

## DIVERSITÉ

Dissertation in partial fulfillment of the requirements for the degree of

### TRAVAIL DE BACHELOR SCIENCES INFORMATIQUES



Université de Genève  
Faculté des sciences  
Département d'informatique

Maxime Pasquier

14.06.2021

## DIVERSITÉ

Dissertation in partial fulfillment of the requirements for the degree of

### TRAVAIL DE BACHELOR SCIENCES INFORMATIQUES

Université de Genève  
Département d'informatique

Approuvé par

Superviseurs, Professeur Bastien Chopard & Franck Raynaud

Examiner, Professeur Bastien Chopard & Franck Raynaud

14.06.2021

# **Abstract**

## Acknowledgements

# Table of Contents

<b>Abstract</b>	<b>II</b>
<b>Acknowledgements</b>	<b>III</b>
<b>List of Tables</b>	<b>V</b>
<b>List of Figures</b>	<b>VI</b>
<b>List of Acronyms</b>	<b>VII</b>
<b>1 Résumé</b>	<b>1</b>
1.1 first section . . . . .	1
<b>2 Introduction</b>	<b>2</b>
2.1 first section . . . . .	2
<b>3 Approche du problème, méthodes, outils utilisés</b>	<b>3</b>
3.1 Code génétique . . . . .	3
3.2 Compatibilité entre individus et agents pathogènes . . . . .	3
3.3 Outils . . . . .	4
<b>4 Modèle</b>	<b>5</b>
4.1 Acteurs . . . . .	5
4.1.1 Individus . . . . .	6
4.1.2 Agents pathogènes . . . . .	6
4.2 Espace physique . . . . .	7
4.2.1 Cellule . . . . .	7
4.2.2 Grille régulière . . . . .	7
4.2.3 Voisinage . . . . .	9
4.3 Simulation . . . . .	9
4.3.1 Itérations . . . . .	10
4.3.2 Mouvements des acteurs . . . . .	11
4.3.3 Mise à jour des acteurs . . . . .	12
4.4 Interactions . . . . .	12
4.4.1 Interactions sur la même cellule . . . . .	13
4.4.2 Interactions sur cellules voisines . . . . .	13
4.4.3 Contamination de cellule . . . . .	14

4.5	Diversité génétique . . . . .	14
4.5.1	Distance de Hamming . . . . .	15
4.6	Immunisation et résistance naturelle . . . . .	16
4.7	Mutations . . . . .	17
<b>5</b>	<b>Implémentation</b>	<b>19</b>
5.1	Main . . . . .	19
5.2	Initialisation . . . . .	20
5.3	Exécution de la simulation . . . . .	21
5.4	Prise des mesures . . . . .	25
5.5	Fermeture . . . . .	25
<b>6</b>	<b>Objectifs</b>	<b>26</b>
<b>7</b>	<b>Modèles Compartimentaux</b>	<b>28</b>
7.1	Modèle SI . . . . .	29
7.2	Modèle SIR . . . . .	30
7.3	Simulations de références . . . . .	31
7.3.1	Simulations de SI . . . . .	32
7.3.2	Simulations de SIR . . . . .	33
<b>8</b>	<b>Modèle SI</b>	<b>35</b>
8.1	Mesures et méthodologie SI . . . . .	35
8.2	Résultats . . . . .	36
8.3	Analyses . . . . .	40
8.3.1	Mean Absolute Error . . . . .	40
8.3.2	Moyenne de voisinage . . . . .	40
8.3.3	Variations aléatoires . . . . .	41
8.3.4	Positions des individus . . . . .	43
<b>9</b>	<b>Modèle SIR</b>	<b>45</b>
9.1	Mesures et méthodologie SIR . . . . .	45
9.2	Résultats . . . . .	45
9.3	Analyses . . . . .	48
9.3.1	Mean Absolute Error . . . . .	48
9.3.2	Variations aléatoires . . . . .	50
9.3.3	Latence des simulations . . . . .	51
9.3.4	Mouvements variable . . . . .	52
9.3.5	Comparaison 1000 mouvements . . . . .	53

# List of Tables

Table 8.1: Mean Abosolute Error Normalized : SI . . . . .	40
Table 8.2: Voisinage moyen : SI . . . . .	41
Table 8.3: Variations : SI . . . . .	42
Table 9.1: Mean Abosolute Error Normalized : SI . . . . .	49
Table 9.2: Variations : SIR . . . . .	51

# List of Figures

Figure 4.1: grille régulière . . . . .	8
Figure 4.2: grille régulière . . . . .	9
Figure 4.3: grille régulière . . . . .	10
Figure 4.4: grille régulière . . . . .	11
Figure 4.5: Calcul de la distance de Hamming . . . . .	15
Figure 5.1: Matrices de pointeurs . . . . .	21
Figure 5.2: paramètre = 1 . . . . .	23
Figure 5.3: paramètre = 2 . . . . .	23
Figure 5.4: paramètre = 4 . . . . .	23
Figure 5.5: paramètre = 8 . . . . .	23
Figure 7.1: Modèle SI . . . . .	30
Figure 7.2: Modèle SIR . . . . .	31
Figure 8.1: Simulations de SI, densité 1/2 . . . . .	36
Figure 8.2: Simulations de SI, densité 1/4 . . . . .	37
Figure 8.3: Simulations de SI, densité 1/8 . . . . .	38
Figure 8.4: Simulations de SI, densité 1/8 . . . . .	39
Figure 8.5: Divergences SI . . . . .	42
Figure 8.6: test . . . . .	43
Figure 8.7: test . . . . .	44
Figure 9.1: test . . . . .	45
Figure 9.2: test . . . . .	46
Figure 9.3: test . . . . .	47
Figure 9.4: test . . . . .	48
Figure 9.5: test . . . . .	50
Figure 9.6: test . . . . .	52
Figure 9.7: test . . . . .	53
Figure 9.8: test . . . . .	54
Figure 9.9: test . . . . .	54

## **List of Acronyms**

# 1 Résumé

## 1.1 first section

Une pandémie est une épidémie présente sur une grande zone et affectant un grand nombre de personnes. Ces épidémies peuvent être causées par de multiples facteurs. L'espèce humaine est touchée par des pandémies à une fréquence d'une fois par siècle mais depuis les années 2000, ce rythme s'est grandement accéléré. En effet, nous avons été victime de pandémies comme Ebola en 2013, SARS-2 en 2008, COVID en 2020 et bien d'autres. Ces épidémies à grande échelle deviennent de plus en plus fréquentes pour plusieurs raisons que nous ne traiterons pas ici.

Le travail se concentre sur l'aspect de diversité écologique des populations touchées lors de pandémies. La diversité d'une espèce est un facteur clé permettant sa survie et son évolution. L'objectif de l'étude est de constater l'impact de la diversité des populations sur la propagation de pandémies. En effet, une grande diversité au sein d'une population ralentit la propagation d'une pandémie. Il s'agit dans ce travail d'étudier les différentes issues de simulations en modifiant le paramètre de diversité de la population étudiée.

L'étude se concentre sur la réalisation d'une simulation permettant de modéliser une population avec des caractéristiques paramétrables. Nous contaminons un individu et observons au fil du temps la propagation ou non de l'épidémie.

La simulation se base sur le principe d'automate cellulaire. Le système se présente sous forme d'une grille bidimensionnelle tel un échiquier de taille voulue. La population navigue sur cet échiquier et les individus interagissent entre eux avec les autres. Le but est de simuler les mouvements au sein d'une population et les contacts permettant la propagation d'agents pathogènes. Au fil du temps de la simulation nous pouvons étudier la propagation de la pandémie en fonction des paramètres du modèle.

L'objectif est d'étudier les résultats de simulations en faisant varier la diversité de la population ainsi que d'autres facteurs. Les résultats permettraient de quantifier l'impact qu'à la diversité d'une espèce sur une éventuelle pandémie.

## 2 Introduction

### 2.1 first section

L'élément principal de la recherche scientifique est l'étude de propagation de pandémies en fonction de la diversité de la population étudiée. Pour des questions de simplification, beaucoup de facteurs externes ont été ignorés. En effet, l'apparition et la propagation d'une épidémie est le résultat d'une multitude de facteurs complexes. La simulation se base exclusivement sur des paramètres simples tout en ignorant des aspects comme l'âge, la condition de vie ou encore l'accès aux soins. Le modèle se focalise sur le principe de génomes, mutations et immunités. D'autres facteurs sont paramétrables, comme la taille du système ou le nombre de personnes et bien d'autres mais les résultats recherchés sont basés sur l'aspect de diversité.

La simulation ne comporte que deux types d'acteurs : des individus d'une population donnée et des agents pathogènes. L'espace représenté dans la simulation est une grille bidimensionnelle permettant aux individus de se déplacer librement. Ce modèle simplifié permet de représenter les interactions entre les acteurs du système. Initialement, un seul individu du système est contaminé et il s'agit de déterminer l'évolution de la situation. On cherche à observer si l'agent pathogène parvient à se propager ou non parmi la population d'individu. Une multitude d'autres paramètres peuvent être modifiés pour influencer la simulation.

La recherche principale est d'observer et mesurer l'évolution du système en faisant varier les diversités des acteurs, en sachant que chaque acteur du système possède un indicateur représentant son code génétique.

La question que l'on se pose pour ce travail est de déterminer si la diversité d'une population touchée par une épidémie modifie la propagation d'un agent pathogène ou non. Il s'agit ici d'illustrer et de quantifier cet effet à l'aide d'un modèle numérique.

## 3 Approche du problème, méthodes, outils utilisés

### 3.1 Code génétique

Le travail se concentre sur la notion de diversité. Il faut donc un moyen pour représenter la diversité d'une population et les interactions entre différents acteurs en prenant en considération leur diversité. La méthode utilisée se base sur le principe de génome. On attribue un génome, qui représente le code génétique d'un être vivant à chaque acteur du système. Par conséquent, chaque individu et chaque agent pathogène possède un génome. Ce génome se présente sous la forme d'un entier codé sur 4 octets. Nous avons donc 32 bits disponibles afin de représenter le génome d'un acteur.

La notion de diversité d'une population peut à présent se définir par de grandes différences dans ces valeurs sur 32 bits d'un individu à un autre. En effet, il y a une grande diversité au sein d'une population si les génomes des individus sont différents les uns des autres. Nous pouvons donc générer des séquences pour les génomes en prenant en compte le facteur de diversité du modèle. Il est possible de construire ces génomes, principalement pour la population d'individu, en complémentant un certain nombre de bits d'une séquence de référence. L'idée ici est de commencer par attribuer à tous les individus un génome fixe et identique. Dans cette configuration, nous avons une diversité nulle étant donné que tout le monde a le même code génétique. Dans le cas d'une diversité non nulle, nous modifions (complémentons) certains bits de la séquence de génome des individus. Ce processus se fait aléatoirement et le nombre de complémentation dépend du paramètre de diversité du modèle. Avec cette méthode on finit avec des génomes déviant plus ou moins d'un certain génome de référence.

### 3.2 Compatibilité entre individus et agents pathogènes

Les génomes étant défini, il reste à définir la manière dont les agents pathogènes et les individus réagissent les uns avec les autres. Le problème est l'utilisation des génomes, que ce soit ceux des agents pathogènes ou ceux des individus. Comment gérer les interactions entre individus et agents pathogènes en se basant sur des génomes de 32 bits ? La technique utilisée est basée sur la distance de Hamming entre les séquences des génomes. Cette méthode fournit une solution interprétable en cas de contact entre deux

acteurs interagissants. Cette technique permet de traduire l'interaction de deux génomes en une action comme la contamination d'un individu sain.

La distance de Hamming est une notion mathématique qui permet de calculer les différences entre deux séquences binaires par exemple. Cette technique consiste simplement à compter le nombre de symboles différents pour deux suites du même nombre de symboles. Il s'agit donc de parcourir une des séquence et pour chaque indice comparer avec le symbole correspondant de l'autre séquence. Pour chaque symbole différent, la distance de Hamming est incrémentée de 1.

Nous pouvons donc représenter la compatibilité entre un individu et un agent pathogène par cette distance sous forme d'un entier. Il s'agit ensuite de convertir cette valeur en une probabilité. En effet, dans notre exemple il existe uniquement 33 valeurs possible de distance de Hamming qui caractérisent un match de génomes. Si on applique une fonction de seuillage sur si peu de valeurs, les résultats vont être trop tranchés. Par conséquent on convertit cette distance calculé entre deux génomes en probabilité. Cette probabilité peut ensuite être évaluée ce qui détermine les bonnes actions à effectuer.

### 3.3 Outils

L'outil principal de la simulation est le C++, un langage puissant et performant. L'avantage de ce langage est qu'il est très rapide, ce qui est nécessaire pour exécuter des simulations de plus grande taille. De plus le C++ permet la programmation orientée objet, utilisé ici pour modéliser les individus ainsi que les agents pathogènes.

Le langage utilisé permettant les analyses et représentations graphiques est Python. Ce langage à l'avantage d'être facile à manipuler et puissant pour faire de l'analyse. C'est donc un langage idéal pour des représentations de données. De plus Python dispose de bibliothèques logicielles comme Numpy ou encore Matplotlib qui permettent de faire du calcul scientifique ainsi que de représenter des données.

Le travail complet est disponible sur un dépôtsitoire GitHub et est accessible librement. Ceci fournit de la transparence ainsi qu'un moyen de distribuer le projet. De plus toutes les versions antérieurs de la recherche y figurent.

## 4 Modèle

Le modèle cherche à simuler la propagation de pandémies dans une population. Le système inclut des acteurs qui peuvent faire des actions ainsi qu'un \*\*espace physique\*\* sur lequel les acteurs peuvent se déplacer. Les acteurs sont des organismes vivants qui se déclinent en deux espèces distinctes. Le premier groupe fait partie des organismes victimes de l'épidémie et le second groupe est responsables de l'infection. Par conséquent nous sommes dans une situation où une espèce attaque une autre. Par contre dans le fonctionnement du modèle, il y a des restrictions sur la nature de l'espèce attaquante. Nous parlons ici d'organismes capables de contaminer un hôte.

Les organismes évoluent dans une monde plat et parfaitement géométrique en deux dimensions. L'espace est comparable à un échiquier avec les acteurs étant des pions.

### 4.1 Acteurs

Il existe exclusivement deux espèces d'organismes dans le système :

Les agents pathogènes sont les agents infectieux responsables de l'épidémie.

Les hôtes potentiels, éléments de la population victime de l'épidémie.

La classe "agents pathogènes" ne fait pas référence à une espèce en particulier mais reflète des organismes avec le pouvoir d'infecter une autre espèce. Il peut s'agir de virus, bactéries ou encore parasites. La caractéristique de ce groupe est qu'il a le pouvoir de rendre malade un hôte et donc d'affecter une espèce par une maladie qu'il génère. Il peut s'agir de maladies touchant des humains ou de zoonoses suivant la nature des espèces du système. Le terme "pathogène" est aussi utilisé pour caractériser cette classe.

Les "Individus" sont les organismes susceptibles de développer la maladie suite à la contamination par un agent pathogène. Il peut s'agir d'humains tout comme d'animaux ou de plantes. La restriction est que cette espèce doit être affectée par les agents pathogènes ainsi que par la maladie. Le terme "hôte potentiel" signifie que l'organisme peut devenir l'hôte de l'agent pathogène. Nous parlons de "hôte du pathogène" ou de "hôte contaminé" lorsque l'organisme est contaminé et porteur de l'agent pathogène. Le travail utilise souvent le mot "individu" et ceci pour représenter un acteur de ce groupe. Le fait que l'individu soit sain ou contaminé est toujours précisé.

#### 4.1.1 Individus

Un individu peut se présenter sous deux formes distinctes. Il est soit sain, soit contaminé par un agent pathogène.

Le premier cas représente un hôte potentiel qui n'est pas contaminé par un agent pathogène et donc n'est pas touché par la maladie. Contrairement à ce que l'on pourrait penser, l'illustration ne fait pas référence à un humain mais à un individu potentiellement vulnérable à la classe des agents pathogènes.

Le second cas fait référence à la même espèce mais cette fois ci, l'individu est contaminé par un agent pathogène. Le modèle ne fait pas la distinction entre malade et porteur du virus. Par conséquent, un hôte porteur du pathogène est d'office considéré comme malade. De plus, le modèle définit un taux de contagion qui est paramétrable. Le niveau de contagion est donc constant pour tous les organismes contaminés et ceci tout du long de la simulation. Un hôte infecté par un agent pathogène est donc contagieux, ce qui signifie qu'il a la pouvoir de transmettre l'agent pathogène à d'autres individus de la même espèce. Tout comme pour le point précédent, l'illustration ne fait pas forcément référence à un humain.

#### 4.1.2 Agents pathogènes

Un agent pathogène peut se présenter sous deux formes distinctes. Il est soit contaminant un hôte, soit contaminant un espace physique.

Le premier cas est identique à celui précédemment développé mais cette fois ci, la situation est perçue du point de vue de l'agent pathogène. Quand un hôte devient contaminé par un agent pathogène, ce dernier est absorbé par l'hôte. C'est-à-dire que l'agent pathogène n'est plus une entité distincte mais fait partie intégrante de l'individu. Par conséquent l'individu devient un hôte contaminé et non pas un hôte sain associé à un agent pathogène.

