

```
from math import pi, sin, cos, floor, log
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Define first 100 non-trivial Riemann
zero imaginary parts
gamma = [14.134725, 21.022040,
25.010858, 30.424876, 32.935061,
37.586178,
        40.918719, 43.327073, 48.005150,
49.773832, 52.970321, 56.446247,
        59.347045, 60.831780, 65.112544,
67.079811, 69.546402, 72.067158,
        75.704690, 77.144840, 79.337376,
82.910380, 84.735493, 87.425274,
        88.809111, 92.491899, 94.651345,
95.870634, 98.831194, 101.317852,
        103.725539, 105.446623,
107.168611, 111.029535, 111.874659,
114.320221,
```

116.226680, 118.790782,
121.370125, 122.946829, 124.256818,
127.516084,
129.578704, 131.087688,
133.497737, 134.756509, 138.116042,
139.736208,
141.123707, 143.111845,
146.000982, 147.422765, 150.053520,
151.879449,
153.024693, 156.112909,
157.597591, 158.849988, 161.188964,
163.030709,
165.537069, 167.184440,
169.094515, 170.458347, 173.411536,
174.754192,
176.441434, 178.377407,
179.916484, 182.207078, 184.874468,
185.598783,
187.228922, 189.416158,
192.026656, 193.079726, 195.265397,
196.876481,

198.015310, 201.264751,
202.493594, 204.189671, 205.394697,
207.906258,
209.576509, 211.690862,
213.347919, 214.547045, 216.169538,
219.067596,
220.714918, 221.430705,
224.007000, 224.983324, 227.421444,
229.337413,
231.250188, 231.987235,
233.693404, 236.524229]

Constants

tau_alpha = 0.007297 # Fine-structure
constant torsion
tau_psi = 0.006944 # Harmonic 144Hz
torsion
threshold = 0.001 # Resonance
threshold

Storage

```
resonant_indices_alpha = []
resonant_indices_psi = []
tau_vals = []

# Compute torsions and find resonances
for i, g in enumerate(gamma):
    tau = g / (g**2 + 1)
    tau_vals.append(tau)
    if abs(tau - tau_alpha) <= threshold:
        resonant_indices_alpha.append(i + 1)
    if abs(tau - tau_psi) <= threshold:
        resonant_indices_psi.append(i + 1)

# Union of unique resonant indices
resonant_indices =
sorted(set(resonant_indices_alpha +
resonant_indices_psi))

# Function to compute sync hits
def hit_count(g_n, omega_target):
    delta = abs(g_n - omega_target)
```

```
return floor(5 * delta)
```

```
# Total phase-sync hits
total_hits_alpha =
sum(hit_count(gamma[i-1], 137.036) for i
in resonant_indices_alpha)
total_hits_psi =
sum(hit_count(gamma[i-1], 144.0) for i in
resonant_indices_psi)
```

```
# Print Codex Summary
import pprint
summary = {
    "Total Resonant Strands": len(resonant_indices),
    "Alpha-resonant Indices": resonant_indices_alpha,
    "Psi-resonant Indices": resonant_indices_psi,
    "Total Hits (Alpha)": total_hits_alpha,
    "Total Hits (Psi)": total_hits_psi,
```

```
"Combined Total Hits": total_hits_alpha  
+ total_hits_psi  
}  
pprint.pprint(summary)
```