Projet d'analyse numérique MAM3 - Modélisation de la propagation d'une épidémie

Yassine Zouhri Ahmed Talbo S

9 April 2024

Sommaire

1 Introduction

Au cours des dernières décennies, la modélisation de la propagation d'agents infectieux au sein des populations a représenté un domaine crucial de la recherche en épidémiologie. Les avancées technologiques et la compréhension croissante des mécanismes de transmission des maladies ont permis le développement de modèles mathématiques sophistiqués visant à prédire et à contrôler la diffusion de ces agents pathogènes. Un modèle fondamental dans cette discipline est le modèle SIR, qui divise la population en trois compartiments : les sujets susceptibles d'être infectés (S), les personnes infectieuses (I), et les individus récupérés ou immunisés (R). Ce modèle a été largement utilisé pour étudier des maladies telles que la rougeole, les oreillons et la rubéole, où la résistance acquise confère une immunité durable. Cependant, afin de mieux comprendre et anticiper la dynamique des épidémies, il est impératif d'intégrer des variables pertinentes telles que l'efficacité des traitements. Dans cette optique, nous nous proposons d'enrichir le modèle SIR en tenant compte de la présence et de l'impact des traitements, afin d'évaluer plus précisément leur efficacité dans la gestion et la prévention des épidémies.

2 Objectifs

Dans le cadre de ce projet, notre équipe s'est fixée des objectifs essentiels. Premièrement, nous cherchons à analyser minutieusement comment un agent infectieux se répand au sein d'une communauté.

Cela nécessite de suivre l'évolution du nombre de personnes saines et infectées sur une période donnée, en tenant compte des interactions favorisant la contagion. Pour modéliser cette dynamique complexe, nous prévoyons d'utiliser des équations différentielles qui reflètent fidèlement les relations entre les individus et l'agent infectieux.

L'objectif principal est de trouver des solutions numériques à ces équations grâce à des méthodes numériques adéquates. Par la suite, nous comparerons nos résultats avec d'autres approches de modélisation et testerons notre code face à des programmes existants en Python.

Afin de mener à bien ces objectifs, nous avons mis au point un plan de travail précis et documenté nos progrès dans un journal de bord, disponible en annexe pour consultation.

3 Explications théoriques

3.1 Modélisation mathématique

Pour modéliser le traitement, nous considérons que les individus traités deviennent insensibles à la maladie. Une fraction α d'individus infectés est sélectionnée pour être traitée chaque unité de temps, et les individus traités acquièrent une immunité ou décèdent selon un taux η. Ainsi, la population est subdivisée en quatre compartiments : S, I, T et R, où T désigne les individus traités. Les naissances et les décès naturels ne sont pas pris en compte dans ce modèle.

$$\begin{cases} S' = -\dfrac{\beta}{N}(I + \delta T)S \\ I' = \dfrac{\beta}{N}S(I + \delta T) - (\alpha - \gamma)I \\ T' = \alpha I - \eta T \\ R' = \gamma I - \eta T \end{cases}$$

Afin de résoudre ce système différentiel nous allons utiliser 3 méthodes différente : **Euler implicite** , **Euler explicite** et **Kutta4** .

3.2 Euler implicite

La méthode d'Euler implicite est une méthode numérique utilisée pour résoudre des équations différentielles ordinaires . Elle discrétise le temps en pas de taille h et utilise une itération itérative pour estimer les valeurs des fonctions inconnues à chaque pas de temps. Dans le contexte de la modélisation de la dynamique d'une maladie infectieuse, nous appliquons cette méthode pour estimer l'évolution des populations de susceptibles (S), d'infectés (I), de porteurs sains (T), et de récupérés (R) dans une population donnée.

- *Initialisation*

  Nous commençons par définir les conditions initiales pour chaque variable du système : $S(0)$, $I(0)$, $T(0)$, et $R(0)$. Ces conditions représentent les valeurs des populations à l'instant initial.

