

Pattern Field Theory × Allen Orbital Lattice × Riemann Hypothesis — run002

Generated: 2025-09-23 19:21 UTC

Preparation & Sources

All paths are relative to this run folder: `run002/`

Zeros (Odlyzko)	zeros6.gz — full gzip copied for provenance; analysis in this run used the first 100,000 rows.
Figures	figures/r002_spacing_ecdf_small.png · figures/r002_spacing_ecdf_full.png · figures/r002_fft_magnitude_comp.png
Metrics	metrics/r002_metrics_demo_unfolded.csv metrics/r002_metrics_first100k_raw.csv (raw = not unfolded) metrics/r002_metrics_first100k_unfolded.csv (unfolded = mean spacing forced to 1.0)
Manifest	manifest.json
External references (original sources)	Odlyzko zeta tables — index.html zeros6.gz zeros6 (text)

Naming convention (per run)

- `figures/rNNN_spacing_ecdf_small.png`,
`figures/rNNN_spacing_ecdf_full.png`, `figures/rNNN_fft_magnitude_comp.png`
- `metrics/rNNN_metrics_demo_unfolded.csv`
- `metrics/rNNN_metrics_first100k_raw.csv` and
`metrics/rNNN_metrics_first100k_unfolded.csv`
- `manifest.json` lists every artifact produced by the run

Inputs & Method

- **Lattice spokes:** $n(r) = 3r^2 - r + c$, $c \in \{2, 3, 4, 5, 6, 7\}$, tested $\leq N_{\max} = 1,000,000$.
- **Spacings:** consecutive differences ($\Delta\gamma$ for zeros; Δn for lattice primes).
- **Unfolding:** each spacing series normalized to mean 1.0 (asserted in code).

- **Comparisons:** Kolmogorov–Smirnov, simple Cramér–von Mises proxy (ECDF L^2), FFT magnitude correlation.

Results (inline preview of CSVs)

Demo sanity run — unfolded spacings

File: [metrics/r002_metrics_demo_unfolded.csv](#) (unfolded by construction)

index	spacing
0	6.336292463660334
1	3.6696911502270613
2	4.980868076531093
3	2.309357185908735
4	4.279002343986177
5	3.065919742869566
6	2.2156730335366444
7	4.303806346321148
8	1.6271776000017693
9	2.9407527665094166
10	3.1978335119272585
11	2.6687170711207853
12	1.365947811208963
13	3.938281362626432
14	1.8098746300900717
15	2.2692506817055333
16	2.319081990235961
17	3.3465109082302824
18	1.3249297100707353
19	2.0171204164387353
20	3.2871462379785403
21	1.6790933854817356

index	spacing
22	2.4745847007795843
23	1.273122258560326
24	3.388145300902045
25	1.9866776270782767
26	1.1217404441554562
27	2.723699340435575
28	2.2877109317974926
29	2.215059180908919
30	1.583388996626268
31	1.5842198621910983
32	3.5520297395088276
33	0.7775092186233091
34	2.249903705813805
35	1.7539324466414339
36	2.358960613605296
37	2.3729809557349615
38	1.4505595072175372
39	1.2051831057271432
40	2.9990586471227063
41	1.8970476546731865
42	1.3882575056593633
43	2.217231875225097
44	1.158064006733181
45	3.0907517314416872
46	1.490544871125694
47	1.276491147395314
48	1.8290766875051503
49	2.6579902767045493

index	spacing
50	1.3080326155221316
51	2.420280587118296
52	0.801993548578575
53	1.9314700630345523
54	2.8411417105484995
55	1.365899973327375
56	1.1521979396303377
57	2.1518453671369544
58	1.6943960446690804
59	2.305837322272308
60	1.5155723070404101
61	1.757259176721774
62	0.7520598014042805
63	3.2195765016355584
64	1.2352354143999784
65	1.5522543197721745
66	1.7810852348678265
67	1.415941904512083
68	2.1073346434037776
69	2.4539839341042
70	0.6663666857289511
71	1.499719066017175
72	2.0122454761280775
73	2.4016439124130367
74	0.9688189855904451
75	2.0108047681921857
76	1.482189722627843
77	1.0477155110980148

