



PRÉPASIMPLE

prepasimple.fr

HEC

Mathématiques III — Option économique

2006

Mathématiques appliquées

Mathématiques appliquées — Prépa ECG

Annales de concours · Corrigé

Exercice

1. a) La matrice B est réelle et symétrique : ses coefficients vérifient $b_{i,j} = b_{j,i}$. Le résultat du programme selon lequel toute matrice symétrique réelle est diagonalisable s'applique donc. Ainsi B est diagonalisable sur $\mathbb{R}$.

1. b) Méthode 1 — Propagation des coordonnées

On cherche les réels a pour lesquels $BX = aX$ possède une solution non nulle. Les cinq lignes de cette égalité donnent successivement

$$x_2 = ax_1, \quad x_3 = ax_2 - x_1, \quad x_4 = ax_3 - x_2, \quad x_5 = ax_4 - x_3, \quad x_4 = ax_5.$$

Si $x_1 = 0$, les quatre premières relations imposent que toutes les coordonnées soient nulles. Pour un vecteur propre, $x_1 \neq 0$; on peut donc le normaliser à 1. On obtient

$$x_2 = a, \quad x_3 = a^2 - 1, \quad x_4 = a^3 - 2a, \quad x_5 = a^4 - 3a^2 + 1.$$

La dernière ligne impose $a^3 - 2a = a(a^4 - 3a^2 + 1)$, soit

$$a(a^4 - 4a^2 + 3) = a(a^2 - 1)(a^2 - 3) = 0.$$

Réciproquement, pour chacune des cinq racines réelles de cette expression, les coordonnées précédentes vérifient toutes les lignes. On en déduit

$$\text{Sp}(B) = \{-\sqrt{3}, -1, 0, 1, \sqrt{3}\}.$$

Les sous-espaces propres sont les droites suivantes :

$$E_0 = \text{Vect} \left(\begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 1 \end{pmatrix} \right),$$

$$E_a = \text{Vect} \left(\begin{pmatrix} 1 \\ a \\ 0 \\ -a \\ -1 \end{pmatrix} \right) \quad \text{pour } a \in \{-1, 1\},$$

$$E_a = \text{Vect} \left(\begin{pmatrix} 1 \\ a \\ 2 \\ a \\ 1 \end{pmatrix} \right) \quad \text{pour } a \in \{-\sqrt{3}, \sqrt{3}\}.$$

Les vecteurs propres sont exactement les vecteurs non nuls de ces droites. Les cinq valeurs propres sont distinctes ; les cinq vecteurs affichés forment donc une base de $\mathbb{R}^5$.

Méthode 2 — Exploitation de la symétrie des coordonnées

Toute colonne de $\mathbb{R}^5$ se décompose de manière unique en une colonne de la forme ${}^t(u, v, w, v, u)$ et une colonne de la forme ${}^t(u, v, 0, -v, -u)$: il suffit d'additionner, puis de soustraire, la colonne obtenue en renversant l'ordre des coordonnées, et de diviser par 2. Ces deux sous-espaces, de dimensions 3 et 2, sont conservés par B .

Sur le premier, l'équation propre s'écrit

$$v = au, \quad u + w = av, \quad 2v = aw.$$

Pour $a = 0$, elle donne $v = 0$ et $w = -u$, d'où le vecteur de E_0 ci-dessus. Pour $a \neq 0$, un vecteur propre a $u \neq 0$, puis $v = au$ et $w = 2u$; la deuxième égalité devient $3u = a^2u$. On obtient $a = \pm\sqrt{3}$ et les deux vecteurs déjà affichés.

Sur le second sous-espace, les relations sont $v = au$ et $u = av$. Pour une colonne non nulle, $u \neq 0$, donc $a^2 = 1$. Cela redonne les vecteurs associés à $a = \pm 1$. On a ainsi obtenu cinq vecteurs propres indépendants et toutes les valeurs propres. Cette approche réduit le calcul grâce à la symétrie de l'exemple; la propagation des coordonnées, elle, prépare directement la question générale suivante.

2. Les lignes de $A_a X = 0$, avec $x_0 = x_{n+1} = 0$, s'écrivent toutes sous la même forme :

$$\mu x_{j-1} - ax_j + \lambda x_{j+1} = 0, \quad 1 \leq j \leq n.$$

Ce sont exactement les relations (R) aux rangs $0, \dots, n-1$. Si $A_a X = 0$, on peut prolonger les termes donnés en définissant successivement

$$x_{k+2} = \frac{a}{\lambda} x_{k+1} - \frac{\mu}{\lambda} x_k, \quad k \geq n,$$

car $\lambda > 0$. On obtient une suite vérifiant (R) à tous les rangs. Réciproquement, si les $n+2$ termes sont les premiers termes d'une telle suite, ses relations aux rangs $0, \dots, n-1$ donnent les n lignes de $A_a X = 0$. L'équivalence demandée est donc établie.

