

DIVERSITÉ

TRAVAIL DE BACHELOR  
SCIENCES INFORMATIQUES



**UNIVERSITÉ  
DE GENÈVE**

**FACULTÉ DES SCIENCES**  
Département d'informatique

Université de Genève  
Faculté des sciences  
Département d'informatique

Maxime Pasquier

Printemps 2021

# DIVERSITÉ

## TRAVAIL DE BACHELOR SCIENCES INFORMATIQUES

Université de Genève  
Département d'informatique

Approuvé par

Superviseurs, Professeur Bastien Chopard & Franck Raynaud

Examineurs, Professeur Bastien Chopard & Franck Raynaud

Printemps 2021

# Table des matières

Liste des tableaux	IV
Table des figures	V
<b>1 Résumé</b>	<b>1</b>
<b>2 Introduction</b>	<b>2</b>
<b>3 Approche du problème, méthodes, outils utilisés</b>	<b>3</b>
3.1 Approche du problème . . . . .	3
3.2 Méthodes . . . . .	3
3.3 Outils . . . . .	3
<b>4 Modèle</b>	<b>4</b>
4.1 Acteurs . . . . .	4
4.1.1 Individus . . . . .	4
4.1.2 Agents pathogènes . . . . .	5
4.2 Espace physique . . . . .	5
4.2.1 Cellule . . . . .	5
4.2.2 Grille régulière . . . . .	7
4.2.3 Voisinage . . . . .	8
4.3 Simulation . . . . .	8
4.3.1 Itérations . . . . .	9
4.3.2 Mouvements des acteurs . . . . .	10
4.3.3 Mise à jour des acteurs . . . . .	11
4.4 Interactions . . . . .	12
4.4.1 Interactions sur la même cellule . . . . .	12
4.4.2 Interactions sur cellules voisines . . . . .	12
4.4.3 Contamination de cellule . . . . .	13
4.5 Diversité génétique . . . . .	13
4.5.1 Distance de Hamming . . . . .	15
4.6 Immunisation et résistance naturelle . . . . .	16
4.7 Mutations . . . . .	17
<b>5 Implémentation</b>	<b>18</b>
5.1 Main . . . . .	18
5.2 Initialisation . . . . .	19

5.3	Exécution de la simulation . . . . .	20
5.4	Fermeture . . . . .	24
<b>6</b>	<b>Objectifs</b>	<b>25</b>
<b>7</b>	<b>Modèles Compartimentaux</b>	<b>27</b>
7.1	Modèle SI . . . . .	28
7.2	Modèle SIR . . . . .	29
7.3	Simulations de références . . . . .	31
7.3.1	Simulations de SI . . . . .	31
7.3.2	Simulations de SIR . . . . .	33
<b>8</b>	<b>Modèle SI</b>	<b>34</b>
8.1	Mesures et méthodologie SI . . . . .	34
8.2	Résultats . . . . .	35
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 . . . . .	49
9.3.1	Mean Absolute Error . . . . .	49
9.3.2	Variations aléatoires . . . . .	50
9.3.3	Latence des simulations . . . . .	51
9.3.4	Mouvements variable . . . . .	54
9.3.5	Comparaison 1000 mouvements . . . . .	55
<b>10</b>	<b>Diversité</b>	<b>57</b>
10.1	Mesures et méthodologie . . . . .	57
10.2	Résultats . . . . .	58
10.2.1	Mélange parfait . . . . .	58
10.2.2	Mouvements variable . . . . .	60
10.2.3	Conclusion des résultats . . . . .	63
<b>11</b>	<b>Conclusion</b>	<b>66</b>

## Liste des tableaux

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

## Table des figures

FIGURE 4.1: Différents états de la cellule . . . . .	6
FIGURE 4.2: Grille régulière . . . . .	7
FIGURE 4.3: Voisinage d'une cellule . . . . .	8
FIGURE 4.4: Configuration initiale . . . . .	9
FIGURE 4.5: Mouvements des individus . . . . .	10
FIGURE 4.6: Calcul de la distance de Hamming . . . . .	15
FIGURE 5.1: Matrices de pointeurs . . . . .	20
FIGURE 5.2: Fonction de conversion en probabilité (facteur 1) . . . . .	22
FIGURE 5.3: Fonction de conversion en probabilité (facteur 2) . . . . .	22
FIGURE 5.4: Fonction de conversion en probabilité (facteur 4) . . . . .	22
FIGURE 5.5: Fonction de conversion en probabilité (facteur 8) . . . . .	22
FIGURE 7.1: Modèle mathématique SI . . . . .	29
FIGURE 7.2: Modèle mathématique SIR . . . . .	30
FIGURE 8.1: Simulations de SI, densité 1/2 . . . . .	35
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 1/16 . . . . .	39
FIGURE 8.5: Variations aléatoires : SI . . . . .	42
FIGURE 8.6: Positions des individus : mélange parfait . . . . .	43
FIGURE 8.7: Positions des individus : 1000 mouvements, densité variable . . . . .	44
FIGURE 9.1: Simulations de SI, densité 1/2 . . . . .	45
FIGURE 9.2: Simulations de SI, densité 1/4 . . . . .	46
FIGURE 9.3: Simulations de SI, densité 1/8 . . . . .	47
FIGURE 9.4: Simulations de SI, densité 1/16 . . . . .	48
FIGURE 9.5: Variations SIR . . . . .	50
FIGURE 9.6: Simulation SIR avec latence . . . . .	53
FIGURE 9.7: Mesures de latences SIR . . . . .	53
FIGURE 9.8: Mouvements variable : SIR . . . . .	54
FIGURE 9.9: Comparaison 1000 mouvements . . . . .	55
FIGURE 10.1: Comparaison 50 mouvements . . . . .	59
FIGURE 10.2: 50 mouvements . . . . .	60
FIGURE 10.3: Comparaison 10 mouvements . . . . .	62

FIGURE 10.4: Comparaison 1 mouvements . . . . .	63
FIGURE 10.5: vague propagation . . . . .	64

# 1 Résumé

Une pandémie est une épidémie présente sur une grande zone et affectant un grand nombre de personnes. Ces épidémies sont le résultats de plusieurs causes. 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 une multitude de raisons.

Le travail se concentre sur l'élaboration d'un modèle numérique de propagation de pandémie. L'objectif est de construire ce modèle en incorporant la notion de diversité écologique. La diversité d'une espèce est un facteur clef permettant la survie et l'évolution d'une espèce par conséquent nous voulions quantifier l'impacte de la diversité sur la propagation de pandémies. Le premier objectif est de valider le modèle en le confrontant à des modèles compartimentaux déjà prouvés. Il s'agit de soumettre le modèle à différents examens, de comprendre son fonctionnement et de pouvoir l'expliquer. Le modèle contient une multitude de paramètres, le travail n'en couvre que quelques uns.

Les simulations se basent sur le principe d'automate cellulaire. Le système se présente sous forme d'une grille bidimensionnelle tel un échiquier de taille voulu. La population navigue sur cet échiquier et les individus interagissent les uns avec les autres. Le but est de simuler les mouvements et les contacts au sein d'une population qui permettraient la propagation d'agents pathogènes. Chaque simulation peut être paramétrée et les informations sont enregistrées et peuvent être analysées par la suite.

L'objectif est d'étudier les résultats de simulations en faisant varier les paramètres du modèle dont la diversité écologique. Les résultats permettraient de quantifier l'impacte qu'à la diversité écologique sur une pandémie.



## 2 Introduction

L'élément principal de la recherche scientifique est de valider un modèle numérique et d'étudier la propagation de pandémies en fonction de la diversité. 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 cherche à simuler des mécanismes comme les génomes, les mutations et les immunités. D'autres facteurs sont paramétrables, comme la taille du système, le nombre d'individus et bien d'autres. En un premier temps le travail se concentre sur la validation du modèle puis cherche à démontrer l'impacte qu'à les mécanismes de diversité sur l'apparitions d'événements de grande ampleur.

La simulation ne comporte que deux types d'acteurs : des individus d'une population donnée et des agent pathogènes. La dimension spatiale est représentée par une grille bi-dimensionnelle permettant aux individus de se déplacer librement. Ce modèle simplifié permet de modéliser les interactions entre les acteurs du système.

Initialement, un seul individu du système est contaminé, il s'agit du patient zéro. La simulation fait évoluer le système pendant un certain temps et nous cherchons 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 Approche du problème

Il existe un grand nombre de modèles numériques simulant la propagation de pandémies. L'approche initiale était plutôt orientée vers la notion de diversité génétique. Nous voulions un moyen d'inclure et d'interpréter la diversité dans le modèle. Hormis la diversité, le modèle possède une représentation spatiale en deux dimensions et des objets pouvant interagir.

### 3.2 Méthodes

La mécanique de diversité se base sur l'attribution d'un code à chaque acteur du système. Ce code influence les comportements du système ainsi que l'émergence de pandémies. Le modèle numérique intègre un mécanisme simple pour interpréter ces codes.

### 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 simulations ainsi que les individus et 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 dépositaire GitHub et est accessible librement. Ceci fournit de la transparence ainsi qu'un moyen de distribuer le projet. De plus toutes les versions antérieures de la recherche y figurent.

## 4 Modèle

Le modèle cherche à simuler la propagation de pandémies dans une population. Le système inclut un espace physique ainsi que des acteurs se déplaçant sur cette surface. 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 en affecte 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 individu.

Les organismes évoluent dans un 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 :

1. Les agents pathogènes sont les agents infectieux responsables de l'épidémie.
2. Les individus, 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 de parasites. La caractéristique de ce groupe est qu'il a le pouvoir de rendre malade un individu 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.

#### 4.1.1 Individus

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

1. La première forme décrit un individu qui n'est pas contaminé par un agent pathogène et donc n'est pas touché par la maladie.
2. La seconde état survient lorsque l'individu est contaminé par un agent pathogène. Le modèle ne fait pas la distinction entre malade et porteur du virus, la notion de malade n'est pas prise en compte. Un individu infecté par un agent pathogène est contagieux, ce qui signifie qu'il a la pouvoir de transmettre l'agent pathogène à d'autres individus.

#### 4.1.2 Agents pathogènes

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

1. 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 individu devient contaminé par un agent pathogène, ce dernier est absorbé par l'individu. 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 sain devient un individu contaminé et non pas un individu sain associé à un agent pathogène.
2. Le second cas indique qu'une surface physique est contaminée par un agent pathogène. Un individu 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'individu 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 est constituée cellules dans lesquelles on peut y placer des acteurs. 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.

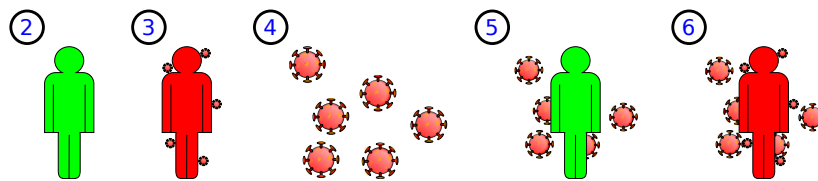
#### 4.2.1 Cellule

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 s'y trouvent. 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 individu sain se trouve sur la cellule.
3. Un individu infecté se trouve sur la cellule.
4. Un agent pathogène se trouve sur la cellule, infectant l'espace, sans individu.
5. Un individu sain se trouve sur une cellule contaminée par un agent pathogène.
6. Un individu 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.



**FIGURE 4.1** – Une cellule possède différents états : (2) individu sain, (3) individu contaminé, (4) agent pathogène contaminant une cellule, (5) individu sain sur un espace contaminé, (6) individu contaminé sur un espace contaminé. Sans oublier l'état où la cellule est vide.

### 4.2.2 Grille régulière

La grille régulière est l'espace sur lequel les acteurs évoluent. Chaque acteur est représenté au niveau spatial par ses coordonnées sur la grille et cette dernière leur permet de se déplacer et d'interagir. Les acteurs ne peuvent évoluer spatialement qu'en respectant la géométrie de la grille. Les bords du système sont périodiques, c'est-à-dire 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. Les bords sont donc connectés afin d'éviter les effets de bords.

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

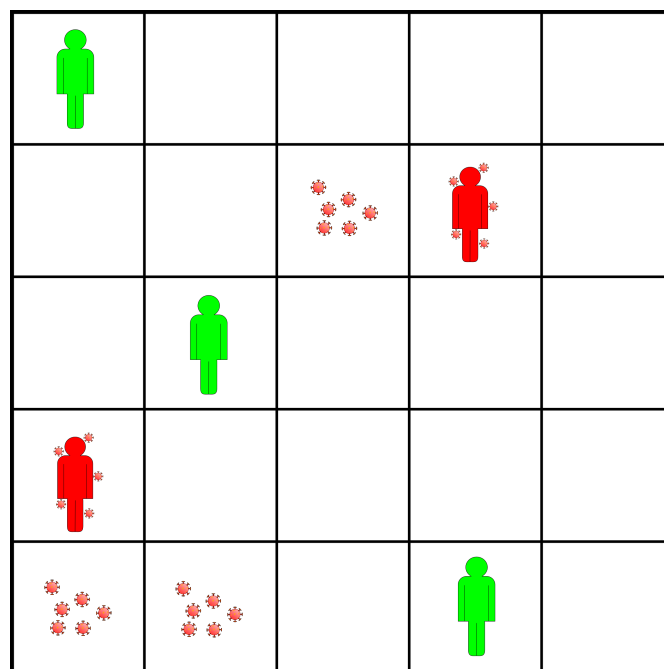


FIGURE 4.2 – Exemple d'une simulation en cours avec une grille de  $5 \times 5$  et 5 individus.

