

Plaque 3 - Analyse numérique : équations, intégrales, équations différentielles

Python et calcul scientifique — Licence 1 (L1)

Corrections détaillées

Boîte à outils

Point fixe $x_{n+1} = g(x_n)$: si $|g'| \leq k < 1$ près de α , convergence **linéaire**, $|x_n - \alpha| \leq k^n |x_0 - \alpha|$.

— **Newton** : $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$, convergence **quadratique** près d'une racine simple, mais peut diverger si x_0 est mal choisi.

— **Quadratures** sur n sous-intervalles de pas h : rectangles $O(h)$, milieu et trapèzes $O(h^2)$, Simpson $O(h^4)$:

$$\text{ordre } p \implies \text{erreur}(h/2) \approx \frac{\text{erreur}(h)}{2^p}$$

— **Euler** pour $y' = F(t, y)$: $y_{k+1} = y_k + hF(t_k, y_k)$, ordre 1 ; **Heun** (Euler corrigé) : ordre 2.

— **Différence finie** : $\frac{f(x+h) - f(x)}{h}$ (erreur $O(h)$) ; centrée $\frac{f(x+h) - f(x-h)}{2h}$ (erreur $O(h^2)$) ; les arrondis imposent de ne pas prendre h trop petit.

Correction de l'exercice 1 - Point fixe : résoudre $\cos x = x$

a)

```
import math

x = 1.0
n = 0
while abs(math.cos(x) - x) > 1e-10:
    x = math.cos(x)
    n = n + 1
print(n, x)
```

b) $h(x) = \cos x - x$ est continue, strictement décroissante ($h' = -\sin x - 1 < 0$ sur $]0, 1]$), avec $h(0) = 1 > 0$ et $h(1) = \cos 1 - 1 < 0$: une unique racine α (TVI).

c) Par l'inégalité des accroissements finis, $x_{n+1} - \alpha = \cos x_n - \cos \alpha = -\sin(c_n)(x_n - \alpha)$ avec c_n entre x_n et α ; quand $x_n \rightarrow \alpha$, le facteur tend vers $\sin \alpha \approx 0,674$:

$$|x_{n+1} - \alpha| \approx 0,674 |x_n - \alpha|$$

d) Chaque itération gagne $\log_{10} \frac{1}{0,674} \approx 0,17$ décimale : pour passer d'une erreur $\approx 0,3$ à 10^{-10} , il faut environ $\frac{9,5}{0,17} \approx 56$ itérations ✓ C'est une convergence **linéaire** : lente, mais sans dérivée à calculer.

Réponse. $\alpha \approx 0,7390851332$ en 57 itérations ; facteur $\sin \alpha \approx 0,674$, soit 0,17 décimale par pas.

Correction de l'exercice 2 - Newton : le nombre de décimales double

a) $x_{k+1} = x_k - \frac{x_k^2 - 2}{2x_k} = \frac{x_k}{2} + \frac{1}{x_k}$ (méthode de Héron).

```
import math

x = 1.0
for k in range(5):
    x = x - (x * x - 2) / (2 * x)
    print(k + 1, x, abs(x - math.sqrt(2)))
```

b) Le nombre de décimales exactes passe de 1 à 2, 5, 11 : il **double** environ à chaque itération (convergence quadratique), contre un gain constant pour le point fixe.

c) $x_{k+1} - \sqrt{2} = \frac{x_k^2 + 2 - 2\sqrt{2}x_k}{2x_k} :$

$$x_{k+1} - \sqrt{2} = \frac{(x_k - \sqrt{2})^2}{2x_k}$$

— Contrôle : $\frac{0,0858^2}{2 \times 1,5} \approx 0,00245 \checkmark$

d) L'erreur théorique au rang 5 serait $\approx \frac{(1,6 \times 10^{-12})^2}{2,8} \approx 10^{-24}$, bien en dessous de la précision des flottants ($\approx 2 \times 10^{-16}$ relative) : x_5 est le flottant le plus proche de $\sqrt{2}$, égal à `math.sqrt(2)`.