Le second cas indique qu'une surface physique est contaminée par un agent pathogène. Un hôte contaminé, en plus de contaminer d'autres individus sains, peut contaminer une surface. Si une surface est contaminée, une copie de l'agent pathogène de l'hôte se dépose sur la surface. Un agent pathogène qui contamine une surface est inerte, c'est-à-dire qu'il ne peut pas se déplacer ni muter par contre il a toujours le potentiel de contaminer des individus sains qui se trouveraient sur cette surface. Un agent pathogène isolé sur une surface ainsi n'a une durée de vie que très limitée. Sa chance survie est paramétrable dans le modèle.

## 4.2 Espace physique

L'espace physique est une surface plane sur laquelle on peut placer et déplacer des acteurs. La surface se présente sous la forme d'une grille régulière tel un échiquier. Cette grille contient des cellules dans lesquelles on peut y placer un acteur. Cet espace bidimensionnel est la seule représentation spatiale du modèle. Chaque acteur a des coordonnées dans cet espace qui définissent sa position. Par conséquent, chaque cellule de la grille est indexée par une paire d'entiers.

### 4.2.1 Cellule

Dans la représentation du modèle, une cellule n'a que 6 états possibles, c'est-à-dire que chaque cellule de la grille ne peut se retrouver que dans 6 configurations différentes. L'état d'une cellule fait référence à l'acteur ou aux acteurs qui se trouvent dessus. Concrètement une cellule est simplement une surface carrée mais nous associons les acteurs se trouvant sur la cellule à cette dernière et ceci décrit son état. Les 6 états possibles d'une cellule sont listés ci-dessous.

1. La cellule est vide, aucun acteur ne l'occupe. Cette situation signifie simplement que la surface n'est occupée par aucun acteur et est libre pour en accueillir un.
2. Un hôte potentiel se trouve sur la cellule.
3. Un hôte infecté se trouve sur la cellule.
4. Un agent pathogène se trouve sur la cellule, infectant l'espace, sans hôte.
5. Un hôte potentiel sain se trouve sur une cellule contaminée par un agent pathogène.
6. Un hôte contaminé se trouve sur une cellule déjà contaminée par un agent pathogène.

Le cas 5 et 6 sont les deux seuls cas où deux acteurs se trouvent simultanément sur la même cellule. Il est par conséquent impossible que deux individus se retrouvent sur la même cellule au même moment.

### 4.2.2 Grille régulière

La grille régulière est l'espace dans lequel les acteurs évoluent. Chaque acteur est représenté au niveau spatial par ses coordonnées sur la grille. Par conséquent nous utilisons une grille contenant les acteurs du système. Cette dernière leur permet de se déplacer et d'interagir. Cet espace est la seule représentation spatiale du modèle, c'est-à-dire que les acteurs ne peuvent évoluer spatialement qu'en respectant la géométrie de la grille. L'illustration suivante semble avoir des bords fixes mais en réalité ce sont des bords périodiques. Cela signifie qu'un acteur n'est pas bloqué par un bord du système. Dans le cas où un individu souhaite se déplacer en dehors du système nous le faisons sauter à l'opposé de la grille. Par conséquent les bords sont connectés afin d'éviter les

effets de bords.

Un exemple de système en cours de simulation est illustré ci-dessous.

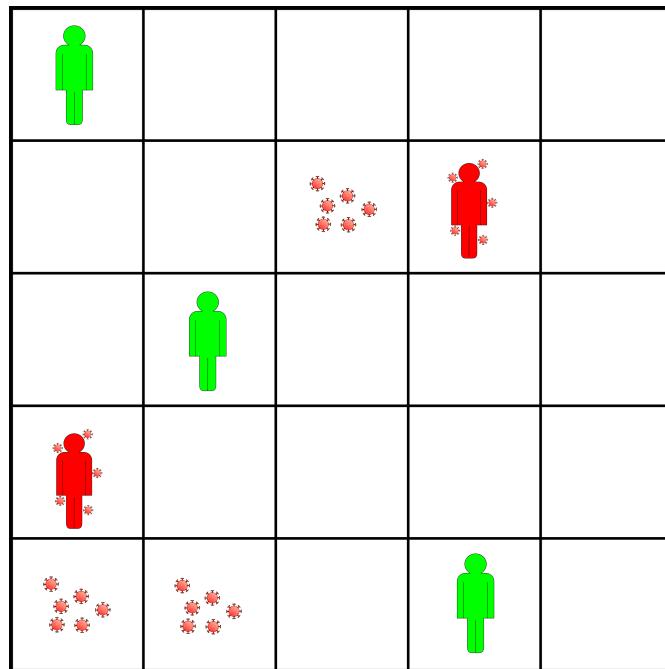


Figure 4.1: grille régulière

L'espace bidimensionnelle permet de donner une représentation spatiale aux acteurs mais également de les faire interagir les uns avec les autres. Les interactions se font par la notion de voisinage.

### 4.2.3 Voisinage

La représentation spatiale permet de situer tous les acteurs du système relativement les uns aux autres. Nous définissons que dans cet espace, les acteurs avec des coordonnées proches sont géographiquement rapprochés et peuvent s'influencer. Le voisinage est la transcription de cette proximité entre les acteurs et permet de définir la portée de l'influence des acteurs. Le voisinage d'une cellule est l'espace avoisinant à la cellule. Dans le modèle, le voisinage est représentée ainsi :

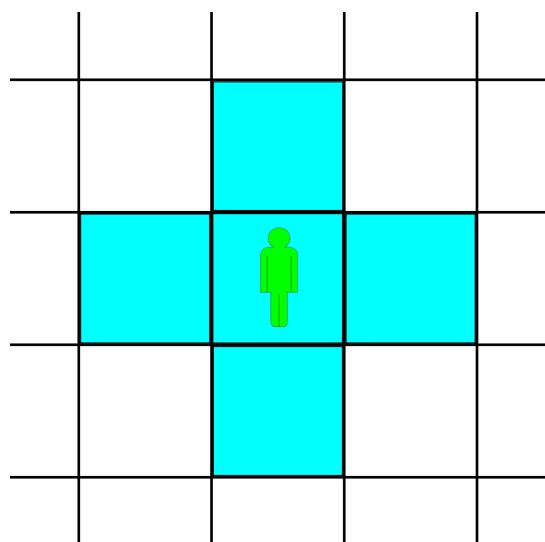


Figure 4.2: grille régulière

Le voisinage de la cellule contenant un acteur est les 4 cellules avoisinantes directes en plus de la cellule centrale. Par conséquent notre humain dans cette cellule ne pourra interagir qu'avec d'autres acteurs dans les cellules voisines (en cyan) ou avec un autre acteur sur sa cellule. Le reste du système lui est hors de portée et donc invisible. Par conséquent, les actions et les états d'un acteur du système ne sont influencés que par l'état actuel de l'acteur ainsi que les autres acteurs se trouvant dans son voisinage. Toutes les autres cellules ainsi que leurs acteurs n'ont aucun impacte sur l'individu étudié.

## 4.3 Simulation

Afin d'initialiser une simulation nous commençons par définir la taille de la grille et le nombre de personnes et nous contaminons un individu. Il existe une multitude d'autres paramètres définissant les comportements des différentes mécaniques de la simulation que nous expliquerons plus loin. Par conséquent, une situation initiale se présente toujours sous la forme ci-dessous. Notre population est saine, sauf un seul individu qui porte l'agent pathogène initial. C'est le point de départ de toute épidémie avec la contamination du premier hôte. Il s'agit ensuite d'observer la propagation ou non du

pathogène initial. Un exemple de configuration initiale pourrait être :

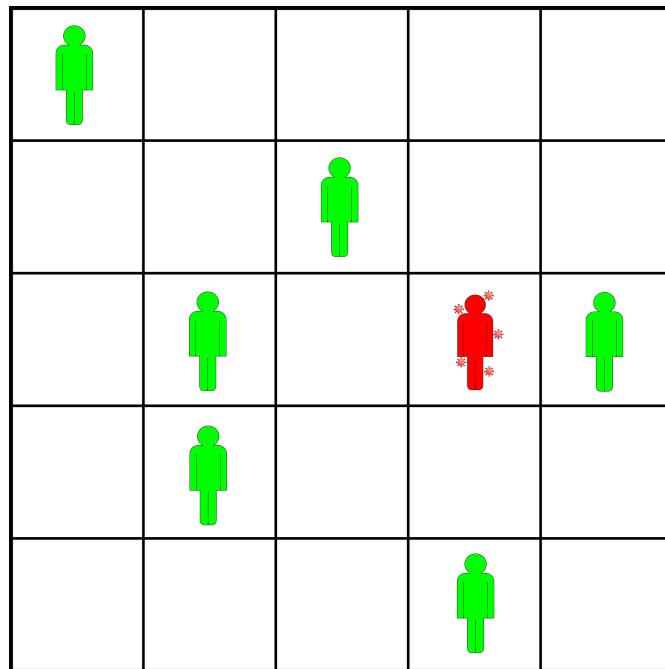


Figure 4.3: grille régulière

Pour toutes configurations initiales, le placement des individus est aléatoire. A partir de là, un certain nombre d'itérations vont se produire permettant de faire évoluer le système.

#### 4.3.1 Itérations

Depuis une configuration, le système peut évoluer par un itération. Une itération est une transition d'un état à un autre du système. Par exemple, depuis l'état initial de la simulation, donc l'état 0 on peut passer dans l'état 1 par une transition qui est la première itération. Cette itération modifie les états de tous les acteurs et leur permet d'effectuer une ou plusieurs actions.

Pour être plus précis, une itération du système consiste en deux phases. La première est de mettre à jour tous les acteurs du système et la seconde est de permettre aux acteurs de se déplacer dans l'espace. La phase de mise à jour comprend l'actualisation de tous les acteurs ainsi que d'effectuer toutes les interactions entre les différents acteurs. La phase de mouvement permet uniquement aux individus de se déplacer sur la grille. Le détail de ces deux phases est explicité plus loin.

En réalité les deux phases sont entrelacées et ne se déroulent donc pas l'une après l'autre. Nous ne mettons pas à jour tous les acteurs puis les faisons tous se déplacer. La méthode utilisée est d'itérer parmi tous les acteur du système et pour chacun d'eux, nous

commençons par faire l'étape de mise à jour puis on fait déplacer l'acteur si possible. Par conséquent, le processus est purement séquentiel avec les acteurs se mettant à jour et se déplaçant les uns après les autres. L'ordre choisi pour la sélection des acteurs est aléatoire.

La simulation se termine après le déroulement d'un certain nombre d'itérations défini. La notion d'itération représente une certaine évolution dans le temps d'un système. C'est la seule représentation temporelle de la simulation.

#### 4.3.2 Mouvements des acteurs

La phase de mouvement permet simplement aux acteurs du système de physiquement se déplacer dans le domaine. Chaque acteur est sur une cellule caractérisée par des coordonnées et peut à cette phase bouger en changeant de cellule. Tous les acteurs ne peuvent pas se déplacer, seuls l'espèce hôte peut se mouvoir. Les agents pathogènes ne peuvent pas se déplacer par eux mêmes. Par conséquent, chaque individu peut se déplacer à chaque itération et ceci d'une seule cellule. C'est-à-dire qu'un individu ne peut bouger que d'une cellule à la fois et ne peut pas se déplacer en diagonale. La portée de déplacement est illustrée ci-dessous et un individu a l'impossibilité de se déplacer sur une cellule si celle-ci est déjà occupée par un autre individu.

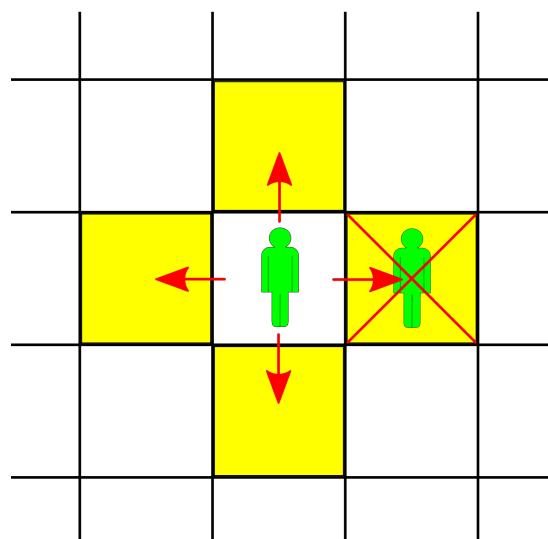


Figure 4.4: grille régulière

Sur cet exemple, l'individu sur la cellule centrale souhaite se déplacer. Sur les 4 cases possible pour un déplacement à cette itération, 3 sont libres et 1 occupée. Par conséquent notre humain ne peut pas se déplacer vers la droite. Le choix de l'individu pour le

déplacement est aléatoire.

La procédure pour se déplacer est la suivante. Tout d'abord, l'individu choisit aléatoirement une des 4 cellule voisine. Il vérifie ensuite que la cellule choisie soit libre. Si la cellule est libre il s'y déplace, sinon il ne bouge pas à cette itération et par conséquent reste là où il est.

#### 4.3.3 Mise à jour des acteurs

La phase de mise à jour se déroule en deux étapes. La première est d'actualiser l'état interne de l'acteur et se produit dans deux situations. Le premier cas survient lorsqu'un agent pathogène contamine une cellule. Cet agent pathogène n'est dans aucun hôte donc sa survie est incertaine. Par conséquent il se peut que cet agent pathogène meurt. La phase de mise à jour interne de cet acteur sert à déterminer s'il meurt à cette itération ou non. La deuxième situation ou l'actualisation de l'état interne d'un acteur est nécessaire est lorsqu'un hôte est contaminé par un agent pathogène. En effet, à chaque itération les hôtes contaminées peuvent développer une immunité au pathogène les contaminant. Cette vérification s'effectue à chaque itération et consiste à recalculer la compatibilité entre l'individu et l'hôte. Si l'agent pathogène contaminant cet hôte survit à l'étape précédente, il a la possibilité de muter. La deuxième étape de la mise à jour d'un acteur est l'analyse du voisinage. A chaque itération, les acteurs analysent leur voisinage et interagissent avec les acteurs à proximité. Tous les différentes interactions entre acteurs sont explicités plus loin.

Par conséquent, le système peut évoluer de deux manières différentes. Premièrement, les acteurs appartenant à l'espèce hôte peuvent se déplacent à chaque itération, modifiant leur position sur la grille. Deuxièmement les états de tous les acteurs du système peut changer grâce aux interactions entre acteurs mais aussi à l'actualisation de leurs états internes.

Les interactions se basent sur la notion de voisinage. Un acteur ne prend en considération durant sa mise à jour que les autres acteurs dans les cases voisines.

### 4.4 Interactions

Une interaction est réaction réciproque d'un acteur sur un autre. Les interactions se produisent entre les différents acteurs du système et permettent de faire évoluer le système. La première phase de toute interaction est la notion de **\*\*charge virale\*\***. La charge virale est un paramètre fixe du modèle qui détermine la probabilité qu'une transmission soit possible entre deux acteurs. Autrement dit, il s'agit d'un facteur déterminant le niveau de contagion de l'espèce des pathogènes. Par conséquent, une faible charge virale signifie que les individus contaminés sont peu contagieux et ont donc peu de chance

de transmettre leur agent pathogène. A l'opposé, une grande charge virale signifie que les hôtes contaminés sont très contagieux et risquent de contaminer d'autres individus sains rapidement.

Si la transmission est possible entre deux acteurs, cela ne suffit pas pour contaminer un individu sain. Le deuxième facteur est le calcul de la compatibilité entre les acteurs par la notion de diversité. La notion de diversité est détaillée plus loin.

#### 4.4.1 Interactions sur la même cellule

Le premier groupe d'interaction est celui des collisions d'acteurs sur la même cellule. Étant donné que deux humains ne peuvent pas se trouver sur la même cellule à la même itération, nous analyserons que le cas où un humain se trouve sur une cellule déjà contaminée par un agent pathogène. Il y a deux cas de collisions possibles dans toutes simulations.

1. Le premier cas se produit quand un hôte potentiel sain se trouve sur une cellule déjà contaminée. Dans cette configuration, un individu sain se retrouve en contact direct avec un agent pathogène contaminant une cellule et risque donc d'être contaminé et de devenir hôte de cet agent pathogène. Il existe deux manières pour contaminer un humain initialement sain, ceci est la première.
2. Le second cas se produit lorsqu'un hôte déjà contaminé se retrouve sur une cellule déjà contaminée. L'agent pathogène contaminant la cellule n'a aucun impact sur l'individu car ce dernier est déjà l'hôte d'un autre agent pathogène. Par conséquent, les deux acteurs n'ont aucune influence l'un sur l'autre dans cette configuration.

#### 4.4.2 Interactions sur cellules voisines

Le deuxième type d'interactions se produit entre des acteurs qui sont spatialement proches, sur des cellules différentes. Afin de permettre aux acteurs d'interagir sans être sur la même cellule, ils doivent appartenir à leur voisinage respectif. Par exemple, deux individus sur deux cellules adjacentes interagissent.

1. Dans le cas où nous avons deux hôtes potentiels sains sur deux cellules voisines, l'interaction ne produit aucun résultat. Par conséquent, les individus n'ont pas d'impacts les uns sur les autres.
2. Un cas plus intéressant survient lorsqu'un individu sain rentre en contact avec un autre contaminé. Lors de cette interaction il se peut que l'individu sain soit contaminé par l'individu contaminé. L'agent pathogène contenu dans l'hôte contaminé de droite pourrait avec une certaine probabilité se propager sur l'humain de gauche. Un hôte potentiel sain peut donc être contaminé par un hôte contaminé seulement si l'individu hôte se trouve dans son voisinage. La contamination de l'individu à l'individu est la seconde méthode de contamination d'un être initialement sain.

