Shadowing by High-Precision Optimization

Henrik Bernt Håkonsen, Bergen, Norway; h_haakonsen@hotmail.com
Robert Knapp, Wolfram Research., Inc., Champaign, Illinois 61820; rknapp@wolfram.com
Stan Wagon, Macalester College, St. Paul, Minnesota 55104; wagon@macalester.edu (corresponding author)
July 10, 2009

Abstract. We use high-precision optimization to obtain explicit shadows for machine-precision trajectories of several sensitive dynamical systems: the quadratic map $r\,x(1-x)$, a billiard problem from the SIAM 100-Digit Challenge, the Duffing differential equation and the Ikeda delay differential equation. The main tools are the numerical optimization algorithms as implemented in *Mathematica*, good estimates of the working precision required to get reliable trajectories, and a way of estimating the sensitivity of the system so as to be able to feed a Jacobian to the optimization algorithm. In these examples one can get a sense of how the optimal shadowing error grows as the length of the noisy trajectory grows. Something quite special happens in the case of the quadratic map: because the system is bounded, it must eventually become periodic and by computing the periodic part explicitly we can apply ideas of Chow and Palmer to show that for any N-term trajectory there is a good shadow (error less than 10^{-9}), regardless of how large N is.

1. Introduction

An important question for sequences of numbers computed in some fixed precision is the relation of the computed sequence to a true sequence, computed in infinite precision with no roundoff error. This is what shadowing theory tries to address. One has a discrete or continuous dynamical system Y, such as iterates of a function or the solution to a differential equation computed using fixed precision (typically, IEEE 53-bit machine precision), and one wants to know if there is an initial condition X such that the infinitely precise computation starting from that initial condition will be very close to the computed data Y. If such an X exists, its trajectory is called a *shadow* of the noisy trajectory and X a *shadow seed*. The error between X and Y will refer to the maximum absolute value of the difference (the *sup error*) though we will also use the sum-of-squares error in the optimization work.

We use the term *noisy trajectory* to refer to machine-precision trajectories of either a discrete or continuous dynamical system. More generally, a *pseudo-orbit* is any sequence where terms differ by a small amount from the true value that the preceding term would generate. There is a well-developed theory of shadowing for hyperbolic dynamical systems, but here we are working with more general systems and taking a computational approach to finding shadows. The benefit is that we can in many cases generate the exact value of the shadow.

In §2 we show how optimization can find explicit shadows for the quadratic map, out to several thousand terms. In §3 we show how work of Chow and Palmer can be extended to reveal that a close shadow can exist for an infinitely long machine-precision trajectory of the quadratic map. Section 4 brings the optimization technique to bear on a certain billiard problem, and in §5 we show that the same optimization techniques can be used to find shadows for sensitive differential equations such as the Duffing equation or the Ikeda delay differential equation.

All computations were done on a MacBook Pro laptop with a 2.16 GHz Intel Core 2 Duo chip. Machine precision computations on other platforms will vary in detail. All the optimizations were done using *Mathematica*'s `FindMinimum` algorithm, which allows options for specifying either the classic Gauss–Newton method or the Levenberg–Marquardt method, and the Jacobian in either case.

Acknowledgement. We're grateful to Wayne Hayes and Tim Sauer for helpful conversations about shadowing.

2. The quadratic map

Let $f_r(x) = r\,x(1-x)$; it is well-known that f_r is chaotic for many values of r, including $r = 4$. Because shadowing for $r = 3.8$ has been studied in detail previously [HYG, CP] and also because $r = 4$ is very special in that f_4 is conjugate to the tent map and thus admits alternative methods for finding shadows, we will focus on the 3.8 case. It is not known that the true orbit for $r = 38/10$ is chaotic, though the numerical evidence is strong that it exhibits sensitive dependence. Recall that for f_4 the sensitivity is very well understood: the Lyapunov exponent λ is $\ln 2$ and the precision loss is one bit per iteration (and therefore all precision is lost when using machine precision—roughly 16 decimal digits—and iterating 55 times). For 38/10 the situation appears to be similiar but with slower divergence: λ is about 0.484 (see [W, chap. 7]).

