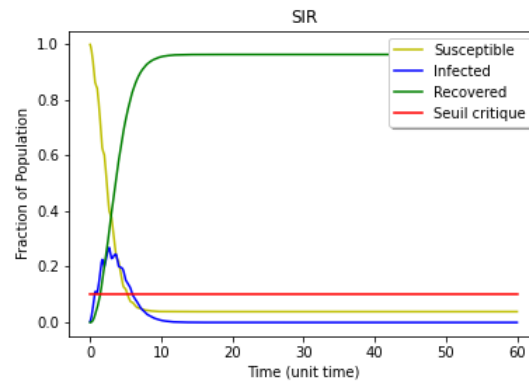


FIGURE 25 – Population soumise à un couvre-feu ( $\beta = 2, \delta = 0.8$ )

Sur le graphe, on peut remarquer notamment sur la courbe des infectés certains endroits où cette dernière se stabilise voir diminue. Ceci est dû à la façon dont nous avons implémenté le couvre-feu : pendant le tiers d'une unité de temps, les contacts sont tellement contraints que le processus d'infection s'arrête, ce qui implique des portions de stagnation sur la courbe des infectés.

### 3.4.6 Stratégie de confinement léger et de couvre-feu simultanés

Le couvre-feu peut être rendu plus efficace en ne rétablissant pas la totalité des liens supprimés. En effet, dès la fin du couvre-feu, s'il n'y a pas d'autres mesures en même temps, toutes les chaînes de transmission sont immédiatement rétablies. Cependant, en liant le couvre-feu à un confinement, c'est à dire en supprimant sur le long-terme un certain nombre de liens, et encore plus pendant le couvre-feu, l'effet devrait être plus important sur le nombre de personnes ayant été touchées par le virus.

<i>Seuil de confinement</i>	2%	4%	6%	8%	10%	12%	14%	16%
<i>Durée moyenne de l'épidémie</i>	40	36,67	35,5	34,58	32,4	30,71	28,83	30,14
<i>Moyenne du pic d'infectés</i>	8,7%	9,7%	11%	12,5%	14,1%	15,3%	16,9%	18,1%
<i>Moyenne des noeuds R</i>	50,1%	54,9%	58,7%	61,3%	63,5%	67,1%	70,5%	71%

On obtient des résultats assez attendus. Allier confinement à couvre-feu, malgré le fait que l'épidémie dure plus longtemps, permet de fortement diminuer le pic d'infectés, ainsi que le nombre de personnes touchées par le virus. On remarque aussi ce qui avait pu être vu auparavant : plus le confinement est déclenché tard, moins l'épidémie dure, mais plus de personnes sont touchées, et le pic d'infectés est plus important.

Cette stratégie semble intéressante, même s'il faut confiner plus rapidement pour ne pas dépasser un certain pic d'infectés, qui croît plus rapidement qu'avec un confinement plus strict.

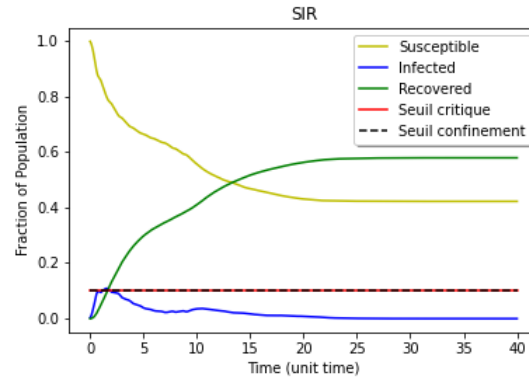


FIGURE 26 – Population soumise à un couvre-feu et à un confinement léger (seuil de confinement à 8%,  $\beta = 2$ ,  $\delta = 0.8$ )

Notons que par rapport au graphe du couvre-feu, les courbes de cette stratégie sont plus lisses. En effet, il n'y a pas les petits intervalles de stabilisation car la population est soumise à un confinement, et donc le couvre-feu supprime beaucoup moins de liens que lors de la situation précédente.

### 3.4.7 Conclusion sur les mesures restrictives

Finalement, on peut affirmer que les mesures restrictives peuvent avoir un gros impact sur la propagation d'un virus.

De plus, malgré la simplicité du modèle SIR, elles représentent plutôt bien l'évolution du covid dans certains pays.

Prenons l'exemple de la France. Le couvre-feu a simplement servi à freiner l'épidémie pendant un certain temps, avec un pic de contaminations prédit par les scientifiques qui a fini par être atteint. D'une manière globale, nous sommes soumis à une stratégie de Stop&Go, et nous avons pu remarquer l'impact d'une stratégie couvre-feu et confinement, qui a permis de réduire fortement le pic d'infections auquel on aurait pu s'attendre.

Cependant, il faut prendre du recul sur notre modèle, bien trop simpliste pour représenter l'évolution d'une épidémie dans un pays. Il reste tout de même un bon moyen d'estimer facilement l'impact que diverses mesures restrictives peuvent avoir. Ainsi, nous ne ferons pas l'erreur d'affirmer qu'une stratégie semble plus adaptée qu'une autre, à cause de la simplicité de notre modélisation. De plus, nous ne pouvons ici qu'analyser l'impact comptable, alors que d'autres aspects sont aussi à prendre en compte (psychologique, économique, ...).

### 3.4.8 À propos du $R_0$

Rappelons que le  $R_0$ , aussi appelé nombre de reproduction de base, est un moyen de déterminer si un virus circule ou non à travers la population. Il représente le nombre attendu de cas causés par une personne infectée. Ainsi, si ce nombre est supérieur à 1, notre épidémie circule.

Dès lors que le pic d'infectés est atteint, et donc que le nombre d'infectés diminue, cela signifie que l'épidémie régresse, et donc le  $R_0$  est inférieur à 1.



Finalement, le  $R_0$  devient inférieur à 1 dès lors que la courbe des infectés est décroissante.

Au cours de nos simulations, nous avons pu vérifier que c'était bel et bien le cas, comme le montre la figure suivante.

