

INSTITUT NATIONAL DES SCIENCES  
APPLIQUÉES



# Rapport du Projet Modélisation

---

Le modèle SIS et SIR stochastique

Hugo Julia et Xiaoya Wang  
Groupe B

25 mai 2021

# Sommaire

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Modèles déterministes</b>	<b>1</b>
2.1	Modèle SIS sans démographie . . . . .	1
2.2	Le modèle SIR . . . . .	3
2.2.1	Le modèle SIR sans démographie . . . . .	3
2.2.2	Le modèle SIR avec démographie . . . . .	5
<b>3</b>	<b>Modèles probabilistes</b>	<b>6</b>
3.1	Modélisation avec des chaînes de Markov . . . . .	7
3.1.1	Modèle SIS . . . . .	7
3.1.2	Modèle SIR . . . . .	9
3.1.3	Comportement des modèles avec chaîne de Markov en temps long	10
3.2	Processus de branchement . . . . .	11
3.2.1	Introduction . . . . .	11
3.2.2	Etude théorique de la probabilité d’extinction . . . . .	12
3.2.3	Simulations et estimations de la probabilité d’extinction . . . .	14
<b>4</b>	<b>Conclusion</b>	<b>16</b>
<b>5</b>	<b>Annexes</b>	<b>18</b>
5.1	Preuve 1 . . . . .	18
5.2	Preuve 2 . . . . .	19
5.3	Code PYTHON 1 . . . . .	21
5.4	Code PYTHON 2 . . . . .	24
5.5	Code PYTHON 3 . . . . .	25
5.6	Code PYTHON 4 . . . . .	26
5.7	Code PYTHON 5 . . . . .	28
5.8	Code R 6 . . . . .	33

# 1 Introduction

Les épidémies, causées par des virus ou par des bactéries, sont connues par un accroissement anormal du nombre de cas d'une maladie infectieuse, dans une région donnée ou au sein d'une population. Leur histoire est aussi ancienne que celle de l'humanité, à savoir que la plus ancienne épidémie a eu lieu de 430 à 426 avant JC, qui a tué plus d'un quart de la population grecque.

Pour mieux connaître et ensuite contrôler une épidémie, on peut mettre en œuvre des modèles mathématiques pour prévoir son évolution. Dans ce projet de modélisation, nous allons modéliser l'évolution d'une épidémie dans une population donnée, nous nous concentrerons sur deux modèles en particulier : SIS et SIR.

Nous allons d'abord commencer par analyser les modèles déterministes, où nous obtenons une trajectoire de l'épidémie avec un jeu de paramètres et conditions initiales fixes. Puis nous verrons différentes manières de modéliser les épidémies en prenant en compte la nature aléatoire de leur propagation.

## 2 Modèles déterministes

Les modèles déterministes reposent sur des systèmes d'équations différentielles dont la programmation et la résolution sont devenues aisées grâce à l'informatique. Ils permettent de disposer très rapidement de résultats. Ils présentent cependant l'inconvénient de ne pas prendre en compte des événements aléatoires inéluctables lors des démarrages épidémiques, comme les événements à très haut potentiel de contagion (super-spread events). Nous traiterons ces cas dans une deuxième partie.

### 2.1 Modèle SIS sans démographie

Les modèles SIS (susceptible-infectious-susceptible) sont souvent utilisés pour décrire les maladies infectieuses, en particulier les infections sexuellement transmissibles, par exemple comme la gonorrhée. Pour ces maladies, les individus ne peuvent pas être immunisés et donc un individu peut être infecté plusieurs fois au cours de sa vie.

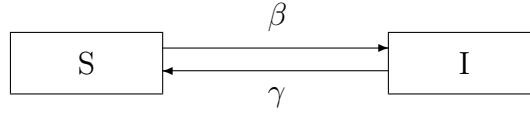
Dans cette partie on s'intéresse au modèle SIS sans démographie, les naissances et les décès naturels de la population ont été ignorés et on considère les fonctions suivantes :

- $S$  : la proportion des individus susceptibles, un individu susceptible peut devenir malade après contacter les personnes infectées.
- $I$  : la proportion des individus infectés, un individu infecté peut transmettre l'infection à d'autres individus susceptibles.

Ces fonctions sont reliées dans les équations différentielles par les paramètres suivants :

- $\beta$  : taux d'infection, une probabilité individuelle d'être infecté au contact d'une autre personne infectée.

- $\gamma$  : taux de récupération, on s'intéresse souvent au terme  $\frac{1}{\gamma}$ , qui représente la période infectieuse en moyenne.



Le modèle SIS se représente sous la forme de deux équations différentielles ordinaires :

### Equations

$$\begin{aligned}\frac{dS}{dt} &= -\beta IS + \gamma I \\ \frac{dI}{dt} &= \beta IS - \gamma I\end{aligned}$$

Si on regarde la première équation, on constate que le terme  $IS$  signifie qu'un malade rencontre un susceptible, et le contamine avec un taux  $\beta$ , d'où l'augmentation du nombre de cas contaminés de  $\beta IS$ , et la diminution du nombre de susceptibles, où on retrouve le même terme avec un signe négatif. Le deuxième terme  $\gamma I$  représente les personnes infectées qui guérissent puis reviennent aux susceptibles, ce qui fait croître le nombre des susceptibles et diminuer le nombre d'infectés.

Pour que l'épidémie s'arrête, il faut que le nombre d'infectés diminue, donc sa dérivée  $\frac{dI}{dt}$  doit être négative. En résolvant  $\beta IS - \gamma I < 0$ , on obtient une condition limite :  $\frac{\beta S(0)}{\gamma} = 1$ , et on introduit donc le nombre de reproduction de base  $R_0 = \frac{\beta}{\gamma}$  (avec la population initiale de susceptibles égale à 1), qui est un élément important pour déterminer si l'épidémie va s'arrêter. Il représente le nombre moyen de personnes qu'une personne contagieuse peut infecter.

On voit 2 cas émerger selon la valeur de  $R_0$  :

- Si  $R_0 > 1$ , le nombre de susceptibles diminue tant que le nombre d'infectés continue à augmenter, l'épidémie va continuer à se propager. Plus  $R_0$  est grand, moins il faut de temps pour stabiliser la courbe.
- Si  $R_0 \leq 1$ , l'épidémie va s'éteindre, le nombre d'infectés  $I$  reste à 0 en temps infini.

Comme on a une équation différentielle avec la condition initiale  $I(t=0)$  prédéfinie, cela se transforme en problème de Cauchy, donc la solution du problème peut être approchée par différentes méthodes de résolution numériques connues, notamment celles de Runge-Kutta.

Nous l'avons modélisé sur Python, avec les paramètres :  $\beta = 1.4247$ ,  $\gamma = 0.14286$

et  $R_0 = 9.97$ , puis dans un deuxième temps :  $\beta = 0.78$ ,  $\gamma = 1$  et  $R_0 = 0.78$ . Voici les graphes que nous avons obtenu :

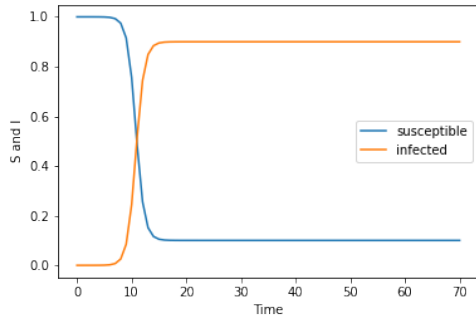


FIGURE 1 – Résolution numérique des équations,  $R_0 = 9.97 > 1$

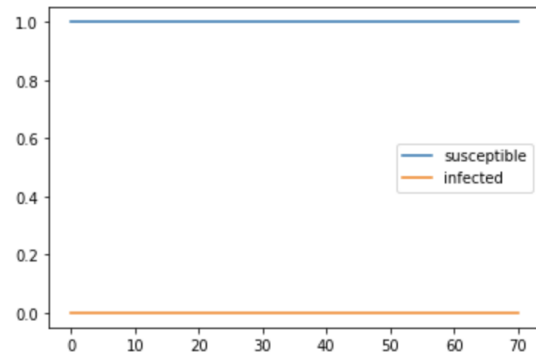


FIGURE 2 – Résolution numérique des équations,  $R_0 = 0.78 < 1$

*Le code correspondant est en annexe rubrique "Code PYTHON 3".*

Nos hypothèses sur l'influence de  $R_0$  sont bien confirmées pour la résolution numérique.

## 2.2 Le modèle SIR

Dans ce modèle, on divise la population en 3 catégories :

- Les Susceptibles (S), qui n'ont jamais été en contact avec la maladie considérée
- Les Infectés (I), qui portent la maladie et sont capables de la transmettre
- Les individus qui ne transmettent plus la maladie (R)

Ce modèle est surtout adapté pour modéliser des maladies qui confèrent une immunité définitive ou très longue aux malades qui ont guéris. On va séparer deux cas : le modèle SIR sans démographie et avec démographie.

