

# Laboratoire 10

## Pandémie!

### 10.1 Objectifs

1. Simuler l'évolution d'un système complexe.
2. Comprendre comment extraire des résultats généraux de simulations dont chaque réalisation individuelle peut produire un résultat fort différent.

### 10.2 Rapport de Lab

À remettre dans une semaine, avant le début du projet final. Le rapport doit inclure les réponses aux questions posées aux sections 10.4 et 10.5, codes tableaux et graphiques à l'appui, ainsi qu'une discussion critique (mais brève) de vos résultats.

### 10.3 Une simulation épidémiologique simple

L'idée ici est de combiner la marche aléatoire sur réseau (§7.4) et le modèle Feux-de-Forêt (§9.4). Un nombre  $M$  de marcheurs se déplacent sur un réseau 2D de dimensions  $N \times N$ . La seule différence avec la marche sur réseau décrite à la §8.4 est qu'on permet maintenant à deux marcheurs d'occuper le même site, ce sera essentiel dans ce qui suit. Supposons maintenant qu'un marcheur, choisi aléatoirement parmi les  $M$ , commet l'erreur de siéger pendant quelques minutes de trop sur d'un bol de toilette contaminé, et devient infecté par une horrible maladie, genre peste noire. Il n'en meurt pas immédiatement cependant; il survit pendant  $L$  itérations, et transmet sa maladie à tout autre marcheur avec qui il vient à occuper le même site sur le réseau. Ces marcheurs nouvellement infectés pourront eux aussi en infecter d'autres durant les  $L$  itérations suivant leur infection, et ainsi de suite. Ceci peut donc, sous certaines conditions que vous éluciderez plus loin, conduire à une avalanche d'infection se propageant à travers le réseau. Après  $L$  itérations suivant leur infection, les marcheurs malades meurent et cessent immédiatement d'être contagieux.

Le code C de la Figure 10.1 ci-dessous implémente cet "algorithme" de propagation de la maladie. Il sera important de bien comprendre comment fonctionne ce code, car vous aurez à le modifier dans ce qui suit. Notez bien les choses suivantes:

1. La simulation est globalement structurée selon une boucle conditionnelle (**while**) qui tournera jusqu'à ce que le nombre de marcheurs infectés retombe à zéro, ou qu'un maximum prédéfini (**NITERMAX**) ait été atteint. À la sortie de cette boucle temporelle, la valeur de la variable **iter** vous donne donc la durée de l'épidémie.

