Le rythme de valse chiffré, et la vraie formule d'inversion (Riemann-von Mangoldt via W, la fonction de Lambert)
Denise Vella-Chemla pilotant l'ia claude, juillet 2026.

1. Ce que l'ia claude entend par corrélations numériques

Un rapprochement numérique mérite une telle appellation quand :

1. aucune dérivation n'explique pourquoi les deux quantités devraient coïncider ;
2. la formule comporte des constantes ajustées empiriquement ;
3. le test porte sur un échantillon trop petit pour exclure la coïncidence ;
4. l'écart n'est jamais mesuré précisément. Le signe distinctif : le rapprochement se dégrade si on l'étend. C'est exactement le cas ici, de façon chiffrée.

2. Reconstruction exacte de la formule

D'après le document source, https://denisevellachemla.eu/doc-p3-gemini-dvc-20260708.pdf , étant lui-même une analyse de la note *Résumé des petits résultats de novembre dans le plan complexe*, 24 novembre 2024, https://denisevellachemla.eu/resume-nov-complexe.pdf dernier item), pour $z_k = \mathrm{Im}(\rho_k)$ (ρ_k le k-ième zéro non trivial de ζ) :

$$f(z_k) = \exp\left(W\left(-\left(\frac{z_k}{2}\right)^2 \frac{2\pi}{180}\right)\right).$$

3. Premier constat : W est complexe, pas réelle

Pour tout $k \geq 1$, l'argument $-\left(\frac{z_k}{2}\right)^2 \frac{2\pi}{180}$ est inférieur à $-\frac{1}{e} \approx -0.368$ (déjà -1.744 pour $k=1$, avec $z_1 = 14.1347\ldots$). Or W n'a de valeur réelle que pour un argument $\geq -1/e$. Donc $f(z_k)$ est nécessairement un nombre *complexe* pour tout k : parler de “partie entière ” suppose implicitement d'en prendre le module (ou une autre réduction au réel), ce que le document d'origine ne précise pas.

k	argument de W	W(argument)
1	-1.744	$0.0744 + 1.6168i$
2	-3.857	$0.6518 + 1.9011i$
5	-9.466	$1.3279 + 2.1286i$

4. En prenant le module : la reconstruction est exacte

En calculant $|f(z_k)|$ avec les 30 premières hauteurs z_k réelles (via `mpmath.zetazero`), on retrouve **exactement** votre suite pour les 12 premiers termes :

$$\lfloor |f(z_k)| \rfloor_{k=1}^{12} = [1, 1, 2, 3, 3, 4, 5, 5, 6, 7, 7, 8].$$

La reconstruction de votre calcul original est donc confirmée : c'est bien le module qui a été utilisé (implicitement ou via l'affichage d'un nombre complexe par le langage de programmation employé).

5. En continuant au-delà de 12 termes : la coïncidence se rompt

En poussant à $k = 30$, le rapport $g(k)/|f(z_k)|$, au lieu de rester proche de 1 (ce qu'impliquerait un vrai alignement des deux suites), **dérive et se stabilise autour de** 2,00 :

k	z_k	$\|f(z_k)\|$	$g(k) = 22k/15$	rapport $g(k)/\|f(z_k)\|$
1	14.135	1.077	1.467	1.362
5	32.935	3.773	7.333	1.944
10	49.774	7.170	14.667	2.046
15	65.113	10.991	22.000	2.002
20	77.145	14.444	29.333	2.031
25	88.809	18.159	36.667	2.019
30	101.318	22.531	44.000	1.953

J'avais proposé $(2 * k/3 + 4/5)$ pour obtenir un rythme du genre [1,2,2,3,4,4,...] mais je l'avais mal écrit $(2/3 + 4/5)k$. C'est à cette idée que correspond la fraction 22/15 ci-dessous.

Diagnostic : $g(k) = 22k/15$ ne suit pas $|f(z_k)|$: il en vaut sensiblement le *double*, de façon stable dès $k \approx 8$. La coïncidence observée sur les 12 premiers termes n'était pas un vrai alignement de tendance : les deux suites démarrent près de 1 et croissent lentement au début, ce qui suffit à faire coïncider leurs parties entières par hasard sur une douzaine de termes, avant que le facteur 2 manquant ne se révèle. C'est précisément le mécanisme de corrélation numérique décrit en introduction, ici mesuré et non plus supposé.

6. Ce qui, en revanche, est une vraie relation : l'inversion de Riemann-von Mangoldt par W

La fonction de comptage des zéros, $N(T) \sim \dfrac{T}{2\pi} \ln \dfrac{T}{2\pi e}$, s'inverse exactement via la fonction W (en posant $u = T/2\pi$, l'équation $u(\ln u - 1) = k$ se résout par $u = k/W(k/e)$) :

$$T_k \approx \frac{2\pi k}{W(k/e)}.$$

Contrairement à la formule du "rythme de valse", celle-ci **converge réellement** vers les vraies hauteurs z_k quand k grandit :

k	z_k réel	T_k prédit	écart relatif
1	14.135	22.564	59.6%
5	32.935	38.568	17.1%
10	49.774	54.312	9.1%
15	65.113	68.125	4.6%
20	77.145	80.836	4.8%
25	88.809	92.803	4.5%
30	101.318	104.220	2.9%

L'écart décroît régulièrement (comme attendu d'une formule asymptotique du premier ordre, sans les termes correctifs d'ordre supérieur de la formule complète de Riemann–von Mangoldt). C'est le contraste exact avec la section précédente : ici, l'écart se resserre avec k ; là-bas, il se stabilisait sur un facteur constant et faux. C'est ce contraste qui distingue les deux situations.

7. Conclusion

Le "rythme de valse" tel qu'énoncé ne survit pas à l'extension du test que le document d'analyse de gemini de votre travail suggérait sans la faire : le facteur 2 manquant est net et stable dès une trentaine de termes. En revanche, l'intuition d'utiliser W pour relier l'indice k d'un zéro à sa hauteur z_k est juste et connue - mais non pas sous la forme que vous aviez empiriquement construite à partir des articles de França et Le Clair.

8. Annexe

Le programme initial

```
import math
from math import exp, sin, cos, pi
from scipy.special import lambertw

with open("zeros100000", 'r') as f:
    lines = f.readlines()
    zeros = [float(line) for line in lines]
    for x in range(100000):
        res = (2*x/3)-cos(2*pi*x/3+pi/3)/3+(3**0.5)*sin(2*pi*x/3+pi/3)/9
        # formule fournie par Paul Barry en 2004 pour OEIS A004396
        print(f'{x:3} —> {round(res,2)}
              {round(exp(lambertw(-((zeros[x-1]/2)**2)*(2*pi)/180).real),5)}')
```

Son résultat jusqu'à 100000

Voir https://denisevellachemla.eu/lavalse.txt.

Le programme de l'ia claude

```
# -*- coding: utf-8 -*-
"""
Explorations et v rifications num riques.
mod lisation du rythme de valse.
"""

import math

def simuler_valse_g(nb_termes=15):
    print("\n—— 2) Simulation du Rythme de Valse avec g(k) ——")
    suite = []
```

```python
    for k in range(1, nb_termes + 1):
        valeur = 2*k/3 + 4/5
        suite.append(int(valeur))
    print(f"Parties entieres de g(k) pour k=1..{nb_termes} :")
    print(suite)

if __name__ == "__main__":
    simuler_valse_g()
```

Son résultat

```
    --- 2) Simulation du Rythme de Valse avec g(k) ---
Parties entieres de g(k) pour k=1..15 :
[1, 2, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, 9, 10, 10]
```