Réponse. $x_{k+1} = \frac{x_k}{2} + \frac{1}{x_k}$; erreur au carré à chaque pas ; x_5 exact à la précision machine.

Correction de l'exercice 3 – Quand Newton diverge

a) $f(x) = \frac{1}{1+x^2}$, donc $\frac{f(x)}{f'(x)} = (1+x^2) \arctan x$.

```
import math

x = 1.5
for k in range(5):
    x = x - math.atan(x) * (1 + x * x)
    print(round(x, 4))
```

b) Depuis 1 : les itérés changent de signe mais se rapprochent de 0 de plus en plus vite (quadratique). Depuis 1,5 : ils changent de signe et leur valeur absolue **explose** : divergence.

c) Loin de 0, $\arctan$ est presque plate ($f'(x) = \frac{1}{1+x^2}$ petit) : la tangente coupe l'axe **très loin** de l'autre côté, où la courbe est encore plus plate. Il existe un seuil $x^* \approx 1,39$ tel que $|x_0| < x^*$ converge et $|x_0| > x^*$ diverge.

$$|x_1| > |x_0| \Leftrightarrow (1+x_0^2) \arctan x_0 > 2x_0$$

d) Encadrer la racine par dichotomie jusqu'à un intervalle petit (où f' ne varie plus beaucoup), puis lancer Newton ; si un itéré de Newton sort de l'encadrement, refaire un pas de dichotomie (principe de la méthode de Brent).

Réponse. Convergence rapide depuis 1, divergence depuis 1,5 (seuil $\approx 1,39$) ; combiner dichotomie et Newton.

Correction de l'exercice 4 – Milieu, trapèzes, Simpson

a)

```
def milieu(f, a, b, n):
    h = (b - a) / n
    return h * sum(f(a + (k + 0.5) * h) for k in range(n))
```

```
def trapezes(f, a, b, n):
    h = (b - a) / n
    return h * ((f(a) + f(b)) / 2 + sum(f(a + k * h) for k in range(1, n)))

def simpson(f, a, b, n):
    h = (b - a) / n
    s = f(a) + f(b)
    for k in range(1, n):
        s = s + (4 if k % 2 == 1 else 2) * f(a + k * h)
    return h / 3 * s
```

b) Même nombre d'évaluations de f ; Simpson est environ 1500 fois plus précis que les trapèzes.

c) $\frac{1,43 \times 10^{-3}}{-7,16 \times 10^{-4}} \approx -2$. Les termes principaux se compensent dans

$$\frac{2M+T}{3} - I \approx \frac{2 \times (-7,16 \times 10^{-4}) + 1,43 \times 10^{-3}}{3} \approx 0$$

— C'est exactement Simpson (sur $2n$ demi-pas) : combiner deux méthodes d'ordre 2 fait gagner deux ordres.

d) exp est convexe : la corde est **au-dessus** de la courbe (trapèzes par excès, erreur > 0) et la tangente au milieu **en dessous** (point milieu par défaut, erreur < 0).

Réponse. Erreurs $-7,2 \times 10^{-4}$ (milieu), $1,4 \times 10^{-3}$ (trapèzes), $9,5 \times 10^{-7}$ (Simpson) ; $\frac{2M+T}{3} = \text{Simpson}$.

Correction de l'exercice 5 - Mesurer l'ordre d'une méthode

a) Milieu : $\frac{7,157}{1,790} \approx 4,0$; trapèzes : $\frac{14,317}{3,580} \approx 4,0$; Simpson : $\frac{9,535}{0,5964} \approx 16,0$.

b) Si l'erreur vaut Ch^p , diviser h par 2 la divise par 2^p :

$$2^p = 4 \Leftrightarrow p = 2 \text{ (milieu, trapèzes)}, \quad 2^p = 16 \Leftrightarrow p = 4 \text{ (Simpson)}$$

c) Trapèzes : $1,43 \times 10^{-3} \times \left(\frac{10}{n}\right)^2 = 10^{-10}$ donne $n \approx 10 \times \sqrt{1,43 \times 10^7} \approx 3,8 \times 10^4$.

