



PRÉPASIMPLE

prepasimple.fr

EDHEC

Mathématiques — Option économique

1996

Mathématiques appliquées

Mathématiques appliquées — Prépa ECG

Annales de concours · Corrigé

Exercice 1

1. a) On applique successivement la relation de récurrence, avec les deux termes déjà connus :

$$d_2 = 1(d_1 + d_0) = 1, \quad d_3 = 2(d_2 + d_1) = 2,$$

$$d_4 = 3(d_3 + d_2) = 9, \quad d_5 = 4(d_4 + d_3) = 44.$$

1. b) On procède par récurrence à deux termes. Les nombres $d_0 = 1$ et $d_1 = 0$ sont des entiers naturels. Si, pour un entier $n \geq 1$, d_{n-1} et d_n sont dans $\mathbb{N}$, leur somme puis son produit par n sont dans $\mathbb{N}$. Ainsi $d_{n+1} = n(d_n + d_{n-1}) \in \mathbb{N}$. La propriété est donc vraie pour tout $n \in \mathbb{N}$.

2. On cherche à transformer la récurrence d'ordre deux en une relation entre deux termes consécutifs. Posons $r_n = d_{n+1} - (n+1)d_n$ pour $n \geq 0$. On a $r_0 = d_1 - d_0 = -1$ et, pour $n \geq 1$,

$$r_n = n(d_n + d_{n-1}) - (n+1)d_n = nd_{n-1} - d_n = -r_{n-1}.$$

La suite (r_n) est géométrique de raison -1 , d'où $r_n = (-1)^{n+1}$. On obtient bien, y compris pour $n = 0$,

$$d_{n+1} = (n+1)d_n + (-1)^{n+1}.$$

3. Les intégrandes $t \mapsto t^n e^t$ sont continues sur $[0, 1]$, ce qui assure l'existence de toutes les intégrales utilisées.

3. a) Pour $n = 0$, $I_0 = [e^t]_0^1 = e - 1$. Pour exprimer I_{n+1} , on intègre par parties les fonctions de classe C^1 $t \mapsto t^{n+1}$ et $t \mapsto e^t$ sur $[0, 1]$:

$$\int_0^1 t^{n+1} e^t dt = [t^{n+1} e^t]_0^1 - (n+1) \int_0^1 t^n e^t dt = e - (n+1)n! I_n.$$

Après division par $(n+1)!$, il vient

$$I_{n+1} = \frac{e}{(n+1)!} - I_n.$$

3. b) On utilise simultanément les récurrences des questions 2 et 3. a). Pour $n = 0$, $0![1 + I_0] = 1 + e - 1 = e = ed_0$. Supposons $ed_n = n![1 + (-1)^n I_n]$. Alors

$$ed_{n+1} = (n+1)ed_n + e(-1)^{n+1} = (n+1)![1 + (-1)^n I_n] + e(-1)^{n+1}.$$

Or

$$(n+1)![1 + (-1)^{n+1} I_{n+1}] = (n+1)![1 - (-1)^{n+1} I_n] + e(-1)^{n+1},$$

et $-(-1)^{n+1} = (-1)^n$. Les deux expressions coïncident. La récurrence donne donc $ed_n = n![1 + (-1)^n I_n]$ pour tout entier naturel n .

3. c) Le résultat précédent donne

$$d_n - \frac{n!}{e} = \frac{(-1)^n}{e} \int_0^1 t^n e^t dt.$$

Sur $[0, 1]$, $0 \leq t^n e^t \leq et^n$. La positivité de l'intégrale permet donc d'écrire

$$\left| d_n - \frac{n!}{e} \right| = \frac{1}{e} \int_0^1 t^n e^t dt \leq \int_0^1 t^n dt = \frac{1}{n+1}.$$

L'inégalité vaut aussi pour $n = 0$.

3. d) Rectification du renvoi de l'original : la valeur de d_5 avait été calculée en 1. a), et non à la deuxième question.

Pour $n \geq 2$, le nombre d_n est entier et sa distance à $n!/e$ est au plus $1/(n+1) \leq 1/3 < 1/2$. Deux entiers distincts ne peuvent vérifier cette condition : leur distance serait au plus $2/(n+1) < 1$, alors qu'elle vaut au moins 1. L'existence de d_n étant acquise, cette condition le détermine de manière unique : c'est l'entier le plus proche de $n!/e$.