3. Similairement au cas numéro 1, si deux individus contaminés entrent en contact, aucune interaction ne se produit. Par conséquent, un individu contaminé n'a aucune influence sur un autre individu contaminé.

#### 4.4.3 Contamination de cellule

Une cellule se trouve dans l'état si cet espace a été contaminé par un hôte infecté. En effet, lorsqu'un hôte infecté se déplace il a une certaine probabilité de contaminer l'espace qu'il occupait. Un paramètre fixe du modèle permet de déterminer cette probabilité. Une probabilité paramétrée élevée signifie que les hôtes contaminés contaminent souvent les cellules qu'ils visitent et à l'inverse pour une probabilité faible.

Après un déplacement, l'hôte contaminé à une certaine probabilité d'avoir contaminé la cellule qu'il occupait précédemment. L'agent pathogène contaminant à présent la cellule est le même que celui contenu dans l'individu.

Un cas particulier peut se produire si un hôte contamine un espace déjà contaminé par un autre agent pathogène. C'est-à-dire qu'un hôte contaminé se trouvant sur une cellule préalablement contaminée, essaie de la contaminer à nouveau. On a dit précédemment qu'un hôte déjà infecté n'était pas sensible à un agent pathogène externe contaminant une cellule par contre il se pourrait que notre individu contamine lui aussi cette cellule. Dans ce cas précis nous écrasons l'agent pathogène initialement présent et le remplaçons par une copie de l'agent pathogène de l'hôte.

### 4.5 Diversité génétique

Le travail est orienté sur l'aspect de diversité génétique et ceci dicte l'issue des différentes interactions entre les acteurs. Pour simplifier nous nous contentons d'attribuer une valeur sous forme d'un entier à chaque acteur. Cette valeur se code sur 4 octets donc 32 bits et cette séquence représente le **\*\*code génétique\*\***. Tous les agents du système ont un code génétique sur 4 octets que ce soit des humains ou des agents pathogènes.

La valeur de diversité de chaque acteur est calculé initialement au début de la simulation. Un génome est attribué à l'unique agent pathogène et tous les individus se voient attribuer des génomes en fonction de deux paramètres du système. Le premier paramètre est le génome de référence et le second est le niveau de déviation des génomes. C'est-à-dire qu'il est possible de générer les génomes de tous les individus en suivant une distribution Gaussienne avec comme base le génome de référence. Le deuxième paramètre de diversité est donc une forme de variance.

La notion de diversité intervient lors des interactions entre les acteurs du système. En effet la charge virale permet de déterminer si la transmission du pathogène est possible

mais ce n'est pas suffisant pour contaminer un individu sain. La compatibilité entre un hôte sain et un pathogène est nécessaire à la contamination de l'hôte. Plus généralement, le calcul et l'interprétation de la distance de Hamming est utilisé lors des interactions entre individus et agents pathogènes.

#### 4.5.1 Distance de Hamming

La distance de Hamming est un calcul s'effectuant sur deux séquences de symboles de même longueur. Il s'agit de quantifier la différence entre ces deux séquences par un entier. Un exemple sur deux séquences de 1 octet est donné ci-dessous.

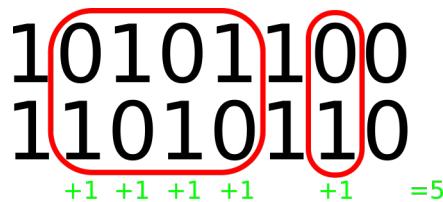


Figure 4.5: Calcul de la distance de Hamming

Dans cet exemple on peut voir en comparant les deux séquences de bits que sur 5 positions les symboles diffèrent. Par conséquent la distance de Hamming entre la séquence 10101100 et 11010110 est égale à 5.

Le calcul de la distance de Hamming s'effectue toujours entre le génome d'un individu et celui d'un agent pathogène et détermine la compatibilité des deux organismes.

Il est ensuite nécessaire d'interpréter la distance de Hamming et de lui donner du sens. En effet, connaître la distance de Hamming entre deux génomes ne dit pas comment le système doit se comporter. Il faut traduire cette information sous une forme utilisable. Un mécanisme de traduction produit une probabilité en utilisant une distance de Hamming. L'idée est de générer une probabilité représentant le plus précisément possible la similarité ou la différence entre les deux séquences. Lorsque le calcul de la distance de Hamming est demandé, l'objectif est de déterminer si une action va avoir lieu ou non. Dans ce cas nous aurions pu utiliser un fonction de seuillage qui fait l'action à partir d'une certaine distance et ne la fait pas autrement. Au lieu de ce mécanismes, nous générerons une probabilité reflétant cette distance et déterminant si l'action se produit ou non.

La distance de Hamming s'interprète de la manière suivante dans le modèle :

- Une faible distance de Hamming donne l'ascendant à l'agent pathogène.
- Un grande distance de Hamming donne l'ascendant à l'individu.

L'idée principale est qu'un pathogène est efficace si son génome est "proche" de celui de l'individu qu'il attaque et inversement, l'agent pathogène est peu efficace si son génome est "éloigné" de celui de l'individu.

Par conséquent, lors d'une interaction entre un individu et un agent pathogène, deux séquences de génome très proches implique une grande probabilité que l'agent pathogène contamine l'individu et s'installe. Par contre, deux séquences très différentes implique une grande probabilité que l'individu résiste à l'agent pathogène et ne soit pas contaminé.

Le calcul de la distance de Hamming s'effectue dans deux situations. A chaque fois que ce calcul est appelé, c'est pour déterminer l'issue d'un contact entre un pathogène et un individu contaminé ou non. La distance de Hamming traduite en probabilité dicte le comportement de l'interaction. Il existe donc deux cas possibles nécessitant le calcul de la distance de Hamming :

1. Un individu sain est en contact avec un voisin contaminé ou se trouve sur une cellule déjà contaminée par un agent pathogène. Dans cette situation, l'individu sain est exposé à cet agent pathogène, en plus du facteur de charge virale il reste à déterminer la compatibilité de ces deux organismes et de déterminer si notre individu devient hôte du pathogène ou reste sain. La compatibilité se calcule à partir de la distance de Hamming traduite en probabilité.
2. La deuxième situation où la distance de Hamming est utilisée est pour déterminer l'évolution de l'état d'un hôte contaminé. En effet un individu contaminé est l'hôte d'un agent pathogène et chacun essaie de prendre le dessus sur l'autre organisme. A chaque itération la distance de Hamming est recalculée entre les deux génomes afin de déterminer si l'humain s'immunise ou si le pathogène reste dans son hôte. Dans le cas où l'agent pathogène reste il se peut qu'il tue son hôte.

## 4.6 Immunisation et résistance naturelle

Le modèle intègre un principe d'immunisation et de résistance naturelle. Ces notions ne s'appliquent qu'aux organismes de l'espèce hôte.

Nous parlons de résistance naturelle quand un individu n'est pas affecté par un certain agent pathogène pour des questions génétiques. C'est-à-dire que l'individu ne développe pas d'immunité mais est insensible à l'agent pathogène. Le modèle ne considère pas ce mécanisme comme immunité mais uniquement comme une insensibilité, une résistance. Cette insensibilité de certains individus à certains agents pathogènes est déterminée par la distance de Hamming des génomes. Si deux acteurs interagissant ont

une très mauvaise compatibilité, c'est-a-dire que l'agent pathogène a très peu d'emprise sur l'hôte alors l'individu est naturellement résistant à cet agent pathogène.

Contrairement à la résistance naturelle, l'immunisation est une résistance acquise, c'est-à-dire que l'individu contaminé a développé une immunité en présence d'un agent pathogène. C'est la situation où l'individu reste contaminé un certain temps tout en contenant l'agent pathogène puis développe une immunité en combattant ce dernier. La notion de temps dans le modèle est donnée par les itérations, par conséquent si un hôte contaminé porte un agent pathogène pendant un certain nombre d'itérations alors cela se traduit par une certaine durée dans l'état contaminé.

La différence entre immunisation et résistance naturelle est donc une différence temporelle. Dans le cas de la résistance naturelle, l'individu est immédiatement débarrassé du pathogène car non affecté. Par contre dans le cas de l'immunisation, il se déroule un certain temps (itérations) avant que l'individu ne se débarrasse de l'agent pathogène.

Le modèle intègre les immunisations comme une liste d'attributs pour chaque individu. C'est-à-dire que chaque individu peut obtenir des immunités aux pathogènes qui les ont contaminés. Pour des questions de simplicité, les immunités des individus ne les protègent que contre les agents pathogènes déjà rencontrés. Par exemple, si un individu développe une immunité contre un agent pathogène, l'individu ne sera plus affecté par ce génome d'agent pathogène dans le futur. Les individus du système ne peuvent donc pas développer des immunités à des groupes similaires de pathogènes, c'est des immunités au cas par cas. Pour ce qui est des résistances naturelles, il n'y a pas d'attributs ou de valeur à mémoriser. La résistance naturelle se détermine qu'à partir des génomes des acteurs.

L'opération d'immunisation ne s'effectue que dans le cas où un hôte est contaminé par un agent pathogène. Le calcul de la distance de Hamming traduite en probabilité détermine si à cette itération l'individu s'immunise ou non. Si il ne s'immunise pas alors le pathogène reste, sinon l'individu se débarrasse de son agent pathogène et intègre le génome du pathogène comme immunité. Il s'agit ici d'immunité car l'individu contenait le pathogène avant de s'en débarrasser. Le mécanisme de résistance naturelle fonctionne exactement de la même manière sauf que le rejet de l'agent pathogène s'effectue immédiatement. Dans ce cas le modèle ne considère pas cela comme immunité et rejette le pathogène sans pour autant développer une immunité à son génome.

## 4.7 Mutations

Le modèle ne représente que des situations qui se déroulent sur le court terme. C'est-à-dire que le temps d'une simulation à l'échelle de la vie d'un individu est assez faible. Par

conséquent, les individus ne peuvent pas muter car nous estimons que durant toute la simulation, les individus ne changent pas de code génétique. Par contre les agents pathogènes peuvent évoluer rapidement, ce sont les seuls à pouvoir muter dans le modèle. Le modèle intègre un paramètre déterminant la vitesse à laquelle les agents pathogènes du système peuvent muter.

La mutation d'un agent pathogène se caractérise par la modification de son génome. En fonction du paramètre de la simulation, les différents agents pathogènes du système changent plus ou moins leur séquence de code génétique ce qui modifie les issues des interactions avec les autres acteurs.

Tous les agents pathogènes ne peuvent muter. Il est nécessaire que l'agent pathogène soit dans un hôte afin de muter. Par conséquent, un pathogène contaminant une cellule ne peut pas muter. Sans un hôte il est impossible pour un agent pathogène de muter. Les agents pathogènes contenus dans des hôtes peuvent muter à chaque itération mais ont une certaine probabilité de le faire. Cette est déterminée par un paramètre fixe du modèle.

## 5 Implémentation

Le modèle se construit sur différents niveaux d'abstractions. L'exécution d'une simulation s'effectue en plusieurs étapes, du plus général au plus détaillé.

### 5.1 Main

Le main est la structure du plus haut niveau du modèle est sert d'interface pour l'utilisateur. A ce niveau, il est possible d'instancier plusieurs simulations avec des paramètres choisis. Afin de pouvoir exécuter une simulation il est nécessaire d'y ajouter un fichier de configuration contenant les paramètres de la simulation ainsi que de lui fournir un générateur de nombres aléatoires.

Par conséquent, chaque simulation demande deux arguments pour se lancer. Le premier est un chemin vers un dossier dans lequel se trouve le fichier de configuration. Le deuxième argument est générateur pseudo-aléatoire permettant la génération de nombres aléatoires indispensables à la simulation.

Le chemin vers le dossier est la racine de la simulation. Dans ce dossier racine doit se trouver le fichier de configuration. Le fichier de configuration doit avoir le nom suivant : **config.txt**. A la fin de la simulation, le programme écrit les résultats dans un sous-dossier à la racine. Ces résultats sont des fichiers CSV contenant des informations sur la simulation.

L'avantage de dédier un générateur pseudo-aléatoire à chaque simulation est de permettre de ré-exécuter les simulations en obtenant toujours les mêmes résultats, ce qui confère à l'algorithme la propriété d'idempotence.

Un ficher de configuration est un ficher texte contenant des lignes formées de mots clefs accompagnée de leur valeur. Un analyseur syntaxique parse le fichier puis les tokens sont analysés par un analyseur lexical qui s'occupe d'assigner les variables du modèles avec les valeurs lues dans le fichier de configuration.

La syntaxe du fichier de configuration doit être respectée. Les valeurs peuvent être modifiées par contre les noms doivent être identiques. L'analyseur lexique n'est pas sensible aux espaces insérés entre les noms, le "=" ainsi que les valeurs.

---

```

1 TAILLE_SYSTEME = 894
2 NOMBRE_INDIVIDUS = 100000
3 ITERATIONS = 150
4 RERUN_LIMIT = 1000
5 FAIL_SEUIL = 10
6 GENOME_INIT_I = 0
7 GENOME_DIVERSITY_I = 0
8 GENOME_INIT_AP = 0
9 VITESSE_MUTATIONS_AP = 0
10 CHARGE_VIRALE = 1
11 PARAMETRE_FONCTION = 1
12 CELLULE_AP = 0
13 SURVIE_AP = 0
14 NOMBRE_MOUVEMENT = 1
15 PERFECT_MIX = false
16 TEMPS_AVANT_IMMUNITE = 1
17 IMMUNITE_MECANISME = false
18 RESISTANCE_MECANISME = false

```

---

Voici la structure d'un fichier de configuration. Chaque ligne est composée d'un paramètre avec une valeur associée.

## 5.2 Initialisation

La phase d'initialisation prépare le système pour une simulation. La première étape est de lire le fichier de configuration et d'assigner les valeurs lues aux variables de la simulation. Ensuite il faut allouer de la mémoire pour tous les objets du système. Un objet du système est un acteur donc il s'agit soit d'un individu, soit d'un pathogène. Le nombre d'individus du système est déterminé par le paramètre "NOMBRE\_INDIVIDUS" et une liste de pointeurs sur ces individus est créée. Cette liste est de taille fixe et contient tous les individus du système. Contrairement aux individus, les agents pathogènes ne sont pas contenus dans une liste fixe à cause de leur nombre pouvant changer. En effet, un agent pathogène est un objet à part entière si et seulement si il contamine une cellule. Nous utilisons donc une liste doublement chaînée pour stocker ces objets. Cette structure permet de supprimer un élément au milieu de la liste sans créer de trou.

Le paramètre "TAILLE\_SYSTEME" nous indique la taille de la grille. La grille allouée est une matrice de pointeurs pouvant pointer sur les acteurs qui sont les objets du système.

Après avoir créé les structures de données il faut générer les génomes des individus. Le génome initial de référence est donnée par le paramètre "GENOME\_INIT\_I". Ce dernier sert de base pour la génération des génomes des individus. Le paramètre "GENOME\_DIVERSITY\_I" détermine le niveau de déviation par rapport aux génome de référence. Par conséquent, pour attribuer des génomes aux individus on part du

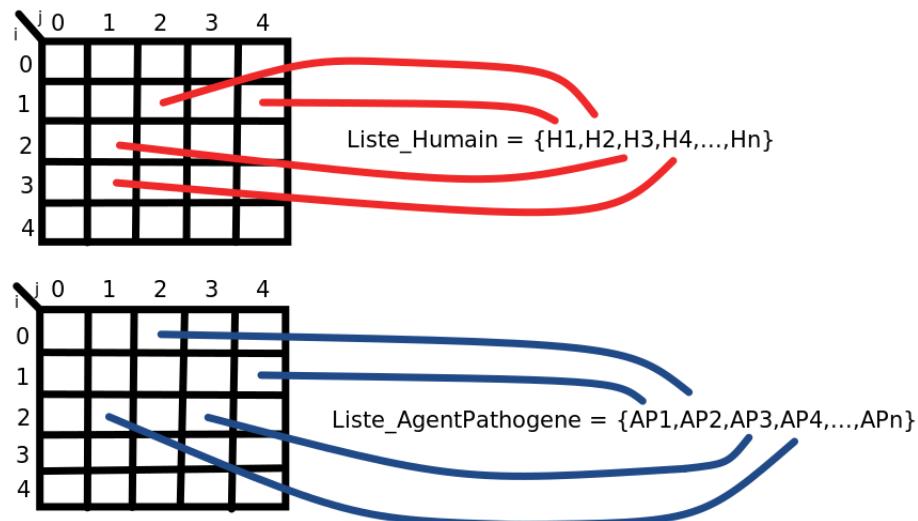


Figure 5.1: Matrices de pointeurs

génome de référence et modifions aléatoirement un nombre de bits déterminé par le paramètre "GENOME\_DIVERSITY\_I". Par exemple, avec GENOME\_DIVERSITY\_I = 1, tous les individus auront le génome de référence avec un bits choisi aléatoirement dans la séquence qui sera modifié. Ce mécanisme permet d'obtenir des génomes avec un certain degré de proximité à une certaine valeur choisie. Le modèle n'intègre pas de mécanisme pour générer des génomes pour les agents pathogènes car la simulation ne commence qu'avec un seul agent pathogène qui porte le génome défini par "GENOME\_INIT\_AP".