index	spacing
78	2.9894694895661162
79	1.1305285853735123
80	1.5603820253536265
81	1.1086169155668433
82	2.3106233046206435
83	1.5366218204243431
84	1.9451933278530487
85	1.5244833521832468
86	1.1031889692343755
87	1.4926855404282366
88	2.666197697982108
89	1.5155278712989977
90	0.6585199465124644
91	2.370177330930457
92	0.8982132347153139
93	2.243056987897773
94	1.7626812461284398
95	1.7597431217866293
96	0.6780788827307013
97	1.569666277364606
98	2.6043442922533484
99	1.145936881106117
100	1.6427949645176632
101	1.3741778082351261
102	1.6321755086367988
103	1.1478097583670455
104	2.8207878363963457
105	0.8877948197444981

index	spacing
106	1.3539557389559425
107	1.3259497817345138
108	1.8906248344102872
109	2.057356157809869
110	0.9884949074078816
111	2.0513357965648966
112	1.1628433310802835
113	0.8562183311015302
114	2.5472898242375113
115	1.825230677486749
116	0.9725465278982572
117	1.202378999765415
118	1.884640258197115
119	1.4017099314139607
120	1.8082987390608773
121	1.957641391772477
122	0.7953876769809557
123	1.654788874920028
124	0.9002215673879567
125	2.9769774057428506
126	0.6863812962154004
127	1.4947872893573055
128	1.6849530659817595
129	1.1449104642392458
130	1.534490743906008
131	2.0851093538621814
132	1.5751622186613734
133	1.2943731090926014

index	spacing
134	0.5592511846795857
135	2.2135275269022636
136	1.7121551088815212
137	1.6642697702666363
138	0.9636245900188762
139	1.9942004058199816
140	0.7953733315044376
141	1.371328863574356
142	2.6587541850590353
143	0.9712184604269132
144	1.1616401188422678
145	1.4328774474769301
146	1.3710959503698026
147	2.0784425694190136
148	1.0288284616897978
149	2.122455297290287
150	0.9056651435095097
151	1.2166109351437373
152	1.2842173010735505
153	2.374538574750503
154	1.4620282763958736
155	0.8465496777275687
156	1.3995214821894548
157	1.9972264439302516
158	0.520695173213382
159	2.420041864147546
160	1.3782830083452007
161	1.396757835752335

index	spacing
162	1.0893144959358718
163	0.9316016656938146
164	2.3982654390738225
165	1.5512651884483484
166	0.8508175052912414
167	1.8800856830220996
168	1.0047799525182717
169	1.3526034305456187
170	1.4814227258513324
171	2.326366990841427
172	1.0430230232056075
173	0.7372679315244468
174	1.6477746738186654
175	1.421949928628843
176	1.8786004399853524
177	1.292310531695582
178	1.3585433741385409
179	1.6383864440127374
180	0.8968683856215841
181	0.9958768312893276
182	2.7701123067388123
183	0.7387055970380829
184	1.8041452484317457
185	0.4583224397069008
186	1.9435696670218037
187	1.3213837995459208
188	1.4825651807228981
189	1.8023254778618236

index	spacing
190	1.3915722776492512
191	0.8327644623095818
192	1.252654936188356
193	1.493370390882787
194	2.4011448199174765
195	0.7258755697414837
196	1.0880413319049294
197	1.9827045354393715
198	0.7350611974063817
199	1.4139231998197974

Showing up to 200 of 4 999 rows.

