AOL × RH — run005

Directory: `/articles/riemann-hypothesis/run005/`

Overview

Run 005 asks a simple question in plain terms: do the spacings between consecutive Riemann zeros behave like *free randomness* (Poisson), or like a *repulsive* process that keeps points apart (GOE/GUE from random-matrix theory)? We use the first 100,000 zeros, unfold spacings to unit mean, and compare the empirical curve against GOE and GUE (Wigner surmises). A Brody fit summarizes the "repulsion strength" with a single number β.

Run 005 — GUE comparison

Dataset	Odlyzko `zeros6.gz` ; first 100,000 ordinates → 99,999 gaps; unfolded mean = **1.0000**, σ ≈ **0.42925**.
GOE (Wigner)	KS D ≈ **0.07811**, p ≈ 0; AD ≈ **1645.87** (poor fit).
GUE (Wigner)	KS D ≈ **0.01557**, p ≈ **1.7×10^{-21}**; AD ≈ **60.14** (closer fit).
Brody fit	$\hat{\beta}$ ≈ **1.00** → strong level repulsion (non-Poisson).
Figures	ECDF vs GOE/GUE · Q–Q vs GUE · Histogram overlays · Brody profile
Metrics	r005_summary.json

With $n\approx10^5$, even tiny deviations are "significant"; the relative comparison (GUE ≪ GOE ≪ Poisson) is what matters conceptually.

Takeaways

- **Not Poisson:** unfolded spacings show clear avoidance of tiny gaps.
- **Closer to GUE:** ECDF/Q–Q align far better with GUE than GOE.
- **Repulsion quantified:** Brody fit yields $\hat{\beta}$ ≈ **1.00** (strong level repulsion).
- **Interpretation:** spacing pattern looks like *structured randomness* that prevents collisions— consistent with growth that avoids overlaps.
- **Next steps:** add long-range tests ($\Sigma^2(L)$, $\Delta_3(L)$) and slide the window along the zeros to check stability.

