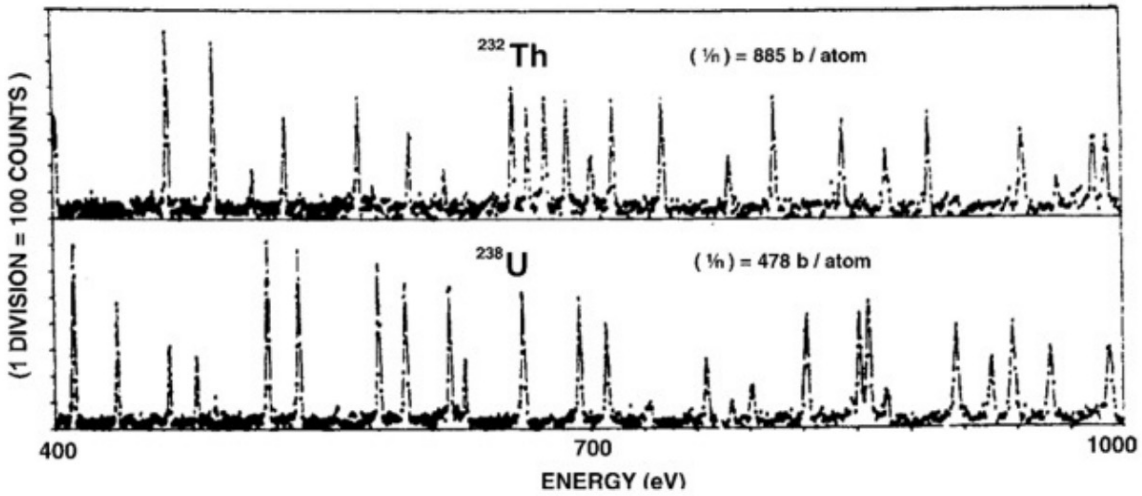
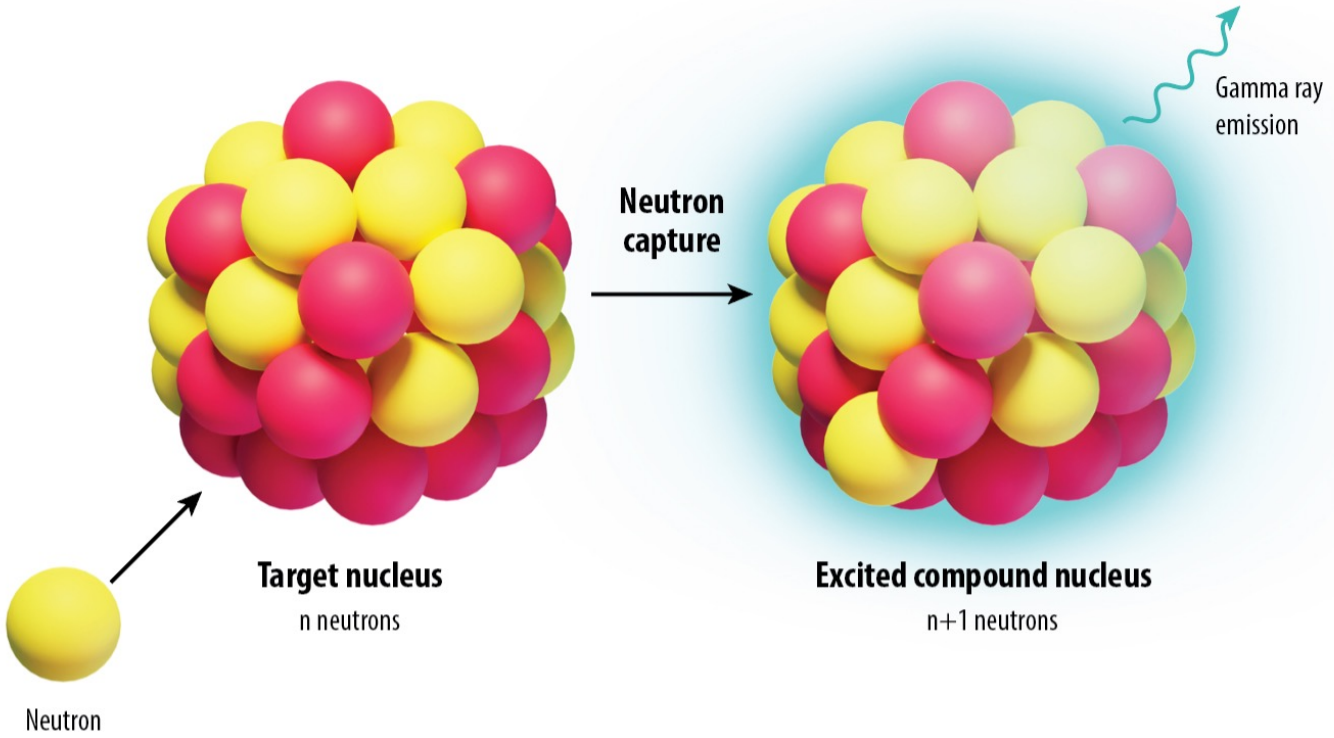