3. a) L'équation caractéristique de la récurrence réelle du second ordre est

$\lambda r^2 - ar + \mu = 0$. Son discriminant $D = a^2 - 4\lambda\mu$ est strictement positif par hypothèse. Ses deux racines réelles distinctes sont

$$r_+ = \frac{a + \sqrt{D}}{2\lambda}, \quad r_- = \frac{a - \sqrt{D}}{2\lambda}.$$

Le résultat du programme sur les suites récurrentes linéaires d'ordre 2 à racines caractéristiques réelles donne exactement les suites

$$u_k = \alpha r_+^k + \beta r_-^k, \quad k \in \mathbb{N}, \quad (\alpha, \beta) \in \mathbb{R}^2.$$

Il s'agit de l'équation d'une récurrence scalaire, sans calcul de déterminant matriciel.

3. b) Par 2., les coordonnées prolongées s'écrivent sous la forme précédente, avec

$$x_0 = x_{n+1} = 0. \text{ La première condition donne } \beta = -\alpha, \text{ et la seconde}$$

$$\alpha(r_+^{n+1} - r_-^{n+1}) = 0.$$

Or $r_+ r_- = \mu/\lambda > 0$: les deux racines sont non nulles et de même signe. Leur quotient est donc strictement positif et différent de 1 ; sa puissance d'exposant $n+1 \geq 1$ ne vaut pas 1. Ainsi $r_+^{n+1} \neq r_-^{n+1}$, donc $\alpha = \beta = 0$. Toutes les coordonnées de X sont nulles.

4. a) L'équation $\lambda r^2 - ar + \mu = 0$ a maintenant une racine réelle double $r = a/(2\lambda)$.

Cette racine n'est pas nulle car $a^2 = 4\lambda\mu > 0$. Le résultat du programme sur la racine caractéristique double donne exactement

$$u_k = (\alpha + \beta k)r^k, \quad k \in \mathbb{N}, \quad (\alpha, \beta) \in \mathbb{R}^2.$$

4. b) La condition $x_0 = 0$ impose $\alpha = 0$. La condition $x_{n+1} = 0$ impose alors

$$\beta(n+1)r^{n+1} = 0. \text{ Comme } n+1 > 0 \text{ et } r \neq 0, \text{ on a } \beta = 0. \text{ Le vecteur } X \text{ est donc nul.}$$

5. a) Si a est une valeur propre de B , il existe un vecteur non nul X tel que

$$(B - aI_n)X = 0. \text{ Les questions 3. b) et 4. b) excluent respectivement } a^2 > 4\lambda\mu \text{ et } a^2 = 4\lambda\mu. \text{ Toute valeur propre réelle vérifie donc } a^2 < 4\lambda\mu, \text{ c'est-à-dire}$$

$$-2\sqrt{\lambda\mu} < a < 2\sqrt{\lambda\mu}.$$

5. b) Résultat admis dans cette version adaptée : pour cette matrice B , toute valeur propre réelle a vérifie $|a| \leq \lambda + \mu$. C'est le résultat particulier d'Hadamard donné dans le sujet ; il n'est pas nécessaire pour démontrer 5. a), mais sert de comparaison.

Comme $(\sqrt{\lambda} - \sqrt{\mu})^2 \geq 0$, on a $2\sqrt{\lambda\mu} \leq \lambda + \mu$. Notre encadrement strict est donc contenu dans celui donné par Hadamard. Si $\lambda \neq \mu$, il améliore aussi la taille de la borne ; si $\lambda = \mu$, la borne numérique est la même mais les extrémités sont exclues. Le résultat de 5. a) est ainsi plus précis dans les deux cas.

Problème

Partie I : Modèle discret

1. a) Les événements $[X_n = 1]$ et $[X_n = 0]$ forment un système complet. Par les probabilités totales et les transitions données,

$$p_{n+1} = p_n P(X_{n+1} = 1 \mid X_n = 1) + (1 - p_n) P(X_{n+1} = 1 \mid X_n = 0),$$

$$p_{n+1} = p_n^2 + \lambda p_n(1 - p_n) = (1 - \lambda)p_n^2 + \lambda p_n.$$

1. b) L'hypothèse initiale donne $0 < p_0 < 1$. Si $0 < p_n < 1$, alors

$$p_{n+1} = p_n[\lambda + (1 - \lambda)p_n],$$

avec $0 < \lambda + (1 - \lambda)p_n < 1$. Il en résulte $0 < p_{n+1} < p_n < 1$. Par récurrence, $0 < p_n < 1$ pour tout n , ce qui confirme aussi que les deux événements conditionnants ont une probabilité non nulle à chaque rang.

