EXAMEN TERMINAL (2 heures)

Consignes :

1. Tous les documents disponibles sur Moodle sont autorisés. L'utilisation d'Internet est strictement **interdite**.
2. À la fin de l'examen, vous devez fournir un programme Python par exercice. Vos programmes doivent être **exécutables** avec la commande `python3 filename.py` et doivent **afficher leurs résultats**.

I. Modélisation mathématique des maladies infectieuses : le modèle SCIR

Un modèle répandu en épidémiologie pour décrire la propagation d'une maladie dans une population est le modèle SCIR. La population totale est divisée en quatre catégories :

- S représente la fraction d'individus *susceptibles* qui n'ont pas encore contracté la maladie ;
- C représente la fraction d'individus *contaminés* qui ont contracté la maladie mais qui ne sont pas encore contagieux ;
- I représente la fraction d'individus qui ont contracté la maladie et qui sont *infectieux* ;
- R représente la fraction d'individus *rétablis* qui sont de nouveau sains et maintenant immunisés.

Ces quatre populations sont reliées par le système d'équations différentielles ordinaires suivant :

$$\begin{cases} \dfrac{\mathrm{d}S}{\mathrm{d}t} = -\beta I S, \\ \dfrac{\mathrm{d}C}{\mathrm{d}t} = \beta I S - \dfrac{C}{\tau}, \\ \dfrac{\mathrm{d}I}{\mathrm{d}t} = \dfrac{C}{\tau} - \gamma I - \mu I, \\ \dfrac{\mathrm{d}R}{\mathrm{d}t} = \gamma I, \end{cases} \tag{1}$$

dans lequel apparaissent des constantes, à savoir le taux d'incidence β, la durée d'incubation τ, le taux de rétablissement γ et le taux de mortalité μ.

Question 1 : En utilisant une méthode d'intégration explicite vue en cours (*i.e.*, Euler ou RK4), intégrer le système ci-dessus pour des valeurs de paramètres $\beta = 1/2$, $\tau = 10$, $\gamma = 1/12$, $\mu = 1/50$ jusqu'à un temps final $t_\mathrm{f} = 100$, en prenant comme conditions initiales $S(0) = 1 - C_0$, $C(0) = C_0$, $I(0) = 0$, $R(0) = 0$, avec $C_0 = 1/2$. Ajouter un commentaire dans votre code pour justifier votre choix de valeur numérique du pas de temps d'intégration h à partir des temps caractéristiques du problème.
Important : Des points bonus seront attribués pour l'utilisation de tableaux NumPy et pour la vectorisation.

Question 2 : La fraction d'individus encore en vie est définie par $A = S + C + I + R$, et la fraction d'individus décédés est $D = 1 - A$. Tracer sur le même graphique l'évolution temporelle des populations S, C, I, R, A et D. Toutes les populations devront être tracées avec des lignes continues avec le code couleur suivant : S en vert, C en orange, I en rouge, R en bleu, A en violet et D en noir. Le graphique devra comporter une grille. La figure devra se suffire à elle-même et devra donc être compréhensible par une personne n'ayant pas fait l'exercice.

Question 3 : La question précédente montre que $I(t)$ est non monotone et atteint un maximum I_max à un temps t_max. Copier et modifier votre programme de sorte que l'intégration s'arrête automatiquement quand $I(t)$ devient inférieure à $I_\mathrm{max}/100$ après avoir atteint son maximum. Afficher le temps $t_{1\%}$ auquel cette condition est atteinte.
Indice : La valeur de I_max devra être calculée à la volée au cours de l'intégration.

II. L'aiguille de Buffon

Georges-Louis Leclerc de Buffon pose la question suivante (voir Fig. 1) : quelle est la probabilité qu'une aiguille de longueur ℓ lancée sur un sol présentant des lignes parallèles équidistantes de d intersecte l'une de ces lignes ? Il se trouve qu'il existe une expression analytique simple pour cette probabilité, qui ne dépend que du rapport $z = \ell/d$:

$$\mathcal{P}(z) = \begin{cases} \dfrac{2z}{\pi} & \text{si} \quad z < 1, \\ \dfrac{2}{\pi}\left[z - \sqrt{z^2 - 1} + \arccos\left(\dfrac{1}{z}\right)\right] & \text{sinon.} \end{cases} \tag{2}$$