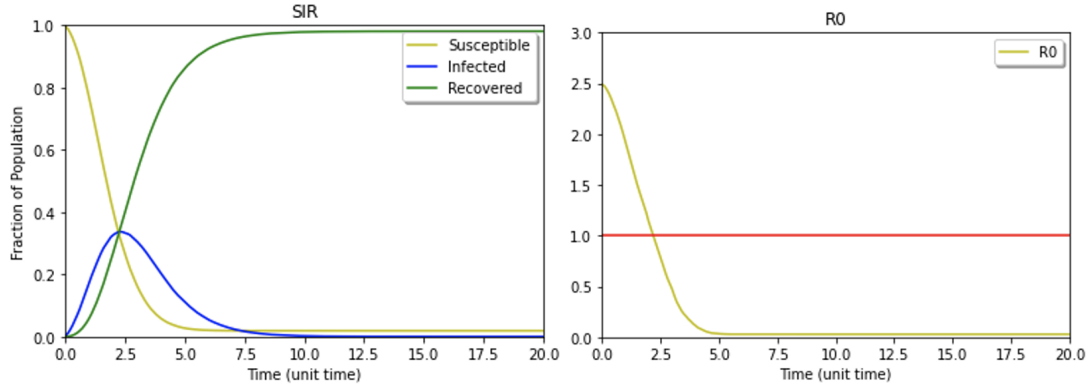


FIGURE 27 – Simulation pour  $\beta = 2$  et  $\delta = 0.8$

### 3.5 Implémentation du modèle déterministe

Au niveau informatique, implémenter le modèle déterministe revient à coder la résolution du système d'équations différentielles suivant au cours du temps.

$$\begin{aligned}\frac{dS}{dt} &= -aSI \\ \frac{dI}{dt} &= aSI - bI \\ \frac{dR}{dt} &= bI\end{aligned}$$

Pour cela, nous utilisons la méthode de Runge-Kutta d'ordre 4. Le code est [en annexe](#), avec les implémentations des mesures restrictives.

#### 3.5.1 Comparaison avec notre modèle stochastique

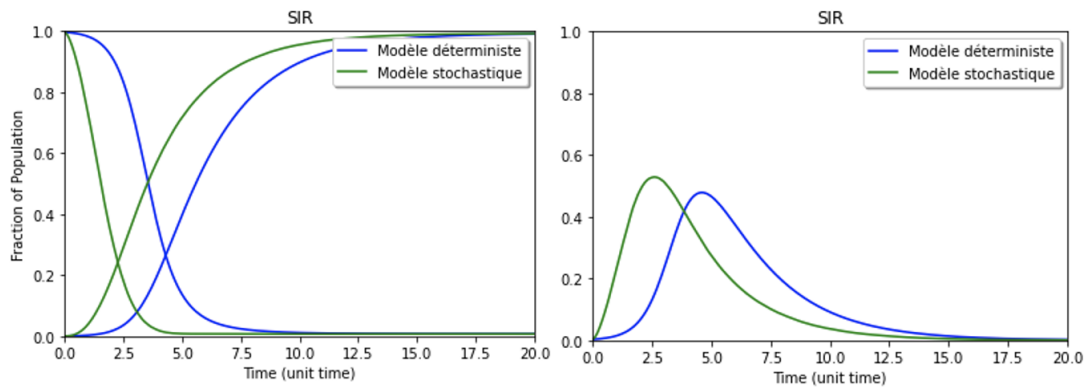


FIGURE 28 – À gauche courbe des Susceptible et des Recovered, à droite des Infected ( $\beta = 2$ ,  $\delta = 0.4$ )

Nous avons ainsi voulu comparer les résultats donnés par la résolution du modèle déterministe et ceux de notre modèle stochastique, afin de vérifier la pertinence de nos résultats. Cependant,



quelques différences se font ressentir, comme on peut le voir sur les deux graphiques précédents.

Tout d'abord, le pic est atteint plus rapidement pour le modèle stochastique. De plus, on remarque aussi qu'il atteint des valeurs plus importantes. Hormis ceci, les courbes sont sinon assez semblables.

Nous avons deux explications à ce phénomène. Tout d'abord, on peut remarquer que l'augmentation du nombre d'infectés est plus lente pour le modèle déterministe. Cela s'explique notamment par le fait que, par construction, le premier événement de notre modèle stochastique est une infection. La courbe représentant les infectés croît donc directement pour le modèle stochastique. De l'autre côté, le modèle déterministe est un peu plus lent au niveau des calculs. Tout ceci permet notamment d'expliquer le décalage du pic. La hauteur de ce pic peut quant à elle être expliquée par la lenteur du modèle déterministe. En effet, le fait que le nombre d'infectés ne croît pas directement permet aux individus initialement infectés de commencer leur processus de guérison. Ainsi, alors que dans notre modèle stochastique les premiers événements seront forcément des infections, certains individus du modèle déterministe auront commencé à guérir, ce qui aura tendance à diminuer le pic d'infectés.

Un autre point peut aussi expliquer ces quelques différences. Le modèle déterministe considère une population totalement uniforme, alors que le graphe sur lequel on applique notre algorithme ne l'est pas. En effet, nous avons plusieurs petits clusters, et des noeuds plus isolés, autrement dit, certains noeuds sont très enclins à être connectés à des noeuds infectés, alors que d'autres ne le seront jamais.

### 3.5.2 Correction du modèle stochastique

Nous avons cherché à modifier notre graphe dans le but de faire correspondre plus exactement le modèle stochastique et déterministe. Pour cela, nous avons tout d'abord commencé par créer un graphe où tous les noeuds ont exactement le même nombre de noeuds en commun, comme par exemple le graphe suivant.

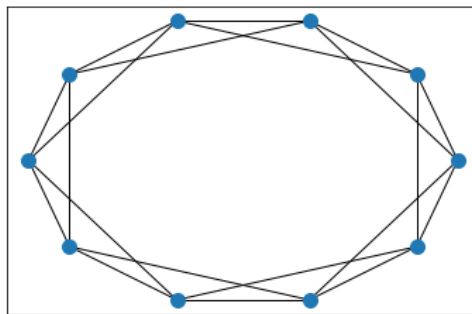


FIGURE 29 – Exemple de graphe pour  $N = 10$

Cependant, cette tentative ne s'est pas avérée fructueuse, et ce peu importe le nombre de noeuds et le nombre de liens que nous attribuons entre chaque noeud.

## 3.6 Implémentation de notre propre modèle

Finalement, nous avons décidé d'utiliser une des forces de GEMFSim : la possibilité de rajouter très simplement son modèle. Après avoir expliqué pourquoi nous avons fait cela, nous présenterons notre nouveau modèle et expliquerons comment nous avons pu l'implémenter sur GEMFSim, avant de conclure sur notre modèle en présentant quelques résultats.

