NEW ZEALAND JOURNAL OF MATHEMATICS
Volume 52 (2021), 453–467
https://doi.org/10.53733/129

TOWARDS UNDERSTANDING THE THEORETICAL CHALLENGES OF NUMERICAL MODELING OF DYNAMICAL SYSTEMS

Michael Yampolsky

(Received 23 June, 2021)

Abstract. Asking and answering the right questions about computability and computational complexity of dynamical systems is more important than many may realize. In this note, I discuss the principal challenges, their implications, and some instructive open problems.

1. Preface

It is a privilege to contribute to the volume in memory of Vaughan Jones. I was one of the invited speakers at NZMRI Summer meeting in January 2009. These "Summer schools" were a wonderful initiative of Jones', leading to the creation of NZMRI which he co-founded and chaired. For me it was a remarkable trip. Although I delivered a series of lectures, I also felt a bit like a student, being a newcomer to the questions of algorithmic computability and complexity which were the focus of the meeting. The study of computational complexity in dynamics was making its first baby steps (it is only a little more mature now). It was fun to talk to a receptive audience about results which I was finding exciting, and it was instructive for me to see it as a part of the broader development of the field of algorithmic complexity in continuous (as opposed to discrete) mathematics. It helped me to focus on what I see as the main theoretical challenges of modeling dynamics on a computer, which I attempt to convey in this paper.

2. Motivation

In discrete mathematics, techniques of numerical modeling and theoretical study of computational complexity of the corresponding problems go hand-in-hand. Indeed, typically *the same people* develop practical algorithms and establish rigorous complexity bounds. The culture of modeling continuous phenomena is completely different. Thus, practitioners bravely use computers to model nuclear explosions, weather systems, or, indeed, the Earth's climate decades from now without ever asking whether such modeling is *theoretically possible.* I see two principal reasons for this:

(I) the modern development of the mathematical study of dynamical systems has been led by computer experiments since the 1950's. Numerical studies have informed the general paradigm of the expected behaviour of typical dynamical systems. In a way, this has become a self-fulfilling prophecy – we look to model natural phenomena using mathematics which was developed to interpret the results of the modeling;

(II) numerical experiments of this nature seem to "work" – that is, they produce plausible results which agree with the established paradigm.

In this note I aim to describe the modern paradigm of numerical modeling of dynamical systems, and its limitations from the point of view of computational complexity and computability, posing questions on what can and cannot be computed in theory. I also speculate on what happens when computer simulations of complex natural processes are performed, why the results appear to make sense, and what they actually describe.

3. Modern Paradigm of Numerical Modeling of a Dynamical System

3.1. Computers versus chaos. In what follows, I will only consider *discrete-time* dynamical systems on a finite-dimensional space. The latter condition means that a state of the system can be described by finitely many parameters, conveniently represented as coordinates of a point in $\mathbb{R}^m$. Discreteness means that the evolution of the system will be represented by a map F from a domain $\Omega \subset \mathbb{R}^m$ to $\mathbb{R}^m$: in one tick of the clock a point $\bar{x} \in \Omega$ will transition to $F(\bar{x})$. Assuming the successive images of $\bar{x}$ remain in Ω, we can continue applying iterates of F, to obtain the orbit

$$\bar{x} \mapsto F(\bar{x}) \mapsto F(F(\bar{x})) \mapsto \cdots \mapsto \underbrace{F(\cdots(F}_{n \text{ times}}(\bar{x})\cdots) \equiv F^{\circ n}(\bar{x}) \mapsto \cdots$$

which represents the long-term evolution of the system starting at $\bar{x}$. This is in line with actual computational practice. Even if studying a continuous-time dynamical system, which is the flow [1] $\Phi(t;\bar{x}_0)$ of a system of differential equations

$$\bar{x}' = G(\bar{x}), \tag{3.1}$$

in practice one would use a numerical method to estimate

$$F(\bar{x}_0) = \Phi(\Delta t; \bar{x}_0)$$

for some suitably small Δt.

When computing the values of $F(\bar{x})$, by necessity, the real point $\bar{x}$ has to be rounded up to a rational approximation $\bar{x}_r$. Round-off errors in numerical experiments have inadvertently led to the empirical discovery of "chaos", the key to the modern theory of dynamical systems. The most famous and, probably, the most influential of these experiments was performed by a meteorologist E. Lorenz in 1961 [**16**]. To save time, Lorenz attempted to reproduce a simulation of a weather pattern, using the output corresponding to some intermediate point in time. The predictions of the model were completely different from the original calculation. The culprit: a round-off error introduced when outputting the intermediate results.

In retrospect, one of the most surprising aspects of this and other similar experimental discoveries is that they *were* surprising. The general question of (in)stability in dynamical system was investigated by A.M. Lyapunov [**18**] in the late 1800's. An example of what we would now describe as chaotic dynamics was discovered

[1] That is, the solution of (3.1) with the initial condition $\bar{x}(0) = \bar{x}_0$ is given by $\bar{x}(t) = \Phi(t;\bar{x}_0)$; we implicitly assume that existence and uniqueness of the solution is guaranteed.

around the same time by H. Poincaré in his seminal study [**28**] of the Three-Body Problem of the relative motion of the Sun, the Earth, and the Moon.[2]

Rounding off does not need to lead to a dramatic change in the results of the numerical experiments, if the orbit of $\bar{x}$ is *Lyapunov-stable*. For instance, the contracting one-dimensional map $F(x) = 0.5x$ has a single fixed point $x_0 = 0$ which attracts the orbits of every point $x \in \mathbb{R}$. Round-off error is decreased by half with every iterate, thus the precision of the calculation actually improves (by a single dyadic bit) with each step. The scenario when an orbit converges to an attracting periodic point is commonly referred to as *regular* dynamics – it presents no obstacles to numerical modeling.

