Information and complexity

Kolmogorov complexity and algorithmic information

Dimitri Petritis

Institut de recherche mathématique de Rennes
Université de Rennes 1 et CNRS (UMR 6625)

September 2022

Shannon's information
First critique

"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is they refer to or are correlated according to some systems with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one selected from a set of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design."

Claude Shannon (1948),
A mathematical theory of communication.

Shannon's information
First critique (cont'd)

- X random variable distributed according to $\mathbf{p}$ on finite set $\mathbb{X}$.
- One of interpretations of $H(\mathbf{p}) = H(X) =$ mean number of binary questions needed to determine X.
- Another (equivalent) interpretation: $\mathbb{E}(-\log p(X))$.
- Can define **descriptive complexity** of event $\{X = x\}$ the number $\lceil \log \frac{1}{p(x)} \rceil$, since $\mathbb{E}(\lceil \log \frac{1}{p(x)} \rceil) \simeq H(\mathbf{p}) =$ the number of questions needed to determine whether $X = x$.
- But numerous situations where $\mathbf{p}$ unknown or worse not existing. E.g.
  - What is the information content of these transparencies? Can they be viewed as element of a set of all possible transparencies with a probability vector on it?
  - What is the heredity information of a biological organism encoded in its DNA? Again, can it be viewed as a DNA realisation in the set of all possible ones with a probability vector on it?

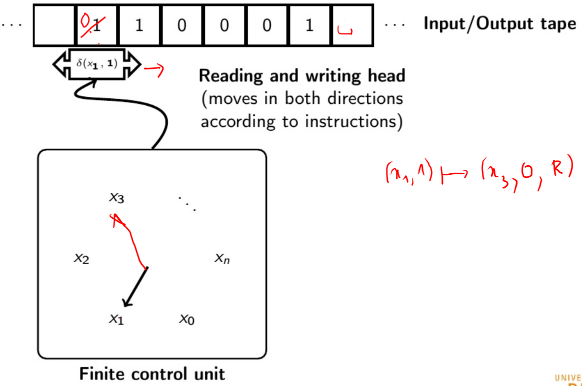
Shannon's information
Second critique

- H assigns information to an ensemble of possible messages on finite set $\mathbb{X}$
- If all messages equiprobable in $\mathbb{X}$, then $H = \log|\mathbb{X}|$ =number of bits to describe generic message.
- Says nothing about number of bits to convey individual message.

Example

- $\mathbb{A} = \{0, 1\}$.
- $\mathbb{X}$ = set of binary strings of 2×10^9 bits, i.e. $\mathbb{X} = \mathbb{A}^{2\times 10^9}$.
- If $\mathbf{p}$ uniform probability vector, Shannon $H(\mathbf{p}) = 2 \times 10^9$ bits.
- Hence, generically, words of $\mathbb{X}$ require 2×10^9 bits to be described.
- But, appealing to meaning of message, some words admit substantially shorter description, e.g.
  - among the words of $\mathbb{X}$, consider $\alpha := (01)^{10^9} = \underbrace{01\cdots01}_{2\times 10^9}$.
  - α admits description "the repetition one billion times of the word 01" requiring only 47 letters (and digits) of the Latin alphabet. Using ISO-8859-1 coding, only $47 \times 8 = 376$ bits necessary.

Intermezzo
What is a Turing machine?



Intermezzo
What is a Turing machine (cont'd)?

Definition

A **Turing machine** M is the sextuple $(\mathbb{X}, \mathbb{A}, \mathbb{B}, \delta, x_0, \mathbb{F})$ where

- $\mathbb{X}$ is the finite set of internal states,
- $\mathbb{A}$ is the finite alphabet in which words of the language are written,
- $\mathbb{B} \supset \mathbb{A}$ is the finite (extended) alphabet of the tape,
- $\delta : \mathbb{X} \times \mathbb{B} \to \mathbb{X} \times \mathbb{B} \times \{L, R\}$ is the transition function ($L := -1$ means "move the head leftwards", $R := +1$ "move rightwards"),
- x_0 is the initial state,
- $\mathbb{F} = \mathbb{F}_{\mathrm{acc}} \sqcup \mathbb{F}_{\mathrm{rej}}$ is the set of halting states, split into $\mathbb{F}_{\mathrm{acc}} = \{x_{\mathrm{acc}}\}$ and $\mathbb{F}_{\mathrm{rej}} = \{x_{\mathrm{rej}}\}$ (always assume that $x_0 \notin \mathbb{F}$).