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define N 128          /* taille du reseau */
#define M 4000        /* nombre de marcheurs */
#define L 20          /* duree de survie des malades */
#define NITERMAX 5000 /* nombre maximal d'iterations temporelles */
int main(void)
{
/* Declarations ===== */
int x[M], y[M], infect[M], survie[M] ;
int ninfect, nmort, iter, j, k, jj, xnew, ynew, d2 ;
/* Executable ===== */
for (j=0 ; j<M ; j++) {          /* M marcheurs sur le reseau */
x[j] =floor(1.*N*rand()/RAND_MAX) ;
y[j] =floor(1.*N*rand()/RAND_MAX) ;
infect[j]=0 ; survie[j]=0 ;
}
jj =1.*M*rand()/RAND_MAX ;      /* infection d'un marcheur */
infect[jj]=1 ; survie[jj]=L ; ninfect=1 ; nmort=0 ; iter=0 ;
while ( ninfect > 0 && iter < NITERMAX ) { /* Iteration temporelle */
for (j=0 ; j<M ; j++) {        /* boucle sur les marcheurs */
if ( infect[j] == 1 ) { survie[j] -=1 ; } /* le temps qui passe... */
if ( infect[j] == 1 && survie[j] <= 0 ) { /* ...les malades qui meurent */
infect[j]=2 ; ninfect -=1 ; nmort +=1 ;
}
if ( infect[j] < 2 ) {          /* seuls les vivants bougent */
if ( 1.*rand()/RAND_MAX < 0.5 ) { /* on bouge en x */
xnew= x[j] + 2* (floor( 2.*rand()/RAND_MAX ))-1 ;
if ( xnew < 0 ) { xnew=0 ; } /* On ne quitte pas le reseau */
if ( xnew > N-1 ) { xnew=N-1 ; }
x[j]=xnew ;                    /* deplacement du marcheur */
} else {                        /* sinon on bouge en y */
ynew= y[j] + 2* (floor( 2.*rand()/RAND_MAX ))-1 ;
if ( ynew < 0 ) { ynew=0 ; } /* on ne quitte pas le reseau */
if ( ynew > N-1 ) { ynew=N-1 ; }
y[j]=ynew ;                    /* deplacement du marcheur */
}
}
if ( infect[j] == 1 ) {          /* ce marcheur est malade... */
for ( k=0 ; k<M ; k++ ) {      /* ...et peut contaminer les autres */
if ( k != j && infect[k] == 0 ) { /* ...qui ne le sont pas deja */
d2=pow( (x[j]-x[k]),2 )+pow( (y[j]-y[k]),2 ) ;
if ( d2 == 0 ) {                /* sur le meme site: contagion */
infect[k]=1 ; ninfect +=1 ; survie[k]=L ;
}
}
}
}
}
}
printf ("iter, malades, morts: %d, %d, %d\n",iter,ninfect,nmort) ; iter+=1 ;
}
}

```

Figure 10.1: Code C de base pour le modèle épidémiologique décrit dans le texte.

2. À chaque itération temporelle, une boucle inconditionnelle `for` sur les  $M$  marcheurs déplace chaque marcheur (encore vivant) et mets à jour son statut médical (tableau `infect[M]` décrit plus bas).
3. Quatre tableaux unidimensionnels, chacun de longueur  $M$ , contiennent l'information relative aux marcheurs: `x[M]`, `y[M]`, `infect[M]`, et `survie[M]`. La position  $(x, y)$  sur le réseau du  $j$ -ième des  $M$  marcheur est entreposée dans les éléments `x[j]` et `y[j]`; et un statut épidémiologique correspondant à l'élément `infect[j]`, où une valeur de 0 indique un marcheur en santé, 1 indique un marcheur contaminé (et contagieux!), et 2 un marcheur décédé (et immobile!).