2. a) La différence des termes consécutifs est

$$p_{n+1} - p_n = -(1 - \lambda)p_n(1 - p_n) < 0.$$

La suite est strictement décroissante et minorée par 0 ; elle converge donc vers une limite $b \in [0, p_0]$. La continuité du polynôme définissant la récurrence entraîne

$$b = (1 - \lambda)b^2 + \lambda b, \quad (1 - \lambda)b(b - 1) = 0.$$

Comme $\lambda \neq 1$, la limite est 0 ou 1. Mais $b \leq p_0 < 1$; ainsi $p_n \rightarrow 0$.

2. b) Puisque $0 < p_0 < 1$ et $0 < \lambda < 1$, le réel $a = \lambda + (1 - \lambda)p_0$ appartient à $]0, 1[$. La décroissance donne $p_n \leq p_0$, donc

$$p_{n+1} = p_n[\lambda + (1 - \lambda)p_n] \leq ap_n.$$

Une récurrence fournit $p_n \leq p_0 a^n \leq a^n$. La série géométrique

$$\sum_{n \geq 0} a^n$$

converge car $0 < a < 1$. La comparaison de séries positives établit la convergence de

$$\sum_{n \geq 0} p_n.$$

3. a) Chaque variable Y_n est une somme finie de variables bornées ; elle admet donc une espérance. La linéarité, qui ne nécessite pas l'indépendance, donne

$$E(Y_n) = \sum_{k=0}^n E(X_k) = \sum_{k=0}^n p_k.$$

La série de 2. b) converge ; ses sommes partielles admettent donc une limite finie

$$L = \sum_{k=0}^{+\infty} p_k.$$

3. b) Question adaptée au programme actuel. Code adapté en Python.

On calcule les $n + 1$ termes $p_0, \dots, p_n$ puis leur somme. Aucun import n'est nécessaire.

Python

```
def approx(n, p0, lam):
    p = p0
```

```

somme = p
for k in range(n):
    p = (1 - lam) * p * p + lam * p
    somme = somme + p
return somme

```

Au début, somme vaut p_0 . À chaque passage, p est remplacé par le terme suivant et ajouté à la somme ; la valeur renvoyée est donc $E(Y_n)$. Pour $n = 0$, la boucle n'est pas exécutée et le résultat est correctement p_0 .

La précision peut être contrôlée grâce à la majoration plus forte de 2. b) :

$$0 \leq L - \text{approx}(n, p_0, \lambda) = \sum_{k=n+1}^{+\infty} p_k \leq \frac{p_0 a^{n+1}}{1-a}.$$

La fonction a été exécutée, notamment pour $n = 0$ et pour des valeurs de λ et p_0 dans $]0, 1[$; les sommes obtenues et cette borne d'erreur ont été contrôlées numériquement.

4. a) Les deux variables sont bornées, donc leurs espérances, variances et covariance existent. Leur produit vaut l'indicatrice de l'événement $[X_n = 1, X_{n+1} = 1]$. Ainsi

$$E(X_n X_{n+1}) = P(X_n = 1)P(X_{n+1} = 1 \mid X_n = 1) = p_n^2,$$

$$\text{Cov}(X_n, X_{n+1}) = p_n^2 - p_n p_{n+1} = p_n(p_n - p_{n+1}).$$

La stricte décroissance de (p_n) et sa positivité montrent que cette covariance est strictement positive. Deux variables indépendantes ayant une variance finie ont une covariance nulle ; X_n et X_{n+1} ne sont donc indépendantes pour aucun rang n .

4. b) Comme $p_n > 0$, la récurrence donne

$$\frac{p_{n+1}}{p_n} = \lambda + (1 - \lambda)p_n.$$

Puisque $p_n \rightarrow 0$, ce quotient tend vers λ .