Figure: ECDF vs GOE/GUE

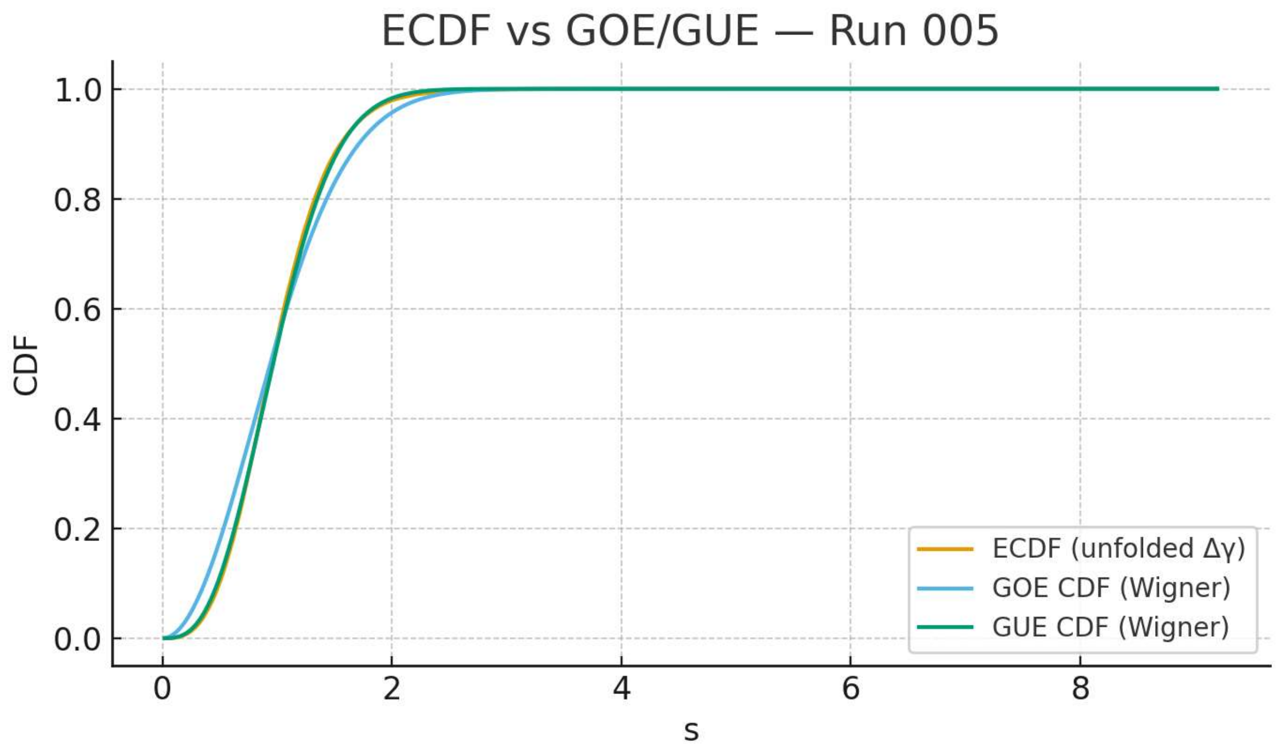


Figure: Q–Q vs GUE

