



PRÉPASIMPLE

prepasimple.fr

HEC

Mathématiques 1

1992

Corrigé de mathématiques approfondies

Mathématiques approfondies — Prépa ECG

Annales de concours · Corrigé

Dans tout le corrigé, $N \geq 1$, $\theta \in [0, \frac{\pi}{4}[$, f est la restriction de la tangente à $] -\frac{\pi}{2}, \frac{\pi}{2}[$ et $g = f^{-1}$. Les soustractions sont comptées comme des additions. Les valeurs α_k sont supposées connues conformément à la convention de l'énoncé.

Partie I

1. Étude de la fonction tangente

(a) Variations, courbe et tangentes

Sur $] -\frac{\pi}{2}, \frac{\pi}{2}[$, le cosinus est strictement positif. La fonction tangente est donc de classe C^∞ et :

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

Ainsi, f est strictement croissante. Les limites du sinus et du cosinus aux extrémités donnent :

$$\lim_{x \rightarrow -\pi/2^+} \tan x = -\infty, \quad \tan 0 = 0, \quad \lim_{x \rightarrow \pi/2^-} \tan x = +\infty.$$

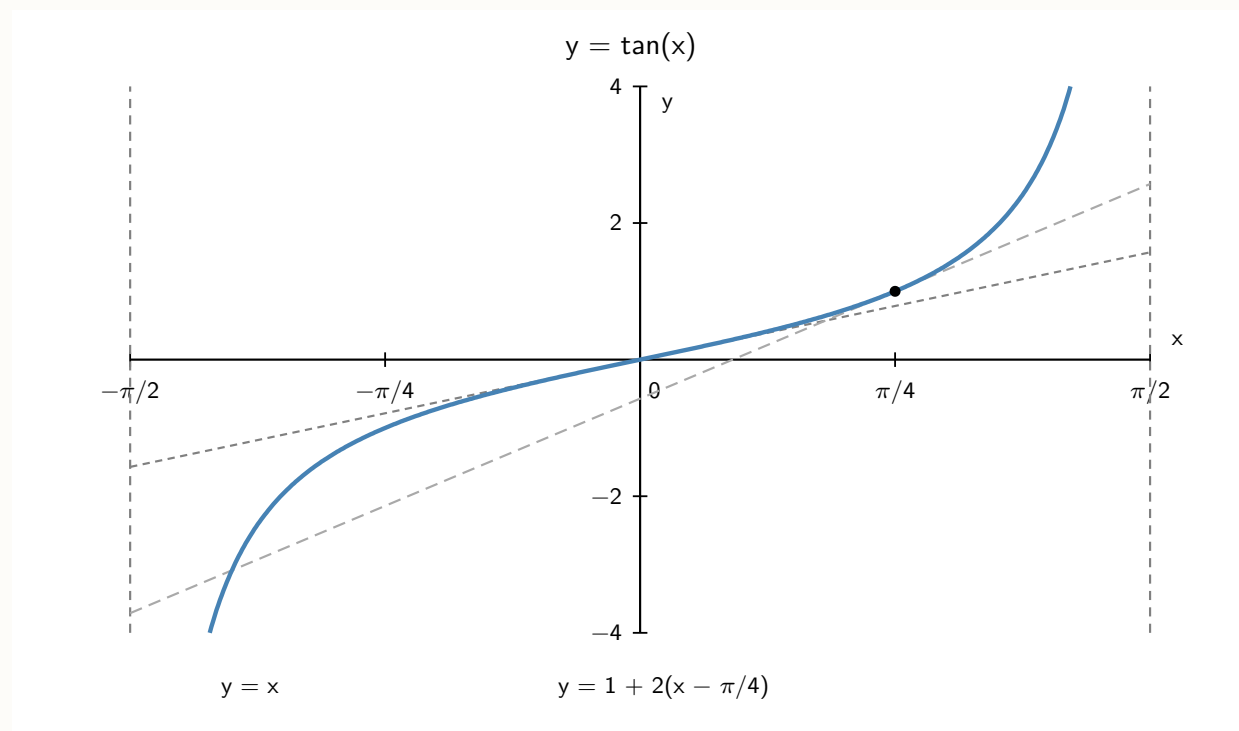
x	$-\pi/2$	0	$\pi/2$
$f'(x)$		+	
$f(x)$	$-\infty$	0	$+\infty$
Variation		$\nearrow$	

La courbe est symétrique par rapport à l'origine, car la tangente est impaire, et admet les asymptotes verticales $x = -\frac{\pi}{2}$ et $x = \frac{\pi}{2}$.

De plus, $f''(x) = 2 \tan x (1 + \tan^2 x)$: la courbe est concave sur $] -\frac{\pi}{2}, 0[$ et convexe sur $] 0, \frac{\pi}{2}[$, avec un point d'inflexion à l'origine.

Une tangente en x_0 a pour équation $y = f(x_0) + f'(x_0)(x - x_0)$. Comme $f(0) = 0$, $f'(0) = 1$, $f(\frac{\pi}{4}) = 1$ et $f'(\frac{\pi}{4}) = 2$, les deux tangentes demandées sont :

$$y = x, \quad y = 1 + 2\left(x - \frac{\pi}{4}\right).$$



(b) Réduction à l'intervalle de calcul

On utilise la périodicité, l'imparité et une identité de complémentarité. Les formules d'addition du sinus et du cosinus donnent $\tan(x + \pi) = \tan x$ et $\tan(-x) = -\tan x$. Elles donnent aussi, pour $0 < x < \frac{\pi}{2}$:

$$\sin\left(\frac{\pi}{2} - x\right) = \cos x, \quad \cos\left(\frac{\pi}{2} - x\right) = \sin x,$$

$$\text{puis } \tan x = \frac{1}{\tan(\pi/2 - x)}.$$

Pour un réel θ où la tangente est définie, on peut choisir un entier q tel que $u = \theta - q\pi \in]-\frac{\pi}{2}, \frac{\pi}{2}[$. Par exemple, $q = \left\lfloor \frac{\theta}{\pi} + \frac{1}{2} \right\rfloor$ convient, les valeurs aux pôles étant exclues. On pose $v = |u|$.

