



PRÉPASIMPLE

prepasimple.fr

ESSEC

Mathématiques 2 — Option économique

1990

Mathématiques appliquées

Mathématiques appliquées — Prépa ECG

Annales de concours · Corrigé

Partie I

1. a) La fonction f est la restriction à $[0, +\infty[$ d'une fonction de classe C^∞ sur $] -1, +\infty[$. En dérivant chaque terme et en réduisant au même dénominateur,

$$f'(x) = \frac{1}{2} + \frac{1}{2(1+x)^2} - \frac{1}{1+x} = \frac{(1+x)^2 + 1 - 2(1+x)}{2(1+x)^2} = \frac{x^2}{2(1+x)^2}.$$

Ainsi $f'(0) = 0$ et $f'(x) > 0$ pour $x > 0$. La fonction f est strictement croissante sur $[0, +\infty[$, à partir de $f(0) = 0$. Pour sa limite,

$$\frac{f(x)}{x} = \frac{1}{2} + \frac{1}{2(1+x)} - \frac{\ln(1+x)}{x} \rightarrow \frac{1}{2}.$$

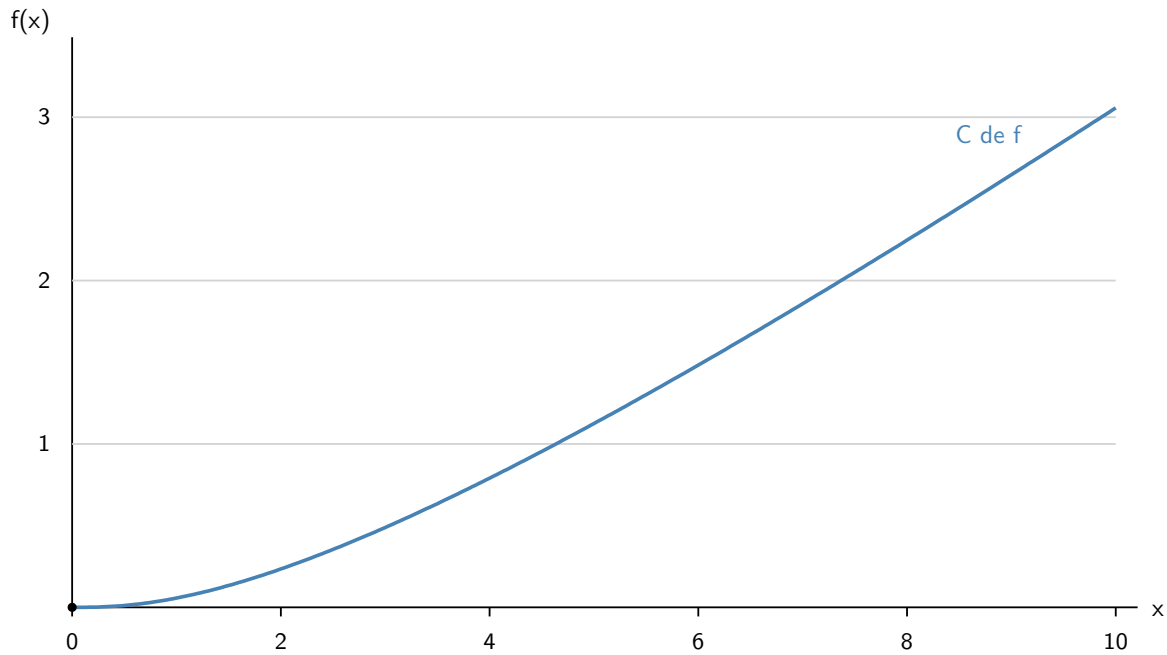
La dernière limite utilise les croissances comparées du logarithme et des puissances. Elle montre que $f(x) \rightarrow +\infty$. Le tableau de variation est donc

x	0	$+\infty$
$f'(x)$	0	+
$f(x)$	0	$\nearrow +\infty$

Pour préciser l'allure, on dérive encore :

$$f''(x) = \frac{x}{(1+x)^3} \geq 0.$$

La courbe est convexe sur $[0, +\infty[$, avec une tangente horizontale à l'origine. Elle monte avec une pente qui tend vers $1/2$. Le dessin suivant est calculé à partir de la formule de f ; l'intervalle représenté est $[0, 10]$.



b) Comme $(1+x)^2 \geq 1$ pour $x \geq 0$, la formule de a) donne

$$0 \leq f'(x) = \frac{x^2}{2(1+x)^2} \leq \frac{x^2}{2}.$$

La fonction f' est continue sur tout segment $[0, x]$. On intègre cette inégalité sur ce segment et on utilise $f(0) = 0$:

$$0 \leq f(x) - f(0) = \int_0^x f'(t) dt \leq \int_0^x \frac{t^2}{2} dt = \frac{x^3}{6}.$$

Les deux encadrements de (1) sont donc établis, y compris pour $x = 0$.