Consider a finite machine-precision trajectory $Y = \{y_0, y_1, \ldots\}$ for $f_{3.8}$ starting at $y_0 = 0.1$. Let Y_m denote the initial segment up to the mth term. Then Y_{85} already ends in a value that has diverged completely from the true trajectory of $f_{38/10}$ starting at $1/10$. But Y is a pseudo-orbit of $f_{3.8}$ in the sense that each term $y_m \in Y$ is quite close to (within $3 \cdot 10^{-16}$ of) the true value $f_{3.8}(y_{m-1})$ computed without roundoff (this bound is proved in [CP, p. 370]).

We seek a shadow for Y, meaning a value x_0 (which we call the *shadow seed* and denote by $seed_m$ when shadowing Y_m), given to very high precision, so that the trajectory of x_0, computed in high-precision, is very close to Y. The importance of this is that it shows that Y is close to a true trajectory of a nearby initial value. The idea is simple: set up an objective function that measures the difference between a high-precision trajectory and Y and then minimize the objective. The main complication is that the objective function can be excessively complicated and fractal-like, while optimization methods typically work under the assumption that a quadratic model successfully describes the objective function. This difficulty is overcome by setting things up so that a quadratic model does apply, and that is done by shadowing only a small number of terms, say 50, in one step, and then using $seed_{50}$ as the starting value for the next step of 50 additional terms. One can work up in this way to find explicit shadow seeds even for several thousand terms, the limiting factor being the slowness of high-precision work.

A key observation is that if Y starts at 0.1, then 1/10 is an obvious approximation to the shadow seed. But the trajectory of 1/10 diverges from Y after about 85 terms. So it is reasonable to use 1/10 as a starting value in an optimization for Y_{50}. To understand the nature of the optimization problem it is helpful to understand the objective function. Figure 1 shows how the error changes with the scale; it appears that standard optimization techniques will have a chance of finding the apparent minimum, which occurs roughly 10^{-17} to the right of 1/10. While we cannot prove that the global minimum apparent in the figures is truly the global minimum, the numerical evidence is strong that the minimum shown, as well as all the others computed in this paper, is indeed the global minimum, and so the shadows we find are the best shadows. Once we have $seed_{50}$, we can use that as the starting value in a search for Y_{100}.

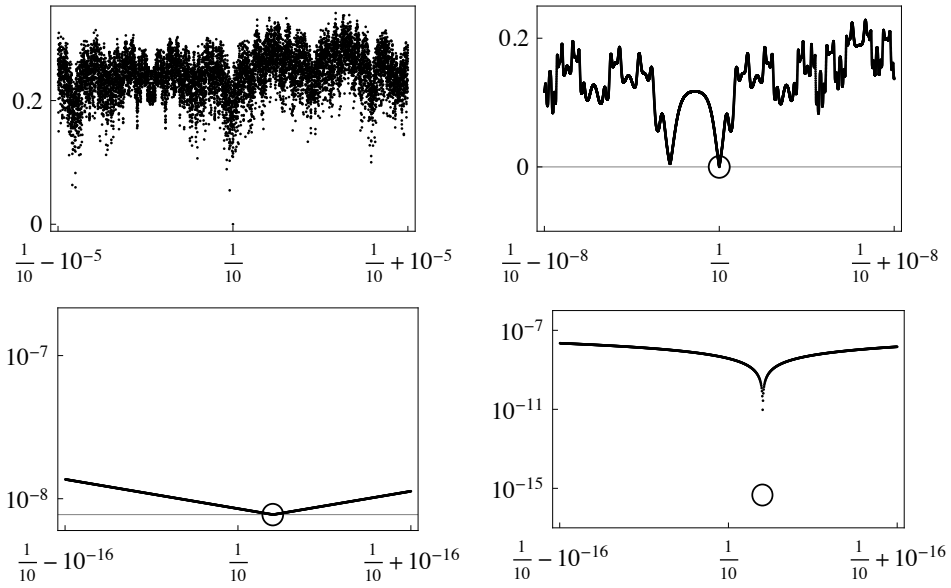


Figure 1. The shadowing error for the trajectory of 0.1 shown at different scales; the last plot uses a logarithmic scale. The circles mark the minima.

Similar plots for Y_{100} show that the error is not nearly so well behaved and that is why we succeed for Y_{100} only when we use $seed_{50}$ as a starting value. The error plot is then centered at that seed, and is again well behaved (Fig. 2).