Simpson : $9,5 \times 10^{-7} \times \left(\frac{10}{n}\right)^4 = 10^{-10}$ donne $n \approx 10 \times (9500)^{1/4} \approx 99$: cent fois moins d'évaluations.

d) Une faute de programmation (mauvais coefficient, indice décalé) conserve souvent la convergence mais **dégrade l'ordre** : un Simpson qui n'est que d'ordre 2 trahit une erreur de code, invisible sur une seule valeur.

Réponse. Rapports 4, 4, 16 : ordres 2, 2, 4 ; pour 10^{-10} : $\approx 38\,000$ pas (trapèzes) contre ≈ 100 (Simpson).

Correction de l'exercice 6 - Euler et Heun

a)

```
def heun(n):
    h = 1 / n
    y = 1.0
    for k in range(n):
        p = y + h * y
        y = y + h / 2 * (y + p)
    return y
```

b) n multiplié par 10, erreur divisée par $\approx 93 \approx 10^2$: ordre 2. Pour $n = 100$, Heun est 300

fois plus précis qu'Euler pour seulement deux fois plus de calculs.

c) $y_{k+1} = y_k + \frac{h}{2}(y_k + y_k + hy_k) = \left(1 + h + \frac{h^2}{2}\right)y_k$: c'est le développement de e^h à l'ordre 2 (Euler s'arrêtait à l'ordre 1).

d) $n \ln\left(1 + h + \frac{h^2}{2}\right) = n\left(h - \frac{h^3}{6} + O(h^4)\right) = 1 - \frac{1}{6n^2} + O\left(\frac{1}{n^3}\right)$, d'où

$$e - y_n \sim \frac{e}{6n^2}$$

— $n = 100$: $\frac{e}{60000} \approx 4,53 \times 10^{-5}$, proche de l'erreur mesurée $4,50 \times 10^{-5}$ ✓

Réponse. Heun d'ordre 2 : $y_{k+1} = (1 + h + \frac{h^2}{2})y_k$, erreur $\sim \frac{e}{6n^2}$.

Correction de l'exercice 7 - Accélérer la série de Leibniz

a)

```
def leibniz(n):
    s = 0.0
    for k in range(n):
        s = s + (-1) ** k / (2 * k + 1)
    return 4 * s
```

b) Pour une série alternée à termes décroissant vers 0, l'erreur est majorée par le premier terme négligé :

$$|\pi - S_n| \leq \frac{4}{2n+1} \approx \frac{2}{n} = 2 \times 10^{-3}$$

— (valeur réelle $\approx 10^{-3}$, soit environ la moitié : convergence très lente, en $\frac{1}{n}$).

c) Les sommes partielles encadrent π en oscillant : $S_{1000} < \pi < S_{1001}$, avec des erreurs presque opposées ($\approx +\frac{1}{n}$ et $-\frac{1}{n}$). Leur moyenne annule le terme principal et l'erreur devient d'ordre $\frac{1}{n^2}$ (ici $\approx \frac{1}{2n^2} = 5 \times 10^{-7}$).

d) Il faudrait $\frac{1}{n} \approx 10^{-7}$, soit $n \approx 10^7$ termes ; avec la moyenne, $\frac{1}{2n^2} \approx 10^{-7}$ donne $n \approx 2240$: le travail est divisé par plus de 4000.

Réponse. Erreur $\approx \frac{1}{n}$ (10^{-3} pour $n = 1000$) ; la moyenne de deux sommes consécutives donne 5×10^{-7} .