To see how things can go wrong, note first that if the map F is smooth on Ω, and $||DF||$ is bounded by some $M > 0$, when we replace $\bar{x}$ with a rounded off value $\bar{x}_r$, the distance $||\bar{x} - \bar{x}_r||$ may grow at most M times after one iterate. But, if this happens at every step of the iteration, then in order to calculate $k = [\log_2 M]$ steps of the orbit of $\bar{x}$, we have to increase the precision of the computation by one dyadic bit – which will quickly become prohibitive.

As an elementary example, consider the doubling map $\mathbf{D}(x) = 2x \bmod 1$ defined on the circle $\mathbb{T} = \mathbb{R}/\mathbb{Z}$. Since $\mathbf{D}'(x) = 2$ at every point, iterating one step further requires adding one binary digit to the precision of computing the function, so an orbit cannot be followed too long in practice, the phenomenon known as *sensitive dependence on initial conditions* which is the main hallmark of "chaos". And the behavior of an orbit can be complicated indeed: some of the sequences $\{\mathbf{D}^n(x)\}_{n\geq 0}$ are dense in the circle $\mathbb{T}$ (for instance, if x is a number whose binary expansion consists of all possible finite binary sequences ordered lexicographically). This is another common feature of chaos, known as *transitivity.*

Tracing individual orbits of $\mathbf{D}(x)$ on a computer is out of reach. However, it is easy to describe statistical properties of *typical* orbits. Recall, that a number $x \in \mathbb{T}$ is *normal base b* if for any string $\Theta = \theta_1\theta_2 \ldots \theta_n$ of base b digits of length $n \in \mathbb{N}$, denoting $N_l(\Theta, x)$ the number of times Θ appears as a substring in the first l digits in base b expansion of x, we have

$$\lim_{l\to\infty} \frac{N_l(\Theta, x)}{l} = b^{-n},$$

that is, all finite strings appear with equal frequency in the base b expansion of x. This concept was introduced by Émile Borel, who showed that almost all $x \in \mathbb{T}$ have this property (this is a consequence of Borel-Cantelli Lemma). Wall [**35**] proved that normality is equivalent to the fact that the sequence $\{b^n x\}$ is uniformly distributed in $\mathbb{T}$. Given an arc $A \subset \mathbb{T}$ and a point $x \in \mathbb{T}$, we will generally not be able to identify the iterates $\mathbf{D}^n(x) \in A$, however, the fraction of time the orbit of a typical point x spends in A is equal to the length of A.

[2] This situation is far from unique in the history of the subject. In 1951, a Russian chemist B.P. Belousov made an experimental discovery of a periodically oscillating chemical reaction. Quasiperiodic behaviour is central to the study of flows in two dimensions since the work of H. Poincaré (Poincaré-Bendixson Theorem is a standard feature of textbooks on differental equations). Still, Belousov's discovery was met with a complete disbelief by fellow practitioners, and was only accepted 15 years later after it was "defended" by a young Russian chemist A.M. Zhabotinsky. Oscillating chemical reactions are now favorite examples of applications of Poincaré-Bendixson Theorem in textbooks.

3.2. Attractors and physical measures. To formalize the discussion of statistical averages, let us say that a probability measure μ on Ω is a *physical measure* of F if the set $B(\mu) \subset \Omega$ (the *basin* of μ) of points $\bar{x}$ for which

$$\frac{1}{n}\sum_{m=1}^{n} \delta_{F^{\circ m}(\bar{x})} \underset{\text{weak}-*}{\longrightarrow} \mu$$

has positive Lebesgue measure. A physical measure which is ergodic is known as *Sinai-Ruelle-Bowen (SRB) measure.* The support of an SRB measure is an *attractor* for F. Regular dynamics gives one such example: if

$$\bar{x}_1 \mapsto \bar{x}_2 \mapsto \cdots \bar{x}_p \mapsto \bar{x}_1$$

is an attracting periodic orbit of F, then the combination of delta-masses

$$\mu = \frac{1}{p}\sum_{k=1}^{p} \delta_{\bar{x}_k}$$

is an ergodic physical measure. The unique SRB measure of $\mathbf{D}(x)$ is the Lebesgue measure on the circle, which is another simple example. Attractors (and measures) may be much more complicated, however, such as the Lorenz “butterfly” attractor made famous by E. Lorenz’s discovery, whose existence was rigorously proved by W. Tucker [**32**] (see Figure 1).