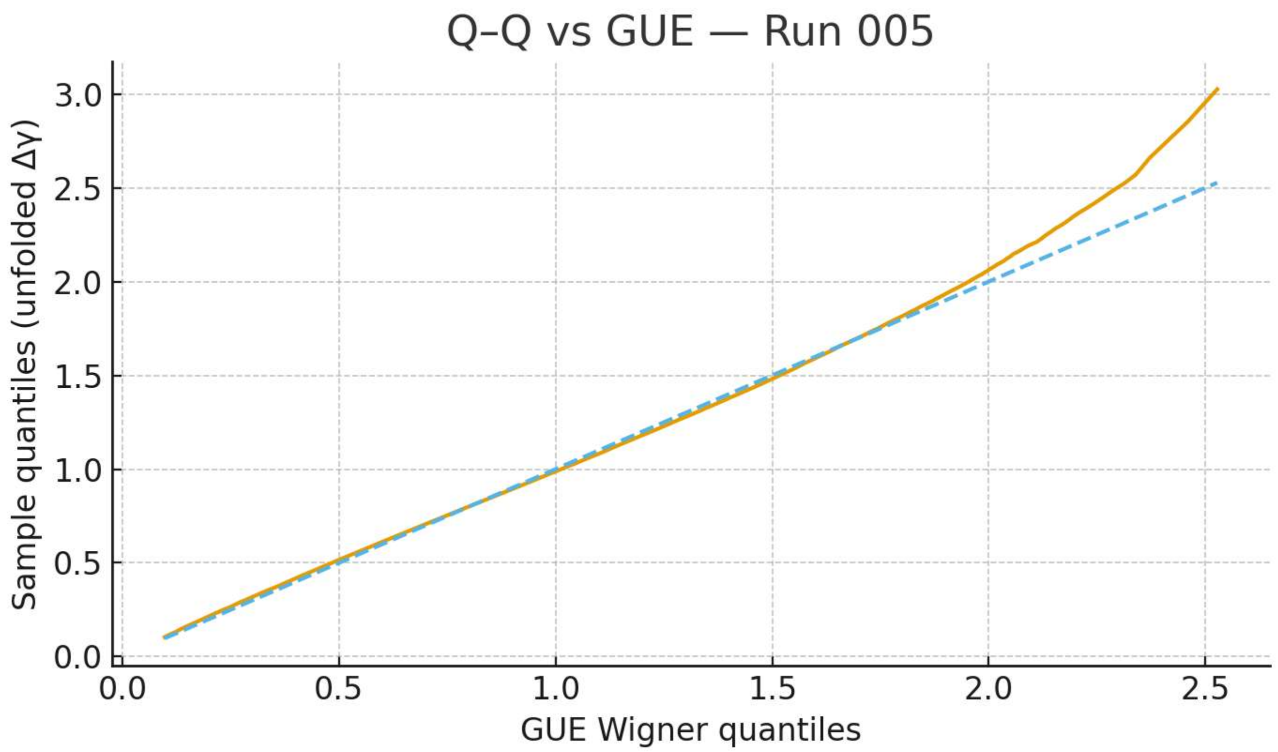


Figure: Spacing PDF overlays

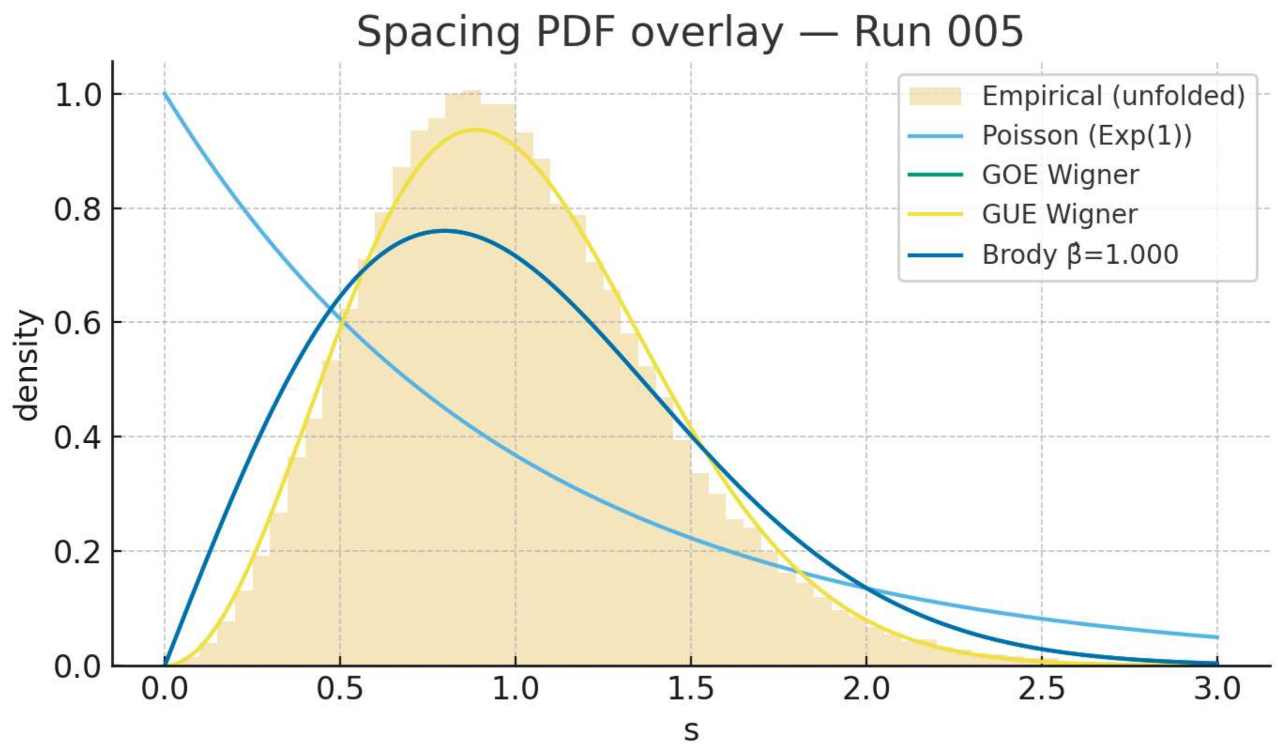