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 :

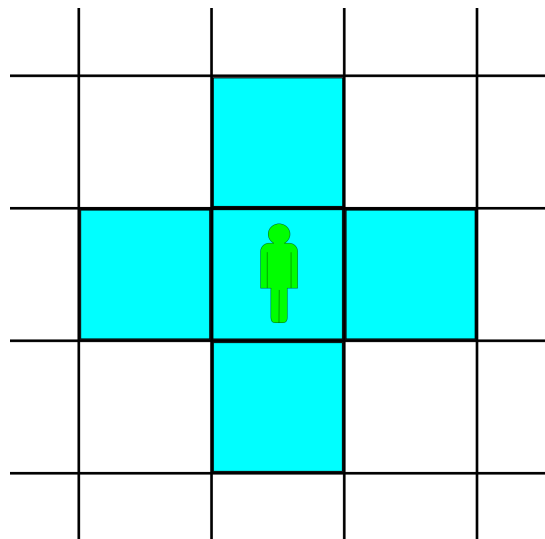


FIGURE 4.3 – Le voisinage de la cellule centrale est les 4 cellules en contact direct ainsi que la cellule centrale.

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 des 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 l'état d'un acteur ne sont influencés que par son l'état actuel ainsi que par 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 d'individus 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 individu, le patient zéro. Il s'agit ensuite d'observer la pro-

pagation ou non du pathogène initial. Un exemple de configuration initiale pourrait être :

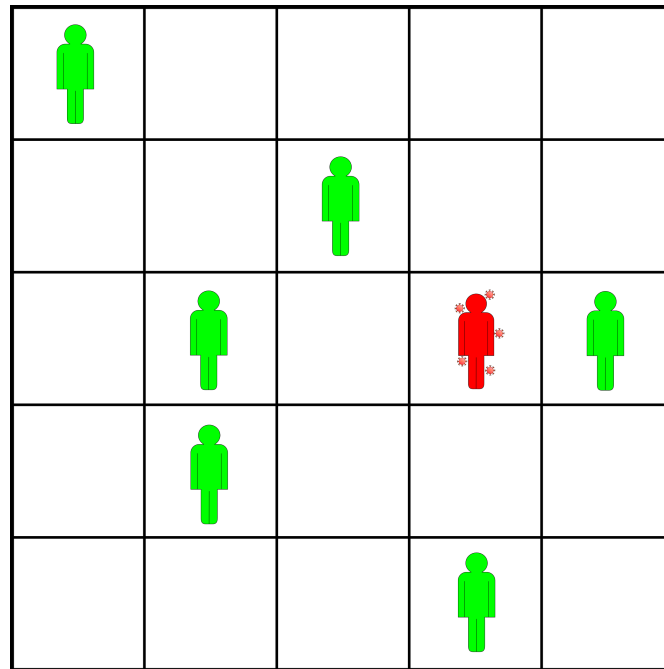


FIGURE 4.4 – Exemple d’une configuration initiale pour un système de taille  $5 \times 5$  avec 7 individus.

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 une 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 à 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.

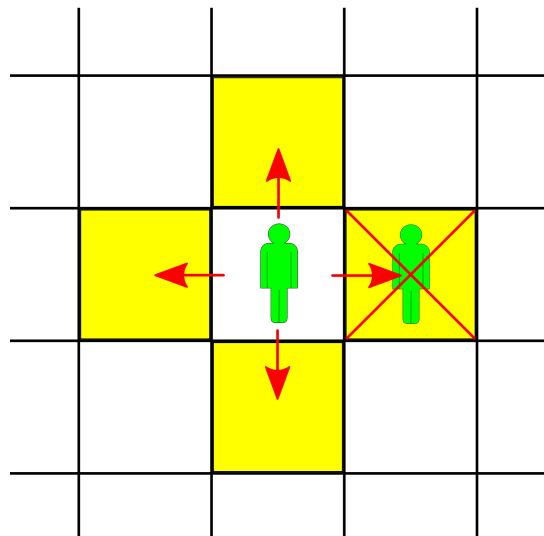
Pour être plus précis, une itération du système consiste en deux phases. La première est de mettre à jour tous les acteurs du système et la seconde est de permettre aux acteurs de se déplacer dans l’espace. La phase de mise à jour comprend l’actualisation de tous les acteurs ainsi que d’effectuer toutes 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.



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 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 "individu" 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 par déplacement. 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. Un paramètre du modèle permet d'effectuer plusieurs mouvement par itération, c'est-à-dire qu'un individu peut répéter  $n$  fois ce processus de mouvement si le paramètre est fixé sur  $n$ .



**FIGURE 4.5** – Lors d'un déplacement, un individu a 4 choix possible (cellules en jaune) et en choisi un aléatoirement. Si la cellule sur laquelle l'individu veut aller est occupée alors il ne se déplace pas. Le processus peut être répété plusieurs fois par itération.

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. Une tentative non fructueuse de déplacement est quand même considérée comme déplacement, c'est-à-dire que si un individu se déplace 5 fois par itération mais est bloqué 2 fois à cette itération alors il n'effectuera que 3 réels déplacements.

### 4.3.3 Mise à jour des acteurs

#### Actualisation des agents pathogènes

Un agent pathogène peut se trouver dans deux états distincts. La procédure de mise à jour ne se produit que dans le cas où un agent pathogène contamine une cellule sans individu. Dans cette configuration sa survie est incertaine. Par conséquent il se peut que cet agent pathogène meurt. La phase de mise à jour de cet acteur sert à déterminer s'il meurt à cette itération ou non.

#### Actualisation des individus

La procédure de mise à jour d'un individu dépend du fait qu'il soit infecté ou non.

1. Dans le cas où l'individu mis à jour est sain, la seule étape à effectuer est l'analyse du voisinage. Cette phase consiste à regarder les cellules voisines et déterminer si l'individu finit contaminé ou non à cette itération. Une éventuelle contamination dépend exclusivement du voisinage. Si l'individu est contaminé alors il intègre l'agent pathogène.
2. Dans le cas où l'individu est déjà contaminé il est inutile d'analyser le voisinage. Par contre cet individu met à jour son état et ses attributs et ce processus a deux issues possibles.
  - a) Dans le premier cas, l'individu se débarrasse de son agent pathogène et redevient sain.
  - b) Dans le second cas, l'individu conserve son agent pathogène ce qui lui laisse la possibilité de muter.

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

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 et permettent de faire évoluer le système. La première barrière à toutes interactions est la 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 des agents 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 en cas de contact. A l'opposé, une charge virale élevée signifie que les individus contaminés sont très contagieux et risquent de contaminer d'autres individus sains rapidement.

### 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 individus ne peuvent pas se trouver sur la même cellule à la même itération, le seul cas à analyser survient lorsqu'un individu se trouve sur une cellule déjà contaminée par un agent pathogène. Il y a deux cas de collisions possible dans toutes simulations.

1. Le premier cas se produit quand un individu 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 l'hôte de cet agent pathogène.
2. Le second cas se produit lorsqu'un individu contaminé se retrouve sur une cellule préalablement contaminée. L'agent pathogène contaminant la cellule n'a aucun impacte 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 mais sur des cellules différentes. Afin de permettre aux acteurs d'interagir sans être sur la même cellule, ils doivent appartenir à leur voisinage respectif. Seuls les individus peuvent effectuer ce type d'interactions. Par exemple, deux individus sur deux cellules adjacentes interagissent. Les différentes interactions possible dans le modèle sont développées ci-dessous.

1. Dans le cas où nous avons deux individus 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. 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.
3. 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'individu contaminé pourrait avec une certaine probabilité se propager sur l'individu sain. Un individu sain peut donc être contaminé par un individu contaminé seulement si ce dernier se trouve dans son voisinage.

#### 4.4.3 Contamination de cellule

Une cellule est porteuse d'un agent pathogène si cet espace a été contaminé par un individu infecté. En effet, lorsqu'un individu 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 individus contaminés infectent souvent les cellules qu'ils visitent et à l'inverse pour une probabilité faible.

Après un déplacement, l'individu contaminé à une certaine probabilité d'avoir contaminé la cellule qu'il occupait précédemment. Si la contamination a lieu, 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 individu contamine un espace déjà contaminé par un autre agent pathogène. C'est-à-dire qu'un individu contaminé se trouvant sur une cellule préalablement contaminée et essaie de la contaminer à nouveau. Nous avons dit précédemment qu'un individu 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 sur la surface et le remplaçons par une copie de l'agent pathogène de l'individu.

### 4.5 Diversité génétique

Le modèle se concentre sur la notion de diversité. Il faut donc un moyen pour représenter la diversité d'une population et gérer les interactions entre différents acteurs en prenant en considération leur diversité. La méthode utilisée se base sur des codes attribués à tous les acteurs du système. Chaque acteur possède un code et ce dernier simule son matériel génétique. Par conséquent, chaque individu et chaque agent pathogène possède un code que nous allons appeler "génom". Ce génome est un nombre entier non signé et se présente sous la forme d'une séquence codée 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 les génomes 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 très différents les uns des autres. Le modèle intègre un paramètre permettant de définir le niveau de diversité des individus. L'unique agent pathogène initial n'a pas de notion de diversité, son génome est directement fixé dans les paramètres de la simulation.

La première étape de la générations de génomes est de commencer par attribuer à tous les individus un génome fixe et identique défini dans les paramètres de la simulation. Par conséquent, tous les individus ont ce même génome de référence, la diversité est nulle dans cette configuration. Il est ensuite possible de modifier ces génomes en complétant un certain nombre de bits à partir de cette séquence de référence. Le processus de complémentation se fait aléatoirement et le nombre de complémentation dépend du paramètre de diversité du modèle. Par exemple, si le paramètre de diversité est défini à 5 alors tous les individus effectueront 5 modifications aléatoires sur leur séquence de référence. Etant donné que les changements sont aléatoires, les individus vont finir avec des génomes différents. Avec cette méthode nous finissons avec des génomes déviant plus ou moins d'un certain génome de référence.

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 de savoir comment interpréter et utiliser des génomes. Comment gérer les comportements des acteurs 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 permettant d'interpréter les génomes et de prendre des décisions. La principale utilité de cette interprétation survient avec les individus contaminés. Nous nous intéressons aux génomes afin de déterminer les compatibilités entre des individus et des agents pathogènes.

La distance de Hamming est une notion mathématique qui permet de calculer les différences entre deux séquences de symboles. Cette technique consiste simplement à compter le nombre de symboles différents pour deux suites de même longueur. 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 qui est la distance de Hamming entre les deux séquences. 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 nous définissons un seuil sur si peu de valeurs,

les résultats vont être trop tranchés. Par conséquent nous convertissons cette distance calculé entre deux génomes en probabilité. Les actions à effectuer sont ensuite déterminée en fonction de cette probabilité.

#### 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.

$$\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}$$

+1 +1 +1 +1 +1 =5

FIGURE 4.6 – Calcul de la distance de Hamming entre deux séquences binaires.

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.

Le calcul de la distance de hamming sert à déterminer si une actions se déroule ou non. Il est donc nécessaire d'interpréter la distance de Hamming en lui donnant 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 en une forme utilisable. Un mécanisme de traduction produit une probabilité à partir d'une distance de Hamming. Dans ce cas nous aurions pu utiliser un seuil au delà duquel l'action s'exécute. Au lieu de ce mécanismes, nous générons 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. Inversement l'agent pathogène est peu efficace si son génome est "éloigné" de celui de l'individu.

L'utilisation de la distance de Hamming ne survient que dans la situation où un individu est déjà contaminé. Les individus contaminés, lors de leur mise à jour calculent la distance de Hamming qu'ils ont avec leur agent pathogène. Cette distance est ensuite traduite en probabilité et cette dernière détermine la chance qu'à les individus d'éliminer leur agent pathogène.

## 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 individu.

L'immunisation est une résistance acquise, c'est-à-dire qu'un 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 individu reste contaminé pendant un certain nombre d'itérations puis finit par se débarrasser du pathogène alors il s'immunise.

Le mécanisme de résistance naturelle est identique au mécanisme d'immunisation, la seule différence est temporelle. Si un individu parvient à se débarrasser rapidement de son agent pathogène alors c'est une résistance naturelle. Par contre si il faut davantage de temps, c'est une immunisation. Un paramètre du modèle définit le seuil en itérations découpant les deux processus. Par exemple, si le paramètre est défini à 5 alors tous les individus qui parviennent à se débarrasser de leur agent pathogène en moins de 5 tours (depuis leur contamination) sont considérés comme naturellement résistants. Les autres seront immunisés s'ils parviennent à éliminer le pathogène.

La particularité de l'immunisation est que le modèle intègre les immunités aux pathogènes sous la forme d'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 uniquement 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 pas développer des immunités à des groupes similaires de pathogènes, c'est des immunités au cas par cas.

Contrairement aux immunités, les résistances naturelles ne sont pas une réponse immunitaire. Par conséquent, les individus ne développent pas de résistance particulière aux agents pathogènes, ils sont "génétiquement" résistants.