The principal conjecture (see [**27**]) in the field can be formulated as follows:

Finitness of attractors conjecture. *For a typical dynamical system F almost every orbit lies in the basin of an SRB measure, and there are finitely many such measures.*

Typicality is understood with respect to Lebesgue measure in a finite-parameter family of maps, presumably given by an application of a physical law, and could

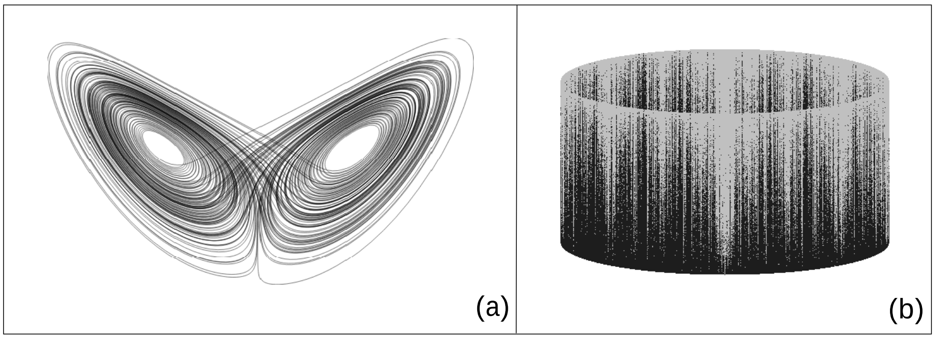

FIGURE 1. (a) Lorenz butterfly attractor. (b) Two intermingled basins of attraction for a diffeomorphism of the cylinder [**12, 3**]; picture courtesy A. Bonifant and J. Milnor. The upper and lower circles are the measure-theoretic attractors, each supporting an SRB measure. Each of the two basins has positive Lebesgue measure in every open set; the shading reflects their local prevalence.

be suitably interpreted in an infinite dimensional space of all maps of a given regularity. This statement is clearly motivated by the numerical experimentation and in its turn informs how we approach computer experiments. There is a lot of supporting evidence in its favor (more on that below), but there have also been voices of caution. Notably, D. Ruelle [**31**] has discussed examples of maps for which the averages

$$\frac{1}{n}\sum_{m=1}^{n}\delta_{F^{\circ m}(\bar{x})}$$

do not converge for points $\bar{x}$ in some set of positive Lebegue measure, and conjectured that in high enough dimension such dynamical systems may occupy open subsets of the parameter space. But at this point no convincing evidence of this has been offered, to my knowledge.

To round off the discussion, let me turn now to the principal example of discrete dynamical systems, which has attracted more attention than the rest of the field combined: quadratic polynomial maps.

3.3. The star of the show. Studying the dynamics of quadratic polynomials has kept everyone in the subject busy from the 1970's to this day. Since linear changes of coordinate turn one quadratic into another, two out of the three coefficients of a degree 2 map can be normalized to our liking. Most commonly, the family

$$f_\lambda(x) = 4\lambda x(1-x) \tag{3.2}$$

is considered; with the additional restriction $\lambda \in [0,1]$, the function f_λ maps the closed interval $[0,1]$ to itself. In this family, a very strong version of the Finitness conjecture holds. To formulate it, let us first introduce some terminology. As usual, the *ω-limit set* $\omega(x)$ of a point $x \in [0,1]$ is the set of limit points of its orbit:

$$\omega(x) \equiv \bigcap_{k\geq 0}\overline{\cup_{n\geq k} f_\lambda^{\circ n}(x)}.$$

Let us say that f_λ is *renormalizable with period* p if there exists a periodic cycle of disjoint intervals

$$\mathcal{C} = (I_0,\ldots,I_{p-1}) \text{ where } f_\lambda(I_j) = I_{j+1 \bmod p}.$$

It is *infinitely renormalizable* if there exists an infinite nested sequence of cycles of periodic sub-intervals of $[0,1]$: $\mathcal{C}_0 \supset \mathcal{C}_1 \supset \mathcal{C}_2 \supset \cdots$ with increasing periods. We say that the *Feigenbaum-like Cantor set* of an infinitely renormalizable polynomial is the intersection $\cap_{k\in\mathbb{N}}\mathcal{C}_k$; it actually is a Cantor set in this case.

There are only three possibilities for the structure of the ω-limit set of a typical orbit of a real quadratic polynomial (see [**19**] where the classification was completed, and references therein):

Theorem 3.1. *Let* $\lambda \in [0,1]$*. Then there is a unique set* $\mathcal{A}_\lambda$ *(a measure-theoretic attractor in the sense of Milnor* [**24**]*) such that* $\mathcal{A}_\lambda = \omega(x)$ *for Lebesgue almost all* $x \in [0,1]$*, and only one of the following three possibilities can occur:*

(1) $\mathcal{A}_\lambda$ *is an attracting periodic cycle;*
(2) $\mathcal{A}_\lambda$ *is a periodic cycle of intervals;*
(3) $\mathcal{A}_\lambda = \omega(0.5)$ *is a Feigenbaum-like Cantor set.*

In all of the above cases, $\mathcal{A}_\lambda = \omega(x)$ *for a dense-*G_δ *set of points in* $[0,1]$ *(the topological attractor of* f_λ *in the sense of Milnor* [**24**]*).*

Furthermore, M. Lyubich [**20**] showed that:

Theorem 3.2. *For almost every $\lambda \in [0,1]$, the attractor $\mathcal{A}_\lambda$ is as in (1) or (2) of Theorem 3.1, is equal to $\omega(0.5)$, and is the support of the unique SRB measure μ_λ of f_λ, whose basin has full measure in $[0,1]$. Moreover, in case (2), the measure μ_λ is absolutely continuous on each of the intervals forming $\mathcal{A}_\lambda$.*

Thus, for almost all parameters λ, the maps of the family f_λ possess a unique SRB measure μ_λ supported on an attractor $\mathcal{A}_\lambda$ whose structure is well understood, and which attracts typical (in every reasonable sense) orbits of f_λ. One potential cause of concern is the instability of the statistical properties of orbits with respect to λ. M. Misiurewicz showed in [**25**], that there exists a test function $\phi : [0,1] \to \mathbb{R}$, a set $S \subset [0,1]$ of positive measure, and $\epsilon > 0$, such that for any $\lambda \in S$ and a neighborhood $U \subset [0,1]$ of λ, there is a set $Z \subset U$ of positive measure with

$$\left| \int \phi d\mu_\lambda - \int \phi d\mu_\ell \right| > \epsilon \text{ for all } \ell \in Z,$$

the property which he called *strong structural instability*, and which is likely common in other natural families of examples.