RANDOM MATRIX THEORY

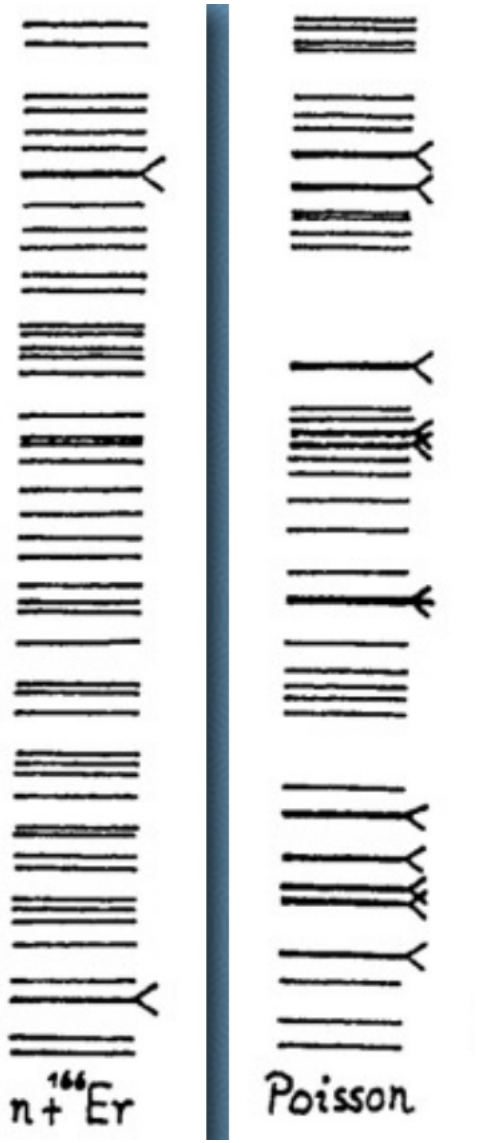
complex quantum system

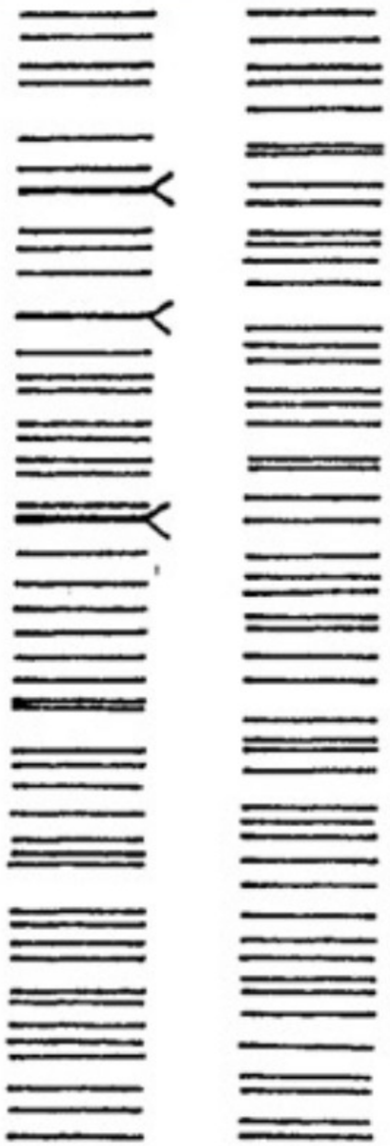
intractable Hamiltonian

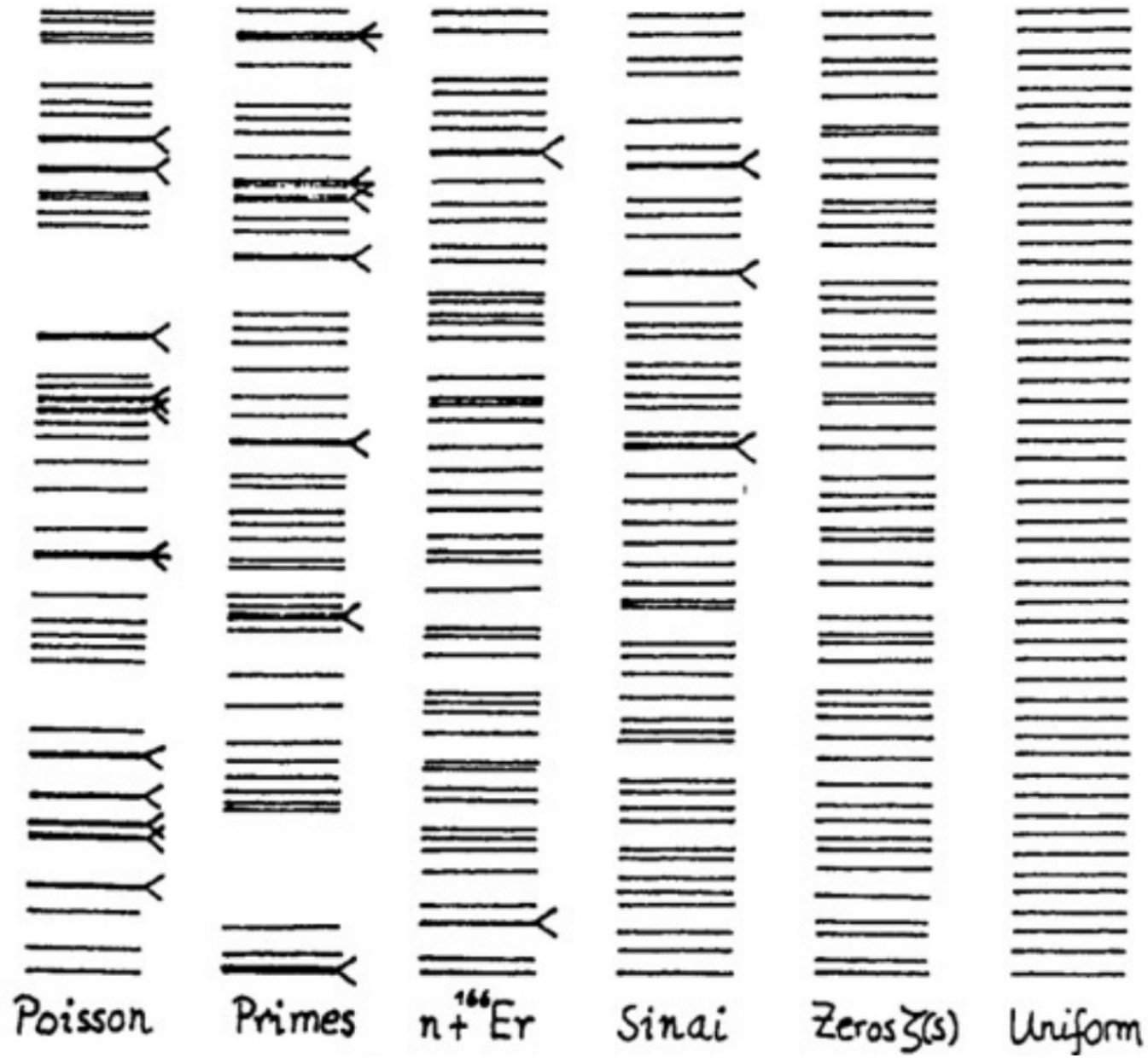
how to characterize the eigenenergies?