Si $v < \frac{\pi}{4}$, on applique directement l'algorithme à v , puis on rétablit le signe de u . Si

$v = \frac{\pi}{4}$, on utilise la valeur exacte $\tan v = 1$. Si $\frac{\pi}{4} < v < \frac{\pi}{2}$, on pose $w = \frac{\pi}{2} - v \in]0, \frac{\pi}{4}[$ et on approche $\tan v$ en inversant une approximation de $\tan w$.

Il faut contrôler l'erreur lors de cette inversion : si $t = \tan w > 0$ et $|\tilde{t} - t| \leq \delta < t/2$, alors $\tilde{t} > t/2$ et :

$$\left| \frac{1}{\tilde{t}} - \frac{1}{t} \right| = \frac{|t - \tilde{t}|}{t\tilde{t}} \leq \frac{2\delta}{t^2}.$$

Pour garantir une erreur finale au plus $\varepsilon > 0$, il suffit donc de prendre une précision initiale assez fine. Sans connaître t , on peut utiliser $t = \tan w \geq w$, car la dérivée de $\tan x - x$ est $\tan^2 x \geq 0$ sur $[0, w]$. Le choix $\delta \leq \min(w/4, \varepsilon w^2/2)$ garantit alors l'erreur requise.

On peut obtenir une telle précision en augmentant le paramètre de l'algorithme. Le calcul approché se ramène donc à l'intervalle annoncé ; la même erreur absolue n'est pas conservée automatiquement par inversion.

2. Étude de la fonction réciproque de la fonction tangente

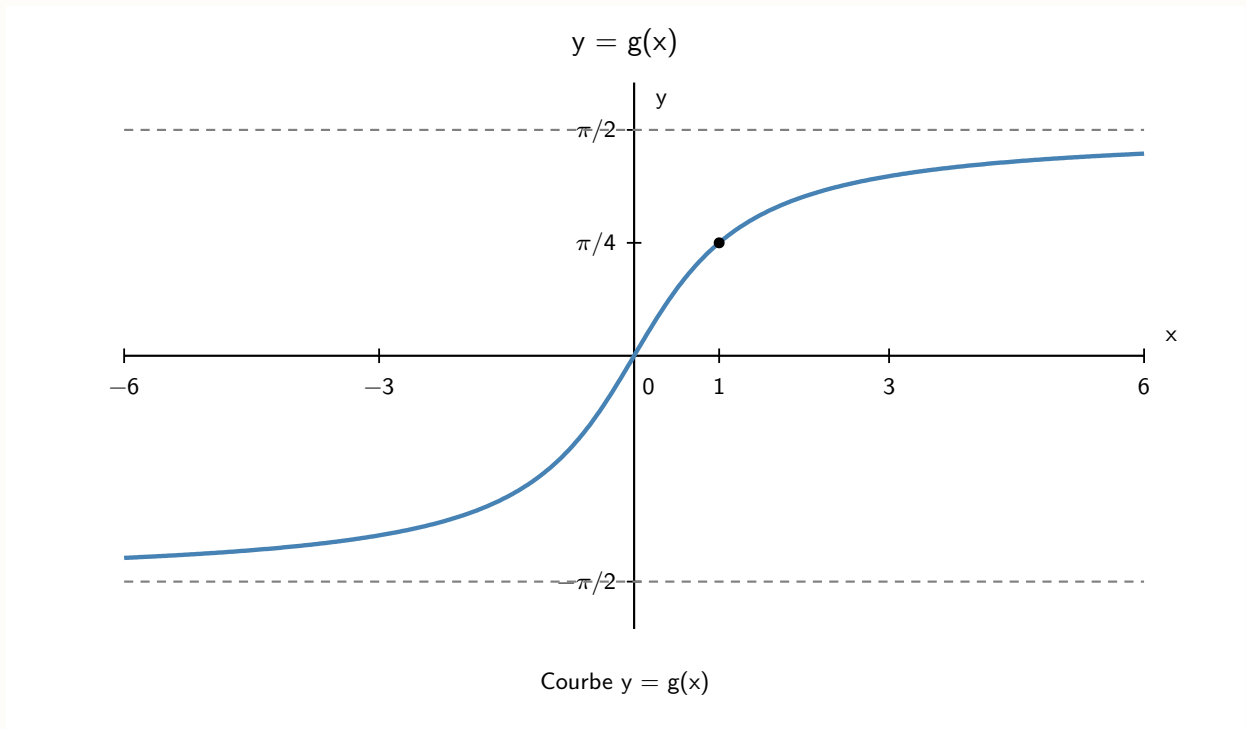
(a) Bijection et allure de la courbe de g

La fonction f est continue et strictement croissante sur $] -\frac{\pi}{2}, \frac{\pi}{2}[$; ses limites aux extrémités sont $-\infty$ et $+\infty$. Le théorème de la bijection assure que son image est $\mathbb{R}$ et qu'elle possède une réciproque $g : \mathbb{R} \rightarrow] -\frac{\pi}{2}, \frac{\pi}{2}[$, continue et strictement croissante.

La relation $f(-x) = -f(x)$ et l'unicité de l'antécédent donnent $g(-t) = -g(t)$. En particulier, $g(0) = 0$ et $g(1) = \frac{\pi}{4}$.

La courbe de g est la symétrique de celle de f par rapport à la droite $y = x$. Pour tout $a < \frac{\pi}{2}$, $t > f(a)$ implique $g(t) > a$; comme $g(t) < \frac{\pi}{2}$, on en déduit $g(t) \rightarrow \frac{\pi}{2}$ lorsque $t \rightarrow +\infty$. L'imparité donne la limite $-\frac{\pi}{2}$ en $-\infty$.

