

Modélisation d'une épidémie

1 Description du modèle choisi

Pour modéliser une épidémie, on partage la population entre quatre catégories : les individus sains, malades, rétablis (ils sont alors immunisés), et décédés. Pour tout $t \in \mathbb{R}$, on note :

- $S(t)$ la proportion d'individus sains à l'instant t ;
- $M(t)$ la proportion d'individus malades à l'instant t ;
- $R(t)$ la proportion d'individus rétablis à l'instant t ;
- $D(t)$ la proportion d'individus décédés à l'instant t .

Notez que $S + M + R + D$ est constante égale à 1.

On suppose l'existence d'une constante $c \in \mathbb{R}^{+*}$, appelée « taux de contagion », telle que $S' = -cSM$ (le nombre de nouveaux infectés est proportionnel à SM , lui-même proportionnelle à la probabilité qu'un individu sain entre en contact avec un porteur).

Pour les autres fonctions, on suppose qu'il existe deux constantes d et r telle que

$$\forall t \in \mathbb{R}, \begin{cases} S'(t) = -cS(t)M(t) \\ D'(t) = dM(t) \\ R'(t) = rM(t) \\ M'(t) = cS(t)M(t) - (d+r)M(t) \end{cases} . \quad (1)$$

On notera t_0 l'instant de début de la simulation. On suppose $M(t_0) > 0$.

2 Analyse mathématique

Ce système n'est pas résoluble analytiquement, ce qui signifie qu'on ne sait pas donner une formule utilisant uniquement les fonctions usuelles. Voyons ce que nous pouvons en dire néanmoins.

Pour simplifier les formules, on prend ici $t_0 = 0$.

1. Vérifier que $S + M + D + R$ est constant, et préciser sa valeur.
2. Au début de l'épidémie, D , M et R sont négligeables devant S , donc S équivaut à 1.
Calculer M .

3 Euler

Pour tout $t \in \mathbb{R}$, on note $X(t) = \begin{pmatrix} S(t) \\ D(t) \\ R(t) \\ M(t) \end{pmatrix}$. Le vecteur $X(t)$ sera enregistré par un tableau numpy dans Python.

1. Pourquoi le type `ndarray` des tableaux numpy ?
2. Écrire une fonction `Xprime` qui prend $X(t)$, c , r , d et qui renvoie $X'(t)$.
3. En déduire une fonction `euler` pour simuler l'évolution de S, I, D, M .

4 Détermination des paramètres

4.1 r et d

Les paramètres r et d ne sont pas connus. En revanche, on connaît la durée moyenne de la maladie, ainsi que son taux de mortalité. Voyons comment déduire les premiers des suivants.

1. Notons m le taux de mortalité (à peu près connu) de la maladie. C'est le rapport du nombre de morts sur le nombre total de malades. Donner une formule liant m , r et d .
2. Supposons ici que $c = 0$: on étudie une situation où aucun nouvel individu est infecté, et on va calculer la durée moyenne de la maladie.

(a) Calculer alors M .

(b) La durée moyenne de la maladie, notée τ , est $\frac{1}{M(t_0)} \int_{t_0}^{\infty} -M'(t)dt$. En effet, penser que $-M'(t)dt$ représente le nombre de personnes pour lesquelles la maladie a duré entre t et $t + dt$.

L'intégrale sur $[t_0, \infty[$ est définie par $\int_{t_0}^{\infty} -M'(t)dt = \lim_{t_f \rightarrow \infty} \int_{t_0}^{t_f} -M'(t)dt$.

Calculer τ en fonction de r et d .

3. Finalement, exprimer r et d en fonction de m et τ . Définir τ et m en variables globales à partir des données que vous aurez trouvées, dans la presse où sur internet. Définir alors r et d en variables globales également.

4.2 Taux de contagion (c)

Le fichier `time_series_19-covid-Confirmed.csv` contient le nombre de personnes détectée positives au COVID-19 chaque jour depuis le 22 janvier 2020 dans chaque pays du monde. Il y a une entrée par jour.

Dans le fichier `bib_épidémie.py`,

- La fonction `extraire_données` récupère les données de la France. Les paramètres `début` et `fin` permettent de sélectionner une période. Ils sont mesurés en nombre de jours à partir du début des mesures, soit du 22 janvier 2020.
- La fonction `dessins_données_passé` affiche le graphe issu de ces données, dans un repère classique d'abord puis semilog. En pratique, le deuxième graphe affiché est celui de $\ln(M(t))$ en fonction de t .

1. Au vu de l'allure générale des graphes, est-ce que les calculs de la partie 2 sont confirmés ?
2. Pour une période située dans la première phase de l'épidémie, le deuxième graphe est supposé être une droite ; donner l'équation théorique de cette droite. On prendra $t_0 = 0$.
3. La fonction `polyfit` de `numpy` permet d'effectuer une régression linéaire. Si on lui fournit `tx` le tableau des abscisses et `ty` le tableau des ordonnées d'un ensemble de points, et qu'on lui précise qu'on veut une regression de degré 1 en tapant `np.polyfit(tx,ty,1)`, elle renvoie un couple (a, b) tel que la droite d'équation $y = ax + b$ est la droite passant au plus près de ces points.

Utiliser `np.polyfit` pour trouver $M(0)$ ainsi que la constante λ telle que au début de l'épidémie, $M(t) = M(0) e^{\lambda t}$.

4. En déduire la valeur de c .

4.3 Conclusion

Tracer l'évolution de l'épidémie avec ces valeurs des paramètres.

5 Durée de confinement

Notons M_c la proportion maximale de population infectée avant que les hopitaux ne soient débordés. Mettons que le taux de mortalité est multiplié par deux lorsque $M(t) > M_c$.

Soit t_{fc} l'instant de fin de confinement. Mettons que c est multiplié par deux après t_{fc} .

1. Écrire une nouvelle variante de votre modélisation pour prendre en compte ces données.
2. Déterminer t_{fc} pour que le nombre de morts au final soit le plus petit possible.