L'étape suivante est de disposer tous les individus sur la grille et ceci aléatoirement. Finalement il faut contaminer un premier individu avec le premier pathogène défini par le paramètre "GENOME\_INIT\_AP".

### 5.3 Exécution de la simulation

Une grande partie des calculs de la simulation se font à cette étape. Cette phase s'effectue un nombre de fois équivalent à la valeur du paramètre "ITERATIONS". Chaque itération sert à actualiser, à un moment donné le système en entier. La première partie de cette étape est la prise de mesures et l'écriture des mesures dans les fichier CSV. Les données récoltées sont le nombre de contaminés, le nombre d'agents pathogènes différents ainsi que le nombre d'individus ayant au moins une immunité. Toutes les mesures du système qui doivent s'effectuer à chaque itération se font à cette étape. Faire des mesures au fur et à mesure de la simulation nous évite de devoir sauvegarder toutes les données.

La prochaine étape est la construction d'une permutation des indices de la liste contenant les individus. Pour que le système soit aléatoire il est important de ne pas actualiser tous les individus dans le même ordre à chaque itération. C'est la raison pour laquelle

nous permutions les indices de la liste des individus à chaque itération et la parcourons en suivant les indices permutés. Ceci permet d'actualiser tous les individus du système dans un ordre aléatoire.

Vient ensuite les phases de mise à jour et actions des acteurs du système. Le type d'acteur le plus simple est les agents pathogènes contaminant des cellules. Ces derniers sont détachés des individus et sont donc des objets à part entière. L'actualisation de ces agents pathogène s'effectue en une seule étape. Nous parcourons tous les agents pathogènes contaminant des cellules (l'ordre ici n'est pas important) et déterminons si les pathogènes survivent à cette itération. La probabilité de survie d'un tel agent pathogène est déterminé par le paramètre "SURVIE\_AP". Ce facteur donne la probabilité qu'un agent pathogène contaminant une cellule survive à cette itération.

Finalement, la dernière étape de la phase est l'actualisation de tous les individus du système. Cette étape est de loin la plus coûteuse en temps. En premier il s'agit de parcourir un à un tous les individus en respectant les indices permutés. L'actualisation d'un individu se fait en deux étapes. La première étape consiste à calculer les interactions et mettre à jour son état interne. La seconde est d'effectuer un déplacement.

La première chose à faire lorsque nous souhaitons actualiser un individu est de regarder si cet individu est contaminé. Si l'individu est déjà contaminé il est inutile de calculer les interactions avec ses voisins étant donnée qu'il est hôte d'un pathogène. Il nous reste donc à actualiser son état interne. Dans cette situation, notre individu n'a que trois issues. La première est d'être naturellement résistant au pathogène. La seconde est de l'immuniser au pathogène et la troisième est de conserver le pathogène. La distinction entre résistance naturelle et immunisation n'est qu'une question de temps. Il s'agit en réalité exactement du même mécanisme mais à partir d'un certain temps en étant contaminé, nous déterminons que l'individu s'immunise. Chaque individu compte le nombre d'itération depuis lequel il est contaminé. Si l'individu à une certaine itération se débarrasse du pathogène, le compteur déterminera si il s'agit d'une immunité ou d'une résistance naturelle.

Le paramètre booléen "IMMUNITE\_MECANISME" du modèle détermine si le mécanisme d'immunisation est actif ou non. Il en est de même pour le paramètre booléen "RESISTANCE\_MECANISME". Désactiver ces mécaniques permet de simuler des systèmes où l'immunisation est impossible pour quelques raisons. Il est aussi possible de déterminer le seuil temporelle différenciant la résistance naturelle de l'immunisation avec le paramètre "TEMPS\_AVANT\_IMMUNITE". C'est-à-dire que cette variable détermine le temps minimal nécessaire pour que le rejet du pathogène soit considéré comme immunité.

Pour déterminer si un hôte contaminé rejette ou non son agent pathogène à une certaine itération, nous utilisons une fonction calculant la compatibilité entre le pathogène et l'individu. La première étape pour calculer la compatibilité entre deux organismes est d'évaluer la distance de Hamming entre les deux séquences de génomes. L'entier résultant doit ensuite être converti en probabilité car l'individu a une certaine probabilité de se débarrasser du pathogène à cette itération. La conversion s'effectue par une fonction dont une variable est définie par le paramètre du modèle "PARAMETRE\_FONCTION".

$$f(d) = \frac{d^{\text{PARAMETRE\_FONCTION}}}{32^{\text{PARAMETRE\_FONCTION}}} = \left(\frac{d}{32}\right)^{\text{PARAMETRE\_FONCTION}}$$

Le paramètre  $d$  est la distance de Hamming entre les deux génomes des acteurs. Cette fonction traduit la distance de Hamming en une probabilité car  $d$  ne peut varier que entre  $[0, 32]$ .

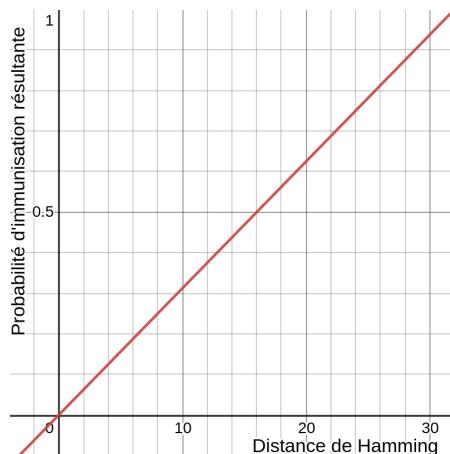


Figure 5.2: paramètre = 1

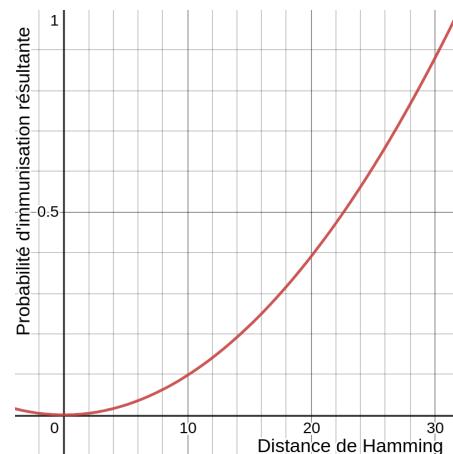


Figure 5.3: paramètre = 2

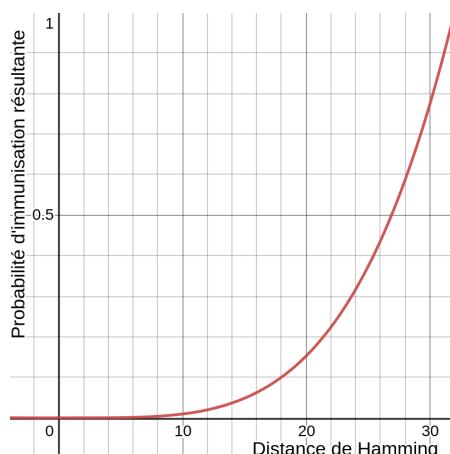


Figure 5.4: paramètre = 4

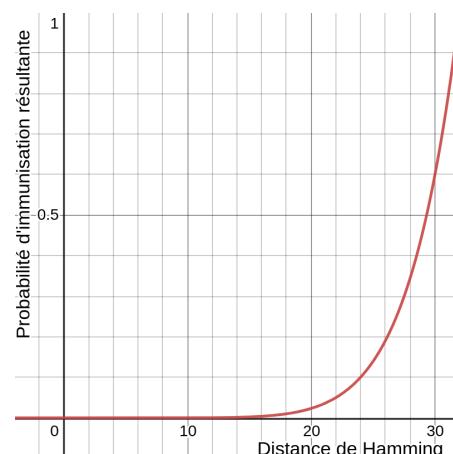


Figure 5.5: paramètre = 8

Par conséquent les images de notre formules sont bornées entre  $[0, 1]$ . Le paramètre de fonction permet de modifier l'allure de la courbe sans modifier les bornes. Une petite valeur "PARAMETRE\_FONCTION" traduit de grandes probabilités d'immuniser

l'individu et inversement une grande valeur du paramètre traduit en de faibles probabilités d'immunisation.

Lorsqu'un individu est décrété naturellement résistant à un pathogène, il se contente uniquement de s'en débarrasser. Par contre si l'individu s'immunise au pathogène, le génome de ce dernier est enregistré dans les attributs de l'individu.

Comme expliqué ci-dessus, à chaque itération, chaque individu contaminé a une chance de se débarrasser de son pathogène. Si l'hôte contaminé n'y parvient pas, alors le pathogène a la possibilité de muter. Le facteur de mutation de la simulation est donné par le paramètre "VITESSE\_MUTATIONS\_AP". Ce paramètre borné entre [0, 1] détermine la probabilité qu'a un pathogène de muter dans un individu. En cas de mutation, un agent pathogène complémente aléatoirement un seul bit de sa séquence de génome.

Le processus de mise à jour pour les individus non contaminés est bien différent. Ces derniers n'intègrent pas de pathogène par conséquent nous nous intéressons aux interactions avec les autres acteurs. Un individu sain est soumis à deux types d'interactions, la première est l'interaction avec des agents pathogènes contaminant des cellules de la grille et la seconde est l'interaction avec des individus dans le voisinage.

Les deux types d'interactions suivent la même méthodologie. La première étape de toutes interactions est donné par le paramètre "CHARGE\_VIRALE". Le paramètre borné entre [0, 1] détermine la probabilité qu'une interaction se produise entre deux acteurs en contact. La deuxième étape est la vérification des immunités. En effet, si l'individu est déjà immunisé au pathogène le contaminant, rien ne se produit. Sans protection l'individu devient contaminé par le pathogène.

L'actualisation d'un individu se termine par une série de mouvements dans l'espace. A chaque itération un individu peut se déplacer un nombre de fois égal au paramètre "NOMBRE\_MOUVEMENT". Par conséquent un individu peut se déplacer de une cellule le nombre de fois décrit par le paramètre. Le choix de la direction du déplacement est déterminé avant la vérification des cellules disponibles. En cas d'échec de déplacement dû à la non disponibilité de la cellule destination, le mouvement non effectué est tout de même comptabilisé. Cela signifie que si le paramètre de mouvement est défini à 10, les individus se déplacent à chaque itération d'un nombre de cellules variant entre [0, 10] en fonction de la disponibilité.

Un individu contaminé se déplaçant a une chance de contaminer l'espace qu'il quitte. La probabilité de réaliser cet événement est donnée par le paramètre "CELLULE\_AP"

borné de  $[0,1]$ . En cas de contamination de cellule un nouvel objet agent pathogène est créé et porte le génome du pathogène de l'individu contaminé.

## 5.4 Prise des mesures

La phase des mesures s'effectue après le déroulement de la simulation. Toutes les mesures nécessitant que la simulation ait abouti s'exécutent dans cette section. Comme vu dans la partie précédente, certaines mesures doivent s'effectuer au fil de la simulation.

La seule mesure du modèle de cette catégorie est le calcul du nombre de fois que les individus ont été contaminés. Chaque individu du système compte le nombre de fois où il a été contaminé par des agents pathogènes différents. Est compté comme contamination tout cycle débutant par la contraction du pathogène jusqu'à l'immunisation à ce dernier.

## 5.5 Fermeture

La phase de fermeture de la simulation effectue deux actions. La première est de fermer tous les fichiers CSV et la seconde est de libérer tous les espaces mémoire alloués par la simulation.

## 6 Objectifs

Nous cherchons à produire un modèle numérique cohérent qui fournirait des résultats que l'on a pu observer par le passé. Le modèle construit son propre monde mais en se basant sur une certaine réalité. Par exemple, la notion de génome implémentée dans ce modèle n'est pas représentative de la réalité. En effet le génome d'un individu n'est pas aussi simple qu'un seul entier. Il s'agit donc ici de créer un abstraction simplifiée de fonctionnements réels afin de l'implanter dans le modèle et donc de simuler un processus réel et très complexe. Cette simplification permet une analyse plus aisée des résultats. En effet, les corrélations de causes à effet sont plus facilement interprétables grâce à la simplicité du modèle.

La validité du modèle peut aussi se tester en le soumettant à des cas simples dont on peut inférer les résultats. Connaissant à l'avance l'orientation de la simulation, nous pouvons vérifier que le programme fournisse bien des résultats conformément aux attentes. Pour ce faire on peut se référer au modèles compartimentaux en épidémiologie pour tester le modèle numérique. A quel point le modèle reflète la réalité que nous pourrions observer dans les modèles compartimentaux ? La validité du modèle sur des cas connus est nécessaire pour conclure de nouveaux résultats.

Le but d'une simulation est de fournir des résultats. Quels résultats pouvons nous extraire d'une simulation ? Nous pouvons déterminer des différences dans les résultats en faisant varier les paramètres du système. Étant donné que le modèle possède sa propre réalité, nous ne pouvons extraire des informations qu'en comparant des résultats d'une simulation avec une autre. Il s'agit donc de mesurer et quantifier des différences de comportements en faisant varier un ou plusieurs paramètres du système. La recherche se concentre sur la notion de diversité, introduite dans le modèle par les génomes des acteurs. Par conséquent, le facteur de diversité est le paramètre principal étudié. Quel est l'impact de la diversité d'une population sur la propagation d'une épidémie ? L'objectif est de déterminer l'impact qu'a la diversité des organismes sur la propagation d'épidémies. La variation d'autres paramètres peuvent aussi être étudié comme par exemple la vitesse de mutation des agents pathogènes. Il s'agit donc d'illustrer et de quantifier l'impact qu'a ces paramètres sur l'émergence d'événement de grande taille comme les pandémies.

De bonnes performances du modèles sont nécessaire pour effectuer des simulation de grande taille. En fonction des performances du modèle, quelles sont les ordres de

grandeur des simulations effectuées ? En effet le travail cherche à observer des événement à grande échelle que le modèle doit supporter.

L'objectif du travail est de fournir un modèle numérique capable de faire des prédictions de propagation de pandémies dans une population donnée en se basant sur la notion de diversité. Pour quel type de problème pourrons nous faire de la prédition ? Le modèle ne se limite pas à deux espèces en particulier mais à deux catégories d'espèces. Par conséquent on peut appliquer le modèle à un grand nombre de cas.

Le dernier objectif de la recherche est de pouvoir faire de la prédition explicative. Contrairement à la prédition anticipative, notre modèle pourrait permettre d'expliquer l'émergence ou non d'événement à grande échelle en fonction de différents paramètres. Il s'agirait de pouvoir expliquer les facteurs qui ont permis la propagation d'un pathogène.

## 7 Modèles Compartimentaux

Les modèles compartimentaux sont une catégorie de modèles mathématiques permettant de représenter l'évolution de maladies infectieuses au sein d'une population. Ces modèles épidémiologiques se basent sur la notion de classes épidémiologiques et de transitions entre ces différentes classes. Nous appelons ces classes des **compartiments** et les transitions entre les classes les **règles**. Un compartiment est un état possible dans lequel les acteurs du système peuvent se trouver. Chaque acteur du système se trouve à tous moment dans un compartiment et ce dernier définit son état. Une règle est une fonction de transition entre les compartiments. Les acteurs changent de compartiments lorsque leur état change. Nous pouvons donc définir des règles permettant de changer de compartiment.

Il existe 7 type de compartiments différents. Le premier compartiment est  $S$  et représente tous les individus sains du système. Le second est  $I$  et désigne tous les acteurs contaminés. Le troisième est un compartiment de transition entre  $S$  et  $I$  et représente une période de latence. En effet, suivant les maladies il se peut qu'un acteur sain ne devienne pas directement contaminé après un contact positif. Le quatrième compartiment est  $D$  et symbolise les individus morts. Le cinquième est  $R$  pour "recovered" et inclut tous les individus qui ont développé une immunité suite à une période infecté. Le sixième compartiment est le compartiment  $M$  et reflète ce que nous avons appelé "résistance naturelle" dans le modèle. C'est-à-dire le groupe d'individus naturellement insensible au pathogène. Finalement le dernier compartiment est  $C$  et illustre les individus asymptomatiques, c'est-à-dire les hôtes contaminés qui ne présentent aucun symptômes et semblent donc sains.

Le modèle actuel n'implémente pas tous les compartiments mais uniquement trois d'entre eux. Nous n'étudierons que les compartiments  $S$ ,  $I$  et  $R$ . Les règles dépendent du modèle et sont toutes explicités dans la description du modèle.

L'objectif de ce chapitre est valider le notre modèle sur les bases des modèles compartimentaux. Il s'agit donc de déterminer les paramètres du modèle afin d'observer des résultats suivant ces modèles. Les modèles compartimentaux sont des modèles mathématiques représentant l'évolution de pandémies et on prouvé leur validité par le passé. Pour tester notre implémentation du modèle, il est essentiel de le comparer à des résultats déjà prouvés. Dans ce chapitre, nous allons essayer de reproduire les

résultats fournis par ces modèles bien connus et allons appliquer des variations sur les paramètres du modèle pour en constater l'impact.

## 7.1 Modèle SI

Le modèle SI est le modèle épidémiologique le plus simple. Nous avons deux classes d'individus, les individus sains ( $S$ ) et les individus contaminées ( $I$ ). Les deux compartiments sont les seuls états possibles pour les acteurs du système. Cela signifie que les individus ne peuvent pas s'immuniser ou guérir.