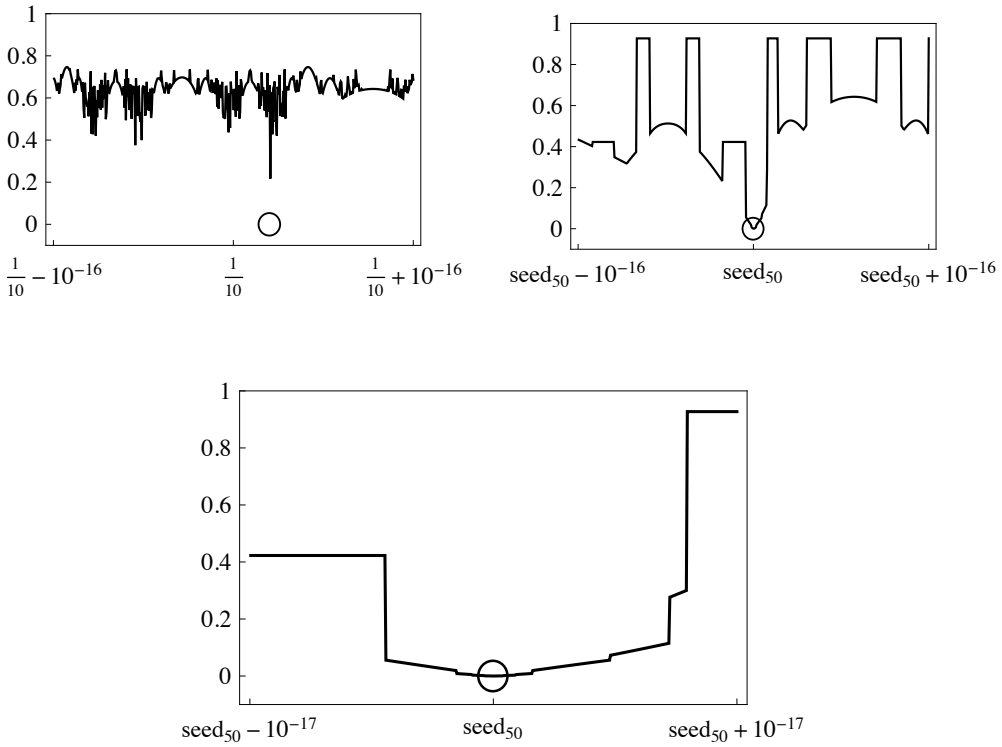


Figure 2. The shadowing error for Y_{100}, centered at $\frac{1}{10}$ (upper left) and then centered at $seed_{50}$.

Another key point is that optimization methods sometimes work better when some sensitivity information is provided. For f_r such information is quite easy to obtain, since the chain rule allows us to get the vector by just accumulating the product of the derivatives, $r\,(1 - 2\,y_n)$.

Shadowing Algorithm for N iterates of f_r starting from y_0

Step 1. Compute the noisy trajectory2 $\{y_n\}_{n=0}^{N}$: N iterations of f_r starting from y_0, using machine precison.

Step 2. Determine an appropriate level, p, of high precision for a shadow of N terms. A safe choice is $p = N$ digits, since the precision loss is well under one decimal digit per iteration. Compute y_n^* to be the best precision-p representation of y_n; in high precision computations y_n^* is used instead of y_n.

Step 3. Compute the sensitivity vector (the Jacobian) J by evaluating the derivative $r(1 - 2\,y_n^*)$ at each point and then accumulating the products: precisely, $J = \left\{\prod_0^k r\left(1 - 2\,y_k^*\right)\right\}_{k=0}^{N}$.

Step 4. Set up an objective function $E(x, M)$ that, given a high-precision value x_0, computes the high-precision M-iteration trajectory of x_0 and forms the sum-of-squares error of the residual vector $\{y_n^* - x_n\}_{n=0}^{M}$.

Step 5. Use high precision and the Gauss–Newton optimization technique to find x_0 that minimizes $E(x, 50)$, starting from y_0^*. The Gauss–Newton method uses a locally quadratic model to compute successive approximations to a minimum.

Step 6. Repeat steps 2–4 increasing M by 50 each time and starting the optimization from the preceding seed value.