Class of Turing machines denoted $\mathcal{T}$, more precisely $\mathcal{T}(\mathbb{X}, \mathbb{A}, \mathbb{B}, \delta, x_0, \mathbb{F})$.

Intermezzo

Example of Turing machine

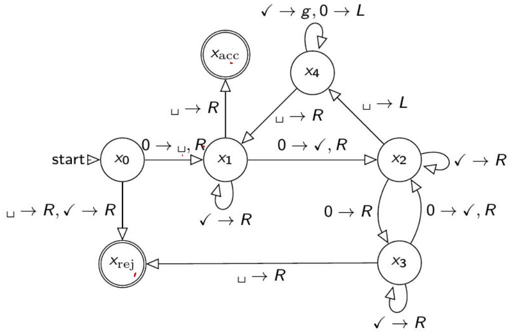


Figure: Directed graph description of Turing machine recognising whether input of the form 0^{2^n}, $n \in \mathbb{N}$.

Intermezzo
What does a Turing machine?

- Turing machine = abstract theoretical construction modelling algorithms that can be executed on classical computers.
- Configuration space $\mathbb{S} = \mathbb{X} \times \mathbb{A}^{\mathbb{N}} \times \mathbb{N}_{>}$. At initial time $t = 0$,
  - machine gets input word $\alpha = \alpha_1 \dots \alpha_{|\alpha|} \in \mathbb{A}^*$,
  - internal state of the machine is $X_0 = x_0$, and
  - the position of the head is at $P_0 = 1$ (1st cell of the tape),
- i.e. initial configuration $S_0 := (X_0, A_0, P_0) = (x_0, \alpha^0, 1)$.
- Suppose configuration at time t is $S_t := (X_t, A_t, P_t) = (x, \beta, p)$, denote $c = \alpha_p$ the letter at cell p of the tape.

$$\delta(x, c) = (x', c', D) \in \mathbb{X} \times \mathbb{B} \times \{-1, 1\} \Rightarrow S_{t+1} = (x', \alpha', p + D),$$

  where $\alpha' = \alpha_1 \dots \alpha_{p-1} c' \alpha_{p+1} \cdots$.
- Hence, function δ induces discrete time dynamical system on $\mathbb{S}$.

Intermezzo
How the Turing machine computes?

Definition

- Let $\tau := \tau_M(\alpha) = \inf\{t \geq 0 : X_t \in \mathbb{F}\} \in \mathbb{N} \cup \{+\infty\}$ be the **stopping time** of the machine[a].

  $\mathbb{F}_{acc} \cup \mathbb{F}_{rej}$
- If $\tau < \infty$ and
  - if $X_\tau = x_{\text{acc}}$ then input α is accepted,
  - if $X_\tau = x_{\text{rej}}$ then input α is rejected.
- Suppose α accepted. Stripping word A_τ from all blank cells, results in a finite word $\gamma \in \mathbb{A}^*$. γ is the result of the computation corresponding to input α.
- I.e. a Turing machine implements partial function

$$\mathbb{A}^* \ni \alpha \mapsto \text{Tur}_M(\alpha) = \gamma \in \mathbb{A}^*$$

  with domain

$$\text{dom}(\text{Tur}_M) = \{\alpha \in \mathbb{A}^* : \tau = \tau(\alpha) < \infty, X_\tau = x_{\text{acc}}\}.$$

- A function $f : \mathbb{A}^* \to \mathbb{A}^*$ is **computable** if there exists a Turing machine M such that $f = \text{Tur}_M$. Mind: $\mathbb{A}^*$ isomorphic to $\mathbb{N}$, hence $f : \mathbb{N} \to \mathbb{N}$.
- A Turing machine = a theoretical model that can compute anything a classical computer can conceivably compute. : Explain

[a] As a matter of fact, $t \geq 1$ because $x_0 \notin \mathbb{F}$.