Quadratic polynomials play a similarly central role in the study of dynamics of functions of one complex variable. In this case, the normalization

$$P_c(z) = z^2 + c, \text{ with the parameter } c \in \mathbb{C}$$

is more common. A polynomial P_c has a single *repeller* J_c, which attracts all of the inverse orbits

$$z \underset{P_c}{\hookleftarrow} z_{-1} \underset{P_c}{\hookleftarrow} z_{-2} \underset{P_c}{\hookleftarrow} \cdots$$

except at most one[3]. It is called the *Julia set* of P_c. Again, for all points $z \in \mathbb{C}$ but at most one, the weighted averages

$$\frac{1}{2^k} \sum_{f^{\circ k}(w)=z} \delta_w$$

weakly converge to a measure ω_c whose support is equal to J_c. It is known as *Brolin-Lyubich measure* and coincides with the harmonic measure of the Julia set seen from the point at infinity. The subject of computability and computational complexity of Julia sets is rich and subtle, to keep the discussion focused I will mostly leave it out of this note. I refer the interested reader to my monograph with M. Braverman [**8**], and also the paper [**2**], in which the computability of ω_c is addressed.

4. Computability and Computational Complexity: Briefly

The formal model of computation widely accepted in computer science is a Turing machine (TM); it is provably equivalent to a program in any common programming language running on a computer with unlimited storage. I will use the terms *TM, computer program,* or *an algorithm* interchengeably. Definitions of computability of continuous objects took a while to develop; the subject is not without controversy (see [**4**] for a review). The first step was taken by Turing in his foundational work

[3] The only quadratic map which has such an exceptional inverse orbit is $P_0(z) = z^2$. The unique inverse orbit of 0 under P_0 remains stuck at 0 and does not converge to $J_0 = \{|z| = 1\}$

[**33**], in which he defined a *computable real number* x as a number for which there exists a TM which takes a single input $n \in \mathbb{N}$ and outputs a rational number d_n such that $|x - d_n| < 2^{-n}$. The bound 2^{-n} can be replaced by any other quantity that converges to 0 with n in a constructive fashion – for instance, using $\frac{1}{n}$ in its stead will give an equivalent defintion. This specific choice reflects the fact that modern computers think in binary. If $d_n \in \mathbb{Z}[2^{-1}]$ is a dyadic rational, represented in base 2 as a finite string of dyadic bits, then we can think of it as an approximation of x to the n-th bit. Most numbers are not, in fact, computable, since in any programming language there can be only countably many programs; in his paper, Turing gave a "specific" example of a non-computable real. Starting with computable reals, one can define, for instance, computable complex numbers, computable points in $\mathbb{R}^m$ and so on.

Turing's idea clearly corresponds to the actual computational practice, and gives the direction for further development of computability in analysis. It is interesting to note (and is not as well known) that further progress owes much to S. Banach and S. Mazur [**1, 21**]. Most of the subtlety in this development lies in the difficulty of defining a computable function $f(x)$ of a real variable x. Since only countably many real numbers x are computable, one has to somehow account for the inputs which are not. The modern approach again corresponds to the actual computing practice:

a real function $f(x)$ is computable if there is an algorithm to approximate the value of $f(x)$ with an arbitrary desired precision, assuming that the value of x can be read by the algorithm with whatever precision is required.

To formalize this, let us say that a function $\phi : \mathbb{N} \to \mathbb{Q}$ is an *oracle* for x if $|x - \phi(x)| < 2^{-n}$. An oracle for $z \in \mathbb{C}$ or $\bar{x} \in \mathbb{R}^m$ is defined in the same way. A TM may be allowed to *query* an oracle, that is, to read the values $\phi(n)$. A real function f is computable if there exists a TM which takes a single input $n \in \mathbb{N}$ and, given an oracle for x in the domain of definition of f, outputs $d_n \in \mathbb{Q}$ such that $|f(x) - d_n| < 2^{-n}$.

If all of this sounds a bit confusing, think, for example, of the problem of computing the value of the exponential function $\exp(x)$ with a given precision 2^{-n}. To fix the ideas, let us suppose that we use the Maclaurin series for the exponential. The formula for the remainder term is

$$R_m(x) = \frac{\exp(c)x^m}{m!},$$

where c is a real number between 0 and x. The algorithm can first ask the user for a rough approximation of the value of x to get an integer upper bound $B > |x| + 1$ for it (in practice, a uniform such bound may be imposed by restricting the range of allowed inputs x). The value of $|R_m(x)|$ can be bounded from above crudely by

$$E_m = \frac{3^B B^m}{m!}.$$

From this, the algorithm can calculate the value of m needed to get $E_m < 2^{-(n+1)}$, and the precision 2^{-s} with which x needs to be known to compute the sum of the first m terms of the Maclaurin series with error of at most $2^{-(n+1)}$. Now the user is again asked to input the value of x, this time with precision 2^{-s}, and the calculation is completed. This is cumbersome, of course, but the key point here is that it can actually be done for the exponential function, so there is no theoretical obstacle to approximating the value $\exp(x)$ if we set our mind to it.