### 3.6.1 Limites du modèle SIR

Il s'avère que le modèle SIR est extrêmement limité (c'est encore plus le cas pour le modèle SIS). Cependant, c'est quelque chose de voulu par le modèle, qui permet de représenter de manière très simplifiée une épidémie.

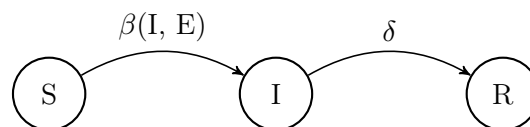
En effet, le modèle SIR ne laisse que trois états possibles : être susceptible d'être contaminé par le virus, être infecté (et contagieux), et finalement guérir de la maladie et devenir immunisé. Cependant, en pratique, une épidémie est bien plus complexe. Certaines fois, même après avoir récupéré du virus, on peut être infecté une seconde fois. De plus, être infecté, ce n'est pas seulement être contagieux : dans beaucoup de cas, les malades doivent être hospitalisés, et certaines fois dans des unités spécialisées, telles que la réanimation. De plus, dans le cadre d'une épidémie, il convient d'isoler rapidement les malades, et un gros accent est mis sur les cas-contacts, personnes ayant été exposées au virus mais dont on ne sait pas si elles sont infectées. Finalement, une seule issue est possible dans le modèle SIR, être immunisé du virus.

Cette vision très simpliste possède de nombreux avantages, comme une analyse plutôt facile. Cependant, nous avons décidé d'implémenter un modèle plus complexe et un peu plus réaliste, afin de voir dans un cadre plus commun comment peut se propager une épidémie.

### 3.6.2 Notre modèle

Nous allons créer notre propre modèle compartimental à partir du modèle SIR. En effet, nous souhaitons complexifier le modèle en ajoutant de nouveaux états et taux de transitions pour décrire au mieux la réalité d'une épidémie, comme celle du covid-19 par exemple. Le modèle SIR sur lequel nous nous appuyons est ainsi assez simpliste puisqu'il ne regroupe que trois états : l'état sain, l'état infecté et l'état immunisé. Nous avons décidé de garder une population constante : nous ne considérons pas le nombre de naissances pendant toute la durée de l'épidémie. En ajoutant un nouvel état comptabilisant le nombre de personnes décédées après avoir contracté la maladie, nous laissons ces personnes au sein de la population. Finalement, tout au long de l'épidémie, la population totale restera constante.

**Choix des différents états du modèle** Rappelons le modèle SIR de base.



$$\begin{aligned}\frac{dS}{dt} &= -\beta SI \\ \frac{dI}{dt} &= \beta SI - \delta I \\ \frac{dR}{dt} &= \delta I\end{aligned}$$

On retrouve donc :

- $S(t)$  : le nombre de personnes susceptibles d'être infectées
- $I(t)$  : le nombre de personnes infectées
- $R(t)$  : le nombre de personnes immunisées
- $\beta$  : le taux d'infection, taux de passage entre l'état S et R
- $\delta$  : le taux de rémission, taux de passage entre l'état R et I

Nous allons complexifier ce modèle pour qu'il soit plus proche de la réalité. Ainsi nous allons rajouter un état  $M(t)$  correspondant au nombre de personnes décédées après avoir contracté la

maladie. Pour cela nous rajoutons le taux  $\gamma$ , le taux de mortalité des personnes n'étant pas en salle de réanimation (taux de passage entre l'état I et l'état M).

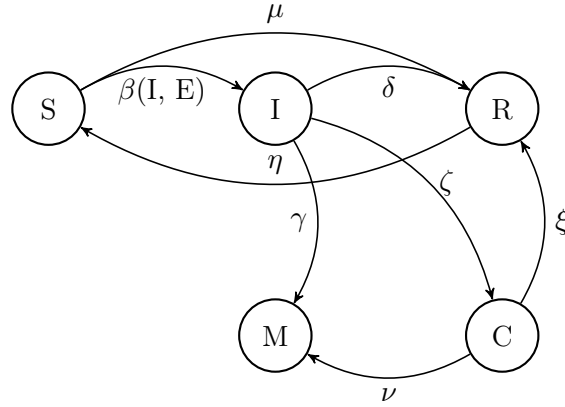
Par ailleurs, pour faire face à cette épidémie, nous imaginons qu'un vaccin a été trouvé pour contrer la maladie. Nous introduisons donc le taux  $\mu$ , le taux de vaccination (taux de passage entre les états S et R).

Enfin, on va aussi prendre en compte la durée limitée de l'immunité acquise soit par le vaccin soit après avoir contracté la maladie. On rajoute donc le taux  $\eta$ , taux correspondant à la fin de l'immunité acquise (taux de passage entre l'état R et S).

On rajoute également un état C(t) correspondant au nombre de personnes en salle de réanimation. Pour cela nous rajoutons trois taux de transition ;

- le taux  $\zeta$  : taux de passage entre les états I et C, représentant la population en salle de réanimation
- le taux  $\xi$  : taux de rémission des personnes en salle de réanimation, taux de transition entre les états C et R
- le taux  $\nu$  : taux de mortalité en salle de réanimation, taux de transition entre les états C et M

Voici maintenant un graphe représentant notre nouveau modèle.



Voici les équations définissant notre nouveau modèle.

$$\begin{aligned}
 \frac{dS}{dt} &= -\beta SI - \mu S + \eta R \\
 \frac{dI}{dt} &= \beta SI - \delta I - \gamma I - \zeta I \\
 \frac{dR}{dt} &= \delta I + \mu S - \eta R + \xi C \\
 \frac{dC}{dt} &= \zeta I - \nu C - \xi C \\
 \frac{dM}{dt} &= \gamma I + \nu C
 \end{aligned}$$

Ainsi à présent nous avons un modèle plus complet et rendant compte de manière plus précise de l'évolution de l'épidémie.