Initialement tous les individus sont dans le compartiment  $S$  car tout le monde est sain. Pour lancer la simulation, nous infectons un individu. Par conséquent il passe dans le second compartiment  $I$ . Il s'agit ensuite d'étudier l'évolution du système.

Le modèle SI est décrit mathématiques par les équations différentielles ordinaires suivantes :

$$\frac{dS}{dt} = -\frac{\beta SI}{N} \frac{dI}{dt} = \frac{\beta SI}{N}$$

La variable  $N$  comptabilise tout la population d'individus de tous les compartiments confondus, par conséquent  $N = S + I$ . Le facteur  $\beta$  est une valeur définie entre 0 et 1 et déterminant le taux d'infection du modèle.

Les équations différentielles s'implémentent dans python de la manière suivante :

```
1 S, I = S - beta * ((S * I / N)), I + beta * ((S * I) / N)
```

Python intègre un moyen de modifier simultanément plusieurs variables utilisé ci-dessus. Le compartiment  $S$  du modèle ne fait que décroître et au contraire le compartiment  $I$  ne fait qu'augmenter afin de conserver le total  $N = S + I$ . La modification du nombre d'individus par compartiment s'effectue en un certain nombre de pas représentant un écoulement dans le temps. Dans notre simulation ce sont les itérations qui font office de représentation temporelle.

Un modèle SI avec une population de 1000 individus, 999 dans le compartiment  $S$  et 1 dans le compartiment  $I$  avec un taux  $\beta = 0.3$  et déroulé sur 50 itérations, nous obtenons le graphique suivant.

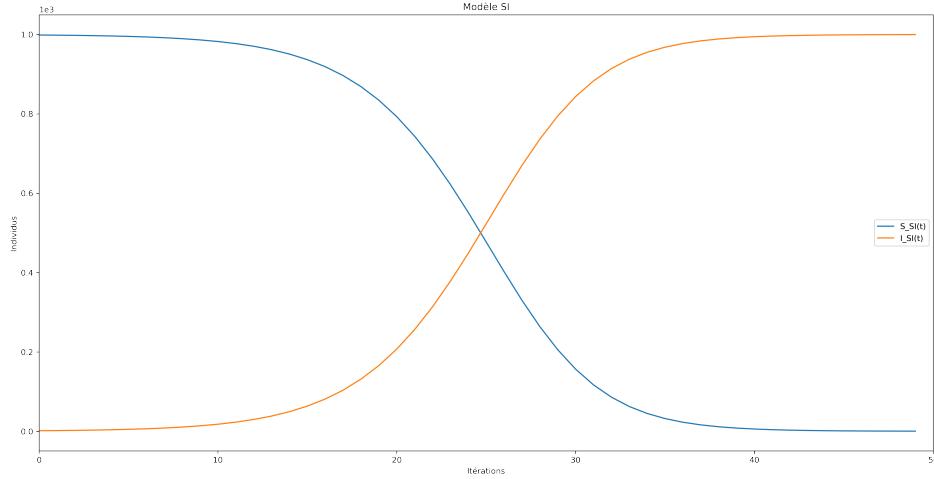


Figure 7.1: Modèle SI

## 7.2 Modèle SIR

Les modèles compartimentaux SIR sont comparables aux modèles SI sauf qu'ils ont un compartiment supplémentaire  $R$  incluant tous les individus guéris. Initialement les individus commencent dans le compartiment  $S$  et passent dans le compartiment  $I$  si ils sont contaminés. Après une certaine période étant contaminés ils peuvent développer une immunité au pathogène et guérir. Tout individus développant une immunité à la maladie passe dans le dernier compartiment  $R$ . Ce modèle possède les mêmes paramètres que le modèle SI avec l'ajout d'une variable  $\gamma$  comprise entre 0 et 1 et définissant le taux de guérison.

Le modèle ayant trois compartiments, il est décrit par trois équations différentielles ordinaires.

$$\frac{dS}{dt} = -\frac{\beta SI}{N} \quad (7.1)$$

$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I \quad (7.2)$$

$$\frac{dR}{dt} = \gamma I \quad (7.3)$$

Le modèle mathématique est très similaire au précédent. La différence vient du fait que certains individus du compartiment  $I$  migrent vers  $R$  avec une certaine probabilité. Par conséquent, une quantité est tronquée à l'ajout du compartiment  $I$  pour le compartiment  $R$ .

Comme pour le point précédent, python implémente les trois équations différentielles de la manière suivante :

```
1 S, I, R = S - beta * ((S * I) / N), I + beta * ((S * I) / N) - gamma * I, R + gamma * I
```

Les trois compartiments sont mis à jour à chaque pas de l'algorithme et ceci simultanément.

Une représentation graphique sur 100 itérations avec les paramètres :  $N = 1000, S = 999, I = 1, R = 0, \beta = 0.3, \gamma = 0.06$  :

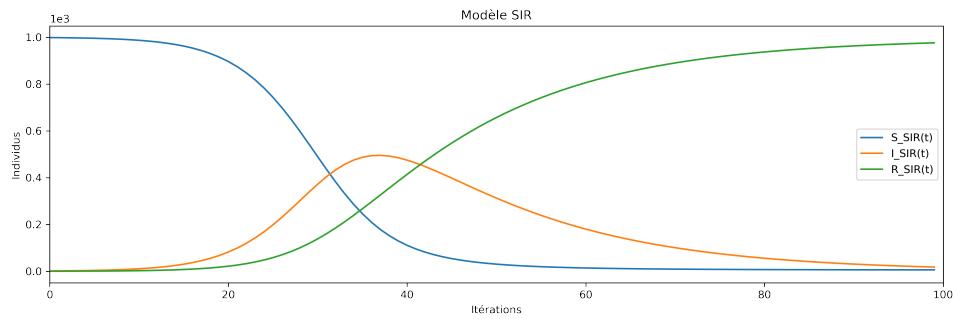


Figure 7.2: Modèle SIR

### 7.3 Simulations de références

Les modèles SI et SIR ont été prouvés par le passé et ont par conséquent une certaine validité. L'objectif ici est de valider notre modèle en le comparant aux résultats des modèles SI et SIR. Il s'agit donc de paramétrier notre modèle afin de le faire correspondre aux courbes des modèles valides déjà existant.

Notre modèle paramétré et validé par SI et SIR peut servir de référentiel pour des analyses. Cette simulation de base nous permettrait de quantifier les différences dans les résultats avec d'autres simulations paramétrées différemment. En effet la simulation crée son propre monde par conséquent un résultat seul ne révèle aucune information. Pour déduire des résultats de simulations il faut les comparer entre eux. C'est la raison pour laquelle nous établissons une simulation de référence validée par SI et SIR.

Afin d'établir une référence nous pouvons paramétrier les modèles SI et SIR directement ainsi que les paramètres de notre modèle. Les paramètres de notre modèles doivent être cohérent et les plus globaux possible.

### 7.3.1 Simulations de SI

Dans ce chapitre nous essayons de simuler des situations décrites par le modèle SI et d'imiter son comportement. Notre modèle est donc paramétré en conséquence pour adopter les mêmes comportements que le modèle SI. L'objectif de cette section est de valider le modèle construit en montrant les similitudes dans les résultats avec le modèle SI déjà prouvé.

Les modèles compartimentaux n'ont pas de notion d'espace, c'est-à-dire que les individus du système n'ont pas de position géographiques. Ces modèles considèrent tous les individus sont en contact les uns avec les autres et ceci atténue par le facteur  $\beta$ . Étant donné que le modèle implémenté possède une notion d'espace, il a fallu l'adapter afin qu'il satisfasse les conditions des modèles comparatimentaux. Deux solutions ont été trouvées pour palier à ce problème. La première est de désactiver la fonction de mouvement chez les individus et de la remplacer par une redistribution complète et aléatoire de toute la population. De ce fait nous redistribuons tous les individus et obtenons un mélange parfait. Cette méthode est appelée "perfect\_mix". La seconde méthode est de permettre aux individus du système de se déplacer d'avantage. Nous pouvons en effet simuler un mélange parfait en déplaçant un grand nombre de fois les individus à chaque itération. Initialement le modèle était conçu pour permettre à chaque itération à chaque individu de se déplacer de une cellule. Mais un paramètre permet de déterminer le nombre de mouvements effectué par un individu à chaque itération. Pour simuler le modèle SI avec le plus de précision possible il fallait un grand nombre de déplacements. Ce nombre a été fixé à 5000 ce qui signifie que un individu a le potentiel de se déplacer de 5000 cellules en une seule itération.

ADD recherche de paramètre optimaux !

Le modèle doit être paramétré spécifiquement pour adopter les mêmes comportements qu'un modèle SI. Un exemple de paramètres est donné si-dessous :

```

1 TAILLE_SYSTEME = 200
2 NOMBRE_INDIVIDUS = 5000
3 ITERATIONS = 150
4 RERUN_LIMIT = 1000
5 FAIL_SEUIL = 10
6 GENOME_INIT_I = 0
7 GENOME_DIVERSITY_I = 0
8 GENOME_INIT_AP = 0
9 VITESSE_MUTATIONS_AP = 0
10 CHARGE_VIRALE = 1
11 PARAMETRE_FONCTION = 1
12 CELLULE_AP = 0
13 SURVIE_AP = 0

```

```

14 NOMBRE_MOUVEMENT = 5000
15 PERFECT_MIX = false
16 TEMPS_AVANT_IMMUNITE = 1
17 IMMUNITE_MECHANISME = false
18 RESISTANCE_MECHANISME = false

```

Lorsqu'on veut simuler le modèle SI il est nécessaire de désactiver la mécanique d'immunisation et de résistance naturelle.

```

1 IMMUNITE_MECHANISME = false
2 RESISTANCE_MECHANISME = false

```

Il faut ensuite paramétriser le modèle en fonction de la simulations que nous cherchons à effectuer. Les paramètres principaux pour simuler le modèle SI sont donnés ci-dessous. Tous les autres paramètres du modèles ne sont pas révélateurs dans ces configurations.

```

1 TAILLE_SYSTEME = 200
2 NOMBRE_INDIVIDUS = 5000
3 ITERATIONS = 150
4 CHARGE_VIRALE = 1
5 CELLULE_AP = 0
6 NOMBRE_MOUVEMENT = 5000
7 PERFECT_MIX = false
8 IMMUNITE_MECHANISME = false
9 RESISTANCE_MECHANISME = false

```

Pour toutes les mesures SI du document, les mécanismes de contaminations de cellules sont désactivés et le facteur de charge virale est par défaut à 1.

```

1 CHARGE_VIRALE = 1
2 CELLULE_AP = 0

```

Il reste ensuite à paramétriser la taille de la grille, le nombre d'individus, le nombre d'itérations et finalement le mode de déplacement.

### 7.3.2 Simulations de SIR

ADD recherche de paramètre optimaux !

Pour modéliser une situation reflétant un modèle SIR il faut un mécanisme d'immunités chez les individus. Notre modèle fournit un paramètre permettant d'activer ou de désactiver les processus d'immunités.

Un exemple de paramètres pour simuler un modèle SIR est donné ci-dessous :

```
1 TAILLE_SYSTEME = 200
2 NOMBRE_INDIVIDUS = 5000
3 ITERATIONS = 300
4 RERUN_LIMIT = 1000
5 FAIL_SEUIL = 10
6 GENOME_INIT_I = 65535
7 GENOME_DIVERSITY_I = 0
8 GENOME_INIT_AP = 0
9 VITESSE_MUTATIONS_AP = 0
10 CHARGE_VIRALE = 1
11 PARAMETRE_FONCTION = 4
12 CELLULE_AP = 0
13 SURVIE_AP = 0
14 NOMBRE_MOUVEMENT = 5000
15 PERFECT_MIX = false
16 TEMPS_AVANT_IMMUNITE = 1
17 IMMUNITE_MECANISME = true
18 RESISTANCE_MECANISME = false
```

La différence entre la configuration d'un modèle SI et celle d'un modèle SIR est l'ajout du mécanisme d'immunisation. Pour des questions de simplifications le paramètre de résistance naturelle est resté désactivé pour les simulations de SIR. Les autres paramètres sont tous activés par l'activation du paramètre d'immunisation :

```
1 IMMUNITE_MECANISME = true
```

Pour simuler un modèle SIR classique il est aussi nécessaire de désactiver le mécanisme de mutation des agents pathogènes ainsi que le mécanisme de diversité chez les individus. Il reste ensuite à paramétriser les génomes initiaux, le paramètre de fonction ainsi que les autres paramètres du modèle.

## 8 Modèle SI

### 8.1 Mesures et méthodologie SI

L'objectif de ces résultats est de valider le modèle implémenté. Toute une série de mesures ont été prise pour différentes densités de population et pour différentes tailles de système. Quatre niveaux de densité de population ont été mesuré ainsi que quatre taille de population. Les densité choisies pour analyses sont :  $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$  et les tailles de populations sont 5000, 20000, 50000, 100000 individus. Nous avons donc un total de 16 figures. Sur chacune de ces figure nous trouvons 3 courbes. En bleu nous avons la courbe du modèle mathématique SI, en orange nous avons la simulation avec le mode de mouvement défini à 1000 mouvements par individu par itération et finalement en vert nous avons la simulation avec le mélange parfait.

## 8.2 Résultats

Le modèle est très précis pour des systèmes très denses. Les similitudes entre les courbes du modèle SI et celle du mélange parfait montrent que le modèle suit les mêmes comportements que ceux du modèle mathématique SI. Le déroulement des simulations sur des systèmes denses sont presque déterministes car elles produisent toujours les mêmes résultats sans variations apparentes. Ceci est dû au fait que pour des systèmes à forte densité, le facteur chance de déclencher un événement est plus faible que sur les systèmes moins denses. Un chapitre ultérieur est dédié à ces variations. Sur les quatre figures nous voyons sans ambiguïté que les simulations aux 1000 mouvements (orange) déclenchent leur pandémies moins rapidement et moins brusquement. Deux phénomènes causent ces différences.

Le premier est dû au fait que le système est très dense. En densité  $\frac{1}{2}$ , la moitié des cellules du système sont occupées par un individu. Par conséquent ces derniers ont la peine à se déplacer car ils se gênent les uns les autres dans leurs déplacements. Sans oublier que un individu, dans ces simulations, essaie de se déplacer 1000 fois ce qui ne signifie pas qu'il se déplace effectivement 1000 fois. Nous observons donc du retard comparé au modèle SI pour les simulations aux 1000 mouvements.

Le deuxième phénomène qui explique le retard croissant est que pour toutes les simulations, le nombre de déplacements pour est constant. Par conséquent les grands systèmes se mélangent moins relativement à leur taille que les plus petits. C'est la raison pour laquelle les courbes oranges s'aplatissent de plus en plus sur des systèmes de plus en plus grands.

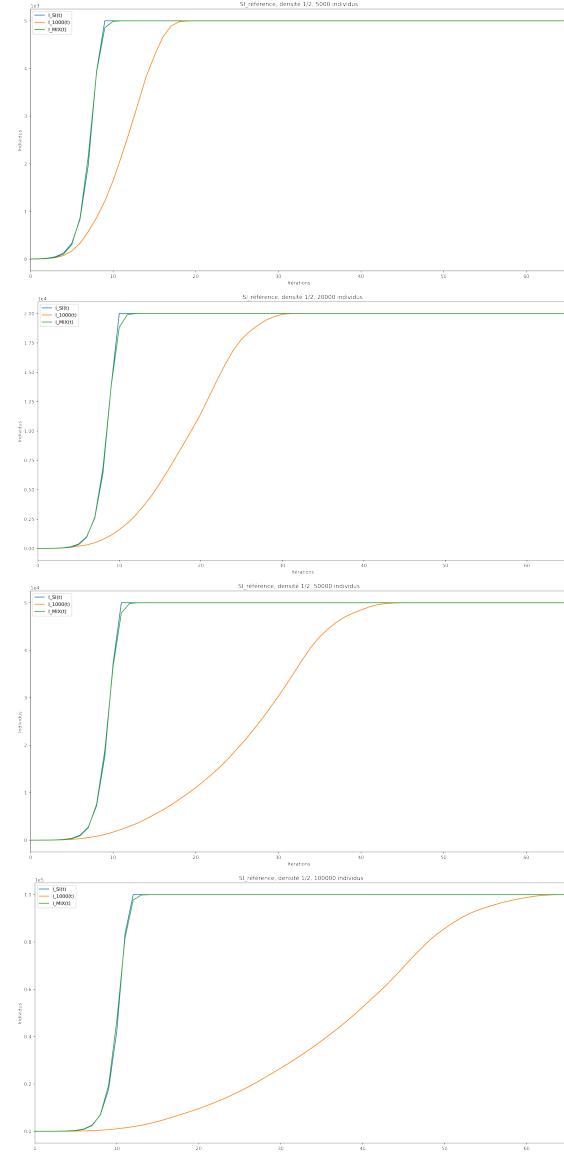


Figure 8.1: Simulations de SI, densité 1/2

Une densité de population de  $\frac{1}{4}$  permet d'avantages de déplacement pour les individus, par conséquent le mélange pour la méthode à 1000 mouvements est de meilleure qualité. Nous avons donc des courbes oranges qui subissent moins les blocages et elle approchent d'avantage le modèle SI.