Ainsi $5!/e \simeq 44,14553$ donne $d_5 = 44$, tandis que $10!/e \simeq 1\,334\,960,91612$ donne $d_{10} = 1\,334\,961$.

Pour contrôler les arrondis sans supposer une valeur décimale de e suffisamment précise, la relation de la question 2 fournit successivement $d_6 = 265$, $d_7 = 1854$, $d_8 = 14833$, $d_9 = 133496$ et $d_{10} = 1334961$. Ces entiers concordent avec l'encadrement.

Exercice 2

Partie I

1. Chaque coefficient du produit J^2 est la somme de trois produits égaux à 1, donc $J^2 = 3J$. On en déduit par récurrence que $J^n = 3^{n-1}J$ pour tout $n \geq 1$: c'est vrai au rang 1, et $J^{n+1} = 3^{n-1}J^2 = 3^nJ$ assure l'hérédité. Cela répond en particulier à la demande pour $n \geq 2$.

2. Méthode 1 — Binôme de matrices qui commutent.

On observe $M = \frac{1}{2}I + \frac{1}{6}J$. Les matrices I et J commutent, puisque $IJ = JI = J$; la formule du binôme s'applique donc. Pour $n \geq 1$,

$$M^n = \frac{1}{2^n} I + \sum_{k=1}^n \binom{n}{k} \left(\frac{1}{2}\right)^{n-k} \left(\frac{1}{6}\right)^k 3^{k-1} J.$$

Comme $3^{k-1}/6^k = 1/(3 \cdot 2^k)$, le coefficient de J est

$$\frac{1}{3 \cdot 2^n} \sum_{k=1}^n \binom{n}{k} = \frac{2^n - 1}{3 \cdot 2^n} = \frac{1}{3} \left(1 - \frac{1}{2^n}\right).$$

Pour $n = 0$, l'expression annoncée vaut $I = M^0$. La formule est donc valable pour tout $n \in \mathbb{N}$.

Méthode 2 — Récurrence matricielle.

Cette méthode vérifie directement une formule de puissance sans développer le binôme.

Posons $\alpha_n = 2^{-n}$ et $\beta_n = (1 - 2^{-n})/3$. Au rang 0, $M^0 = \alpha_0 I + \beta_0 J = I$. Si

$M^n = \alpha_n I + \beta_n J$, alors, en utilisant $J^2 = 3J$,

$$M^{n+1} = (\alpha_n I + \beta_n J) \left(\frac{1}{2} I + \frac{1}{6} J\right) = \frac{\alpha_n}{2} I + \left(\frac{\alpha_n}{6} + \beta_n\right) J.$$

Or $\alpha_n/2 = \alpha_{n+1}$ et

$$\frac{\alpha_n}{6} + \beta_n = \frac{2^{-n}}{6} + \frac{1 - 2^{-n}}{3} = \frac{1 - 2^{-(n+1)}}{3} = \beta_{n+1}.$$

La récurrence est établie. Le binôme explique l'origine de la formule ; la récurrence est utile dès qu'une forme candidate a été proposée.

Partie II

1. À chaque instant, le mobile occupe exactement un sommet. Les événements A_n , B_n et C_n forment donc un système complet d'événements. Par additivité, $a_n + b_n + c_n = 1$ pour tout entier naturel n .
2. a) La probabilité de quitter un sommet est $1 - 2/3 = 1/3$; les deux autres sommets sont équiprobables, donc chaque transition vers un autre sommet a pour probabilité $1/6$. La formule des probabilités totales donne

$$a_{n+1} = \frac{2}{3}a_n + \frac{1}{6}b_n + \frac{1}{6}c_n,$$

$$b_{n+1} = \frac{1}{6}a_n + \frac{2}{3}b_n + \frac{1}{6}c_n,$$

$$c_{n+1} = \frac{1}{6}a_n + \frac{1}{6}b_n + \frac{2}{3}c_n.$$

Ces formules utilisent les probabilités de transition spécifiées par le modèle. Si un événement de départ a une probabilité nulle, sa contribution est nulle : aucune probabilité conditionnelle sur cet événement nul n'a besoin d'être définie.