Load full table (may be heavy)

First 100k zeros — raw spacings (not unfolded)

File: [metrics/r002_metrics_first100k_raw.csv](#)

Unfolding check (**WARN**): n=99 999, mean=0.7491, σ =0.3215, $|\text{mean}-1|=25.09\%$

index	spacing
0	6.887314496999998
1	3.9888179410000006
2	5.414018546000001
3	2.5101854620000026
4	4.651116570999996
5	3.3325408529999976
6	2.408354269
7	4.678077600000002
8	1.768681597000004
9	3.1964889999999997
10	3.4759262189999944
11	2.9007963060000037

index	spacing
12	1.4847345219999966
13	4.280765523000007
14	1.9672664809999958
15	2.466591182000002
16	2.520755962999999
17	3.637533024999996
18	1.4401493700000003
19	2.192534950999999
20	3.573005834
21	1.825112126999997
22	2.689781632000006
23	1.3838365950000053
24	3.6827880629999896
25	2.1594447700000075
26	1.2192901869999986
27	2.960559989999993
28	2.486656788000005
29	2.4076870340000056
30	1.7210850120000032
31	1.7219881319999928
32	3.860924358999995
33	0.8451236340000037
34	2.445561737999995
35	1.9064594060000104
36	2.5641025449999972
37	2.579342135999994
38	1.5767042920000023
39	1.309989260000009

index	spacing
40	3.2598653259999963
41	2.062020320000002
42	1.5089843309999935
43	2.4100486720000163
44	1.2587725499999975
45	3.3595323020000194
46	1.62016689699999
47	1.3874984519999884
48	1.9881384040000114
49	2.8891366789999893
50	1.4217828560000214
51	2.630755077999993
52	0.8717371909999994
53	2.099436198999996
54	3.088215482999999
55	1.4846825240000214
56	1.2523963529999946
57	2.338975966999982
58	1.8417455490000236
59	2.5063595009999915
60	1.6473707899999965
61	1.910075438000007
62	0.8174610629999961
63	3.499560040999995
64	1.3426550030000044
65	1.6872427749999872
66	1.9359734780000224
67	1.5390762440000003

index	spacing
68	2.2905944639999802
69	2.6673893640000017
70	0.7243158299999948
71	1.6301389060000133
72	2.1872360719999904
73	2.6104977050000002
74	1.0530702430000076
75	2.1856700760000008
76	1.6110851609999977
77	1.1388278350000007
78	3.249442268000024
79	1.2288425699999834
80	1.6960772890000158
81	1.2050253989999993
82	2.5115616859999993
83	1.6702508289999969
84	2.1143528780000054
85	1.6570567649999993
86	1.1991254229999981
87	1.6224937249999982
88	2.898057841000025
89	1.6473224899999934
90	0.7157867159999967
91	2.5762947000000054
92	0.9763244149999935
93	2.438119610000001
94	1.915969025999999
95	1.9127753939999934

index	spacing
96	0.737046552999999
97	1.7061689260000037
98	2.8308254869999985
99	1.2455908150000141
100	1.7856570919999797
101	1.4936802230000126
102	1.7741141379999874
103	1.2476265630000114
104	3.066091577999998
105	0.9649999849999915
106	1.4716995850000103
107	1.4412581500000101
108	2.055038952999979
109	2.2362697070000195
110	1.0744572389999973
111	2.229725798000004
112	1.2639674979999995
113	0.9306775150000135
114	2.7688093999999523
115	1.9839579340000455
116	1.05712194299997
117	1.3069413009999948
118	2.0485339410000165
119	1.5236066179999739
120	1.9655535460000237
121	2.1278834610000104
122	0.8645568539999999
123	1.7986940270000105

index	spacing
124	0.9785073979999765
125	3.235863837000011
126	0.7460709679999695
127	1.624778248000041
128	1.8314813819999927
129	1.2444751379999843
130	1.6679344280000237
131	2.266436399999975
132	1.712142809999989
133	1.4069354800000156
134	0.6078852599999891
135	2.4060221830000046
136	1.8610489920000077
137	1.808999407999977
138	1.0474241280000456
139	2.167621750999956
140	0.8645412609999994
141	1.490583526000023
142	2.8899670189999824
143	1.0556783830000427
144	1.2626596509999786
145	1.5574845499999697
146	1.490330358000051
147	2.2591898529999526
148	1.1182983140000147
149	2.3070300530000054
150	0.9844243630000165
151	1.3224108860000001