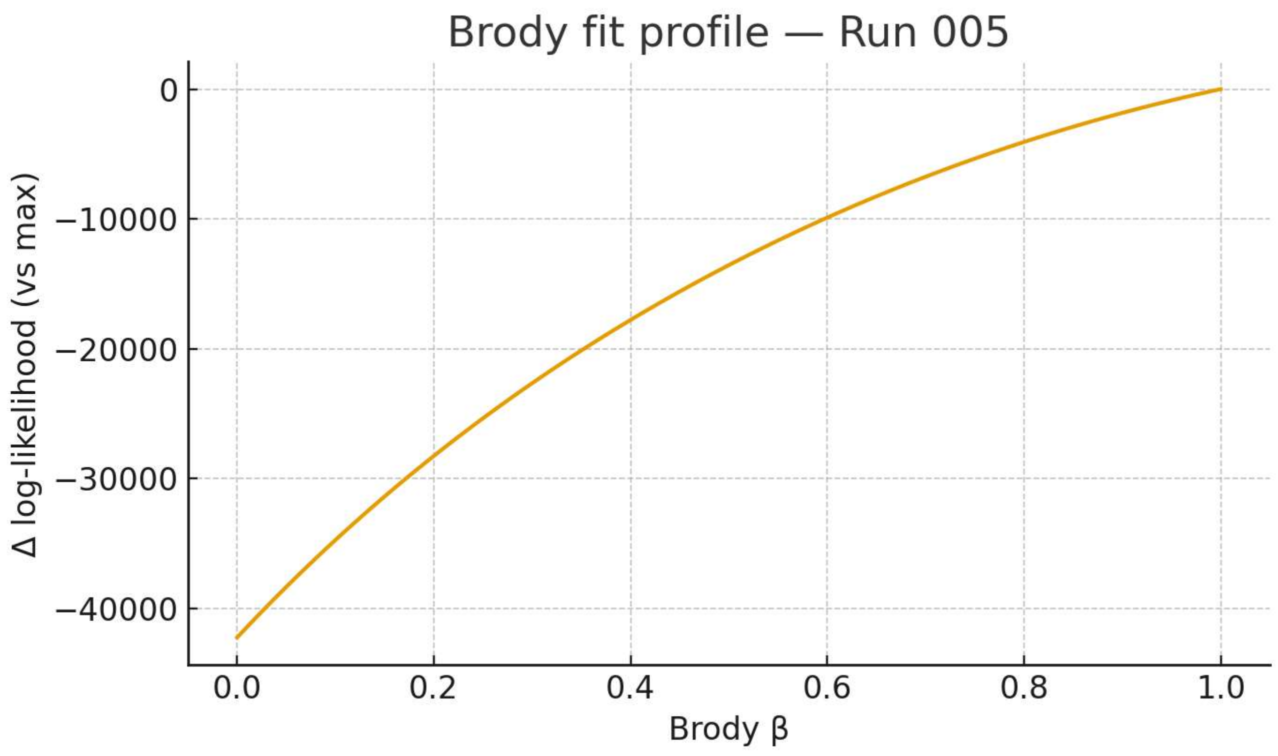
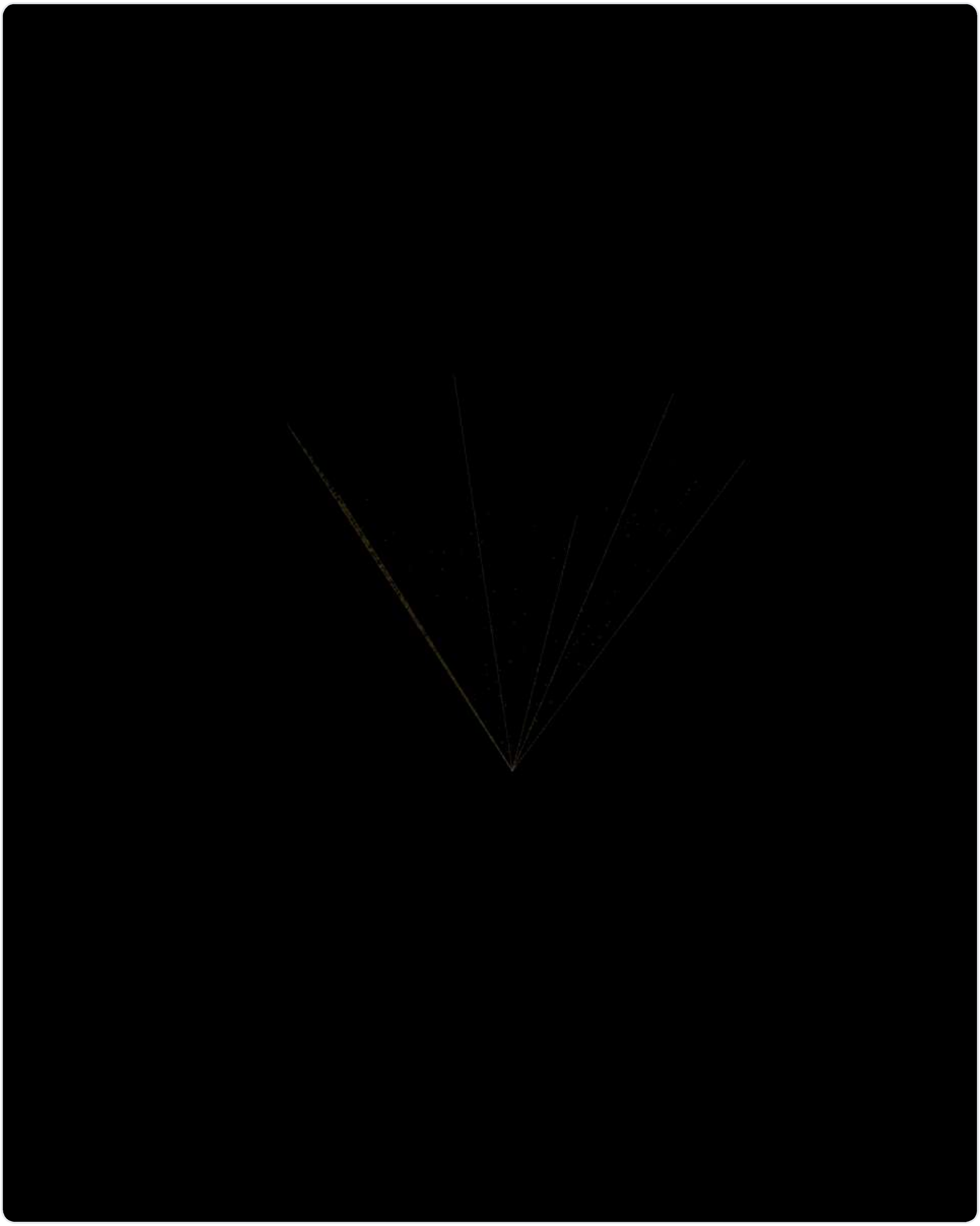


