

Projet : Epidémiologie

Xavier JUVIGNY

December 8, 2021

Contents

1	Présentation du projet	1
1.1	Vue d'ensemble	1
1.1.1	But	1
1.1.2	Structure	1
1.1.3	Echelles	2
1.2	Processus et déroulement	3
1.2.1	Préparation de l'épidémie de grippe	3
1.2.2	Transmission des pathogènes	3
1.2.3	Création et vieillissement des individus	3
1.2.4	Mise à jour des états infectieux	4
2	Objectifs du projet	4
2.1	Mesure du temps	4
2.2	Parallélisation affichage contre simulation	4
2.3	Parallélisation affichage asynchrone contre simulation	4
2.4	Parallélisation OpenMP	5
2.5	Parallélisation MPI de la simulation	5
2.5.1	Parallélisation finale	5
2.5.2	Bilan	5

1 Présentation du projet

1.1 Vue d'ensemble

1.1.1 But

L'objectif du projet est de simuler stochastiquement la co-circulation d'un virus (de la grippe par exemple) et d'un second agent pathogène, en interaction dans une population humaine virtuelle.

Pour le second agent, on aura le choix entre un agent de type épidémique (causant une épidémie annuelle mais absent le reste de l'année) ou un agent de type endémique (présent dans la population tout au long de l'année).

Le projet est librement inspiré d'une thèse qu'on peut télécharger sur le [lien suivant](#).

1.1.2 Structure

Variables d'état : Un ensemble de variables caractérise chacun des individus composant la population virtuelle décrite dans le modèle. Ces variables sont utilisées par les différents processus qui se succèdent au cours de chaque pas de temps. Elles sont regroupées au sein du tableau suivant :

Variable	Type	Actualisation
Âge	Entier (en jours)	À chaque pas de temps
<i>Concernant la grippe</i>		
Sain/Sensible	Etat (énuméré)	Au changement d'état
Contagieux	Etat (énuméré)	Au changement d'état
Immunisé	Etat (énuméré)	Au changement d'état
Temps incubation	Entier (en jours)	À chaque pas de temps
Durée incubation	Entier (Loi $\Gamma(k = 40, \theta = 0.05)$)	À chaque nouvelle infection
Temps passé symptomatique	Entier (en jours)	À chaque pas de temps
Durée symptomatique	Entier (Loi $\Gamma(k = 16; \theta = 0.25)$)	À chaque nouvelle infection
<i>Concernant le second pathogène</i>		
Sain/Sensible	Etat (énuméré)	Au changement d'état
Infecté	Etat (énuméré)	Au changement d'état
Contagieux	Etat (énuméré)	Au changement d'état
Immunisé	Etat (énuméré)	Au changement d'état
Temps passé asymptomatique	Nombre en jours	À chaque nouvelle contamination
Durée asymptomatique	jours (Loi $\Gamma(k = \frac{21^2}{25}; \theta = \frac{25}{21})$)	À chaque nouvelle contamination
Temps passé symptomatique	Jours	À chaque pas de temps
Durée symptomatique	Jours (Loi $\Gamma(k = \frac{144}{16}; \theta = \frac{16}{12})$)	À chaque nouvelle infection
Temps passé immunisé	nombre jours	À chaque pas de temps

Variables globales Ce sont les différentes variables régissant la simulation. Elles sont résumées dans le tableau suivant :

Variable	Valeur par défaut	Justification
Taille population	En séquentiel, 100 000 individus	Temps de calcul
Espérance vie	80 ans	–
Nombre moyen contact/jour	13	–
Distance max. déplacement	10 cellules	–
Paramètre acquisition	$\alpha_{BA} = 1$	À faire varier
Paramètre transmission	$\beta_A = 1$	À faire varier
<i>Concernant la grippe</i>		
Taux transmission	Loi normale $\beta_A = \mathcal{N}(\mu = 0.033, \sigma = \frac{\mu}{30})$	Moyenne calibrée
Date importation grippe	$\Gamma(k = \frac{71^2}{28}, \theta = \frac{28}{71})$	Idem
Probabilité cas rapporté	20%	Calibrée
Probabilité individu immunisé	23%	Calibrée
Période individu contagieux	[incubation + 1 jour; symptôme + 2 jours]	–
<i>Concernant le pathogène</i>		
Taux infection (asympt. → sympt.)	4.2×10^{-5} par jour	Calibré
Cas symptomatique rapporté	100%	Arbitraire
Période contagieux	[jour 0; symptômes + 2 jours]	–
Immunité après contamination	Faux	Arbitraire
Durée immunité (si vrai)	300 jours	Arbitraire
<i>Si pathogène endémique</i>		
Taux portage pathogène	20%	–
Paramère interaction	$\theta_{BA} = 1$	À faire varier
<i>Si pathogène épidémique</i>		
Début épidémie	30-60 jours après début grippe	À adapter selon pathogène
Nombre cas importés	20-30 jours	Idem
Proportion initiale individus immunisés	25%	Idem
Paramètre interaction immunité croisée	$\mu_{BA} = 1$	À faire varier