**Implémentation de notre modèle avec GEMFSim** Pour implémenter ce modèle, il nous faut définir une nouvelle méthode *Para*. En effet, comme nous l'avons expliqué dans [la partie 2.4.4](#), pour simuler la propagation du virus et obtenir l'évolution des populations des différents états, GEMFSim a besoin de la liste *Para*, définie à l'aide de la méthode *Para\_SIR* dans le cas du modèle SIR. Cette liste *Para* permet de définir le nombre d'états pour les noeuds, l'état-influeur (qui sera toujours

l'état I pour notre nouveau modèle), tout comme les deux matrices  $A_\beta$  et  $A_\delta$ . Notre nouveau modèle s'appellera SIRCM.

```
1 Para = Para_SIRCM(delta, beta, nu, xi, gamma, mu, eta, zeta)
```

Algorithme 5 – Déclaration de la liste Para pour le modèle SIRCM

Dans le cadre du modèle SIRCM, la liste *Para* sera décrite comme suit :

- $M = 5$  (nombre d'états possibles pour chaque noeud)
- $q = 1$  ( $q$  représente l'état-influenceur, dans notre cas l'état I, qui est codé par 1 dans le simulateur)
- les deux matrices  $Ab$  (transitions edge-based) et  $Ad$  (transitions node-based) telles que

$$Ab = \begin{pmatrix} 0 & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad Ad = \begin{pmatrix} 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & \delta & \zeta & \gamma \\ \eta & 0 & 0 & 0 & 0 \\ 0 & 0 & \xi & 0 & \nu \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

### 3.6.3 Quelques résultats obtenus

Nous avons ainsi pu simuler différentes situations épidémiques en faisant varier les paramètres de notre modèle.

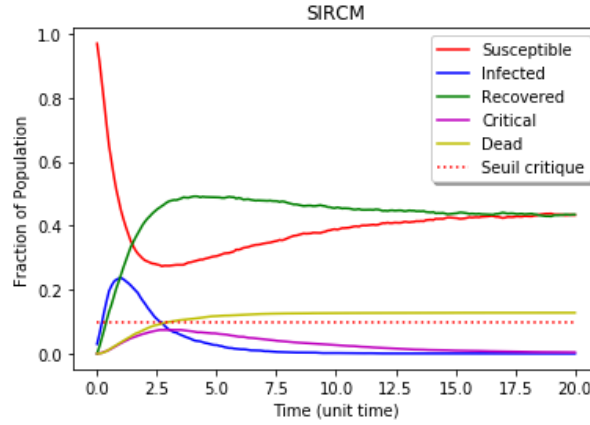


FIGURE 30 – Épidémie sans mesures restrictives ( $\beta = 1.5, \delta = 0.8, \nu = 0.016, \xi = 0.2, \eta = 0.2, \zeta = 0.2, \gamma = 0.2, \mu = 0.2$ )

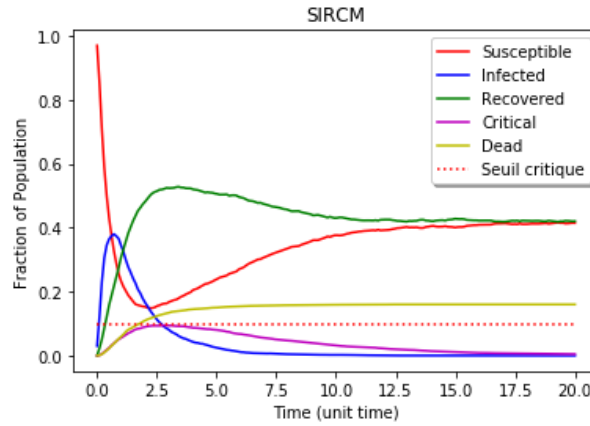


FIGURE 31 – Épidémie sans mesures restrictives ( $\beta = 3, \delta = 0.8, \nu = 0.016, \xi = 0.2, \eta = 0.2, \zeta = 0.2, \gamma = 0.2, \mu = 0.2$ )

On observe bien la même tendance que celle que nous avons trouvé précédemment concernant le taux  $\beta$  comme vous pouvez le constater sur le graphe ci-dessus. Si on augmente le taux  $\beta$  le virus circule plus facilement. Ainsi on a bien que plus  $\beta$  grandit, plus le pic d'infectés est haut, et est atteint rapidement.

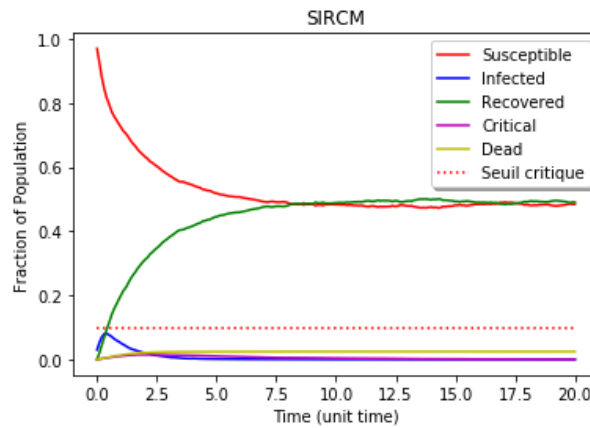


FIGURE 32 – Épidémie avec confinement strict (seuil de confinement à 10%,  $\beta = 1.5, \delta = 0.8, \nu = 0.016, \xi = 0.2, \eta = 0.2, \zeta = 0.2, \gamma = 0.2, \mu = 0.2$ )

On remarque également que le confinement porte ses fruits et est efficace, en réduisant considérablement la propagation du virus et en limitant le nombre de personnes décédées après avoir été déclenché.

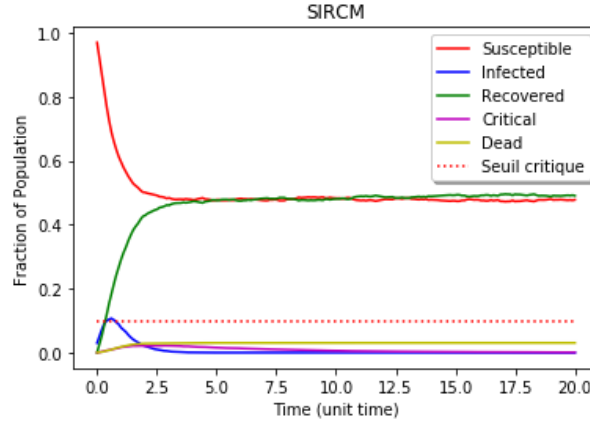


FIGURE 33 – Épidémie sans mesures restrictives ( $\beta = 1.5, \delta = 2, \nu = 0.016, \xi = 0.2, \eta = 0.2, \zeta = 0.2, \gamma = 0.2, \mu = 0.2$ )

Si on augmente la valeur de  $\delta$ , on retrouve bien que le nombre de personnes infectées est bien moindre, simplement car on guérit bien plus rapidement du virus.