2. b) En soustrayant la deuxième égalité de la première,

$$a_{n+1} - b_{n+1} = \left(\frac{2}{3} - \frac{1}{6}\right)a_n + \left(\frac{1}{6} - \frac{2}{3}\right)b_n = \frac{1}{2}(a_n - b_n).$$

En soustrayant la troisième de la première, les termes en b_n s'annulent et l'on obtient de même $a_{n+1} - c_{n+1} = \frac{1}{2}(a_n - c_n)$.

3. L'hypothèse de départ donne $a_0 = 1$ et $b_0 = c_0 = 0$.

3. a) Méthode 1 — Différences géométriques.

Par 2. b), les suites $(a_n - b_n)$ et $(a_n - c_n)$ sont géométriques de raison $1/2$, de premier terme 1. Ainsi $a_n - b_n = a_n - c_n = 2^{-n}$, donc $b_n = c_n = a_n - 2^{-n}$. Avec $a_n + b_n + c_n = 1$, on obtient $3a_n - 2 \cdot 2^{-n} = 1$, puis

$$a_n = \frac{1}{3} + \frac{2}{3 \cdot 2^n}, \quad b_n = c_n = \frac{1}{3} - \frac{1}{3 \cdot 2^n}.$$

Méthode 2 — Récurrence affine de chaque probabilité.

En utilisant la somme des probabilités dans 2. a), on peut écrire $a_{n+1} = \frac{1}{2}a_n + \frac{1}{6}$. Ainsi $a_{n+1} - 1/3 = \frac{1}{2}(a_n - 1/3)$ et $a_0 - 1/3 = 2/3$, d'où $a_n - 1/3 = 2/(3 \cdot 2^n)$. De même, $b_{n+1} - 1/3 = \frac{1}{2}(b_n - 1/3)$ et $c_{n+1} - 1/3 = \frac{1}{2}(c_n - 1/3)$. Leurs premiers termes valent $-1/3$, donc $b_n = c_n = 1/3 - 1/(3 \cdot 2^n)$. Les différences exploitent les relations demandées en 2. b) ; le recentrage affine permet de traiter chaque sommet séparément.

3. b) La première colonne de la formule de la partie I est

$$\begin{pmatrix} 2^{-n} + \frac{1 - 2^{-n}}{3} \\ \frac{1 - 2^{-n}}{3} \\ \frac{1 - 2^{-n}}{3} \end{pmatrix} = \begin{pmatrix} 1/3 + 2/(3 \cdot 2^n) \\ 1/3 - 1/(3 \cdot 2^n) \\ 1/3 - 1/(3 \cdot 2^n) \end{pmatrix}.$$

La comparaison avec 3. a) vérifie donc l'égalité demandée.

3. c) On donne maintenant une preuve structurelle de cette égalité, sans utiliser les formules explicites de 3. a). Avec la convention du programme, la loi est la matrice

ligne $V_n = (a_n \ b_n \ c_n)$ et 2. a) s'écrit $V_{n+1} = V_n M$, donc $V_n = V_0 M^n$ par récurrence.

L'original présente les probabilités en colonne. En posant $X_n = {}^t V_n$, la transposition donne $X_{n+1} = {}^t M X_n$. Ici M est symétrique, donc ${}^t M = M$: la relation en colonne est

$$X_{n+1} = M X_n. \text{ Par récurrence, } X_n = M^n X_0. \text{ Puisque } X_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \text{ le produit } M^n X_0$$

sélectionne précisément la première colonne de M^n . Cela démontre le résultat et explique la compatibilité des deux conventions dans cet exemple.

4. Pour obtenir la deuxième colonne, on fait partir le mobile de B ; sa loi initiale en colonne est alors le deuxième vecteur de la base canonique. Pour la troisième, on le fait partir de C , avec le troisième vecteur de la base canonique. Le même raisonnement $X_n = M^n X_0$ sélectionne respectivement les deuxième et troisième colonnes. Aucun calcul supplémentaire n'est nécessaire.