La courbe est donc croissante, passe par l'origine et possède les asymptotes horizontales $y = \pm \frac{\pi}{2}$.



(b) Dérivée et régularité de g

La fonction f est de classe C^1 et sa dérivée ne s'annule jamais. Le théorème de dérivation d'une réciproque s'applique donc sur toute son image $\mathbb{R}$ et donne :

$$g'(t) = \frac{1}{f'(g(t))} = \frac{1}{1 + \tan^2(g(t))} = \frac{1}{1 + t^2}.$$

Cette expression est continue sur $\mathbb{R}$, son dénominateur étant strictement positif. Ainsi $g \in C^1(\mathbb{R})$ avec la dérivée annoncée. En particulier, le théorème fondamental du calcul intégral et $g(0) = 0$ donnent :

$$g(x) = \int_0^x \frac{dt}{1 + t^2}.$$

(c) Identité algébrique finie

On utilise la somme géométrique finie, sans aucune série infinie. Pour $p \in \mathbb{N}$ et $t \in \mathbb{R}$, posons

$$S_p(t) = \sum_{j=0}^p (-1)^j t^{2j}.$$

En développant le produit, les termes intermédiaires s'annulent :

$$(1 + t^2)S_p(t) = 1 + (-1)^p t^{2p+2}.$$

Ainsi :

$$1 = (1 + t^2)S_p(t) + (-1)^{p+1} t^{2p+2}.$$

On peut diviser par $1 + t^2 > 0$ et obtenir :

$$\frac{1}{1 + t^2} = \sum_{j=0}^p (-1)^j t^{2j} + (-1)^{p+1} \frac{t^{2p+2}}{1 + t^2}.$$

Cette identité vaut pour tout réel t , sans restriction sur sa valeur absolue.

(d) Encadrement de $g(x)$

Méthode 1 — Intégration d'un encadrement

Pour $t \geq 0$, $\frac{1}{1 + t^2} = 1 - \frac{t^2}{1 + t^2} \leq 1$ et :

$$\frac{1}{1 + t^2} - (1 - t^2) = \frac{t^4}{1 + t^2} \geq 0.$$

En intégrant les inégalités entre fonctions continues sur $[0, x]$, avec $x \geq 0$, on obtient :

$$\int_0^x (1 - t^2) dt \leq \int_0^x \frac{dt}{1 + t^2} \leq \int_0^x 1 dt,$$

soit $x - \frac{x^3}{3} \leq g(x) \leq x$. Pour $x = 0$, les deux bornes valent 0.

Méthode 2 — Monotonie des différences

Les fonctions $U(x) = x - g(x)$ et $V(x) = g(x) - x + \frac{x^3}{3}$ sont de classe C^1 sur $[0, +\infty[$.
Leurs dérivées sont :

$$U'(x) = \frac{x^2}{1+x^2} \geq 0, \quad V'(x) = \frac{x^4}{1+x^2} \geq 0.$$

Comme $U(0) = V(0) = 0$, leur monotonie donne $U(x) \geq 0$ et $V(x) \geq 0$, donc le même encadrement.

La première méthode prépare le calcul du reste intégral ; la seconde illustre la preuve d'une inégalité par étude de la différence des deux membres.

(e) Développement fini et reste

On intègre l'identité de I.2(c) sur le segment $[0, x]$. Il n'y a qu'une somme finie de fonctions continues, donc la linéarité de l'intégrale s'applique :

$$g(x) = \sum_{j=0}^p (-1)^j \frac{x^{2j+1}}{2j+1} + R_p(x),$$

avec :

$$R_p(x) = (-1)^{p+1} \int_0^x \frac{t^{2p+2}}{1+t^2} dt.$$

Pour $x \geq 0$, $0 \leq \frac{t^{2p+2}}{1+t^2} \leq t^{2p+2}$ sur $[0, x]$. Par conséquent :

$$|R_p(x)| \leq \int_0^x t^{2p+2} dt = \frac{x^{2p+3}}{2p+3}.$$

On a aussi $(-1)^{p+1} R_p(x) \geq 0$. Ainsi les polynômes obtenus avec p pair majorent $g(x)$, et ceux avec p impair le minorent. Ces conclusions valent aussi pour $x = 0$.

3. Construction de la suite (α_n)

(a) Existence, unicité et valeur initiale

La restriction de la tangente à $]0, \frac{\pi}{4}]$ est une bijection strictement croissante sur $]0, 1]$.
 Pour $0 \leq n \leq N$, $10^{-n} \in]0, 1]$ possède donc un unique antécédent dans cet intervalle :

$$\alpha_n = g(10^{-n}).$$

Pour $n = 0$, $\tan(\alpha_0) = 1 = \tan(\frac{\pi}{4})$, donc $\alpha_0 = \frac{\pi}{4}$.

(b) Décroissance stricte

Pour $0 \leq n < N$, $10^{-(n+1)} < 10^{-n}$. La stricte croissance de g donne $\alpha_{n+1} < \alpha_n$. Enfin $\alpha_{N+1} = 0 < \alpha_N$, car $\tan(\alpha_N) = 10^{-N} > 0$.

Ainsi $\alpha_0 > \alpha_1 > \dots > \alpha_N > \alpha_{N+1} = 0$.

Attention : la formule $\alpha_n = g(10^{-n})$ ne vaut que jusqu'à $n = N$; le dernier terme est défini séparément.

(c) Partition de l'intervalle

Soit $x \in [0, \alpha_0[$. L'ensemble des indices $j \in \{1, \dots, N+1\}$ tels que $\alpha_j \leq x$ est non vide, puisque $\alpha_{N+1} = 0 \leq x$. On prend son plus petit élément n .

Si $n = 1$, $x < \alpha_0$ est déjà connu ; si $n > 1$, la minimalité implique $x < \alpha_{n-1}$. Dans tous les cas, $\alpha_n \leq x < \alpha_{n-1}$.