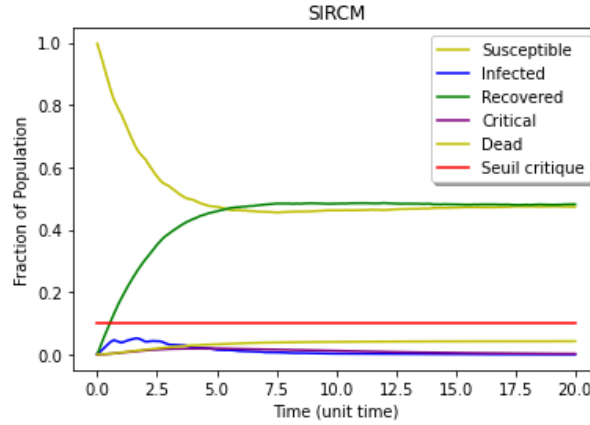


FIGURE 34 – Épidémie avec couvre-feu ( $\beta = 1.5, \delta = 0.8, \nu = 0.016, \xi = 0.2, \eta = 0.2, \zeta = 0.2, \gamma = 0.2, \mu = 0.2$ )

Observons à présent l'impact d'un couvre-feu sur la circulation du virus avec notre modèle. Ce dernier permet en effet de diminuer le nombre d'infectés, mais n'a pas d'impact sur le nombre de personne touchées par le virus. Le virus met aussi plus de temps à s'éteindre.

### 3.6.4 Améliorations possibles

Nous avons songé à des améliorations possibles de notre modèle. Nous n'avons pas pu les réaliser faute de temps mais nous souhaitons les présenter succinctement.

Tout d'abord, nous avons pensé à prendre en compte l'influence de l'âge dans notre modèle. Grâce au modèle compartimental, nous pouvons compartimenter la population en plusieurs catégories correspondant à des tranches d'âge qui auraient leurs propres taux de transitions, reflétant leur réaction face au virus. Par exemple, nous pourrions implémenter un taux de mortalité plus important pour les personnes de plus de 60 ans ou encore un taux de transmission du virus plus important pour les personnes entre 20 et 30 ans.

Nous avons aussi envisagé de considérer une population non constante, en ajoutant le nombre de naissances pendant la durée de l'épidémie. Il faudrait alors considérer cette nouvelle tranche de la population comme moins affectée et transmettant moins le virus.

Aussi, nous avons pensé à intégrer la vaccination de manière différente. Celle-ci serait disponible à la population au bout d'un certain temps, et donnerait une immunité totale et qui ne disparaîtrait pas au cours du temps.

Un élément important qu'il nous a semblé important de mettre en avant serait de considérer une relance épidémique une fois le virus disparu. Cela nous permettrait de modéliser de manière plus précise la situation d'un pays comme la France. En effet, après un confinement, même si le virus ne circule plus dans le pays, on peut imaginer qu'il peut être amené d'un pays voisin.

Finalement, un modèle avec cas-contact semble aussi intéressant. On se confine de soi-même quand on sait que l'on a été cas-contact d'une personne vraiment infectée, mais on peut tout de même à notre tour devenir aussi infecté si cette dernière personne nous a transmis le virus.

## Conclusion

Dans ce projet, nous nous étions donnés pour objectif d'analyser les modèles SIS et SIR stochastiques. Nous nous sommes pour cela basés sur GEMFSim, un simulateur stochastique pour la propagation d'une épidémie. Riche en fonctionnalités, nous avons décidé d'en implémenter d'autres : mise en place d'un confinement ou d'un couvre-feu par exemple.

Après avoir analysé en détail les paramètres de ce modèle, nous avons décidé de vérifier l'impact de ces mesures restrictives sur un petit panel de paramètres, qui nous permettaient d'avoir une idée précise de l'influence de ces dernières sur la propagation d'une épidémie. Par la suite, après avoir vérifié le comportement du nombre de reproduction de base, le  $R_0$ , nous avons souhaité vérifier la pertinence de notre modèle, en le comparant avec la résolution du système différentiel définissant le modèle SIR déterministe. Nous avons trouvé quelques différences, même si la lenteur du modèle déterministe en explique certaines. Enfin, nous avons profité de la simplicité d'utilisation de GEMFSim pour implémenter notre propre modèle, plus complexe que le modèle SIR, et nous avons donné quelques résultats sur ce dernier.

Nous souhaitions finalement de même faire une autocritique au niveau de la ligne directrice que nous avons emprunté pour le projet.

Tout d'abord, nous avons fait le choix d'utiliser un code déjà tout fait. Nous avons fait ce choix après nos recherches préliminaires, quand nous sommes tombés sur l'outil GEMFSim. Un tel choix peut être bénéfique, mais nous a aussi contraints quelquefois.

Tout d'abord, nous avons mis beaucoup de temps (sûrement trop) à comprendre le fonctionnement précis de ce simulateur. Il ne fait nul doute que si nous avions nous même codé un modèle, nous aurions certes mis un certain temps à l'implémenter, mais nous aurions été sûrs de ce que nous faisons. D'un autre côté, utiliser ce simulateur nous a permis d'apprendre de nouvelles choses, et une façon d'implémenter et de simuler un processus de Markov.

Ce temps perdu nous a peut-être empêché d'aller plus loin dans le projet, par exemple d'implémenter d'autres stratégies, ou encore d'implémenter un petit algorithme qui permet de relancer l'épidémie à un certain temps, en sélectionnant au hasard un noeud, et en l'infectant arbitrairement. À ce niveau-là, les temps de simulations (plusieurs jours non-stop) ont aussi pu nous handicaper, mais malheureusement indépendamment de notre volonté.

Nous avons aussi perdu beaucoup de temps au début de notre projet, car nous ne sommes pas donné une ligne directrice, un objectif précis, à atteindre. Ce reproche nous a d'ailleurs été fait par un autre groupe qui travaillait sur le même sujet. Nous donner dès les toutes premières séances un objectif nous aurait sûrement fait gagner du temps.

Au niveau plus mathématique, nous regrettons notamment deux choses :

- ne pas avoir réussi à déterminer l'équilibre du modèle SIS pour notre modèle, sûrement dû au fait que nous avons utilisé un code déjà fait, que nous ne maîtrisons pas parfaitement
- ne pas être allés plus loin dans la comparaison entre le modèle stochastique et le modèle déterministe. Nous avons réussi, malgré quelques différences, à le faire pour une situation sans mesure restrictive, mais pas dans le cas où ces dernières seraient mises en vigueur.