1.1.3 Echelles

Le pas de temps par défaut est le jour. Le premier jour dans la simulation correspond au 1^{er} Octobre. **À chaque pas de temps, on déplace tous les individus d'une à dix cases au hasard.** Le maillage dans lequel se promènent les individus est une grille cartésienne torique (en sortant de la grille par la gauche, on arrive à droite de la grille, etc.).

La dimension de la grille est calculée de sorte qu'il y ait environ 14 individus en moyenne par cellule ce qui permet d'obtenir en moyenne treize contacts par individus et par jour.

Deux individus sont rentrés en contact si ils se retrouvent dans la même cellule de la grille.

1.2 Processus et déroulement

À chaque pas de temps, les processus suivants sont exécutés dans l'ordre suivant :

1. Remise à zéro des compteurs d'incidence pour les deux pathogènes;
2. Préparation épidémie de grippe;
3. Transmission de la grippe;
4. Transmission du second pathogène;
5. Création de nouveaux individus (pour remplacer ceux qui sont morts au pas de temps précédent)
6. Vieillessement des individus;
7. Mise à jour des états des individus liés à la grippe;
8. Mise à jour des états des individus liés au second pathogène;
9. Mise à jour des compteurs de durée pour les états des deux pathogènes;
10. Déplacement des individus dans l'espace de simulation.

1.2.1 Préparation de l'épidémie de grippe

Ce processus est exécuté deux fois chaque année (une année dure 365 jours). Le premier jour (premier Octobre) de chaque nouvelle année de simulation, le jour d'apparition de la grippe saisonnière est tiré aléatoirement selon une loi Gamma (voir tableau), le taux de transmission de la grippe est tiré selon une loi normale (voir tableau) et une partie de la population (23%) est immunisée contre la grippe cette année là. Au jour d'importation de la grippe, un nombre aléatoire d'individu (entre 20 et 30) entre en phase d'incubation de la grippe.

1.2.2 Transmission des pathogènes

C'est au cours de cette étape que l'interaction entre les deux pathogènes entre en jeu. Supposons qu'il y ait un contact entre un transmetteur portant le virus de la grippe est un receveur non immunisé pour la grippe :

- Si le transmetteur n'est pas affecté par le second pathogène :
 - Si le receveur est sensible au second pathogène : chance d'attraper la grippe = β_A ;
 - Si le receveur est infecté par le second pathogène : chance d'attraper la grippe : $\beta_A \times \alpha_{BA}$
 - Si le receveur est en contact avec le transmetteur durant la période d'interaction des deux pathogènes, il a $\beta_A \times \mu_{BA}$ chance d'attraper la grippe;
 - Sinon il a simplement β_A chance d'attraper la grippe;
- Si le transmetteur est affecté par le second pathogène
 - Si le receveur est sensible au second pathogène : $\beta_A \times \theta_{BA}$ chance d'attraper la grippe
 - Si le receveur est infecté par le second pathogène : $\beta_A \times \alpha_{BA} \times \theta_{BA}$ chance d'attraper la grippe;
 - Si le receveur est immunisé et qu'il est entré en contact durant la période d'interaction des deux pathogènes : $\beta_A \times \theta_{BA} \times \mu_{BA}$ chance d'attraper la grippe;
 - Si le receveur est immunisé mais qu'il n'est pas rentré en contact durant la période d'interaction des deux pathogènes : $\beta_A \times \theta_{BA}$ chance d'attraper la grippe.

1.2.3 Création et vieillissement des individus

De nouveaux individus sont créés aléatoirement de manière à garder la population stable. Chaque individu vieillit d'un jour par pas de temps et meurt lorsque son nombre de jours correspondant à 80 ans.

1.2.4 Mise à jour des états infectieux

Les différents états infectieux de chacun des deux pathogènes ont des durées définies par l'utilisateur comme résumé dans le tableau plus haut. Les mises à jour des compteurs de durée et des changements d'état interviennent au cours de cette étape.