In our discussion of computability of attractors in the quadratic family, we will talk about computable compact sets in the plane. In practice, by computing a set, we mean visualizing it on a computer screen. A visualization is a finite collection of pixels. If we fix a specific pixel size (commonly taken to be 2^{-n} for some n) then to accurately draw the set within one pixel size, we should fill in the pixels which are close to the set (for instance, within distance 2^{-n} from it), and leave blank the pixels which are far from it (for instance, at least $2^{-(n-1)}$-far). Thus, for the set K to be computable, there has to exist an algorithm which for every square of size 2^{-n} with dyadic rational vertices correctly decides whether it should be filled in or not according to the above criteria. [4] This definition easily generalizes to subsets of $\mathbb{R}^m$ for any m.

Suppose that we are trying to compute the attractor $\mathcal{A}_\lambda$ of $f_\lambda(x)$. Then we should assume that the algorithm has an access to the oracle for λ. In other words, we will treat the dependence of the attractor on λ as a function

$$\Phi : \lambda \mapsto \mathcal{A}_\lambda,$$

and will ask whether it is computable at a point $\{\lambda_0\}$, or, more generally, on some subset of $[0,1]$. This is completely consistent with the computing practice – if we would like to compute an object (an invariant set, an SRB measure, etc) associated with a dynamical system $F : \Omega \to \mathbb{R}^m$, then we expect the algorithm to "know" what the function F is, that is, to be able to approximate the values of F with any given precision.[5]

To talk about computability of measures, we can follow the same strategy, using approximations by a dense subset of explicitly defined measures, such as rational combinations of delta-masses at rational points, for instance. However, the way it typically is with measures, it is more convenient to take the dual approach. We will say that *a measure μ is computable* if the following is true. There is a TM which, given a computable test function ϕ (as a subprogram, or a black box) and $n \in \mathbb{N}$ outputs $d_n \in \mathbb{Q}$ such that

$$\left| \int \phi d\mu - d_n \right| < 2^{-n}.$$

I would like to emphasize that all of the above definitions are consistent with the actual computing practice. While it is true that practitioners will often cut corners and not worry too much about the round-off error in the calculation, at least, for a computable object (real number, value of a real function, an integral over a measure, etc) the calculation can in principle be carried out with the desired precision. For a non-computable object no algorithm can do the job, all of them will provably fail. Moreover, they must fail in such a way that we would not have a systematic way (an algorithm) of detecting it (which would otherwise enable us to pick a correct finite precision approximation by discarding the incorrect ones).

[4] Note that we let the algorithm treat the "in-between" pixels in an arbitrary fashion, to reflect the fact that a finite-precision computation may have round-off errors. We only require that these errors are small enough, so the "fuzz" around the approximation of the set is at most one pixel thick.

[5] Think of having a black box which, given an oracle for x, outputs the values of $F(x)$ with any desired precision.

An extremely important practical consideration for a computable object is its computational complexity, since computing resources are finite in practice. In this note I will only address the time complexity of the computations. A complexity bound is expressed by a function $T : \mathbb{N} \to \mathbb{N}$, whose argument is the desired precision, and the output is a bound on the computing time. For instance, we say that the complexity of a set $K \Subset \mathbb{R}^2$ is bounded above by $T(n)$ if there exists an algorithm to visualize K on a computer screen, and $N \in \mathbb{N}$ such that for every $n \geq N$ and each pixel of size 2^{-n} the time that the algorithm takes to decide the pixel is bounded above by $T(n)$. If this is not the case, we say that $T(n)$ is a lower complexity bound for the set. In other words, the time complexity is bounded below by $T(n)$ if for every program computing the set, there exists an infinite sequence of pixels P_{n_k} of size 2^{-n_k} such that the time it takes the program to decide P_{n_k} is greater than $T(n_k)$.

You will have noticed that the above definition of complexity depends on the programming language that is used, the computation may run faster in some of them than in others. However, changing the language changes the computing time for each step by at most a constant multiple – and we are generally only interested in the rate at which $T(n)$ increases with n. Any computation whose time grows faster than polynomial in n becomes impossible in practice for large values of n.[6] Problems with polynomial-time complexity are generally considered to be computably tractable; having a polynomial upper bound for time complexity is not the same, of course, as having an efficient algorithm, but the two usually go hand-in-hand. If the lower bound on $T(n)$ is, for example, exponential in n, then there is no realistic hope of computing a high-precision approximation of a continuous object, no matter what we try.

5. Can the Statistical Behavior of a Dynamical System be Modeled on a Computer?

5.1. Monte Carlo modeling. Lorenz' observations on the difficulty of modeling the weather have not put an end to weather modeling. Instead, weather forecasts are made in the language of statistics (e.g. there is a 40% chance of rain tomorrow). The probabilistic approach to modeling is consistent with the paradigm which we have presented above, in which a typical dynamical system has finitely many stable limiting regimes into which almost all orbits eventually fall. The technique itself, known as *Monte Carlo method*, dates to the very dawn of computer modeling of natural processes. It was pioneered by S. Ulam and J. von Neumann during their work in Los Alamos in 1946 [**34, 23, 22**] to model neutron diffusion. Informally speaking, given a numerical simulation of a dynamical system $F : \Omega \to \mathbb{R}^m$, one can throw random darts to select a large number of initial values; run the simulation for the desired duration for each of them; then statistically average the outcomes. We then expect these averages to reflect the true average statistics of our system.