EXAMPLE. Applying the algorithm to Y_{100} first gets $seed_{50}$ as $\frac{1}{10} + 2.02514302 \cdot 10^{-17}$ and the corresponding sup error as $4.589996 \cdot 10^{-16}$; at the second round (Y_{100}), we get $seed_{100} = \frac{1}{10} + 2.025143008078537154 26 \cdot 10^{-17}$, with shadow error rising to $7.69 \cdot 10^{-16}$. Extending this to $seed_{500}$ takes under two seconds and yields a shadow error of $1.20 \cdot 10^{-15}$; see Figure 3 which shows that $seed_{500}$ achieves visual perfection to more than 550 terms. These computations were on a Macintosh with an Intel chip; the noisy trajectory, and therefore the shadow seeds and errors, will vary with the platform.

The method specified above minimizes the sum of squares error, which is not the same as minimizing the sup-error. However, one can use the answer to the former in an optimization for the latter, from which one learns that the differences between the two are negligible (i.e., the error plots as in Figure 1 are qualitatively the same and the difference in location of the minima is very small).

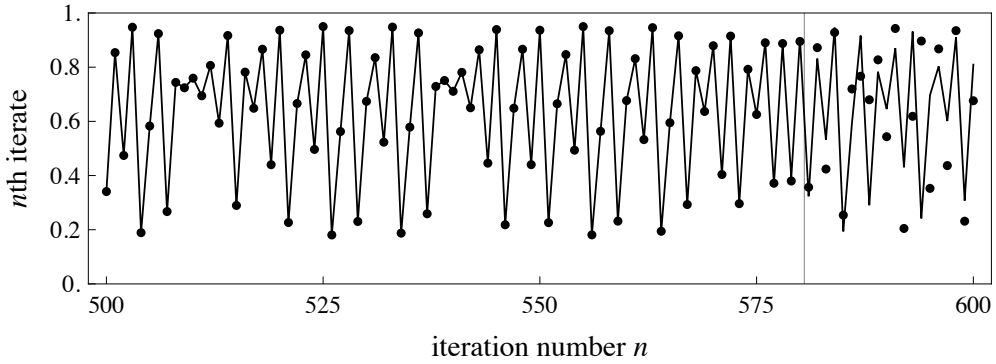


Figure 3. The tail end of the noisy trajectory Y_{600} is shown by dots; the trajectory of $seed_{500}$ is the broken line. There is visual agreement to 580 terms.

In light of the work of Hammel et al [HYG] and Chow and Palmer [CP] (see §3 below) it is not all surprising that good shadows exist for f_r, the reason being that the derivatives of f_r never get too small. The approach via optimization brings concreteness to the issue by showing how to find the explicit shadow values.

The approach given here can be used to obtain a shadow to 4000 terms in about 1.5 hours. The shadowing error for Y_{4000} is about $4.874985994773 \cdot 10^{-14}$.

3. Shadowing forever

The optimization method of §2 gets an explicit shadow for a few thousand terms. Hammel et al [HYG], working in 14-digit precision, showed that one can shadow for 10^7 terms with error under 10^{-8}. Chow and Palmer [CP] showed that for trajectories of functions mapping [0, 1] to itself one can bound the shadowing error using two easily computed functions, σ and τ. These functions measure expansiveness and, loosely speaking, σ can be viewed as the reciprocal of the derivative while τ estimates the amount of change needed to eliminate the noise.

DEFINITION. Given a noisy trajectory $Y = \{y_i\}_{i=0}^{N+1}$ for a twice continuously differentiable function $F : [0, 1] \to [0, 1]$, define σ, τ, M, and B by:

$$\sigma = \sup_{n=0}^{N} \Sigma_{m=n}^{N} \left| \frac{1}{F'(y_n)\,F'(y_{n+1})\cdots F'(y_m)} \right|$$

$$\tau = \sup_{n=0}^{N} \left| \Sigma_{m=n}^{N} \frac{1}{F'(y_n)\,F'(y_{n+1})\cdots F'(y_m)} \,(y_{m+1} - F(y_m)) \right|$$

$$M = \sup(\,|F''(x)| : 0 \le x \le 1) \qquad\qquad B = \frac{2\tau}{1+\sqrt{1-2M\sigma\tau}}$$

Following [CP], if we restrict the sum in the definition of τ to just the first p terms (with m still bounded by N) as opposed to $N - n + 1$ terms, we get τ_p, which can be an excellent approximation to τ for p not too large; this is how τ is computed when N is large. To be precise, so long as

$$\mu_p = \sup_{n=0}^{N-p} \left| \frac{1}{F'(y_n)\,F'(y_{n+1})\cdots F'(y_{n+p})} \right| < 1,$$