### 2.2.1 Le modèle SIR sans démographie

Pour introduire le modèle, il est plus simple de considérer une population "fermée", c'est à dire qui reste constante, sans considérer les morts naturelles, les naissances, ou les migrations. En terme de réalité physique, on peut imaginer l'introduction d'une maladie dans une population dont les membres n'ont de contact qu'entre eux et de manière à ce que l'épidémie résultante soit assez rapide pour ne pas à avoir à considérer les processus démographiques naturels.

On peut alors imaginer les scénarios suivants, qui grâce à l'introduction de paramètres intrinsèques à l'épidémie nous permettent de construire les équations différentielles représentant la dynamique du problème :

- Un individu infecté (I) contamine un individu susceptible (S) au taux  $\beta$ , qui correspond au nombre moyen de contact par personne par unité de temps. Dans d'autres termes, lors d'une rencontre entre un susceptible et un infecté, on a une chance  $\beta$  que le susceptible soit contaminé. Cela explique donc le terme  $\beta IS$  qui a un signe positif dans l'équation régissant I et un signe négatif dans l'équation régissant S.
- Un individu infecté (I) guérit ou meurt au taux  $\gamma$ , l'inverse du temps moyen de guérison. Il rejoint alors la population R (Removed en anglais), d'où le terme  $\gamma I$  dans les équations de I et R.
- La population reste constante :  $S + I + R = 1$  à tout instant.



On obtient alors les équations suivantes :

### Equations

$$\frac{dS}{dt} = -\beta IS$$

$$\frac{dI}{dt} = \beta IS - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

On ne peut pas trouver de solution exacte à ce problème mais on peut l'approcher par des méthodes d'intégration numérique des équations différentielles comme Euler ou Runge Kutta, comme on l'a fait dans l'étude du modèle SIS.

Pour toute maladie, un enjeu sanitaire majeur est de savoir si elle se propage dans la population et à quelle vitesse. L'épidémie se propage si  $\frac{dI}{dt} > 0 \Leftrightarrow I(\beta S - \gamma) > 0 \Leftrightarrow \frac{\beta S}{\gamma} > 1$ . En supposant que tout le monde est susceptible d'être contaminé au temps initial, ( $S(0) = 1$ ), on voit alors que l'épidémie se propage si et seulement si au temps 0,  $\frac{\beta}{\gamma} > 1 \Leftrightarrow R_0 > 1$ .

Ce  $R_0 = \beta/\gamma$  que l'on a déjà introduit pour le modèle SIS représente en fait le nombre moyen d'individus contaminés par contact avec un infecté. Il est donc logique que l'épidémie se propage seulement si ce nombre est supérieur à un.

Nous avons tracé les dynamiques du modèle pour plusieurs valeurs de  $\gamma$  et  $\beta$  et donc de  $R_0$ . Sur le premier graphe on voit les courbes pour  $\beta = 1.4247$ ,  $\gamma = 0.14286$ ,  $R_0 = 9.97$ , des valeurs proches de celles de la varicelle. Sur le deuxième,  $\beta = 0.78$  et  $\gamma = 1$ , donc  $R_0 = 0.78$ .

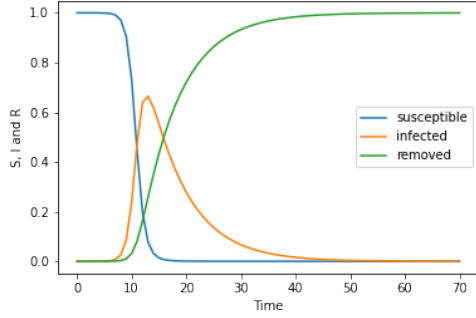


FIGURE 3 – Résolution numérique des équations,  $R_0 = 9.97 > 1$

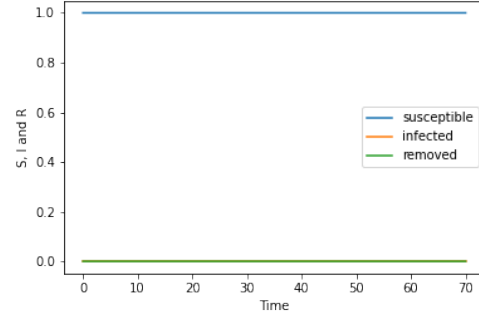


FIGURE 4 – Résolution numérique des équations,  $R_0 = 0.78 < 1$

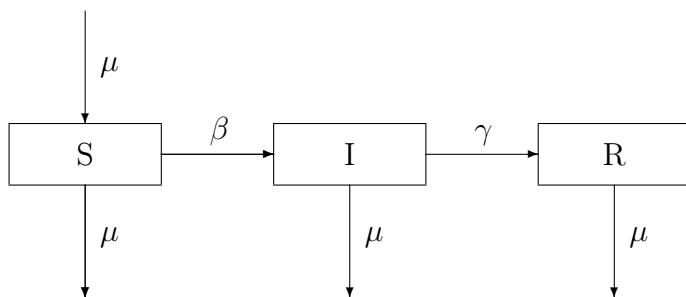
*Le code correspondant est en annexe rubrique "Code PYTHON 1".*

Ici, on voit bien que avec des valeurs de  $R_0 < 1$ , l'épidémie ne se propage pas, et dans le cas inverse, la dynamique des susceptibles et des infectés est similaire à la modélisation SIS au début, puis lorsque la majorité de la population commence à gagner l'immunité, les infectés baissent fortement et on arrive à un état d'immunité collective.

### 2.2.2 Le modèle SIR avec démographie

Dans le modèle précédent, on avait supposé que la population considérée était fermée et que la propagation de l'épidémie était assez rapide pour ne pas avoir à considérer les morts et les naissances naturelles. Pour pouvoir appliquer notre modèle à plus de situations réelles, on introduit un nouveau paramètre  $\mu$ , qui correspond au taux de mortalité naturelle, l'inverse de la durée de vie moyenne des individus de la population. Pour garantir que la population totale reste aussi constante, on suppose que  $\mu$  est aussi le taux de natalité naturel de la population considérée.

Ainsi, on a des nouveaux individus qui rentrent dans la population saine au taux  $\mu$ , et des individus qui meurent naturellement peu importe si ils étaient sains (S), infectés (I), ou guéris (R).



On a donc les nouvelles équations :

### Equations

$$\frac{dS}{dt} = \mu - \beta IS - \mu S$$

$$\frac{dI}{dt} = \beta IS - \gamma I - \mu I$$

$$\frac{dR}{dt} = \gamma I - \mu R$$

On voit l'apparition de termes en  $\mu$  dans toutes les populations car peu importe l'état de santé d'un individu, il peut mourir d'une autre cause au taux  $\mu$ , et on a  $\mu$  nouveaux nés qui sont susceptibles.

On peut appliquer les mêmes méthodes de résolution numériques à ces équations et on trouve des résultats similaires. Cependant, l'expression de  $R_0$  va changer. Le nouveau R-effectif est alors défini comme  $R_0 = \frac{\beta}{\gamma + \mu}$ . Il est logique que le  $R_0$  ait diminué car le fait de considérer les morts naturelles diminue la probabilité d'infection moyen par chaque personne infectée.

**CONCLUSION :** Considérons une maladie qui a un taux de transmission  $\beta$  et un taux de guérison  $\gamma$ . Dans une population fermée et sans considérer le renouvellement naturel de la population, l'épidémie va se répandre seulement si  $R_0 = \beta/\gamma > 1$ . Si on considère une population au taux de renouvellement  $\mu$ , l'épidémie va se répandre seulement si  $R_0 = \frac{\beta}{\gamma + \mu} > 1$ .

## 3 Modèles probabilistes

Tous les modèles considérés jusqu'à présent ont été déterministes, qui se basaient sur l'hypothèse d'une population de grande taille. Étant donné les mêmes conditions



initiales, la même trajectoire est toujours observée. Mais en réalité, on ne peut pas voir exactement les mêmes personnes infectées exactement au même moment.

Donc maintenant, on s'intéresse aux cas aléatoires, qui sont appliqués lorsque la taille de la population est petite, soit au début de l'épidémie, soit après que les mesure de contrôle ont réussies etc... Dans ces cas là, il est particulièrement important que la stochasticité soit prise en compte et intégrée dans les modèles.

### 3.1 Modélisation avec des chaînes de Markov

#### 3.1.1 Modèle SIS

Dans cette partie, on va modéliser les compartiments S et I comme des chaînes de Markov.

On note  $S_n$  le nombre d'individus susceptibles,  $I_n$  le nombre d'individus infectés au temps  $t_n = n\Delta t$  où  $\Delta t > 0$  est le pas.

Comme nous l'avons supposé dans les cas déterministes, la population  $N$  reste constante et  $N = S_n + I_n$ , on pourra donc considérer seulement le nombre d'individus infectés  $I_n$  comme une chaîne de Markov.