## 4.7 Mutations

Le modèle ne représente que des situation qui se déroule 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 acteurs à pouvoir muter dans le modèle.

Le modèle intègre un paramètre déterminant la probabilité qu'ont les agents pathogènes de muter à chaque itération. Seuls les agents pathogènes contaminant un individu peuvent muter. A chaque itération, l'individu essaie de se débarrasser de son agent pathogène. Si il échoue, le pathogène a la possibilité de muter. Une mutation se produit avec une certaine probabilité définie dans le modèle.

Une mutation est une modification du génome de l'acteur. Lorsqu'un agent pathogène mute il modifie un seul bit de sa séquence choisi aléatoirement.



## 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. L'instanciation d'une simulations se fait de la manière suivante :

```
1 threads.push_back(std::thread(thread_function, "./path_exemple"));
```

Le programme utilise des threads afin de gagner en performances. Etant donné que les simulations sont difficilement parallélisables nous pouvons uniquement lancer plusieurs simulations simultanément. Une variable de précompilation dans le main définit le nombre de thread alloué au programme.

```
1 #define NB_THREADS 8
```

Cette valeur signifie que 8 simulations peuvent être exécutée simultanément. Définissez cette valeur en fonction des ressources disponibles.

Une simulation nécessite un chemin où les informations seront lues et stockées. Afin de pouvoir exécuter une simulation il est nécessaire d'y ajouter un fichier de configuration contenant les paramètres de la simulation. Dans l'exemple, le dossier "path\_exemple" est accédé depuis la racine du projet et doit contenir le fichier de configuration de la simulation.

Le chemin vers le dossier est la racine de la simulation. Le fichier de configuration dans le dossier de la simulation doit avoir le nom suivant : **config.txt**. A la fin de la simulation, le programme écrit les résultats dans un sous-dossier nommé "data\_csv". Les résultats sont des fichiers CSV contenant des informations sur la simulation.

Un fichier de configuration est un fichier texte contenant des lignes formées de mots clés 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 lexicaux 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. La simulation ainsi que les individus et les agents pathogènes sont des objets. Le nombre d'individus du système est déterminé par le paramètre "NOMBRE\_INDIVIDUS" et chaque individu est pointé par un pointeur appartenant à une liste. Cette liste est de taille fixe et contient tous les individus du système. Contrairement aux individus, les objets 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 s'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. Deux grilles sont instanciées comme des matrices de pointeurs pouvant pointer sur des objets. Une grille est dédiée à pointer sur les individus et une autre sur les agents pathogènes. Il est nécessaire d'utiliser deux grilles car il se peut que un individu et un agent pathogène se retrouvent sur la même cellule.

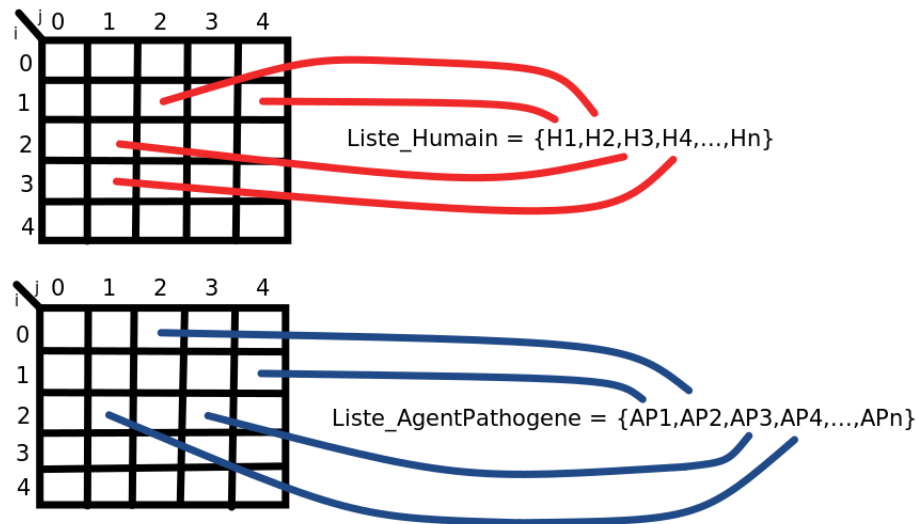


FIGURE 5.1 – L'espace est représenté par deux matrices de pointeurs de la taille du système. Chacune de ces matrices pointe exclusivement sur un type d'acteurs.

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 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 complémenté. Ce mécanisme permet d'obtenir des génomes avec un certain degré de proximité à une certaine valeur choisie. Le génome de l'unique agent pathogène initial est 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 pathogène initial.

### 5.3 Exécution de la simulation

Une grande majorité du travail de la simulation s'effectue à cette étape. Cette phase se répète un nombre de fois équivalent à la valeur du paramètre "ITERATIONS". Chaque itération sert à actualiser, à un moment donné le système en entier.

## Permutation

La première 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 permutons 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.

## Agents pathogènes

Vient ensuite le phase de mise à jour des 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.

## Individus

L'étape d'actualisation et de déplacement des individus est l'étape la plus coûteuse en temps. L'actualisation d'un individu se fait en deux étapes. La première étape consiste à calculer les interactions et mettre à jour les états et la seconde est d'effectuer un ou plusieurs déplacements.

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. Dans cette situation, notre individu n'a que trois issues. La première est d'être naturellement résistant au pathogène. La seconde est de s'immuniser au pathogène et la troisième est de conserver le pathogène.

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. 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 individu 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. 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  $[0, 32]$ .

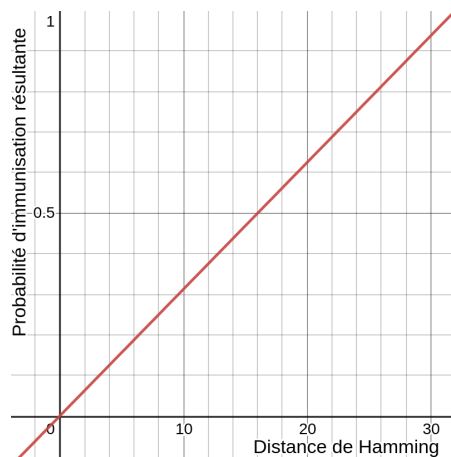


FIGURE 5.2 – Fonction de conversion de la distance de Hamming en probabilité avec  $\text{PARAMETRE\_FONCTION} = 1$

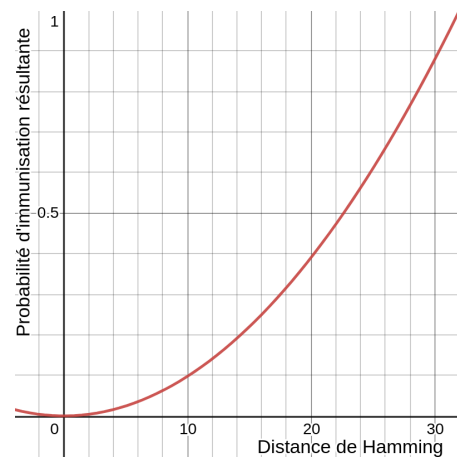


FIGURE 5.3 – Fonction de conversion de la distance de Hamming en probabilité avec  $\text{PARAMETRE\_FONCTION} = 2$

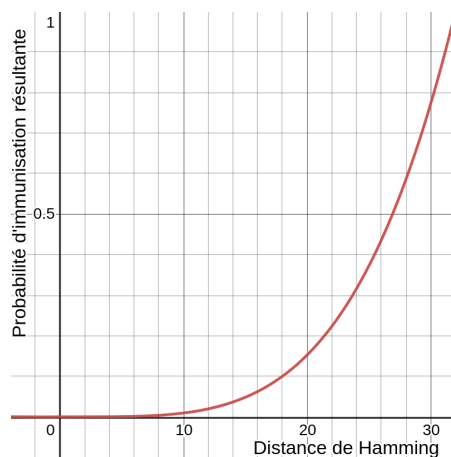


FIGURE 5.4 – Fonction de conversion de la distance de Hamming en probabilité avec  $\text{PARAMETRE\_FONCTION} = 4$

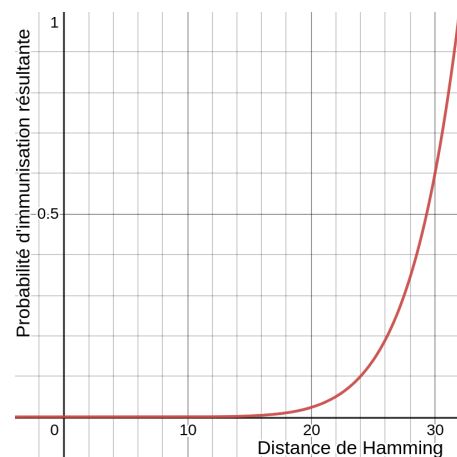


FIGURE 5.5 – Fonction de conversion de la distance de Hamming en probabilité avec  $\text{PARAMETRE\_FONCTION} = 8$

Les 4 fonctions sont bornées entre  $[0, 1]$ . L'axe en abscisse représente la probabilité de rejeter l'agent pathogène en fonction de la distance de Hamming. Le paramètre de

fonction permet de modifier l'allure de la courbe sans modifier les bornes. Une petite valeur de "PARAMETRE\_FONCTION" traduit de grandes probabilités d'éliminer l'agent pathogène et inversement une grande valeur du paramètre traduit en de faibles probabilités d'éliminer le pathogène.

Comme expliqué ci-dessus, à chaque itération, chaque individu contaminé a une chance de se débarrasser de son pathogène. Si l'individu 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'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 qu'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". Ce 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 est inévitablement contaminé par le pathogène.

La phase de déplacement se produit après la phase de mise à jour. A chaque itération un individu peut se déplacer un nombre de fois égal au paramètre "NOMBRE\_MOUVEMENT". 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 donné 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.

## Mesures

L'étape finale est la prise de mesures lors de la simulation, le traitement des données et l'écriture dans les fichier CSV. Les données récoltées sont le nombre de contaminés,

le nombre d'agents pathogènes différents, le nombre d'individus ayant au moins une immunité ainsi que quelques autres informations relative à la simulation. Faire des mesures et analyses pendant la simulation nous évite de devoir sauvegarder toutes les données.

## **5.4 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 mémoires 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 une abstraction simplifiée de fonctionnements réels et de l'implanter dans le modèle afin de simuler des processus réels très complexes. Cette simplification permet une analyse plus aisée des résultats. En effet, les corrélations et causalités sont plus facilement interprétables lorsque le modèle est simplifié.

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 aux modèles compartimentaux en épidémiologie pour tester notre 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 observer 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 variations de comportements en faisant varier un ou plusieurs paramètres du modèle. La recherche se concentre sur la notion de diversité, introduite dans le modèle par les génomes des acteurs. Le facteur de diversité est le paramètre principal étudié. Quel est l'impacte de la diversité d'une population sur la propagation d'une épidémie ? La variation d'autres paramètres peuvent aussi être étudiée comme par exemple la vitesse de mutation des agents pathogènes. Il s'agit donc d'illustrer et de quantifier l'impacte qu'ont ces paramètres sur l'émergence d'événement de grande taille comme les pandémies.

Un modèle performant est nécessaire pour effectuer des simulations de grande taille. Dépendant des performances du modèle, quelles sont les ordres de grandeur des simulations effectuées ? et quel quantité d'expériences pouvons nous effectuer ? En effet le travail cherche à observer des événements à grande échelle que le modèle doit supporter



et pouvoir répéter un certain nombre de fois.

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 pourrions nous faire de la prédiction ? Le modèle ne se limite pas à deux espèces en particulier mais à deux catégories d'espèces. Nous pouvons donc appliquer le modèle à un grand nombre de cas.

Le dernier objectif de la recherche est de pouvoir faire de la prédiction explicative. Contrairement à la prédiction 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 Compartmentaux

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 moments 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 types 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 infectée. 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 symptôme 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 explicitées dans la description du modèle.

L'objectif de ce chapitre est de valider 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 a 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 mathématiquement par les équations différentielles ordinaires suivantes :

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

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.

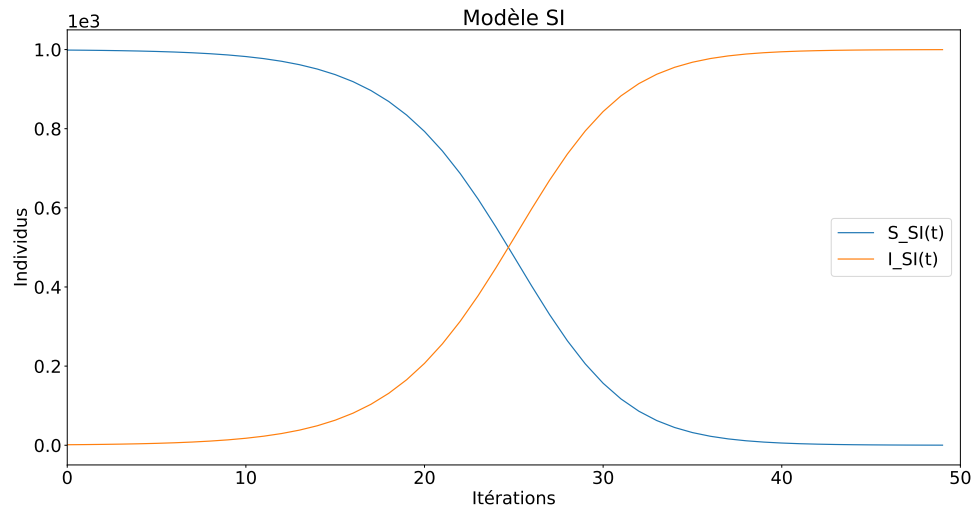


FIGURE 7.1 – Modèle mathématique SI construit par les équations différentielles ordinaires.