Kolmogorov's complexity of a string

Definition

Definition

Let $\mathbb{A} = \{0, 1\}$, $\mathbb{A}^* = \cup_{n\in\mathbb{N}}\mathbb{A}^n$, $M \in \mathcal{T}$ Turing machine. **Kolmogorov's complexity** of $\alpha \in \mathbb{A}^*$ w.r.t. M:

$$K_M(\alpha) := \inf\{|\beta|, \beta \in \mathbb{A}^* \text{ and } \mathrm{Tur}_M(\beta) = \alpha\},$$

with convention $K_M(\alpha) = \infty$ if no such β exists.

Remark

- β must be thought as a programme that when fed as input to computer M produces output α.
- Komogorov's complexity: the minimum length over programmes that halt and print out α when run on computer M.
- Machine $U \in \mathcal{T}$ is **universal**, if for any other $M \in \mathcal{T}$, $\exists \gamma_M \in \mathbb{A}^*$, s.t.

$$\forall \beta \in \mathbb{A}^*, \mathrm{Tur}_M(\beta) = \mathrm{Tur}_U(\gamma_M \beta).$$

UNIVERSITÉ DE RENNES 1

Kolmogorov's complexity of a string
Universality of complexity

Theorem

If $U \in \mathcal{T}$ universal,

$$\forall M \in \mathcal{T}, \exists c := c_M : K_U(\alpha) \leq K_M(\alpha) + c_M, \forall \alpha \in \mathbb{A}^*.$$

Proof.

- Let $\beta := \beta_M$ be programme s.t. $\mathrm{Tur}_M(\beta_M) = \alpha$.
- From universality of U, there exists programme γ_M simulating computer M on U. Let $c := c_M = |\gamma_M|$.
- When string $\delta = \gamma_M \beta_M$ fed to U, then U starts by simulating M and then M uses β_M as input to produce α. Now

$$|\delta| = |\gamma_M| + |\beta_M| = c_M + |\beta_M|.$$

Hence

$$K_U(\alpha) = \inf_{\xi:\mathrm{Tur}_U(\xi)=\alpha} |\xi| \leq \inf_{\zeta:\mathrm{Tur}_M(\zeta)=\alpha} (c_M + |\zeta|) = c_M + K_M(\alpha).$$

□

Kolmogorov's complexity of a string
Universality of complexity (cont'd)

Theorem (Invariance theorem)

For every pair of universal machines $U, V \in \mathcal{T}$, there exists c, s.t.

$$\forall \alpha \in \mathbb{A}^*, |K_U(\alpha) - K_V(\alpha)| \leq c.$$

Proof.

By universality: $K_U(\alpha) \leq K_V(\alpha) + c_V$ and $K_V(\alpha) \leq K_U(\alpha) + c_U$. Hence

$$-c_U \leq K_U(\alpha) - K_V(\alpha) \leq c_V \Rightarrow |K_U(\alpha) - K_V(\alpha)| \leq c := \max(c_U, c_V).$$

□

Kolmogorov's complexity of a string
Conditional complexity

Definition

- A **pairing function** $\langle\ \cdot,\cdot\ \rangle : \mathbb{A}^* \times \mathbb{A}^* \to \mathbb{A}^*$ is the map defined by
$$\langle\ \alpha, \beta\ \rangle = 0^{|\alpha|}1\alpha\beta.$$
- The **conditional Kolmogorov complexity** of a α, given the hint γ, is
$$K_U(\alpha \mid \gamma) = \inf_{\beta}\{|\beta| : \mathrm{Tur}_U(\langle\ \alpha, \beta\ \rangle) = \alpha\},$$
if such a β exists, $+\infty$ otherwise.

Remark

- $K_U(\alpha \mid \varepsilon) = \inf_{\beta}\{|\beta| : \mathrm{Tur}_U(1\beta) = \alpha\} = K_U(\alpha)$.
- The hint γ is supposed to reduce the complexity of α

Kolmogorov's complexity of a string
Upper and lower bounds

Because of invariance, complexities w.r.t. different universal computers differ only by constant $\Rightarrow$ we can drop dependence on U.

Theorem (Upper bound)

There exist constants c and c', s.t. for all $\alpha \in \mathbb{A}^*$,

$$K(\alpha) \leq |\alpha| + \log|\alpha| + c$$

and

$$K(\alpha \mid \mathrm{rep}(|\alpha|)) \leq |\alpha| + c'.$$

Theorem (Lower bound)

Let $\mathbb{A} = \{0, 1\}$. For integer $k \geq 1$,

$$\mathrm{card}\{\alpha \in \mathbb{A}^* : K(\alpha) < k\} < 2^k.$$

Remark

Meaning of lower bound: although some (very few) input words have short descriptions, most of them have not.