2 Objectifs du projet

Le but final du projet est de paralléliser le code fourni avec ce projet en respectant les charges suivantes :

- On utilisera conjointement MPI pour faire du parallélisme distribué et OpenMP pour la parallélisation en mémoire partagée;
- Le processus maître (le processus 0) ne s'occupera que de l'affichage graphique tandis que les autres processus effectueront la gestion des individus, de la grippe et des agents pathogènes en parallèle;
- L'affichage se fera de manière asynchrone, c'est à dire que la mise à jour de l'affichage ne se fera pas obligatoirement pour chaque jour passé;
- Le projet séquentiel utilise des séquences pseudo-aléatoires contrôlées par des graines, garantissant d'avoir le même résultat à chaque exécution du programme. Il faudra veiller en parallèle d'obtenir la même séquence qu'en séquentiel en contrôlant la valeur des graines initialisées dans le code.

Pour chaque étape du projet, on rendra **la version du code correspondant** avec les noms de fichier suivant (ou des branches de git différentes) :

- `simulation_sync_affiche_mpi.cpp`
- `simulation_async_affiche_mpi.cpp`
- `simulation_async_omp.cpp`
- `simulation_async_mpi.cpp`
- `simulation_async_mpi_omp.cpp`

Il est également important pour vous de conserver la version d'origine afin de contrôler chaque étape de votre parallélisation.

Voici les différentes étapes à mener pour le projet :

2.1 Mesure du temps

Dans un premier temps, mesurez le temps passé dans la simulation par pas de temps avec et hors affichage et le temps passé à l'affichage. Que constatez-vous ?

2.2 Parallélisation affichage contre simulation

On écrira une première version MPI du code fonctionnant seulement sur deux processus. Le premier processus s'occupera de l'affichage en synchrone et le second processus de la simulation proprement dite. Mesurez le temps passé par pas de temps pour la simulation et mesurez l'accélération. Par rapport à la question précédente, essayez d'analyser et d'expliquer les résultats obtenus.

2.3 Parallélisation affichage asynchrone contre simulation

Nous allons maintenant rendre asynchrone l'affichage, c'est à dire que le processus s'occupant de la simulation ne devra pas envoyer de données au processus zéro sauf si celui-ci est en attente d'un message pour mettre à jour son affichage.

Pour tester si le processus 0 est en attente d'un message, on utilisera la fonction `MPI_Iprobe` proposée par MPI (pour un exemple d'utilisation de cette fonction, voir [l'exemple donné sur ce lien](#)).

Mesurez le temps passé par pas de temps pour la simulation. Que constatez-vous par rapport au temps mesuré précédemment ? Cela va-t-il dans le même sens que l'analyse que vous aviez faite dans la partie précédente ?

2.4 Parallélisation OpenMP

Parallélisez la simulation faite par le processus 1 à l'aide d'OpenMP tout en veillant à obtenir les mêmes résultats pour la courbe de sortie que la simulation d'origine (attention à la valeur des graines aléatoires !).

Mesurez l'accélération obtenue pour un nombre de threads variables en :

- prenant un nombre d'individus global constant;
- prenant un nombre d'individus constant par thread (le résultat obtenu sera alors différent...)

2.5 Parallélisation MPI de la simulation

Reprenez la version sans OpenMP et parallélisez la partie simulation. On créera un groupe pour l'affichage (seulement le processus 0) et un groupe pour la simulation (les autres processus) à l'aide de la fonction `MPI_Comm_split` (voir documentation et exemple sur [ce lien](#)).

La parallélisation se fera sur les individus qu'on distribuera sur les processus s'occupant de la simulation. De même qu'avec OpenMP, assurez vous que vous obtenez exactement la même simulation, quel que soit le nombre de processus s'occupant de la simulation.

Mesurez l'accélération obtenue pour un nombre de procesus variables en :

- prenant un nombre d'individus global constant;
- prenant un nombre d'individus constant par processus (le résultat obtenu sera alors différent...)

2.5.1 Parallélisation finale

Parallélisez la partie simulation à l'aide d'OpenMP et MPI en vous servant des deux versions précédentes. Testez votre programme sur votre ordinateur, puis essayez de lancer les processus MPI sur divers ordinateurs reliés par un réseau (il faudra faire un fichier `machinefile.dat` contenant les noms des ordinateurs que vous voulez utiliser et lancer MPI avec l'option `-machinefile machinefile.dat`).

Mesurez les diverses accélérations obtenues avec un nombre d'individus constant globalement puis par processus et par thread.

2.5.2 Bilan

Dans le bilan, faites un retour d'expérience du projet :

- Sur les résultats obtenus;
- Sur la connaissance que vous avez acquise avec ce projet.