index	spacing
152	1.3958964939999987
153	2.581035209999982
154	1.5891704179999806
155	0.9201680480000505
156	1.5212278549999496
157	2.1709109420000345
158	0.5659763079999607
159	2.6304955950000135
160	1.4981424229999902
161	1.5182238740000003
162	1.1840443860000391
163	1.0126163989999668
164	2.6068254300000015
165	1.6861676259999854
166	0.9248070180000241
167	2.04358328699999
168	1.0921584779999876
169	1.4702296760000308
170	1.6102514639999868
171	2.5286744879999787
172	1.1337272750000125
173	0.8013828499999818
174	1.7910698510000316
175	1.5456067430000076
176	2.0419688829999814
177	1.4046935349999785
178	1.4766861740000081
179	1.7808651940000004

index	spacing
180	0.974862614000017
181	1.0824811159999967
182	3.0110091600000146
183	0.8029455390000066
184	1.9610388559999592
185	0.49817946400003166
186	2.112588018999986
187	1.436295097000027
188	1.6114932699999827
189	1.9590608330000236
190	1.5125873649999448
191	0.905184031000033
192	1.3615893759999835
193	1.6232381319999831
194	2.6099552100000096
195	0.7889997760000256
196	1.1826605040000118
197	2.1551261669999917
198	0.7989842119999935
199	1.5368819870000152

Showing up to 200 of 99 999 rows.

Load full table (may be heavy)

Figure: ECDF (small demo run)