Finalement, nous retirons beaucoup de positif de ce projet. Nous avons pu apprendre beaucoup de nombreuses choses, qui se sont avérées très intéressantes, et donc nous sommes curieux quant à approfondir ces connaissances. Nous sommes sûrement passés à côté de certaines choses en utilisant un simulateur déjà implémenté, mais c'est un choix que nous assumons, et on ne peut de toute façon pas tout apprendre. Nous sommes aussi contents d'avoir pu implémenté des mesures restrictives, qui se sont avérées très fonctionnelles, et assez représentatives de la réalité.



## Références

- [1] F. Darabi Sahneh, C. Scoglio, and P. Van Mieghem, “Generalized epidemic mean-field model for spreading processes over multilayer complex networks,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1609–1620, 2013.
- [2] E. Bender and G. Williamson, *List, Decisions and Graphs with an Introduction to Probability*. University of California at San Diego, 2002.
- [3] J. L. Doob, “Topics in the theory of markoff chains,” *Transactions of the American Mathematical Society*, vol. 52, pp. 37–64, 1942.
- [4] —, “Markoff chains—denumerable case,” *Transactions of the American Mathematical Society*, vol. 58, no. 3, pp. 455–473, 1945. [Online]. Available : <http://www.jstor.org/stable/1990339>
- [5] D. T. Gillespie, “A general method for numerically simulating the stochastic time evolution of coupled chemical reactions,” *Journal of Computational Physics*, vol. 22, no. 4, pp. 403–434, 1976. [Online]. Available : <https://www.sciencedirect.com/science/article/pii/0021999176900413>
- [6] —, “Exact stochastic simulation of coupled chemical reactions,” *The Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 12 1977. [Online]. Available : <https://doi.org/10.1021/j100540a008>
- [7] F. D. Sahneh, A. Vajdi, H. Shakeri, F. Fan, and C. Scoglio, “Gemfsim : A stochastic simulator for the generalized epidemic modeling framework,” *Journal of Computational Science*, vol. 22, pp. 36–44, 2017. [Online]. Available : <https://www.sciencedirect.com/science/article/pii/S1877750317305227>
- [8] A. Joulin, *Méthodes de Monte-Carlo*. Insitut National des Sciences Appliquées - Toulouse, 2019. [Online]. Available : <https://perso.math.univ-toulouse.fr/joulin/files/2020/04/Monte-Carlo.pdf>
- [9] J. M. Heffernan, R. J. Smith, and L. M. Wahl, “Perspectives on the basic reproductive ratio,” *Journal of the Royal Society, Interface*, vol. 2, no. 4, pp. 281–293, 09 2005. [Online]. Available : <https://pubmed.ncbi.nlm.nih.gov/16849186>

## 4 Annexes

### 4.1 Minimum d'un échantillon de loi exponentielles indépendantes - 2.3.3

Soient  $X_1 \dots X_n$  des variables aléatoires indépendantes telles que

$$\forall i \in \llbracket 1; n \rrbracket, X_i \sim \mathcal{E}(\lambda_i) \text{ où } \lambda_i > 0$$

Posons  $Y = \min(X_1, \dots, X_n)$ , et montrons que

$$Y \sim \mathcal{E}\left(\sum_{i=1}^n \lambda_i\right)$$

Soit  $t > 0$  (le cas  $t \leq 0$  n'a pas d'intérêt, les  $X_i$  (et par construction  $Y$ ) étant à valeurs dans  $\mathbb{R}_+$ ), calculons  $\mathcal{P}(X_i > t)$ .

$$\mathcal{P}(X_i > t) = \int_t^{+\infty} \lambda_i e^{-\lambda_i t} dt = e^{-\lambda_i t}$$

Regardons maintenant  $\mathcal{P}(Y > t)$ .

$$\mathcal{P}(Y > t) = \mathcal{P}(X_1 > t \cap \dots \cap X_n > t)$$

Cependant, par indépendance des  $X_i$ , nous avons

$$\mathcal{P}(X_1 > t \cap \dots \cap X_n > t) = \prod_{i=1}^n \mathcal{P}(X_i > t) = \exp\left(-\sum_{i=1}^n \lambda_i t\right)$$

Ainsi, nous avons

$$\mathcal{P}(Y > t) = \exp\left(-\sum_{i=1}^n \lambda_i t\right)$$

On a donc trouvé la fonction de répartition de  $Y$ , notée  $F_Y(t)$  et telle que

$$F_Y(t) = \begin{cases} 0 & \text{si } t \leq 0 \\ 1 - \exp\left(-\sum_{i=1}^n \lambda_i t\right) & \text{si } t > 0 \end{cases}$$

On reconnaît la fonction de répartition d'une loi exponentielle de paramètre  $\sum_{i=1}^n \lambda_i$ , et la fonction de répartition caractérisant la loi, on en déduit que

$$Y \sim \mathcal{E}\left(\sum_{i=1}^n \lambda_i\right)$$

### 4.2 Détermination de $\Delta t$ - 2.4.5

Montrons que  $T \sim \mathcal{E}(R) \implies \frac{-\ln(U)}{R} \sim \mathcal{E}(R)$  avec  $U \sim \mathcal{U}[0, 1]$ .

Dans un premier temps, on a

$$U(\Omega) = [0, 1] \implies \ln(U)(\Omega) = ]-\infty, 0] \implies -\ln(U)(\Omega) = [0, +\infty[ \implies \frac{-\ln(U)}{R}(\Omega) = [0, +\infty[$$

On pose désormais  $T' = \frac{-\ln(U)}{R}$ , et nous cherchons à déterminer  $F_{T'}$  la fonction de répartition de  $T'$ .

$$\forall t \in \mathbb{R}, F_{T'}(t) = \begin{cases} 0 & \text{si } t < 0 \\ \mathcal{P}\left(\frac{-\ln(U)}{R} \leq t\right) & \text{si } t \geq 0 \end{cases}$$

Cependant,

$$\begin{aligned} & \mathcal{P}\left(\frac{-\ln(U)}{R} \leq t\right) \\ &= \mathcal{P}(-\ln(U) \leq Rt) \\ &= \mathcal{P}(\ln(U) \geq -Rt) \\ &= \mathcal{P}(U \geq \exp(-Rt)) \\ &= \mathcal{P}(U \leq 1 - \exp(-Rt)) \\ &= F_U(1 - \exp(-Rt)) \text{ avec } F_U \text{ la fonction de répartition d'une } \mathcal{U}[0, 1] \end{aligned}$$

Cependant,  $F_U(1 - \exp(-Rt)) = 1 - \exp(-Rt)$ . On déduit donc directement que

$$\forall t \in \mathbb{R}, F_{T'}(t) = \begin{cases} 0 & \text{si } t < 0 \\ 1 - \exp(-Rt) & \text{si } t \geq 0 \end{cases}$$

et donc  $T' = \frac{-\ln(U)}{R} \sim \mathcal{E}(R)$ .

