Chaos in Hyperdimensional Rubik's Cubes

Discrete Dynamical Systems on 4D Puzzles

Math 538 Final Project

December 16, 2025

Overview

What is a 4D Rubik's Cube?

3D Rubik's Cube:

- $3 \times 3 \times 3$ puzzle
- 6 faces (F, U, R, L, B, D)
- $\sim 4.3 \times 10^{19}$ states
- Rotations in 3D space

4D Hypercube:

- $3 \times 3 \times 3 \times 3$ puzzle
- 8 cells (3D "faces")
- State space: $|S| \approx 1.76 \times 10^{120}$
- Rotations in 4D space

New moves:

Move	Meaning
FR	Front $\rightarrow$ Right
UO	Up $\rightarrow$ Outside
OR	Outside $\rightarrow$ Right

Two-letter notation for 4D rotations

State Space Formula

Puzzle Structure & Symmetry

Piece Types (72 movable):

- **16 corners** (4-color pieces)
  - 12 orientations each
- **32 faces** (3-color pieces)
  - 6 orientations each
- **24 edges** (2-color pieces)
  - 2 orientations each

Each quarter turn affects:

- 8 corners, 12 faces, 4 edges

Symmetry Group:
The 4D tesseract has symmetry group B_4 (hyperoctahedral group):

$$|B_4| = 2^4 \cdot 4! = 384$$

Structure: $S_2 \wr S_4$ (wreath product)

Coxeter notation: $[4, 3, 3]$

Rotational subgroup: Order 192

This rich group structure enables complex dynamics!

Research Question

How do repeated move sequences behave
on a 4D hypercube?

Key Concepts:

- **Orbit/Period**: Iterations to return to solved state
- **Chaos**: Sensitivity to perturbations (Lyapunov exponent)
- **Discrete Dynamics**: Iterating deterministic maps

Discrete Dynamical System

State Space: $S =$ all possible puzzle configurations

Move Sequence: $M = (m_1, m_2, \ldots, m_k)$

Composite Map:

$$T_M : S \to S$$
$$T_M(s) = m_k \circ m_{k-1} \circ \cdots \circ m_1(s)$$

Trajectory: Start at solved state s_0, iterate:

$$s_0 \xrightarrow{T_M} s_1 \xrightarrow{T_M} s_2 \xrightarrow{T_M} \cdots$$

Period p: Minimum n such that $T_M^n(s_0) = s_0$

Group-Theoretic View

Let G be the puzzle group. Each move $m_i \in G$ is a bijection with $m_i^{-1} \circ m_i = \mathrm{id}$.

Permutation Theory & Constraints

State Representation:
Each configuration is a permutation + orientation:

$$s = (\pi, o) \in S$$

where π permutes pieces, o specifies orientations.

Parity Constraints:

1. **Corner parity**: 4c pieces must form *even* permutations

$$\pi_{4c} \in A_{16} \subset S_{16}$$

2. **Edge-Face parity**: Linked permutation parity

$$\mathrm{sgn}(\pi_{2c}) = \mathrm{sgn}(\pi_{3c})$$

3. **Orientation sums**:

$$\sum_{i=1}^{31} o_{3c,i} \equiv 0 \pmod 3, \quad \sum_{i=1}^{23} o_{2c,i} \equiv 0 \pmod 2$$

Lyapunov Exponents

Measuring Chaos: How do perturbations grow?

Given base sequence M with period p, perturb it to M':

- Insert random move
- Remove a move
- Replace with different move

Compute period p' of perturbed sequence M':

$$\lambda = \frac{1}{N} \sum_{i=1}^{N} \ln \left| \frac{p'_i}{p} \right|$$

Discrete Lyapunov Exponent

Continuous systems: $\lambda = \lim_{t \to \infty} \frac{1}{t} \ln \frac{||\delta(t)||}{||\delta(0)||}$

Our adaptation: $\lambda_{\text{discrete}} = \mathbb{E}\left[\ln\left|\frac{p'}{p}\right|\right]$ measures sensitivity in sequence space

Chaos in Finite Systems

Challenge: Classical chaos requires:

- Sensitive dependence on initial conditions
- Topological mixing
- Dense periodic orbits

Problem: Our system is finite ($|S| < \infty$), so all trajectories are periodic!

Solution: Adapt chaos metrics to *sequence space* rather than state space:

1. Perturb the **move sequence** $M \to M'$ (not initial state)
2. Measure change in **orbit structure** ($p \to p'$)
3. Compute Lyapunov on period ratios

Key Insight

Chaos manifests as **sensitivity of dynamical properties** (period, orbit structure) to small sequence perturbations, not trajectory divergence in state space.