4. c) Les variances de Bernoulli sont $p_n(1 - p_n)$ et $p_{n+1}(1 - p_{n+1})$, toutes deux strictement positives. Le coefficient est donc bien défini et vaut

$$r_n = \frac{p_n(p_n - p_{n+1})}{\sqrt{p_n(1 - p_n)p_{n+1}(1 - p_{n+1})}}.$$

On peut aussi utiliser $p_n - p_{n+1} = (1 - \lambda)p_n(1 - p_n)$ pour écrire

$$\frac{r_n}{p_n} = (1 - \lambda) \sqrt{\frac{p_n(1 - p_n)}{p_{n+1}(1 - p_{n+1})}}.$$

Par 4. b), $p_n/p_{n+1} \rightarrow 1/\lambda$; par 2. a), les deux facteurs $1 - p_n$ et $1 - p_{n+1}$ tendent vers 1. La continuité de la racine carrée sur $]0, +\infty[$ donne

$$\frac{r_n}{p_n} \rightarrow \frac{1 - \lambda}{\sqrt{\lambda}} > 0.$$

Cela établit précisément l'équivalent $r_n \sim \frac{1 - \lambda}{\sqrt{\lambda}} p_n$.

Partie II : Simulation

1. Question adaptée au programme actuel. Code adapté en Python.

Soit U la variable uniforme représentée par `random()`. Pour $p \in [0, 1]$, la fonction renvoie 1 si $U \leq p$ et 0 sinon. Comme $P(U \leq p) = p$, sa sortie suit une loi de Bernoulli de paramètre p . Les valeurs aux extrémités $p = 0$ et $p = 1$ correspondent respectivement à une sortie presque sûrement nulle et presque sûrement égale à 1.

Python

```
from numpy.random import random

def bernoulli(p):
    if random() <= p:
        return 1
    return 0
```

Le code a été exécuté et ses sorties ont été contrôlées ; la détermination exacte de la loi vient du raisonnement sur U , non de l'expérimentation.

2. Question adaptée au programme actuel. Code adapté en Python.

La fonction commence par simuler X_0 avec le paramètre p_0 et initialise C à sa valeur. Au début du passage d'indice i , avec $1 \leq i \leq N$, p vaut p_{i-1} et x contient la valeur simulée de X_{i-1} .

Si x vaut 1, le paramètre de la prochaine Bernoulli est p_{i-1} ; s'il vaut 0, ce paramètre est λp_{i-1} . Après le tirage, x contient donc une valeur de X_i selon les transitions précisées dans cette partie. Elle est ajoutée au cumul. Enfin, la récurrence de I.1. a) transforme p en p_i , prêt pour le passage suivant.

Après les N passages, C contient $X_0 + \dots + X_N$; la fonction renvoie ainsi une réalisation de Y_N . Voici le code complet du calcul de la distribution empirique, avec paramètres explicites; il ne nécessite pas de saisie interactive.

Python

```
from numpy.random import random

def bernoulli(p):
    if random() <= p:
        return 1
    return 0

def simulation(N, p0, lam):
    p = p0
    x = bernoulli(p)
    C = x
    for i in range(1, N + 1):
        q = p
        if x == 0:
            q = lam * p
        x = bernoulli(q)
        C = C + x
        p = (1 - lam) * p * p + lam * p
    return C
```

```
def evaluation(N, p0, lam, repetitions=10000):
    T = [0] * (N + 2)
    for k in range(repetitions):
        y = simulation(N, p0, lam)
        T[y] = T[y] + 1
    return T

def afficher_frequencies(N, p0, lam, repetitions=10000):
    T = evaluation(N, p0, lam, repetitions)
    for k in range(N + 2):
        print(k, T[k] / repetitions)
```

La fonction `evaluation` renvoie les effectifs ; `afficher_frequencies` affiche les valeurs et leurs fréquences. Les paramètres doivent vérifier $0 < p_0 < 1, 0 < \lambda < 1, 1 \leq N \leq 200$, et `repetitions` doit être un entier strictement positif.

Rectification de la source : Y_N est la somme de $N + 1$ Bernoulli et peut valoir $N + 1$. En particulier, la probabilité que tous les tirages soient des succès est

$$p_0 \prod_{i=0}^{N-1} p_i > 0.$$

Il faut donc $N + 2$ cases, indexées de 0 à $N + 1$, et afficher également la dernière. Les dimensions et bornes d’affichage du programme source ont été corrigées en conséquence.

Autre précision nécessaire : les seules probabilités conditionnelles de la partie I, données conditionnellement à X_i , ne déterminent pas en général la loi de toute la suite. La construction de cette partie précise que les transitions restent les mêmes conditionnellement au passé entier. Les tirages successifs indépendants de la fonction reproduisent cette réalisation markovienne, représentée par l’arbre du sujet.

