

Computer arithmetic and sensitivity of natural measure

TIM SAUER*

Department of Mathematical Sciences, George Mason University, Fairfax, VA 22030, USA

(Received 5 July 2004; in final form)

In computer simulations of deterministic dynamical systems, floating-point rounding errors and other truncation errors contaminate the simulation results. We investigate the effect of computations using IEEE standard double precision arithmetic on inference of natural measure of chaotic attractors.

Keywords: Natural measure; Chaotic attractor; Floating-point computation; Trajectories

AMS Classification: 39A11; 37M25

1. Introduction

For difference equations with bounded, non-periodic solutions, the natural measure of the equation is a useful descriptor of the geometry of the solution. When it exists, it catalogs the density of typical bounded solutions of the equation, and can be used to characterize its dynamics.

By their nature, non-periodic solutions evade easy description. Computer methods are usually needed. In case of chaotic solutions, it is futile to try to approximate a particular solution for a large number of steps using floating-point computation, due to sensitive dependence on initial conditions. However, even if individual trajectories are unstable, a compact attractor defined as the totality of a number of solutions may be stable. Given that it is mathematically stable, we ask whether it is computable.

In this article we discuss the practical and theoretical obstructions to approximating the natural measure on a computer by creating long trajectories. In particular, we want to know whether the small errors made in computer arithmetic lead to correspondingly small errors in natural measure, or whether they can lead to disproportionately poor estimates.

To some extent this is a question about the sensitivity of natural measure to small perturbations, and more precisely the sensitivity to rounding errors made in machine computation. For example, it is clear that near global bifurcation points, natural measure will be sensitive to small changes. We begin by investigating examples of this type, and proceed to an example where there is no nearby bifurcation. In both cases, we find examples of extreme failure of computer arithmetic to approximate natural measure, in that the errors are many orders of magnitude greater than the arithmetic rounding errors.

*Email: tsauer@gmu.edu

Let S be a Borel set, f a discrete map on R^n . The natural measure μ is a probability measure satisfying,

$$\mu(S) = \text{asymptotic proportion of trajectory in } S$$

for almost every initial condition. The natural measure may not exist, but if it does it is f -invariant, i.e.

$$\mu(f^{-1}(S)) = \mu(S).$$

The properties of natural measures are a key topic of ergodic theory. The Frobenius-Perron operator (see [7] for an introduction) can be used to determine the measure in many cases. In this article we consider only the most elementary approach of computing long trajectories of typical initial conditions.

Example 1 Bernoulli map $x_{n+1} = f(x_n) = 2x_n \pmod{1}$.

The natural measure is the constant density $\rho(x) = 1$ on $[0,1]$. Almost every initial condition in the sense of Lebesgue measure on $[0,1]$ will map out the uniform probability density $\rho = 1$. The measure zero set of exceptions includes $x_0 = 0$, which generates the atomic measure concentrated at 0 instead of the natural measure. Also in this category are all initial conditions that are expressible as a finite binary expansion.

Example 2 Logistic map $x_{n+1} = f(x_n) = 4x_n(1 - x_n)$.

The map f is plotted in figure 1(a). It can be shown (for example, see [1]) that the natural measure has density $\rho(x) = 1/\pi\sqrt{x(1-x)}$. For a subinterval $[a, b]$ of the unit interval the measure is

$$\mu([a, b]) = \int_a^b \frac{dx}{\pi\sqrt{x(1-x)}} = \frac{2}{\pi} \arcsin\sqrt{b} - \frac{2}{\pi} \arcsin\sqrt{a}. \tag{1}$$

Almost every initial condition traces out the measure (1), whose density is shown in figure 2. Exceptional initial conditions include $x_0 = 0, 1/2,$ and 1 , which generate the atomic measure at 0.

Although the natural measure for these two examples is well known, it is rare to have explicit expressions in terms of elementary functions. More typically, computer simulation is needed. The most obvious way to compute a natural measure is to calculate a long trajectory and use it to produce a histogram showing the density of points in each area. Since the natural measure is

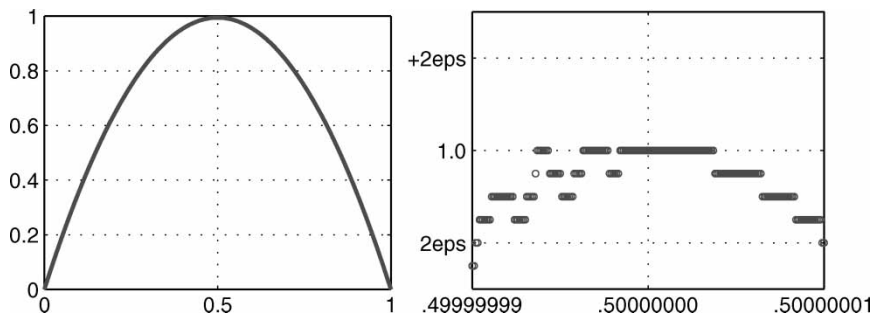


Figure 1. (a) The logistic equation $x_{n+1} = f(x_n) = 4x_n(1 - x_n)$. (b) Magnification of the double precision version \bar{f} of (a). eps represents $2^{-52} \approx 2.2 \times 10^{-16}$.