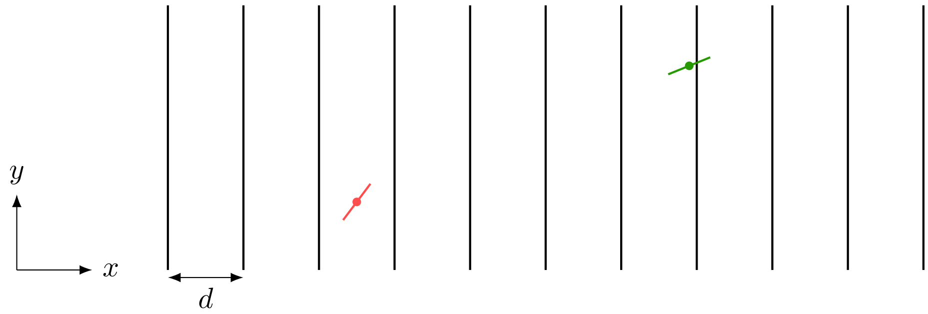

FIGURE 1 – Illustration du problème de l'aiguille de Buffon. Deux aiguilles ont été lancées sur un sol sur lequel ont été dessinées des lignes parallèles équidistantes de d. L'aiguille rose n'a pas croisé une ligne, l'aiguille verte a intersecté une ligne.

Question 1 : Définir une fonction `crossing_proba_th(z)` qui prend en argument une variable `z` qui peut être un flottant ou un tableau unidimensionnel, et retourne la valeur donnée par l'Éq. (2). Dans le cas où l'argument est un tableau, la fonction doit retourner un tableau de même taille, dont les éléments correspondent à la valeur calculée par l'Éq. (2) pour chaque élément du tableau donné en entrée. Dans le cas où l'argument est un flottant, la fonction doit retourner un flottant.
Indice : La fonction $\arccos$ *est implémentée dans NumPy (*`numpy.acos`*).*
Important : Des points bonus seront attribués pour la vectorisation.

Question 2 : Dans la suite, on considère la situation schématisée Fig. 1. Le sol comporte 11 lignes parallèles séparées de $d = 1$ (sans perte de généralité). Le lancer d'une aiguille est paramétré par l'abscisse $x \in [0, 10]$ du centre de l'aiguille (représenté par un point sur le schéma ci-dessus), et par l'angle $\theta \in [0, 2\pi]$ que fait l'aiguille avec l'axe des abscisses. On suppose que l'abscisse du centre de l'aiguille et l'angle fait par l'aiguille avec l'axe des abscisses sont des variables aléatoires réelles uniformément distribuées (l'ordonnée des aiguilles ne joue aucun rôle ici).
Définir une fonction `crossing_proba(ell, N)` qui prend comme argument un flottant `ell` (la longueur des aiguilles), simule le lancer de `N` aiguilles sur le sol (comme décrit précédemment) et renvoie la fraction d'aiguilles qui ont intersecté une ligne sur le sol.
Indice : Pour une aiguille faisant un angle θ *avec l'axe des abscisses et dont le centre a pour abscisse* x*, les extrémités de l'aiguille ont pour abscisses*

$$x_1 = x - \frac{\ell}{2}\cos\theta, \quad \text{et} \quad x_2 = x + \frac{\ell}{2}\cos\theta. \tag{3}$$

Comme les lignes sur le sol sont espacées d'un nombre entier $d = 1$*, il existe un critère très simple s'appuyant sur la partie entière de* x_1 *et* x_2 *pour déterminer si une aiguille a intersecté une ligne.*
Important : Des points bonus seront attribués pour l'utilisation de tableaux NumPy et pour la vectorisation.

Question 3 : Tracer sur le même graphique la solution analytique, en ligne noire continue, et le résultat des simulations Monte Carlo précédentes, avec des points rouges non reliés, pour des valeurs de ℓ entre 0 et 2. La figure devra se suffire à elle-même et devra donc être compréhensible par une personne n'ayant pas fait l'exercice.

Question 4 : Pour $\ell = d$, la probabilité d'intersection vaut $2/\pi$. Cette relation a été utilisée par Buffon pour estimer la valeur de π. Calculer l'estimation de π pour différents nombres de lancers $N = 10^k$ avec $k \in \{2, \ldots, 8\}$. À partir d'un tracé en double échelle logarithmique (log-log), illustrer comment l'estimation Monte Carlo de π converge vers sa valeur exacte en fonction de N. À partir de ce tracé, ajouter un commentaire à votre code pour décrire de quelle manière la convergence se fait en fonction de N.