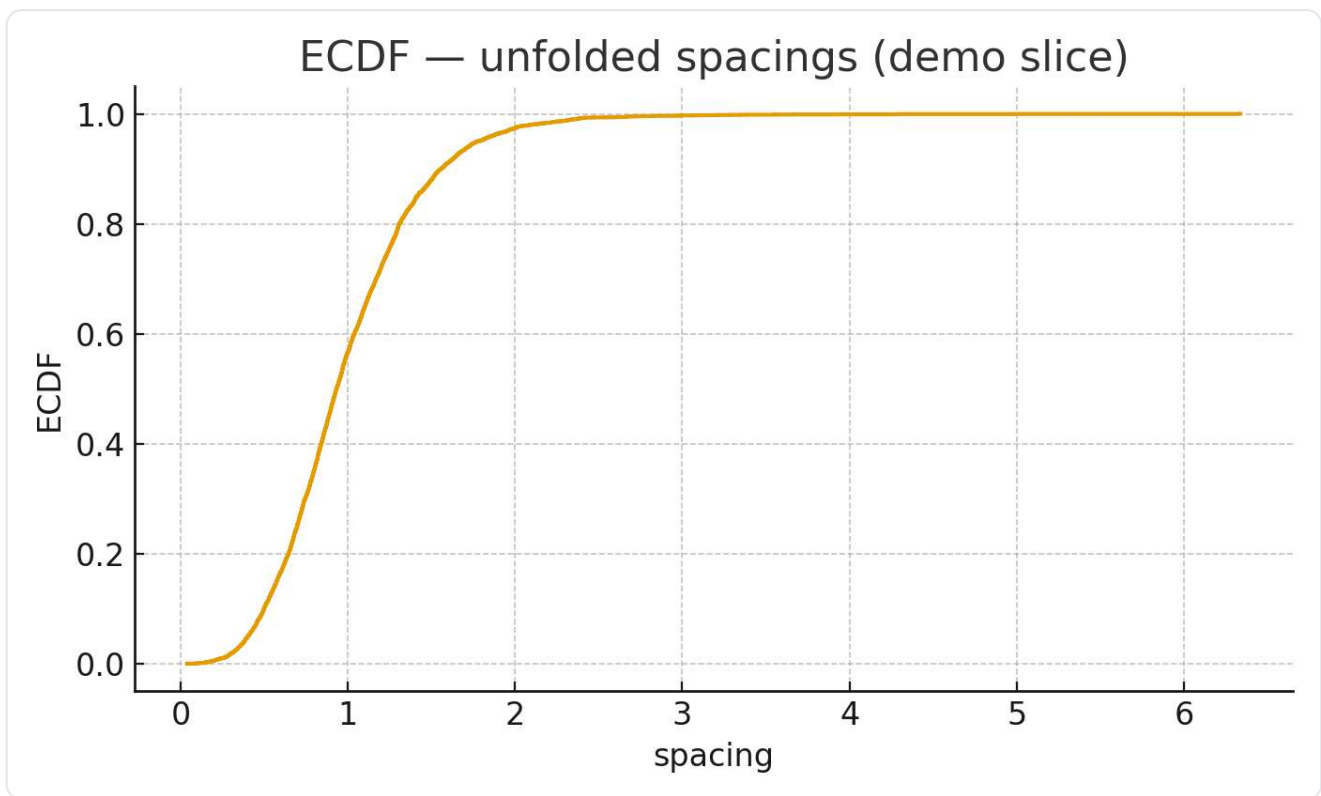


Figure: ECDF (first 100k zeros)