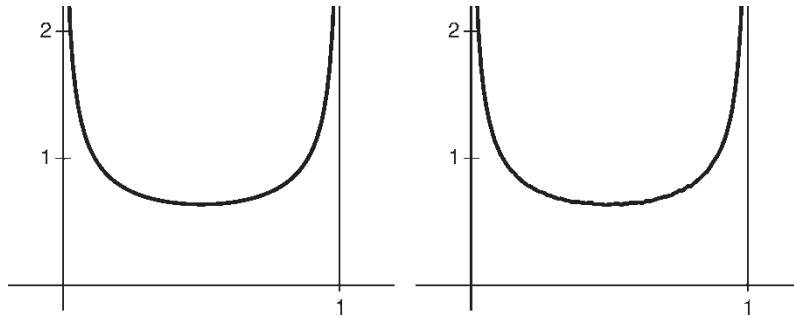


Figure 2. (a) Theoretical natural density of logistic equation. (b) Histogram of period 56,38,349 orbit. Both panels are plotted with step size 0.01. Differences are barely perceptible at the scale of this figure. Under double precision calculation, the trajectory of most initial conditions will yield (b) as the computed natural measure.

an asymptotic invariant, only the tail of the trajectory is relevant, and often reasonable approximations can be made by computing a trajectory and throwing away the initial transient.

However, using floating-point computation to approximate natural measure immediately raises a dilemma. Computer arithmetic systems have a finite number of states. (There are about 10^{16} double precision numbers between 0 and 1.) Due to finiteness and the determinism of computer arithmetic, every computer trajectory is periodic, unlike true trajectories. Even so, if a reasonable approximation to natural measure can be made with a long-period computer trajectory, we may be satisfied for practical purposes.

The main question of this article is to determine how reliably the correct natural measure can be approximated in double precision computer arithmetic. We will find it is possible, in an example like the logistic equation, to reach complete failure in this approximation. Because the IEEE-754 double precision arithmetic protocol [5] has become the *de facto* standard for scientific computing, it is enlightening to investigate as a case study what happens when the logistic equation is run for long trajectories, in hopes of approximating the natural measure. As we explain below, (i) over 2/3 of initial conditions in $[0,1]$ eventually iterate under the IEEE-754 version of the logistic equation to a single periodic solution, of period 56 38 349 and (ii) another 16% of initial conditions eventually iterate to the zero fixed point solution. The former initial conditions lead to a trajectory that approximates the natural measure (1) rather well. The latter initial conditions fail catastrophically.

The logistic map is special in that it sits at a bifurcation value of its parameter. We investigate its double precision characteristics because of its ubiquity in discussion of one-dimensional maps, but it is not typical for the purpose of natural measure since the attractor touches the basin boundary, making it very sensitive to perturbations. After studying the logistic example we show an additional second-order equation that fails to reproduce a good double precision approximation in a more robust way.

2. Computing measure in double precision arithmetic

After several years of conflicting standards, scientific computation has largely settled to a single computer arithmetic system, IEEE-754 double precision. Most floating-point compilers and computer software systems built on them (for example, Matlab) will give precisely reproducible results under this standard.

In this arithmetic system, each double precision machine number is represented as

$$\pm 1.bbb\dots b \times 2^e \quad (2)$$

where the leftmost 1 is assumed, and where there is a 52 binary digit mantissa to the right of the radix. A total of 11 bits are assigned to store the integer exponent e , and 1 bit for the sign. The *unit roundoff*, or *machine epsilon* of the number system, is the distance between 1 and the next largest double precision number, which is 2^{-52} . The default rounding protocol for IEEE-754 double precision is called “rounding to nearest”, and does exactly that after each arithmetic operation: rounds to the nearest 52-bit double precision number. There is a deterministic procedure for rounding in case of a tie. We will often denote the double precision floating-point version of a real number x by \bar{x} , and informally call it a machine number.