More formally, let $\bar{x}_1, \ldots, \bar{x}_k$ be k points in Ω randomly chosen, for some $k \gg 1$ and consider the probability measure

$$\mu_{k,n} = \frac{1}{kn} \sum_{l=1}^{k} \sum_{m=1}^{n} \delta_{F^{\circ m}(\bar{x}_l)}, \tag{5.1}$$

[6] This is equally true if $T(n)$ is a polynomial of a high degree.

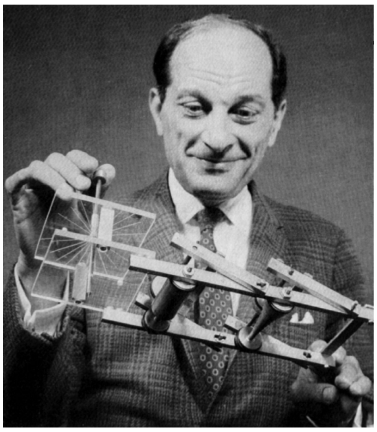

FIGURE 2. Stanisław Ulam holding the FERMIAC – an analogue device for performing Monte Carlo simulation of neutron diffusion designed by E. Fermi.

where $\delta_{\bar{x}}$ is the delta-mass at the point $\bar{x} \in \mathbb{R}^n$. The expectation is then that for $k, n \to \infty$ the probabilities $\mu_{k,n}$ converge to a limiting statistical distribution that we can use to make meaningful long-term *statistical* predictions of our system. If the finitness of attractors postulate holds for F, then the limit of the sums (5.1) is almost surely equal to the weighted average of the physical measures, where the weights correspond to the proportions of Ω occupied by the basins of the measures. In particular, if there is a unique attractor with an SRB measure in our dynamical system, as is the case for a typical quadratic map f_λ, then one random dart to pick an orbit in (5.1) will suffice to obtain this measure in the limit. Of course, there are examples of systems with infinitely many physical measures. For instance, Newhouse [**26**] showed that a polynomial map f in dimension 2 can have infinitely many attracting basins, on each of which the dynamics will converge to a different stable periodic regime. This in itself, however, is not necessarily an obstacle to the Monte-Carlo method, since the weighted average of the attracting basins is still a meaningful statistic. Indeed, the empirical belief is that Monte Carlo method still succeeds in these cases.

5.2. When limiting statistics cannot be computed. It is hard to overestimate the importance of Monte Carlo simulations in computing practice, it is the go-to tool of numerical simulations. In light of this, it was particuarly surprising for C. Rojas and I [**30**] to discover the following:

Theorem 5.1. *There exist uncountably many parameters $\lambda \in [0, 1]$ such that:*

- *the quadratic map f_λ has a unique SRB measure μ_λ whose basin has full measure in $[0, 1]$;*
- *the measure μ_λ is not computable by a TM with an oracle for λ.*

Monte Carlo method (as well as any other numerical algorithm) would thus definitely fail to compute the limiting statistics of the orbits – even though almost every orbit has the same limiting statistics.

The measures μ_λ constructed in [**30**] are not absolutely continuous with respect to the Lebesgue measure, however, this does not seem to be a distinguishing feature. It is not yet known whether absolutely continuous non-computable examples exist in this family, but I suspect that this is the case.

The first instinct of a theoretical computer scientist would be to ask whether a value of λ with non-computable μ_λ can be made computable. While we have not proved that it is so, it is very likely to be the case. In fact, the proof of Theorem 5.1 has little to do with the parameter λ itself, but rather is based on a very precise description of the instability of the dependence $\lambda \mapsto \mu_\lambda$, similar in spirit to the strong structural instability of [**25**]. Instability does therefore become a theoretical barrier to computing the limiting distribution.

A more important question to ask is whether such parameters λ can form a set of positive Lebesgue measure. The answer to this is likely "no": we expect μ_λ to be computable for almost all λ. The set of parameters produced in [**30**] has zero Lebesgue measure. However, the situation may change when one considers higher dimensional dynamical systems – it is possible that there exist large sets of parameters for which limiting statistics cannot be computed.

Even if in the quadratic family f_λ, the measure μ_λ is almost always computable, would it almost always be computable in polynomial time? The existence of non-computable examples may forecast trouble.

Modelers should be at least aware that even in the simplest families with the nicest limiting statistical properties, numerical methods such as Monte Carlo may fail for reasons of non-computability or high computational complexity. There is no practical way around this obstacle. The limiting distribution μ_λ for maps in Theorem 5.1 is truly unknowable.

6. Can one Compute an Attractor as a Set?

To be more precise, can we compute the set of limit points of a typical orbit? This is another very natural question from the point of view of modeling. It is clearly complementary to the statistical modeling discussed above, and the evidence is mixed as we will see below.