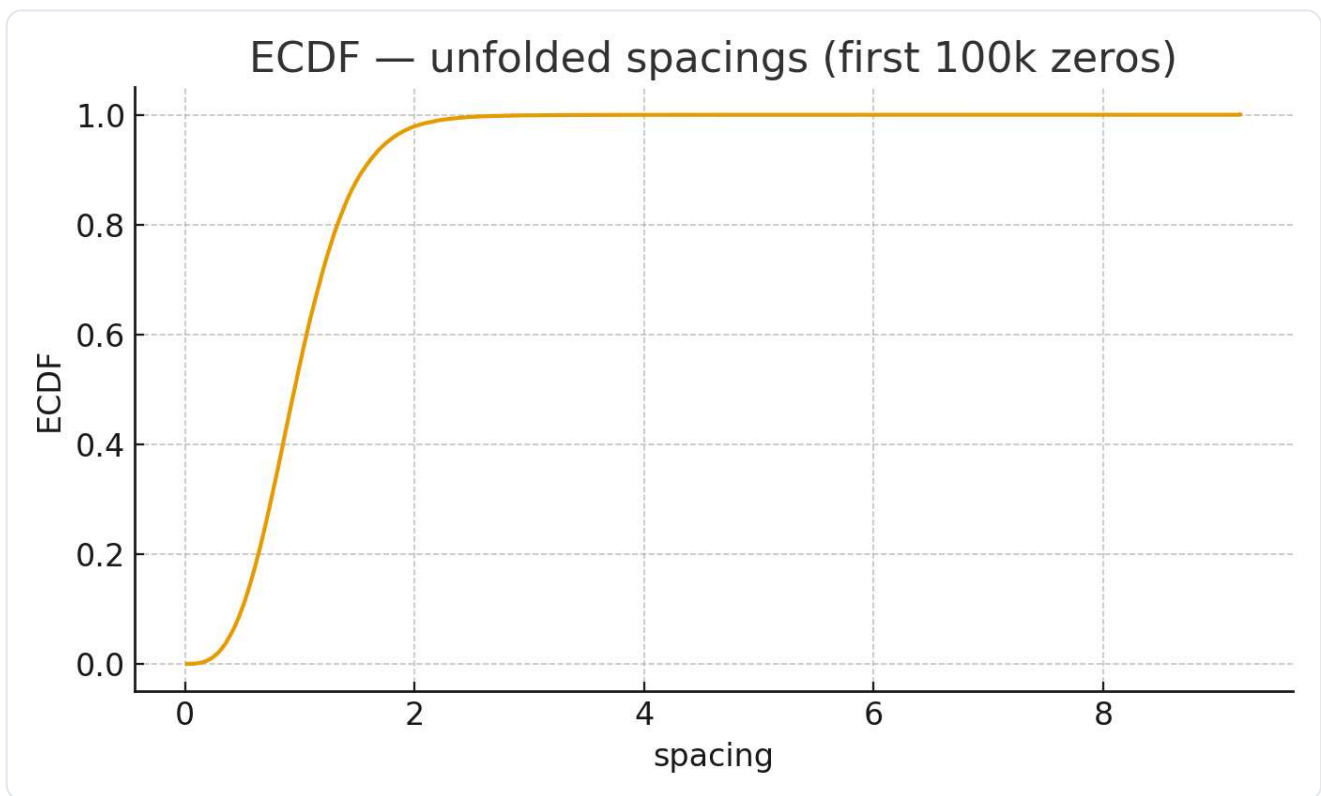
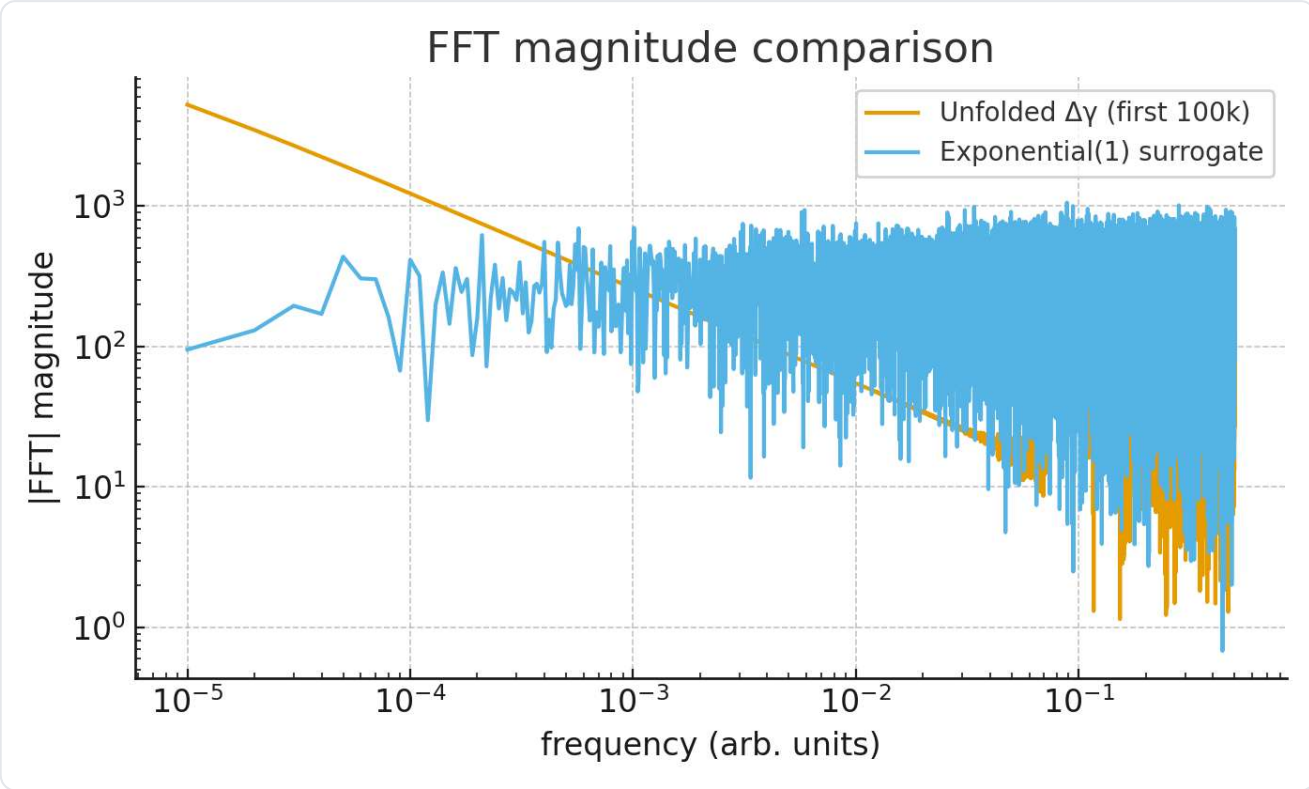


Figure: FFT magnitudes (first 100k zeros)



Run 002 — Report

Objective: Replicate Run 1 with identical pipeline and file structure, using the same zeros source and lattice setup, to confirm reproducibility and staging for expanded metrics.