Exercice 3

1. Méthode 1 — Étude d'une différence.

Posons $h(t) = \ln(1+t) - t/(1+t)$ sur $[0, +\infty[$. Cette fonction est dérivable et $h'(t) = 1/(1+t) - 1/(1+t)^2 = t/(1+t)^2$. Sa dérivée est strictement positive pour $t > 0$, et $h(0) = 0$. Donc $h(t) > 0$ pour tout $t > 0$, ce qui donne l'inégalité.

Méthode 2 — Comparaison d'intégrales.

Pour $t > 0$ fixé et $s \in [0, t]$, $1/(1+s) \geq 1/(1+t)$, avec inégalité stricte pour $s < t$. L'intégrale de cette différence continue et positive, non identiquement nulle, est strictement positive. Par conséquent,

$$\ln(1+t) = \int_0^t \frac{ds}{1+s} > \int_0^t \frac{ds}{1+t} = \frac{t}{1+t}.$$

L'étude de la différence est réutilisable pour comparer deux fonctions ; l'intégrale relie ici l'inégalité à la décroissance de l'intégrande.

2. La quantité $1 + e^x$ est strictement positive pour tout x ; la fonction f est donc de classe C^1 sur $\mathbb{R}$.

2. a) La dérivation du produit donne

$$f'(x) = -e^{-x} \ln(1 + e^x) + e^{-x} \frac{e^x}{1 + e^x} = e^{-x} \left[\frac{e^x}{1 + e^x} - \ln(1 + e^x) \right].$$

La question 1 appliquée à $t = e^x > 0$ montre que le crochet est strictement négatif. Comme $e^{-x} > 0$, $f'(x) < 0$ partout : f est strictement décroissante sur $\mathbb{R}$.

2. b) À $-\infty$, posons $t = e^x$, qui tend vers 0 par valeurs positives. Alors

$f(x) = \ln(1+t)/t$. La dérivée de $t \mapsto \ln(1+t)$ en 0 vaut 1, donc son taux d'accroissement $[\ln(1+t) - \ln 1]/t$ tend vers 1. Ainsi $\lim_{x \rightarrow -\infty} f(x) = 1$.

À $+\infty$, pour $x \geq 0$, on a $1 + e^x \leq 2e^x$ et donc $0 < f(x) \leq e^{-x}(x + \ln 2)$. Les croissances comparées donnent $xe^{-x} \rightarrow 0$ et $e^{-x} \ln 2 \rightarrow 0$. Par encadrement, $\lim_{x \rightarrow +\infty} f(x) = 0$.

Le tableau de variations est donc :

x	$-\infty$		$+\infty$
$f'(x)$		—	
$f(x)$	1	$\searrow$	0

3. a) La dérivée calculée ci-dessus vérifie $f'(x) = -f(x) + 1/(1 + e^x)$. Or $1/(1 + e^x) = 1 - e^x/(1 + e^x)$, d'où $f(x) = 1 - f'(x) - e^x/(1 + e^x)$.

Les termes de droite ont pour primitives respectives x , $-f(x)$ et $-\ln(1 + e^x)$. Ainsi une primitive sur $\mathbb{R}$ est

$$F(x) = x - f(x) - \ln(1 + e^x) = -f(x) - \ln(1 + e^{-x}).$$

La dernière écriture résulte de $1 + e^x = e^x(1 + e^{-x})$ et sera utile à l'infini.

3. b) Pour tout $R \geq 0$, la continuité de f sur $[0, R]$ permet d'écrire

$$\int_0^R f(x) dx = F(R) - F(0). \text{ Par 2. b) et par continuité de } \ln \text{ en } 1,$$

$$F(R) = -f(R) - \ln(1 + e^{-R}) \rightarrow 0.$$

Comme $f(0) = \ln 2$, on a $F(0) = -2 \ln 2$. La limite des intégrales sur segment existe et est finie : l'intégrale impropre converge et

$$\int_0^{+\infty} f(x) dx = 2 \ln 2.$$

4. La fonction f est strictement positive. La positivité de g nécessite donc $\alpha \geq 0$. La fonction g est continue sur chaque demi-droite ouverte et admet des limites finies de part et d'autre de 0, à savoir 0 à gauche et $\alpha \ln 2$ à droite. Elle satisfait donc les conditions de régularité attendues d'une densité.