2. a) Question adaptée au programme actuel : la méthode imposée est remplacée par un calcul algébrique particulier.

Méthode 1 — Réduire au même dénominateur.

Pour $k \geq 2$, tous les dénominateurs suivants sont strictement positifs et

$$\frac{1}{2(k-1)^2} - \frac{1}{2k^2} - \frac{1}{k^3} = \frac{k(2k-1) - 2(k-1)^2}{2k^3(k-1)^2} = \frac{3k-2}{2k^3(k-1)^2} > 0.$$

En ajoutant $1/k^3$, on obtient l'inégalité (3).

Méthode 2 — Comparer une intégrale à une constante.

Sur $[k-1, k]$, la fonction $t \mapsto 1/t^3$ est continue et décroissante. On a donc $1/t^3 \geq 1/k^3$.

L'intégration sur un segment de longueur 1 donne

$$\frac{1}{k^3} \leq \int_{k-1}^k \frac{dt}{t^3} = \left[-\frac{1}{2t^2} \right]_{k-1}^k = \frac{1}{2(k-1)^2} - \frac{1}{2k^2}.$$

Le calcul algébrique vérifie directement les constantes; la comparaison intégrale explique la forme télescopique du majorant, utile pour la question suivante.

b) Pour chaque $k \geq n \geq 2$, I.1.b, appliquée à $x = 1/k$, puis I.2.a donnent

$$0 \leq f(1/k) \leq \frac{1}{6k^3} \leq \frac{1}{12} \left(\frac{1}{(k-1)^2} - \frac{1}{k^2} \right).$$

On somme ces inégalités pour $k = n, n+1, \dots, n+p-1$. La somme de droite se simplifie par annulations des termes intermédiaires :

$$0 \leq \sum_{k=n}^{n+p-1} f(1/k) \leq \frac{1}{12} \left(\frac{1}{(n-1)^2} - \frac{1}{(n+p-1)^2} \right) \leq \frac{1}{12(n-1)^2}.$$

Il s'agit de sommes finies; aucun échange de limite et de somme n'est utilisé. Cela prouve (4).

3. a) Pour $t \in [n, n+1]$, avec $n \geq 1$, on a

$$\frac{1}{n+1} \leq \frac{1}{t} \leq \frac{1}{n}.$$

La fonction $t \mapsto 1/t$ est continue sur ce segment. On intègre et on utilise la primitive $\ln t$:

$$\frac{1}{n+1} \leq \int_n^{n+1} \frac{dt}{t} = \ln(n+1) - \ln n \leq \frac{1}{n}.$$

Cet encadrement est obtenu par intégration d'inégalités, avec des outils du programme.

b) Les différences consécutives s'écrivent

$$u_{n+1} - u_n = \frac{1}{n+1} - (\ln(n+1) - \ln n) \leq 0,$$

$$v_{n+1} - v_n = u_{n+1} - u_n - \frac{1}{n+1} + \frac{1}{n} = \frac{1}{n} - (\ln(n+1) - \ln n) \geq 0.$$

Ainsi (u_n) est décroissante et (v_n) est croissante. De plus,

$$u_n - v_n = \frac{1}{n} \geq 0, \quad u_n - v_n \longrightarrow 0.$$

Ces suites sont donc adjacentes. Pour expliciter également leurs bornes, $v_1 = 0$, $u_1 = 1$, et la monotonie donne

$$0 \leq v_n \leq u_n \leq 1.$$

Le théorème des suites adjacentes assure leur convergence vers une même limite réelle γ . Pour tout $n \geq 1$,

$$v_n \leq \gamma \leq u_n.$$

4. a) La moyenne $m_n = (u_n + v_n)/2$ est le milieu du segment $[v_n, u_n]$, qui contient γ . La distance de son milieu à l'un quelconque de ses points est au plus la demi-longueur du segment. Ainsi,

$$|m_n - \gamma| \leq \frac{u_n - v_n}{2} = \frac{1}{2n}.$$

On peut aussi écrire directement $v_n = m_n - 1/(2n)$ et $u_n = m_n + 1/(2n)$; l'encadrement de I.3.b donne la même conclusion.

b) Code adapté en Python. En posant

$$H_n = \sum_{k=1}^n 1/k,$$

les définitions donnent

$$m_n = H_n - \ln n - \frac{1}{2n}.$$

Le programme complet suivant accumule H_n puis renvoie cette expression. L'entier n est supposé au moins égal à 1.