Ainsi, on peut déterminer directement  $\Delta t$ , tel que  $\Delta t = \frac{-\ln(u)}{R}$ , avec  $u \in [0, 1]$ .

### 4.3 Compléments sur les modifications apportées au code

#### 4.3.1 Implémentation du confinement - 2.5.1

Comme nous l'avons expliqué précédemment, dans notre modèle, confiner revient à supprimer des liens entre les noeuds. L'implémentation de cette mesure restrictive devient assez aisée.

```
1 if compteur/len(X) > seuil_confinement:
2     graphe=nx.random_geometric_graph(N,dist=0.035,pos=pos)
3     print("Mise en place du confinement")
4     #Mise a jour des variables essentielles
```

Algorithme 6 – Implémentation du confinement

Précisons le rôle de quelques variables en premier lieu.

- *compteur* représente le nombre de noeuds infectés à chaque instant
- *len(X)* représente la population totale, i.e. le nombre de noeuds de notre graphe ( $N$ )

Dès lors, dès que la fraction de population dans un état  $I$  (le rapport *compteur*/*len(X)*) dépasse un certain seuil, nous mettons à jour notre graphe : les noeuds sont aux mêmes positions, nous changeons juste la distance maximale par rapport à laquelle les noeuds sont considérés comme en contact, et donc liés.

Une fois ce changement de graphe, nous mettons à jour toutes les variables nécessaires à la simulation.

Pour le déconfinement, le fonctionnement est très similaire. Dès que la fraction de population dans un état  $I$  devient inférieure à un certain seuil, nous changeons à nouveau la distance maximale par rapport à laquelle les noeuds sont considérés comme en contact, et nous mettons à jour les variables essentielles.

#### 4.3.2 Implémentation du couvre-feu - 2.5.2

Le principe de couvre-feu revient à restreindre sur une courte période de temps les relations entre les gens. Pour notre modèle, cela revient à supprimer des liens sur le graphe entre les noeuds

durant une courte période de temps, et de répéter ce processus sur l'intervalle  $[0, t_{max}]$  jusqu'à ce que l'épidémie s'éteigne.

Nous avons dans notre cas choisi de diviser une période de temps en 3 portions : 2 où la mesure de couvre-feu est inactive, et 1 dernière pendant laquelle le couvre-feu est en vigueur (cela représente par exemple un couvre-feu entre 22h et 6h).

Un moyen d'implémenter ceci est d'ajouter ces lignes de code à la fonction *GEMFSim*.

```

1 if Tf > curfew_index - 1/3:
2     print("Debut du couvre-feu")
3     graphe=nx.random_geometric_graph(N,0.032,pos=pos)
4     #Mise a jour des variables essentielles
5     while Tf < curfew_index and Tf > curfew_index - 1/3:
6         #Algorithme de simulation
7         graphe=nx.random_geometric_graph(N,0.051,pos=pos)
8         #Mise a jour des variables essentielles
9         curfew_index+=1
10    print("Fin du couvre-feu")

```

Algorithme 7 – Implémentation du couvre-feu

Rappelons tout d'abord que  $Tf$  représente la valeur actuelle du temps. Nous ajoutons une variable *curfew\_index*, qui représente le nombre de fois où le couvre-feu a été mis en place. Cette variable peut aussi s'interpréter comme la période de temps dans laquelle on est, et elle permet de déterminer le moment où il faut mettre en vigueur le couvre-feu. En effet, si le temps actuel  $Tf$  est plus grand que la période de temps à laquelle on se trouve -  $1/3$ , cela signifie que nous sommes dans l'intervalle où il faut déclencher le couvre-feu.

Dès lors, nous modifions le graphe en supprimant un certain nombre de liens et nous mettons à jour les variables essentielles, puis nous effectuons notre algorithme tant que

$$Tf \in [\text{curfew\_index} - \frac{1}{3}, \text{curfew\_index}]$$

Cette intervalle correspond en fait à la dernière fraction de notre période de temps que nous avons découpée en 3, soit la portion de temps où le couvre-feu est en vigueur.

Nous pouvons représenter ceci-visuellement, dans le cas où  $\text{curfew\_index} = t_1$ .



Dès lors que nous sortons de cet intervalle, le couvre-feu est levé, nous rétablissons alors le graphe originel, nous remettons à jour les valeurs essentielles, et enfin nous incrémentons *curfew\_index*, pour signifier qu'une période de temps a passé.

### 4.3.3 Implémentation du $R_0$ - 2.5.3

L'implémentation du calcul du  $R_0$  se fait très simplement. Dans un premier temps, nous définissons notre  $R_0$  comme une liste que l'on va incrémenter à chaque itération de notre fonction *GEMFSim*.

```

1 compteur = 0
2 for i in range(len(X)):
3     if X[i] == 0:
4         compteur+=1
5     if compteur==0:
6         R0.append(0)
7     else:

```

```
8 R0.append(compteur*beta/(N*delta))
```

#### Algorithme 8 – Calcul du R0

Expliquons désormais en détail le code précédent. On établit dans un premier temps la variable *compteur*, qui va contenir le nombre de noeuds dans un état *S*. Ensuite, on regarde pour chacun des noeuds leur état. Dès qu'un est dans l'état *S*, on incrémente *compteur* de 1.

Finalement, on rajoute dans notre liste le *R0* de l'instant considéré, défini comme la proportion de noeuds *S* multiplié par le rapport  $\frac{\beta}{\delta}$ .