## 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 même 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 :  $N = 1000, S = 999, I = 1, R = 0, \beta = 0.3, \gamma = 0.06$  :

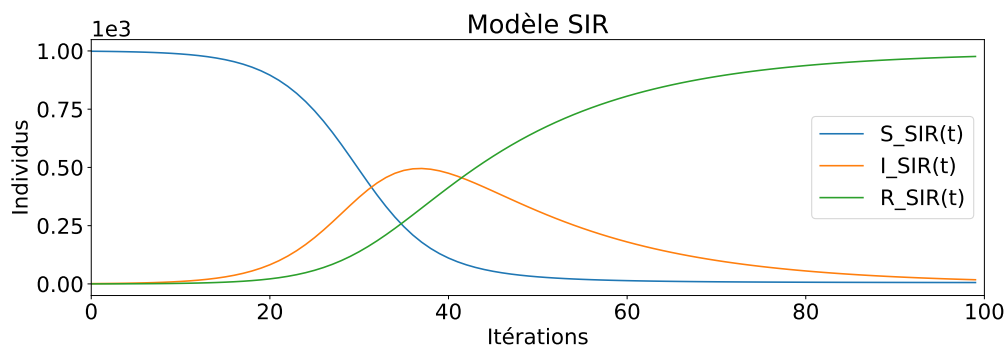


FIGURE 7.2 – Modèle mathématique SIR construit par les équations différentielles ordinaires.

## 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 paramétrer 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 paramétrer 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 atténué 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.

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_MECANISME = false
18 RESISTANCE_MECANISME = false
```

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