On suppose que le pas  $\Delta t$  est suffisamment petit, de cette manière il y aura un seul changement d'état dans chaque intervalle de temps : un seul individu passe de susceptible à infecté, ou inversement, ou rien ne se passe.

Le graphe ci-dessous représente la chaîne de Markov construite :

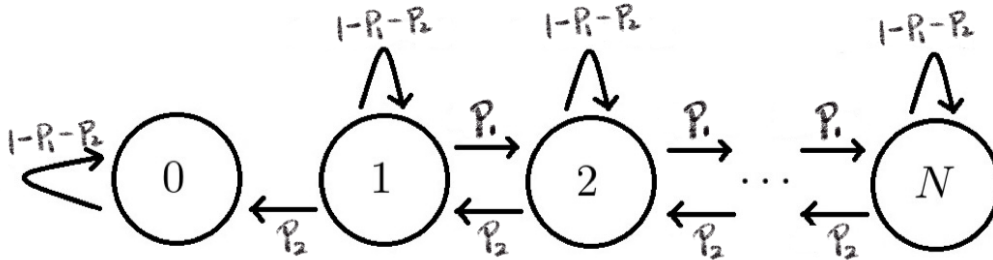


FIGURE 5 – Présentation graphique de la chaîne de Markov

On construit notre matrice de transition avec les probabilités suivantes :

<b>Probabilités</b>
$P(I_{n+1} = i + 1 \mid I_n = i) = P_1 = \frac{\beta i(N - i)}{N} \Delta t$
$P(I_{n+1} = i - 1 \mid I_n = i) = P_2 = (b + \gamma) i \Delta t$
$P(I_{n+1} = i \mid I_n = i) = 1 - P_1 - P_2 = 1 - \left( \frac{\beta i(N - i)}{N} - (b + \gamma) i \right) \Delta t$

- $P_1$  : chaque personne infectée est en contact avec une personne saine à vitesse  $\beta$ , alors cette personne saine devient contaminée, le nombre de infecté augmente,  $I_{n+1} \rightarrow I_n + 1$ .
- $P_2$  : une personne infectée meurt à taux  $b$  ou guérit à taux  $\gamma$ , le nombre de infecté diminue,  $I_{n+1} \rightarrow I_n - 1$ .
- $1 - P_1 - P_2$  : le nombre d'infecté ne change pas,  $I_{n+1} \rightarrow I_n$ .

On peut vérifier que la chaîne de Markov est récurrente apériodique et positive, et que l'état  $I_n = 0$  est absorbant alors que tous les autres sont transitoires. Par propriété des chaînes de Markov, on a alors que  $\lim_{n \rightarrow +\infty} I_n = 0$ . Cependant, on va vouloir savoir au bout de combien de temps le nombre d'infectés va être égal à 0.

Voici le graphe qu'on obtient en simulant cette chaîne de Markov sur Python, avec les paramètres suivants :  $\beta=0.25$ ,  $\gamma=0.1$ ,  $b = 0.25$  et  $R_0 < 1$  ;  $\beta=0.3$ ,  $\gamma=0.1$ ,  $b = 0.2$  et  $R_0 = 1$  ;  $\beta=0.5$ ,  $\gamma=0.1$ ,  $b = 0.2$  et  $R_0 > 1$ .

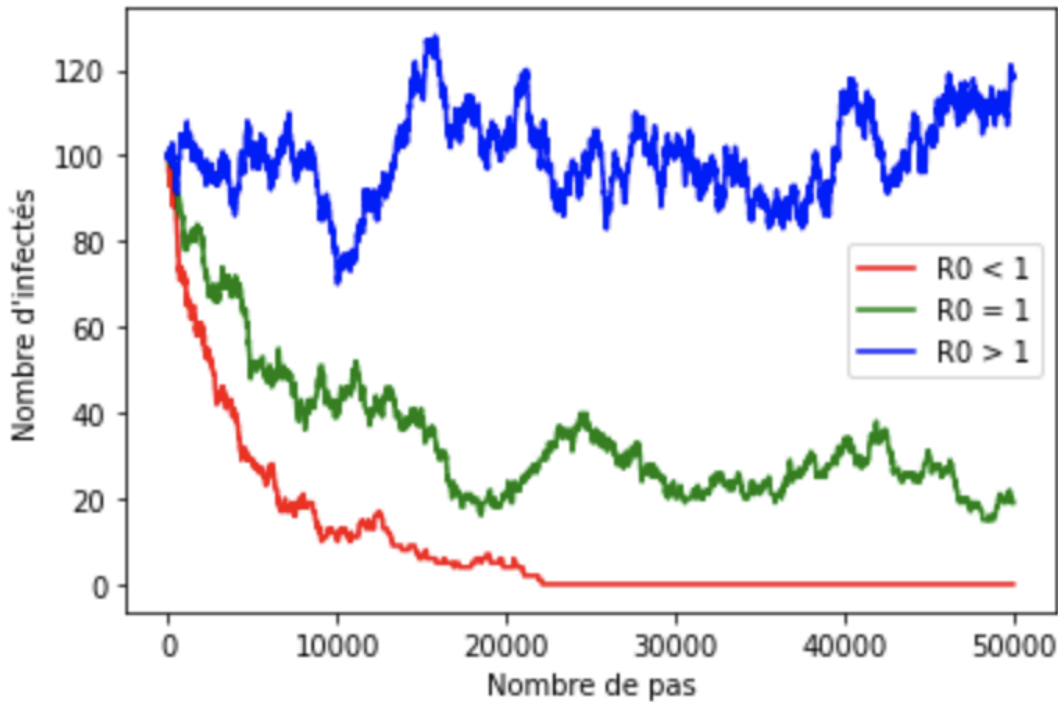


FIGURE 6 – Simulation de la chaîne de Markov avec plusieurs valeurs de  $R_0$

*Le code correspondant est en annexe rubrique "Code PYTHON 4".*

On constate que le temps de retour en 0 varie selon la valeur de  $R_0$  :

- Avec  $R_0 < 1$ , l'épidémie s'éteint comme analysé précédemment, avec un temps de retour en 0 petit.
- Avec  $R_0 = 1$ , l'épidémie s'éteint avec un temps de retour en 0 plus grand.

- Avec  $R_0 > 1$ , l'épidémie ne semble pas s'éteindre, et le temps de retour en 0 est très grand.

On peut alors se poser la question du temps de retour en temps long, si  $R_0 > 1$ , quand-est-ce que l'épidémie va s'éteindre ? Nous reviendrons sur ce point plus tard.

### 3.1.2 Modèle SIR

Comme on l'a fait pour le modèle SIS, on peut aussi reprendre le modèle SIR et le modéliser en utilisant un processus aléatoire. On partage la population en trois compartiments : S, I et R comme on l'a fait quand on a établi la version déterministe de ce modèle. On se met dans le cas d'une épidémie au taux de transmission  $\beta$ , taux de guérison  $\gamma$  se propageant au sein d'une population de  $N$  individus au taux de natalité/mortalité  $\mu$ . Comme on sait que  $R = N - S - I$ , on peut se contenter de s'intéresser qu'au couple  $(S, I)$ .

Comme on l'a fait pour le modèle SIS, on discrétise le temps de manière à ce que  $t \in \{0, \Delta t, 2\Delta t, \dots\}$ , avec  $\Delta t$  assez petit afin que durant une unité de temps, il n'y ait qu'un changement d'état. A chaque étape, on note le nombre d'individus dans chaque compartiment  $S_n = S(n\Delta t)$  et  $I_n = I(n\Delta t)$ .

Alors,  $Z_n = (S_n, I_n)$  est une chaîne de Markov discrète dont on va pouvoir étudier les propriétés.

La première chose à trouver sont les probabilités de transition afin de pouvoir construire la matrice de transition de la chaîne. Avec la manière dont on a discrétisé le temps, on a seulement 6 possibilités et la matrice de transition va être assez creuse. En couvrant tous les cas possibles, on arrive aux probabilités de transition suivantes :

#### Probabilités

$$\begin{aligned}
 P((S_n - 1, I_n + 1) | (S_n, I_n)) &= \frac{\beta S_n I_n}{N \Delta t} \\
 P((S_n, I_n - 1) | (S_n, I_n)) &= \gamma I_n \Delta t \\
 P((S_n + 1, I_n - 1) | (S_n, I_n)) &= \mu I_n \Delta t \\
 P((S_n + 1, I_n) | (S_n, I_n)) &= \mu(N - S_n - I_n) \Delta t \\
 P((S_n, I_n) | (S_n, I_n)) &= 1 - \frac{\beta S_n I_n}{N \Delta t} - \gamma I_n \Delta t - \mu I_n \Delta t - \mu(N - S_n - I_n) \Delta t
 \end{aligned}$$

On peut vérifier que la chaîne de Markov est récurrente apériodique et positive, et que l'état  $(N, 0)$  est absorbant alors que tous les autres sont transitoires. Par propriété des chaînes de Markov, on a alors que  $\lim_{n \rightarrow +\infty} (S_n, I_n) = (N, 0)$ .