La normalisation impose $\int_{-\infty}^{+\infty} g(x) dx = \alpha 2 \ln 2 = 1$. Comme $\ln 2 > 0$, l'unique choix est $\alpha = 1/(2 \ln 2)$, qui est positif et satisfait toutes les conditions. C'est donc bien la valeur cherchée.

Problème

Partie I

1. La fonction $f(x) = x - \ln x$ est de classe C^1 sur $]0, +\infty[$ et $f'(x) = 1 - 1/x = (x-1)/x$. La dérivée est négative sur $]0, 1[$, nulle en 1 et positive sur $]1, +\infty[$. Ainsi f décroît strictement puis croît strictement et admet le minimum $f(1) = 1$.

Quand $x \rightarrow 0^+$, $x \rightarrow 0$ et $-\ln x \rightarrow +\infty$, donc $f(x) \rightarrow +\infty$. Quand $x \rightarrow +\infty$, les croissances comparées donnent $\ln x/x \rightarrow 0$, d'où $f(x) = x(1 - \ln x/x) \rightarrow +\infty$.

x	0		1		$+\infty$
$f'(x)$		—	0	+	
$f(x)$	$+\infty$	$\searrow$	1	$\nearrow$	$+\infty$

En particulier, $f(x) \geq 1$ pour tout $x > 0$, avec égalité si et seulement si $x = 1$.

2. On a $f(x) - x = -\ln x$. Le logarithme étant strictement croissant et nul en 1, cette différence est strictement positive pour $0 < x < 1$, nulle pour $x = 1$ et strictement négative pour $x > 1$.

Partie II

Code adapté en Python. Le nom `resultat` désigne une variable locale, créée à chaque appel, dont la valeur est renvoyée par `return`. Elle permet de conserver le comptage de l'affectation interne de l'ancien algorithme. Voici le code complet correspondant à l'énoncé :

Python

```
import numpy as np

def f(x):
    if x > 0:
        resultat = x - np.log(x)
    return resultat
```

```
n = int(input("n = "))
a = float(input("a = "))
u = a
p = a
for k in range(1, n + 1):
    u = f(u)
    p = p * u
print(u, p)
```

Après k tours, u contient u_k et p contient le produit $u_0 u_1 \cdots u_k$. Pour $n = 0$, aucun tour n'est exécuté, et les valeurs affichées sont toutes deux a . Les calculs mathématiques qui suivent portent sur $a > 0$; pour un argument non positif, le code ne définit pas de valeur réelle de f utilisable par la suite du calcul.

1. Pour $a = 2$, on applique successivement $u_{k+1} = u_k - \ln u_k$ et $p_{k+1} = p_k u_{k+1}$, à partir de $u_0 = p_0 = 2$. Les valeurs numériques sont :

k	u_k	p_k
0	2	2
1	1,306852819440	2,613705638880
2	1,039231000596	2,716243926356
3	1,000749983454	2,718281064358

Les approximations par défaut à 10^{-4} près sont donc $u_3 \simeq 1,0007$ et $p_3 \simeq 2,7182$. En effet, $1,0007 \leq u_3 < 1,0008$ et $2,7182 \leq p_3 < 2,7183$.

Code exécuté avec les saisies $n = 3$, $a = 2$; les valeurs affichées concordent avec ce tableau. Les valeurs décimales illustrent le calcul et ne remplacent pas les preuves des questions suivantes.

2. a) Pour que $u_1 = f(a)$ soit défini comme réel, il faut $a > 0$, puisque $\ln a$ intervient. Cette condition est suffisante : d'après la partie I, $f(x) \geq 1 > 0$ pour tout $x > 0$. Une récurrence montre donc que, à partir de $a > 0$, tous les termes existent et restent strictement positifs. La suite entière est définie si et seulement si $a > 0$. Le fait que le code n'effectue aucun appel pour $n = 0$ ne permet pas de définir la suite à tous les indices lorsque $a \leq 0$.
2. b) Question adaptée au programme actuel : le comptage a été précisé pour Python. La variable locale `resultat` correspond à l'affectation du résultat de fonction dans le code d'origine ; `return` n'est pas une affectation.