Python

```
import numpy as np

def moyenne(n):
    h = 0.0
    for k in range(1, n + 1):
        h = h + 1 / k
    return h - np.log(n) - 1 / (2 * n)

print(moyenne(5))
print(moyenne(50))
```

Après le passage d'indice k , la variable h vaut la somme $1 + 1/2 + \dots + 1/k$. La borne exclusive de `range` fait inclure le terme $1/n$. La fonction renvoie un réel approchant m_n ; les deux appels l'affichent pour les indices demandés. Ce code a été exécuté. Il donne, à l'arrondi indiqué,

$$m_5 \simeq 0,573895420899233, \quad m_{50} \simeq 0,577182332901279.$$

Les garanties mathématiques de I.4.a sont

$$|m_5 - \gamma| \leq 0,1, \quad |m_{50} - \gamma| \leq 0,01.$$

On peut donc prendre m_5 comme approximation à 0,1 près et m_{50} comme approximation à 0,01 près. Les décimales affichées ne signifient pas que toutes ces décimales de γ sont déjà garanties.

Les deux valeurs numériques ne diffèrent que d'environ 0,003286912002046. A posteriori, la majoration triangulaire

$$|\gamma - m_5| \leq |\gamma - m_{50}| + |m_{50} - m_5| \leq 0,01 + |m_{50} - m_5|$$

donne une erreur inférieure à 0,013287, déjà nettement plus petite que 0,1. L'encadrement plus fin de I.5.b établira même que l'erreur de m_5 est inférieure à 0,003322.

5. a) On remplace m_k par l'expression obtenue en I.4.b :

$$m_{k+1} - m_k = \frac{1}{k+1} - \ln\left(1 + \frac{1}{k}\right) - \frac{1}{2(k+1)} + \frac{1}{2k}.$$

En regroupant les termes rationnels,

$$m_{k+1} - m_k = \frac{1}{2k} + \frac{1}{2(k+1)} - \ln\left(1 + \frac{1}{k}\right) = f\left(\frac{1}{k}\right).$$

L'égalité demandée est donc exacte pour tout $k \geq 1$. Comme f est positive sur $[0, +\infty[$, la suite (m_k) est croissante.

b) Question adaptée au programme actuel : le renvoi à la valeur de m_{50} est rectifié en I.4.b.

Pour $n \geq 2$ et $p \geq 1$, on somme l'égalité de a) ; la somme est télescopique :

$$m_{n+p} - m_n = \sum_{k=n}^{n+p-1} f(1/k).$$

L'inégalité (4) donne donc

$$0 \leq m_{n+p} - m_n \leq \frac{1}{12(n-1)^2}.$$

Puisque $m_j = (u_j + v_j)/2 \rightarrow \gamma$, on fait tendre p vers $+\infty$, pour n fixé. La compatibilité des limites avec l'ordre donne

$$0 \leq \gamma - m_n \leq \frac{1}{12(n-1)^2}, \quad m_n \leq \gamma \leq m_n + \frac{1}{12(n-1)^2}.$$

Pour $n = 50$, l'encadrement exact issu de cette méthode est

$$m_{50} \leq \gamma \leq m_{50} + \frac{1}{28812}.$$

Avec la valeur de I.4.b et des bornes décimales arrondies vers l'extérieur, il donne

$$0,57718233290 \leq \gamma \leq 0,57721704067.$$

La largeur de cet intervalle est d'environ 0,000034707761, bien plus petite que la largeur 0,02 de l'encadrement symétrique de I.4.a pour $n = 50$. En soustrayant m_5 aux deux bornes exactes, on obtient aussi

$$0,0032869 \leq \gamma - m_5 \leq 0,0033217.$$

Cela confirme la bonne précision de m_5 , malgré la borne initiale 0,1.

Partie II

Les tirages successifs sans remise rendent équiprobables les suites ordonnées de numéros distincts. En effet, une suite donnée de longueur p a pour probabilité

$$\frac{1}{n} \times \frac{1}{n-1} \times \dots \times \frac{1}{n-p+1} = \frac{(n-p)!}{n!}.$$

Les variables considérées ont un nombre fini de valeurs ; elles admettent donc toutes une espérance. On utilisera la linéarité sans supposer d'indépendance entre les tirages.

