

Boucles, fonctions, listes, chaînes de caractères

Faire apparaître dans votre code votre NOM et votre prénom.

Exercice 1.

Modélisation d'une épidémie par la méthode SIR

La méthode SIR (pour **Sains-Infecteds-Remis**) est l'une des modélisations les plus simples de la propagation d'une épidémie. Dans cette modélisation, on ne s'intéresse qu'à la propagation de l'épidémie, et le groupe des individus **Remis** comprend en fait les individus guéris aussi bien que ceux décédés de la maladie.

Le but de cet exercice est d'en donner une des variantes.

Modélisation avec immunité permanente

- 1) On modélise l'épidémie par trois suites $(S_n)_{n \in \mathbb{N}}$, $(I_n)_{n \in \mathbb{N}}$ et $(R_n)_{n \in \mathbb{N}}$ donnant respectivement le nombre d'individus **sains**, **infectés** et **remis** lors de la semaine n°n.

Ces trois suites sont définies par les relations de récurrence suivantes :

$$\begin{cases} S_0 &= 67.10^6 & I_0 = 1 & R_0 = 0 \\ S_{n+1} &= S_n - \{\alpha S_n I_n\} \\ I_{n+1} &= \{\alpha S_n I_n\} \\ R_{n+1} &= R_n + I_n \end{cases}$$

où $\alpha = 2,2389.10^{-8}$ et

où $\{r\}$ désigne l'entier le plus proche de r . En Python, la fonction `round(r)` permet d'obtenir cet entier.

Écrire une fonction `suites(n)` qui, étant donné le numéro `n` de la semaine, renvoie **les trois listes** constituées des termes S_0, S_1, \dots, S_n , I_0, I_1, \dots, I_n et R_0, R_1, \dots, R_n respectivement.

- 2) Pour tracer une représentation graphique,

- on importe le module `matplotlib.pyplot` :

```
import matplotlib.pyplot as plt
```

- puis on utilise la fonction `plot` de ce module qui permet de tracer une représentation graphique en fournissant la liste des abscisses et la liste des ordonnées de ses points :

```
plt.plot(X,Y)
```

Tracer les trois représentations graphiques constituées des points

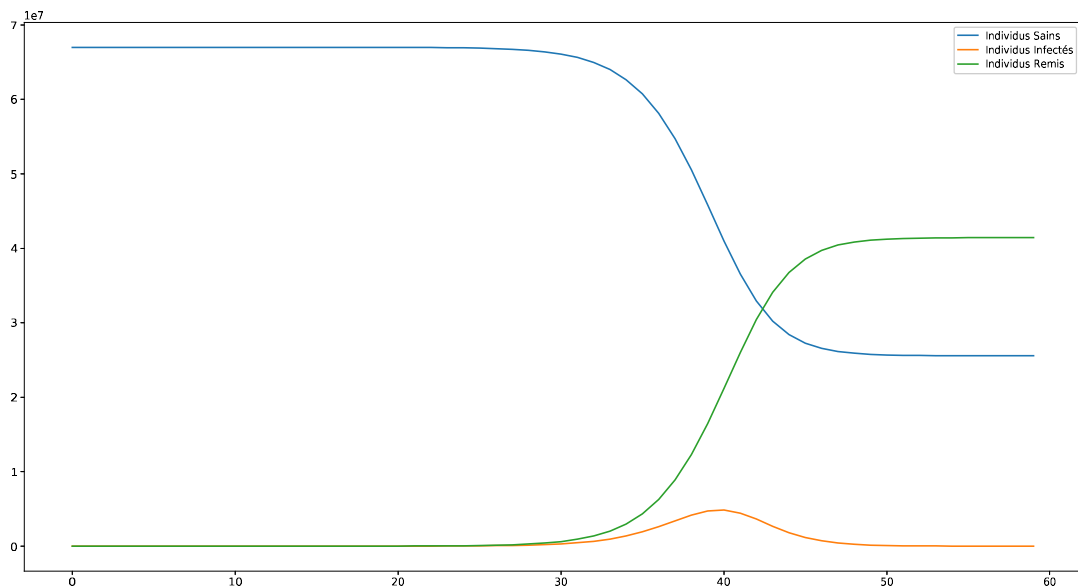
- d'abscisses $\llbracket 0; 60 \rrbracket$ et d'ordonnées $\{S_0; S_1; \dots; S_{60}\}$;
- d'abscisses $\llbracket 0; 60 \rrbracket$ et d'ordonnées $\{I_0; I_1; \dots; I_{60}\}$;
- d'abscisses $\llbracket 0; 60 \rrbracket$ et d'ordonnées $\{R_0; R_1; \dots; R_{60}\}$.

Ces représentations graphiques devraient ressembler au graphe donné au verso.

- 3) Finir cette première partie par la ligne suivante :

```
plt.figure()
```

qui fait apparaître une nouvelle représentation graphique vide.



Modélisation avec immunité temporaire

- 1) Pour tenir compte du fait que l'immunité pourrait ne pas être permanente, on modifie les relations de récurrence de la façon suivante :

$$\begin{cases} S_0 = 67.10^6 & I_0 = 1 & R_0 = 0 \\ S_{n+1} = S_n - \{\alpha S_n I_n\} & \text{si } n < 10 \\ I_{n+1} = \{\alpha S_n I_n\} & \text{si } n < 10 \\ R_{n+1} = R_n + I_n & \text{si } n < 10 \\ S_{n+1} = S_n - \{\alpha S_n I_n\} + \{\beta R_{n-10}\} & \text{si } n \geq 10 \\ I_{n+1} = \{\alpha S_n I_n\} & \text{si } n \geq 10 \\ R_{n+1} = R_n + I_n - \{\beta R_{n-10}\} & \text{si } n \geq 10 \end{cases}$$

où $\beta = 0,03$.

Écrire une fonction `suites2(n)` qui, étant donné le numéro `n` de la semaine, renvoie **les trois listes** constituées des termes S_0, S_1, \dots, S_n , I_0, I_1, \dots, I_n et R_0, R_1, \dots, R_n ainsi modifiées.

- 2) Tracer les trois représentations graphiques constituées des points
- d'abscisses $\llbracket 0; 520 \rrbracket$ et d'ordonnées $\{S_0; S_1; \dots; S_{520}\}$;
 - d'abscisses $\llbracket 0; 520 \rrbracket$ et d'ordonnées $\{I_0; I_1; \dots; I_{520}\}$;
 - d'abscisses $\llbracket 0; 520 \rrbracket$ et d'ordonnées $\{R_0; R_1; \dots; R_{520}\}$.
- 3) **On répondra aux questions ci-dessous par des commentaires dans le fichier Python**
- À quel intervalle de temps correspondent ces trois représentations graphiques ?
 - Quel nouveau phénomène apparaît lorsque l'immunité est temporaire et non pas permanente ?
 - D'après les relations de récurrence, à partir de combien de temps l'immunité s'atténue-t-elle dans cette nouvelle modélisation ?
 - Pouvez-vous donner une explication simple au phénomène observé ?