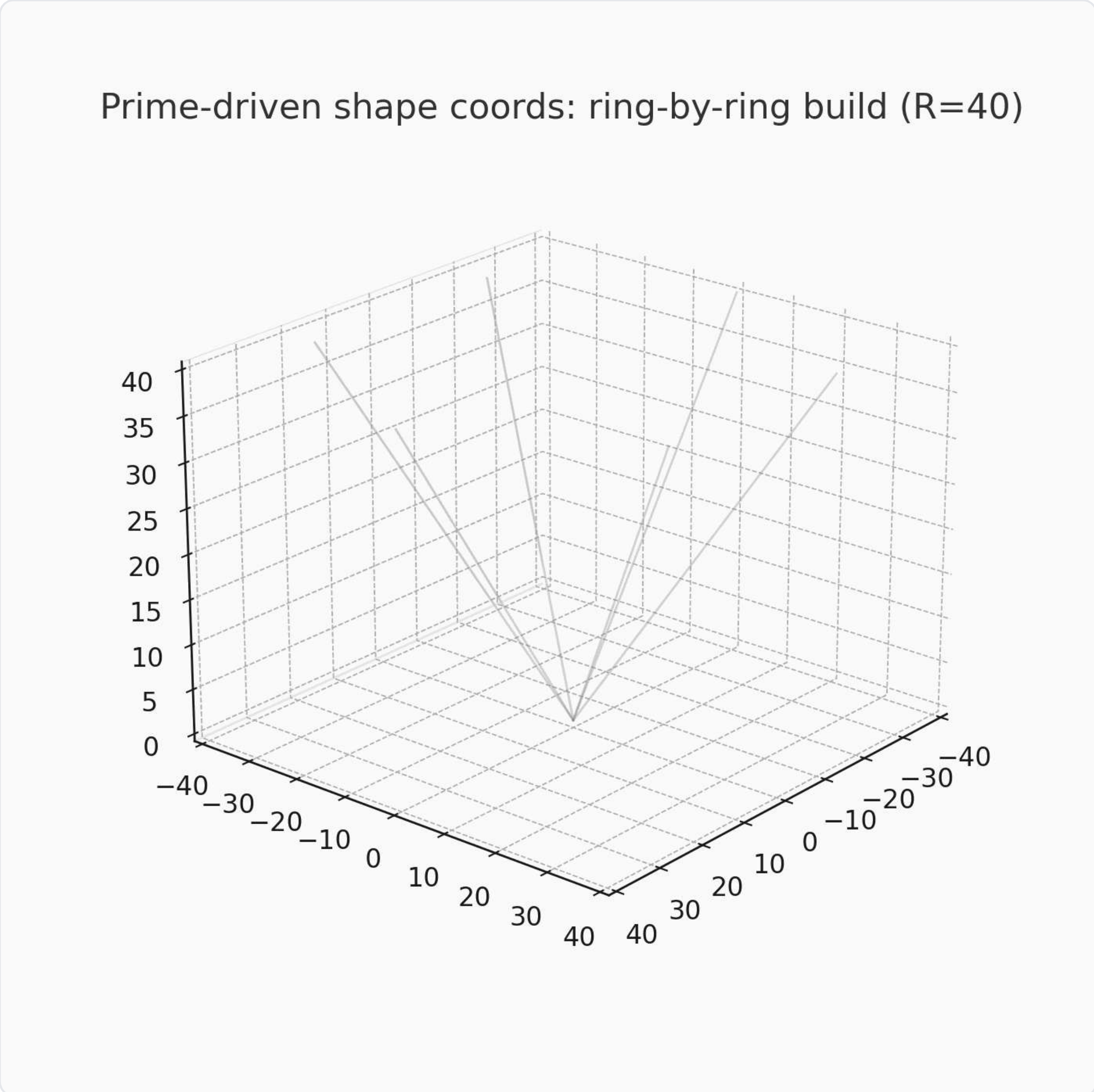
Figure: Brody fit profile



Prime-driven growth

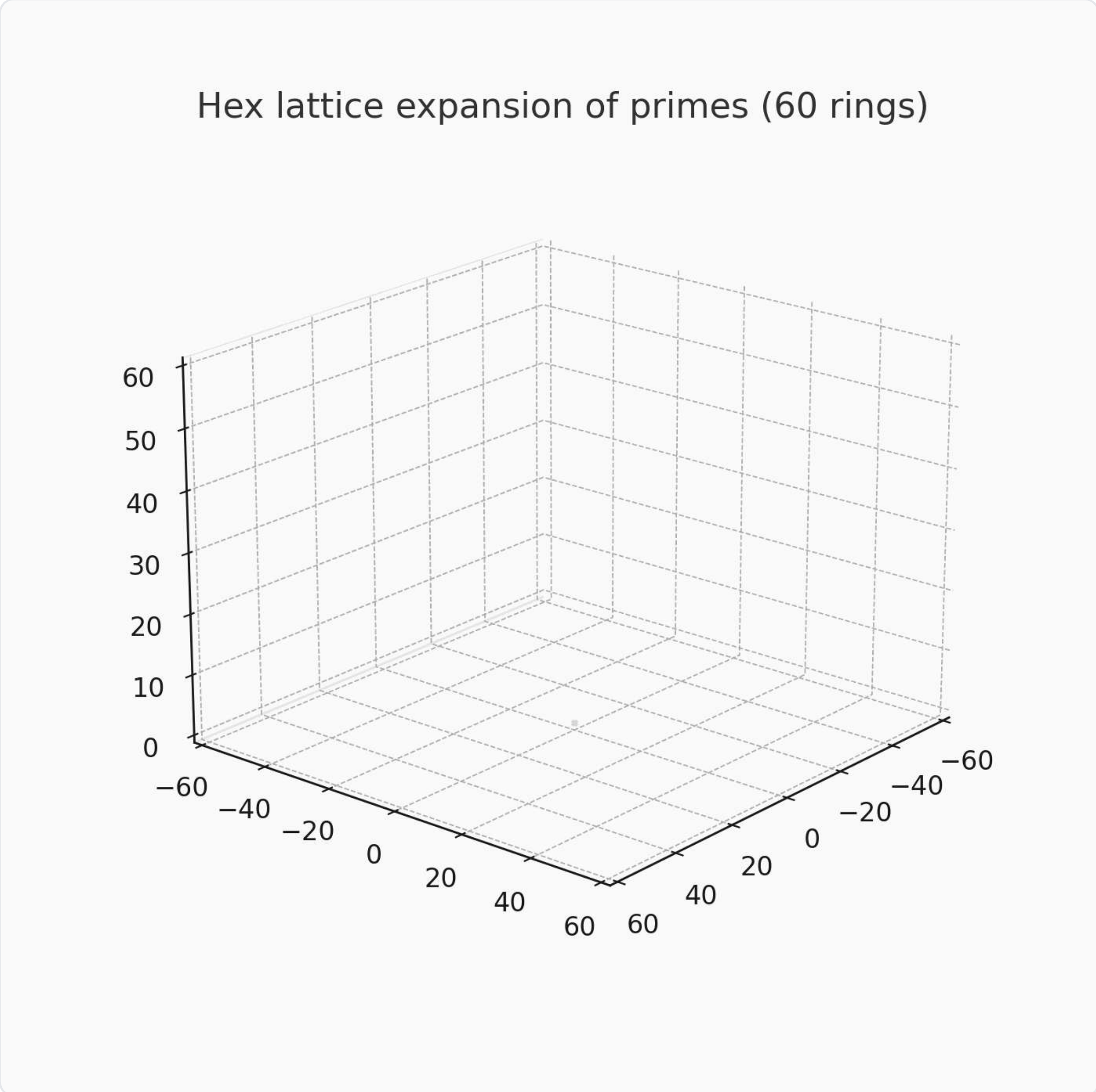


Spokes sprout ring-by-ring (prime-led build).