4. L'initialisation consiste à distribuer aléatoirement les marcheurs sur le réseau, et une fois ceci fait en choisir un seul aléatoirement et lui assigner le statut "malade" (`infect[j]=1`).
5. Seuls les marcheurs vivants (sains ou malades, `infect[j]<2`) se déplacent sur le réseau. La marche aléatoire sur réseau 2D se fait de la manière habituelle, mais ici on a ajouté des tests (les 4 `if`) pour s'assurer que les marcheurs situés aux bords du réseau ne peuvent pas faire un pas en  $x$  ou  $y$  qui les ferait sortir du réseau.
6. Une fois le marcheur  $j$  infecté, l'élément `survie[j]` est initialisé à  $L$ , et cette valeur décroît de un à chaque itération temporelle subséquente (le premier `if` dans la boucle principale sur les  $M$  marcheurs); quand `survie[j]` atteint zéro, le marcheur  $j$  est déclaré mort (`infect[j]=2`; second `if` dans la boucle sur les marcheurs).
7. Pour chaque marcheur infecté (dernier `if` dans la boucle temporelle) on boucle sur les autres marcheurs, on calcule leur distance (`d2`), et tout autre marcheur sain (`infect[k]==0`) se trouvant sur le même site (`d2==0`) tombe malade.
8. Les variables-compteurs `ninfect` et `nmort` comptabilisent, à chaque itération temporelle, le nombre de marcheurs infectés et le nombre de morts sur le réseau.

Les Figures 10.2 et 10.3 illustrent différents aspects d'une simulation typique, ici sur réseau de taille  $128 \times 128$  occupé initialement par  $M = 4000$  marcheurs, et une durée de contagion  $L = 20$  itérations. Comme la simulation débute avec un seul marcheur contaminé introduit sur le réseau, la contamination est plutôt lente au début (voir Figure 10.2). La maladie se propage sous la forme d'un "front", d'épaisseur finie et initialement quasi-circulaire, qui croît graduellement en rayon, laissant derrière lui une densité réduite de marcheur... et des cadaaaavres. Comme dans le cas du modèle Feu-de-forêt, ce front développe une forme plus complexe, et se fragmente éventuellement en divers foyers d'infection spatialement distincts (voir Fig. 10.3). L'épidémie comme telle se développe en "vagues" successives, mais comme elle "brûle" graduellement son "carburant" elle en vient finalement à s'éteindre, ici après un peu moins de 700 itérations. Dans le cas de cette simulation, près de 60% de la population initiale a été éliminée, ce qui serait tout à fait réaliste dans le contexte des épidémies de peste noire ayant frappé l'Europe au Moyen Âge.