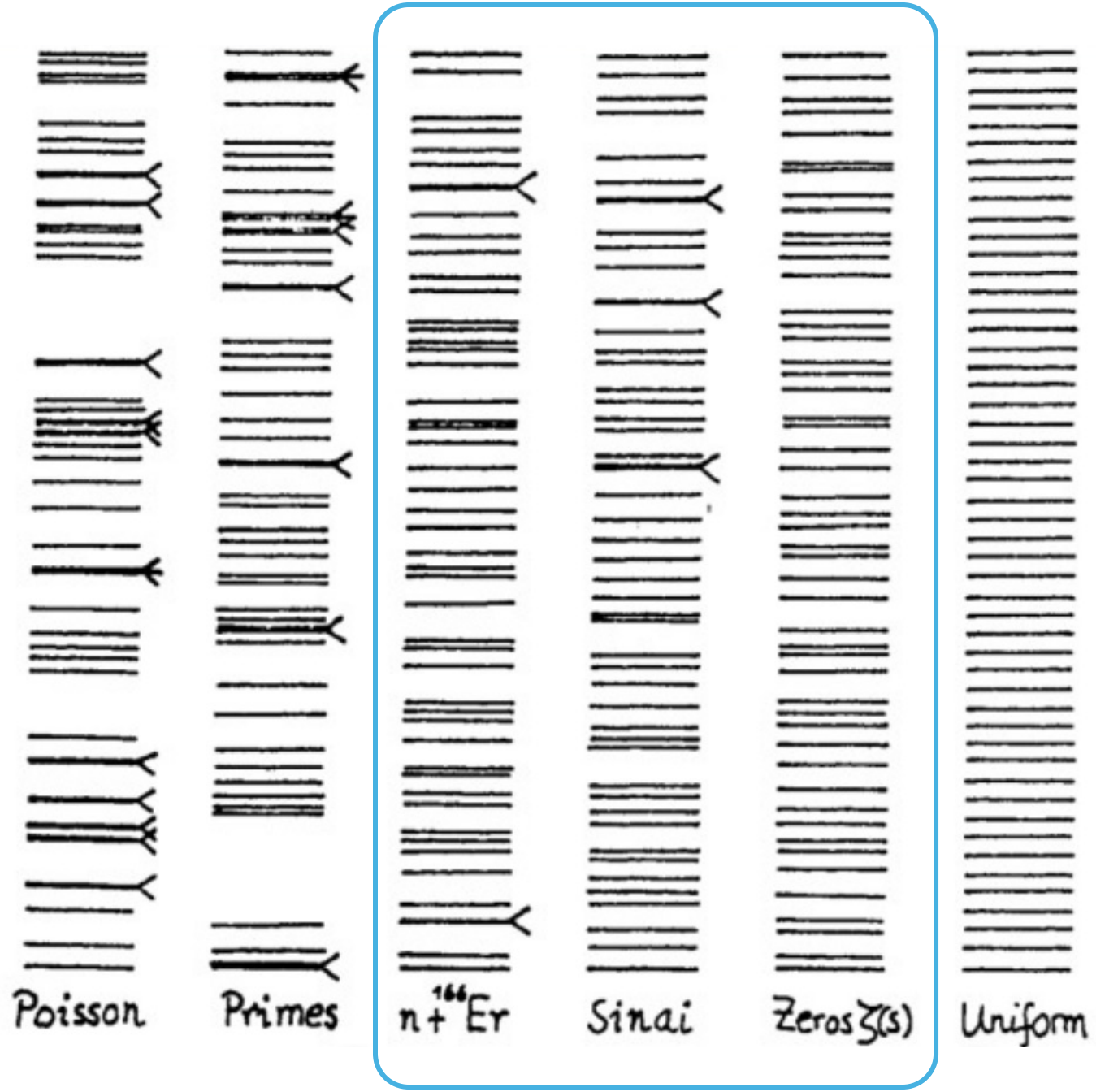







Poisson
Primes
n+√n Er
Sinai
Zeros ζ(s)
Uniform

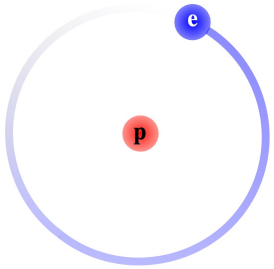

They're the same picture.