Correction de l'exercice 8 - Calculer e avec une précision contrôlée

a)

```
import math

s, t, k = 1.0, 1.0, 0
while t > 1e-15:
    k = k + 1
    t = t / k
    s = s + t
print(k, s, math.e - s)
```

— t passe de $\frac{1}{(k-1)!}$ à $\frac{1}{k!}$ par une seule division : pas de factorielle géante.

b) Pour $k > n$, $\frac{1}{k!} \leq \frac{1}{n!} \frac{1}{(n+1)^{k-n}}$, et la série géométrique donne

$$\sum_{k > n} \frac{1}{k!} \leq \frac{1}{n!} \sum_{j \geq 1} \frac{1}{(n+1)^j} = \frac{1}{n!n}$$

c) $\frac{1}{17! \times 17} \approx 1,7 \times 10^{-16} < 10^{-15}$: 18 termes (de $k = 0$ à 17) suffisent ;
 $\frac{1}{16! \times 16} \approx 3 \times 10^{-15}$ ne suffit pas.

d) Les flottants ont une précision relative $\approx 2,2 \times 10^{-16}$: pour une valeur proche de 2,7, l'écart entre deux flottants voisins vaut $4,4 \times 10^{-16}$. L'erreur observée est **d'arrondi**, pas de troncature.

Réponse. 18 termes ; reste $\leq \frac{1}{n!n}$; erreur finale $4,4 \times 10^{-16}$ (limite des flottants).

Correction de l'exercice 9 - Dérivée numérique : le bon pas

a) L'erreur diminue d'abord avec h (linéairement pour D_h , quadratiquement pour C_h), atteint un minimum, puis **remonte** : pour $h = 10^{-14}$ on n'a plus que 3 décimales.

b) $e^h = 1 + h + \frac{h^2}{2} + \frac{h^3}{6} + \dots$; $D_h = 1 + \frac{h}{2} + O(h^2)$; les termes pairs s'annulent dans $e^h - e^{-h} = 2h + \frac{h^3}{3} + \dots$, d'où $C_h = 1 + \frac{h^2}{6} + O(h^4)$. **Contrôle** : $\frac{10^{-4}}{6} \approx 1,7 \times 10^{-5}$ ✓

c) Erreur totale $\approx \frac{h}{2} + \frac{\varepsilon}{h}$, minimale pour $h \approx \sqrt{2\varepsilon} \approx 10^{-8}$; pour la formule centrée, $\frac{h^2}{6} + \frac{\varepsilon}{h}$ est minimale pour

$$h \approx (3\varepsilon)^{1/3} \approx 10^{-5}, \text{ erreur } \approx 10^{-11}$$

d) Non : en dessous du pas optimal, on soustrait deux nombres presque égaux (annulation catastrophique) et l'erreur d'arrondi, divisée par h , domine.

Réponse. Pas optimal $\approx 10^{-8}$ pour D_h (erreur $\approx 10^{-8}$), $\approx 10^{-5}$ pour C_h (erreur $\approx 10^{-11}$).

Correction de l'exercice 10 - La suite $u(n+1) = \sqrt{2 + u(n)}$

a) Par récurrence, $0 \leq u_n < 2$ implique $u_{n+1} = \sqrt{2 + u_n} < 2$, et $u_{n+1} > u_n \Leftrightarrow 2 + u_n > u_n^2 \Leftrightarrow (2 - u_n)(1 + u_n) > 0$ ✓ Croissante majorée : convergente, vers $\ell = \sqrt{2 + \ell}$, soit $\ell = 2$.

b) Entre $n = 5$ et $n = 10$, l'erreur est divisée par $\approx 1024 = 4^5$: facteur $\frac{1}{4}$ par itération.

c) $g'(x) = \frac{1}{2\sqrt{2+x}}$, $g'(2) = \frac{1}{4}$: $2 - u_{n+1} \approx \frac{1}{4}(2 - u_n)$, convergence linéaire de rapport $\frac{1}{4}$.

d) $2 \cos \frac{\pi}{2^{n+2}} = \sqrt{2 + 2 \cos \frac{\pi}{2^{n+1}}}$ (formule $1 + \cos 2\theta = 2 \cos^2 \theta$), et $u_0 = 2 \cos \frac{\pi}{2} = 0$ ✓ Alors

$$2 - u_n = 2 \left(1 - \cos \frac{\pi}{2^{n+1}} \right) \sim \frac{\pi^2}{4^{n+1}}$$

— $n = 10$: $\frac{\pi^2}{4^{11}} \approx 2,35 \times 10^{-6}$ ✓

Réponse. $u_n \rightarrow 2$ avec un rapport $\frac{1}{4}$; $u_n = 2 \cos \frac{\pi}{2^{n+1}}$, $2 - u_n \sim \frac{\pi^2}{4^{n+1}}$.