Les fonctions complètes ont été exécutées, y compris pour $N = 200$; la sortie de `simulation` appartient toujours à $\{0, \dots, N + 1\}$ et la somme des effectifs de `evaluation` est le nombre de répétitions. Ces essais contrôlent le code et ne remplacent pas la justification probabiliste.

3. Question adaptée au programme actuel.

Pour $j \in \{0, \dots, N + 1\}$, notons $q_j = P(Y_N = j)$ dans le modèle précisé en partie II. Pour chaque répétition indépendante r , l'indicatrice $I_{r,j}$ de l'événement « la sortie vaut j » est une Bernoulli de paramètre q_j . Les indicatrices des répétitions distinctes sont indépendantes et de même loi.

Après M répétitions, la fréquence de j vaut

$$\hat{q}_{M,j} = M^{-1} \sum_{r=1}^M I_{r,j}.$$

La loi faible des grands nombres donne, pour tout $\eta > 0$,

$$P(|\hat{q}_{M,j} - q_j| > \eta) \rightarrow 0 \quad \text{quand } M \rightarrow +\infty.$$

On peut aussi le quantifier directement par Bienaymé-Tchebychev :

$$P(|\hat{q}_{M,j} - q_j| > \eta) \leq \frac{q_j(1 - q_j)}{M\eta^2} \leq \frac{1}{4M\eta^2}.$$

Le programme approche donc la loi de Y_N par les fréquences $T[j]/10000$. Il affiche seulement des effectifs dans sa version du sujet ; leur division par 10000 donne les probabilités estimées. Pour un nombre fini de répétitions, il s'agit d'approximations aléatoires, sans égalité garantie avec les probabilités exactes.

Partie III : Modèle continu

On pose $c = 1 - \ell > 0$ pour alléger les calculs.

1. La fonction $f(t) = p(t)e^{ct}$ est dérivable sur $[0, T]$, avec les dérivées unilatérales aux extrémités. L'équation vérifiée par p donne

$$f'(t) = e^{ct}[p'(t) + cp(t)] = ce^{ct}p(t)^2 \geq 0.$$

La fonction f est donc croissante. Pour tout $t \in [0, T]$, $f(t) \geq f(0) = p_0 > 0$. Comme $e^{ct} > 0$, on obtient $p(t) > 0$. Ainsi p ne s'annule pas sur l'intervalle, ce qui autorise la division de la question suivante.

2. a) La fonction $g(t) = e^{-ct}/p(t)$ est dérivable puisque p est dérivable et strictement positive. En utilisant $p' = cp(p-1)$,

$$g'(t) = -\frac{ce^{-ct}}{p(t)} - \frac{e^{-ct}cp(t)(p(t)-1)}{p(t)^2} = -ce^{-ct}.$$

C'est la dérivée de $t \mapsto e^{-ct}$. La différence $g(t) - e^{-ct}$ a donc une dérivée nulle sur l'intervalle et y est constante. Il existe $k \in \mathbb{R}$ tel que

$$g(t) = k + e^{-ct} = k + e^{(\ell-1)t}.$$

2. b) En prenant $t = 0$, on obtient $k = 1/p_0 - 1 = (1 - p_0)/p_0$. L'égalité de 2. a), multipliée par e^{ct} , devient

$$\frac{1}{p(t)} = 1 + \frac{1 - p_0}{p_0} e^{ct}.$$

Le membre de droite est strictement positif. Son inversion donne

$$p(t) = \frac{p_0}{p_0 + (1 - p_0)e^{ct}}.$$

Cette expression vérifie directement la condition initiale et l'équation donnée. Elle définit aussi le prolongement de la même fonction pour tous les réels $t \geq 0$, utile lorsque T varie dans la suite.

2. c) Le dénominateur est strictement positif et croît avec t , donc p est strictement décroissante. Plus explicitement, $0 < p(t) \leq p_0 < 1$ entraîne $p'(t) = cp(t)(p(t) - 1) < 0$. Le tableau est

t	0		T
$p'(t)$		—	
$p(t)$	p_0	$\searrow$	$p(T)$

Ici $p(T) = p_0/[p_0 + (1 - p_0)e^{cT}]$.