#### 4.3.4 Implémentation du modèle SIRCM - 3.6.2

```
1 def Para_SIRCM(delta, beta, nu, xi, gamma, mu, eta, zeta):
2     M=5 #nombre d'etats : S, I, R, C, M
3     q= np.array([1])#definit l'etat considere comme infecte
4     L=len(q)
5     A_d=np.zeros((M,M)) #matrice des transitions node-based
6     A_d[0][2]=mu #taux de transition entre S et R
7     A_d[1][2]=delta # taux de transition entre I et R
8     A_d[1][3]=zeta #taux de transition entre I et C
9     A_d[1][4]=gamma #taux de transition entre I et M
10    A_d[2][0]=eta #taux de transition entre R et S
11    A_d[3][2]=xi #taux de transition entre C et R
12    A_d[3][4]=nu #taux de transition entre C et M
13    A_b = []
14    for l in range(L):
15        A_b.append(np.zeros((M,M))) #matrice des transitions edge-based
16    A_b[0][0][1]=beta
17    Para=[M,q,L,A_d,A_b]
18    return Para
```

#### Algorithme 9 – Para SIRCM

Détaillons le fonctionnement de cette méthode. Cette fonction prend en argument les huit taux de transition définis dans notre modèle SIRCM. Elle renvoie d'autre part, le nombre *M* d'états du modèle, les états influenceurs *q* et leur nombre *L*. Dans notre cas, comme expliqué auparavant, on ne considère qu'un état influenceur, à savoir l'état *I*. C'est en effet l'unique état pour lequel regarder le voisinage du noeud est important pour déterminer la transition entre l'état *S* (sain) et *I* (infecté). Par la suite on va créer les deux matrices de transitions node-based  $A_d$  et edge-based  $A_b$  qui vont être utilisées par l'algorithme GEMFsim pour déterminer l'évolution de l'épidémie. Dans la matrice node-based, de taille le nombre d'états du modèle, on insère les taux de transitions des différents passages. Il faut imaginer que par exemple le point  $[0][0]$  correspond à la transition entre les états *S* et *S*. A titre d'exemple, en listant les états de la manière suivante : *S, I, R, C, M* on peut remplir la matrice. Le taux de transition entre les états *S* et *R* correspond par exemple à la case  $[0][2]$  de la matrice  $A_d$ , la transition entre les deux états s'effectuant des états des lignes vers les états des colonnes. Une fois remplie notre matrice node-based, on remplit la matrice edge-based correspondant à celle des états influenceurs, donc ici la transition entre l'état *S* et l'état *I*. On insère donc le taux de transition  $\beta$  en position  $[0][1]$ .

### 4.4 Implémentation du modèle déterministe - 3.5

#### 4.4.1 Réflexion mathématique

Nous travaillons sur le modèle suivant.

$$\begin{aligned}\frac{dS}{dt} &= -aSI \\ \frac{dI}{dt} &= aSI - bI \\ \frac{dR}{dt} &= bI\end{aligned}$$

Ce modèle s'écrit facilement sous la forme d'un système différentiel autonome  $X'(t) = F(X(t))$  avec  $F(X)$  telle que

$$F = [-aSI, aSI - bI, bI] \text{ avec } X = [S, I, R]$$

On rappelle aussi la méthode de Runge-Kutta d'ordre 4, avec un pas de temps  $dt$ , qui pour un problème du type  $X'(t) = F(X(t))$  avec  $F(0) = X_0$ .

$$X_{n+1} = X_n + \frac{dt}{6}(X_n^1 + 2X_n^2 + 2X_n^3 + X_n^4)$$

avec

$$X_n^1 = F(X_n), X_n^2 = F\left(X_n + \frac{dt}{2}X_n^1\right), X_n^3 = F\left(X_n + \frac{dt}{2}X_n^2\right), X_n^4 = F(X_n + dtX_n^3)$$

#### 4.4.2 Implémentation informatique

```
1 def SIR(x,a,b):
2     return np.array([-a*x[0]*x[1], a*x[0]*x[1]-b*x[1], b*x[1]])
```

Algorithme 10 – Modèle SIR

```
1 def RK4(x,dt,f,a,b):
2     x1=f(x,a,b)
3     x2=f(x+(dt/2)*x1,a,b)
4     x3=f(x+(dt/2)*x2,a,b)
5     x4=f(x+dt*x3,a,b)
6     return x+(dt/6)*(x1+2*x2+2*x3+x4)
```

Algorithme 11 – Méthode RK4

```
1 def epidemie(f,x0,dt,Tfin,Tdeb,couvre_feu,confinement,deconfinement,seuil_conf,
2     seuil_deconf,a,b):
3     n=int((Tfin-Tdeb)/dt)
4     Xsol=np.zeros((n,3))
5     X=x0
6     i=0
7     while i<n-1:
8         Xsol[i]=X
9         X=RK4(X,dt,f,a,b)
10        i+=1
11        if couvre_feu and i>(1/3)/dt*curfew_index:
12            #Mise en place du couvre-feu
13            #Modification du taux de contact
14            while i>(1/3)/dt*curfew_index and i<curfew_index/dt:
15                #Algorithme de resolution
16                curfew_index+=1
17                #Modification du taux de contact
18            if confinement and Xsol[i-1,1] > seuil_conf:
19                #Mise en place du confinement
20                #Modification du taux de contact
21            if deconfinement and Xsol[i-1,1] < seuil_deconf:
22                #Fin du confinement
```

```

22         #Modification du taux de contact
23     return Xsol

```

### Algorithme 12 – Propagation de l'épidémie

Le fonctionnement est très simple.

La variable  $n$  représente le nombre d'itérations à réaliser. Si aucune mesure restrictive n'est en place, on effectue le schéma RK4 jusqu'au temps final. Si le confinement a besoin d'être mis en place (i.e. il y a plus d'infectés que le seuil de déclenchement du confinement), on met en place ce dernier en modifiant le taux de contact. Quand le confinement est levé, on modifie à nouveau ce taux de contact.

Pour le couvre-feu, à la manière de l'implémentation pour le couvre-feu du modèle stochastique, on modifie le taux de contact lorsque la variable  $i$ , représentant le temps actuel, est dans l'intervalle  $\left[ \frac{1}{3} \times \frac{1}{dt} \times curfew\_index, \frac{1}{dt} \times curfew\_index \right]$ <sup>10</sup>, où  $curfew\_index$  représente la période de temps actuelle. On peut résumer cela sur le schéma suivant, dans le cas où  $curfew\_index = t_1$ .