Poisson
Primes
n+"Er
Sinai
Zeros ζ(s)
Uniform

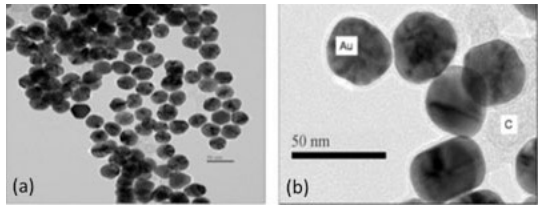
notion of **"level repulsion"**

found in complex physical systems, including

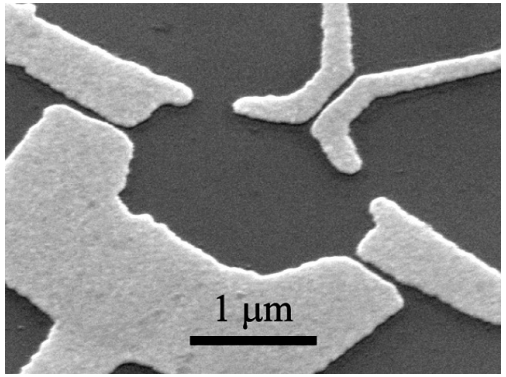
H atom in a B field

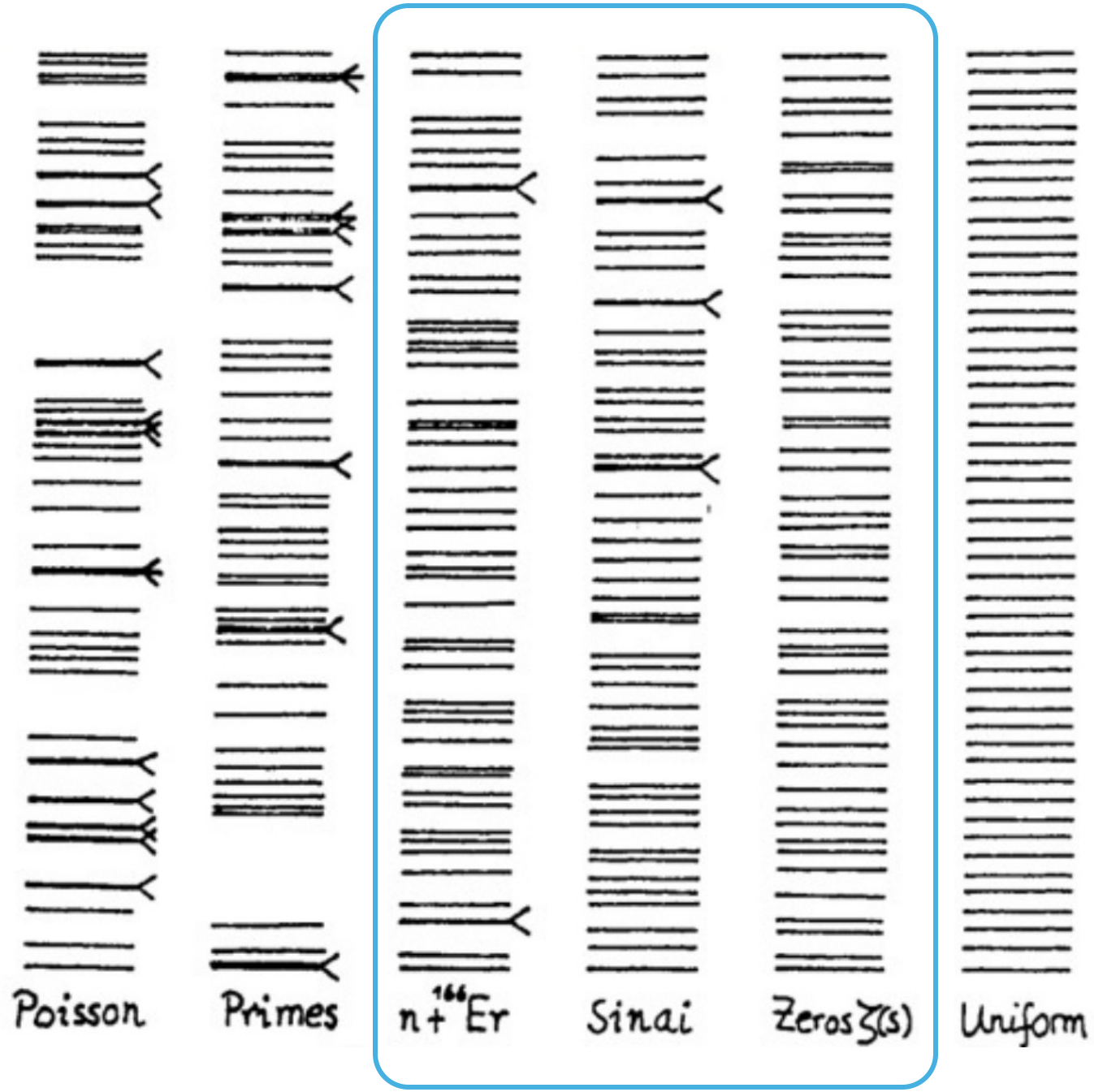


metal grain



quantum dots

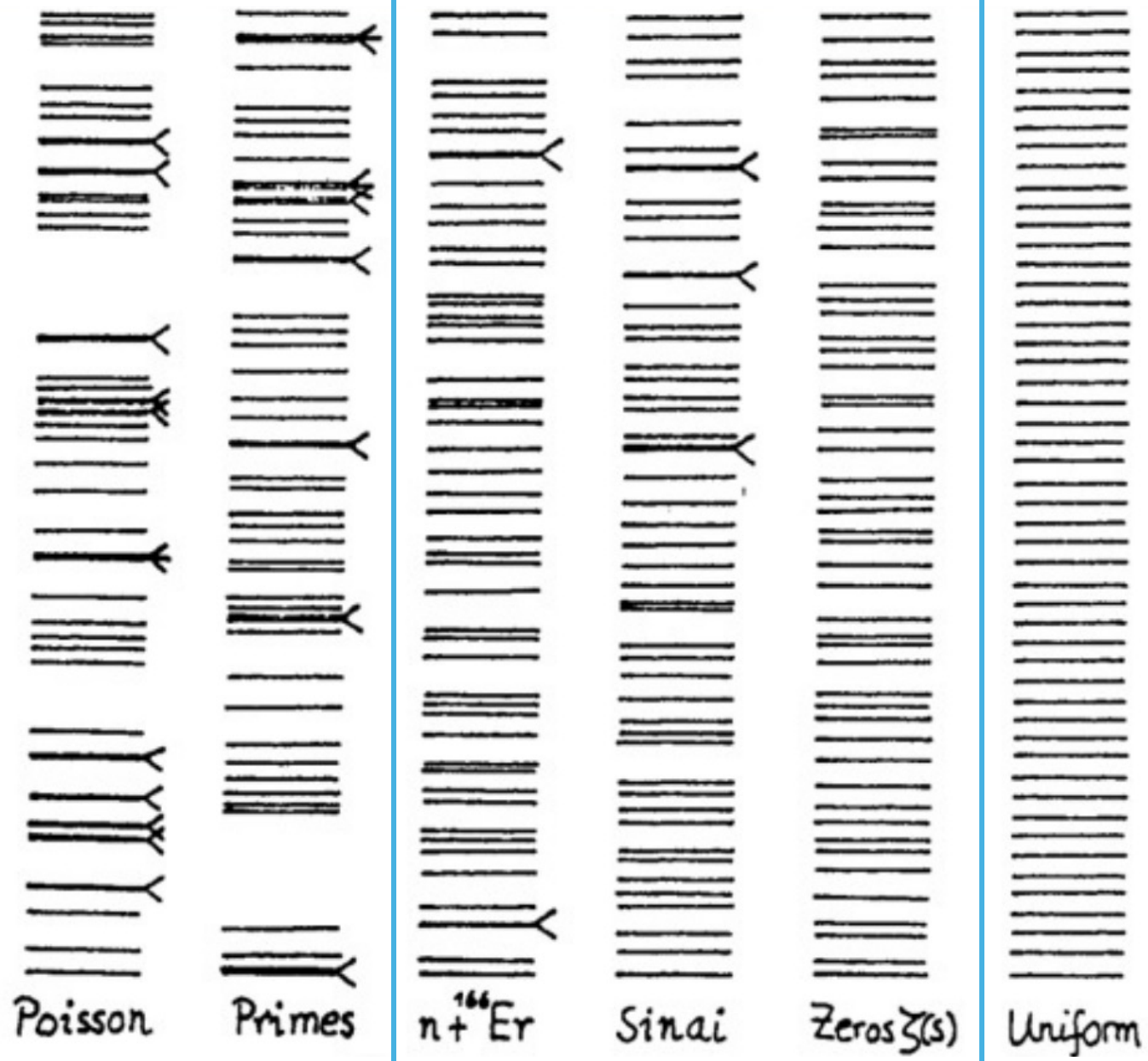


