

TP #4

Responsable: Antoine Allard

À remettre au plus tard le **22 avril à 8h00****TP4.1 Intégration numérique de la dynamique épidémiologique SIS (50 points)**

Soit la dynamique épidémiologique SIS

$$\begin{aligned}\frac{ds}{dt} &= -\beta si + \alpha i \\ \frac{di}{dt} &= \beta si - \alpha i\end{aligned}$$

où $0 \leq s, i \leq 1$ sont les fractions de la population susceptible et infectée, respectivement, et où $\alpha, \beta > 0$ sont des paramètres de contrôle. Cette dynamique est utilisée pour modéliser la propagation d'agents infectieux face auxquels les individus ne développent pas d'immunité. La conservation de la population, $s + i = 1$, permet de réduire la dynamique SIS au système d'ordre 1

$$\frac{di}{dt} = \beta i(1 - i) - \alpha i.$$

De plus, il est possible d'*adimensionner* cette équation en divisant par α de part et d'autre et de définir le temps *sans dimension* $\tau = \alpha t$ (exprimé en termes de la période moyenne de guérison α^{-1})

$$\frac{di}{d\tau} = (R_0 - 1)i - R_0 i^2,$$

où nous avons défini le *nombre de reproduction de base* $R_0 = \beta/\alpha$ correspondant au nombre moyen attendu de cas directement générés par un cas dans une population où tous les individus sont sensibles à l'infection.

- a. (5 points) Identifiez les deux états stationnaires $0 \leq i_{1,2}^* \leq 1$ pour lesquels $\frac{di}{d\tau} = 0$, de même que les conditions sur R_0 pour lesquelles ces états stationnaires sont possibles (s'il y a lieu).
- b. (10 points) Obtenez une solution analytique $i(\tau)$ (valide pour $i \geq 0$). Vous noterez $i(0) = i_0$. Indice : Il s'agit d'une EDO de Bernoulli.
- c. (5 points) Implémentez les intégrateurs d'Euler, de Runge-Kutta d'ordre 2 et de Runge-Kutta d'ordre 4. Vos fonctions devront retourner les trajectoires composées des fractions de la population infectée $\{i_s\}_{s=0,\dots,T}$ et des temps $\{\tau_s\}_{s=0,\dots,T}$ auxquels ces fractions correspondent. La valeur de T sera fixée ci-dessous.
- d. (20 points) Implémentez la méthode du ratio doré (ou un autre optimiseur de votre choix) qui vous permettra de trouver la longueur de pas h permettant d'obtenir une erreur sur la trajectoire numérique $\varepsilon(h)$ bornée par $(1 \pm 0.01)\delta$, où δ sera spécifiée ci-dessous. L'erreur sera calculée selon

$$\varepsilon(h) = \sqrt{\frac{1}{T+1} \sum_{s=0}^T \left(i_s - i(\tau_s)\right)^2},$$

où $i(\cdot)$ est la solution analytique obtenue en **b**. N'hésitez pas à considérer une méthode heuristique pour bien cerner l'intervalle de départ pour h fourni à la méthode du ratio doré. À l'aide de votre algorithme, identifiez une valeur de pas d'intégration h permettant de limiter $\varepsilon(h)$ à l'intervalle $[0.99\delta, 1.01\delta]$ pour plusieurs valeurs de $\delta \in [10^{-9}, 10^{-6}]$, pour plusieurs couples (i_0, R_0) et pour chacun des trois intégrateurs implémentés en **c**. Utilisez $T = 3$. Une fois les valeurs de h obtenues, représentez quelques-unes des trajectoires sur un graphique (pour des fins de visualisation seulement).

e. (10 points) À l'aide des résultats obtenus en **d**, obtenez numériquement l'ordre d de l'erreur globale $\varepsilon(h) \propto h^d$ effectuée par chacun des trois intégrateurs implémentés en **c**. Il est possible d'obtenir d via une régression (linéaire) en observant que $\log \varepsilon(h) \propto d \log h$.