Kolmogorov's complexity of a string
Complexity vs. entropy

Theorem (Asymptotically: average complexity = entropy)

Let $(X_k)_{k\in\mathbb{N}}$ be sequence of independent $\mathbb{X}$-valued r.v., with finite $\mathbb{X}$, and identically distributed with $\mathbf{p}$. Write $\mathbf{p}^{(n)}$ for probability vector of the joint law of n r.v., i.e. $\mathbf{p}^{(n)}(x_1,\ldots,x_n)=\prod_{k=1}^n p(x_k)$. Then, there exists constant c, s.t. for all n,

$$H(\mathbf{p}) \leq \frac{1}{n}\sum_{\mathbf{x}:=(x_1,\ldots,x_n)\in\mathbb{X}^n} \mathbf{p}^{(n)}(x_1,\ldots,x_n)K(\mathbf{x}\mid \mathrm{rep}(n)) \leq H(\mathbf{p}) + \frac{|\mathbb{X}|\log n}{n} + \frac{c}{n},$$

hence

$$\mathbb{E}\left(\frac{K(\mathbf{X}\mid \mathrm{rep}(n))}{n}\right) \to H(\mathbf{p}).$$

A question of the utmost importance

How to play "heads or tails"?

> "Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin. For, as has been pointed out several times, there is no such thing as a random number — there are only methods to produce random numbers, and a strict arithmetic procedure of course is not such a method."
>
> John von Neumann (1951),
> *Various techniques used in connection with random digits.*

- Kolmogorov's theorem states that there *exist*
  - an abstract probability space $(\Omega, \mathcal{F}, \mathbb{P})$,
  - on which can be defined an infinite independent sequence of random variables $X_n : \Omega \to \{0, 1\}$ such that $\mathbb{P}(X_n = 1) = 1/2$ for all $n \in \mathbb{N}$.
- But this answer is existential, not constructive. We have merely displaced the problem: how to simulate $(\Omega, \mathcal{F}, \mathbb{P})$?
- Not surprisingly, the person who more deeply searched for a convincing constructive answer to this problem was Kolmogorov himself [Kolmogorov1965].

5-A: Example.

Stochasticity, chaoticity, typicality
Three aspects of randomness

Sequence will be termed

- **stochastic**, if fulfills conditions of frequency stability,
- **chaotic**, if disordered with a Kolmogorov's complexity (another measure of its informational content) proportional to its length, and
- **typical**, if belongs to an effectively full measure set (in the sense that non-typical sequences belong to an effectively negligible set.)

Stochasticity
Frequency stability

- Let $\boldsymbol{\omega} = \omega_0\omega_1\omega_2\cdots$, with $\omega_i \in \mathbb{B} = \{0,1\}$, a binary infinite sequence and $\nu_b^{(n)}(\boldsymbol{\omega}) = \sum_{k=0}^{n-1} \mathbb{1}_{\{b\}}(\omega_k)$, for $b \in \mathbb{B}$. The sequence $\boldsymbol{\omega}$ is **frequencially stable** if $\lim \frac{\nu_b^{(n)}(\boldsymbol{\omega})}{n} = p_b$, where $p_b \in [0,1]$ and $\sum_{b\in\mathbb{B}} p_b = 1$.
- First attempt to define randomness = frequency stability [von Mises 1936, 1956, 1964]: sequence $\boldsymbol{\omega}$ is random if $\lim \frac{\nu_b^{(n)}(\boldsymbol{\omega})}{n} = \frac{1}{2}$.
- First order objection: $\boldsymbol{\omega} = 010101010101\cdots$ is frequencially stable but …does not look very random.
- First order correction: not only sequence but also subsequences must be frequentially stable. Here even $(0000\cdots)$ and odd $(1111\cdots)$ subsequences are not.
- Second order objection: frequency stability cannot be true for every subsequence. Eg. for $\boldsymbol{\omega} = (\omega_0\omega_1\omega_2,\cdots)$ construct integer sequence by

$$n_0 = \inf\{n \geq 0 : \omega_n = 0\}$$

and recursively, as long as n_{m-1} is finite,

$$n_m = \inf\{n > n_{m-1} : x_n = 0\}.$$

If $\inf\{m : n_m = +\infty\} = +\infty$, the subsequence $(\omega_{n_0}, \omega_{n_1}, \omega_{n_2}, \cdots)$ fails — by construction — to be frequentially stable.