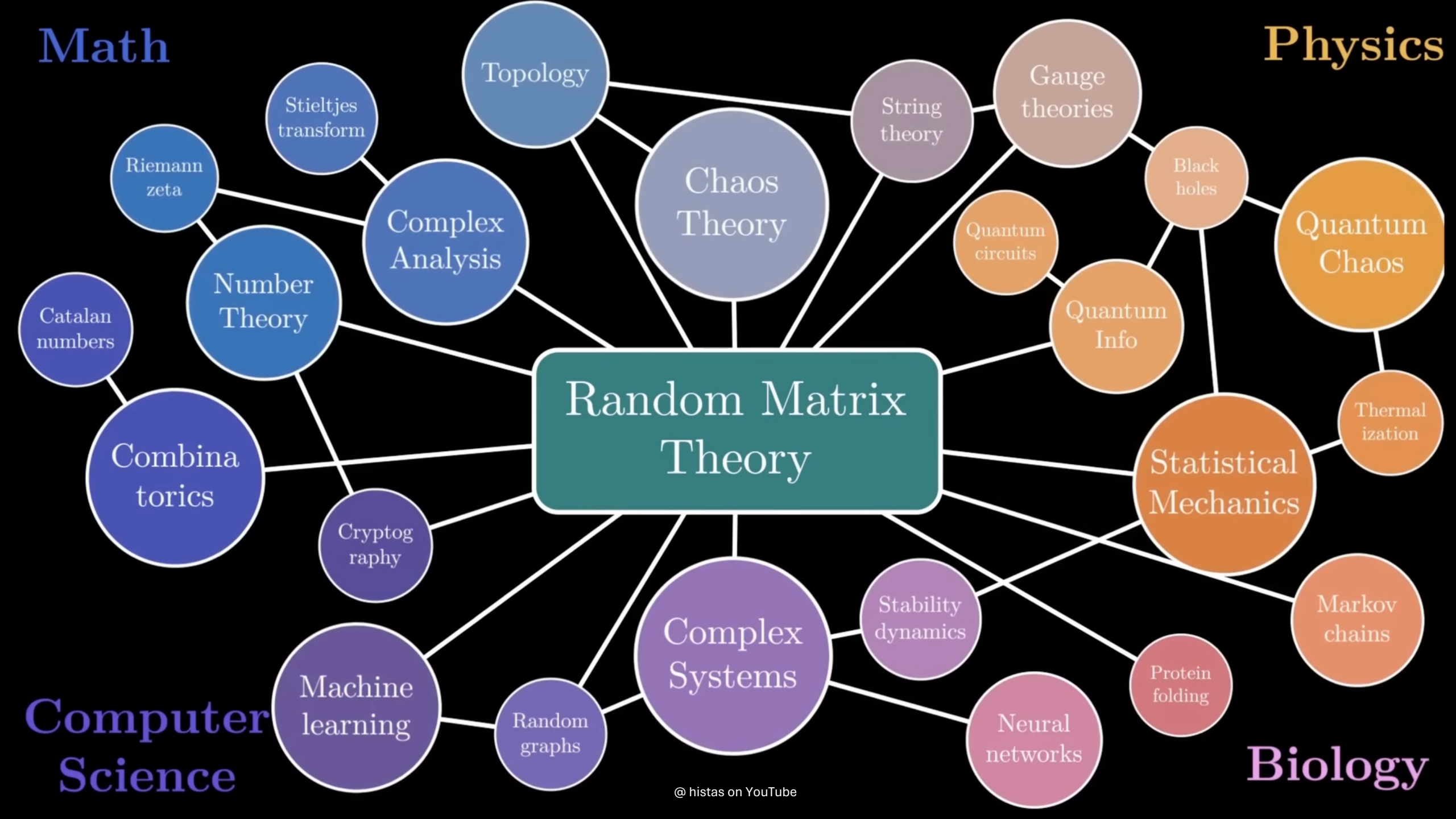


@ statistical theory of energy levels by Wigner, Dyson, Mehta (60s-70s)


deterministic
dynamics


random
matrix
theory


Poisson
Primes
n+"Er
Sinai
Zeros ζ(s)
Uniform


Math
Physics
Topology
Stieltjes transform
String theory
Gauge theories
Riemann zeta
Chaos Theory
Black holes
Complex Analysis
Quantum circuits
Quantum Chaos
Number Theory
Quantum Info
Catalan numbers
Random Matrix Theory
Thermal ization
Combina torics
Statistical Mechanics
Cryptog raphy
Markov chains
Stability dynamics
Complex Systems
Machine learning
Protein folding
Random graphs
Neural networks
Computer Science
Biology
@ histas on YouTube