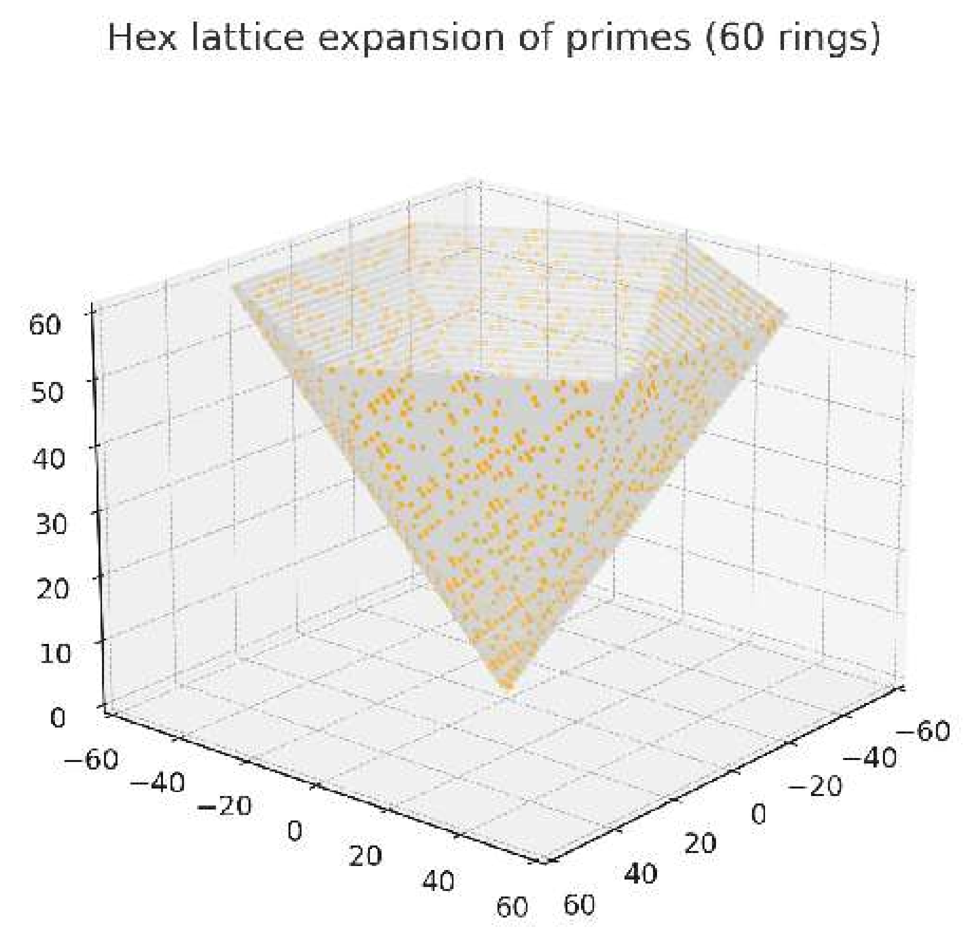


Prime-driven shape coordinates per ring (R=40).