Des résultats similaires sont observés pour une densité de  $\frac{1}{4}$  que pour une densité de  $\frac{1}{2}$ . Une différence notable est que les courbes oranges s'approchent d'avantage au mélange parfait que précédemment. Un système à cette densité reste très déterministe et par conséquent toutes les simulations au mélange parfait produisent des résultats précis qui collent au modèle SI.

De même pour cette densité, le taux de mélange de 1000 mouvements fixe impacte les grands systèmes en s'éloignant du modèle SI. Par contre la simulation sur 5000 individus particulièrement précise et ceci bien plus que les simulations en densité  $\frac{1}{2}$ . Ceci est dû au fait qu'une densité de  $\frac{1}{4}$  permet aux acteur de se déplacer plus aisément et donc d'obtenir un meilleur mélange. Par conséquent, sur un système de taille  $141 \times 141$  avec 5000 individus, le mélange des 1000 mouvements est presque aussi bon que le mélange parfait.

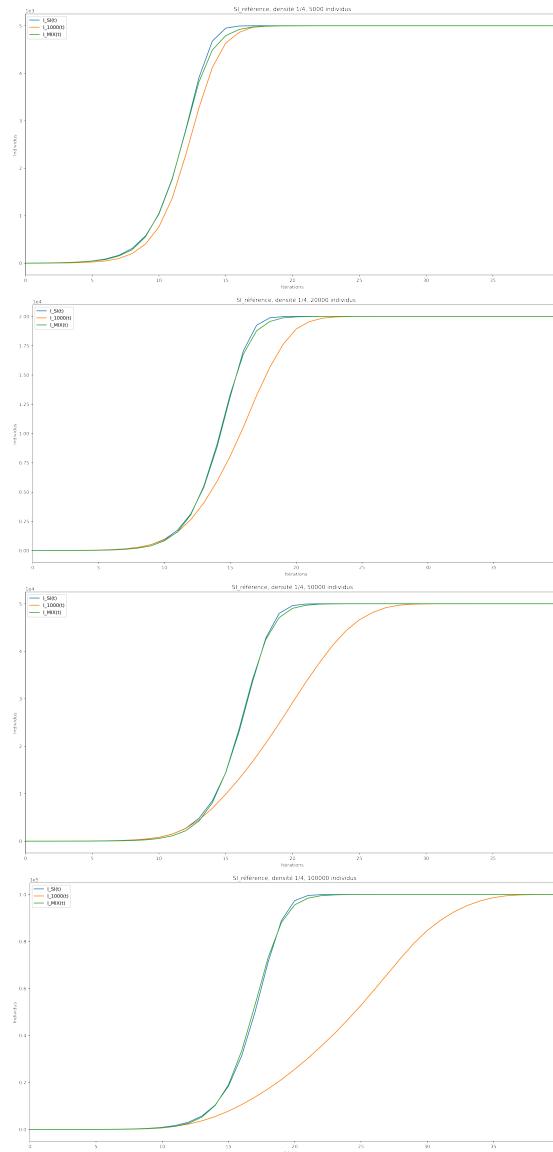


Figure 8.2: Simulations de SI, densité 1/4

Les mêmes comportements sont observés pour les simulations en densité  $\frac{1}{8}$ . La qualité du mélange décroît avec l'augmentations de la taille des systèmes et le mélange parfait est bon comparé au modèle mathématique SI. On remarque tout de même sur les trois simulations (5000, 20000, 50000) que les courbes de mélange parfait évoluent moins vite que le modèle SI. Ce comportement est logique car en diminuant la densité du système on réduit la probabilité d'entrer en contact avec d'autres individus et par conséquent de propager la contamination. Cette baisse de contact est en partie compensée par le fait que les individus ont (par la densité faible) la liberté de se déplacement plus aisément.

Un comportement intrigue sur la simulation à 20000 individus. La courbe des 1000 mouvements semble être décalée vers la droite sans pour autant qu'elle soit plus plate que le modèle SI. Ce comportement est du aux variations aléatoires sur les simulations. Lorsque la densité est faible le facteur chance croît ce qui a pour conséquence de déclencher des pandémies à des instants bien différents d'une simulation à l'autre. Ce phénomène est illustré et détaillé dans un chapitre ultérieur.

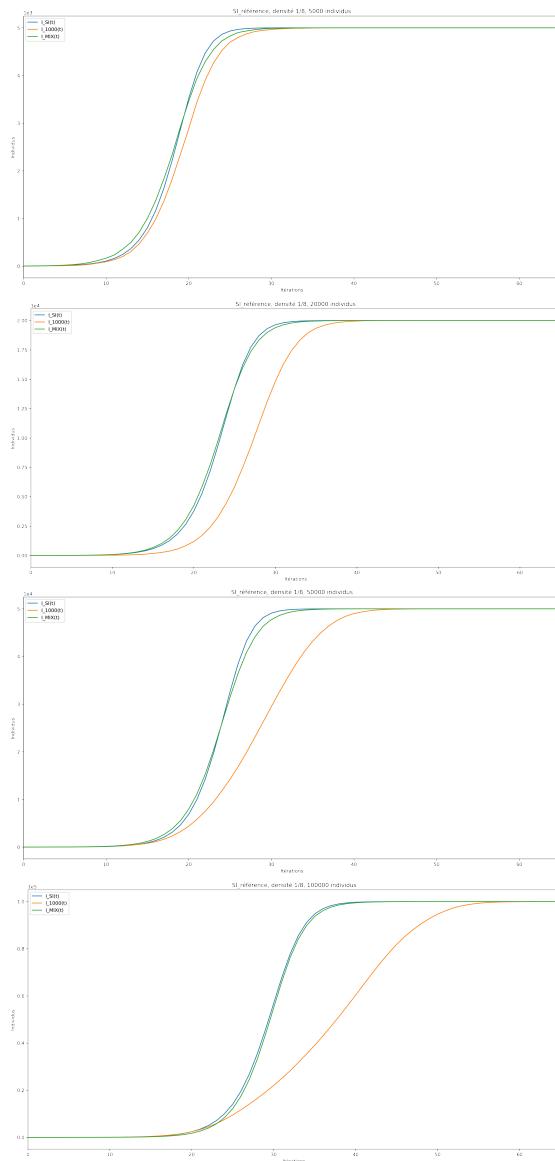


Figure 8.3: Simulations de SI, densité 1/8

Finalement les simulations en densité  $\frac{1}{16}$  donnent aussi de bons résultats. Les mêmes comportements sont observés sur ces simulations mais la simulation à 50000 individus montre un nouveau comportement pas encore observé sur les autres simulations. L'effet ici est assez faible mais on peut distinguer que la courbe du mélange parfait croît plus rapidement que le modèle mathématique SI. Nous avons donc une simulation qui évolue plus rapidement que le modèle SI. Cet effet n'est pas du au fait que le modèle implémenté est plus rapide mais uniquement au temps qu'il a fallu pour la pandémie se déclare. En effet le modèle SI n'a aucun temps de latence et commence donc à l'itération 0. Les mesures prises sur SI souffrent peu de de phénomène contrairement à certaines simulations par la suite.

Tout comme pour les simulations en densité  $\frac{1}{8}$ , nous pouvons observer de fortes variations sur les moments de déclenchement de pandémies. Sur la première simulations (5000 individus), la courbe des 1000 mouvement est retardée. Sur les trois autres c'est l'effet inverse. Ce comportement est illustré et mesuré plus loin.

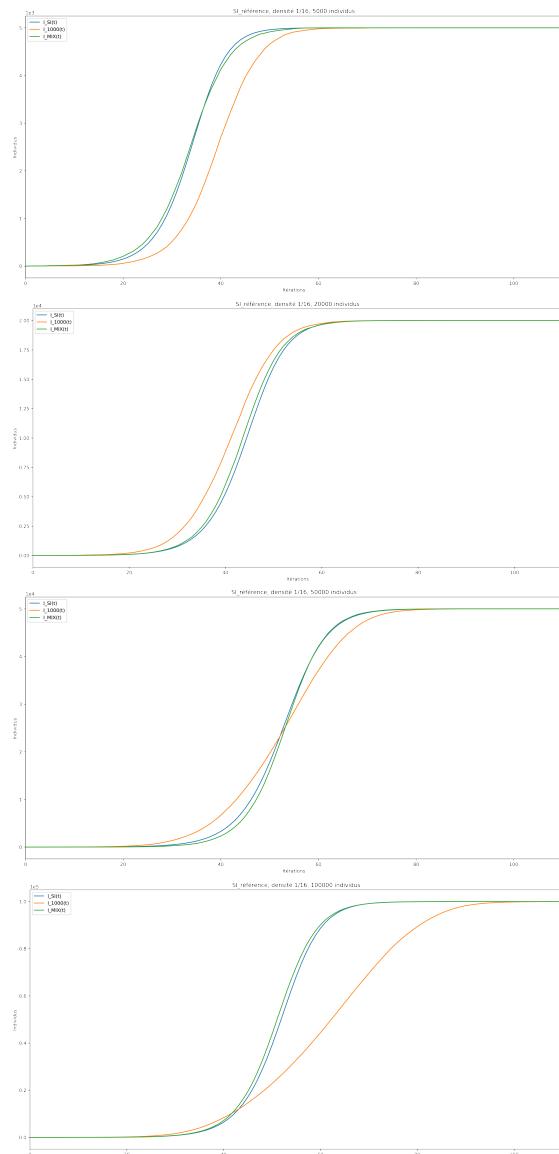


Figure 8.4: Simulations de SI, densité 1/8

## 8.3 Analyses

### 8.3.1 Mean Absolute Error

L'objectif du chapitre précédent est de prouver la validité du modèle implémenté en comparant les résultats avec des résultats bien connus du modèle SI. Le mean absolute error est un calcul d'erreur qui consiste à sommer toutes les différences entre les courbes et à calculer la moyenne de ces différences. Les résultats sont normalisés c'est-à-dire que nous divisons le MAE (mean absolute error) par le nombre d'individus. Ceci permet de comparer les résultats pour des simulations de taille différente.

Les mesures qui suivent comparent le modèle mathématique SI avec les simulations au mélange parfait.

Table 8.1: Mean Abosolute Error Normalized : modèle SI, mélange parfait

	5000	20000	50000	100000
1/2	6.2e-4	6.4e-4	6.4e-4	7.3e-4
1/4	9.0e-4	7.9e-4	7.7e-4	9.5e-4
1/8	3.2e-3	1.9e-3	2.8e-3	2.0e-3
1/16	4.2e-3	4.6e-3	4.2e-3	4.6e-3

Les résultats suivent les observations du chapitre précédent. Tout d'abord nous remarquons que l'erreur est constante pour une densité fixée, c'est-à-dire que l'erreur ne croit pas avec la taille du système. Pour ces analyses, la magnétude des erreurs ne sont pas importantes, ce qui nous intéresse est de comparer les valeurs les unes avec les autres.

Le deuxième résultat intéressant est que l'erreur augmente lorsque nous diminuons la densité du système. Ce comportement suit les observations des figures au chapitre précédent. L'erreur est due au temps de latence dont souffrent les systèmes moins denses. Une section ultérieure est dédié aux latences des grands systèmes.

### 8.3.2 Moyenne de voisinage

Les courbes des simulations ont montré des comportements peu intuitifs, surtout pour les systèmes peu denses. Cette section analyse le nombre de voisins moyens par individu pour ces simulations afin de déterminer si les résultats sont influencés par des mécaniques cachées derrière les mouvements des individus. Il s'agit ici de comparer la densité du voisinage de simulations au mélange parfait avec des simulations aux 1000 mouvements. L'hypothèse derrière cette recherche est de déterminer si le déplacement des individus implique d'avantages de contacts que la méthode du mélange parfait.

Nous nous intéressons aux simulations de faible densité car ce sont sur ces simulations que les comportements non intuitifs apparaissent. Les mesures sont prises sur des systèmes de densité  $\frac{1}{8}$  ainsi que  $\frac{1}{16}$  avec 5000 individus. Les simulations les plus petites ont été choisies pour des questions de temps de calcul mais aussi du fait que les micro mécaniques que nous essayons de détecter ne sont pas sensibles à la taille du système car c'est une analyse local à chaque individu.

Les mesures pour le système en densité  $\frac{1}{8}$  ont été effectuées sur 65 itérations puis moyennées. Les mesures pour le système de densité  $\frac{1}{16}$  ont été effectuées sur 110 itérations puis moyennées. Ces itérations ont été choisies par rapport aux bornes des figures de la section "Résultats".

Table 8.2: Voisinage moyen : modèle SI, mélange parfait, 1000 mouvements

	1000	perfect_mix
1/8	0.498	0.494
1/16	0.248	0.250

Les résultats montrent que le mode de mouvements n'influence pas la densité du voisinage, car conséquent les phénomènes observés dans les figures ne sont pas causés par la densité du voisinage. De plus les résultats nous réconforment dans le fonctionnement du modèle implémenté. En effet les valeurs sont attendues, pour une densité de  $\frac{1}{8}$  nous nous attendions à avoir en moyenne  $4 \times \frac{1}{8} = 0.5$  et pour une densité de  $\frac{1}{16}$  nous nous attendions à une moyenne de  $4 \times \frac{1}{16} = 0.25$ .

### 8.3.3 Variations aléatoires

Une autre hypothèse qui pourrait expliquer les comportements observés est que les simulations ne soient pas déterministe, c'est-à-dire qu'un facteur aléatoire détermine à quel instant la pandémie se déclare. Cela signifie que deux simulations aux paramètres identiques produisent deux résultats différents.

L'objectif de cette section est de mesurer cette variation sur des simulations à faible densité. Les densités étudiées sont  $\frac{1}{8}$  et  $\frac{1}{16}$  avec une population de 20000 individus. Pour chaque densité, les deux modes de déplacements sont étudiés et ceci sur un total de 20 simulations. Nous représentons donc 20 simulations avec des paramètres identiques et observons les différences.

Sur chacune des figure sont représentées les 20 simulations aux mêmes paramètres. Les mesures effectuées sur les simulations permettent de mettre ces courbes en relations les unes avec les autres. L'intérêt des mesures est de calculer les décalages sur l'axe des abscisses. Pour ce faire nous mesurons à quelle itération la moitié du système est contaminé. Les simulations ont toutes un nombre d'individus de 20000 par conséquent nous mesurons à quelle itérations les simulations ont atteint 10000 infectés. C'est à cet endroit que les déviations sont les plus grandes. En effectuant la même mesure en calculant l'itération des 20000 infectés, nous obtenons des variations moins importantes.

Avec les mesures nous pouvons calculer l'itération minimale ainsi que l'itération maximale des simulations pour atteindre le seuil des 10000 individus infectés. Il est ensuite possible de calculer la moyenne et la moyenne des déviations à cette moyenne.

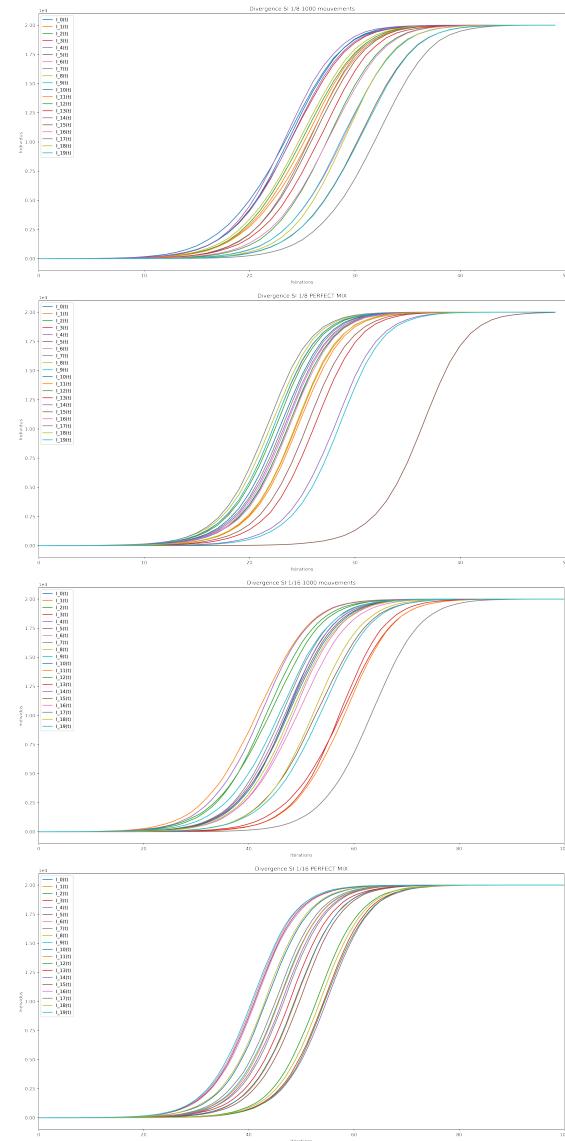


Figure 8.5: Divergences SI

Table 8.3: Voisinage : modèle SI

	1000 mouvements		Mélange parfait	
densité	1/8	1/16	1/8	1/16
min	24	42	22	41
max	32	64	37	55
mean	26.9	50.15	25.1	47.7
std	2.52	5.81	3.27	4.63

Plus les systèmes sont grands et plus les variations sont grandes comme on peut le

voir dans les résultats. Les simulations en densité  $\frac{1}{16}$  ont en moyenne une plus grande déviation ainsi qu'une plus grande différence entre le minimum et le maximum.