```
1  IMMUNITE_MECANISME = false
2  RESISTANCE_MECANISME = false
```

Il faut ensuite paramétrer 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_MECANISME = false
9  RESISTANCE_MECANISME = 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 paramétrer 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

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_MECHANISME = true
18 RESISTANCE_MECHANISME = 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_MECHANISME = 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 paramétrer les génomes initiaux, le paramètre de fonction ainsi que les autres paramètres du modèle.



## 8 Modèle SI

### 8.1 Mesures et méthodologie SI

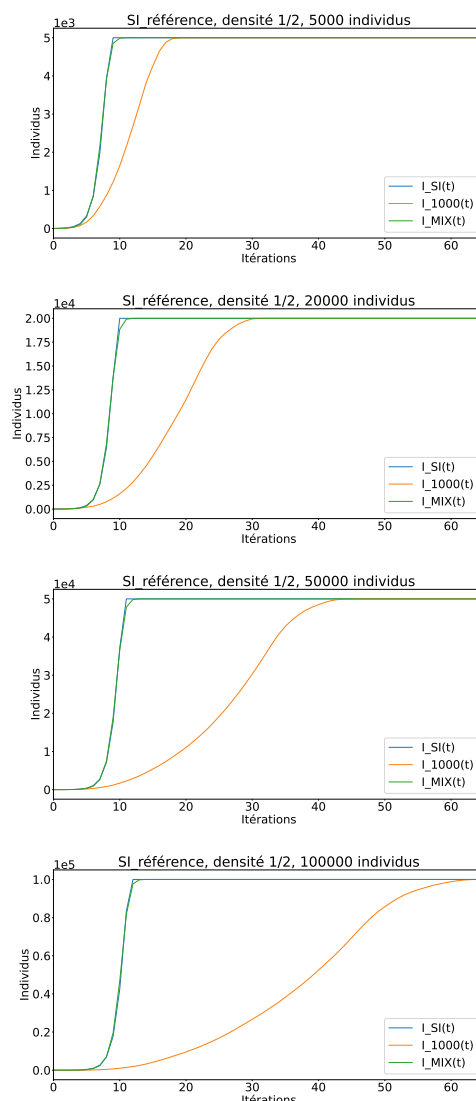
L'objectif de ces résultats est de valider le modèle implémenté. Toute une série de mesures ont été prise pour différentes densités de population et pour différentes tailles de système. Quatre niveaux de densité de population ont été mesuré ainsi que quatre taille de population. Les densité choisies 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 le 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



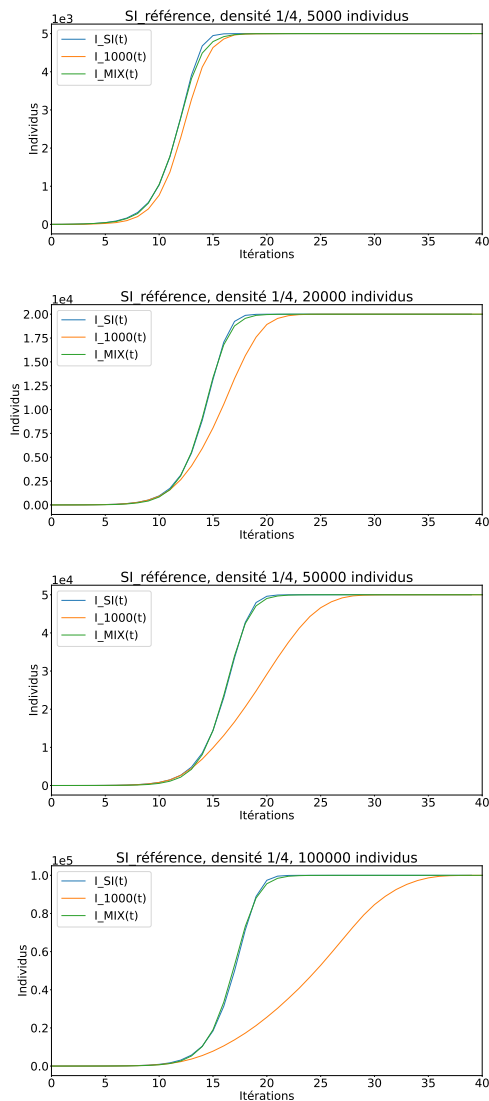
**FIGURE 8.1** – Simulations du modèle SI sur des systèmes en densité 1/2 avec le nombre d'individu 5000, 20000, 50000 et 100000. Les courbes bleues proviennent du modèle mathématique. Les courbes vertes proviennent des simulations au mélange parfait et les courbes oranges proviennent des simulations aux 1000 mouvements.

plus grands.

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.

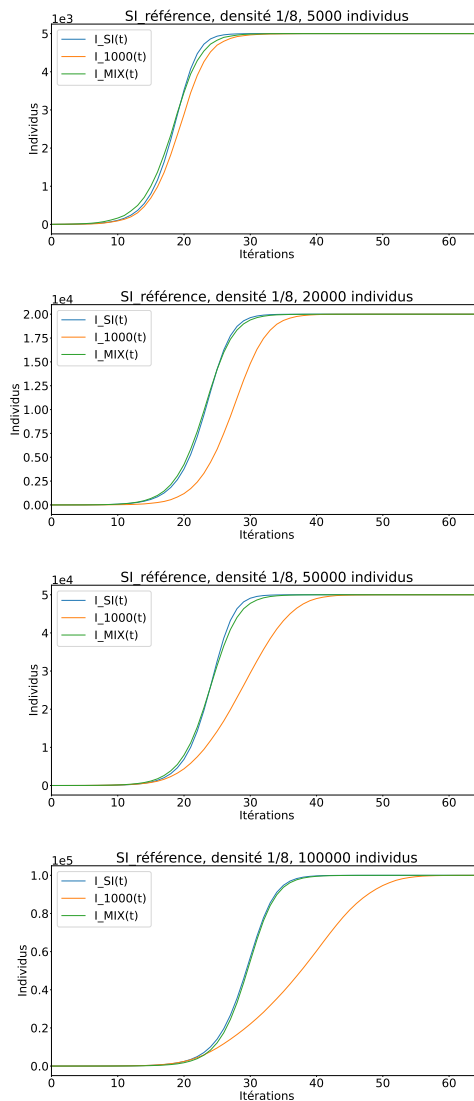
De même pour cette densité, le taux de mélange de 1000 mouvements fixe impacte les grands systèmes en s'éloignant du modèle SI. Par contre la simulation sur 5000 individus particulièrement précise et ceci bien plus que les simulations en densité  $\frac{1}{2}$ . Ceci est dû au fait qu'une densité de  $\frac{1}{4}$  permet aux acteurs 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 du modèle SI sur des systèmes en densité  $1/4$  avec le nombre d'individu 5000, 20000, 50000 et 100000. Les courbes bleues proviennent du modèle mathématique. Les courbes vertes proviennent des simulations au mélange parfait et les courbes oranges proviennent des simulations aux 1000 mouvements.

Les mêmes comportements sont observés pour les simulations en densité  $\frac{1}{8}$ . La qualité du mélange décroît avec l'augmentation 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 déplacer 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 dû 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 du modèle SI sur des systèmes en densité  $1/8$  avec le nombre d'individu 5000, 20000, 50000 et 100000. Les courbes bleues proviennent du modèle mathématique. Les courbes vertes proviennent des simulations au mélange parfait et les courbes oranges proviennent des simulations aux 1000 mouvements.

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 déclencher. 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 ce 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 simulation (5000 individus), la courbe des 1000 mouvements est retardée. Sur les trois autres c'est l'effet inverse. Ce comportement est illustré et mesuré plus loin.

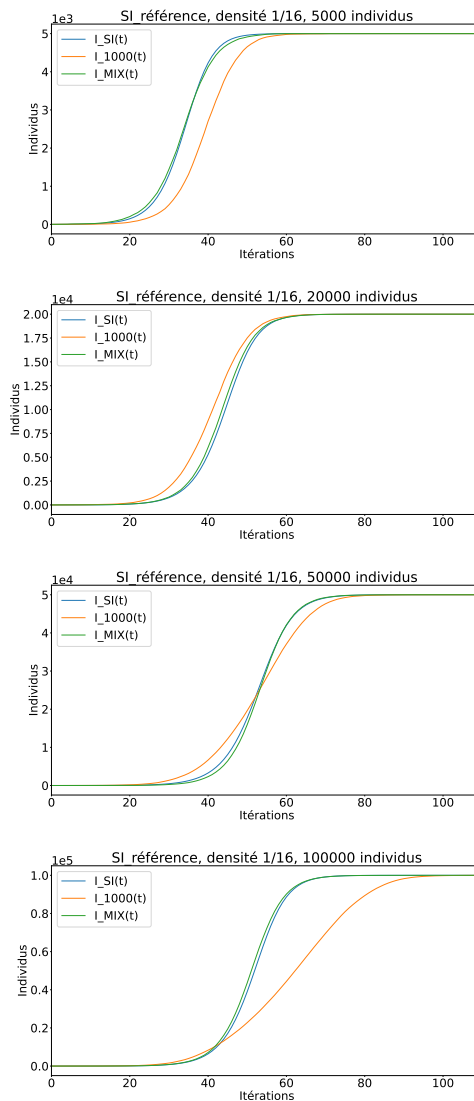


FIGURE 8.4 – Simulations du modèle SI sur des systèmes en densité 1/16 avec le nombre d'individu 5000, 20000, 50000 et 100000. Les courbes bleues proviennent du modèle mathématique. Les courbes vertes proviennent des simulations au mélange parfait et les courbes oranges proviennent des simulations aux 1000 mouvements.

## 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 Absolute Error Normalized : Erreurs relevées des simulations SI pour toutes les densités et nombre d'individus. L'erreur est calculée entre les simulations au mélange parfait et le modèle mathématique.

	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 croît pas avec la taille du système. Pour ces analyses, la magnitude 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 dédiée 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 locale 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** – Calcul du nombre moyen de voisin par individu pour les systèmes de densité  $1/8$  et  $1/16$  avec les deux méthodes de déplacement.

	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 conséquemment les phénomènes observés dans les figures ne sont pas causés par la densité du voisinage. De plus les résultats nous réconfortent 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 déterministes, 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.

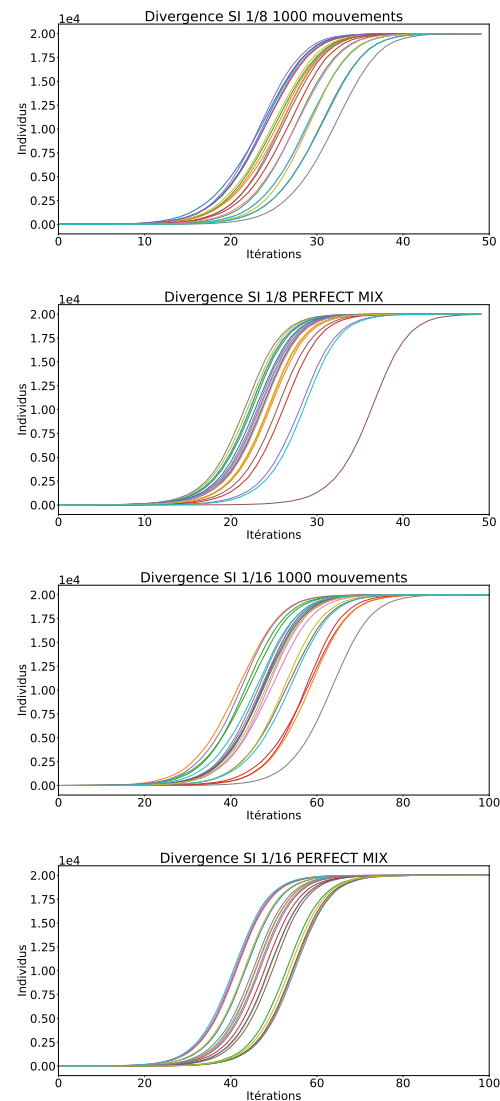