La formule explicite montre que p est de classe C^∞ sur un ouvert contenant $[0, T]$, car son dénominateur y est strictement positif. En dérivant l'équation de p ,

$$p''(t) = cp'(t)(2p(t) - 1) = c^2p(t)(p(t) - 1)(2p(t) - 1).$$

Les facteurs $p(t)$ et $1 - p(t)$ sont strictement positifs; le signe de p'' est donc l'opposé de celui de $2p(t) - 1$. Comme p est strictement décroissante, un changement de convexité à l'intérieur de l'intervalle existe exactement lorsque $p(0) > 1/2$ et $p(T) < 1/2$. Ces conditions sont équivalentes à

$$\frac{1}{2} < p_0 < \frac{e^{cT}}{1 + e^{cT}}.$$

Dans ce cas, la solution unique de $p(t_*) = 1/2$ est

$$t_* = \frac{1}{c} \ln \left(\frac{p_0}{1 - p_0} \right).$$

On a bien $0 < t_* < T$; la courbe est concave avant t_* et convexe après. Son point d'inflexion a pour coordonnées $\left(\frac{1}{1 - \ell} \ln \frac{p_0}{1 - p_0}, \frac{1}{2} \right)$. Lorsque p_0 est égal à l'une des bornes de la condition, l'annulation de p'' se produit à une extrémité et ne constitue pas un point d'inflexion intérieur. La précision «à l'intérieur» explicite ce point de domaine.

3. a) Question adaptée au programme actuel.

Toutes les variables $X(t_k)$ sont bornées; leur somme finie admet une espérance. Par linéarité, sans hypothèse d'indépendance,

$$E(Z_n) = \sum_{k=0}^n p(t_k).$$

Pour chaque $k \in \{0, \dots, n-1\}$, la décroissance et la continuité de p donnent

$$\delta p(t_{k+1}) \leq \int_{t_k}^{t_{k+1}} p(t) dt \leq \delta p(t_k).$$

En sommant puis en divisant par $T = n\delta > 0$, on obtient

$$\frac{1}{n} \sum_{k=1}^n p(t_k) \leq \frac{1}{T} \int_0^T p(t) dt \leq \frac{1}{n} \sum_{k=0}^{n-1} p(t_k).$$

Notons $S_n = E(Z_n)/n$ et $I = T^{-1} \int_0^T p(t) dt$. Cet encadrement s'écrit $S_n - p_0/n \leq I \leq S_n - p(T)/n$, donc

$$\frac{p(T)}{n} \leq S_n - I \leq \frac{p_0}{n}.$$

Les deux bornes tendent vers 0; ainsi $E(Z_n)/n \rightarrow I = m(T)$. Cette démonstration particulière repose uniquement sur la monotonie et les propriétés de l'intégrale sur un segment.

3. b) Le changement $u = e^{ct}$ est de classe C^1 , strictement croissant sur $[0, T]$, de dérivée $ce^{ct} > 0$. Il transforme les bornes en 1 et e^{cT} , avec $dt = du/(cu)$. Le dénominateur $p_0 + (1 - p_0)u$ reste strictement positif sur cet intervalle. Ainsi

$$m(T) = \frac{1}{cT} \int_1^{e^{cT}} \frac{p_0}{u[p_0 + (1 - p_0)u]} du.$$

L'identité algébrique

$$\frac{p_0}{u[p_0 + (1 - p_0)u]} = \frac{1}{u} - \frac{1 - p_0}{p_0 + (1 - p_0)u}$$

s'obtient en réduisant au même dénominateur; elle donne exactement l'expression demandée.

3. c) Sur l'intervalle d'intégration, les arguments des logarithmes sont positifs. Une primitive de l'intégrande de 3. b) est $\ln u - \ln[p_0 + (1 - p_0)u]$. À la borne inférieure $u = 1$, elle vaut 0, car $p_0 + (1 - p_0) = 1$. On obtient

$$m(T) = \frac{cT - \ln[p_0 + (1 - p_0)e^{cT}]}{cT} = -\frac{\ln[1 - p_0 + p_0e^{-cT}]}{cT}.$$

Lorsque $T \rightarrow +\infty$, $e^{-cT} \rightarrow 0$. Par continuité du logarithme au réel strictement positif $1 - p_0$, le numérateur $-\ln[1 - p_0 + p_0e^{-cT}]$ tend vers $-\ln(1 - p_0) > 0$. Par conséquent

$$\frac{m(T)}{-\ln(1 - p_0)/(cT)} = \frac{\ln[1 - p_0 + p_0e^{-cT}]}{\ln(1 - p_0)} \rightarrow 1.$$

Cela prouve $m(T) \sim -\ln(1 - p_0)/[(1 - \ell)T]$, avec p_0 et ℓ fixés.