### 8.3.4 Positions des individus

Cette section a pour but de visualiser l'impacte de la densité des systèmes sur les déplacements des individus et donc sur la qualité du mélange. Les figures suivantes montrent les positions géographiques de tous les individus à une certaine itération. En vert sont affichés les individus sains et en rouge les individus contaminés.

L'image ci-dessous montre les positions des individus pour une simulation au mélange parfait à l'itération 20. Le mode de mélange parfait redistribue tous les individus dans l'espace à chaque itération, par conséquent les individus contaminés sont parfaitement répartis dans l'espace.

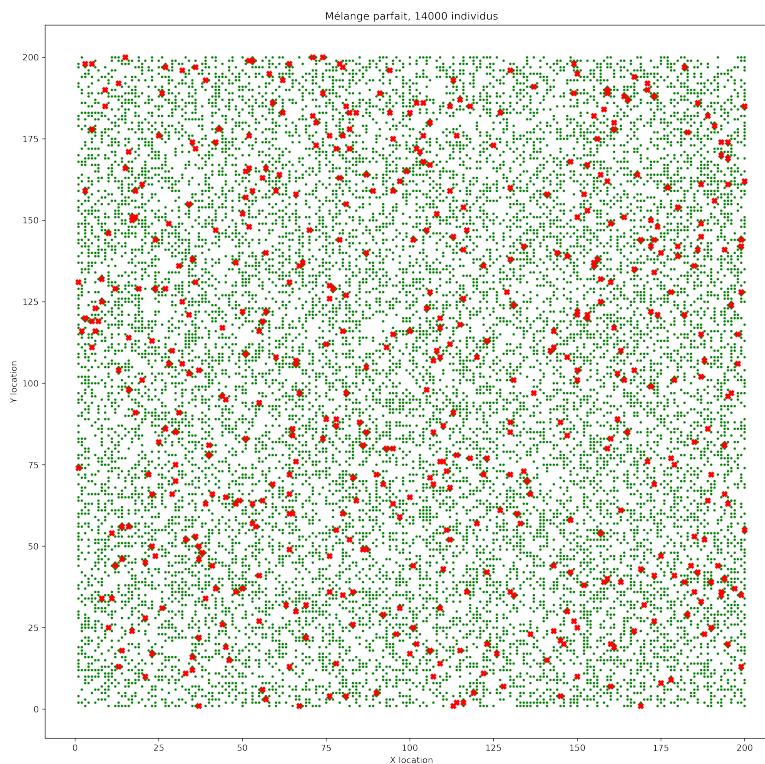


Figure 8.6: test

Sur les figures qui suivent, nous effectuons la même mesure mais cette fois-ci sur des simulations aux 1000 mouvements. L'idée est de pouvoir observer la qualité du mélange de population en fonction de la densité du système.

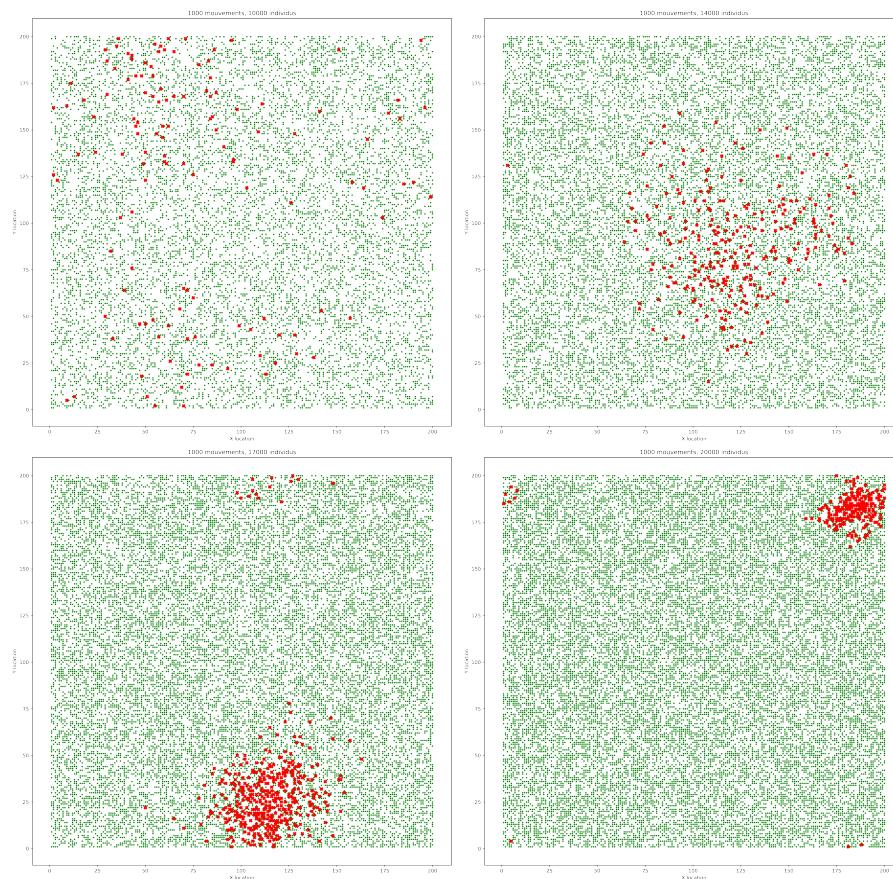


Figure 8.7: test

Modifications à effectuer...

# 9 Modèle SIR

## 9.1 Mesures et méthodologie SIR

Les mesures ainsi que les méthodologie sont les mêmes que pour le modèle SI, c'est-à-dire que les densité  $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$  et les tailles de populations 5000, 20000, 50000, 100000 sont étudiées.

## 9.2 Résultats

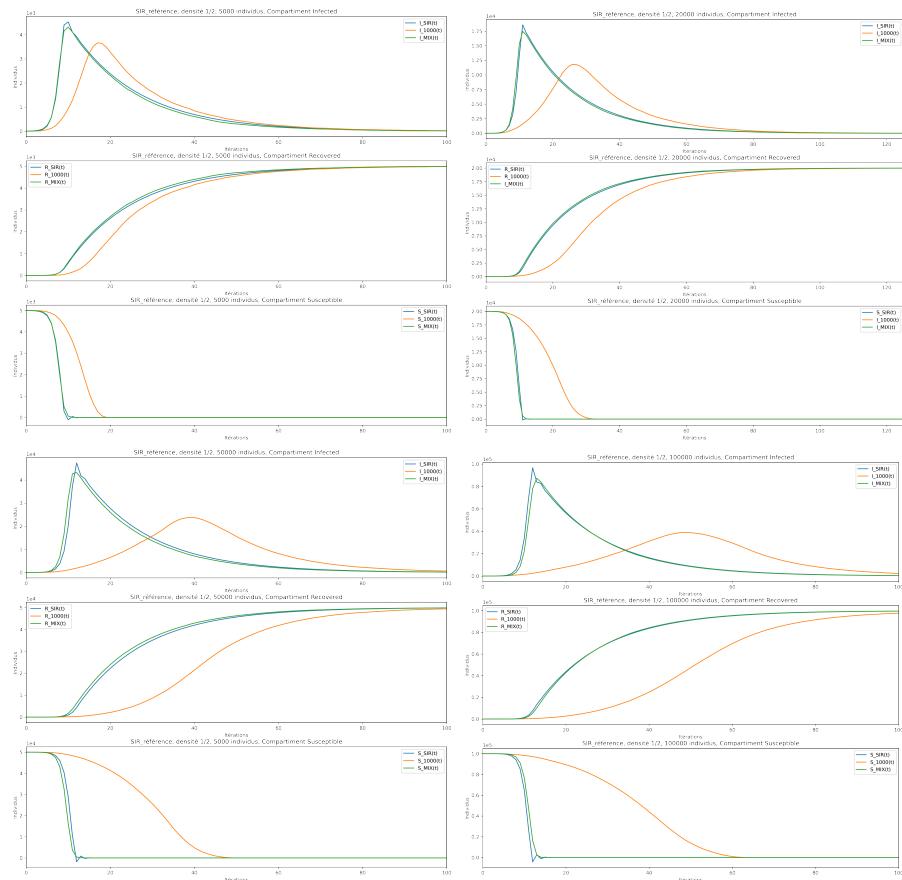


Figure 9.1: test

Pour les systèmes de forte densité, les simulations aux 1000 mouvements peinent à se mélanger. Par conséquent, les événements sont plus tardifs. Nous pouvons observer les mêmes comportements que pour les simulations SI.

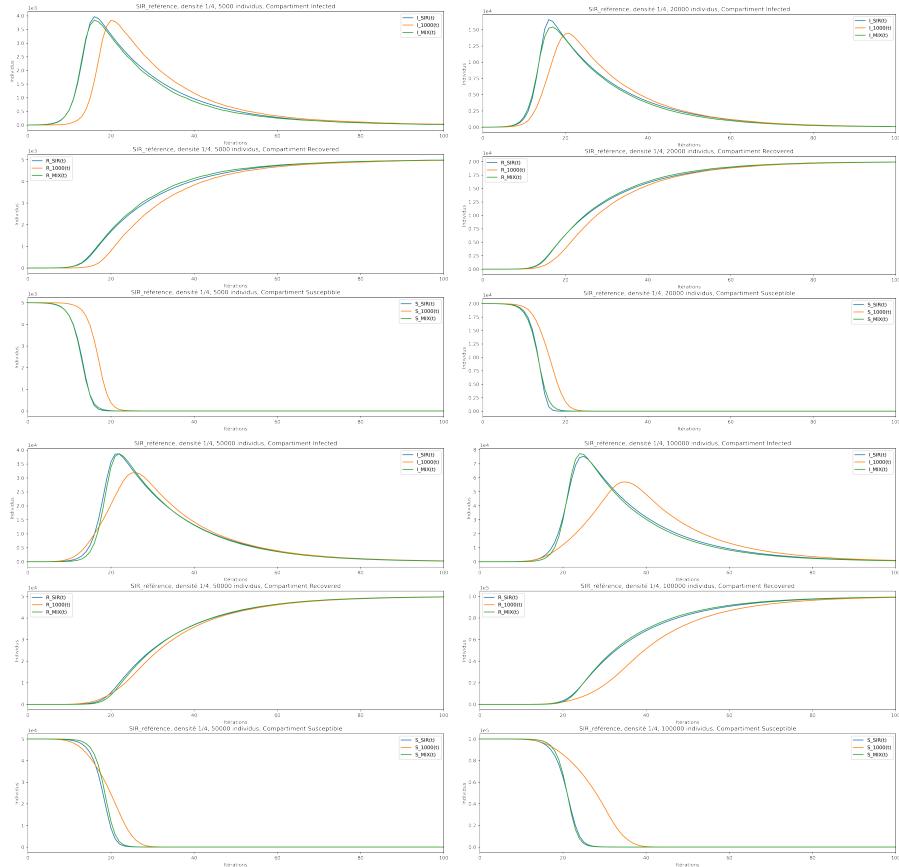


Figure 9.2: test

En densité  $\frac{1}{4}$  les simulations des 1000 mouvements tendent d'avantage vers le mélange parfait. Les courbes sont toujours plus progressives et donc les événements plus lents.

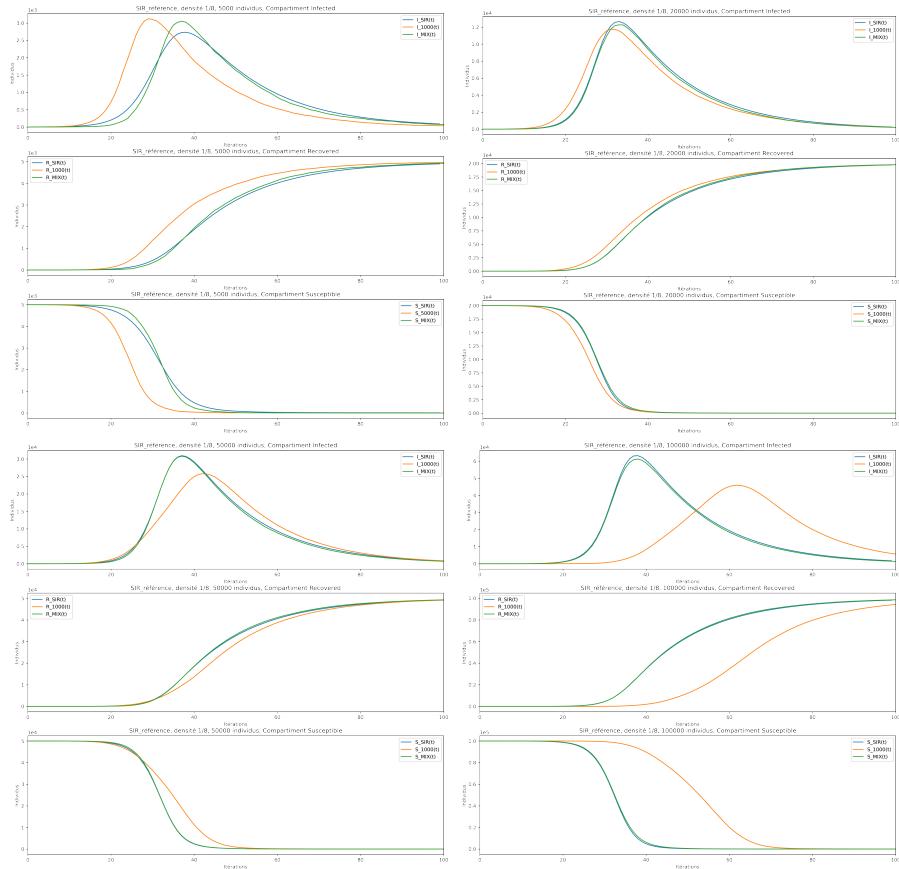


Figure 9.3: test

En plus de retrouver les mêmes comportements qu'observés précédemment, il y a un nouveau phénomène que nous pouvons observer ici. Ce phénomène est particulièrement présent dans les simulations à 5000, 20000 et 100000 individus. Nous remarquons que la croissance des courbes pour les simulations de mélange parfait sont plus rapides que le modèle mathématique SIR. Ce comportement est dû au fait que nos simulations peuvent nécessiter d'un temps de latence avant l'émergence d'un événement. Alors que au contraire le modèle SIR n'est pas victime de cette latence.

Par conséquent, pour joindre le modèle SIR aux simulations il faut prendre en compte le temps de latence et l'ajouter au modèle SI artificiellement. Ce temps de latence est particulièrement présent sur les systèmes peu denses car dans ces configurations les pandémies peuvent prendre du temps à apparaître. Ce phénomène est étudié dans un autre chapitre.

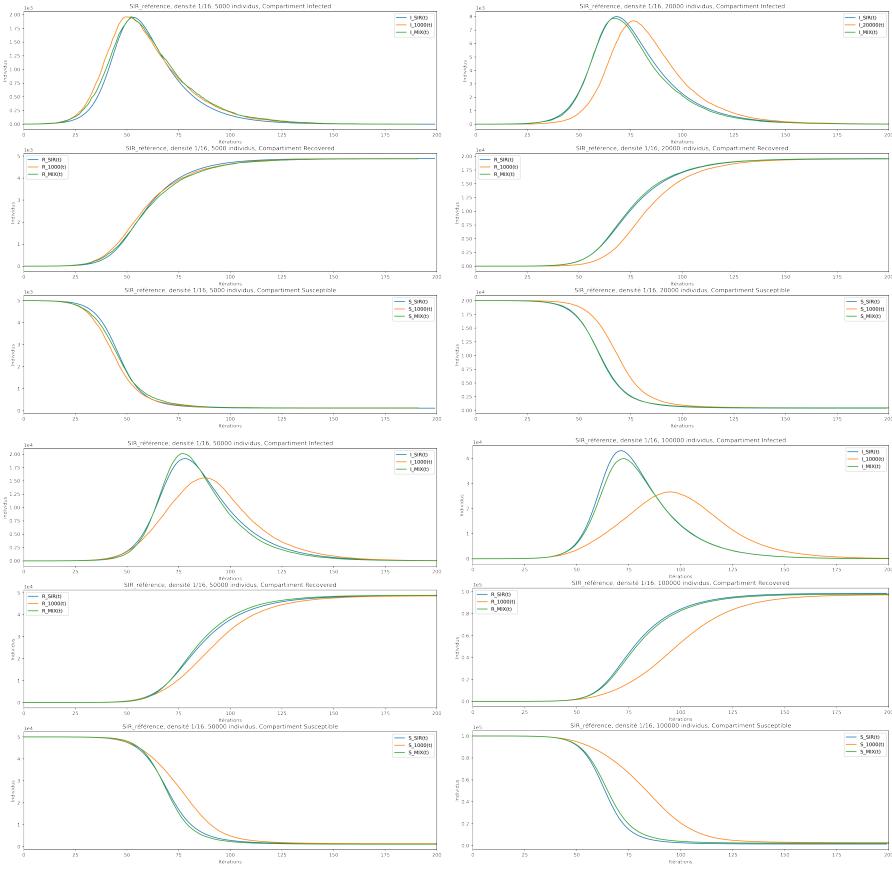


Figure 9.4: test

Tout comme pour le modèle SI, les simulations aux 1000 mouvements sont ralenties par les systèmes de grande taille car le mouvement des individus est réduit par rapport à la taille du système. C'est la raison pour laquelle les courbes au 1000 mouvements sont retardées.