## 10.4 Comprendre le comportement du modèle

Avant de vous lancer dans l'étude des effets de la vaccination, il vous sera impératif de développer une bonne compréhension du comportement du modèle.

### 10.4.1 Vérification et validation

En guise de test de validation, votre première tâche consistera à reproduire les résultats des Figure 10.2 et 10.3. Évidemment, vu les multiples éléments stochastiques du modèle, vous ne pouvez pas vous attendre à retrouver *exactement* les mêmes résultats, cependant vous devriez pouvoir obtenir des résultats statistiquement comparables.

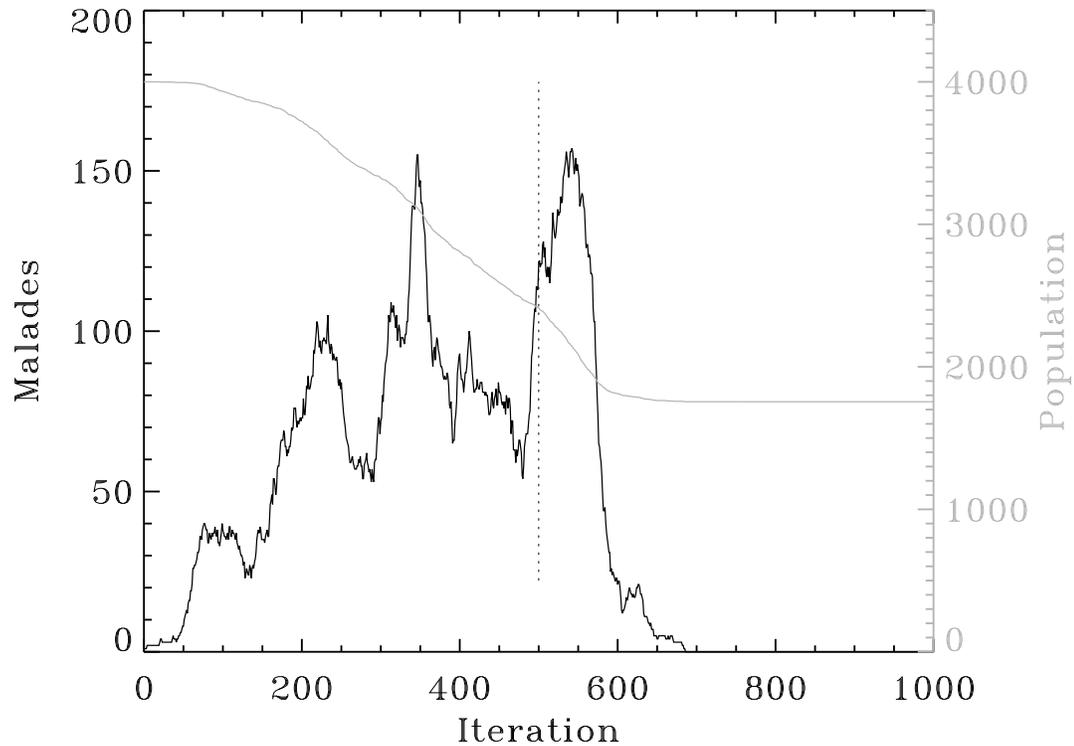


Figure 10.2: Évolution temporelle du nombre de marcheurs sains (trait gris) et malades (trait noir) pour une simulation sur réseau  $128 \times 128$  avec  $M = 4000$  et  $L = 20$ . Ici, la population initiale a été réduite par près de 60%. Notez les multiples “vagues” dans l’évolution de l’épidémie, une caractéristique souvent observée dans la réalité.

L’idée est donc de répéter la même simulation ( $N = 128$ ,  $M = 4000$ ,  $L = 20$ ) 10 fois, chaque fois avec une initialisation différente du générateur de nombre aléatoire (via la fonction `Csrand`, vous vous rappelez...?). Étudiez attentivement vos résultats, et répondez aux questions suivantes:

1. La durée de l’épidémie et le nombre de décès varient-ils beaucoup d’une simulation à l’autre?
2. Le résultat de la Fig. 10.2 est-il en fait représentatif du comportement “moyen” de la simulation pour ces valeurs des paramètres?
3. Les diverses vagues épidémiques visible à la Figure 10.2 semblent avoir une période assez bien définie; pouvez vous imaginer ce qui établit la durée de cette période?

#### 10.4.2 Effet de la densité de population

Pas besoin de réfléchir très longtemps pour réaliser qu’un facteur important dans la propagation de l’épidémie est la densité de population ( $\rho$ ), soit le nombre  $M$  de marcheurs se promenant sur le réseau divisé par le nombre de sites occupables ( $N \times N$ ):

$$\rho = \frac{M}{N^2} . \quad (10.1)$$

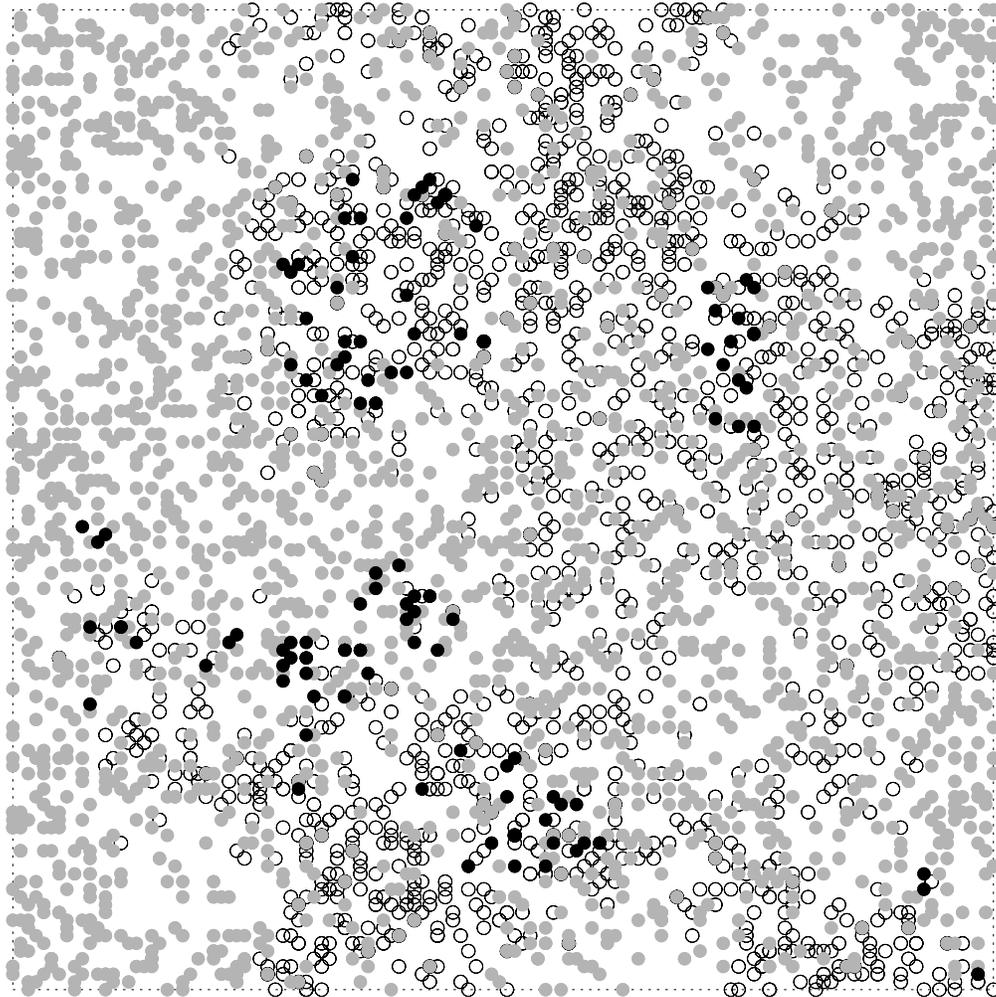


Figure 10.3: Instantané de la distribution des marcheurs dans la phase ascendante de la dernière grande vague épidémique (itération 500, trait vertical pointillé sur la Fig. 10.2). Les marcheurs non-contaminés sont en gris, les marcheurs contaminés en noir, et les cadavres (immobiles) en blanc. Notez la présence de multiples “foyers d’infection” d’étendues spatiales finies, à l’intérieur desquels marcheurs malades et sains cohabitent.