σ_p and τ_p are reasonable approximations: $\sigma_p \le \sigma \le (1-\mu_p)^{-1}\,\sigma_p$ and $\tau_p \le \tau \le (1-\mu_p)^{-1}\,\tau_p$. Another useful restriction comes from viewing the vector defining τ (i.e., eliminating the sup and the absolute values) as being a first estimate to the difference between Y and the shadow trajectory X (indeed, such ideas underlie the proof-by-iteration of the theorem that follows); then looking at only the first term of the vector gives an estimate for the difference between the noisy starting value and the shadow seed. We call this value δ; that is,

$$\delta = \Sigma_{m=0}^{N} \frac{y_{m+1} - F(y_m)}{F'(y_0)\,F'(y_1)\cdots F'(y_m)}$$

Further details are in [CP]. Note that σ can be computed using just machine precision but τ requires high precision because of the difference term. Further, the derivative of $f_r(x)$ is simply $2\,r\left(x - \frac{1}{2}\right)$, so that the main terms in σ and τ are the reciprocals of the distance of y_i from $1/2$, divided by $2\,r$. Here is the main result from [CP].

THEOREM. Let $Y = \{y_i\}_{i=0}^{N+1}$ be a noisy trajectory for a twice continuously differentiable function $F : [0, 1] \to [0, 1]$ and suppose $2\,M\,\sigma\,\tau \le 1$. Then there is an exact trajectory $X = \{x_i\}_{i=0}^{N}$ for F such that $\sup |\, X - Y_N \,| \le B$.

Here are some sample computations of these items; M is just $2\,r$ for the quadratic map.

$n = 50$: $\sigma = 15.313$, $\tau = \mathbf{4.5899}84\,773 \cdot 10^{-16}$, $B = 4.589984773 \cdot 10^{-16}$, $\delta = \mathbf{1.47003}149 \cdot 10^{-17}$
$n = 100$: $\sigma = 15.329$, $\tau = \mathbf{7.685263027}767 \cdot 10^{-16}$, $B = 7.685\,263\,027767 \cdot 10^{-16}$,
$\delta = \mathbf{1.470031495765}85 \cdot 10^{-17}$

Compare to the true story, as found in §2:

$n = 50$: shadow error = $\mathbf{4.5899}9589 \cdot 10^{-16}$, $\text{seed}_{50} - 0.1 = \mathbf{1.47003}151 \cdot 10^{-17}$
$n = 100$: shadow error = $\mathbf{7.6852630}279517 \cdot 10^{-16}$, $\text{seed}_{50} - 0.1 = \mathbf{1.470031495765}59 \cdot 10^{-17}$

For $N = 4000$ the shadow error from §2 is $\mathbf{4.8749859}948 \cdot 10^{-14}$ while the upper bound B is $\mathbf{4.8749859}960 \cdot 10^{-14}$.

So we see here several interesting aspects of these parameters for the case at hand:

• B is very close to τ (for longer trajectories it does become slightly different).

• B is an extrenely sharp upper bound on the actual shadowing error.

• $y_0 + \delta$ is a good estimate of the shadow seed.

Further computations using a variety of starting values, letting N be much larger, and also considering $r = 4$, confirm these observations. But note the important condition that, for any of this to be useful, we must have $4\,r\,\sigma\,\tau \le 1$. This can fail for starting values close to $1/2$ and then the definition of B fails; thus the points above are not universally valid.

These observations have several consequences:

1. In the optimization technique of §2, we can use $y_0 + \delta$ as the starting value, as opposed to just starting at a rational value near y_0 (such as 1/10 when $y_0 = 0.1$). This does speed up the first step of the optimization, so is worth doing. The overall impact is small since this first step plays a small role if one is shadowing hundreds of terms; yet this idea can help with the first step in some troublesome cases, such as when the initial value is very close to 0.5.

2. We can generate a plot of B as the trajectory grows and so gain an understanding of how the shadowing error increases for longer trajectories. Such estimates were done by Chow and Palmer [CP] who found that the shadow error for $Y_{426\,000}$ (starting from 0.6, using single-precision, and using τ_{30} to approximate τ) was bounded by $9.8 \cdot 10^{-6}$. A plot of shadow error vs. N was also presented by Chow and Van Vleck [CV; fig. 1] who comment on the sublinear nature of the error growth. In [HYG] it is stated that when N-digit precision is used it is to be expected that a trajectory of length $10^{N/2}$ can be shadowed with error at most $10^{-N/2}$. In our case $N \sim 16$, so the prediction of [HYG] is that shadowing of 10^8 terms should exist with error less than 10^{-8}; this is indeed what is seen in Figure 4. But we will show in a moment that the long-term error growth is very much better than such predictions: the error becomes constant.