Since the two examples discussed above have known natural measures, it will be helpful to investigate how successful computer arithmetic versions of the difference equations are in reproducing the correct measure. Example 1 is extremely special and not very representative because it is essentially multiplication by 2, the base of the number system being used for computation.

Example 1 Bernoulli map $x_{n+1} = f(x_n) = 2x_n \pmod{1}$.

Let \bar{f} denote the machine version of the map. Multiplication by 2 just changes the integer exponent, and has no effect on the mantissa. The modulo operation, on the other hand, effectively shifts the mantissa to force the exponent to be 0, and chops off everything to the left of the radix. Then the number must be rejustified, changing the exponent to put the leftmost 1 on the left of the radix. For any machine number greater than machine epsilon, at least one significant digit will be lost on each iteration, so it is clear that after 52 iterations of \bar{f} , the computer trajectory will be moved to machine 0, where it will stay on further iterations. The computed natural measure, on the basis of this trajectory, will converge to the atomic measure at 0. This disagrees with the correct natural measure.

This is an example of an extremely unsuccessful measure calculation. It occurs due to the special connection of the map with regard to the binary number system on which computer arithmetic is based.

Example 2 Logistic map $x_{n+1} = f(x_n) = 4x_n(1 - x_n)$.

This example has less resonance with the number system, and is fairly typical for a simple dynamical system in terms of the arithmetic. However, it is special dynamically, in that it lies at a crisis, or bifurcation value of the map parameter $\lambda = 4$. For this parameter, small perturbation noise added to the map near $x = 1/2$ will move a trajectory out of the unit interval, and towards negative infinity.

The logistic example shows the sometimes counter-intuitive effects of machine rounding as compared with random noise. Although we will see that the natural measure will be incorrectly computed by many long trajectories, it happens in a somewhat surprising way. For example, machine rounding error cannot move a trajectory outside of the interval $[0,1]$, as the first of the following facts shows. Let \bar{f} denote the machine version of the logistic map.

Fact 1. Let \bar{x} be a machine number in $[0,1]$. Then $0 \leq \bar{f}(\bar{x}) \leq 1$.

Proof First note that since all operands are nonnegative, the result of computing the equation is nonnegative. To show the upper bound, define $\bar{y} = \bar{x} - 1/2$. Because double precision

numbers are expressed in base 2,

$$\begin{aligned} 4\bar{x}(1 - \bar{x}) &= 4\left(\frac{1}{2} + \bar{y}\right)\left(\frac{1}{2} - \bar{y}\right) = (1 + 2\bar{y})(1 - 2\bar{y}) \\ &= 1 + 2\bar{y} - 2\bar{y} - (2\bar{y})^2 = 1 - (2\bar{y})^2 \leq 1. \end{aligned}$$

Curiously, the analogue of Fact 1 fails for other maps conjugate to the standard logistic map. For example, consider $g(x) = 2.5x(8/5 - x)$ on the interval $[0, 8/5]$. One checks that g is conjugate to the above logistic map f by the change of coordinates $x \rightarrow 8x/5$. The corresponding fact for this map, that $0 \leq \bar{x} \leq 8/5$ implies that $0 \leq \bar{g}(\bar{x}) \leq 8/5$, is false. The reason is that $2.5[(0.8 + \bar{y})(0.8 - \bar{y})] \leq 1.6$ does not hold for all double precision numbers, including for example $\bar{y} = 0$! (This can be easily verified by computing $2.5(0.8 \times 0.8) - 1.6$ in Matlab, or in a C program using the “double” data type.) Therefore, any trajectory whose initial condition is sufficiently close to 0.8 will diverge to negative infinity. This positive measure set of initial conditions clearly fails to approximate the true natural measure. \square

Although the logistic map appears to be “saved” by the luck of computer arithmetic, in fact there are other problems.

Fact 2. Approximately 16% of machine numbers \bar{x} in $[0, 1]$ eventually map to the trajectory 0 under \bar{f} .

Figure 1(b) shows the difficulty—there is a small but positive length neighborhood around $\bar{x} = 1/2$ where \bar{f} maps to exactly 1, and following that maps to exactly 0 for all further steps. For example, the floating-point version \bar{x} of $x = 2/13$ satisfies

$$\bar{f}^{171537}(\bar{x}) \approx \frac{1}{2} + 3 \times 10^{-9},$$

and so according to figure 1(b), $\bar{f}^k(\bar{x}) = 0$ for $k \geq 171539$. Since only a Lebesgue measure 0 set of initial conditions has the atomic measure at 0 in theory, this shows a breakdown of the ability of the simulation to compute natural measure.