1. a) Par l'interprétation du coefficient binomial, le nombre de parties de N_n à p éléments est

$$\binom{n}{p} = \frac{n!}{p!(n-p)!}.$$

- b)** Une suite de longueur p composée d'éléments distincts d'un ensemble A à p éléments utilise chacun de ses éléments exactement une fois. C'est un ordre de ses p éléments. Le premier terme a p choix possibles, le deuxième $p - 1$, puis ainsi de suite. Le nombre de suites est donc

$$p(p - 1) \cdots 1 = p!.$$

- c)** On choisit d'abord l'ensemble A des p numéros, puis leur ordre. Chaque suite détermine de façon unique ces deux choix : il n'y a pas de double comptage. D'après a) et b), leur nombre est

$$\binom{n}{p} p! = \frac{n!}{(n - p)!}.$$

Il coïncide avec le produit $n(n - 1) \cdots (n - p + 1)$ obtenu en choisissant successivement chaque numéro parmi ceux qui restent.

- d)** L'ensemble A possède un unique plus grand élément. On impose cet élément en dernière position ; il reste à ordonner librement les $p - 1$ autres éléments. Le nombre recherché est

$$(p - 1)!.$$

- e)** On choisit l'ensemble A de p éléments de N_n , puis une disposition comptée en d). De nouveau, ces choix sont déterminés par la suite finale. Le nombre de suites favorables est donc

$$\binom{n}{p} (p - 1)! = \frac{n!}{p(n - p)!}.$$

- f)** Méthode 1 — Utiliser les dénombrements précédents.

L'événement $[Z_p > X_{p-1}]$ signifie que le plus grand numéro parmi les p premiers est en dernière position. Il a les suites de e) pour issues favorables, parmi les suites

équiprobables de c). Par conséquent,

$$P(Z_p > X_{p-1}) = \frac{\binom{n}{p} (p-1)!}{\binom{n}{p} p!} = \frac{1}{p}.$$

Les dénominateurs sont non nuls, puisque $2 \leq p \leq n$.

Méthode 2 — Utiliser la symétrie des positions.

Le plus grand des p premiers numéros occupe une seule position parmi $1, \dots, p$. Pour deux positions données, permuter ces positions définit une bijection entre les suites dont le maximum est à l'une et celles dont il est à l'autre. Les suites étant équiprobables, ces p événements ont la même probabilité. Ils sont deux à deux incompatibles et leur réunion est certaine ; chacun a donc une probabilité $1/p$.

En particulier, le maximum est en position p avec probabilité $1/p$, ce qui donne le même résultat. La première méthode suit les comptages demandés ; la seconde évite les factorielles lorsque seule la probabilité d'un rang du maximum est utile.

2. La variable B_p est l'indicatrice de l'événement $[Z_p > X_{p-1}]$. D'après II.1.f, elle suit une loi de Bernoulli de paramètre $1/p$, donc

$$E(B_p) = P(Z_p > X_{p-1}) = \frac{1}{p}.$$

Par linéarité de l'espérance d'une somme finie,

$$E(B_2 + B_3 + \dots + B_n) = \sum_{p=2}^n \frac{1}{p} = H_n - 1 = \ln n + u_n - 1,$$

où

$$H_n = \sum_{p=1}^n 1/p.$$

Cette variable compte le nombre de nouveaux maxima rencontrés après le premier tirage : chaque terme vaut 1 exactement quand le numéro nouvellement tiré dépasse tous les précédents. Son espérance est donc le nombre moyen de mises à jour du maximum après son initialisation. Aucune indépendance entre les B_p n'est nécessaire.

Si $n = 1$, la somme est vide et vaut 0 ; la formule $H_1 - 1 = 0$ reste correcte.

3. a) Code adapté en Python. Les indices de liste sont décalés d'une unité par rapport aux indices mathématiques.

Lorsque la liste est croissante, X reçoit d'abord 1. Chaque valeur suivante est supérieure à la valeur courante ; les valeurs reçues par X sont successivement

1, 2, 3, ..., n.

Lorsque la liste est décroissante, X reçoit d'abord n. Toutes les comparaisons suivantes sont fausses : X conserve n pendant toute la boucle. La seule valeur qu'il reçoit est donc n.

Voici le code complet sous forme de fonction, avec les deux exemples. La fonction renvoie la valeur finale du maximum.

Python

```
def maximum(Z):
    n = len(Z)
    X = Z[0]
    for p in range(1, n):
        if Z[p] > X:
            X = Z[p]
    return X

print(maximum([1, 2, 3, 4, 5]))
print(maximum([5, 4, 3, 2, 1]))
```

Ce code a été exécuté et affiche 5 dans les deux cas. Les instructions $n = \text{len}(Z)$ et `return X` servent à la présentation en fonction ; conformément à l'énoncé, seules les affectations à X entrent dans le coût étudié.

b) Après l'initialisation, X est le maximum du premier élément. Si X est le maximum des éléments déjà lus, la comparaison avec le suivant laisse X inchangé lorsque cet élément ne le dépasse pas, et remplace X par cet élément dans le cas contraire. Dans les deux cas, X devient le maximum de tous les éléments lus jusque-là. Une récurrence sur le nombre d'éléments lus prouve donc que la valeur finale est

$$X = \max(Z_1, \dots, Z_n) = n.$$

La boucle comporte exactement $n - 1$ passages et une comparaison $>$ par passage : il y a donc $n - 1$ comparaisons, quel que soit l'ordre des données.