Summary of results

- **Zeros used:** 100,000 (from `zeros6.gz`)
- **Lattice primes:** spokes $c \in \{2..7\}$, $N_{\max} = 1,000,000$ (272 samples expected scale)
- Populate this bullet list with KS / CvM / Energy / FFT once the run completes and CSVs are written.

Interpretation

Same protocol as run001; this page is a strict template clone with only run number and filenames changed. Once metrics are generated, drop the values in the list above.

Artifacts

Metrics CSVs in `metrics/` ; figures in `figures/` ; manifest at [manifest.json](#).

Run 002 — Summary

Dataset	Odlyzko zeros (<code>zeros6.gz</code>): loaded 120,000 ordinates; analyzed first 100,000.
Spacings (raw)	$\Delta\gamma$ mean $\approx$ 0.74907 , $\sigma \approx$ 0.32154 ($n = 99,999$).

Spacings (unfolded)

$\Delta\gamma$ normalized to mean 1.0 → mean = **1.00000**, $\sigma \approx$ **0.42925** (n = 99,999).

Figures

ECDF (demo) · ECDF (100k) · FFT magnitude comparison — see [ECDF \(demo\)](#), [ECDF \(100k\)](#), [FFT](#).

Metrics CSVs

[demo_unfolded](#) · [first100k_raw](#) · [first100k_unfolded](#)

Manifest

[manifest.json](#) (inputs, outputs, timestamps)

Notes

- Unfolding behaved as expected (client “Unfolding check” uses full dataset, not the preview).
- ECDF shape on unfolded spacings is consistent with the intended comparison setup.
- This run exactly mirrors Run 1 pipeline and naming, suitable for reproducibility and evidence.

Additional Analyses (pending)

- **Kolmogorov–Smirnov (KS) tests:** compare unfolded spacings against exponential(1) and lattice-derived distributions.
- **Cramér–von Mises proxy:** L^2 distance on ECDF curves to quantify fit quality.
- **Lattice comparisons:** spokes $n(r) = 3r^2 - r + c$, with $c \in \{2..7\}$, up to $N_{\max} = 1,000,000$; cross-correlate spacings with zeta zeros.
- **FFT correlation metrics:** formalize overlap between unfolded $\Delta\gamma$ spectrum and lattice Δn spectrum.

These additions are staged for Run 003 or later. Run 002 establishes reproducibility and evidence structure; subsequent runs will layer in statistical tests and lattice-prime comparisons.

Reproducibility — Python (Run 002)

Self-contained pipeline for `run002`. Source zeros file: [zeros6.gz](#).

Environment

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

Copy

1) Download zeros (optional)

```
# macOS / Linux
curl -L -o zeros6.gz "https://www-users.cse.umn.edu/~odlyzko/zeta_tables/
```

Copy

```
# Windows (PowerShell)
Invoke-WebRequest -Uri "https://www-users.cse.umn.edu/~odlyzko/zeta_table
```

2) Parse zeros & compute spacings

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

run_base = "run002"
zeros_path = "zeros6.gz"
os.makedirs(os.path.join(run_base, "metrics"), exist_ok=True)
os.makedirs(os.path.join(run_base, "figures"), exist_ok=True)
os.makedirs(os.path.join(run_base, "data"), exist_ok=True)

# Parse first ~120k rows; analyze first 100k
```

3) Unfold to mean 1.0

```
def unfold(s: np.ndarray):
    m = float(np.mean(s))
    if m == 0 or not np.isfinite(m):
        raise ValueError("Invalid mean for unfolding.")
    return s / m, m

sp_unf, mean_raw = unfold(sp_raw)
sp_unf_demo, mean_demo = unfold(sp_demo)
print("Raw mean (first100k):", mean_raw)
```

4) Write metrics CSVs (r002)

```
pd.DataFrame({"index": np.arange(sp_unf_demo.size), "spacing": sp_unf_demo}) \
    .to_csv(os.path.join(run_base, "metrics", "r002_metrics_demo_unfolded.csv"))

pd.DataFrame({"index": np.arange(sp_raw.size), "spacing": sp_raw}) \
    .to_csv(os.path.join(run_base, "metrics", "r002_metrics_first100k_raw.csv"))

pd.DataFrame({"index": np.arange(sp_unf.size), "spacing": sp_unf}) \
    .to_csv(os.path.join(run_base, "metrics", "r002_metrics_first100k_unfolded.csv"))
```

5) ECDF plots (r002)