Malgré des observations déjà analysées, nous obsevons des oscillations sur la simulation à 5000 individus. Ce phénomène n'était pas présent sur les simulations SI et ceci à cause de l'absence du compartiment *Recovered*. Deux raisons explique ce comportement, le premier est que le mécanisme d'immunisation transfert des individus du compartiment *I* au *R* et donc abaisse la courbe. Le second est dû à la densité du système qui est très faible, par conséquent il y a peu de contacts qui s'effectuent ce qui crée des oscillations. Ces oscillations sont aussi présentes sur des systèmes plus denses mais pas perceptibles car moyennée par un grand nombre d'événements.

## 9.3 Analyses

### 9.3.1 Mean Absolute Error

Le mean absolute error permet de quantifier les différences entre les simulations et le modèle mathématique SIR. Les calculs suivants ont été effectués sur les simulations

au mélange parfait. Contrairement au modèle SI, le modèle SIR contient trois courbes différentes par conséquent chacune de ces courbes a son propre MAE. Le tableau ci-dessous donne la moyenne de ces trois résultats et ceci pour toutes les simulations SIR.

Table 9.1: Mean Absolute Error Normalized : modèle SIR, mélange parfait

	5000	20000	50000	100000
1/2	4.3e-3	2.1e-3	2.7e-3	2.5e-3
1/4	3.8e-3	3.0e-3	2.1e-3	2.6e-3
1/8	9.6e-3	2.8e-3	1.9e-3	2.3e-3
1/16	8.5e-3	3.2e-3	6.2e-3	7.2e-3

Les résultats pour le modèle SIR sont généralement moins précis que pour le modèle SI. Certains résultats sont intéressants par exemple les courbes de densité  $\frac{1}{8}$  avec 5000 individus. L'erreur sur cette courbe est élevée car la simulation souffre de latence, par conséquent le déclenchement de l'événement se fait tardivement. C'est également la raison pour laquelle la simulation au 1000 mouvement évolue plus rapidement que la simulation au mélange parfait.

En densité  $\frac{1}{8}$  d'avantage d'erreurs apparaissent. Premièrement, une plus grande erreur de la simulation à 5000 individus et due aux oscillations causées par le peu d'événements du système, donc les systèmes peu denses ont une plus grande erreur. Deuxièmement les simulations à faible densité prennent du temps déclencher un événement, ce qui engendre un temps de latence. Ce temps de latence est innexistant dans le modèle SIR, par conséquent notre modèle ne se comporte plus exactement comme un modèle mathématique SIR.

### 9.3.2 Variations aléatoires

Tout comme pour les simulations SI, nous calculons les variations d'une simulation à une autre et ceci pour des paramètres identiques. Les mesures sont faites sur un ensemble de simulations de densité  $\frac{1}{8}$  ainsi que  $\frac{1}{16}$  avec une population de 20000 individus.

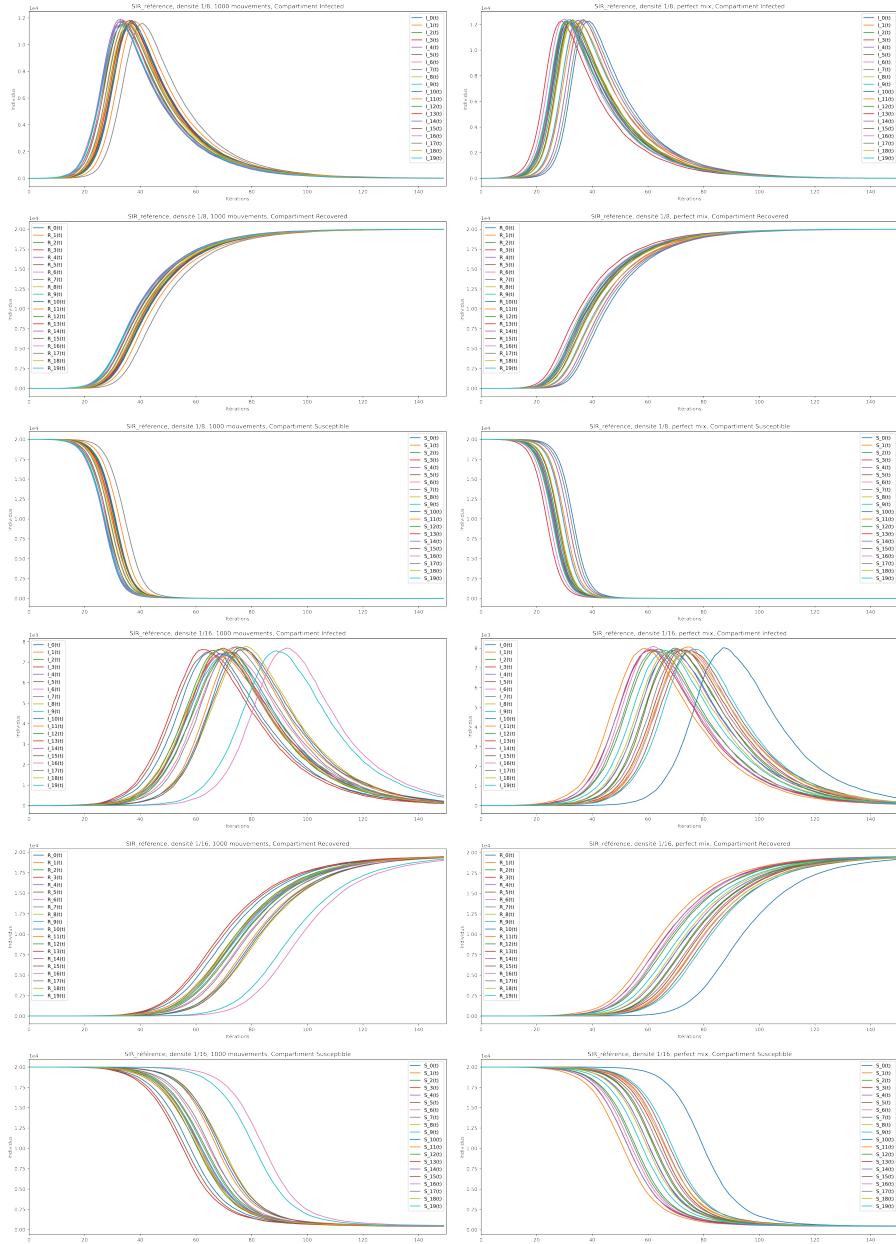


Figure 9.5: test

Idem aux simulations SI, nous pouvons observer de grandes variations d'une simulation à une autre. A nouveau, ce sont les systèmes peu denses qui sont victimes de plus grandes variations. La mesure de la variation n'est pas la même que pour le modèle SI. Ici nous mesurons l'itération du pic de contaminés simultanés, c'est-à-dire l'itération de la valeur maximale des courbes du compartiment *Infected*.

Les analyses permettent de calculer la moyenne ainsi que la moyenne des écarts.

Table 9.2: Voisinage : modèle SIR

	1000 mouvements	Mélange parfait	
densité	1/8	1/16	1/8
min	33	62	29
max	41	93	38
mean	35.35	72.85	33.2
std	2.10	7.28	2.58
			69.35
			6.77

Le tableau confirme plusieurs attentes du modèle. Premièrement, les simulations au mélange parfait sont plus rapides que les simulations aux 1000 mouvements et ceci pour les deux densités étudiées. Deuxièmement, la moyenne des déviations est plus grande sur des systèmes moins denses, ce qui confirme nos hypothèses.

### 9.3.3 Latence des simulations

Certaines simulations du modèle implémenté divergent du modèle SIR. Une des hypothèses expliquant ce phénomène est que les simulations peuvent nécessiter un certain temps de latence avant que la pandémie apparaisse, une période durant laquelle la situation n'évolue pas. Cette latence est théoriquement possible dans le modèle mais contre elle est inexiste dans le modèle mathématique SIR.

L'objectif de cette section est de déterminer l'impact de la latence sur le "mean absolute error" entre les simulations et le modèle mathématique SIR. Pour ce faire nous utilisons une simulation qui subit de la latence. De toutes les simulations du document, peu d'entre elles montrent des signes de latence. Ceci est compréhensible car la latence croît avec la diminution de la densité du système mais décroît en même temps. Un système à faible densité a une certaine latence à cause du manque d'événements mais en même temps le manque d'événements réduit la latence car la progression de la pandémie en sera ralentie et donc SIR s'adaptera mieux à la simulation.

Par conséquent, la latence est généralement peu présente dans les simulations car le phénomène est auto-compensé. La simulation ci-dessous est l'exemple le plus explicite du comportement. La densité de cette simulation est de  $\frac{1}{8}$  ce qui semble être une densité qui révèle de la latence. Tous les systèmes plus ou moins denses du document n'ont pas donné autant de latence.

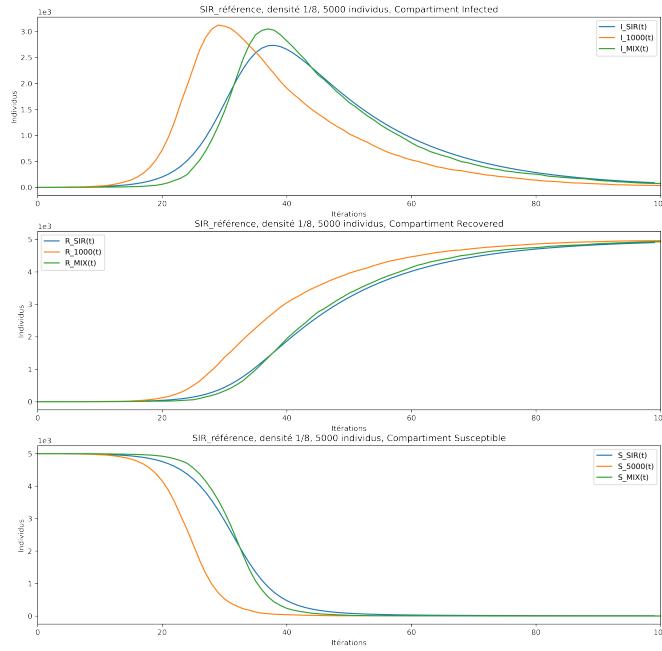


Figure 9.6: test

Pour confirmer la théorie de la latence, nous ajoutons une latence artificielle au modèle SIR et regardons l'impacte sur le "mean absolute error". En effet, si la simulation a quelques itérations de latence, nous devrions pouvoir ajouter de la latence au modèle SIR et observer des améliorations sur le MAE.

La figure ci-dessus montre la valeur du MAE en ajoutant de la latence au modèle SIR. Sur les abscisses nous avons le nombre d'itérations de latence ajoutée à SIR et sur l'axe des ordonnées nous avons le calcul du MAE entre le modèle mathématique et la simulation.

La valeur minimale de MAE est atteinte avec une latence artificielle sur SIR de 7 itérations. Ce qui veut dire que notre simulation a subit une latence de environ 7 itéraisons.

### 9.3.4 Mouvements variable

Jusqu'à présent, toutes les simulations ont utilisé une des deux méthodes de déplacement : mélange parfait ou 1000 mouvements. Nous sommes partis de l'hypothèse que plus le nombre de mouvements est grand, plus le modèle tend vers la méthode du mélange parfait.

Ce paragraphe est destiné à visualiser l'impacte du nombre de mouvements sur le déroulement d'une simulation. La même simulation est exécutée 1 fois avec le mode de mélange parfait et 9 fois avec un nombre de mouvements variable : 1, 10, 50 ,100, 200, 400, 600, 800, 1000 mouvements.

Les résultats suivent notre intuition. Plus le mélange est efficace, plus la pandémie est rapide. Par conséquent, le nombre de mouvements a un impact direct sur la propagation de pandémies et ceci significativement.

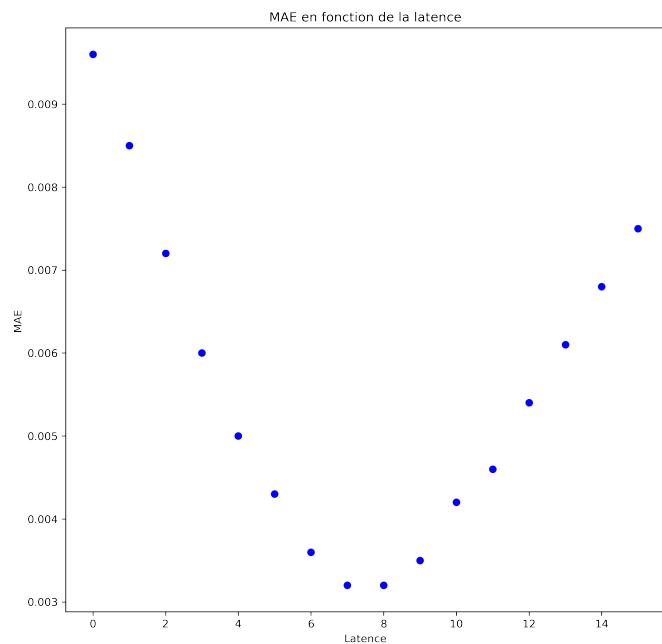


Figure 9.7: test

### 9.3.5 Comparaison 1000 mouvements

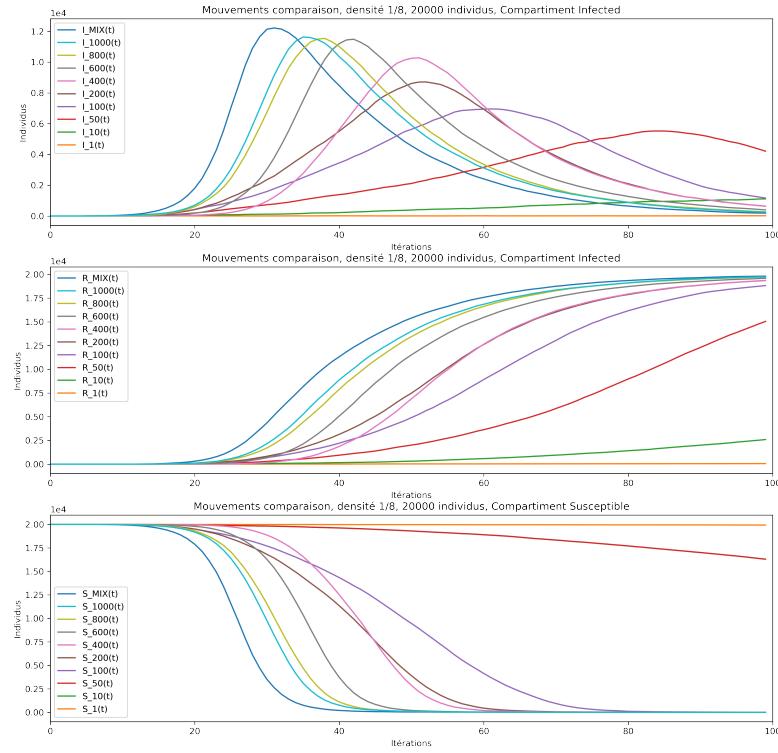


Figure 9.8: test

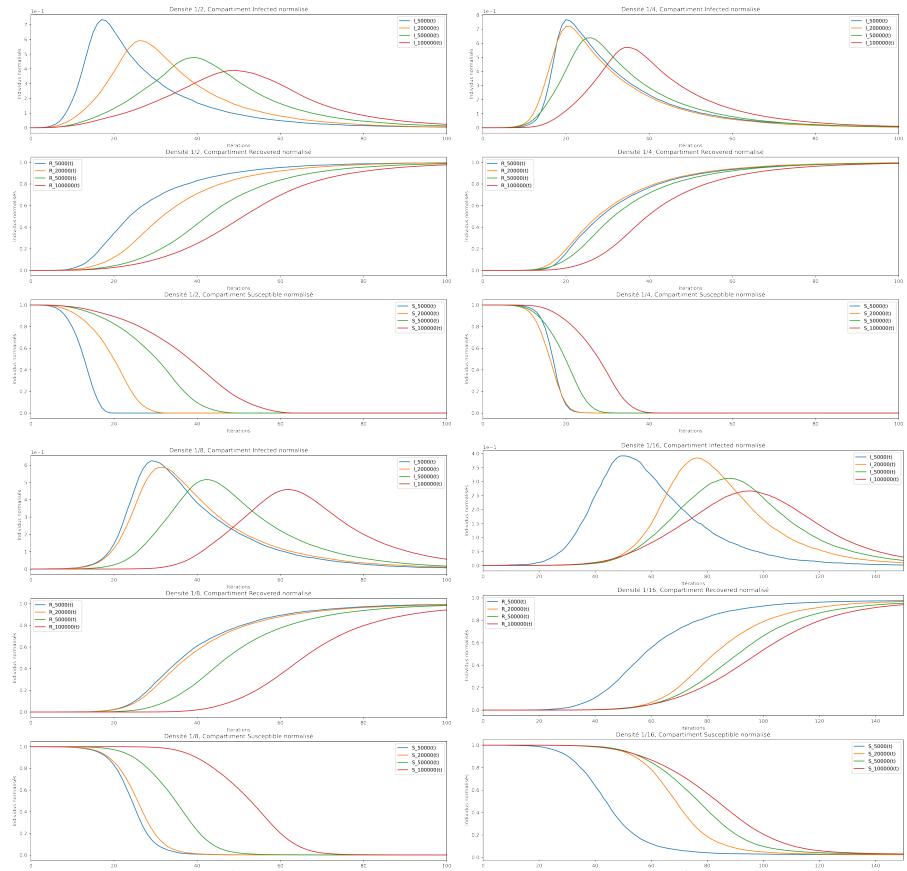


Figure 9.9: test