Stochasticity

Frequency stability (cont'd)

Frequency stability not true for all subsequences but only legal ones.

Definition

Let $\gamma \in \mathbb{B}^*$ arbitrary finite binary word of length $\ell = |\gamma|$ and $\boldsymbol{\omega}$ an infinite binary sequence. Note

$$i_0 = i_0(\omega, \gamma) := \inf\{m \geq \ell : \omega_{[m-\ell:m]} = \gamma\} + 1$$

$$i_k = i_k(\omega, \gamma) := \inf\{m \geq i_{k-1} : \omega_{[m-\ell:m]} = \gamma\} + 1 \text{ for } k > 0 \text{ ifi } i_{k-1} < \infty.$$

Subsequence $\omega_{i_0}\omega_{i_1}\omega_{i_2}\cdots$ is a γ-**legal** subsequence of $\boldsymbol{\omega}$. **legal** subsequences are all γ-legal ones when $\gamma \in \mathbb{B}^*$.

Definition

A sequence is **stochastic** if all its legal subsequences are frequencially stable.

UNIVERSITÉ DE RENNES 1

Stochasticity
Frequency stability (cont'd)

Proposition

Let $\gamma = \gamma_1 \cdots \gamma_k$ an arbitrary finite word of lenght $|\gamma| = k$ and ξ a stochastic sequence generated by a Bernoulli law with parameter 1/2. Then

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{1}_{\{\gamma\}}(\xi_{[i:i+k-1]}) = \frac{1}{2^k}.$$

Proof.

- If $|\gamma| = 1$, the result is a consequence frequencial stability for infinite sequences.
- If $|\gamma| > 1$, proceed by recurrence. Suppose formula correct $\gamma \in \mathbb{B}^k$, $k \geq 1$, i.e. $\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{1}_{\{\gamma\}}(\xi_{[i:i+k-1]}) = \frac{1}{2^k}$. Every word γ inside ξ followed either by 0 or 1 and sequence of successors of γ is a legal subsequence of ξ, hence is frequencially stable. Therefore,

$$\lim_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} \mathbb{1}_{\{\gamma 0\}}(\xi_{[i:i+k]}) = \frac{1}{2^{k+1}} = \lim_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} \mathbb{1}_{\{\gamma 1\}}(\xi_{[i:i+k]}) = \frac{1}{2^{k+1}}$$

Stochasticity
Frequency stability (cont'd)

Remark

The previous proposition means

- every finite word must appear infinitely many times inside an infinite stochastic word,
- in particular, every infinite stochastic sequence is Borel-normal,
- the r.h.s. limit appearing in the proposition can be obtained as an almost sure result of the strong law of large numbers,
- however, here it holds for all stochastic sequences. It is as if stochastic sequences were the subset of the universe stripped from the exceptional sequences (of measure zero) that are precisely ignored by the "almost sure" proviso of the strong law of large numbers.

Chaoticity
Kolmogorov's omplexity

- Intuitively: easier to describe a sequence of 1000000 bits 0 than a random sequence of 1000000 outcomes of a honest coin.
- Because first sequence described by the very short sentence "1000000 bits 0" while the second requires the full display of the sequence.
- Kolmogorov's definition of random sequence as one that is intrinsically algorithmically difficult to describe it shortly.
- Uses equivalence of algorithm with Turing machine to give precise definition of chaoticity in terms of Turing machines.

Chaoticity

Kolmogorov complexity (cont'd)

$x_{n+1} = (\underset{16807}{a} x_n + \cancel{b}) \bmod \underset{m}{\underline{2^{31}-1}}$ $\quad \left(\frac{x_n}{m}\right) \in [0,1]$

Definition

Denote $\mathbb{A} = \{0,1\}$, $\mathbb{A}^* = \cup_{n\in\mathbb{N}}\mathbb{A}^n$, $\mathcal{T}$ set of Turing machines, and rep : $\mathcal{T} \to \mathbb{A}^*$ their binary coding.