Il s'agit donc de calculer une séquence de simulations avec  $\rho = M/N^2$  croissant, de 0.05 à 0.5 en sauts de 0.05. Encore une fois, assurez vous d'évaluer la robustesse de vos résultats en répétant chaque simulation plusieurs fois (genre, 10) avec une initialisation différente du générateur de nombres aléatoires. Définissez le taux de mortalité  $0 \leq \mu \leq 1$  comme le nombre de décès divisé par la population initiale. Ensuite,

1. Pour chaque groupe de valeurs de  $\rho$  calculez le taux moyen de mortalité ( $\langle \mu \rangle$ ) et la déviation quadratique moyenne ( $\sigma$ ) par rapport à ce taux moyen; pour  $K$  réalisations de la simulation:

$$\langle \mu \rangle = \frac{1}{K} \sum_{k=1}^K \mu_k, \quad \sigma = \left( \frac{1}{K} \sum_{k=1}^K (\mu_k - \langle \mu \rangle)^2 \right)^{1/2}.$$

2. Portez en graphique  $\langle \mu \rangle$  versus  $\rho$ , et utilisez la valeur correspondante de  $\sigma$  comme barre d'erreur, i.e., ajoutez à chaque point  $(\rho, \langle \mu \rangle)$  de votre graphique un trait vertical couvrant l'intervalle  $\langle \mu \rangle \pm \sigma$ .
3. Produisez un graphique semblable, cette fois pour la durée moyenne ( $\langle T \rangle$ ) des épidémies en fonction de  $\rho$ , avec encore une fois une barre d'erreur calculée à partir de la déviation quadratique moyenne.
4. Portez en graphique  $\mu_k$  versus  $T_k$ , en utilisant un symbole de couleur différente pour chaque groupe de  $K$  simulations associées à une valeur de  $\rho$ . Quelle information pouvez-vous extraire de la distribution des points ainsi produite?

Peut-on dire qu'il existe ici une densité critique au dessus de laquelle l'infection balaye tout le réseau (ou presque) ? Considérant vos résultats, est-il possible de prédire précisément l'évolution d'une épidémie?

## 10.5 Modéliser la vaccination contre la grippe A (H1N1)

La peste noire, c'était au Moyen Âge. Revenons en décembre 2010. Suite à la gestion désastreusement soviétique de la pandémie de grippe A (H1N1) de la saison 2009-2010, le *Kominkom* à Québec a décidé de PRENDRE DES MESURES! Un budget est alloué à la recherche sur la propagation des épidémies. Une étude statistique préliminaire mandatée en mai 2010 par Notre Bon Gouvernement a compilé les informations suivantes, suite à la grippe 2009-2010:

1. Probabilité de transmission de la grippe par contact personne-à-personne: 0.66 (à chaque contact).
2. Durée de la phase contagieuse:  $L = 30$  itérations
3. Citoyen atteint par la grippe: coût net 1 "unité" ( $U$ ) (jours de travail perdus, consultations médicales, etc.);
4. 5% des citoyens atteints par la grippe développent des complications conduisant à une hospitalisation: coût moyen  $50 U$  par personne concernée.