Pour l'unicité, si $r < s$ étaient deux indices convenables, alors $\alpha_{s-1} \leq \alpha_r \leq x$, ce qui contredirait $x < \alpha_{s-1}$.

Les intervalles $[\alpha_n, \alpha_{n-1}[$, pour $1 \leq n \leq N+1$, constituent donc une partition de $[0, \frac{\pi}{4}[$.
 En particulier, $x = 0$ correspond à $n = N+1$.

4. Étude de la suite (α_n)

(a) Comparaison entre deux échelles

Méthode 1 — Différence et dérivée

Posons $F(x) = 10g(x) - g(10x)$ sur $[0, +\infty[$. On a $F(0) = 0$ et, pour $x > 0$:

$$F'(x) = \frac{10}{1+x^2} - \frac{10}{1+100x^2} = \frac{990x^2}{(1+x^2)(1+100x^2)} > 0.$$

Ainsi $F(x) > 0$ pour $x > 0$, donc $g(10x) < 10g(x)$.

Méthode 2 — Comparaison d'intégrales

Dans $g(10x) = \int_0^{10x} \frac{dt}{1+t^2}$, le changement de variable affine $t = 10u$ donne :

$$g(10x) = 10 \int_0^x \frac{du}{1+100u^2}.$$

Pour $u > 0$, $\frac{1}{1+100u^2} < \frac{1}{1+u^2}$. Si $x > 0$, la différence est continue, positive sur $]0, x]$,

et son intégrale est strictement positive. Donc $g(10x) < 10 \int_0^x \frac{du}{1+u^2} = 10g(x)$.

La première méthode étudie directement l'écart ; la seconde fait apparaître la comparaison des intégrandes après changement d'échelle.

Pour $0 \leq n \leq N-1$, on applique le résultat à $x = 10^{-(n+1)} > 0$:

$$\alpha_n = g(10x) < 10g(x) = 10\alpha_{n+1}.$$

(b) Encadrement explicite

On applique I.2(d) à $x = 10^{-n}$, pour $0 \leq n \leq N$:

$$10^{-n} - \frac{10^{-3n}}{3} \leq \alpha_n \leq 10^{-n}.$$

La borne inférieure se factorise en $10^{-n} \left(1 - \frac{1}{3 \cdot 10^{2n}}\right)$, ce qui donne exactement l'encadrement demandé.

(c) Optimalité du coefficient 10

Les bornes précédentes montrent que, lorsque l'indice n tend vers l'infini, les nombres $u_n = g(10^{-n})$ vérifient :

$$1 - \frac{1}{3 \cdot 10^{2n}} \leq \frac{u_n}{10^{-n}} \leq 1.$$

Le théorème d'encadrement donne $\frac{u_n}{10^{-n}} \rightarrow 1$, puis :

$$\frac{u_n}{u_{n+1}} = 10 \frac{u_n/10^{-n}}{u_{n+1}/10^{-(n+1)}} \rightarrow 10.$$

Tous les dénominateurs sont strictement positifs. Pour une constante $c < 10$, cette limite implique $u_n/u_{n+1} > c$ pour certains indices assez grands. En choisissant $N \geq n + 1$, ces deux termes deviennent α_n et α_{n+1} dans le problème correspondant. L'inégalité $\alpha_n < c\alpha_{n+1}$ échoue alors.

Il est donc impossible de remplacer 10 par une constante plus petite valable pour tous les choix de n et de N . On a fait varier N : aucune limite n'est prise dans une suite finie à N fixé.

(d) Minoration du rapport par 9

Pour $1 \leq n \leq N - 1$, I.4(b) donne :

$$\alpha_n \geq 10^{-n} \left(1 - \frac{1}{3 \cdot 10^{2n}}\right) \geq \frac{299}{300} 10^{-n} \geq \frac{9}{10} 10^{-n}.$$

$$\text{D'autre part, } 9\alpha_{n+1} \leq 9 \cdot 10^{-(n+1)} = \frac{9}{10} 10^{-n}. \text{ D'où } 9\alpha_{n+1} \leq \alpha_n.$$

Si $N = 1$, il n'y a aucun indice dans l'intervalle demandé ; l'assertion est alors vide.

(e) Encadrement du premier rapport

On utilise les polynômes et les signes de reste établis en I.2(e), afin de ne pas supposer une valeur décimale de π .

Avec $p = 3$ et $x = 1$, le reste est positif ou nul. Ainsi :

$$\alpha_0 = g(1) \geq 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} = \frac{76}{105} > \frac{7}{10}.$$

Comme $\alpha_1 \leq \frac{1}{10}$, on obtient $7\alpha_1 \leq \frac{7}{10} < \alpha_0$.

Pour la borne supérieure, posons $a = g(1/2)$ et $b = g(1/3)$. Tous deux sont dans $]0, \frac{\pi}{4}[$, donc leur somme est dans $]0, \frac{\pi}{2}[$. La formule d'addition donne :

$$\tan(a + b) = \frac{\frac{1}{2} + \frac{1}{3}}{1 - \frac{1}{2} \cdot \frac{1}{3}} = 1.$$

Par injectivité de la tangente sur cet intervalle, $a + b = g(1) = \alpha_0$. En utilisant $p = 2$, dont le reste est négatif ou nul :

$$g(1/2) \leq \frac{1}{2} - \frac{1}{24} + \frac{1}{160} = \frac{223}{480},$$

$$g(1/3) \leq \frac{1}{3} - \frac{1}{81} + \frac{1}{1215} = \frac{391}{1215}.$$

On en déduit :

$$\alpha_0 \leq \frac{223}{480} + \frac{391}{1215} = \frac{30575}{38880} < \frac{299}{375}.$$

La dernière comparaison est exacte : $299 \cdot 38880 - 30575 \cdot 375 = 159495 > 0$.