Sur chacune des figure sont représentées les 20 simulations aux mêmes paramètres. Les mesures effectuées sur les simulations permettent de mettre ces courbes en relations les unes avec les autres. L'intérêt des mesures est de calculer les décalages sur l'axe des abscisses. Pour ce faire nous mesurons 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.

**TABLE 8.3** – Résultats sur les variations aléatoires des simulations. Pour chaque densité et type de mouvements, les mesures sont les itérations à laquelle la moitié de la population est contaminée. Nous mesurons donc les simulations les plus rapides à atteindre ce 50% (min), les simulations les plus lentes (max), l'itération moyenne pour atteindre le 50% et finalement l'écart à la moyenne (std).

	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



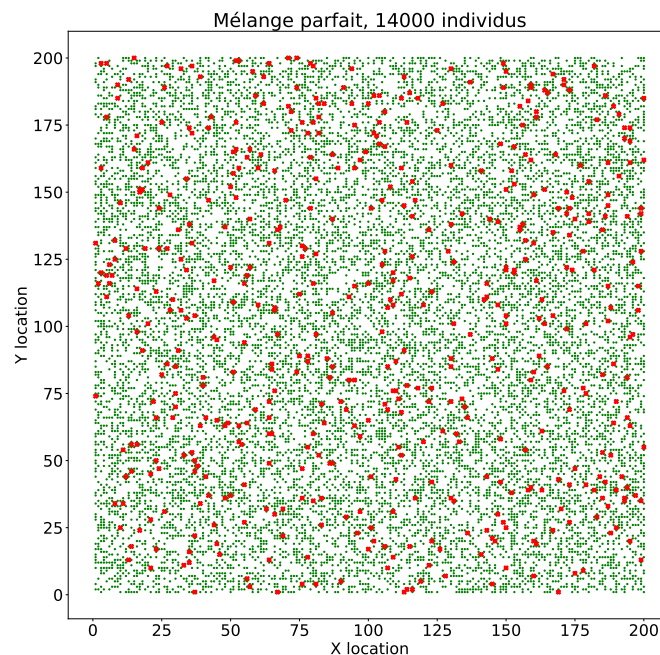
**FIGURE 8.5** – Chaque figure comporte 20 courbes issues de simulations identiques. Les systèmes sont de densité 1/8 pour les deux premières figures et 1/16 pour les deux dernières. Les courbes sont identiques à l'exception des décalages sur l'axe des abscisses.

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

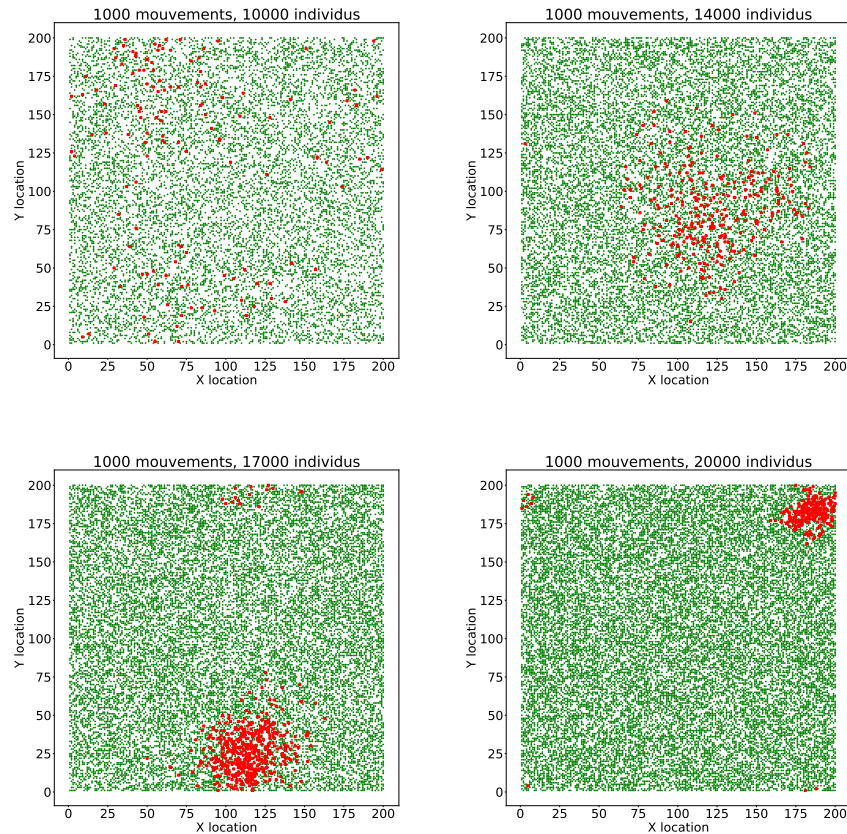
Cette section a pour but de visualiser l'impacte de la densité des systèmes sur les déplacements des individus et donc sur la qualité du mélange. Les figures suivantes montrent les positions géographiques de tous les individus à une certaine itération. En vert sont affichés les individus sains et en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

L'image ci-dessous montre les positions des individus pour une simulation au mélange parfait à l'itération 7. 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** – Exemple de distribution d'individus contaminés par le mélange parfait dans un espace. La figure est une image du système à l'itération 7. Les marques rouges sont les individus contaminés et les marques vertes sont les individus sains.

Sur les figures qui suivent, nous effectuons la même mesure mais cette fois-ci sur des simulations aux 1000 mouvements. Le nombre d'individus est de 10000, 14000, 17000 et 20000 nous onvons donc quatre simulations aux densité différentes mais paramétrées de manière identique. L'idée est de pouvoir observer la qualité du mélange de la population en fonction de la densité du système.



**FIGURE 8.7** – Exemple de distribrution des individus contaminés à l’itération 7 dans des systèmes de densités différentes. Les 4 simulations ont un espace de  $200 \times 200$  mais le nombre d’individus varie de 10000 à 20000.

Afin de pouvoir comparer les simulations, nous avons fixé la taille du système à  $200 \times 200$  et faisons varier uniquement le nombre d'individus. La dispersion des individus confirme le fait que les mouvements sont entravés dans les simulations aux 1000 mouvements. Pour rappel, en mode 1000 mouvements, un individu essaie de se déplacer 1000 fois par itération mais si le passage est obstrué, l'individu ne se déplace pas et ceci est compatible comme mouvement.

Par conséquent, le diamètre de propagation croit avec la diminution de la densité du système.

## 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

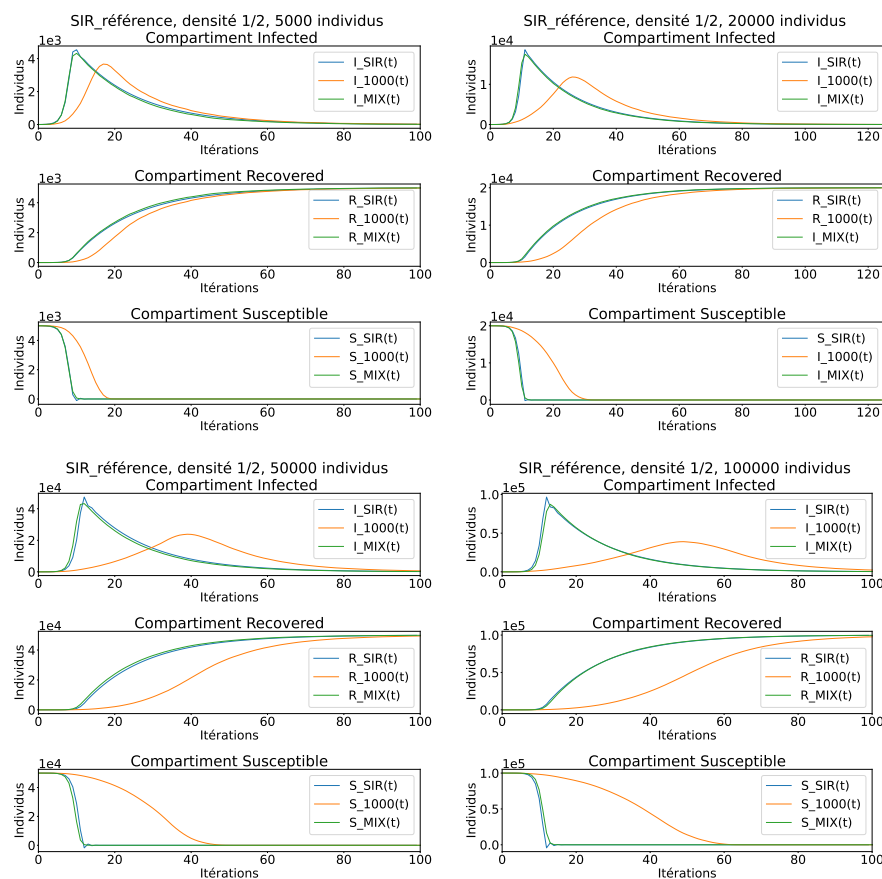


FIGURE 9.1 – Simulations de SI, densité 1/2

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.

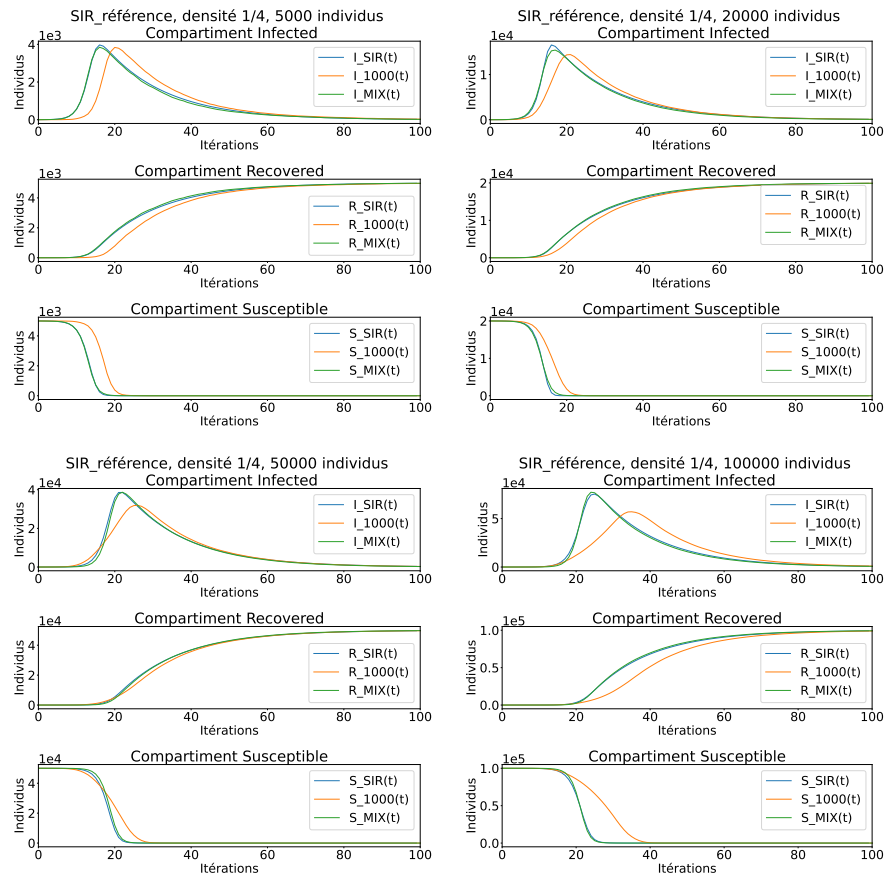


FIGURE 9.2 – Simulations de SI, densité 1/4

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.

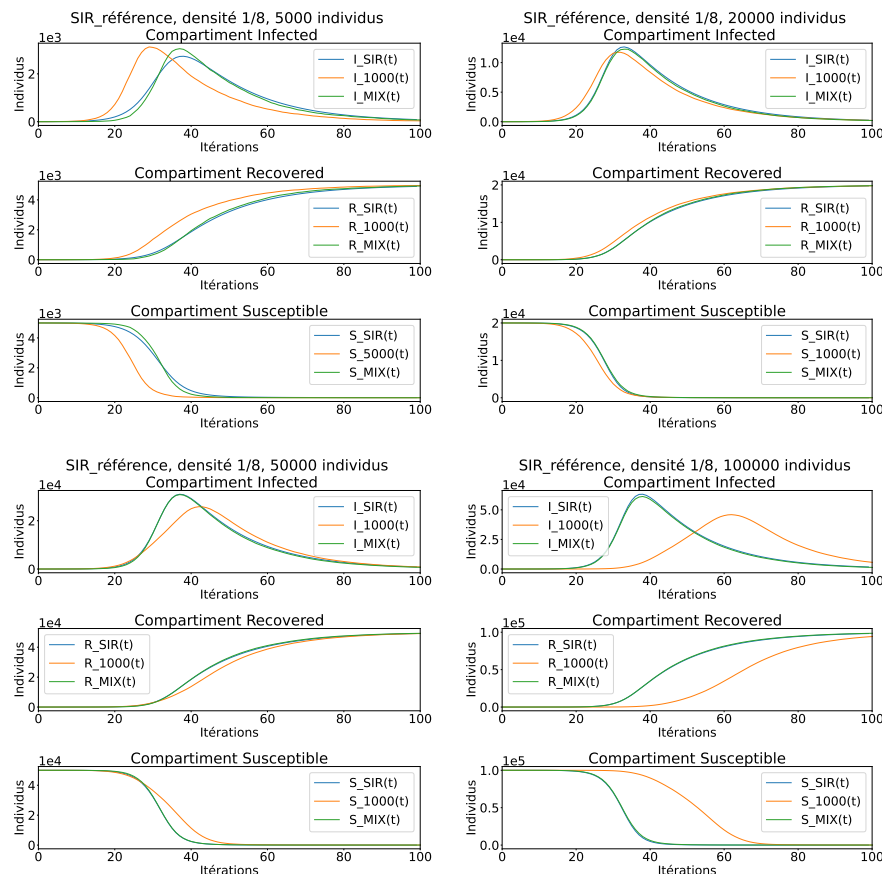


FIGURE 9.3 – Simulations de SI, densité 1/8

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.

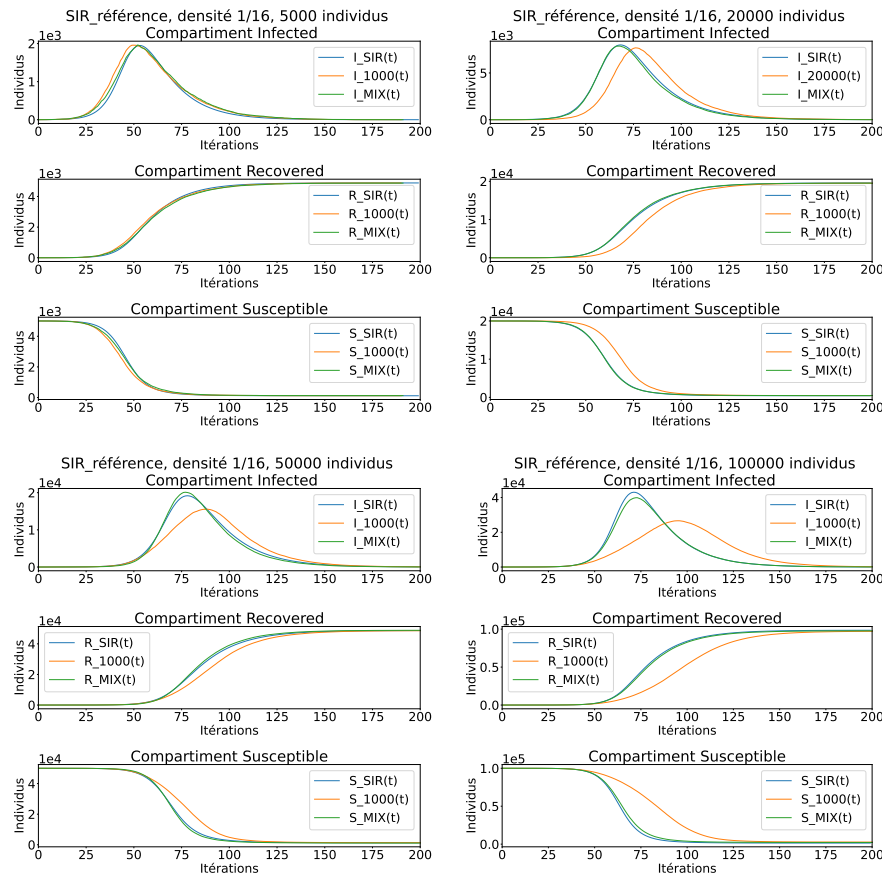


FIGURE 9.4 – Simulations de SI, densité 1/16

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 observons 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 expliquent ce comportement, le premier est que le mécanisme d'immunisation transfère 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 moyennées 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 : Calcul des erreurs entre les simulations au mélange parfait et le modèle mathématique SIR sur 4 niveaux de densités et 4 niveaux de populations.

	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  $\frac{1}{16}$  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 à déclencher un événement, ce qui engendre un temps de latence. Ce temps de latence est inexistant 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.

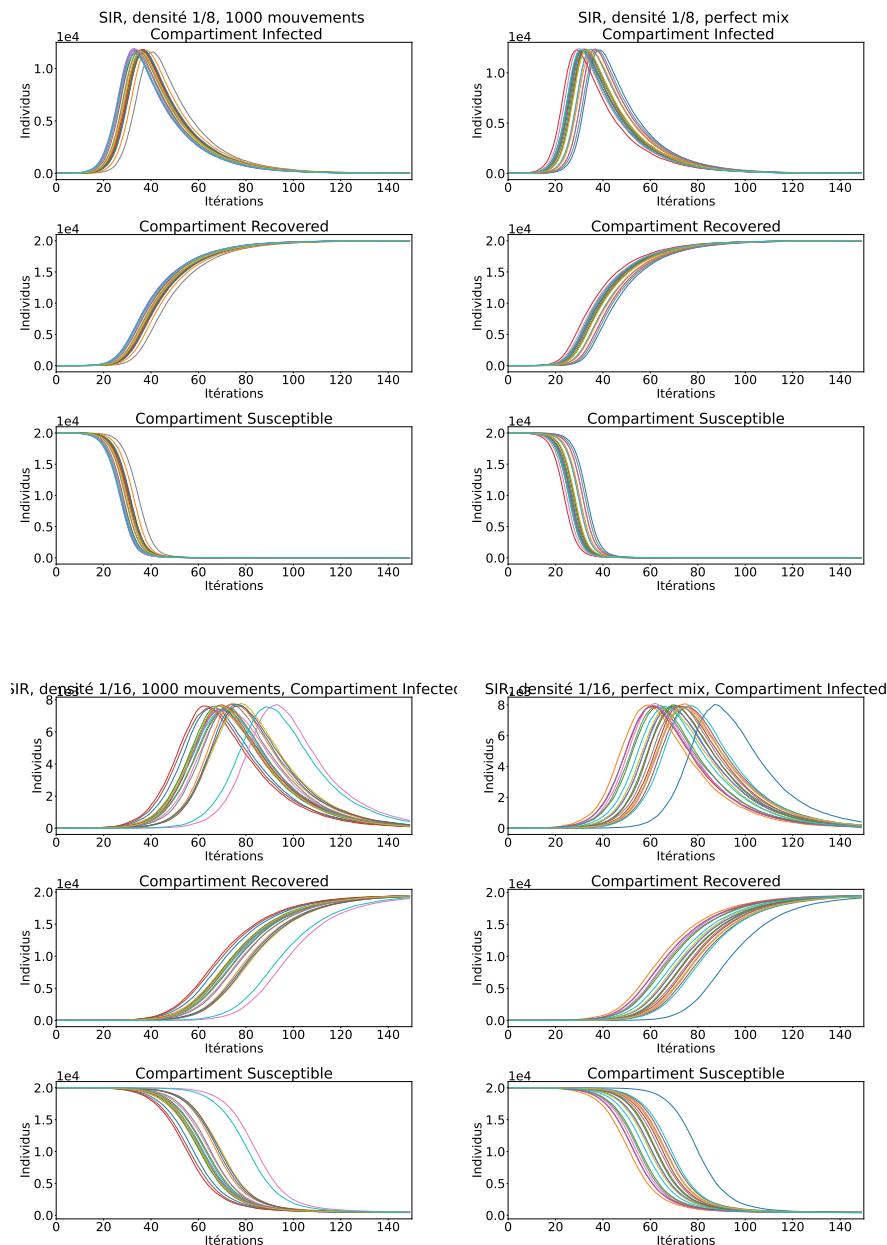


FIGURE 9.5 – Variations SIR

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** – Calcul des variations aléatoires sur 20 simulations identiques. La mesure est prise pour les deux modes de mouvements ainsi que pour deux densités lorsque les 50% de contaminés surviennent. La mesure du minimum est la mesure de la simulation la plus rapide et le maximum est la plus lente. La moyenne ainsi que l'écart à la moyenne est aussi calculé.