Ce qui nous intéresse pour modéliser la propagation de l'épidémie est l'instant auquel est atteint cet état absorbant, défini comme  $\tau_A = \inf(n \geq 0, (S_n, I_n) = (N, 0) = A)$ . En simulant la chaîne de Markov pour différentes valeurs de  $R_0$ , on voit là aussi clairement que le temps d'atteinte de la probabilité invariante et le  $R_0$  sont fortement corrélés.

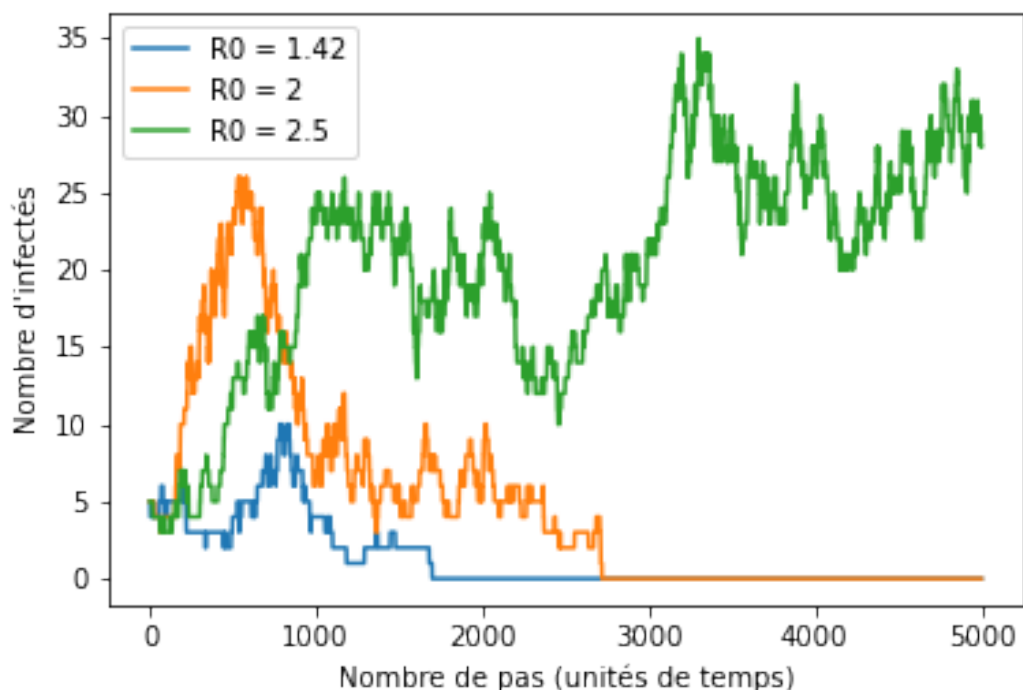


FIGURE 7 – Simulation de la chaîne de Markov avec plusieurs valeurs de  $R_0$

*Le code correspondant est en annexe rubrique "Code PYTHON 5".*

On va donc étudier en particulier le comportement des chaînes de Markov de ce type, c'est à dire qui représente des populations vouées à disparaître presque sûrement.

### 3.1.3 Comportement des modèles avec chaîne de Markov en temps long

Si on résume ce qu'on a vu dans la modélisation que l'on a fait des épidémies grâce à des chaînes de Markov, on peut retenir que l'état où le nombre d'infectés est égal à 0 est absorbant. Ceci paraît logique car si il n'y a plus de malades, il n'y a plus de transmission donc l'épidémie s'éteint.

On sait aussi que d'après la théorie des chaînes de Markov, si la chaîne considérée converge, ce serait vers cet état absorbant. Cependant, on a vu que dans certains cas dépendant du paramètre choisi, l'épidémie paraît se maintenir dans la population. Pour expliquer cela, on introduit la distribution quasi stationnaire :

### Définition

On considère une chaîne de Markov  $Z$  sur  $\mathbb{Z}$  qui représente une population isolée, c'est à dire que l'état 0 est un état absorbant, on le note  $A$ . On note  $\tau_A = \inf(n \geq 0, Z = A)$  le temps de retour en cet état d'extinction.

Soit  $\alpha$  une mesure de probabilité sur  $\mathbb{Z}$ . On dit que  $\alpha$  est une distribution quasi stationnaire (où Quasi-Stationary Distribution en anglais, d'où l'acronyme QSD) si pour tout  $t \geq 0$ , et tout espace mesurable  $A \subset \mathbb{N}$  :

$$\alpha(A) = \mathbb{P}_\alpha(Z \in A | \tau_A > t)$$

Cette QSD nous permet d'avoir une idée du temps d'extinction grâce au résultat suivant :

### Résultat

Si on considère une chaîne de Markov  $Z$  qui possède les mêmes propriétés que dans la définition précédente, et qu'on considère qu'elle admet une QSD  $\alpha$ . Alors, il existe un nombre positif  $\theta(\alpha)$  qui dépend de la QSD tel que :

$$\mathbb{P}_\alpha(\tau_A > t) = e^{-\theta(\alpha)t}$$

Ce résultat nous montre que en connaissant la QSD du processus de Markov qui modélise notre épidémie, on peut trouver le paramètre de la loi exponentielle que suit le temps d'extinction.

## 3.2 Processus de branchement

### 3.2.1 Introduction

Dans la partie précédente, on avait modélisé la population entière en la séparant en compartiments selon l'état de santé de l'individu en question. Les processus de branchements sont aussi basés sur les individus (les transitions de chaque individus sont données) et stochastiques mais modélisent directement le processus de  $I$  individus, sans analyser le mécanisme d'exposition/infection.

Les processus de branchement ont été initié au dix-neuvième siècle par Bienaymé et ensuite par Galton-Watson pour étudier l'extinction des noms de famille. On peut reprendre ces idées pour étudier l'extinction d'une épidémie. On s'intéresse à des individus qui se reproduisent (dans notre cas, reproduction correspond à infection). La population initiale est appelée génération 0. Leurs enfants correspondent à la génération 1 et on définit de même les générations suivantes.

On note  $Z_n$  la taille de la  $n$ -ième génération, autrement dit le nombre d'infectés à l'instant  $n$ . Naturellement, ce nombre va dépendre de  $X_{i,n}$  le nombre de cas secondaires générés à l'instant  $n$  par l'individu  $i$ . Les  $\{X_{i,n}\}_{(n,i) \in \mathbb{N}^2}$  sont supposés indépendants et identiquement distribués car les individus d'une même génération n'interfèrent pas entre eux et à l'instant  $n$ , on suppose que la loi qui gouverne les générations suivantes ne

dépend pas du passé. Cette loi, que l'on appelle  $\mu$ , est suivie par tous les  $\{X_{i,n}\}_{(n,i) \in \mathbb{R}^2}$  et est donc identique pour chaque individu et à tout instant ce qui garantit la propriété de similarité des processus de Galton-Watson qui garantit que tout sous-arbre est identique à l'arbre principal.

Sachant tout cela, on peut définir le processus de Galton-Watson de loi de reproduction  $\mu$  de la manière suivante :

**Définition**

$$Z_0 = 1$$

$$Z_{n+1} = \sum_{i=1}^{Z_n} X_{n,i}$$

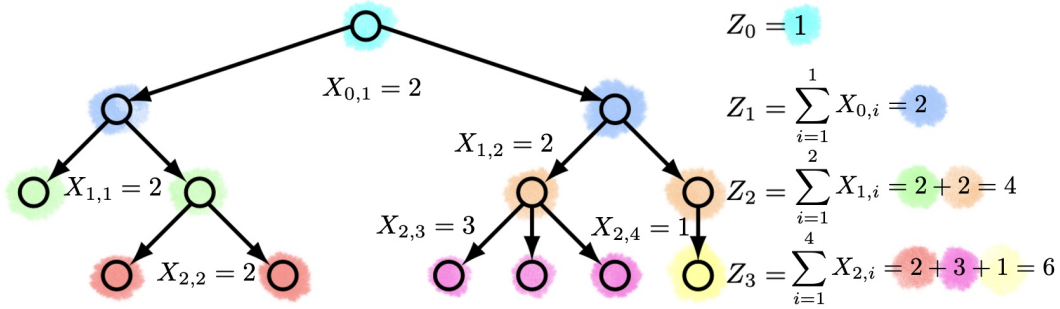


FIGURE 8 – Représentation graphique d'un exemple simple de processus de Galton-Watson

### 3.2.2 Etude théorique de la probabilité d'extinction

La question au centre de la modélisation avec des processus de branchement de type Galton-Watson est alors la suivante : quelle est la probabilité que l'espèce s'éteigne ? Ou encore, dans notre contexte, quelle est la probabilité que l'épidémie disparaisse de la population ?