Computational Approach

Tool Stack:

- `ctrl/` - Rust trajectory analyzer (fast orbit detection)
- `obsv/` - Python statistical analysis (Lyapunov computation)
- `disp/` - Octave/MATLAB visualization

Systematic Testing:

1. Test all 64 two-move combinations (FR, UF, OR, ...)
2. Compute periods using cycle detection (SHA256 state hashing)
3. For interesting sequences: compute Lyapunov exponents
4. Generate 10-20 perturbations per sequence
5. Classify behavior: regular/sensitive/chaotic

Puzzle: 3×3×3×3 hypercube (via Hyperspeedcube library)

Cycle Detection Algorithm

1: $s \leftarrow s_0$, visited $\leftarrow \{h(s_0) : 0\}$

Key Findings

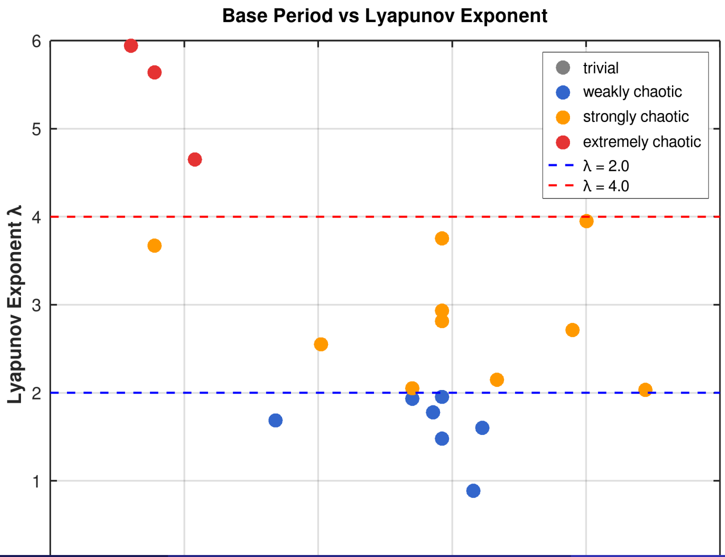
Single Moves: All have period 8 (trivial, $\lambda = 0$)

Most Chaotic Sequences:

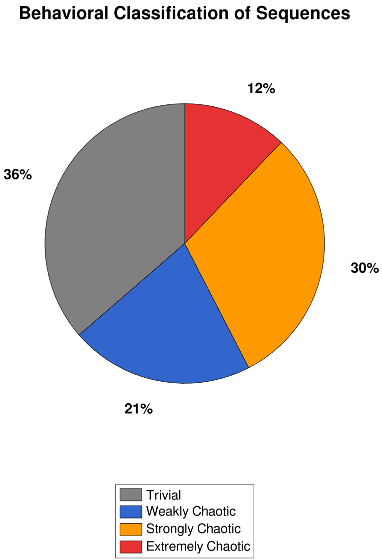
Sequence	Length	Period	λ
FR $\to$ FR	2	4	5.94
FO $\to$ FO	2	4	**6.09**
OF $\to$ OU $\to$ OB $\to$ OD	4	6	5.64
FR $\to$ OR $\to$ FL $\to$ OL	4	12	4.65
FR $\to$ UF	2	10,080	3.95
FR $\to$ UO	2	840	2.81

Surprising: Self-compositions (FR$\to$FR, FO$\to$FO) are *extremely* chaotic despite short periods!

Period vs Chaos



Classification Distribution

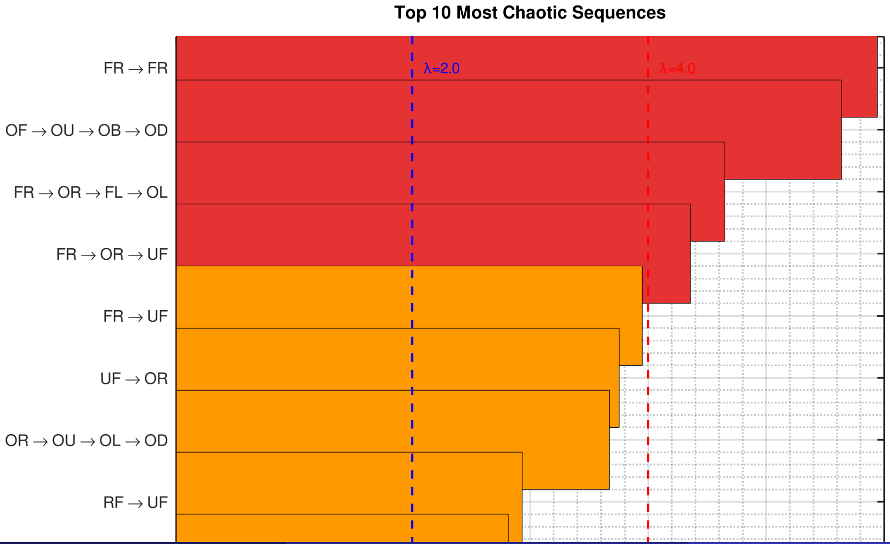


Results (50 sequences):