Correction de l'exercice 11 - Deux séries pour $\ln 2$

a) $\ln(1+x) = \sum_{k \geq 1} \frac{(-1)^{k+1} x^k}{k}$ en $x = 1$; et $-\ln(1-x) = \sum_{k \geq 1} \frac{x^k}{k}$ en $x = \frac{1}{2}$: $-\ln \frac{1}{2} = \ln 2$.

```
import math
```

```
print(sum((-1) ** (k + 1) / k for k in range(1, 11)) - math.log(2))
```

```
print(sum(1 / (k * 2 ** k) for k in range(1, 11)) - math.log(2))
```

b) Même nombre de termes, précision 500 fois meilleure pour la seconde : ses termes décroissent **géométriquement**.

c)

$$\sum_{k>n} \frac{1}{k2^k} \leq \frac{1}{n+1} \sum_{k>n} \frac{1}{2^k} = \frac{1}{(n+1)2^n}$$

— Pour $n = 10$: $\frac{1}{11 \times 1024} \approx 8,9 \times 10^{-5}$, cohérent avec $8,2 \times 10^{-5}$ ✓

d) Série alternée : erreur $\approx \frac{1}{2n}$, il faudrait $n \approx 5 \times 10^{11}$ termes. Seconde : $(n+1)2^n \geq 10^{12}$ dès $n = 35$ ($36 \times 2^{35} \approx 1,24 \times 10^{12}$, alors que $35 \times 2^{34} \approx 6 \times 10^{11}$).

Réponse. Erreurs $4,8 \times 10^{-2}$ contre $8,2 \times 10^{-5}$; $\approx 5 \times 10^{11}$ termes contre 35 pour 10^{-12} .

Correction de l'exercice 12 - Additionner des flottants

a) 0,1 n'est pas représentable exactement ; chaque addition arrondit, et les erreurs se cumulent en $1 - 1,1 \times 10^{-16}$. Le test d'égalité exacte échoue.

b) Chaque somme partielle est arrondie avec une erreur relative $\leq u$; en première approximation

$$|\text{erreur}| \leq (n-1)uS$$

— soit $\approx 10^{-15}$ pour $n = 10$ et $S = 1$: l'erreur observée ($1,1 \times 10^{-16}$) est bien dans cette borne.

c) L'erreur d'une addition est proportionnelle à la **somme partielle** : si l'on ajoute d'abord les gros termes, les petits sont ajoutés à un grand nombre et perdent leurs derniers chiffres.

d) *math.fsum* garde en mémoire les erreurs d'arrondi successives (sommation exacte puis un seul arrondi final) : 1,0. Test robuste : $\text{abs}(s - 1) < 1e-12$, ou *math.isclose*(*s*, 1).

Réponse. $10 \times 0,1$ donne 0,9999999999999999 ; erreur $\leq (n-1)uS$; utiliser *math.fsum* ou *math.isclose*.

Correction de l'exercice 13 - Oscillateur : Euler n'est pas stable

a)

```
def energie_euler(h, T):
    x, v = 1.0, 0.0
    for k in range(round(T / h)):
        x, v = x + h * v, v - h * x
    return x * x + v * v
```

b) $x_{k+1}^2 + v_{k+1}^2 = (x + hv)^2 + (v - hx)^2 = (1 + h^2)(x^2 + v^2)$. Après $n = \frac{T}{h}$ pas :

$$E = (1 + h^2)^{T/h} \approx e^{hT}$$

— $T = 2\pi$: $e^{0,0628} \approx 1,065$ ✓ ; $T = 20\pi$: $e^{0,628} \approx 1,87$ ✓ La trajectoire **spirale vers l'extérieur**.

c)

```
def energie_symplectique(h, T):
    x, v = 1.0, 0.0
    for k in range(round(T / h)):
        v = v - h * x
        x = x + h * v
```

```
return x * x + v * v
```

d) Le schéma symplectique conserve exactement une énergie « modifiée » voisine de E : l'erreur **reste bornée** au lieu de croître exponentiellement. C'est le schéma utilisé en mécanique céleste et en dynamique moléculaire.