Le comportement des modèles de type processus de Galton-Watson ont été étudiées longuement depuis leur apparition et on sait que deux cas se distinguent : les trajectoires du processus s'éteignent ou alors explosent exponentiellement :

$$\mathbb{P}(\left\{ \lim_{n \rightarrow \infty} Z_n = \infty \right\} \cup \left\{ \lim_{n \rightarrow \infty} Z_n = 0 \right\}) = 1$$

Pour savoir quand l'épidémie modélisée va disparaître, on s'intéresse particulièrement à la probabilité d'extinction  $\mathbb{P}_{ext} = \mathbb{P}(\lim_{n \rightarrow \infty} Z_n = 0)$ . On va donc maintenant essayer de voir comment on peut trouver la valeur de cette probabilité et les paramètres qui peuvent influencer dessus.

- **Fonction génératrice :** Le premier élément qu'il faut définir pour étudier la probabilité d'extinction est la fonction génératrice d'une variable aléatoire. On verra par la suite ce qu'elle représente dans le cadre d'un processus de branchement.

### Définition

Pour  $0 \leq s \leq 1$ , on définit la fonction génératrice d'une variable aléatoire  $X$  par

$$G(s) = \mathbb{E}[s^X]$$

Grâce à des propriétés sur les séries, il est possible de montrer que :

- $G$  est bien définie sur  $[0,1]$  et  $y$  est de classe  $C^1$ .
- $G$  est strictement croissante et convexe sur  $]0,1[$ .

Par définition, cette fonction caractérise la loi de  $X$ , ce qui signifie que si  $Y$  et  $Z$  sont deux variables aléatoires et que  $G_Z(s) = G_Y(s)$ ,  $\forall z \in \mathbb{R}$ , alors  $Y$  et  $Z$  ont la même loi.

De plus, on peut remarquer que  $G(s) = \mathbb{E}[s^X] = \sum_{k=1}^{+\infty} s^k \mathbb{P}(X = k)$  donc  $G(0) = \mathbb{P}(X = 0)$ .

De même, on peut calculer  $G'(s) = \sum_{k=1}^{+\infty} k s^{k-1} \mathbb{P}(X = k)$  donc  $G'(0) = \mathbb{P}(X = 1)$  et  $G'(1) = \sum_{k=1}^{+\infty} k \mathbb{P}(X = k) = \mathbb{E}(X)$

- **Fonction génératrice de  $Z_n$  :** Pour notre processus de branchement, on note  $G_n = G_{Z_n}$  la fonction génératrice de  $Z_n$ , donc  $G_n(s) = \mathbb{E}[s^{Z_n}] = \sum_{k=1}^{+\infty} s^k \mathbb{P}(Z_n = k)$  et  $G$  la fonction génératrice des  $\{X_{i,n}\}_{(n,i) \in \mathbb{R}^2}$ , qui ont tous la même loi car ils ont tous la même loi. Alors, on a :

### Résultat

$$\forall n \in \mathbb{N}^*, G_n = \underbrace{G \circ \dots \circ G}_{n \text{ fois}}(z).$$

La preuve de ce résultat se trouve en annexe en première position.

- **Convergence vers la probabilité d'extinction :** Maintenant que l'on sait cela, on remarque que  $\mathbb{P}_{ext} = \mathbb{P}(\lim_{n \rightarrow \infty} Z_n = 0) = \mathbb{P}(\exists n \in \mathbb{N}, Z_n = 0) = \lim_{n \rightarrow \infty} \mathbb{P}(Z_n = 0) = \lim_{n \rightarrow \infty} G_n(0)$ . Cette limite est la limite d'une suite récurrente, vu qu'on a montré que  $G_n = \underbrace{G \circ \dots \circ G}_{n \text{ fois}}(z)$ , et donc on sait que  $\mathbb{P}_{ext}$  est le plus petit point fixe de  $G$  sur  $]0,1[$ .

On en arrive donc au résultat principal qui permet de conclure sur le sort de l'épidémie dans le temps :

### Théorème

Soit un processus de branchement défini par  $Z_n = \sum_{i=1}^{Z_{n-1}} X_{i,n}$ , avec  $\{X_{i,n}\}_{(n,i) \in \mathbb{R}^2} \sim \mu \sim \mathcal{L}(m, \sigma^2)$ . Alors deux cas se distinguent :

- Si  $m > 1$ , alors  $\mathbb{P}_{ext} < 1$ , on dit alors que le processus est **sur-critique**.
- Si  $m \leq 1$ , alors  $\mathbb{P}_{ext} = 1$ , on dit alors que le processus est **sous-critique** si l'inégalité est stricte, ou **critique** si  $m = 1$ .

La preuve de ce théorème est la deuxième annexe.

### 3.2.3 Simulations et estimations de la probabilité d'extinction

On voit donc que le paramètre principal qui décrit l'évolution de l'épidémie est  $m = \mathbb{E}[\mu]$ . Il correspond au nombre moyen de descendants (donc d'infections) par individu. Il joue un rôle analogue à celui de  $R_0$ .

Pour mettre en valeur l'influence de ce paramètre, on peut modéliser un processus de branchement (sur  $\mathbb{R}$ ) pour différentes valeurs de  $m$ . Le code correspondant se trouve en annexe. Pour la modélisation, nous avons choisi une distribution de poisson pour la distribution des noeuds secondaires  $X_{i,n}$ . On remarque bien cette corrélation sur la figure ci-dessous.



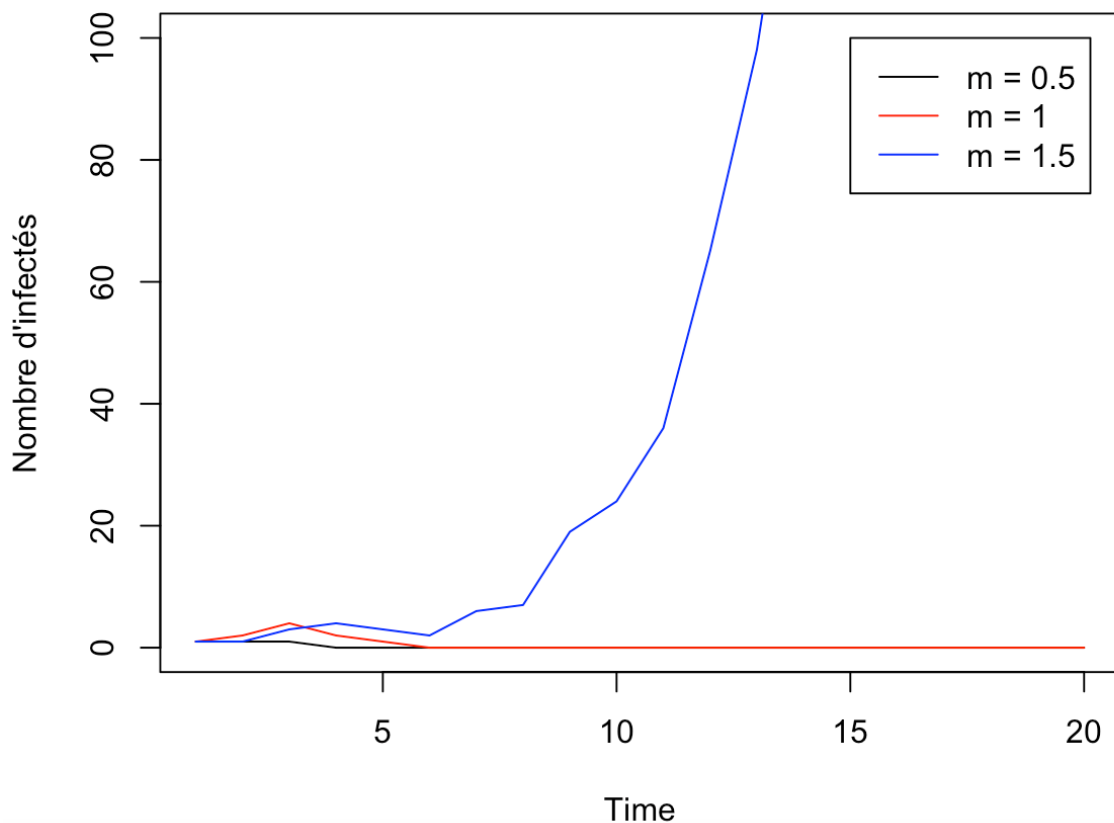


FIGURE 9 – Modélisation d'un processus de branchement où  $X_i \sim \mathcal{P}(m)$  pour différentes valeurs de  $m$

*Le code correspondant est en annexe rubrique "Code R 6".*

Cependant, si on est dans le cas où le processus ne s'éteint rapidement, c'est à dire si  $m > 1$ , on n'a pas d'idée de la probabilité que le processus concerné s'éteigne. On peut cependant l'approcher sur python, en simulant 100 processus avec une valeur de  $m$  donnée, et en regardant combien survivent, autrement dit dans combien de cas l'épidémie représentée par le processus en question aurait persisté et dans combien de cas elle se serait éteinte.