TP4.2 Simulation Monte-Carlo de la dynamique SIS (50 points)

Dans cette section, nous allons modéliser la propagation d'une maladie sur un réseau où les nœuds représentent les individus et les liens représentent un contact sans distanciation sociale entre les individus (voir figure 4.1). Pour se familiariser avec les réseaux, étudions un modèle de construction de graphes aléatoires¹, soit celui d'Erdős-Rényi. Il existe deux manières de configurer ce modèle, mais nous allons nous concentrer sur le modèle $G(n, p)$. Ici, $G(\cdot)$ est un ensemble de graphes, c'est-à-dire la collection de tous les graphes possibles issus des paramètres n et p qui représentent le nombre de nœuds dans le graphe et la probabilité de connecter une paire de nœuds.

Pour générer un graphe dans le modèle $G(n, p)$, on initialise un nombre de nœuds n et l'on itère sur toutes les paires distinctes² possibles en déterminant si le présent lien sera créé (succès) ou non (échec) sachant que la probabilité que cela se produise est p . Chaque lien est donc créé (ou refusé) de manière indépendante.

a. (5 points) Déterminez l'expression qui gouverne la probabilité de générer un graphe de l'ensemble $G(n, p)$ qui contiendra m liens. Détaillez votre raisonnement pour obtenir cette expression.

b. (10 points) Déterminez également le nombre de liens moyens que possède un graphe issu du modèle $G(n, p)$ ainsi que l'écart-type sur le nombre de liens. Détaillez votre raisonnement pour obtenir ces expressions. Si des propriétés sont utilisées, elles doivent être dûment citées et/ou démontrées également.

c. (15 points) Implémentez un algorithme qui génère des graphes en utilisant le modèle d'Erdős-Rényi avec les paramètres $n = 100$ et $p = 0.05$. Une fois programmé, obtenez empiriquement la moyenne, l'écart-type et la distribution du nombre de liens dans les réseaux issus de votre algorithme. Comparez ces résultats aux valeurs obtenues en **a** et **b**. (Indice : un graphe est parfois simplement représenté par sa matrice d'adjacence. Vous pouvez utiliser cette représentation ou une autre structure de votre choix qui permet d'encoder l'information pertinente.)

Le module NetworkX³ permet de générer et manipuler facilement les réseaux en utilisant Python. Pour la question **d**, nous vous demandons d'utiliser ce module. (Il pourrait être judicieux de vérifier vos résultats à la question **c** en utilisant ce module.)

Afin de simuler la propagation de la maladie sur un réseau, il faut d'abord sélectionner aléatoirement quelques nœuds et leur donner l'étiquette « infecté ». On divise alors notre horizon temporel en plusieurs étapes discrètes. À chaque étape, il suffit de vérifier si les nœuds infectés guérissent (avec probabilité α) et redeviennent susceptibles. Si ce n'est pas le cas, il faut déterminer si les voisins des nœuds infectés deviennent infectés (avec probabilité β). Lors d'une étape, la nouvelle infection d'un nœud ne doit pas engendrer de conséquences sur ses voisins avant la prochaine

1. Dans ce travail, on utilisera le mot graphe et réseau de manière interchangeable.

2. En combinatoire, on parle ici de combinaisons, c'est-à-dire que la paire est composée de deux nœuds distincts, mais l'ordre dans lequel les nœuds sont pigés n'a pas d'importance. Par exemple, la paire de nœuds i et j est la même que la paire j et i .

3. <https://networkx.org/>

étape. Par exemple, si le nœud k devient infecté au temps t , les voisins de k peuvent seulement tomber malades à cause de k au temps $t + 1$.

d. (20 points) Implémentez la simulation de la dynamique SIS sur réseau en utilisant le module NetworkX. Une fois votre algorithme fonctionnel, générez les courbes moyennes du nombre de nœuds infectés et susceptibles en fonction du temps pour le réseau fourni (*reseau.adj*). Pour ouvrir ce réseau, utilisez la fonction `read_adjlist()` de NetworkX⁴. Pour vos simulations, utilisez $\alpha = 0.05$, $\beta = 0.1$, une proportion initiale de nœuds infectés de 10% et 100 pas de temps. Comparez les résultats de la simulation avec la dynamique SIS étudiée dans la première partie du travail et discutez.