Partie IV : Retour au modèle discret

On pose $h = 1 - \lambda \in]0, 1[$. Dans cette partie, $\ell = 1/2$, $p_0 = 1/3$, $T = 2nh$ et $\delta = 2h$. La formule de III.2. b) devient

$$p(t) = \frac{1}{1 + 2e^{t/2}}.$$

On utilisera $0 < p(t) \leq 1/3$ pour $t \geq 0$ et $0 < p_k \leq 1/3$ pour $k \geq 0$, d'après les décroissances démontrées précédemment.

1. Le dénominateur de l'expression explicite de p est strictement positif ; la fonction est donc de classe C^∞ , en particulier deux fois dérivable. En dérivant $p' = \frac{1}{2}p(p - 1)$,

$$p'' = \frac{1}{2}p'(2p - 1) = \frac{1}{4}(2p - 1)p(p - 1).$$

Cette égalité vaut sur tout $[0, T]$, comme demandé.

2. a) Question adaptée au programme actuel. Résultat admis dans cette version adaptée : avec $p(t) = 1/(1 + 2e^{t/2})$, $\delta = 2(1 - \lambda)$ et $t_k = k\delta$, pour tout $k \in \{0, \dots, n - 1\}$,

$$|p(t_{k+1}) - p(t_k) - \delta p'(t_k)| \leq \frac{\delta^2}{8}.$$

Cette seule majoration particulière est donnée pour comparer les deux suites de nombres ; elle servira en 2. c). Aucune méthode générale de discrétisation d'une équation différentielle n'est requise.

2. b) On utilise uniquement les relations déjà établies pour les deux modèles. Puisque $\delta = 2h$ et $p' = \frac{1}{2}p(p - 1)$, on a

$$p(t_k) + \delta p'(t_k) = p(t_k) + h p(t_k)(p(t_k) - 1).$$

La récurrence de I.1. a) s'écrit aussi $p_{k+1} = p_k + h p_k(p_k - 1)$. En posant $u = p(t_k)$ et $v = p_k$, la différence vaut donc

$$(u - v) + h[u^2 - u - v^2 + v] = (u - v)[1 + h(u + v - 1)].$$

Comme $u - v = \varepsilon_k$, on obtient bien

$$p(t_k) + \delta p'(t_k) - p_{k+1} = \varepsilon_k[1 - (1 - \lambda)(1 - p(t_k) - p_k)].$$

2. c) Les bornes $0 < p(t_k), p_k \leq 1/3$ donnent

$$\frac{1}{3} \leq 1 - p(t_k) - p_k < 1.$$

En multipliant par $h > 0$ et en soustrayant à 1, il vient

$$\lambda = 1 - h < 1 - h(1 - p(t_k) - p_k) \leq 1 - \frac{h}{3} = \frac{\lambda + 2}{3}.$$

Le facteur de 2. b) est donc positif et majoré par $(\lambda + 2)/3$. D'autre part,

$$\varepsilon_{k+1} = [p(t_{k+1}) - p(t_k) - \delta p'(t_k)] + [p(t_k) + \delta p'(t_k) - p_{k+1}].$$

L'inégalité triangulaire, la majoration admise de 2. a) et l'identité de 2. b) donnent

$$|\varepsilon_{k+1}| \leq \frac{\delta^2}{8} + \frac{\lambda + 2}{3} |\varepsilon_k|.$$

2. d) Méthode 1 — Récurrence avec la borne demandée

On a $\varepsilon_0 = p(0) - p_0 = 0$. Supposons $|\varepsilon_k| \leq 6h$ à un rang $k < n$. Comme $\delta^2/8 = h^2/2$ et $(\lambda + 2)/3 = 1 - h/3$, l'inégalité précédente donne

$$|\varepsilon_{k+1}| \leq \frac{h^2}{2} + \left(1 - \frac{h}{3}\right) 6h = 6h - \frac{3}{2}h^2 \leq 6h.$$

La propriété est donc vraie pour tout $k \in \{0, \dots, n\}$, soit $|\varepsilon_k| \leq 6(1 - \lambda)$.

Méthode 2 — Sommation d'une majoration géométrique

Posons $q = 1 - h/3 \in]0, 1[$. En partant de $|\varepsilon_0| = 0$ et en appliquant successivement 2. c), une récurrence donne

$$|\varepsilon_k| \leq \frac{h^2}{2} \sum_{j=0}^{k-1} q^j = \frac{h^2}{2} \frac{1 - q^k}{1 - q} \leq \frac{3}{2}h,$$

avec une somme vide égale à 0 si $k = 0$. Cette borne plus forte entraîne celle demandée. La première méthode est adaptée à une constante imposée ; la seconde explique l'accumulation des erreurs et fournit une constante meilleure.