	1000 mouvements		Mélange parfait	
densité	1/8	1/16	1/8	1/16
min	33	62	29	59
max	41	93	38	87
mean	35.35	72.85	33.2	69.35
std	2.10	7.28	2.58	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.

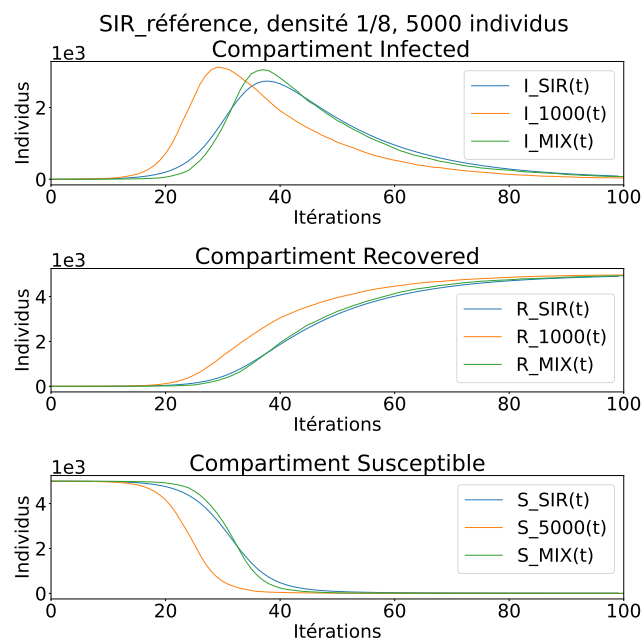
### 9.3.3 Latence des simulations

Certaines simulations du modèle implémenté divergent du modèle SIR. Une des hypothèse 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 par contre elle est inexistante 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 des systèmes 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.

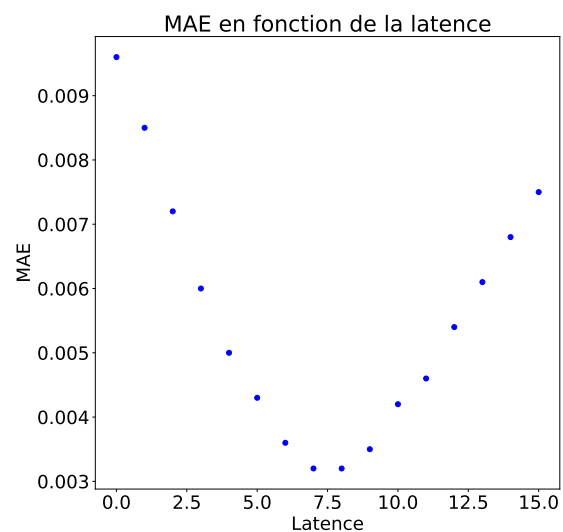


**FIGURE 9.6** – Exemple de simulation SIR avec de la latence. La courbe bleue représente le modèle mathématique, la verte, la simulations au mélnage parfait et la orange la simulations aux 1000 mouvements. Nous nous intéressons ici au mauvais fit du modèle mathématique sur la courbe de la simulation au mélange parfait.

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 subi une latence de environ 7 itérations.

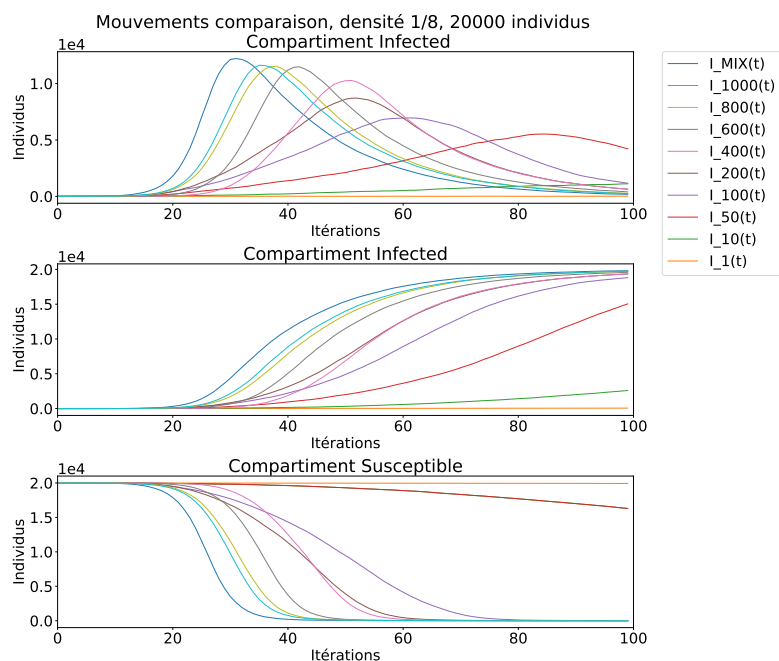


**FIGURE 9.7** – Mesure de la latence de la simulation au mélange parfait. Sur les abscisses nous avons de la latence artificielle que nous ajoutons au modèle SIR et sur les ordonnées nous avons le MAE entre le modèle mathématique et la simulation. L'idée est d'ajouter de la latence au modèle mathématique et chercher un meilleur fit.

### 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.



**FIGURE 9.8** – Comparaison de simulations aux mêmes paramètres sauf que le nombre de mouvements par individu change. Les simulations vont de un seul mouvement par itération et par individu à 1000 mouvements. La courbe au mélange parfait est aussi présente comme référence.

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 ici significativement.

### 9.3.5 Comparaison 1000 mouvements

Afin d'étudier la scalabilité du modèle sur de grandes simulations, nous comparons des simulations aux mêmes paramètres et même densité mais de taille différente. Le but est d'observer les différences entre des simulations presque identiques mais de taille différente. Pour pouvoir comparer des simulations de population différente il faut normaliser les résultats, c'est-à-dire diviser les valeurs par le nombre d'individus.

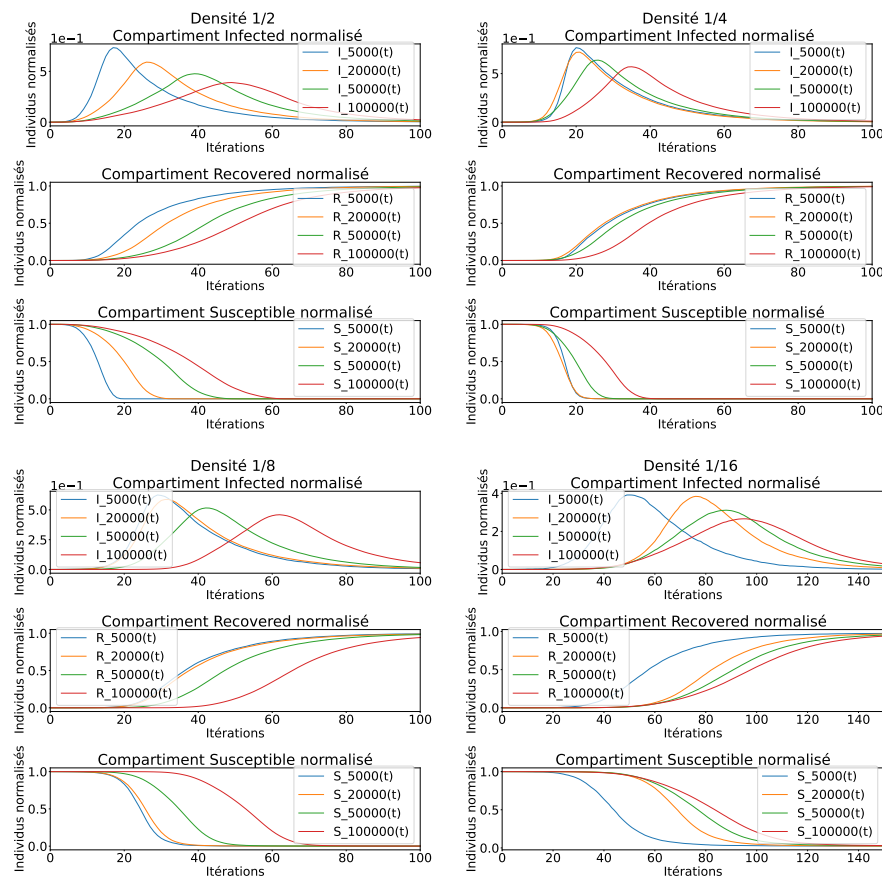


FIGURE 9.9 – Comparaison 1000 mouvements

Nous pouvons observer deux phénomènes. Le premier est le fait que plus le nombre d'individus est élevé, plus la pandémie prend du temps à se propager, ce qui paraît assez logique car il faut contaminer d'avantage d'individus. Le deuxième phénomène est le fait que le pic d'infectés décroît lorsque nous augmentons le nombre d'individus. Mais cette décroissance n'est pas constante sur les quatre densités testées. En effet il semblerait que plus la densité soit élevée, plus les variations entre les pics sont grands.

Ce dernier phénomène est conforme à ce que nous avons étudié précédemment. Les systèmes à forte densité peinent à se mouvoir et propager une pandémie. Par conséquent le rayon de propagation de la pandémie est fixe pour une certaine densité et ne dépend pas du nombre d'individus. Nous avons donc les systèmes à forte densité et grande taille qui peinent à contaminer la population, ce qui produit des événements plus

progressif et donc avec un pic de contaminés moins élevé. Par contre les simulations moins denses pénalisent moins les grands systèmes car le mélange s'effectue mieux. Avec un bon mélange, le nombre de personne n'impacte pas la vitesse de propagation. Cette situation est comparable à concaténer plusieurs simulations de plus petite taille pour en obtenir une grande. Dans cette configuration, les  $N$  simulations constituant la grande ne prennent pas  $N$  fois plus de temps que l'exécution d'une seule.

## 10 Diversité

### 10.1 Mesures et méthodologie

L'objectif du travail est de quantifier l'impacte de la diversité sur la propagation de pandémies. Cette section est dédiée à la prise de mesures avec des niveaux de diversité différents et d'en constater les résultats. Afin de ne pas trop complexifier le modèle au-delà des simulations SIR, nous ne modifions que le paramètre de diversité ainsi que la charge virale. D'autres paramètres du modèle pourraient être étudiés mais ce chapitre ne les explore pas.

Toutes les mesures de diversité ont été effectuées sur des systèmes aux paramètres semblables. La taille des systèmes est définie à  $1264 \times 1264$  avec une population de  $10^5$  individus, il s'agit de systèmes en densité  $\frac{1}{16}$ . Un exemple de fichier de configuration est donné ci-dessous.