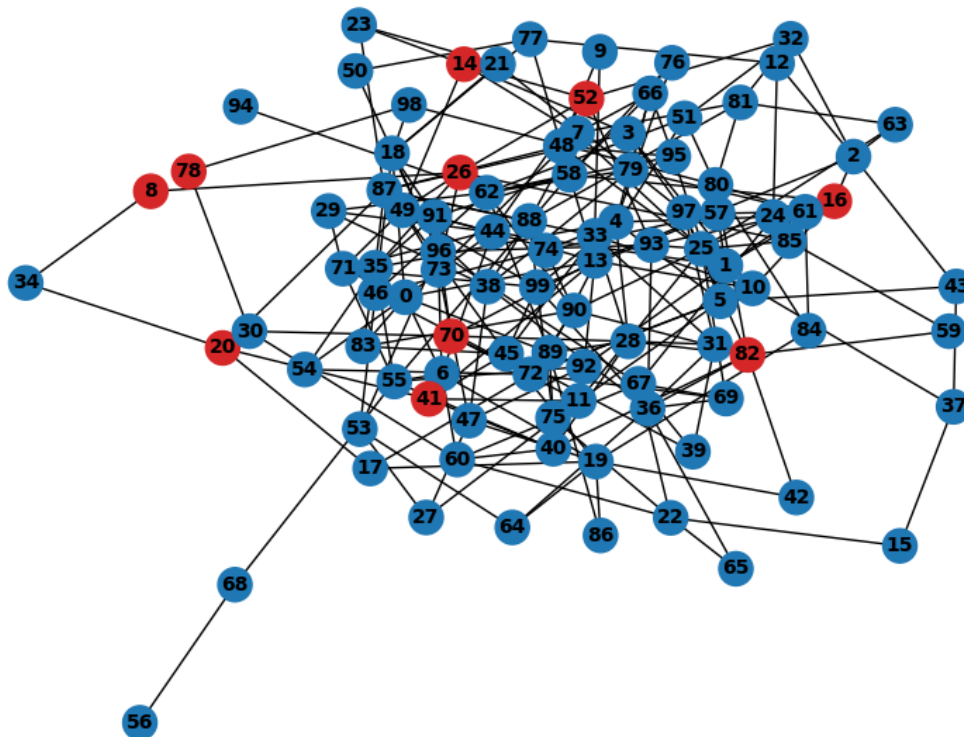


FIGURE 4.1 – Structure du réseau contenu dans le fichier *reseau.adj*. Les nœuds représentent des individus et les liens représentent un contact entre les individus. Les nœuds bleus sont dans l'état « susceptible » tandis que les nœuds rouges sont dans l'état « infecté ». Ce réseau est issu du modèle $G(n = 100, p = 0.05)$.

4. https://networkx.org/documentation/stable/reference/readwrite/generated/networkx.readwrite.adjlist.read_adjlist.html

Le travail devra être complété en équipe d'au plus trois personnes sous format de cahier de bord « jupyter » (*.ipynb) et remis dans la boîte de dépôt prévue à cette fin sur le [portail du cours](#). Ce document contiendra **toutes informations pertinentes** permettant au lecteur d'apprécier vos résultats et conclusions, incluant le code Python utilisé et d'éventuelles références bibliographiques. La qualité de la présentation est très importante (utilisation de sections, de graphiques appropriés, de mise en contexte, etc.).

Prenez soin de bien indiquer vos noms dans le cahier de bord. Pour faciliter la tâche de classification, utilisez la nomenclature suivante pour le fichier transmis (un seul): TP4_nom1_nom2_nom3.ipynb