On applique exactement la convention de l'énoncé. La boucle possède n tours. Chaque tour appelle une fois `f`, qui évalue une fois `np.log`, effectue une soustraction et une affectation à `resultat`. Il effectue ensuite une affectation à `u`, une multiplication et une affectation à `p`. Les deux initialisations de `u` et `p` s'ajoutent à ces affectations.

Le bilan est donc : n appels à `f`, n évaluations de `np.log`, n soustractions, n multiplications et $3n + 2$ affectations. Si l'on regroupe les deux fonctions de calcul `f` et `np.log`, cela représente $2n$ appels au total. Les opérations des entrées-sorties et de la boucle ne font pas partie de ce comptage, conformément à la consigne. Pour $n = 0$, il reste seulement les deux affectations initiales.

3. Pour tout entier naturel n et pour $a > 0$, la mise à jour de `u` donne exactement $u_{n+1} = f(u_n) = u_n - \ln(u_n)$, avec $u_0 = a$.
4. Si la suite est constante, $u_1 = u_0 = a$, donc $a - \ln a = a$ et $\ln a = 0$, c'est-à-dire $a = 1$. Réciproquement, $f(1) = 1$ assure par récurrence que tous les termes valent 1 lorsque $a = 1$. C'est donc l'unique valeur recherchée.
5. On suppose $a > 1$.
5. a) La propriété $u_0 > 1$ est l'hypothèse. Supposons $u_n > 1$. La fonction f est strictement croissante sur $[1, +\infty[$ d'après I.1 ; ainsi $u_{n+1} = f(u_n) > f(1) = 1$. La récurrence donne $u_n > 1$ pour tout $n \in \mathbb{N}$.

5. b) Pour tout n , $u_n > 1$, donc $\ln u_n > 0$. Par la relation de II.3, $u_{n+1} - u_n = -\ln u_n < 0$.
La suite est strictement décroissante.

5. c) Méthode 1 — Limite monotone et point fixe.

La suite est décroissante et minorée par 1, donc elle converge vers un réel $\ell \geq 1$. La fonction f est continue en $\ell > 0$, ce qui permet de passer à la limite dans $u_{n+1} = f(u_n)$. On obtient $\ell = \ell - \ln \ell$, donc $\ln \ell = 0$ et $\ell = 1$.

Méthode 2 — Inégalité des accroissements finis.

On sait que $u_n \in [1, a]$. Sur cet intervalle, $0 \leq f'(x) = 1 - 1/x \leq 1 - 1/a = q$, où $0 < q < 1$. L'inégalité des accroissements finis, appliquée entre 1 et u_n , donne

$$0 < u_{n+1} - 1 = |f(u_n) - f(1)| \leq q|u_n - 1|.$$

Une récurrence conduit à $0 < u_n - 1 \leq q^n(a - 1)$. Comme $q^n \rightarrow 0$, l'encadrement prouve $u_n \rightarrow 1$. La première méthode identifie la limite après avoir établi son existence ; la seconde fournit en plus une majoration de l'erreur.

6. On suppose $0 < a < 1$.

6. a) Sur $]0, 1]$, f est strictement décroissante et atteint la valeur 1 uniquement en 1.
Puisque $a < 1$, $u_1 = f(a) > f(1) = 1$.

6. b) La suite décalée $(u_{n+1})_{n \geq 0}$ suit la même relation de récurrence et son premier terme est $u_1 > 1$. Les résultats de II.5 s'appliquent donc à cette suite : elle est strictement décroissante et converge vers 1. Retirer le premier terme ne modifie pas la limite, donc $u_n \rightarrow 1$ également. La suite initiale n'est pas décroissante dès le rang 0, car $u_1 > 1 > a = u_0$.

Partie III

1. Pour $a > 0$, tous les termes u_k sont positifs. L'initialisation et la boucle donnent

$$p_n = \prod_{k=0}^n u_k > 0,$$

ce qui autorise le logarithme. Par II.3, $\ln u_k = u_k - u_{k+1}$. Par conséquent,

$$\ln p_n = \sum_{k=0}^n \ln u_k = \sum_{k=0}^n (u_k - u_{k+1}) = u_0 - u_{n+1} = a - u_{n+1}.$$

La somme se télescope ; on en déduit $p_n = \exp(a - u_{n+1})$. Dans les trois cas $a > 1$, $0 < a < 1$ et $a = 1$, la partie II donne $u_n \rightarrow 1$. La continuité de l'exponentielle implique donc

$$\lim_{n \rightarrow +\infty} p_n = e^{a-1}.$$

Pour $a = 1$, tous les produits valent 1, en accord avec cette formule.