Wigner surmise (1956): "Perhaps I am now too courageous when I try to guess the distribution of the distances between successive energy levels…"

$$P(s) \propto s^{\beta} e^{-s^2}$$

s – spacing between subsequent energy levels, normalized by the average spacing Δ: $s = (E_{i+1} - E_i)/\Delta$

$$H = \begin{pmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{pmatrix} \xrightarrow[\text{equation}]{\text{characteristic}} s \equiv \lambda_+ - \lambda_- = \sqrt{(H_{11} - H_{22})^2 + 4H_{12}^2}$$

$$P(s) \sim \int dH_{11}\, P(H_{11}) \int dH_{12}\, P(H_{12}) \int dH_{22}\, P(H_{22})\, \delta\left(s - \sqrt{(H_{11} - H_{22})^2 + 4H_{12}^2}\right)$$

$$P(H_{11}) \sim e^{-H_{11}^2/2} \qquad P(H_{22}) \sim e^{-H_{22}^2/2} \qquad P(H_{12}) = e^{-\frac{2H_{12}^2}{2}} \quad \text{↑ symmetry}$$

$$\hookrightarrow \begin{cases} u = (H_{11} - H_{22})/2 \\ w = 2H_{12} \\ v = (H_{11} + H_{22})/2 \end{cases} \longrightarrow P(s) \sim \int du\, dv\, dw\, \delta\left(s - \sqrt{u^2 + w^2}\right) e^{-(u^2 + v^2 + w^2)}$$

$$\ldots \sim s\, e^{-s^2}$$

How do we generate a random matrix H? What is the probability distribution $\mathrm{P}(H)$?

Any physically relevant result should be independent of the basis chosen:
use basis change invariants, the eigenvalues!

$P(H) = f(\{E_i\})$ $\{E_i\}$ – set of eigenvalues

Useful minimal model:

$f(\{E_i\}) = \prod_i f(E_i)$ with Gaussian $f(E_i) \propto e^{-cE_i^2}$

which gives $P(H) = f(\{E_i\}) \propto \prod_i e^{-cE_i^2} = e^{-c\sum_i E_i^2} = e^{-c\,\mathrm{Tr}\,H^2}$,

or in terms of matrix elements H_{ij}: $P(H) \propto e^{-c\,\mathrm{Tr}\,H^2} = e^{-c\,\sum_{ij}|H_{ij}|^2}$

→ independent Gaussian distributions for all H_{ij}

$$P(H) \propto \prod_{i<j} \exp\left(-2c|H_{ij}|^2\right) \prod_i \exp(-c|H_{ii}|^2)$$

Gaussian ensemble symmetries

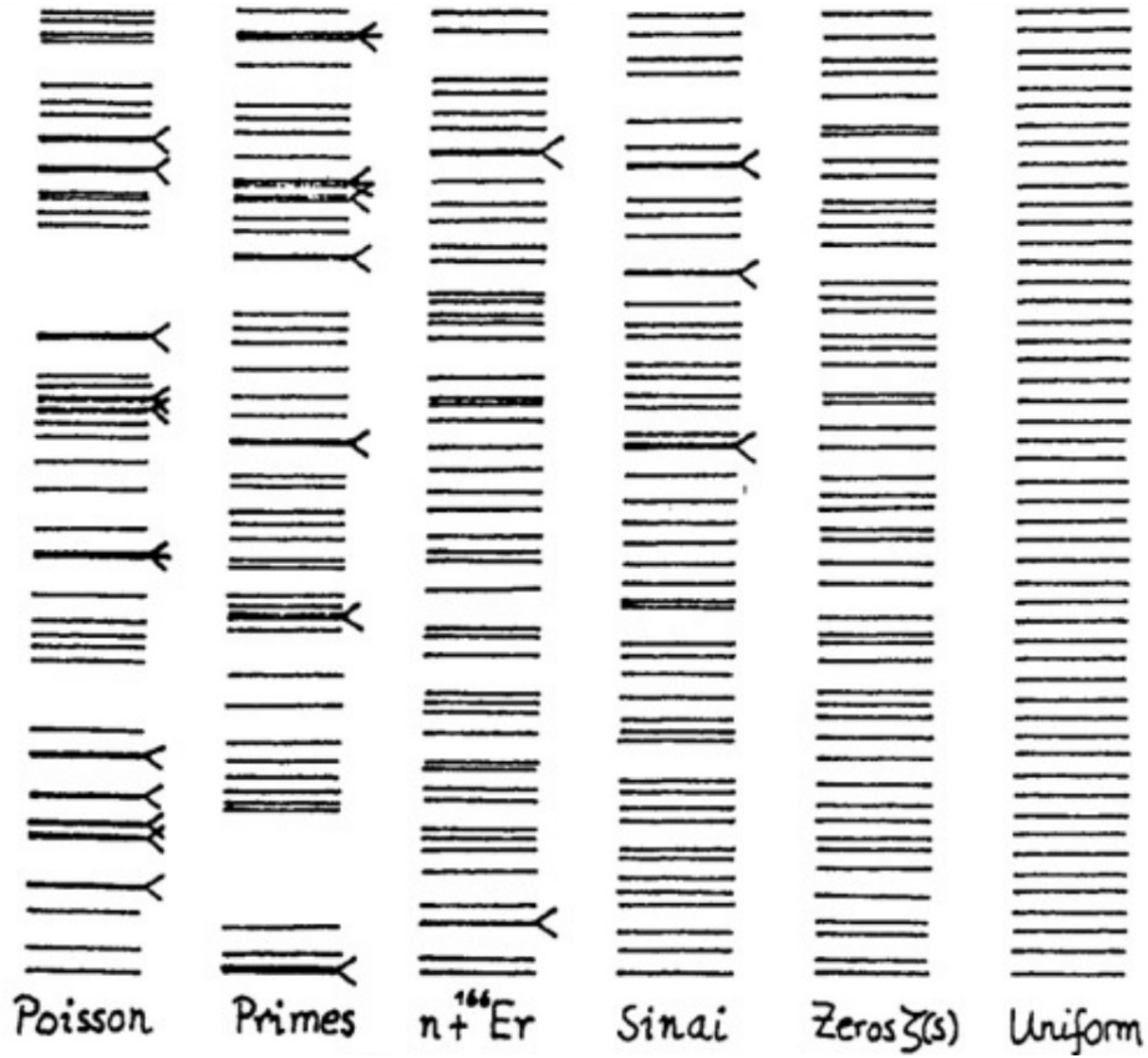
Naming conventions follow from the properties of matrices that diagonalize H