- **Kolmogorov's complexity** of $\alpha \in \mathbb{A}^*$:

  $K(\alpha) := \inf\{|\mathrm{rep}(M)\beta| : M \in \mathcal{T}, \beta \in \mathbb{A}^*$, s.t. on input β, M halts and $\mathrm{Tur}_M(\beta) = \alpha\}$.

- Sequence α is **chaotic**, if

$$K(\alpha) = \mathcal{O}(|\alpha|).$$

Remark

- Previous results have shown that there exist chaotic sequences as well as ones with short description.
- The previous definition implies that no algorithm run on a classical computer can generate chaotic sequence.

UNIVERSITÉ DE RENNES 1

Chaoticity

Impossibility of classical randomness

- Previous result excluded the existence of classical algorithmic randomness.
- Nagging question: is it possible to generate classical true randomness?

Example (Coin tossing revisited)

- Coin viewed as solid body subject to laws of motion.
- Coin idealised as disk with no thickness of radius R and mass m. Its barycenter coincides with geometrical center.
- An initial impulse is exerted on the coin resulting to an initial vertical velocity v_z and an angular velocity α around a rotation axis lying on the disk plane and passing through its center.
- Afterwards, coin evolves subject to earth's gravity following Newton's equations from time $t = 0$ to the first time t_0 it touches the soil (assumed perfectly plastic to stop the coin instantaneously.)
- Equations of motion

$$\frac{\mathrm{d}^2 z}{\mathrm{d}t^2}(t) = -g, \text{ with initial conditions: } z(0) = R, \frac{\mathrm{d}z}{\mathrm{d}t}(0) = v_z,$$
$$\frac{\mathrm{d}^2 \theta}{\mathrm{d}t^2}(t) = 0, \text{ with initial conditions: } \theta(0) = 0, \frac{\mathrm{d}\theta}{\mathrm{d}t}(0) = \alpha,$$

have solution

$$z(t) = v_z t - \frac{1}{2} g t^2 + R; \quad \theta(t) = \alpha t, t \in [0, t_0].$$

Chaoticity
The coin tossing machine

In [DiaconisHolmesMontgomery2007] the previous setting has been physically realised!

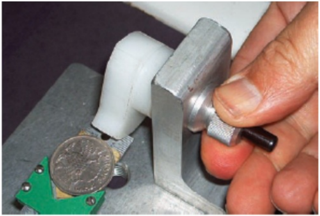



Figure: The coin tossing machine.

Chaoticity

Impossibility of classical randomness (cont'd)

Example (Coin tossing revisited(cont'd))

- t_0 positive solution of $z(t_0) - R|\sin\theta(t_0)| = 0$.
- Coin shows up "heads" if

$$2n\pi - \frac{\pi}{2} < \theta(t_0) < 2n\pi + \frac{\pi}{2}, n \in \mathbb{N}.$$

- Pre-images of "heads" the pairs $(v_z, \alpha) \in \mathbb{R}^2_>$ that show up "heads", i.e. $\alpha t_0 \in [(2n - \frac{1}{2})\pi, (2n + \frac{1}{2})\pi]$.
- Introducing variable $\zeta = \frac{v_z}{g}$ (with dimensions of time), the family of equations

$$\alpha = (2n \pm \frac{1}{2})\frac{\pi}{2}\zeta, n \in \mathbb{N}_>$$

delimits in the (α, ζ)-plane the alternating loci of parameters for which coins end up "heads" or "tails".

Chaoticity

And the solution reads . . .

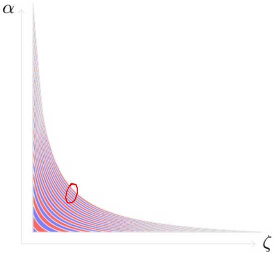


Figure: The phase space (α, ζ), where α initial angular velocity and the initial parameter $\zeta = \frac{v_z}{g}$, where v_z is the initial vertical velocity g the gravity acceleration.

Chaoticity
Lessons from this example

- Apparent randomness due to lack of complete information on initial condition.
- Our fingers too crude to control initial impulse precisely.
- If initial condition known with infinite precision, no randomness.
- In principle: classical randomness is reducible.
- Only true randomness in Nature of quantum origin because quantum randomness is intrinsic and irreducible.