```
1
2     TAILLE_SYSTEME = 1264
3     NOMBRE_INDIVIDUS = 100000
4     ITERATIONS = 5000
5     RERUN_LIMIT = 100
6     FAIL_SEUIL = 30
7     GENOME_INIT_I = 0
8     GENOME_DIVERSITY_I = 8
9     GENOME_INIT_AP = 0
10    VITESSE_MUTATIONS_AP = 0
11    CHARGE_VIRALE = 1
12    PARAMETRE_FONCTION = 4
13    CELLULE_AP = 0
14    SURVIE_AP = 0
15    NOMBRE_MOUVEMENT = 1
16    PERFECT_MIX = true
17    TEMPS_AVANT_IMMUNITE = 1
18    IMMUNITE_MECANISME = true
19    RESISTANCE_MECANISME = false
```

L'exemple de fichier de configuration produit une simulation au mélange parfait avec une diversité de 8. La configuration est similaire à celle d'une simulation SIR, la seule différence ici est l'ajout d'un paramètre de diversité non égal à 0.



Toutes les simulations qui suivent se basent sur ce fichier de configuration mais avec quelques modifications.

Premièrement, le niveau de diversité varie d'une simulation à une autre. Les valeurs de diversité choisies sont : 4, 8, 16, 32. Pour rappel, une diversité de  $x$  signifie que  $x$  bits des génomes des individus seront complétés et ceci aléatoirement. Sans diversité les génomes des individus sont tous identiques.

Deuxièmement, le mode de mouvement ainsi que le nombre de mouvements varie. Seule la première simulation servant d'exemple utilise le mode de mouvement au mélange parfait, toutes les autres ont un nombre de mouvements défini. Les mouvements étudiés sont : 1, 10, 50.

Finalement nous faisons varier le paramètre de charge virale (ici défini à 1). L'idée principale est ralentir la propagation de pandémies en diminuant la contagion des agents pathogènes. Les simulations sont effectuées sur 4 niveaux de charge virale : 0.25, 0.5, 0.75, 1

Les résultats du chapitre se découpent en 5 parties. La première sert d'exemple et permet d'expliquer les mécanismes derrière la diversité. Les 4 autres parties explorent les 4 niveaux de mouvements, les 4 niveaux de diversité et les 4 niveaux de charge virale.

## 10.2 Résultats

Chaque figure est structurée de la même manière. Il y a 3 subplots pour les 3 compartiments du modèle SIR. Les quatre courbes par compartiment représentent le nombre d'infectés de simulations aux niveaux de diversité différents.

### 10.2.1 Mélange parfait

Le document ne contient qu'une seule simulation à diversité non nulle et au mélange parfait. La raison pour laquelle cette voie n'est que peu explorée est parce que le mélange parfait est trop efficace pour la propagation de pandémies. Par conséquent les propagations sont beaucoup trop rapides comparé au ralentissement qu'offre la diversité.

Les résultats des simulations au mélange parfait sont très similaires aux simulations SIR et ceci est dû au fait que l'impact de la diversité est trop faible sur cette simulation. Pour pouvoir étudier l'impact de la diversité il est nécessaire de construire des systèmes plus progressifs dans la propagation des agents pathogènes.

Deux choses sont à noter à propos de cette figure. La première est que la propagation est beaucoup trop rapide comme nous pouvons le voir sur le compartiment *Susceptible*. Tous les individus sont déjà passés dans le compartiment *Infected* très tôt dans la simulation. Par conséquent dans ces configurations la propagation de la pandémie est

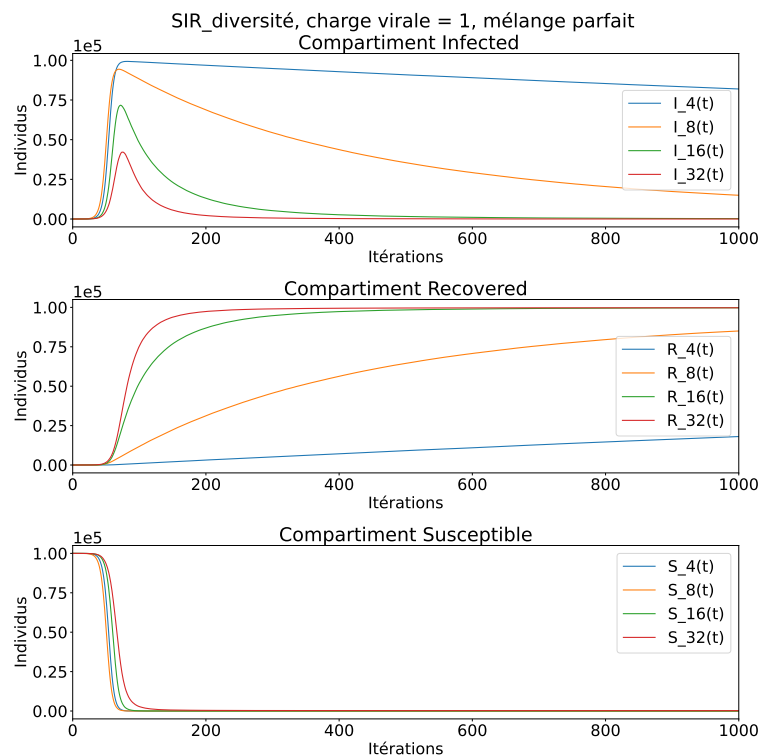


FIGURE 10.1 – Comparaison 50 mouvements

maximale car elle atteint rapidement 100% de la population.

Le deuxième élément se produit dans une deuxième phase et décrit l'immunisation des individus. Comme nous l'avons déjà dit, tout les individus finissent contaminés très vite. A partir de là, les individus s'immunisent mais leur immunisation dépend de la compatibilité qu'ils ont avec leur agent pathogène. Le compartiment *Recovered* montre la différence de vitesse des acquisitions d'immunisation parmi la population et il est assez clair qu'un plus grand niveau de diversité implique une immunisation générale plus rapide. Par conséquent la diversité permet une meilleure immunisation de la population.

### 10.2.2 Mouvements variable

Les simulations qui suivent pour étudier la diversité des individus ont toutes été effectuée avec des mouvements variable. Les mouvements choisis sont : 1, 10, 50.

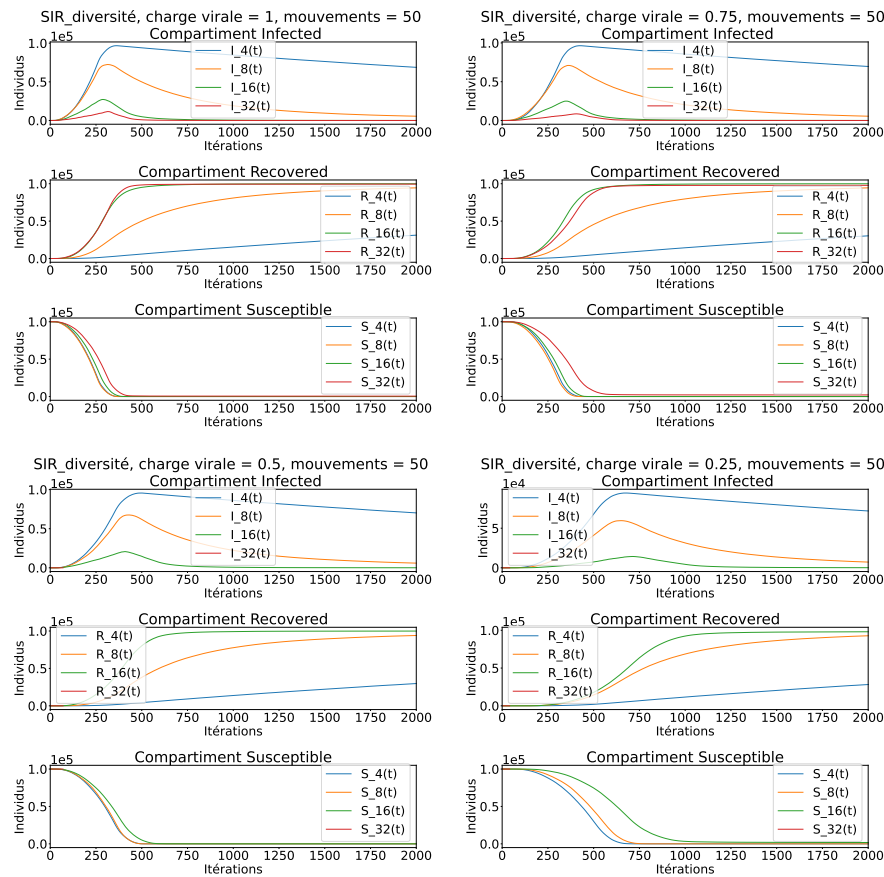


FIGURE 10.2 – 50 mouvements

La première série de graphiques est une ensemble de simulations à 50 mouvements. Les comportements sont très semblables à la simulation de référence au mélange parfait. Très rapidement, tous les individus sortent du compartiment *Susceptible* et donc la pandémie atteint une propagation de 100%. La seule action de la diversité est de permettre une immunisation plus rapide.

Les 4 figures utilisent chacune une valeur de charge virale différente. Les valeurs choisies sont : 1, 0.75, 0.5, 0.25. Le facteur de charge virale est directement lié avec la propagation de l'épidémie. Sur les figures nous constatons un ralentissement de la propagation lorsque la charge virale baisse. Ce comportement est normal car le facteur réduit les contacts entre les voisins.

Pour conclure, ces plots ne montrent que deux types de pandémies. Le premier type est une pandémie totale qui touche toute la population et le second type est une pandémie qui ne démarre pas. Les pandémies qui ne se sont pas propagées ont leur courbe coupée

très tôt dans la simulation comme nous pouvons le voir dans les simulations aux charge virale de 0.5 et 0.25 pour les cas à diversité 32.

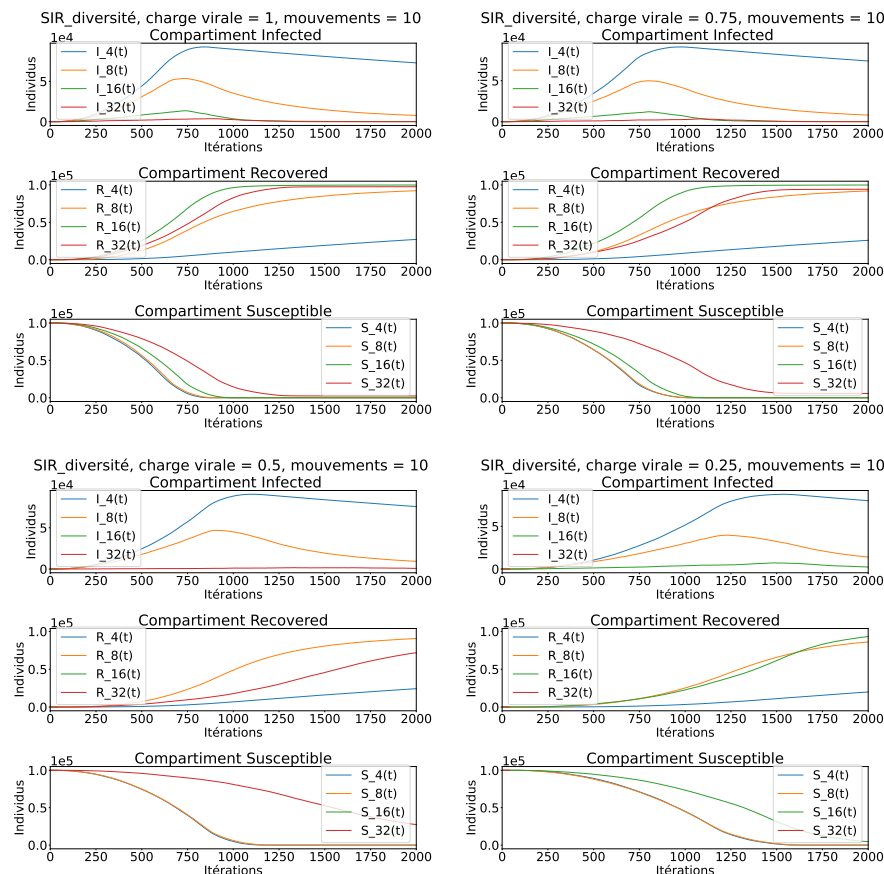


FIGURE 10.3 – Comparaison 10 mouvements

Etant donné que les simulations aux 50 mouvements sont trop rapides, nous baissions les mouvements des individus et les passons à 10. L'objectif est de pouvoir observer des épidémies qui se propagent en partie sans que ce soit du tout ou rien.

Les résultats ne montrent rien de nouveau. La diversité impacte la vitesse d'immunisation mais la propagation de la pandémie n'en est pas affectée. Certes les individus restent plus longtemps dans le compartiment *Susceptible* mais finissent tous par être contaminés donc nous avons encore à faire à des pandémies aux propagations de 100%.

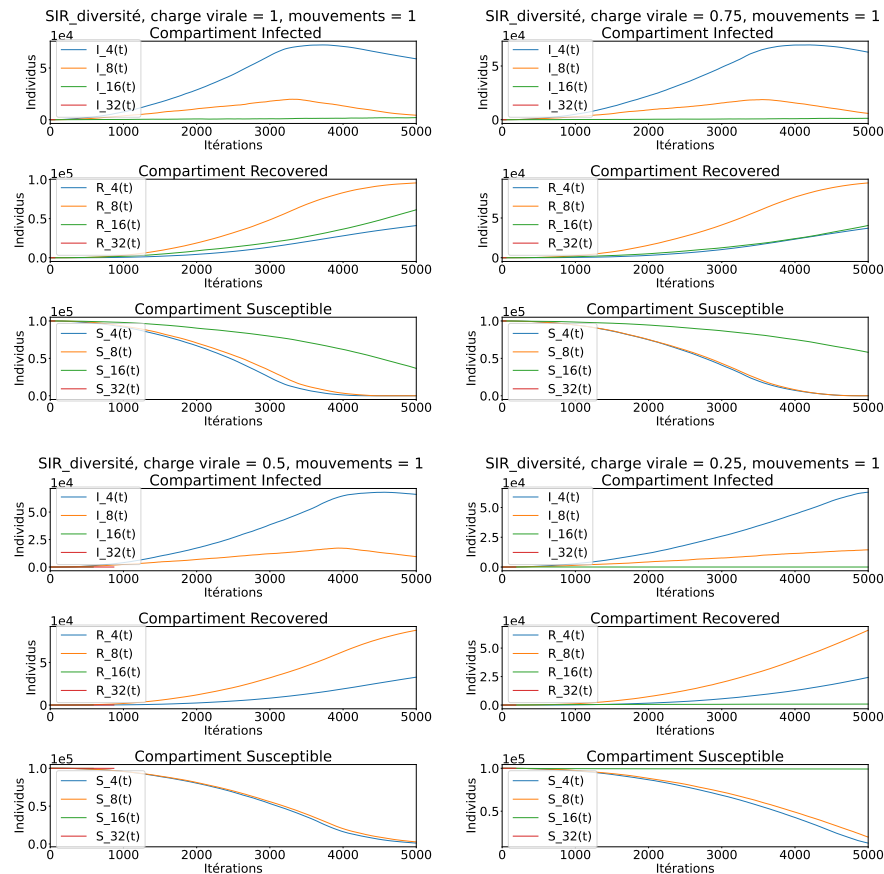


FIGURE 10.4 – Comparaison 1 mouvements

Finalement la dernière analyse de diversité effectue les mêmes mesures sur des systèmes à 1 mouvement. Notez que l'échelle ici est passée à 5000 sur les abscisses. Ces simulations ont un rythme d'évolution beaucoup plus lent mais suivent les mêmes tendances. Pour toutes ces simulations les pandémies se propagent ou non, totalement ou pas du tout.

### 10.2.3 Conclusion des résultats

Les résultats permettent d'observer l'impacte de la diversité sur les systèmes. Apparemment il n'y a pas d'impacte sur la propagation de la pandémie mais uniquement sur la vitesse d'immunisation des individus. Nous cherchions à observer des pandémies qui évoluaient jusqu'à un certain point et disparaissaient ensuite mais ce comportement n'est pas présent dans les mesures.

L'absence de ces événements est dû au fonctionnement du modèle et à la notion de diversité. Le problème est que dans le modèle, la diversité n'a aucun impacte sur la propagation de pandémie. Pour rappel, lorsqu'un individu sain est en contact avec un agent pathogène, l'individu devient contaminé et contagieux directement et ceci peu importe la diversité du système. Le seul effet de la diversité est le fait que cet individu peut s'immuniser plus ou moins rapidement en fonction de cette diversité. Mais le

problème est que la propagation se présente sous forme de vague, comme une onde. L'illustration ci-dessous illustre le phénomène.

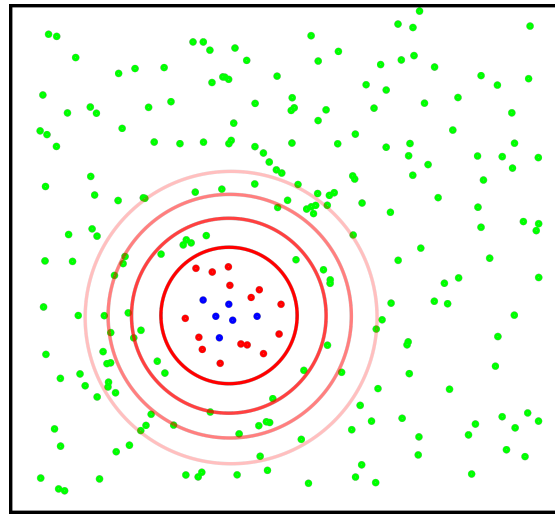


FIGURE 10.5 – vague propagation

La figure montre la propagation d'une pandémie. En vert nous avons les individus sains, en bleu les individus immunisés et en rouge les individus contaminés. Les cercles représentent la propagation de la pandémie sous forme de vagues. Nous pourrions découper le processus en 3 zones distinctes et indépendantes. La première est la zone centrale du cercle avec les individus immunisés. La seconde est la frontière entre les individus infectés et les sains donc la vague (cercle rouge) et finalement la troisième zone est le reste du système avec uniquement des individus sains.

D'après les résultats des simulations, la diversité n'impacte pas la propagation de la pandémie. En effet, la diversité permet l'immunisation plus rapide de la population donc l'apparition d'avantages de cercles bleus sur l'image. Le problème est que cette première zone est indépendante de la deuxième qui est la vague car cette dernière est beaucoup plus rapide que l'immunisation. Un individu fraîchement contaminé propagera la pandémie bien avant qu'il ne s'immunise. Par conséquent, l'ampleur des immunisations n'impacte pas la propagation de la pandémie.

Donc la première zone n'affecte pas la vague mais qu'en est-il de la troisième zone ? En réalité la troisième zone est purement uniforme. Si une vague parvient à un moment donné à se propager dans la troisième zone alors elle le pourra jusqu'à contaminer tous les individus.

La réflexion mène donc à expliquer que une pandémie se propage soit complètement soit pas du tout.

En épidémiologie, le nombre de reproduction de base  $R_0$  représente le nombre moyen de contaminations par un individu contaminé. Si ce taux est au dessous de 1, cela signifie que la pandémie est mourante par contre si le taux est au dessus de 1, la pandémie se propage.

Dans le modèle implémenté et d'après les mesures de diversité, une simulations ne peut pas avoir un  $R_0$  qui varie. Cette valeur est fixe du début à la fin. Par conséquent, toutes les simulations au  $R_0 < 1$  ne montrent aucune pandémie et au contraire toutes les simulations au  $R_0 > 1$  finissent par contaminer tout le monde. Dans une situation réelle, des mesures sanitaires permettent de faire varier ce taux au cours de la propagation de la pandémie. Dans le modèle ce fonctionnement est non présent. Une autre chose n'est pas présente et donc impossible dans le modèle est l'effet d'immunité collective.

L'immunité collective est le fait le niveau d'immunité général est suffisamment élevé afin de faire passer le taux  $R_0$  en dessous de 1 et que la pandémie disparaisse. Ce phénomène peut s'obtenir par trois façons. La première est la vaccination, la seconde est l'immunité croisée et la troisième est une forme de résistance naturelle. Si nous reprenions la figure ci-dessus, l'immunité collective dans le modèle pourraient être obtenue d'une seule façon. Il faudrait pouvoir modifier l'état de la troisième zone (les parties du système non contaminés). Si au fil du temps nous pouvions vacciner des individus ou les immuniser en amont de la pandémie alors nous pourrions influencer le  $R_0$  de la simulation mais le modèle n'inclut pas ces mécanismes. Par conséquent le  $R_0$  reste constant dû à l'homogénéité de la troisième zone.



## **11 Conclusion**