- Gaussian Orthogonal Ensemble (GOE)
  - $\beta = 1$
  - time reversal symmetry
  - H is real, $H = H^*$
  - $P(s) = \frac{\pi}{2} s e^{-\frac{\pi}{4}s^2}$
  - Numerical recipe:
    draw each H_{ij} as an independent Gaussian variable with mean $\mu = 0$ and standard deviation $\sigma = 1$

- Gaussian Unitary Ensemble (GUE)
  - $\beta = 2$
  - no time reversal symmetry
  - H is complex
  - $P(s) = \frac{32}{\pi^2} s^2 e^{-\frac{4}{\pi}s^2}$
  - Numerical recipe: draw x and y from $N(\mu = 0, \sigma = 1)$ and set $H_{ij} = (x + iy)/\sqrt{2}$

In both cases, after drawing H_{ij} return the Hermitian $\mathrm{H} \coloneqq (H + H^\dagger)/2$.

Even though derived for 2x2 matrices, formulas hold within 2% even for very large ones!


Poisson
Primes
n+√n Er
Sinai
Zeros ζ(s)
Uniform

Density of states

We first need to check the level density for some test models. We use only Gaussian ensembles for this lab. Matrix entries of a test matrix are random and uncorrelated. We want to discover a semi-circle law, also attributed to Wigner, which describes the density of states of a large, random matrix. This semicircle law acts as a central limit theorem for large (symmetric) random matrices.

With a Gaussian ensemble one can derive many results analytically; the semicircle law is a limit for N -> ∞. It is known that there are deviations for smaller values of N, which we should also see in our lab.

Task 1

Use subplots, colors and labels to illustrate the results. Calculate eigenvalues of nsample random matrices drawn from GOE ensamble (N x N matrix size). Make a histogram (normalized) of eigenvalues and compare with the analytical Wigner semicircle law. For this task we accumulate all eigenvalues from all generated matrices.

Wigner's semicircle law: $\rho(E) = \frac{2}{\pi R^2}\sqrt{R^2 - E^2}$, where in GOE, $R = \sqrt{2N}$.

Example test values:
(A) N = 6, nsample = 20000;
(B) N = 20, nsample = 10000;
(C) N = 200, nsample = 500

Hints

We can plot the histogram by using the hist function, e.g. with the following parameters

```
n, bins, _ = plt.hist(eigen, 50, density=True,
facecolor='cyan', alpha=0.75)
```

We can also plot an exact function wigner on top of that

```
plt.plot(bins, wigner(bins), 'r-', linewidth=2)
```

Simplest way to get a random Hamiltonian from GOE:

```
h = np.random.randn(N,N)
return (h+h.T)/2
```

Task 2

Calculate the histogram of energy spacings for GOE and GUE ensembles. Normalize the accumulated energy spacings to unit mean level spacing (divide by the average). Compare with the Wigner surmise prediction:

$$P(s) = \begin{cases} \frac{\pi}{2} s e^{-\frac{\pi}{4}s^2} & \text{for } GOE \\ \frac{32}{\pi^2} s^2 e^{-\frac{4}{\pi}s^2} & \text{for } GUE \end{cases}$$

Spectrum unfolding:

We have to get levels with approximately uniform density, so we will only probe the middle part of the whole spectrum. For matrix sizes $N \geq 10$ use 1/4 of the spectrum around the eigenvalue $N/2$, for smaller matrices use only the two middle eigen-energies.

Present the results:

Illustrate the results for the marix sizes $N = 8$ and $N = 200$. Try also to use bigger matrix sizes: how far can we reasonably increse N?

Hints

Sorting the eigenvalues

```
eigen = np.sort(eigen)
```

and calculating the differences (spacings):

```
diff = np.diff(eigen)
```