3. a) L'hypothèse $\alpha > 18h$ entraîne

$$\frac{\alpha}{12h} - \frac{1}{2} > \frac{18}{12} - \frac{1}{2} = 1.$$

L'argument du logarithme est donc strictement supérieur à 1, et son logarithme est strictement positif. Puisque $h > 0$, $N(\alpha) = h^{-1} \ln[\alpha/(12h) - 1/2] > 0$.

3. b) Pour $0 \leq k \leq n$, la décroissance de p donne $p(t_k) \geq p(T) > 0$. La division est donc justifiée. Par 2. d) et l'expression explicite de p ,

$$\left| \frac{p(t_k) - p_k}{p(t_k)} \right| \leq \frac{6h}{p(T)} = 6h(1 + 2e^{T/2}) = 6h(1 + 2e^{nh}).$$

L'hypothèse $n \leq N(\alpha)$ implique, après multiplication par $h > 0$ puis application de l'exponentielle croissante,

$$e^{nh} \leq \frac{\alpha}{12h} - \frac{1}{2}.$$

Ainsi

$$6h(1 + 2e^{nh}) \leq 6h \left[1 + 2 \left(\frac{\alpha}{12h} - \frac{1}{2} \right) \right] = \alpha.$$

La majoration relative demandée vaut donc simultanément à tous les rangs $0, \dots, n$.

3. c) On fixe $\alpha > 0$. Quand $\lambda \rightarrow 1^-$, le réel $h = 1 - \lambda$ tend vers 0^+ . L'hypothèse $\alpha > 18h$ est satisfaite pour tout h assez petit. De plus, $\alpha/(12h) - 1/2 \rightarrow +\infty$, donc son logarithme tend vers $+\infty$. Comme $1/h \rightarrow +\infty$ et ces deux facteurs sont alors positifs,

$$N(\alpha) = \frac{1}{h} \ln \left(\frac{\alpha}{12h} - \frac{1}{2} \right) \rightarrow +\infty.$$

Par exemple, pour toute borne $M > 0$, les deux facteurs finissent par dépasser respectivement M et 1, ce qui prouve directement que leur produit dépasse M .

3. d) Pour toute tolérance relative fixée $\alpha > 0$ et tout nombre fixé de rangs n , lorsque λ est suffisamment proche de 1 par valeurs inférieures, on a à la fois $\alpha > 18(1 - \lambda)$ et $n \leq N(\alpha)$. Le résultat de 3. b) assure alors une erreur relative au plus α pour chacun des paramètres $p_0, \dots, p_n$, comparés à $p(t_0), \dots, p(t_n)$.

Le nombre de rangs pour lesquels cette garantie est disponible peut croître sans borne puisque $N(\alpha) \rightarrow +\infty$. De plus, la borne absolue $6(1 - \lambda)$ de 2. d) tend vers 0. La comparaison concerne les probabilités marginales de succès; elle n'établit pas à elle seule une approximation des lois conjointes ni une identité entre les processus.

Relevé des adaptations

- I.3. b) : code adapté en Python, avec calcul des termes p_0 à p_n et contrôle explicite du reste de série.
- II, préambule et questions 2 et 3 : la construction est précisée conditionnellement au passé entier pour déterminer la loi simulée; les seules transitions conditionnées par le dernier état de la partie I ne suffisent pas à déterminer une loi jointe générale. La simulation construit une réalisation markovienne compatible avec ces transitions.
- II, figure : l'arbre est retranscrit en SVG dans l'énoncé à partir de l'original, avec ses branches et les probabilités symboliques données; aucune mesure numérique de probabilité n'est nécessaire.

- II.1 à II.3 : code adapté en Python, syntaxe de la génération uniforme documentée et fréquences distinguées des effectifs. L'histogramme comporte $N + 2$ cases et affiche les valeurs de 0 à $N + 1$, afin de rectifier l'omission de la valeur possible $Y_N = N + 1$ dans le programme source.
- III.2. c) : la localisation intérieure du point d'inflexion est explicitée ; les deux cas d'annulation de la dérivée seconde à une extrémité sont distingués.
- III.3. a) : indication d'un encadrement par monotonie sur chaque segment, afin d'établir la convergence particulière sans utiliser un théorème général sur les sommes de Riemann.
- IV.2. a) : question adaptée au programme actuel ; seule la majoration particulière utile aux comparaisons suivantes est admise, afin de ne pas exiger la discrétisation d'une équation différentielle ni une formule de Taylor avec reste hors du cadre requis. Les identités algébriques et majorations des questions suivantes sont conservées et démontrées.