En même temps, on peut compter le nombre de processus qui ont survécu parmi les 100 processus. Avec  $\mu = 1.5$ , il y a environs 60 processus survécus parmi 100, avec  $\mu = 1$ , ce nombre devient très faible, à moins de 5, avec  $\mu = 0.5$ , aucun processus ont survécu.

On peut voir que la probabilité d'extinction varie selon la valeur de  $m$ . Si  $m$  est plus petit que 1, alors l'épidémie va sûrement s'éteindre, si  $m$  égale à 1, alors l'épidémie va s'éteindre presque sûrement, si  $m$  est plus grand que 1, alors l'épidémie va continuer à se propager, plus  $m$  est grand, plus l'épidémie va exploser.

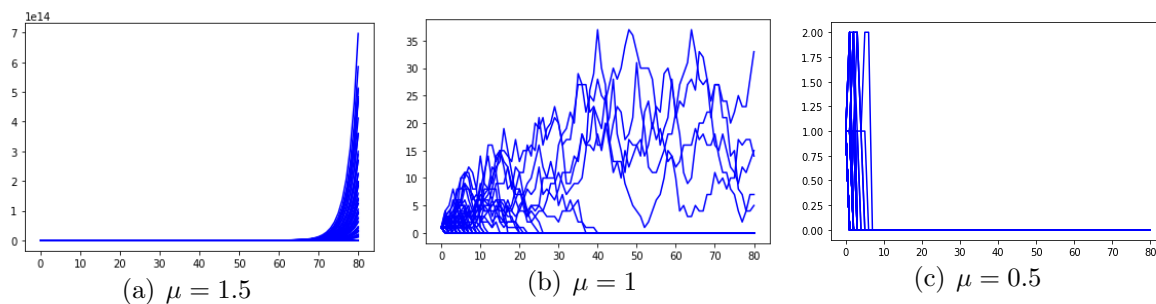


FIGURE 10 – Simulation de 100 trajectoires avec différentes valeurs de  $\mu$

## 4 Conclusion

Dans ce rapport, nous avons fait un résumé de quelques modèles existant pour modéliser la propagation d'une épidémie. Tous les modèles étudiés, qu'ils soient déterministes ou stochastiques, font ressortir un paramètre crucial :  $R_0$ . C'est lui qui détermine l'avenir de l'épidémie au sein de la population étudiée. Dans le cadre de l'application à la réalité, il est donc important de vouloir agir sur sa valeur pour la faire descendre sous la barre de 1, afin de stopper la propagation de l'épidémie. Ceci se fait grâce à des mesures préventives telles que la vaccination, la mise en place de gestes barrières ou de confinements.

## Références

Keeling, M. and Rohani, P., 2011. Modeling Infectious Diseases in Humans and Animals. Princeton : Princeton University Press.

Ross Beckley, Cametria Weatherspoon, Michael Alexander, Marissa Chandler, Anthony Johnson, and Ghan S Bhatt, 2013. Modeling epidemics with differential equations. Tennessee State University, Philander Smith College.

Bongaarts, J. , 1989. A Model of the Spread of HIV Infection and the Demographic Impact of AIDS.

Hugo Falconet et Antoine Jégo, Juin 2015, Modéliser la propagation d'une épidémie.

Reza Yaesoubi and Ted Cohen, 2011. Generalized Markov Models of Infectious Disease Spread : A Novel Framework for Developing Dynamic Health Policies.

Jacob C, 2010. Branching Processes : Their role in Epidemiology

Benaïm El-Karoui. Promenade aléatoire

Sylvie Méléard, Denis Villemonais, 2011. Quasi-stationary distributions and population processes

Sophie Péniisson, 2010. Processus de branchement multitypes conditionnées à l'extinction tardive et illustration en analyse des risques épidémiologiques

Christine Jacob, 2010. Branching Processes : Their Role in Epidemiology

Hethcote, H.W. The Mathematics of Infectious Diseases

Becker, N.G, 1976. Estimation for an epidemic model

Becker, N.G. Estimation for discrete time branching processes with applications to epidemics.

Peter Czappon, François Blanquart, Florence Débarre, 2020. A stochastic epidemiological model to estimate the size of an outbreak at the first case identification

## 5 Annexes

### 5.1 Preuve 1

#### Preuve de la fonction génératrice

On va montrer par récurrence que  $\forall n \in \mathbb{N}^*, G_n = \underbrace{G \circ \dots \circ G}_{n \text{ fois}}(z)$ .

**Initialisation :**

$$G_1(s) = \mathbb{E}(s^{Z_1}) = \mathbb{E}(s^{X_{1,0}}) = G(s)$$

**Hérédité :** On suppose  $\exists n \in \mathbb{N}, G_n = \underbrace{G \circ \dots \circ G}_{n \text{ fois}}(z)$ . On cherche à montrer que

$G_{n+1} = \underbrace{G \circ \dots \circ G}_{n+1 \text{ fois}}(z)$ . On a :

$$\begin{aligned} G_{n+1} &= \mathbb{E} \left[ s^{Z_{n+1}} \right] \\ &= \mathbb{E} \left[ s^{\sum_{i=1}^{Z_n} X_{i,n}} \right] \\ &= \mathbb{E} \left[ \prod_{i=1}^{Z_n} s^{X_{i,n}} \right] \\ &= \mathbb{E} \left[ \sum_{k=0}^{\infty} \left( \prod_{i=1}^k s^{X_{i,n}} \mathbf{1}_{Z_n=k} \right) \right] \quad (\text{réécriture du produit}) \\ &= \sum_{k=0}^{\infty} \mathbb{E} \left[ \prod_{i=1}^k s^{X_{i,n}} \mathbf{1}_{Z_n=k} \right] \quad (\text{propriété de l'espérance}) \end{aligned}$$

Or  $Z_n \perp\!\!\!\perp X_{i,n}$  car  $Z_{n-1}$  ne dépend que de  $Z_{n-1}$  et des  $(X_{i,n-1})_{i \in \mathbb{N}}$  donc par récurrence immédiate on voit que  $Z_n$  ne dépend que des  $(X_{i,j})_{i \geq 0, j < n}$  et vu que les  $X_{i,j}$  sont indépendants, on obtient que  $\forall i \in \mathbb{N}, Z_n \perp\!\!\!\perp X_{i,n}$ . On peut donc séparer le produit et l'indicatrice dans l'espérance :

$$\begin{aligned} G_{n+1} &= \sum_{k=0}^{\infty} \mathbb{E} \left[ \prod_{i=1}^k s^{X_{i,n}} \right] \mathbb{E} [\mathbf{1}_{Z_n=k}] \\ &= \sum_{k=0}^{\infty} \prod_{i=1}^k \mathbb{E} [s^{X_{i,n}}] \mathbb{P}(Z_n = k) \quad (\text{car les } X_{i,n} \text{ sont indépendants}) \\ &= \sum_{k=0}^{\infty} \mathbb{P}(Z_n = k) G(s)^k \\ &= G_n(G(s)) \\ &= \underbrace{G \circ \dots \circ G}_{n+1 \text{ fois}}(z) \quad (\text{par hypothèse de récurrence}) \end{aligned}$$

**Conclusion :**

18

On a donc bien montré la propriété de récurrence, on a  $\forall n \in \mathbb{N}, G_n = \underbrace{G \circ \dots \circ G}_{n \text{ fois}}(z)$


## 5.2 Preuve 2

### Preuve du théorème

Puisque  $G(1) = 1$ , le graphe de  $G$  coupe la droite  $y=x$  (en rouge sur les figures) au moins au point  $(1,1)$ . On va donc se placer dans chacun des cas et montrer que dans le cas sur-critique, il existe un autre point fixe plus petit qui sera la probabilité d'extinction, et dans les cas critique sous-critique, le seul point fixe de  $G$  sur  $]0,1[$  est  $(1,1)$ .

Cas Sur-critique ( $m > 1$ ) :

Pour trouver le point d'extinction de  $G$ , on étudie les points où  $G(x) = x \Leftrightarrow \gamma(x) = G(x) - x = 0$   
 $x \rightarrow \gamma'(x) = G'(x) - 1$  est une fonction croissante de  $G'(0) - 1 < 0$  à  $m - 1 > 0$ , donc par le théorème des valeurs intermédiaire (car  $G \in C^1$  donc sa dérivée est continue), on est sûrs qu'elle s'annule en un point  $\alpha \in ]0, 1[$ . La fonction  $\gamma$  est donc décroissante sur  $[0, \alpha]$  puis croissante sur  $[\alpha, 1]$ . Comme  $G(0) > 0$  et  $G(1) - 1 = 0$ , il existe un point dans l'intervalle  $]0, \alpha]$  où la fonction s'annule.  $\mathbb{P}_{ext}$  est donc l'unique point fixe de  $G$  sur l'intervalle  $]0, 1[$  donc  $\mathbb{P}_{ext} < 1$ .