2. Code adapté en Python ; la restriction « aucune multiplication » est conservée. Pour calculer p_n , il suffit de calculer u_{n+1} , donc de réaliser $n + 1$ mises à jour, puis d'appliquer la formule précédente.

Python

```
import numpy as np

n = int(input("n = "))
a = float(input("a = "))
u = a
for k in range(n + 1):
    u = u - np.log(u)
```

```
p = np.exp(a - u)
print(p)
```

On suppose $n \in \mathbb{N}$ et $a > 0$. $\text{range}(n + 1)$ comporte $n + 1$ valeurs, de 0 à n ; à la sortie, u contient donc u_{n+1} . Le code ne contient aucune multiplication et renvoie bien p_n dans le modèle mathématique exact. Pour $n = 0$, il donne $\exp(a - u_1) = \exp(\ln a) = a = p_0$.

Code exécuté pour $a \in \{0,2; 0,8; 1; 2; 10\}$ et $n \in \{0; 1; 2; 3; 10; 30\}$. Les résultats ont été comparés à ceux de l'algorithme de II : ils coïncident à la précision des calculs flottants. Pour $n = 3$, $a = 2$, on obtient environ 2,718281064358. Le contrôle numérique vérifie les indices et l'implémentation; l'identité exacte est démontrée en III.1.

Partie IV

1. Sous l'hypothèse $a > 1$, la partie II assure $1 < u_{n+1} < u_n$ pour tout n . Le segment $[u_{n+1}, u_n]$ est donc inclus dans $]0, +\infty[$, où f est continue. L'intégrale sur ce segment existe et est finie. La suite (v_n) est donc bien définie.

2. a) Comme f est croissante sur $[1, +\infty[$, pour $t \in [u_{n+1}, u_n]$ on a

$$f(u_{n+1}) \leq f(t) \leq f(u_n), \quad \text{soit} \quad u_{n+2} \leq f(t) \leq u_{n+1}.$$

En intégrant sur le segment, de longueur positive $u_n - u_{n+1}$, il vient

$$u_{n+2}(u_n - u_{n+1}) \leq v_n \leq u_{n+1}(u_n - u_{n+1}).$$

La relation de récurrence donne $u_n - u_{n+1} = \ln u_n$, ce qui conduit à l'encadrement demandé :

$$u_{n+2} \ln(u_n) \leq v_n \leq u_{n+1} \ln(u_n).$$

2. b) Pour tout n , $u_n > 1$, donc $\ln u_n > 0$ et l'on peut diviser l'encadrement précédent sans changer le sens des inégalités :

$$u_{n+2} \leq \frac{v_n}{\ln u_n} \leq u_{n+1}.$$

Les deux suites qui encadrent le quotient tendent vers 1. Par le théorème d'encadrement, $v_n / \ln u_n \rightarrow 1$. C'est précisément $v_n \sim \ln u_n$ lorsque $n \rightarrow +\infty$.

Relevé des adaptations

- Exercice 1, 3. d) : correction du renvoi « deuxième question » en « question 1. a) », où d_5 est effectivement calculé.
- Problème, partie II, algorithme et questions associées : Code adapté en Python, avec documentation de `np.log`, des entrées-sorties et des bornes de range. Une variable locale `resultat` conserve l'affectation interne lors de l'évaluation de f ; le domaine mathématique reste $a > 0$.
- Problème, II.2. b) : conventions de comptage explicitées pour Python, en excluant les liaisons de paramètres et de boucle, en conservant les deux initialisations et l'affectation locale, et en distinguant les appels à f des évaluations de `np.log`. La consigne de compter appels, soustractions, multiplications et affectations est conservée.
- Problème, III.2 : Code adapté en Python, avec documentation de `np.exp`. La restriction imposant un algorithme sans multiplication est conservée.