Il y a une affectation initiale à X , puis au plus une par passage. Le nombre d'affectations est donc compris entre 1 et n . Les deux bornes sont atteintes :

- une liste décroissante, ou plus généralement une liste commençant par n , donne une seule affectation ;
- une liste strictement croissante donne une affectation à chaque passage, soit n au total.

Le minimum est donc 1 et le maximum n . Les affectations à la variable de boucle ne sont pas comptées.

c) Méthode 1 — Compter les records selon leur position.

On note A_n le nombre aléatoire d'affectations à X lorsque l'ordre de la liste est celui des tirages aléatoires du préambule. L'initialisation compte pour 1. Pour chaque tirage de rang $p \geq 2$, une nouvelle affectation a lieu exactement lorsque $Z_p > X_{p-1}$. Ainsi, pour chaque issue,

$$A_n = 1 + \sum_{p=2}^n B_p.$$

En prenant les espérances et en utilisant II.2,

$$E_n = E(A_n) = 1 + \sum_{p=2}^n \frac{1}{p} = H_n = \ln n + u_n.$$

Méthode 2 — Compter les records selon leur valeur.

Pour $k \in \{1, \dots, n\}$, on note C_k l'indicatrice de l'événement «la valeur k provoque une affectation à X », l'initialisation étant incluse. Cette valeur provoque une affectation si et seulement si elle apparaît avant toutes les valeurs strictement supérieures à elle : sinon un maximum au moins égal à l'une de ces valeurs est déjà mémorisé ; réciproquement, si aucune valeur plus grande n'est encore apparue, k dépasse tous les éléments précédents ou constitue le premier élément.

Par symétrie de l'ordre uniforme, chacun des $n - k + 1$ éléments de $\{k, k + 1, \dots, n\}$ a la même probabilité d'apparaître le premier parmi eux. Ces possibilités sont incompatibles et exhaustives, donc

$$E(C_k) = P(C_k = 1) = \frac{1}{n - k + 1}.$$

Chaque affectation correspond à une seule valeur ; on a donc

$$A_n = \sum_{k=1}^n C_k.$$

La linéarité puis le changement d'indice $j = n - k + 1$ donnent

$$E_n = \sum_{k=1}^n \frac{1}{n - k + 1} = \sum_{j=1}^n \frac{1}{j} = H_n.$$

Le premier comptage est adapté au parcours de l'algorithme ; le second explique le rôle des grandes valeurs dans la formation des records.

Enfin, I.3.b assure $u_n \rightarrow \gamma$. En posant

$$a = 1, \quad b = \gamma, \quad \varepsilon_n = u_n - \gamma,$$

on a $\varepsilon_n \rightarrow 0$ et

$$E_n = \ln n + \gamma + \varepsilon_n.$$

L'encadrement $v_n \leq \gamma \leq u_n$ donne même $0 \leq \varepsilon_n \leq 1/n$. Le nombre moyen d'affectations croît donc comme $\ln n$, tandis que le nombre de comparaisons vaut toujours $n - 1$.

Relevé des adaptations

- I.2.a : remplacement de la méthode imposant l'égalité des accroissements finis par un calcul algébrique guidé, avec une seconde preuve particulière par intégration d'une inégalité. L'inégalité demandée et son emploi en I.2.b sont conservés.
- I.4.b : Code adapté en Python ; syntaxe et rôle du logarithme numérique documentés dans le sujet. L'algorithme calcule la même moyenne m_n .
- I.5.b : rectification du renvoi à la « question 5 » en « question 4.b », où m_{50} est calculé.
- Préambule de II : équiprobabilité de chaque boule restante explicitée pour rendre le modèle de tirage sans remise précis ; aucune indépendance entre tirages n'est ajoutée.
- II.2 : convention de somme vide précisée pour le cas $n = 1$ déjà autorisé par le préambule.
- II.3.a–c : Code adapté en Python, avec décalage des indices des listes. Le coût est explicitement le nombre d'affectations à X , initialisation incluse et contrôle de boucle exclu, ce qui conserve la quantité modélisée par les records et l'espérance demandée.