$x$	0	$\mathbb{P}_{ext}$	$\alpha$	1
$G'(x) - 1$	$G'(0) - 1$	-	0	$m - 1$
$G(x) - x$	$G(0)$			

Cas  $m \leq 1$  :

On fait la même étude, sauf que maintenant on remarque que  $\gamma'(x) = G'(x) - 1$  est croissante de  $G'(0) - 1 < 0$  à  $m - 1 < 0$ , donc strictement négative. Alors,  $\gamma$  est une fonction strictement décroissante de  $G(0) > 0$  à 0.  $G$  admet donc un seul point fixe, 1 et donc  $\mathbb{P}_{ext} = 1$ .

$x$	0	$1 = \mathbb{P}_{ext}$
$G'(x) - 1$	$G'(0) - 1$	$m - 1$
$G(x) - x$	$G(0)$	0

# Preuve (suite)

Graphiquement :

On conclut donc que si  $m = G'(1) > 1$ , autrement dit si la pente de la courbe représentative de  $G$  au point 1 est supérieure à 1, alors la convexité de la fonction garantit que  $\mathbb{P}_{ext}$  soit inférieure à 1. Dans le cas inverse, la convexité de  $G$  garantit que la fonction ne coupe pas la droite  $y=x$  en un autre point et garantit donc que  $\mathbb{P}_{ext} = 1$ . On peut le vérifier en traçant les courbes :

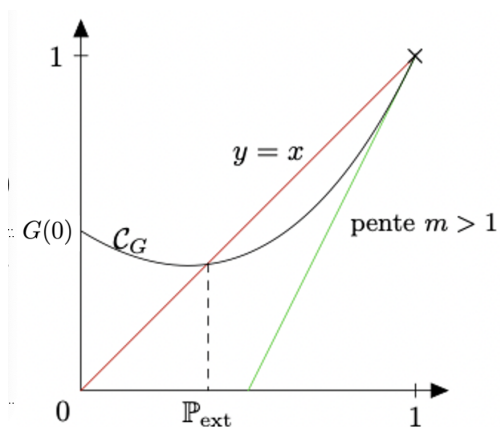


FIGURE 11 –  $G$  dans le cas où  $m > 1$

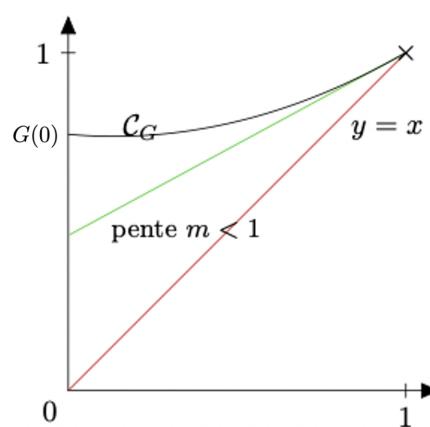


FIGURE 12 –  $G$  dans le cas où  $m < 1$

## 5.3 Code PYTHON 1

### Code pour la simulation du modèle SIR sans démographie

Cas  $R_0 > 1$  :

```
import matplotlib.pyplot as plt
import numpy as np
##### Paramètres #####
beta=1.4247 #beta = transmission rate
gamma=0.14286 #gamma = recovery rate
S0=1-1e-6 # CI
I0=1e-6
X0 = (S0, I0, 0.0)
dt = 1

##### Equation différentielle #####

def SIR1(x):
    F = np.zeros(3)
    F[0] = -beta*x[0]*x[1] #S
    F[1] = beta*x[0]*x[1] - gamma*x[1] #I
    F[2] = gamma*x[1] #R
    return F

##### Méthode de résolution RK4 #####

def RK4 (X, dt):
    k1 = SIR1(X)
    k2 = SIR1(X+(dt/2)*k1)
    k3 = SIR1(X+(dt/2)*k2)
    k4 = SIR1(X+dt*k3)
    return X+ dt* (k1 + 2*k2 + 2*k3 + k4)/6

##### Fonction de résolution itération de la méthode #####

def sol (T, X, dt, method):
    res = np.copy(X0)
    tab = []
    tab.append(X0)
    Inter = X

    for j in range(int(T/dt)):
        res = method(Inter, dt)
        Inter = res
```

## Code pour la simulation du modèle SIR sans démographie (suite)

```

        tab.append(res)

    return np.array(tab).T

##### Tracé #####

SIRwod1 = sol(70, X0, dt, RK4)
plt.plot(SIRwod1[0], label = "susceptible")
plt.plot(SIRwod1[1], label = "infected")
plt.plot(SIRwod1[2], label = "removed")
plt.legend()
plt.xlabel('Time')
plt.ylabel('S, I and R')
plt.savefig('SIR_wod1.png')
plt.show()

Cas  $R_0 > 1$  :

beta=0.78 #beta = transmission rate
gamma=1 #gamma = recovery rate
S0=1-1e-6 # CI
I0=1e-6
X0 = (S0, I0, 0.0)
dt = 1

##### Equation différentielle #####

def SIR1(x):
    F = np.zeros(3)
    F[0] = -beta*x[0]*x[1] #S
    F[1] = beta*x[0]*x[1] - gamma*x[1] #I
    F[2] = gamma*x[1] #R
    return F

##### Méthode de résolution RK4 #####

def RK4 (X, dt):
    k1 = SIR1(X)
    k2 = SIR1(X+(dt/2)*k1)
    k3 = SIR1(X+(dt/2)*k2)
    k4 = SIR1(X+dt*k3)
    return X+ dt* (k1 + 2*k2 + 2*k3 + k4)/6

```



### Code pour la simulation du modèle SIR sans démographie (suite)

##### Fonction de résolution itération de la méthode #####

```
def sol (T, X, dt, method):  
    res = np.copy(X0)  
    tab = []  
    tab.append(X0)  
    Inter = X  
  
    for j in range(int(T/dt)):  
        res = method(Inter, dt)  
        Inter = res  
        tab.append(res)  
  
    return np.array(tab).T
```

##### Tracé #####

```
SIRwod2 = sol(70, X0, dt, RK4)  
plt.plot(SIRwod2[0], label = "susceptible")  
plt.plot(SIRwod2[1], label = "infected")  
plt.plot(SIRwod2[2], label = "removed")  
plt.legend()  
plt.xlabel('Time')  
plt.ylabel('S, I and R')  
plt.savefig('SIR_wod2.png')  
plt.show()
```

## 5.4 Code PYTHON 2

### Code pour la simulation du modèle SIR avec démographie

```
##### Paramètres #####

mu = 0.25/1000 # mu = taux de mortalité naturelle indépendant de l'épidémie
beta=1.4247 #beta = transmission rate
gamma=0.14286 #gamma = recovery rate
S0=1-1e-6 # CI
I0=1e-6
R0=1-S0-I0
X0 = (S0, I0, R0)

##### Equation différentielle #####

def SIR2(x):
    F = np.zeros(3)
    F[0] = mu - beta*x[0]*x[1] -mu*x[0]
    F[1] = beta*x[0]*x[1] - gamma*x[1] -mu*x[1]
    F[2] = gamma*x[1] - mu*x[2]
    return F

##### Méthode de résolution RK4 #####

def RK42 (X, dt):
    k1 = SIR2(X)
    k2 = SIR2(X+(dt/2)*k1)
    k3 = SIR2(X+(dt/2)*k2)
    k4 = SIR2(X+dt*k3)
    return X+ dt* (k1 + 2*k2 + 2*k3 + k4)/6

##### Tracé #####

SIRwd = sol(70, X0, dt, RK42)
plt.plot(SIRwd[0], label = "susceptible")
plt.plot(SIRwd[1], label = "infected")
plt.plot(SIRwd[2], label = "removed")
plt.legend()
plt.savefig('SIR_wd.png')
plt.show()
```

## 5.5 Code PYTHON 3

### Code pour la simulation du modèle SIS

```
##### Paramètres #####

beta=1.4247
gamma=0.14286
io = 1e-6
X0 = [1-io, io]

##### Equation différentielle #####

def SIS(x):
    F = np.zeros(2)
    F[0] = gamma*x[1] - beta*x[1]*x[0]
    F[1] = beta*x[0]*x[1] - gamma*x[1]
    return F

##### Méthode de résolution RK4 #####

def RK42 (X, dt):
    k1 = SIS(X)
    k2 = SIS(X+(dt/2)*k1)
    k3 = SIS(X+(dt/2)*k2)
    k4 = SIS(X+dt*k3)
    return X+ dt* (k1 + 2*k2 + 2*k3 + k4)/6

##### Tracé #####

rk42 = sol(70, X0, dt, RK42)
plt.plot(rk42[0], label = "susceptible")
plt.plot(rk42[1], label = "infected")
plt.xlabel('Time')
plt.ylabel('S and I')
plt.legend()
plt.savefig('SIS.png')
plt.show()
plt.show()]}
```