- *Itérations*

  Pour chaque pas de temps t_n, nous utilisons les équations du système pour estimer les valeurs des populations à l'instant t_{n+1}. Dans la méthode d'Euler implicite, cette estimation implique la résolution d'un système d'équations non linéaires à l'aide de la méthode de Newton.

- *Calcul de la Jacobienne*

  Avant de résoudre le système d'équations non linéaires, nous calculons la jacobienne du système. La jacobienne est une matrice contenant les dérivées partielles de chaque équation par rapport à chaque variable du système. Pour notre système d'EDOs, la jacobienne est donnée par :

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial S} & \frac{\partial F_1}{\partial I} & \frac{\partial F_1}{\partial T} & \frac{\partial F_1}{\partial R} \\ \frac{\partial F_2}{\partial S} & \frac{\partial F_2}{\partial I} & \frac{\partial F_2}{\partial T} & \frac{\partial F_2}{\partial R} \\ \frac{\partial F_3}{\partial S} & \frac{\partial F_3}{\partial I} & \frac{\partial F_3}{\partial T} & \frac{\partial F_3}{\partial R} \\ \frac{\partial F_4}{\partial S} & \frac{\partial F_4}{\partial I} & \frac{\partial F_4}{\partial T} & \frac{\partial F_4}{\partial R} \end{bmatrix}$$

  où F_1, F_2, F_3, et F_4 sont les fonctions du système. Chaque élément de la jacobienne est calculé en prenant les dérivées partielles des fonctions F_i par rapport à chaque variable S, I, T, et R.

- *Utilisation de la Jacobienne dans l'algorithme de Newton*

  Dans l'algorithme de Newton, nous utilisons la jacobienne pour linéariser les équations non linéaires résultant de la discrétisation temporelle. La formule de l'itération de Newton devient :

$$x^{(k+1)} = x^{(k)} - [J(x^{(k)})]^{-1} F(x^{(k)})$$

  où $x^{(k)}$ est l'itération courante, $x^{(k+1)}$ est l'itération suivante, $F(x^{(k)})$ est le vecteur des fonctions du système évaluées à l'itération courante, et $J(x^{(k)})$ est la jacobienne évaluée à l'itération courante.

- *Arrêt*

  Nous répétons les itérations de Newton jusqu'à ce que la norme du résidu $F(x^{(k)})$ soit inférieure à une certaine tolérance ou jusqu'à ce qu'un nombre maximum d'itérations soit atteint.

3.3 Euler explicite

La méthode d'Euler explicite est une méthode numérique utilisée pour résoudre des équations différentielles ordinaires. Contrairement à la méthode d'Euler implicite, où les valeurs actuelles sont calculées en fonction des valeurs futures, la méthode explicite utilise les valeurs actuelles pour calculer les valeurs futures.

- *Initialisation*

  L'initialisation dans la méthode d'Euler explicite est similaire à celle d'Euler implicite.

- *Itérations*

  Pour chaque pas de temps t_n, nous utilisons les équations du système pour estimer les valeurs des populations à l'instant t_{n+1}. Dans la méthode d'Euler explicite, cette estimation implique l'utilisation d'une approximation de la dérivée pour avancer d'un pas de temps. Les équations de mise à jour pour chaque variable du système sont les suivantes :

$$\begin{aligned} S_{n+1} &= S_n + h \cdot \phi_S(S_n, I_n, T_n, \beta, \delta, N) \\ I_{n+1} &= I_n + h \cdot \phi_I(S_n, I_n, T_n, \alpha, \beta, \delta, \gamma, N) \\ T_{n+1} &= T_n + h \cdot \phi_T(I_n, T_n, \alpha, \eta) \\ R_{n+1} &= R_n + h \cdot \phi_R(I_n, T_n, \gamma, \eta) \end{aligned}$$

  où ϕ_S, ϕ_I, ϕ_T, et ϕ_R sont les fonctions de mise à jour pour chaque population, et h est la taille du pas de temps.

- *Arrêt*

  Nous répétons ces itérations jusqu'à atteindre le nombre maximal d'itérations $N_{\max}$ ou jusqu'à ce que le système converge vers une solution stable.