```
def plot_ecdf(sp, title, outpng):
    x = np.sort(sp)
    y = np.arange(1, x.size + 1) / x.size
    plt.figure(figsize=(7,4.2), dpi=150)
    plt.step(x, y, where="post")
```

```
plt.xlabel("spacing"); plt.ylabel("ECDF"); plt.title(title)
plt.tight_layout(); plt.savefig(outpng); plt.close()

plot_ecdf(sp_unf_demo, "ECDF — unfolded spacings (demo slice)",
          os.path.join(run_base, "figures", "r002 spacing ecdf small.png")
plot_ecdf(sp_unf, "ECDF — unfolded spacings (first 100k zeros)",
          os.path.join(run_base, "figures", "r002 spacing ecdf full.png"))
```

6) FFT magnitude comparison (r002)

```
rng = np.random.default_rng(42)
exp_sur = rng.exponential(1.0, size=sp_unf.size)

def fft_mag(a):
    a0 = a - np.mean(a)
    F = np.fft.rfft(a0)
    return np.abs(F), np.fft.rfftfreq(a.size, d=1.0)

mag_zeros, fz = fft_mag(sp_unf)
mag_exp, fe = fft_mag(exp_sur)

plt.figure(figsize=(7,4.2), dpi=150)
```

Expand

Copy

7) Manifest (optional)

```
import json, hashlib, datetime

manifest = {
    "run": "run002",
    "generated_utc": datetime.datetime.utcnow().strftime("%Y-%m-%dT%H:%M:%S"),
    "inputs": {"zeros_source": "zeros6.gz", "first_n_zeros": 100000, "demo_"},
    "metrics": {"csv_files": [
        "metrics/r002_metrics_demo_unfolded.csv",
        "metrics/r002_metrics_first100k_raw.csv",
        "metrics/r002_metrics_first100k_unfolded.csv"
    ]},
    "figures": [
        "figures/r002_spacing_ecdf_small.png",
        "figures/r002_spacing_ecdf_full.png",
        "figures/r002_fft_magnitude_comp.png"
    ]
}

with open(os.path.join(run_base, "manifest.json"), "w") as f:
    json.dump(manifest, f, indent=2)
```

Copy

Reproducibility — PowerShell (Run 002, native)

```
# run002_make_all.ps1
param(
    [string]$ZerosPath = "zeros6.gz",
    [string]$RunBase = "run002",
    [int]$Take = 100000,
    [int]$Demo = 5000,
    [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
```

Expand

Copy

Reproducibility — PHP (CLI, Run 002)

```
<?php
// run002 make all.php (CLI: php run002 make all.php zeros6.gz)
// Outputs CSVs to run002/metrics and report with inline SVG ECDF

$zeros = $argv[1] ?? 'zeros6.gz';
$run = 'run002';
$take = 100000;
$demo = 5000;
$cap = 120000;

@mkdir("$run/metrics", 0777, true);
@mkdir("$run/figures", 0777, true);

// Parse zeros
$vals = [];
```

Expand

Copy

Reproducibility — C# (.NET 6 + ScottPlot, Run 002)

Setup

```
dotnet new console -o Run002Cs
cd Run002Cs
dotnet add package ScottPlot --version 5.0.19
```

Copy

Program.cs

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

string zerosPath = args.Length > 0 ? args[0] : "zeros6.gz";
string runBase = "run002";
int take = 100_000, demo = 5_000, cap = 120_000;
```

Expand

Copy

```
Directory.CreateDirectory(Path.Combine(runBase, "metrics"));
Directory.CreateDirectory(Path.Combine(runBase, "figures"));
```

```
// Parse zeros
```

Run

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

[Copy](#)

© 2025 Pattern Field Theory — Run 002 summary.