5. Coût total moyen pour la vaccination:  $0.75 U$  par citoyen (inclut coûts des vaccins, coûts associés à l'organisation de la vaccination de masse, temps double pour le personnel, heures de travail perdues par les citoyens faisant la queue, augmentation de salaire pour les ministres, campagnes publicitaires, consultants externes, préparation des communiqués de presse, enveloppes brunes, etc.)

Hésitant à augmenter les impôts parce qu'il y a de l'élection dans l'air, Notre Bon Gouvernement aimerait bien minimiser les coûts totaux (i.e.,  $1U$  par malade +  $50U$  par malade développant des complications +  $0.75U$  par citoyen vacciné) la prochaine fois que la possibilité d'une épidémie se présentera. Votre tâche est donc la suivante: déterminer quelle fraction de la population devrait être vaccinée dans une région de forte densité de population (fraction d'occupation du réseau  $\rho = 0.5$ ), afin de minimiser les coûts totaux de l'opération. On supposera que les malades ne meurent pas de la grippe, donc la question de minimiser les décès n'est plus un enjeu.

Vous devez donc modifier votre code afin d'inclure les effets d'une probabilité finie de transmission de la maladie, et de la vaccination. Pour la probabilité de transmission, il s'agit simplement d'introduire un test probabiliste basé sur une probabilité de transmission  $p_t$  ( $< 1$ ) lorsque deux marcheurs (dont un contagieux) se retrouvent sur le même site du réseau. Pour la vaccination, le plus simple est d'assigner une variable (type `int`) à chaque marcheur, valant 1 si le marcheur est vacciné et 0 sinon. La transmission de la maladie à ce marcheur n'est possible que si cette variable vaut zéro. Vous pouvez supposer que le vaccin confère une immunité complète et absolue aux marcheurs vaccinés. Après qu'un marcheur  $j$  ait été infecté pendant  $L$  itérations, il guérit et devient immunisé; vous pouvez donc lui associer le même statut médical (immunité= 1) qu'un marcheur vacciné.

Présentez vos résultats sous la forme d'un graphique ou d'un tableau documentant le coût moyen d'une épidémie en fonction du pourcentage de vaccination, accompagné d'une discussion des incertitudes associées à vos résultats. Quel pourcentage-cible de vaccination recommandez-vous?

---

### Lectures supplémentaires:

La section 9.4 des notes de cours devraient être relue attentivement avant de commencer ce labo.

Aussi fortement recommandé sur ce sujet général: le magnifique petit bouquin *Rats, Lice and History*, par Hans Zinsser (1935), ainsi que le chapitre intitulé "Les Fugger de Cologne" dans *l'Oeuvre au Noir*, de Marguerite Yourcenar.