Réponse. Euler : $E \times (1 + h^2)$ par pas, $E \approx e^{hT}$ (spirale) ; variante symplectique : énergie quasi conservée.

Correction de l'exercice 14 - Monte-Carlo : deux estimateurs de π

a)

```
import math, random

random.seed(0)
n = 100000
dedans = 0
for k in range(n):
    x = random.random()
    if random.random() <= math.sqrt(1 - x * x):
        dedans = dedans + 1
print(4 * dedans / n)

random.seed(0)
s = 0.0
for k in range(n):
    s = s + math.sqrt(1 - random.random() ** 2)
print(4 * s / n)
```

b) A : $|3,14844 - \pi| \approx 6,8 \times 10^{-3}$; B : $\approx 4 \times 10^{-4}$.

c) A : $\sigma = 4 \sqrt{\frac{0,785 \times 0,215}{10^5}} \approx 5,2 \times 10^{-3}$; B :

$$\sigma_B = 4 \sqrt{\frac{0,050}{10^5}} \approx 2,8 \times 10^{-3}$$

— B est deux fois plus précis (quatre fois moins de tirages pour la même précision) : il remplace un tirage « tout ou rien » par une valeur **moyenne**, moins dispersée. L'écart observé pour A ($1,3\sigma$) est normal.

d) Pour que le résultat soit **reproductible** : relancer le programme redonne les mêmes nombres, ce qui permet de déboguer et de comparer deux méthodes sur les mêmes tirages.

Réponse. Erreurs $6,8 \times 10^{-3}$ (A) et 4×10^{-4} (B) ; écarts types $5,2 \times 10^{-3}$ et $2,8 \times 10^{-3}$.

Correction de l'exercice 15 - Synthèse : trois méthodes pour $x = e^{(-x)}$

a)

```
import math

f = lambda x: x - math.exp(-x)
a, b, nd = 0.0, 1.0, 0
while b - a > 1e-12:
    m = (a + b) / 2
    if f(a) * f(m) <= 0:
        b = m
    else:
        a = m
```

```

nd = nd + 1

x, npf = 0.5, 0
while abs(math.exp(-x) - x) > 1e-12:
    x = math.exp(-x)
    npf = npf + 1

y, nn = 0.5, 0
while abs(f(y)) > 1e-12:
    y = y - f(y) / (1 + math.exp(-y))
    nn = nn + 1
print(nd, npf, nn)

```

b) Il faut $2^{-n} \leq 10^{-12}$, soit $n \geq 12 \log_2 10 \approx 39,9$: 40 étapes.

c) $g(x) = e^{-x}$ envoie $[0,5; 0,7]$ dans lui-même avec $|g'| = e^{-x} \leq e^{-0,5} < 1$: contraction. Près de α , $|g'(\alpha)| = e^{-\alpha} = \alpha \approx 0,567$; gain de $\log_{10} \frac{1}{0,567} \approx 0,25$ décimale par pas, soit $\approx \frac{11}{0,25} \approx 45$ itérations ✓

d) $y_{k+1} = y_k - \frac{y_k - e^{-y_k}}{1 + e^{-y_k}}$: convergence quadratique (0,5 est déjà proche de α).
coût : 40 (dichotomie), 45 (point fixe), 3 (Newton)

— Newton est le plus rapide **si** on connaît f et un bon point de départ ; la dichotomie est lente mais **toujours** sûre (il suffit d'un changement de signe). En pratique on les combine.

Réponse. $\alpha \approx 0,5671432904$; 40 (dichotomie), 45 (point fixe, facteur α), 3 (Newton) itérations.