

Quand gemini t'explique ce que tu fais "au feeling"!, Denise Vella-Chemla, mai 2026.

Bonjour, en décembre 2024, à cette adresse [lien](#) j'avais proposé une formule ultra-simple pour calculer les zéros de  $\zeta$  et ils étaient tout de même très bien approchés.

Pourrais-tu m'expliquer pourquoi cette formule, dont je te donne aussi l'image, peut marcher... (Il te faut utiliser un langage très simple : je suis informaticienne, pas mathématicienne ;).

$$\text{Valeurs des zéros non triviaux de } \zeta \approx \frac{2\pi x}{W\left(\frac{x}{e}\right)}$$

FIGURE 1 : Formule qui calcule les parties imaginaires des zéros non triviaux de  $\zeta$ .

RÉPONSE DE L'IA GEMINI : Bonjour! C'est une excellente question et votre intuition de lier les zéros de la fonction zêta ( $\zeta$ ) de Riemann à la fonction  $W$  de Lambert est très élégante.

Pour expliquer pourquoi cette "formule magique" fonctionne de manière si surprenante avec des mots simples (et des concepts d'informaticienne), il faut regarder comment sont répartis ces zéros.

### 1. Le "compteur" de zéros (l'algorithme de base)

En mathématiques, on sait que les zéros non triviaux de la fonction zêta se situent sur une ligne verticale (l'axe critique) et apparaissent à des hauteurs particulières, qu'on appelle  $t$ .

Il existe une formule célèbre (la formule de Riemann-von Mangoldt)<sup>1</sup> qui permet de compter approximativement combien il y a de zéros entre la hauteur 0 et une hauteur  $t$  donnée. Appelons ce compteur  $N(t)$ . Quand  $t$  devient grand, ce compteur se comporte à peu près comme cela :

$$N(t) \approx \frac{t}{2\pi} \ln\left(\frac{t}{2\pi e}\right)$$

Pour une informaticienne, vous pouvez voir  $N(t)$  comme une fonction qui prend en entrée une hauteur  $t$  et renvoie son "index" (le numéro du zéro).

### 2. Le problème inverse (trouver la hauteur à partir de l'index)

---

1. Il se trouve qu'on avait traduit l'article de von Mangoldt qui donne la preuve de la formule en question, depuis l'allemand en n'étant pas germaniste avec les outils google traduction en avril 2024, la traduction est consultable à cette adresse.

Votre objectif est inverse : vous ne voulez pas donner une hauteur  $t$  pour trouver l'index. Vous voulez donner un index (que vous avez appelé  $x$  dans votre formule) et trouver la hauteur estimée du  $x$ -ième zéro (que nous allons appeler  $t$ ).

En informatique, cela revient à chercher la fonction inverse de  $N(t)$ . On veut résoudre l'équation :

$$x = \frac{t}{2\pi} \ln \left( \frac{t}{2\pi e} \right)$$

On cherche à isoler  $t$  en fonction de  $x$ .

### 3. Pourquoi la fonction $W$ de Lambert apparaît ?

C'est là que la "magie" opère. La fonction  $W$  de Lambert a été inventée précisément pour résoudre des équations où une variable est à la fois "toute seule" et "bloquée" à l'intérieur d'un logarithme ou d'une exponentielle (par exemple, résoudre  $y = u \cdot \ln(u)$  ou  $y = u \cdot e^u$ ).

Si on manipule un peu notre équation mathématique pour l'adapter au profil de la fonction  $W$ , on s'aperçoit que la solution exacte pour isoler  $t$  est :

$$t \approx \frac{2\pi x}{W \left( \frac{x}{e} \right)}$$

### En conclusion : pourquoi ça marche ?

Votre formule n'est pas un hasard : elle est la réponse algébrique exacte au problème d'inversion du compteur de zéros.

Comme la formule de Riemann-von Mangoldt donne une excellente approximation globale de la position des zéros (la courbe lisse suit de très près la réalité, même s'il y a de petites fluctuations locales d'un zéro à l'autre), l'inversion que vous avez trouvée grâce à la fonction de Lambert donne instantanément une estimation ultra-rapide et très propre de la valeur du  $x^{\text{ième}}$  zéro.

C'est une très jolie façon d'exprimer de manière condensée et calculable un comportement asymptotique complexe !

1er juin : petit pont : passer d'un indice à un zéro de  $\zeta$  par la fonction ci-dessus puis utiliser la formule de Riemann pour calculer un nombre premier, pour voir.

```

import numpy as np
from scipy.special import lambertw
from mpmath import li
from scipy.optimize import brentq
from sympy import isprime
import time

def zeros_de_zeta_lambert(x):
    w_0 = lambertw(x / np.e, k=0).real
    return (2 * np.pi * x) / w_0

def R_riemann_approx(p):
    if p <= 2:
        return 0
    term1 = li(p) - li(2)
    term2 = 0.5 * (li(np.sqrt(p)) - li(2))
    return float(term1 - term2)

def pont_complet_via_t(t):
    indice_cible = (t / (2 * np.pi)) * np.log(t / (2 * np.pi * np.e))
    def equation_a_resoudre(p):
        return R_riemann_approx(p) - indice_cible
    approx_grossiere = indice_cible * np.log(max(2, indice_cible))
        if indice_cible > 1 else 2
    borne_inf = max(2.001, approx_grossiere * 0.5)
    borne_sup = approx_grossiere * 2.0
    try:
        p_estime = brentq(equation_a_resoudre, borne_inf, borne_sup)
        return p_estime
    except ValueError:
        return approx_grossiere

tic = time.time()
n = 1000000002
lespremiers = []
for k in range(n):
    if isprime(k):
        lespremiers.append(k)
print(f"{'Indice (x)':<10}|{'Vrai Premier':<12} |
      {'Zero calcule (t)':<18}|{'Premier via le pont t':<22}|{'erreur'}")
print("-" * 85)
indices_tests = [10,100,1000,10000,100000,1000000]
for x in indices_tests:
    vrai_p = lespremiers[x]
    t_calc = zeros_de_zeta_lambert(x)
    p_pont = pont_complet_via_t(t_calc)
    erreur = abs(vrai_p - p_pont) / vrai_p * 100
    print(f"{'x':<10} | {'vrai_p':<12} | {'t_calc':<18.2f} | {'p_pont':<22.2f}
          | {'erreur':.2f}%")
tac = time.time()
print(tac-tic, ' s. ')

```

*Résultat du programme*

Indice( $x$ )	Vrai Premier	Zéro calculé( $t$ )	Premier via le pont t	erreur
10	31	54.31	28.00	9.67%
100	547	238.37	525.96	3.85%
1000	7927	1421.11	7891.71	0.45%
10000	104743	9878.80	104694.38	0.05%
100000	1299721	74921.81	1299571.19	0.01%
1000000	15485867	600270.39	15483689.87	0.01%

Exécution en 994.349011182785 s.

Et avec une fonction plus efficace que la fonction `isprime` du package `sympy`, on parvient à ajouter en plus des lignes précédentes la dernière ligne ci-dessous en 141 secondes :

Indice( $x$ )	Vrai Premier	Zéro calculé( $t$ )	Premier via le pont t	erreur
10000000	179424691	4992381.75	179430488.63	0.00%