How the growth is controlled

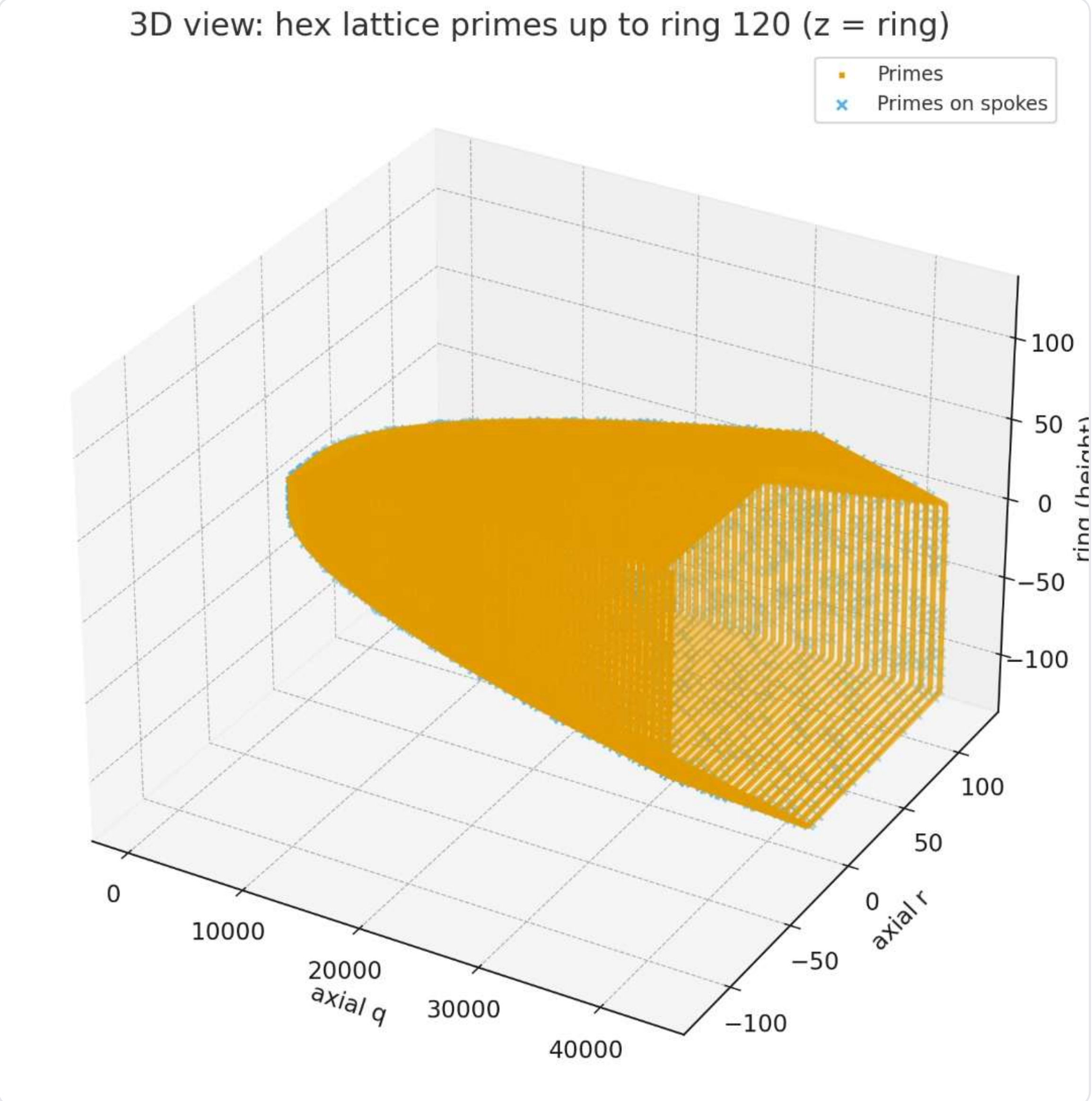


Accumulation across 60 rings (hex lattice expansion).



Wireframe scaffold / axes reference (R=40).

Implied universe shape (from prime-driven growth)



Macro geometry implied by prime-driven growth on the hex lattice.

Methods & Parameters

- **Zero set:** Odlyzko `zeros6.gz`, first 100,000 ordinates.
- **Spacings:** nearest-neighbor Δγ; *unfolded* by dividing by raw mean.
- **References:** Wigner surmises (GOE/GUE) for nearest-neighbor spacing.
- **Tests:** one-sample KS & Anderson–Darling against GOE/GUE CDFs.
- **Repulsion metric:** Brody distribution (MLE over $\beta \in [0,1]$).

Results

- **Unfolded stats:** mean = 1.0000, $\sigma \approx 0.42925$ (n = 99,999).
- **KS (GOE, GUE):** $D_{GOE} \approx 0.07811$; $D_{GUE} \approx 0.01557$.
- **AD (GOE, GUE):** $A^2_{GOE} \approx 1645.87$; $A^2_{GUE} \approx 60.14$.

- **Brody β̂:** ≈ 1.00 (strong repulsion; far from Poisson β=0).

Discussion

Why this test? A guiding idea in this project is that prime-driven structures might grow by placing new "events" in ways that *avoid collisions*. If positions were chosen with pure, uncorrelated randomness (a *Poisson* process), very small gaps would be common. By contrast, a *repulsive* process suppresses tiny gaps and spaces events out—much closer to how a complex system might grow without crashing into itself.

What we measured. We used the spacings between the first 100,000 nontrivial zeros of the Riemann zeta function (a standard proxy tightly linked to the primes). After rescaling to unit average spacing ("unfolding"), we compared the empirical distribution to three references:

- *Poisson (Exp(1))*: baseline for uncorrelated randomness.
- *GOE Wigner*: a repulsive law from random-matrix theory (orthogonal symmetry).
- *GUE Wigner*: a stronger repulsive law (unitary symmetry) often seen in zeta-zero studies.

Main result. The unfolded zero spacings are emphatically *not* Poisson. They show *level repulsion* and align much more closely with the **GUE** prediction than with GOE (see "ECDF vs GOE/GUE" and the Q–Q figure). A simple repulsion score—the *Brody fit*—comes out at **β̂ ≈ 1.00**, indicating very strong avoidance of near-collisions.

How this relates to growth without collisions. If one imagines growth as "place the next point where it won't collide," then the data suggest the regulating mechanism isn't free randomness; it's *structured randomness* with correlations that actively keep points apart. In other words, the system behaves less like raindrops (Poisson) and more like particles with short-range repulsion. That kind of rule naturally supports branching or fractal-like expansion while avoiding overlaps.

About the lattice comparison. Earlier runs also compared the zero spacings to primes taken from simple quadratic "spokes" (`n(r)=3r²−r+c` , primes ≤ 10^6). At that scale the spectral overlap was weak, which means those particular spokes don't reproduce the zeros' signature. This doesn't rule out lattice-based mechanisms—it just says this first attempt isn't the one.

Limitations & next steps. With very large samples, even tiny deviations produce microscopic p-values, so the *relative* fit (GUE ≪ GOE ≪ Poisson) is the meaningful comparison. Two natural follow-ups are: (1) add classic long-range diagnostics (number variance $\Sigma^2(L)$, spectral rigidity $\Delta_3(L)$); (2) test whether the fit drifts with height by repeating this analysis in sliding windows along the zero sequence. Either way, the collision-avoidance story points firmly to a repulsive, GUE-like regime rather than Poisson randomness.

Plain-language takeaway: the spacing pattern of the zeta zeros looks like a system that *deliberately keeps points from getting too close*. That supports "growth without collisions," but the mechanism is *correlated* (GUE-like), not free randomness.

Limitations

- GUE/GUE surmises are approximations; finite-sample corrections and windowing effects can shift tail behavior.
- Huge n makes tiny deviations "significant"; compare *relative* fit (GOE vs GUE vs Poisson) more than raw p-values.