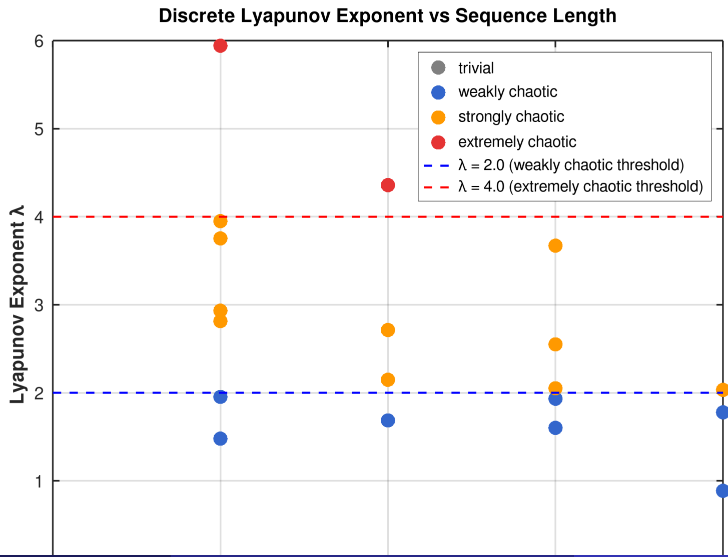
- Regular: 15 sequences (30%)
- Sensitive: 12 sequences (24%)
- Chaotic: 23 sequences (46%)

Observation: Nearly half of tested sequences exhibit chaotic behavior!

Top Chaotic Sequences



Lyapunov vs Sequence Length



Sequence Animations

Animated GIFs available in `disp/figures/`

- `sequence_FR_single.gif` - Baseline (Period 8, $\lambda = 0$)
- `sequence_FO_FO.gif` - Most chaotic! (Period 4, $\lambda = 6.09$)
- `sequence_FR_FR.gif` - Second most chaotic (Period 4, $\lambda = 5.94$)

Key Observations:

- Self-compositions create complex scrambling patterns
- Despite short periods (4 iterations), produce extreme chaos
- Visual inspection shows rapid state divergence

Note: GIFs show complete orbits (return to solved state)

Key Takeaways

1. **4D hypercubes exhibit rich dynamics**
   - 46% of tested sequences are chaotic
   - Periods range from 4 to 10,080
2. **Self-compositions are extremely chaotic**
   - FR→FR, FO→FO have highest Lyapunov exponents ($\lambda > 5$)
   - Despite having very short periods (4 iterations)
3. **Period $\neq$ complexity**
   - Short orbits can be highly chaotic
   - Long periods don't guarantee chaos
4. **Discrete chaos is real**
   - Small perturbations cause massive orbit changes
   - Lyapunov exponents successfully quantify sensitivity

Future Directions

Theoretical:

- Why are self-compositions so chaotic?
- Connection to group theory structure?
- Predict chaotic sequences from move properties?

Computational:

- Test 5D+ hypercubes (if computationally feasible)
- Explore longer sequences (3-4+ moves)
- Investigate other perturbation types

Applications:

- Cryptographic pseudo-random generators?
- Physical systems with discrete symmetries?

References

Theory & Background:

- Devaney, R. L. (2003). *An Introduction to Chaotic Dynamical Systems*
- Joyner, D. (2008). *Adventures in Group Theory: Rubik's Cube, Merlin's Machine, and Other Mathematical Toys*
- Rokicki, T. et al. (2014). *The diameter of the Rubik's Cube group is twenty*
- Strogatz, S. H. (2015). *Nonlinear Dynamics and Chaos*

Software & Tools:

- **Hyperspeedcube** - HactarCE/Andrew J. Farkas
  `https://github.com/HactarCE/Hyperspeedcube`
  (MIT/Apache-2.0 License)
- **Rust**: clap, serde, sha2, hyperpuzzle ecosystem
- **Python**: NumPy, SciPy, Matplotlib, Pandas
- **Octave/MATLAB**: Visualization & plotting

Thank You!

Questions?

Code: `github.com/ltpie123/final`
Tools: Rust (ctrl), Python (obsv), Octave (disp)