3.4 Kutta4

La méthode de Runge-Kutta d'ordre 4 est plus précise que la méthode d'Euler et utilise une approche itérative pour estimer les valeurs des fonctions inconnues à chaque pas de temps.Nous appliquons cette méthode pour estimer l'évolution des populations de susceptibles (S), d'infectés (I), de porteurs sains (T), et de récupérés (R) dans une population donnée.

- *Initialisation*

  L'initialisation dans la méthode de Runge-Kutta d'ordre 4 est similaire à celle des méthodes précédentes

- *Itérations*

  Pour chaque pas de temps t_n, nous utilisons la méthode de Kutta4 pour estimer les valeurs des populations à l'instant t_{n+1}. Cette méthode consiste en quatre étapes intermédiaires pour calculer la pente au point t_n et utiliser cette pente pour estimer les valeurs au point t_{n+1}. Les équations de mise à jour pour chaque variable du système sont calculées comme suit :

$$
\begin{aligned}
k_1 &= h \cdot \phi(S_n, I_n, T_n, R_n) \\
k_2 &= h \cdot \phi(S_n + \frac{k_1}{2}, I_n + \frac{k_1}{2}, T_n + \frac{k_1}{2}, R_n + \frac{k_1}{2}) \\
k_3 &= h \cdot \phi(S_n + \frac{k_2}{2}, I_n + \frac{k_2}{2}, T_n + \frac{k_2}{2}, R_n + \frac{k_2}{2}) \\
k_4 &= h \cdot \phi(S_n + k_3, I_n + k_3, T_n + k_3, R_n + k_3)
\end{aligned}
$$

  Les valeurs mises à jour des populations sont calculées comme suit :

$$
\begin{aligned}
S_{n+1} &= S_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \\
I_{n+1} &= I_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \\
T_{n+1} &= T_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \\
R_{n+1} &= R_n + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}
\end{aligned}
$$

- *Arrêt*

  L'arrêt est similaire à celui d'Euler explicite.

4 Réalisations pratiques

4.1 Objectif de l'algorithme

Notre algorithme vise à résoudre un système différentiel afin de représenter graphiquement l'évolution des quatre classes d'individus dans la propagation d'une maladie : les sujets susceptibles (S), les personnes infectieuses (I), celles qui ne peuvent plus contracter la maladie (R), et la classe des individus traités (T).

4.2 Réalisation de l'algorithme

Pour atteindre cet objectif, nous avons adopté une approche similaire à celle des travaux pratiques d'analyse numérique. Notre principal défi a été la mise en œuvre efficace de la méthode de Newton pour résoudre le système linéaire associé au modèle épidémiologique. En identifiant d'abord les équations différentielles du modèle, nous les avons discrétisées pour obtenir un système d'équations à résoudre numériquement. Ensuite, nous avons implémenté la méthode de Newton pour résoudre ce système, en assurant sa convergence et sa stabilité.

Cette approche nous a permis de développer un algorithme robuste pour simuler l'évolution de l'épidémie et visualiser graphiquement ses résultats.

5 Résultats

Dans cette section, nous présentons les résultats obtenus à partir de l'application des trois méthodes de résolution du système d'équations différentielles décrivant la propagation de l'épidémie.

Nos résultats confirment la cohérence de notre modélisation, mettant en évidence l'effet significatif de paramètres tels que le taux de contact sur la propagation de l'épidémie.