On viewing the figure 1(b), one might wonder why Fact 2 doesn’t state 100% instead of 16%. The true natural measure should bounce the state around until it finds the small subinterval $[1/2 - 3 \times 10^{-9}, 1/2 + 3 \times 10^{-9}]$, and then exit to 0.

Moreover, we can make a prediction for how long this should take. According to the known natural measure (1) of the logistic map, the probability of landing in the subinterval on a typical iteration should be

$$p = C(1/2 + 3 \times 10^{-9}) - C(1/2 - 3 \times 10^{-9}) \approx 3.82 \times 10^{-9},$$

where $C(x) = 2/\pi \arcsin \sqrt{x}$ is the antiderivative of the natural density. After 10^8 iterations, say, the probability that a trajectory maps to 0 would be $1 - (1 - p)^{10^8} \approx 0.32$, and the rest of the initial conditions would just take a little longer. However, something else happens first, keeping the trajectory from moving according to the true natural measure:

Fact 3. The trajectories of over two-thirds of the initial conditions in $[0, 1]$ iterate to a single periodic orbit of \bar{f} .

In addition to the fixed orbit at 0, there is a period 56 38 349 orbit that claims over two-thirds of all initial conditions. This orbit does not enter the interval around 1/2 shown in

figure 1(b). A histogram of the orbit in the interval $[0,1]$ is shown in figure 2(b). It is very close but not equal to the correct histogram implied by (1). The two-thirds of initial conditions that iterate to this orbit will, after an initial transient, generate the histogram figure 2(b) instead of the correct histogram. Although incorrect, it is an improvement on the atomic measure at 0. There is another orbit of period 1 46 32 801 that traps over 10%. More than 95% of initial conditions terminate in one of these three periodic orbits.

Since the double precision numbers form a finite set, every computer trajectory will be eventually periodic. However, given the fact that there are $2^{52} \approx 10^{16}$ distinct machine numbers between 0 and 1, it is rather surprising that the periods are so low. Empirical evidence supports the following:

CONJECTURE For all \bar{x} in $[0,1]$, $\tilde{f}^{200000000}(\bar{x})$ belongs to a periodic orbit of the IEEE double precision logistic map \tilde{f} .

This is consistent with the work of Grebogi, Ott, Yorke [4] and Beck and Roepstorff [3], which predicts that the expected length of periodic computer orbits for a one-dimensional attractor should be on the order of the reciprocal square root of machine epsilon, or 10^8 .

The details of the double precision version of the logistic equation are not important. They will of course differ in any other system of computer arithmetic, for example the extended precision system with 80-bit mantissa that may become a future standard. The purpose of the example is to show the general outline of what to expect for a well-known example on a typical floating-point arithmetic system.

The logistic equation is still not very typical because it lacks robustness. Since the attracting set, the interval $[0,1]$, intersects its basin boundary at 0 and 1, it is not surprising that any small error will cause problems in computing the natural measure. Next we illustrate a more subtle failure. Even when attractors are well within the basin, and are near no bifurcations, it may be impossible for long computer trajectories to produce faithful approximations of the natural measure.

3. A robust, sensitive second-order equation

A third example is a second-order equation, apparently far from bifurcations, whose double precision trajectories give natural measure approximations that are far from the correct measure.

Example 3 Define the second-order equation

$$x_{n+1} = a \sin \pi x_n + \sin \left[\left(\frac{x_n}{a} - \sin \pi x_{n-1} \right) b \cos \pi x_n \right] \quad (3)$$

where a, b are constants. This equation is a variant of an example in [8], and is a second-order extension of the first-order equation

$$x_{n+1} = a \sin \pi x_n, \quad (4)$$

which for $a > 1$ is globally convergent to the chaotic attractor $[-a, a]$.

Figure 3(a) shows a 2000-point trajectory in the (x_n, x_{n+1}) plane computed with no error, setting $a = 1.1, b = 1.82$. Figure 3(b) shows the same computation of (3) using IEEE double precision. These are both representative trajectories; changing initial conditions or running

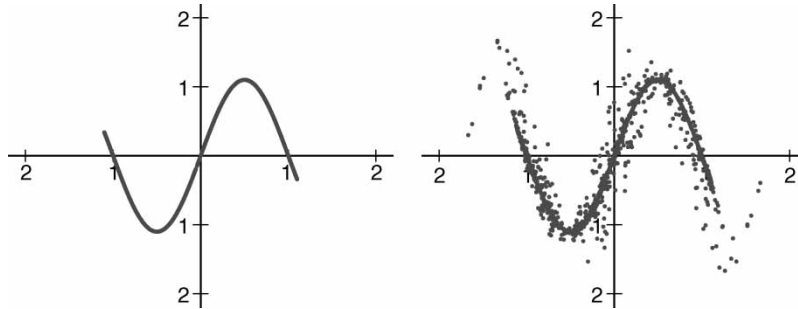


Figure 3. (a) Theoretical natural density of second-order equation (3). (b) Computed natural density from typical trajectory. Each panel shows a single 2000-point trajectory, after deleting the transient.