In the family f_λ, together with C. Rojas [**29**] we have shown:

Theorem 6.1. *The attractor $\mathcal{A}_\lambda$ is always computable by a TM with an oracle for λ. However, for any lower bound $T : \mathbb{N} \to \mathbb{N}$, there exists a parameter $\lambda \in [0, 1]$ such that the attractor $\mathcal{A}_\lambda$ is a Feigenbaum-like Cantor set whose computational complexity (given an oracle for λ) is bounded below by T.*

By M. Lyubich's Theorem 3.2, Feigenbaum-like attractors only occur for a set of parameters of measure zero, and indeed, as shown in [**29**], in all other cases the attractor is computable in polynomial time, given an oracle for λ. I note that despite this encouraging fact, the mechanism of creating a high-complexity attractor in Theorem 6.1 is quite robust, and it is again likely that in higher-dimensional families of maps this phenomenon is much more prevalent.

As I have mentioned, M. Braverman and I have studied the question of computability of the Julia sets J_c of complex quadratic polynomials $P_c(z) = z^2 + c$ and demonstrated [**6, 8, 9**] that there exist values of $c \in \mathbb{C}$ such that J_c is not computable with an oracle for c (and, in fact, such c's can be produced constructively

[**7**], negating the need for an oracle). Values of c for which J_c is not computable form an uncountable set, which is, however, quite small in the measure-theoretic sense. It not only has zero area, but, subject to a natural conjecture, also has zero Hausdorff dimension. It is plausible that for a typical value of c, the set J_c is computable in polynomial time, and this is indeed true for a typical *real* value of c as shown by A. Dudko and I in [**10**].

The situation with Julia sets in two or more complex dimensions is entirely unknown, and any results, either positive or negative, would be most interesting.

7. What does Computer Modeling tell us in Practice?

I am far from being the first person asking this question, and there are numerous papers on this; as a sample of references spanning three decades see [**15, 14, 17**]. Of course, the only theoretical obstacle to rigorously approximating a computable object such as a physical measure or an attractor of a dynamical system with whatever precision is desired are the required computational resources. In the case when there is an efficient (polynomial-time) algorithm, the computation can be carried out with an arbitrary degree of accuracy – but it has to be done carefully, to control the round-off error. There are widely available computing packages which realize computations with an arbitrary precision arithmetic, which automate the process. However, this approach is computationally costly, as I could see first hand from my own very limited experience with this type of computations [**11**]. Arbitrary precision computations are immensely valuable in the situations when a single high-precision estimate needs to be obtained. They are the foundation of computer-assisted proofs in Analysis, since the pioneering work of Lanford [**13**].

Still, in most situations, numerical computations are carried out with a high, but finite precision, usually based on the rounding off conventions of the programming language used – something along the lines of double precision "reals" in *C*. Clearly, we cannot hope to faithfully recreate the properties of the orbits of a dynamical system with this approach. Suppose, we are given a dynamical system F acting on a bounded domain $\Omega \subset \mathbb{R}^m$. A finite-precision discretization of F replaces it with a map G whose domain and range consist of points whose coordinates are dyadic rationals of the form $p \cdot 2^{-n}$ where both n and p are bounded integers. This means that G acts on a finite set, all of its orbits are finite, and so eventually fall into periodic cycles. This is an artefact of the discretization, and, if the latter is not well-chosen, may result in an unfortunate scenario when all of the orbits seem to quickly collapse into very few cycles (see the insightful discussion in [**14**] and references therein). To provide a mathematical justification, iteration of a rounded-off map is usually interpreted as *random iteration.* That is, at the k-th step of the iteration, F is replaced with $F + \epsilon_k$, where ϵ_k is randomly chosen in a ball of radius $\delta > 0$ around the origin. This approach has been studied since the early days of the subject; in [**14**], Lanford credits D. Ruelle with suggesting it to explain the empirical behaviour of discretized chaotic maps.

Random iteration was studied from the point of view of algorithmic computability by the authors of [**5**], who found that, if ϵ_k is chosen uniformly and F is given by an oracle, then for almost all δ all ergodic measures of the random dynamical system are algorithmically computable. Thus, random noise destroys non-computability.

Rounding off, at least if interpreted in this way, masks the non-computable phenomena.

Another approach is to study the computability of the products $\{(\nu, \mathcal{O}_\nu)| \; ||\nu|| < \delta\}$, where ν is the parameter of the dynamical system F_ν, and the object $\mathcal{O}_\nu$ is, for instance, an attractor. This seems to mask non-computability in a similar fashion; for instance, M. Braverman and I showed that the set $\{(c, J_c)\}$ is computable[7] [**8**].

Rather than interpreting such statements as positive news, I see them as yet another reason for caution – since they suggest that the results produced by finite-precision iteration may be much nicer than the underlying dynamics.

8. A few Words in Conclusion

The field of computational complexity of continuous processes is only beginning to take shape now. Practitioners should be aware of the potential theoretical obstacles to numerical modeling of dynamical systems. There are two plausible explanations for the seeming effectiveness of numerical methods: the possibility that a *typical* dynamical system is computably tractable (and thus, all is good with the modeling world), or the less welcome scenario in which our approach to numerical modeling replaces the underlying dynamical systems with ones that are computably tractable and may thus blind us to the reality in some instances. Distinguishing between them motivates most of the open problems I have discussed in this note. Without a doubt, many fascinating discoveries await in the emerging new synthesis of analysis and theoretical computer science.

References

[1] S. Banach and S. Mazur. Sur les fonctions caluclables. *Ann. Polon. Math.*, 16, 1937.

[2] I. Binder, M. Braverman, C. Rojas, and M. Yampolsky. Computability of Brolin-Lyubich measure. *Commun. Math. Phys.*, 308:743–771, 2011.