Enfin I.4(b), appliquée à $n = 1$, donne $\alpha_1 \geq \frac{1}{10}(1 - \frac{1}{300}) = \frac{299}{3000}$; ainsi

$$8\alpha_1 \geq \frac{299}{375} > \alpha_0.$$

On a donc établi $7\alpha_1 \leq \alpha_0 < 8\alpha_1$, et même la stricte inégalité à gauche.

5. Algorithme de calcul des α_n

Pour $1 \leq n \leq N$, on utilise le polynôme fini :

$$S_p(10^{-n}) = \sum_{j=0}^p \frac{(-1)^j 10^{-n(2j+1)}}{2j+1}.$$

Le résultat de I.2(e) assure :

$$|\alpha_n - S_p(10^{-n})| \leq \frac{10^{-n(2p+3)}}{2p+3}.$$

Soit p le plus petit entier naturel tel que $p \geq \frac{K-3}{2n}$, autrement dit

$p = \max\left(0, \left\lceil \frac{K-3}{2n} \right\rceil\right)$. Alors $2np \geq K-3$, donc :

$$n(2p+3) = 2np + 3n \geq K-3 + 3n \geq K,$$

puisque $n \geq 1$. Par conséquent, $|\alpha_n - S_p(10^{-n})| \leq 10^{-K}/(2p+3) \leq 10^{-K}$.

Voici une réalisation Python complète. La boucle qui détermine p évite d'introduire une commande supplémentaire de plafond.

Python

```
def alpha_approchee(n, K):
    p = 0
    while 2 * n * p < K - 3:
        p = p + 1
    x = 10.0 ** (-n)
    terme = x
    somme = 0.0
    for j in range(p + 1):
        somme = somme + terme / (2 * j + 1)
        terme = -terme * x * x
    return somme, p
```

Au début de l'itération j , terme vaut $(-1)^j x^{2j+1}$; la division par $2j+1$ forme le terme à additionner. $\text{range}(p+1)$ parcourt bien $0, \dots, p$. La fonction renvoie l'approximation et l'ordre utilisé. Elle suppose $n \geq 1$ et $K \geq 1$.

Ce calcul préalable des constantes n'est pas soumis à la restriction d'une seule division de la partie II. Celle-ci concerne l'algorithme utilisant les constantes déjà connues. La garantie précédente porte sur le reste mathématique ; pour des précisions extrêmes, les erreurs d'arrondi du calcul flottant doivent aussi être maîtrisées.

Code exécuté pour $1 \leq n \leq 6$ et $1 \leq K \leq 12$: les valeurs ont été comparées à une évaluation numérique de l'arc tangente, avec une tolérance tenant compte des arrondis. La preuve de précision demeure celle donnée ci-dessus.

Partie II

1. Construction de la suite $(\tan(\theta_n))$

(a) Construction et unicité de h

On pose $r_0 = \theta$. Comme $r_0 \in [0, \alpha_0[$, I.3(c) fournit un unique $h(1) \in \{1, \dots, N+1\}$ tel que $\alpha_{h(1)} \leq r_0 < \alpha_{h(1)-1}$.

Supposons les premiers indices construits et posons :

$$r_{n-1} = \theta - \sum_{j=1}^{n-1} \alpha_{h(j)}.$$

Si $r_{n-1} \in [0, \alpha_0[$, on choisit l'unique $h(n)$ fourni par I.3(c), puis on définit

$r_n = r_{n-1} - \alpha_{h(n)}$. On a :

$$0 \leq r_n \leq r_{n-1} < \alpha_0.$$

L'invariant nécessaire à l'étape suivante est donc conservé. Par récurrence, les indices existent à tout rang. Leur unicité à chaque étape montre, aussi par récurrence, l'unicité de l'application h , en complétant par $h(0) = 0$.

Pour $\theta = 0$, le choix est $h(n) = N+1$ pour tout $n \geq 1$ et tous les restes valent 0.

(b) Monotonie et arrêt après un nombre fini d'étapes

Le reste r_n ne croît pas. Si $h(n+1) < h(n)$, alors $h(n+1) \leq h(n) - 1$ et :

$$r_n \geq \alpha_{h(n+1)} \geq \alpha_{h(n)-1} > r_{n-1},$$

ce qui contredit $r_n \leq r_{n-1}$. Ainsi $h(n+1) \geq h(n)$ pour $n \geq 1$, et $h(1) \geq 1 > h(0) = 0$. L'application h est croissante au sens large.

Tant que $h(n) \leq N$, on soustrait au reste un terme $\alpha_{h(n)} \geq \alpha_N > 0$. Si cette situation persistait pendant L étapes, on aurait :

$$0 \leq r_L = \theta - \sum_{j=1}^L \alpha_{h(j)} \leq \theta - L\alpha_N.$$

Cette inégalité est impossible pour un entier $L > \theta/\alpha_N$. On atteint donc nécessairement l'indice $N+1$ en un nombre fini d'étapes.

À partir de cet indice, on soustrait $\alpha_{N+1} = 0$; le reste ne change plus et l'unicité du choix impose $h(n) = N+1$ ensuite. Il existe donc un unique $m \in \mathbb{N}$ tel que $h(m) < N+1$ et $h(n) = N+1$ pour $n \geq m+1$.

La convention $h(0) = 0$ rend cette définition valable pour $m = 0$. Cela arrive exactement lorsque $0 \leq \theta < \alpha_N$: dès le premier choix, le reste appartient à $[\alpha_{N+1}, \alpha_N[$.

(c) Algorithme n'utilisant que des additions et comparaisons

On construit la somme courante s en essayant d'ajouter α_1 , puis α_2 , et ainsi de suite. À un niveau k , on calcule $c = s + \alpha_k$. Si $c \leq \theta$, on accepte cette somme, on enregistre l'indice k et on recommence au même niveau ; sinon, on passe à $k+1$. Après le refus au niveau N , le reste est inférieur à α_N et aucun terme positif n'est encore accepté.