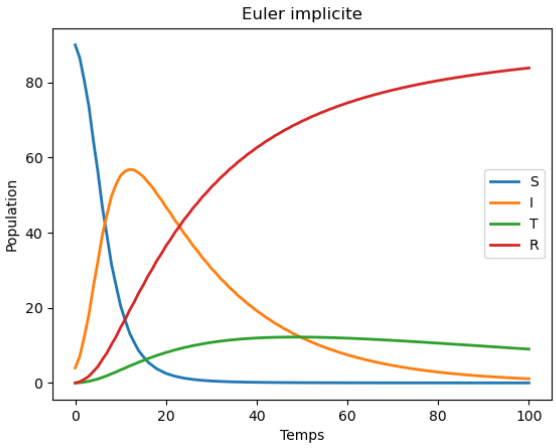

Figure 1: Évolution des populations avec la méthode d'Euler implicite

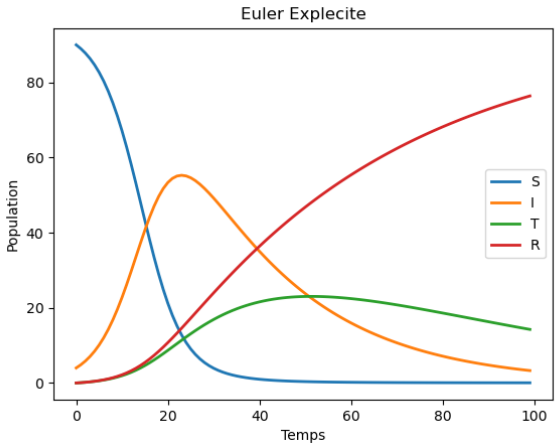

Figure 2: Évolution des populations avec la méthode d'Euler explicite

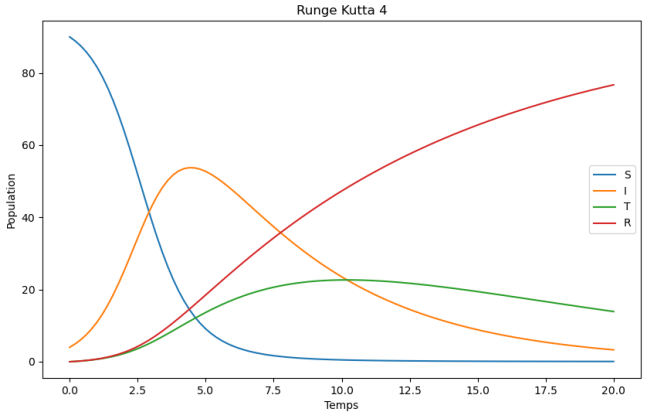

Figure 3: Évolution des populations avec la méthode de Runge-Kutta d'ordre 4

Par exemple, une diminution du taux de contact correspond à une croissance moins rapide des cas infectés, illustrant ainsi l'efficacité des mesures de confinement. Ces observations soulignent l'importance de comprendre et de manipuler les paramètres du modèle pour guider les décisions en santé publique

Conclusion

En conclusion, ce projet a représenté une expérience enrichissante qui nous a permis d'explorer l'application pratique des concepts théoriques à travers l'utilisation de Python. Nous avons constaté que les méthodes numériques telles que l'Euler explicite, l'Euler implicite et la méthode de Runge-Kutta d'ordre 4 se sont avérées efficaces pour résoudre numériquement des équations différentielles ordinaires , ainsi que l'application de la méthode de Newton pour la résolution de systèmes linéaires.

La compréhension approfondie du sujet a nécessité une maîtrise des techniques de résolution d'EDOs, essentielles pour modéliser divers systèmes physiques, biologiques, mécaniques, et plus encore. Nous avons pris conscience de l'importance cruciale de ces méthodes dans la modélisation et la simulation de phénomènes réels.

Ce projet nous a également permis de développer des compétences essentielles pour un ingénieur, notamment la capacité à travailler en équipe et à présenter nos résultats de manière orale et écrite de manière professionnelle. En envisageant une extension de ce projet, nous pourrions explorer des aspects plus complexes de la modélisation, tels que la prise en compte des variations démographiques réelles dans la population.

Dans l'ensemble, nous avons apprécié cette expérience qui nous a permis d'acquérir de nouvelles compétences en programmation Python, tout en appliquant nos connaissances mathématiques et informatiques à un projet concret. Ce projet a renforcé notre capacité à relever des défis techniques et à collaborer efficacement au sein d'une équipe, ce qui sera précieux dans notre parcours professionnel futur.