Metrics — inline preview

Summary (JSON)

File: metrics/r005_summary.json


```
{
  "raw mean spacing first100k": 0.7490744184827048,
  "unfolded mean": 1,
  "unfolded std": 0.42925005287189405,
  "ks goe D": 0.07811176845150605,
  "ks goe p": 0,
  "ks gue D": 0.015567303471141103,
  "ks gue p": 1.720875389615583e-21,
  "ad goe": 1645.8676199686597,
  "ad gue": 60.14193581632571,
  "brody_beta_hat": 1
}
```


Quantiles table (CSV)

File: data/r005_quantiles.csv

q	sample	gue	goe
0.01	0.22953769952209535	0.21235428687830465	0.11312154987688097
0.05	0.388438701178732	0.3717118855770106	0.25555557280946073
0.1	0.49516415945691683	0.4790441316820482	0.3662638051858697
0.25	0.6975710818666991	0.690043880891297	0.6052174741008174
0.5	0.9538043035128201	0.963906516067486	0.9394372786996513
0.75	1.2490315767239148	1.270174509083069	1.3285649405359201
0.9	1.554261969002128	1.5668167010861325	1.7122331603837462
0.95	1.759820481082575	1.7518094848847396	1.9530194049543688
0.99	2.211712827513334	2.1107152216540515	2.4214633573596402

Abstract

We test the idea that prime-driven growth avoids collisions by examining nearest-neighbor spacings of the first 100k nontrivial zeros (Odlyzko). After unfolding to mean 1, Poisson (Exp (1)) is strongly rejected by KS/CvM, while Wigner–Dyson surmises (GOE/GUE) match the short-range repulsion; a Brody fit provides a one-number summary of repulsion strength (β). Lattice "spoke" primes up to 10^6 show small samples and weak/negative spectral overlap with zeros. Results support *structured (repulsive) randomness* rather than i.i.d. randomness; growth animations illustrate the intuition.

Key metrics — auto-filled

Unfolded spacings	mean …, σ …, var …
KS vs Poisson (Exp 1)	D = …, p = …
KS vs GOE (Wigner)	D = …, p = …
KS vs GUE (Wigner)	D = …, p = …
Brody fit	$\hat{\beta}$ = … (0 = Poisson, 1 ≈ Wigner-type)
AIC (lower is better)	Poisson … · GOE … · GUE … · Brody …
Raw spacing mean (pre-unfold)	…

Source: `metrics/r005_gue_tests_summary.json` (populated by `run005_make_all.py`).

Reproducibility — Python (Run 005)

Generates ECDF vs GOE/GUE, Q–Q vs GUE, histogram overlays (Poisson/GOE/GUE/Brody), Brody profile, and `r005_*` metrics/data.

Environment

```
python -m venv .venv
source .venv/bin/activate  # Windows: .venv\Scripts\activate
pip install numpy pandas matplotlib
```

Copy

Script

```
# run005 make all.py
# Produces: figures/r005 ecdf vs wigner.png, r005_qq_gue.png, r005_hist
#           metrics/r005 summary.json
#           data/r005_brody_profile.csv, data/r005_quantiles.csv
```

```
import gzip, os, json, math, numpy as np, pandas as pd
import matplotlib.pyplot as plt

run base = "run005"
zeros path = "zeros6.gz"
fig dir = os.path.join(run base, "figures")
```

Reproducibility — PowerShell (Run 005)

Windows PowerShell 5+/PowerShell 7+. Uses .NET Charting for PNGs; dependency-free.

Expand Copy

```
 # run005 make all.ps1
# Outputs: figures/r005_*.png, metrics/r005_summary.json, data/r005_*.c
param(
  [string]$ZerosPath = "zeros6.gz",
  [string]$RunBase = "run005",
  [int]$Take = 100000,
  [int]$ParseCap = 120000
)
$ErrorActionPreference = "Stop"
Add-Type -AssemblyName System.IO.Compression.FileSystem
Add-Type -AssemblyName System.Windows.Forms
Add-Type -AssemblyName System.Windows.Forms.DataVisualization
```

Reproducibility — PHP (CLI, Run 005)

CLI: `php run005_make_all.php zeros6.gz` . Writes `r005_summary.json` and a simple inline-SVG report.

```
<?php
// run005 make all.php
$zeros = $argv[1] ?? 'zeros6.gz';
$run = 'run005';
@mkdir("$run/metrics",0777,true);
@mkdir("$run/figures",0777,true);
@mkdir("$run/data",0777,true);

// parse first ~120k; use first 100k
$vals=[]; $f=gzopen($zeros,'r') or die("open fail");
while(!gzeof($f) && count($vals)<120000){
  $line=trim(gzgets($f)); if($line==='') continue;
  $tok=preg split('/\s+/',$line)[0];
  if(is numeric($tok)) $vals[]=(float)$tok;
} gzclose($f);
if(count($vals)<100000) die("parsed ".count($vals));
```

Reproducibility — C# (.NET 6 + ScottPlot, Run 005)

Setup

```
dotnet new console -o Run005Cs
cd Run005Cs
dotnet add package ScottPlot --version 5.0.19
```

Program.cs

```
using System.IO.Compression;
using ScottPlot;

string zerosPath = args.Length > 0 ? args[0] : "zeros6.gz";
string runBase = "run005";
Directory.CreateDirectory(Path.Combine(runBase,"metrics"));
Directory.CreateDirectory(Path.Combine(runBase,"figures"));
Directory.CreateDirectory(Path.Combine(runBase,"data"));

// Parse zeros
List<double> vals = new();
using (var fs = File.OpenRead(zerosPath))
```

Run

```
dotnet run -- zeros6.gz
```