La monotonie des indices prouvée en II.1(b) autorise à ne jamais revenir à un niveau précédent. Les indices acceptés sont exactement $h(1), \dots, h(m)$, et les sommes acceptées sont exactement $\theta_1, \dots, \theta_m$.

Voici le code complet, utilisant un tableau d'entrée alpha indexé de 0 à $N+1$. Les listes enregistrent les indices et les sommes successives. La commande `append(v)` ajoute la valeur v en fin de liste ; `len` renvoie sa longueur. Ces commandes ne réalisent pas d'opérations arithmétiques sur les nombres réels étudiés.

Python

```
def decomposition(theta, alpha, N):
    s = 0.0
    indices = []
    sommes = []
    comparaisons = 0
    additions = 0
    for k in range(1, N + 1):
        continuer = True
        while continuer:
            c = s + alpha[k]
            additions = additions + 1
            comparaisons = comparaisons + 1
            if c <= theta:
                s = c
                indices.append(k)
                sommes.append(s)
            else:
                continuer = False
    return indices, sommes, s, comparaisons, additions
```

Le compteur d'additions compte uniquement la formation de c ; les incréments des compteurs et le contrôle des boucles sont des opérations sur des indices entiers et sont exclus de la convention de coût. À chaque succès, une seule addition réelle est effectuée et sa valeur est réutilisée pour l'affectation de s . Aucun calcul de partie entière n'est nécessaire.

Le code ne contient aucune multiplication, division ou soustraction entre nombres réels. Il fournit également $s = \theta_m$. Pour $m = 0$, les deux listes sont vides et $s = 0$.

Code exécuté avec les constantes évaluées numériquement pour $1 \leq N \leq 6$, sur une grille de valeurs de θ et pour $\theta = 0$. Les sorties et les compteurs ont été confrontés aux propriétés et aux bornes établies dans ce corrigé.

2. Propriétés de la suite (θ_n)

(a) Croissance puis stationnarité

Pour $1 \leq j \leq m$, $h(j) \leq N$ et $\alpha_{h(j)} > 0$. Ainsi, pour $0 \leq n < m$:

$$\theta_m - \theta_n = \sum_{j=n+1}^m \alpha_{h(j)} > 0.$$

Pour $j \geq m+1$, $h(j) = N+1$ et $\alpha_{h(j)} = 0$. On a donc $\theta_n = \theta_m$ pour tout $n \geq m$.

Si $m = 0$, il n'y a aucun entier naturel $n < m$ et la suite entière est nulle. Les deux assertions demeurent valables.

(b) Encadrement de l'erreur sur l'angle

Le reste est toujours positif ou nul, donc $\theta_m \leq \theta$. À l'étape suivante, $h(m+1) = N+1$, et la règle de choix impose :

$$\alpha_{N+1} \leq \theta - \theta_m < \alpha_N.$$

Comme $\alpha_{N+1} = 0$, on obtient $\theta_m \leq \theta < \theta_m + \alpha_N$, y compris pour $m = 0$.

(c) Erreur sur la tangente et recentrage

Rectification de l'énoncé : le facteur 26 imprimé dans le PDF est remplacé par 2. C'est la borne que fournit l'inégalité des accroissements finis et qui justifie la précision annoncée après recentrage.

Sur le segment $[\theta_m, \theta] \subset [0, \frac{\pi}{4}[$, la tangente est de classe C^1 et :

$$0 < f'(x) = 1 + \tan^2 x \leq 2.$$

L'inégalité des accroissements finis et la monotonie donnent :

$$0 \leq \tan \theta - \tan \theta_m \leq 2(\theta - \theta_m) < 2\alpha_N \leq 2 \cdot 10^{-N}.$$

La stricte inégalité vient de II.2(b), et la dernière majoration vient de I.4(b). Pour $\theta = \theta_m$, le résultat reste vrai puisque la différence est nulle.

Posons $\varepsilon = 10^{-N}$ et $z = \tan \theta_m + \varepsilon$. L'encadrement précédent donne :

$$-\varepsilon \leq \tan \theta - z < \varepsilon,$$

donc $|\tan \theta - z| \leq \varepsilon = 10^{-N}$. C'est bien la valeur approchée demandée.

3. Calcul de $\tan(\theta_m)$

(a) Récurrence pour les tangentes

Pour $1 \leq n \leq m$, $h(n) \in \{1, \dots, N\}$, donc $\tan(\alpha_{h(n)}) = 10^{-h(n)} = k_n$. De plus, $\theta_{n-1} \in [0, \theta] \subset [0, \frac{\pi}{4}[$ et $\alpha_{h(n)} \leq \alpha_1 < \frac{\pi}{4}$.

La formule d'addition du préambule s'applique à ces deux angles ; leur somme est $\theta_n \leq \theta < \frac{\pi}{4}$. Elle donne :

$$\tan(\theta_n) = \frac{\tan(\theta_{n-1}) + k_n}{1 - k_n \tan(\theta_{n-1})}.$$

Le dénominateur est strictement positif : $0 < k_n \leq 1/10$ et $0 \leq \tan(\theta_{n-1}) < 1$, donc $1 - k_n \tan(\theta_{n-1}) > 9/10$. La division est ainsi justifiée.

La définition $k_0 = 10^{-h(0)} = 1$ n'intervient pas dans cette récurrence. Pour $n > m$, k_n n'est pas la tangente de $\alpha_{h(n)}$, car ce dernier angle vaut 0 ; on n'utilise donc pas la formule au-delà de m .

(b) Justification du quotient b_n/a_n

Méthode 1 — Récurrence avec positivité du dénominateur

Au rang 0, $a_0 = 1 > 0$ et $b_0/a_0 = 0 = \tan \theta_0$. Supposons, pour $1 \leq n \leq m$, que $a_{n-1} > 0$ et $b_{n-1}/a_{n-1} = \tan \theta_{n-1}$.