longer trajectories would show the same general picture. The correct natural measure, shown in panel (a), is supported on the curve (4), while the one in panel (b), computed in double precision, is clearly incorrect. The spray of points off the theoretical attractor is macroscopic phenomena brought up through 16 orders of magnitude from microscopic causes. This example is of course specially constructed to emphasize this problem, but is not otherwise dynamically unusual, and the behavior persists through changes in both parameters a and b . We emphasize that the extra points in figure 3(b) are persistent, not transient phenomena. This is an extreme case of a phenomena called bubbling [2].

To understand the intuition behind the construction of Example 3, note that for points near the curve (4) the approximation

$$x_{n+1} - a \sin \pi x_n = \sin \left[\left(\frac{x_n}{a} - \sin \pi x_{n-1} \right) b \cos \pi x_n \right] \approx \left(\frac{x_n}{a} - \sin \pi x_{n-1} \right) b \cos \pi x_n$$

can be made. The parameter b is adjusted so that for points on the curve, the multiplicative mean of $b \cos \pi x_n$ is less than a . For the theoretical attractor, the distance of points from the curve decreases exponentially to zero. Since the natural measure is an asymptotic limit, it is supported entirely on the curve. For the double precision arithmetic attractor, on the other hand, small rounding errors of size $\approx 2^{-52}$ in the distance are multiplied by sequences of $b \cos \pi x_n$ that despite having multiplicative mean being less than a , occasionally have subsequences that multiply the distance to a macroscopic size, as seen in figure 3(b).

Since solutions to this equation cannot be computed even to a single correct significant digit by double precision arithmetic, the question remains how figure 3(a) was made. It turns out that the mathematically equivalent version

$$x_{n+1} = a \sin \pi x_n + \sin \left[\left(x_n - a \sin \pi x_{n-1} \right) \frac{b}{a} \cos \pi x_n \right] \tag{5}$$

of (3) eliminates the spray of incorrect points through fortuitous rounding error cancellation. Independent of the present discussion, it is interesting that the two versions (3) and (5) of a stable attractor, which are shown to be identical by elementary algebra, disagree macroscopically due to their different handling of size 10^{-16} errors.

Examples like the ones treated in this article throw into question conclusions made from computer simulation of long trajectories of dynamical systems and difference equations. When chaos is involved, faithfulness of computer trajectories on an individual basis is not expected, due to sensitive dependence on initial conditions. However, stability of the attractor

itself, as catalogued by natural measure, may still be hoped for. Example 3, in particular, shows an error magnification of 16 orders of magnitude on an ongoing basis, that cannot be attributed to any simple bifurcation or basin boundary crossing. For this example, double precision computer trajectories are not representative of the correct natural measure. It is not known whether alternate computational methods of inferring natural measure, such as those based on the Frobenius-Perron operator [6], suffer from the same difficulties.

References

- [1] Alligood, K., Sauer, T. and Yorke, J.A., 1996, *Chaos: An Introduction to Dynamical Systems* (NY: Springer-Verlag).
- [2] Ashwin, P., Buescu, J. and Stewart, I., 1994, Bubbling of attractors and synchronization of chaotic oscillators. *Physics Letters A*, 126–139.
- [3] Beck, C. and Roepstorff, G., 1987, Effects of phase space discretization on the long-time behaviour of dynamical systems. *Physica D*, 173.
- [4] Grebogi, C., Ott, E. and Yorke, J.A., 1988, Roundoff-induced periodicity and the correlation dimension of chaotic attractors. *Physical Review A*, 3688–3693.
- [5] IEEE Computer Society, 1985, IEEE standard for binary floating-point arithmetic, ANSI/IEEE Std., pp. 754–1985.
- [6] Li, T.Y., 1976, Finite approximation for the Frobenius-Perron operator, a solution to Ulam’s conjecture. *Journal of Approximation Theory*, 177–186.
- [7] Pollicott, M. and Yuri, M., 1998, *Dynamical Systems and Ergodic Theory* (Cambridge, UK: Cambridge University Press).
- [8] Sauer, T., 2003, Chaotic itinerancy based on attractors of one-dimensional maps. *Chaos*, 947–952.