## 5.6 Code PYTHON 4

Code pour la simulation du modèle SIS avec des chaînes de markov

```
##### paramètres pour  $R_0 < 1$  #####
N=250
beta=0.25
gamma=0.1
b = 0.25
N = 250
deltat =0.0015

##### Création de la matrice de transition #####

P1 = np.zeros([N,N])
P1[0,0] = 1
for i in range (1,N-1):
    P1[i,i+1]= beta*i*(N-i)*deltat/N
    P1[i, i-1] = (beta+gamma)*i*deltat
    P1[i,i] = 1- ((beta*i*(N-i)*deltat/N)+(beta+gamma)*i*deltat)

##### Itérations de la chaine de markov #####

def markov(n,P,x0):

    X=np.zeros(n,dtype=np.int)
    #on indique que X contiendra seulement des entiers
    X[0]=x0
    for k in range(n-1):
        X[k+1]=np.random.choice(a=range(len(P)), p=P[X[k],:])
        # Choix au hasard avec les états numérotés de 1
        # à len(P) et proba la case dans P
    return X

x0 = 100

markov1 = markov(50000, P1, x0)

##### paramètres pour  $R_0 = 1$  #####

beta = 0.3
gamma = 0.1
b = 0.2
```

## Code pour la simulation du modèle SIS avec des chaînes de markov (suite)

```
P2 = np.zeros([N,N])
P2[0,0] = 1
for i in range(1,N-1):
    P2[i,i+1]= beta*i*(N-i)*deltat/N
    P2[i, i-1] = (b+gamma)*i*deltat
    P2[i,i] = 1- ((beta*i*(N-i)*deltat/N)+(b+gamma)*i*deltat)

markov2 = markov(50000, P2, x0)

##### paramètres pour  $R_0 > 1$  #####

beta = 0.5
gamma = 0.1
b = 0.2

P3 = np.zeros([N,N])
P3[0,0] = 1
for i in range(1,N-1):
    P3[i,i+1]= beta*i*(N-i)*deltat/N
    P3[i, i-1] = (b+gamma)*i*deltat
    P3[i,i] = 1- ((beta*i*(N-i)*deltat/N)+(b+gamma)*i*deltat)

markov3 = markov(50000, P3, x0)

plt.plot(markov1, c= 'r', label = "R0 < 1")
plt.plot(markov2, c = 'g', label = "R0 = 1")
plt.plot(markov3, c = 'b', label = "R0 > 1")
plt.legend()
plt.show()
```

## 5.7 Code PYTHON 5

Code pour la simulation du modèle SIR avec des chaînes de markov

```
##### paramètres pour  $R_0 = 1.42$  #####

N=50
beta=1
b=0.2
gamma=0.5
deltat =0.01

(S0, I0) = (N-5, 5)

##### chaine de markov #####

nbpas = 5000
S = np.zeros(nbpas)
I = np.zeros(nbpas)
S[0] = S0
I[0] = I0

for j in range(10):
    for i in range(nbpas-1):
        probas = np.zeros(5)
        probas[0] = beta*S[i]*I[i]/N*deltat
        probas[1] = gamma*I[i]*deltat
        probas[2] = b*I[i]*deltat
        probas[3] = b*(N-S[i]-S[i])*deltat
        probas[4] = 1-probas[0]-probas[1]-probas[2]-probas[3]

        x = np.random.uniform(0, 1)

        if x<= probas[0]:
            S[i+1] = S[i]-1
            I[i+1] = I[i]+1

        elif x<= probas[0]+probas[1]:
            S[i+1] = S[i]
            I[i+1] = I[i]-1
```

## Code pour la simulation du modèle SIR avec des chaînes de markov (suite)

```
        elif x<= probas[0]+probas[1]+probas[2]:
            S[i+1] = S[i]+1
            I[i+1] = I[i]-1

        elif x<= probas[0]+probas[1]+probas[2]+probas[3]:
            S[i+1] = S[i]+1
            I[i+1] = I[i]

        else:

            S[i+1] = S[i]
            I[i+1] = I[i]
    plt.plot(I)

plt.show()

##### paramètres pour R0 = 2.03 #####

beta=1.4247
b=0.5
gamma=0.2
deltat =0.01

(S0, I0) = (N-5, 5)

##### chaine de markov #####

nbpas = 5000
S2 = np.zeros(nbpas)
I2 = np.zeros(nbpas)
S2[0] = S0
I2[0] = I0
```

### Code pour la simulation du modèle SIR avec des chaînes de markov (suite)

```
for j in range(10):
    for i in range(nbpas-1):

        probas = np.zeros(5)
        probas[0] = beta*S2[i]*I2[i]/N*deltat
        probas[1] = gamma*I2[i]*deltat
        probas[2] = b*I2[i]*deltat
        probas[3] = b*(N-S2[i]-S2[i])*deltat
        probas[4] = 1-probas[0]-probas[1]-probas[2]-probas[3]

        x = np.random.uniform(0, 1)

        if x<= probas[0]:
            S2[i+1] = S2[i]-1
            I2[i+1] = I2[i]+1

        elif x<= probas[0]+probas[1]:
            S2[i+1] = S2[i]
            I2[i+1] = I2[i]-1

        elif x<= probas[0]+probas[1]+probas[2]:
            S2[i+1] = S2[i]+1
            I2[i+1] = I2[i]-1

        elif x<= probas[0]+probas[1]+probas[2]+probas[3]:
            S2[i+1] = S2[i]+1
            I2[i+1] = I2[i]

        else:

            S2[i+1] = S2[i]
            I2[i+1] = I2[i]

    plt.plot(I2)
plt.show()
```



## Code pour la simulation du modèle SIR avec des chaînes de markov (suite)

```
##### paramètres #####

beta=1
b=0.3
gamma=0.1    ## R0 = 2.5
deltat =0.01

(S0, I0) = (N-5, 5)

##### chaine de markov #####

nbpas = 5000
S3 = np.zeros(nbpas)
I3 = np.zeros(nbpas)
S3[0] = S0
I3[0] = I0

for j in range(10):
    for i in range(nbpas-1):

        probas = np.zeros(5)
        probas[0] = beta*S3[i]*I3[i]/N*deltat
        probas[1] = gamma*I3[i]*deltat
        probas[2] = b*I3[i]*deltat
        probas[3] = b*(N-S3[i]-I3[i])*deltat
        probas[4] = 1-probas[0]-probas[1]-probas[2]-probas[3]

        x = np.random.uniform(0, 1)

        if x<= probas[0]:
            S3[i+1] = S3[i]-1
            I3[i+1] = I3[i]+1

        elif x<= probas[0]+probas[1]:
            S3[i+1] = S3[i]
            I3[i+1] = I3[i]-1
```

### Code pour la simulation du modèle SIR avec des chaînes de markov (suite)

```
        elif x<= probas[0]+probas[1]+probas[2]:
            S3[i+1] = S3[i]+1
            I3[i+1] = I3[i]-1

        elif x<= probas[0]+probas[1]+probas[2]+probas[3]:
            S3[i+1] = S3[i]+1
            I3[i+1] = I3[i]

        else:

            S3[i+1] = S3[i]
            I3[i+1] = I3[i]
        plt.plot(I3)
plt.show()

plt.plot(I, label = "R0 = 1.42")
plt.plot(I2, label = "R0 = 2")
plt.plot(I3, label = "R0 = 2.5")
plt.ylabel("Nombre d'infectés")
plt.xlabel("Nombre de pas (unités de temps)")
plt.legend()
plt.savefig('SIR_Markov.png')
plt.show()
```

## 5.8 Code R 6

### Code pour la simulation d'un processus de branchement

```
#initialize list of population size at generation n
Z1<-rep(0,20)
Z2<-rep(0,20)
Z3<-rep(0,20)

#pop. size 1 at generation 0
Z1[1] = 1
Z2[1] = 1
Z3[1] = 1

#poisson r.v.
#each member generates poisson(lambda) offspring
#sum offspring

lambda = 0.5 #<1
N = 20
for (i in 2:N)
{
  if(Z1[i-1]==0) {Z1[i]=0} else
  {
    x=rpois(Z1[i-1], lambda)
    Z1[i] = sum(x)
  }
}
Z1

#####
lambda = 1 #=1
N = 20
for (i in 2:N)
{
  if(Z2[i-1]==0) {Z2[i]=0} else
  {
    x=rpois(Z1[i-1], lambda)
    Z2[i] = sum(x)
  }
}
Z2
```

### Code pour la simulation d'un processus de branchement (suite)

```
#####  
lambda = 1.5 #>1  
N = 20  
for (i in 2:N)  
{  
  if(Z3[i-1]==0) {Z3[i]=0} else  
  {  
    x=rpois(Z3[i-1], lambda)  
    Z3[i] = sum(x)  
  }  
}  
Z3  
  
#####  
plot(Z1, type = "l", col = 'black', xlim = c(1,20), ylim = c(1,100))  
lines(Z2, type = "l", col = 'red')  
lines(Z3, type = "l", col = 'blue')  
  
legend(14,100,legend=c("m = 0.5", "m = 1", "m = 1.5"), col=c("black",  
  "red", "blue"),lty=c(1,1,1), ncol=1)
```