Alors :

$$a_n = a_{n-1} - k_n b_{n-1} = a_{n-1}(1 - k_n \tan \theta_{n-1}) > 0,$$

par la positivité prouvée en II.3(a). On peut donc diviser et écrire :

$$\frac{b_n}{a_n} = \frac{k_n a_{n-1} + b_{n-1}}{a_{n-1} - k_n b_{n-1}} = \frac{k_n + b_{n-1}/a_{n-1}}{1 - k_n b_{n-1}/a_{n-1}} = \tan \theta_n.$$

Par récurrence, $a_n > 0$ et $\tan \theta_n = b_n/a_n$ pour tout $0 \leq n \leq m$.

Méthode 2 — Interprétation trigonométrique du couple

Posons $\beta_j = \alpha_{h(j)}$ pour $1 \leq j \leq m$. On a $k_j = \tan \beta_j$ et $\cos \beta_j > 0$. Définissons :

$$P_0 = 1, \quad P_n = \prod_{j=1}^n \frac{1}{\cos \beta_j} > 0.$$

Montrons par récurrence que $a_n = P_n \cos \theta_n$ et $b_n = P_n \sin \theta_n$. C'est vrai au rang 0. Si c'est vrai au rang $n-1$, les formules d'addition donnent :

$$a_n = P_{n-1}(\cos \theta_{n-1} - \tan \beta_n \sin \theta_{n-1}) = \frac{P_{n-1}}{\cos \beta_n} \cos(\theta_{n-1} + \beta_n) = P_n \cos \theta_n,$$

$$b_n = P_{n-1}(\tan \beta_n \cos \theta_{n-1} + \sin \theta_{n-1}) = \frac{P_{n-1}}{\cos \beta_n} \sin(\theta_{n-1} + \beta_n) = P_n \sin \theta_n.$$

Comme $\theta_n \in [0, \frac{\pi}{4}]$, $a_n = P_n \cos \theta_n > 0$ et $b_n/a_n = \tan \theta_n$.

La première méthode suit directement l'algorithme et contrôle chaque division ; la seconde explique que le couple code le cosinus et le sinus avec un même facteur positif. Les fonctions trigonométriques de cette seconde preuve ne sont pas des opérations à effectuer dans l'algorithme.

(c) Nature et nombre des opérations

À chaque rang $1 \leq n \leq m$, on calcule $k_n b_{n-1}$ et $k_n a_{n-1}$: cela représente deux multiplications par la puissance de 10 déjà disponible k_n . Puis on calcule $a_{n-1} - k_n b_{n-1}$ et $k_n a_{n-1} + b_{n-1}$: cela représente deux additions, en comptant la soustraction comme une addition.

Après les m étapes, on effectue la seule division b_m/a_m , autorisée puisque $a_m > 0$. Le coût est donc $2m$ additions, $2m$ multiplications par des puissances de 10, et une division. Pour $m = 0$, on peut conserver cette procédure en effectuant la division $0/1$; une réalisation optimisée peut simplement renvoyer 0.

Voici le code complet, où puissances[k] contient la constante 10^{-k} préalablement connue :

Python

```
def tangente_de_la_somme(indices, puissances):
    a = 1.0
    b = 0.0
    for k in indices:
        a_ancien = a
        b_ancien = b
        q = puissances[k]
        a = a_ancien - q * b_ancien
        b = q * a_ancien + b_ancien
    return b / a

def valeur_approchee(theta, alpha, puissances, N):
    indices, sommes, s, tests, additions = decomposition(theta, alpha, N)
    m = len(indices)
    valeur = tangente_de_la_somme(indices, puissances)
    valeur = valeur + puissances[N]
    return valeur, tests, additions + 2 * m + 1, 2 * m, 1
```

Les anciennes valeurs de a et b sont sauvegardées avant les mises à jour, afin que les deux formules utilisent bien le rang précédent. La fonction finale ajoute 10^{-N} pour recentrer l'approximation, ce qui coûte une addition de plus. Elle renvoie aussi les coûts des comparaisons réelles, additions réelles, multiplications par des puissances de 10 et divisions. Les opérations sur les compteurs entiers sont exclues.

Les codes ont été exécutés, conjointement avec celui de II.1(c), pour $1 \leq N \leq 6$ sur une grille d'angles incluant 0. Le quotient obtenu a été comparé numériquement à la

tangente de la somme construite, et la valeur recentrée à la tangente de l'angle d'entrée. Ces vérifications numériques accompagnent les démonstrations, sans les remplacer.

Partie III

1. Nombre d'utilisations de chaque angle

(a) Somme des effectifs

Les indices $h(1), \dots, h(m)$ appartiennent tous à $\{1, \dots, N\}$. Les indices suivants valent $N + 1$ et ne contribuent à aucun des X_k .

Les ensembles $\{p \in \{1, \dots, m\} : h(p) = k\}$, pour $1 \leq k \leq N$, sont disjoints et leur réunion est $\{1, \dots, m\}$. Le dénombrement donne donc :

$$m = \sum_{k=1}^N X_k.$$

Si $m = 0$, tous les effectifs sont nuls et cette égalité reste vraie.

(b) Calcul et majoration de X_1

Par croissance de h , les occurrences de 1, s'il y en a, sont les premières. L'algorithme ajoute α_1 tant que la somme ne dépasse pas θ . Le nombre X_1 est donc caractérisé par :

$$X_1 \alpha_1 \leq \theta < (X_1 + 1) \alpha_1.$$

Puisque $\alpha_1 > 0$, on divise pour obtenir $X_1 \leq \theta / \alpha_1 < X_1 + 1$. Par définition de la partie entière :

$$X_1 = \text{Int}(\theta / \alpha_1).$$

D'après I.4(e), $\theta < \alpha_0 < 8\alpha_1$, donc $0 \leq \theta / \alpha_1 < 8$. Par conséquent, $X_1 \leq 7$.