[3] A. Bonifant and J. Milnor. Schwarzian derivatives and cylinder maps. In *Holomorphic dynamics and renormalization*, volume 53 of *Fields Inst. Commun.*, pages 1–21. Amer. Math. Soc., Providence, RI, 2008.

[4] M. Braverman and S. Cook. Computing over the reals: Foundations for scientific computing. *Notices of the American Mathematical Society*, 53(3):318–329, 2006.

[5] M. Braverman, A. Grigo, and C. Rojas. Noise vs computational intractability in dynamics. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 128–141. ACM, 2012.

[6] M. Braverman and M. Yampolsky. Non-computable Julia sets. *Journ. Amer. Math. Soc.*, 19(3):551–578, 2006.

[7] M. Braverman and M. Yampolsky. Constructing non-computable Julia sets. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 709–716. Association for Computing Machinery, 2007.

[8] M. Braverman and M. Yampolsky. *Computability of Julia sets*, volume 23 of *Algorithms and Computation in Mathematics*. Springer, 2008.

[7] This question was posed to us by M. Shub.

[9] M. Braverman and M. Yampolsky. Computability of Julia sets. *Moscow Math. Journ.*, 8:185–231, 2008.

[10] A. Dudko and M. Yampolsky. Almost every real quadratic polynomial has a poly-time computable Julia set. *Found. Comput. Math.*, 18(5):1233 1243, 2018.

[11] D. Gaidashev and M. Yampolsky. Cylinder renormalization of Siegel disks. *Experiment. Math.*, 16(2):215–226, 2007.

[12] I. Kan. Open sets of diffeomorphisms having two attractors, each with an everywhere dense basin. *Bulletin of the American Mathematical Society*, 31:68–74, 1994.

[13] O.E. Lanford. A computer-assisted proof of the Feigenbaum conjectures. *Bull. Amer. Math. Soc. (N.S.)*, 6(3):427–434, 1982.

[14] O.E. Lanford. Informal remarks on the orbit structure of discrete approximations to chaotic maps. *Experimental Mathematics*, 7(4):317 – 324, 1998.

[15] Y. Lévy. Some remarks about computer studies of dynamical systems. *Physics Letters A*, 88:1–3, 1982.

[16] E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20:130–141, 1963.

[17] R. Lozi. *Can we trust in numerical computations of chaotic solutions of dynamical systems ?*, pages 63–98. 2013.

[18] A.M. Lyapunov. *The general problem of the stability of motion.* Kharkov Mathematical Society, 1892.

[19] M. Lyubich. Combinatorics, geometry and attractors of quasi-quadratic maps. *Annals of Mathematics*, 140(2):347–404, 1994.

[20] M. Lyubich. Almost every real quadratic map is either regular or stochastic. *Annals of Mathematics (2)*, 156(1):1–78, 2002.

[21] S. Mazur. *Computable Analysis*, volume 33. Rosprawy Matematyczne, Warsaw, 1963.

[22] N. Metropolis. The beginning of the Monte Carlo method. *Los Alamos Science Special Issue*, pages 125–130, 1987.

[23] N. Metropolis and Ulam. S. The Monte Carlo method. *Journal of the American Statistical Association*, 44:335–341, 1949.

[24] J. Milnor. On the concept of attractor. *Commun. Math. Phys*, 99:177–195, 1985.

[25] M. Misiurewicz. Sensitive dependence on parameters. *Journal of Mathematical Physics*, 51(10):102704, 2010.

[26] S. Newhouse. Diffeomorphisms with infinitely many sinks. *Topology*, 13:9–18, 1974.

[27] J. Palis. A global view of dynamics and a conjecture on the denseness of finitude of attractors. In M. Flexor, P. Sentenac, and J.-C. Yoccoz, editors, *Géométrie complexe et systèmes dynamiques - Colloque en l'honneur d'Adrien Douady Orsay, 1995*, number 261 in Astérisque, pages 339 – 351. Société mathématique de France, 2000.

[28] H. Poincaré. *Les méthodes nouvelles de la mécanique céleste. Tomes 1-3.* Gauthier-Villars, Paris, 1892-1899.

[29] C. Rojas and M. Yampolsky. Computational intractability of attractors in the real quadratic family. *Advances in Math.*, 349:941–958, 2019.

[30] C. Rojas and M. Yampolsky. How to lose at Monte Carlo: A simple dynamical system whose typical statistical behavior is non-computable. In *Proceedings of*

the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, page 1066 1072, 2020.

[31] D. Ruelle. Historical behaviour in smooth dynamical systems. In *Global analysis of dynamical systems. Festschrift dedicated to Floris Takens for his 60-th birthday*, pages 63–65. IOP Publishing, 2001.

[32] W. Tucker. A rigorous ODE solver and Smale's 14th problem. *Found. Comp. Math.*, 2:53–117, 2002.

[33] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings, London Mathematical Society*, pages 230–265, 1936.

[34] S. Ulam, R.D. Richtmyer, and J. von Neumann. Statistical methods in neutron diffusion. *Los Alamos Scientific Laboratory report LAMS 551*, 1947.

[35] D. Wall. *Normal numbers.* PhD thesis, University of California, Berkeley, 1949.

Mathematics Department
University of Toronto
40 St George Street
Toronto
Ontario, M5S 2E4
Canada.
yampol@math.toronto.edu