(c) Calcul et majoration des X_k suivants

Pour $2 \leq k \leq N$, posons :

$$r^{(k-1)} = \theta - \sum_{j=1}^{k-1} X_j \alpha_j.$$

C'est le reste après le traitement des niveaux $1, \dots, k-1$. Il vérifie $0 \leq r^{(k-1)} < \alpha_{k-1}$, puisque le dernier test au niveau $k-1$ a échoué. Le niveau k accepte exactement X_k additions, puis refuse la suivante :

$$X_k \alpha_k \leq r^{(k-1)} < (X_k + 1) \alpha_k.$$

En divisant par $\alpha_k > 0$, on obtient :

$$X_k = \text{Int} \left(\frac{r^{(k-1)}}{\alpha_k} \right) = \text{Int} \left(\frac{\theta - X_1 \alpha_1 - \dots - X_{k-1} \alpha_{k-1}}{\alpha_k} \right).$$

L'inégalité I.4(a), avec $n = k-1$, donne $\alpha_{k-1} < 10\alpha_k$. Par conséquent :

$$0 \leq \frac{r^{(k-1)}}{\alpha_k} < \frac{\alpha_{k-1}}{\alpha_k} < 10,$$

donc $X_k \leq 9$. Cette conclusion inclut les niveaux sautés, où $X_k = 0$.

Les parties entières donnent des formules mathématiques pour les effectifs ; elles n'ont pas à être calculées par le programme, qui réalise les soustractions successives sous forme d'additions acceptées.

(d) Majoration du nombre total d'étapes

En utilisant III.1(a), III.1(b) et III.1(c) :

$$m = X_1 + \sum_{k=2}^N X_k \leq 7 + 9(N-1) = 9N - 2.$$

Pour $N = 1$, la somme de $k = 2$ à N est vide et la borne devient $m \leq 7$, exactement comme III.1(b).

2. Complexité de la construction des indices

La réalisation retenue est celle explicitée dans l'énoncé et en II.1(c). Les comptes exacts dépendent de cette réalisation : chaque valeur candidate $c = s + \alpha_k$ est calculée une seule fois et réutilisée si elle est acceptée. On visite les N niveaux, y compris ceux où aucun terme n'est accepté.

(a) Nombre de comparaisons

Au niveau k , il y a X_k tests réussis, correspondant aux occurrences acceptées, et un dernier test échoué, qui provoque le passage au niveau suivant ou l'arrêt si $k = N$.

Ainsi le nombre de tests entre nombres réels est :

$$T_N = \sum_{k=1}^N (X_k + 1) = m + N.$$

La majoration de III.1(d) donne $T_N \leq 10N - 2$, indépendamment de θ .

Pour $m = 0$, les N niveaux sont tous refusés et $T_N = N$, ce qui est cohérent avec la formule.

(b) Nombre d'additions

Chaque test nécessite exactement une addition réelle, celle qui forme $c = s + \alpha_k$. En cas de succès, l'affectation $s = c$ réutilise ce résultat et ne demande aucune addition supplémentaire.

On a donc :

$$A_N = \sum_{k=1}^N (X_k + 1) = m + N \leq 10N - 2.$$

Ce nombre comprend également les candidats refusés. Les affectations, enregistrements et changements d'indice entier ne sont pas des additions réelles comptées dans cette convention.

3. Complexité totale de la valeur approchée

On combine la construction des indices, le calcul des couples (a_n, b_n) et l'addition finale de 10^{-N} . Le calcul préalable des constantes est exclu, conformément à l'hypothèse de la fin de la partie I et à la convention précisée dans la partie III.

(a) Comparaisons

La construction des indices est la seule phase réalisant des comparaisons entre nombres réels. Le total est donc :

$$T_N = m + N \leq 10N - 2.$$

(b) Additions

Il faut $m + N$ additions pour la construction des indices, $2m$ pour les couples, et une addition pour le recentrage. Le total vaut :

$$A_{\text{total}} = (m + N) + 2m + 1 = 3m + N + 1.$$

Comme $m \leq 9N - 2$, on obtient :

$$A_{\text{total}} \leq 3(9N - 2) + N + 1 = 28N - 5.$$

L'addition finale est indispensable à la valeur approchée annoncée, même si elle n'intervient pas dans le seul calcul de $\tan \theta_m$.

(c) Multiplications par des puissances de 10

Les couples nécessitent deux telles multiplications par étape. Les autres phases n'en nécessitent aucune, les constantes étant déjà disponibles. Ainsi :

$$M_{\text{total}} = 2m \leq 18N - 4.$$

(d) Divisions

On effectue seulement le quotient final b_m/a_m , avec $a_m > 0$. Le nombre de divisions est donc égal à 1 pour la procédure décrite, et en particulier majoré par 1 indépendamment de θ .

Pour $m = 0$, on peut supprimer la division $0/1$; cette optimisation ne change pas le majorant.

Les bornes obtenues sont résumées ci-dessous.

Opération	Nombre pour la réalisation choisie	Majorant
Comparaisons réelles	$m + N$	$10N - 2$
Additions réelles	$3m + N + 1$	$28N - 5$
Multiplications par une puissance de 10	$2m$	$18N - 4$
Divisions	1	1

Relevé des adaptations

- II.1(b) : explicitation du cas $m = 0$, déjà compatible avec $h(0) = 0$, pour couvrir les angles $0 \leq \theta < \alpha_N$ et l'arrêt immédiat.
- II.2(c) : rectification du facteur imprimé 26 en 2 dans la borne supérieure de l'erreur ; l'inégalité des accroissements finis donne $2 \cdot 10^{-N}$ et permet le recentrage avec une précision 10^{-N} .
- III.2(a), III.2(b), III.3(a) à III.3(d) : ajout, dans le préambule de la partie III, d'une réalisation explicite de l'algorithme et de sa convention de comptage. Les questions originales demandent des nombres exacts d'opérations sans fixer les détails d'implémentation ; cette précision rend les comptes univoques.

Aucune question n'est admise ni supprimée. Toutes les démonstrations utilisent les outils du référentiel fourni.