Focusing on a single noisy trajectory, starting at 0.1, as being typical, we can compute B as the trajectory grows and use its growth as a guide to how the shadow error grows. Figure 4 shows in log-log form the shadow-error bound B as the number of terms grows. As one expects from the fact that B is computed from τ and σ, most often B does not change at all, since a change in τ or σ requires a new record in terms involving derivatives. But every once in a while the trajectory gets closer to 0.5 than previously, which causes a new maximum in the reciprocal of the derivative, and hence a jump in B.

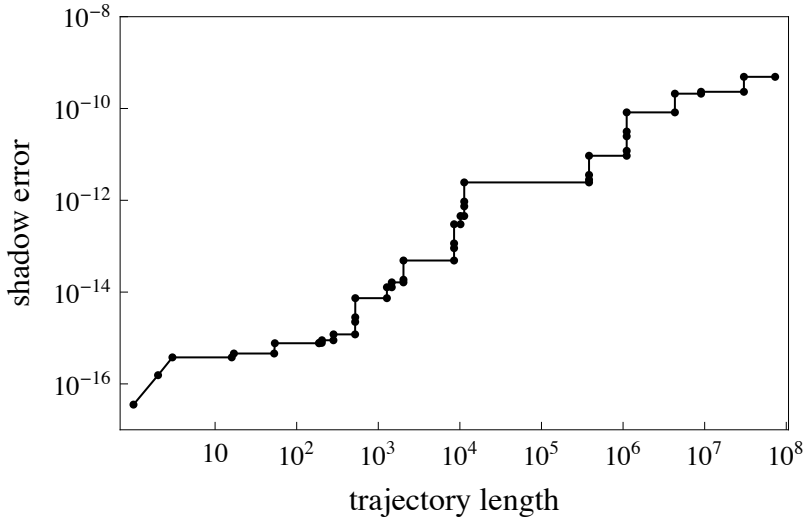


Figure 4. The shadow error for noisy trajectories Y_N of $f_{3.8}$ starting at 0.1. The growth is roughly linear in the log-log scale for the domain shown.

Such a shadow-error plot leads naturally to the question of what happens as N gets really large, the expectation from the graph being that the shadow error will grow until it reaches 1, at which point it is useless, since the zero trajectory always has shadow error at most 1. But the graph is totally misleading! This is because any machine precision trajectory must eventually become periodic: there are only finitely many machine reals. This observation was made by Sauer [S] who computed several of the attracting cycles for machine-precision trajectories of f_4. We can do the same for $f_{3.8}$. And the key point is that if the trajectory is periodic, so are σ, τ, and B. This means that there is a maximum value of B and, if it is small, shadowing stays good forever.

To see exactly how this works, one starts with $y_0 = 0.1$ and computes y_{10^8}. Then restart and find q, the number of steps to first hit y_{10^8}; the period p must then divide $10^8 - q$ and in this example $p = 72\,065\,374$. A closer look finds that the first entry of the p-cycle is $y_{1\,007\,602}$. Now we can compute σ, τ, and B for $Y_{1\,007\,602+72\,065\,374} = Y_{73\,072\,976}$. This is done by first finding a value of p so that μ_p is small. A computation shows that μ_{60} for Y_{10^8} is 0.00974, so we take $p = 60$ guaranteeing error less than 1% when using σ_{60} and τ_{60}. It takes a couple of hours, but we find $\tau_{60} = 4.896 \times 10^{-10}$ and $\sigma_{60} = 16\,209\,825$, and therefore we can shadow $Y_{73\,072\,976}$ with error no greater than $5 \cdot 10^{-10}$. By periodicity we can now say that, for *any* trajectory length N, the shadow error for the machine-precision trajectory Y_N is no greater than $5 \cdot 10^{-10}$. Thus there is good shadow ing forever.

The question of how long good shadowing persists was raised in [HYG], where it is stated that after 10^7 terms, and for their particular numerical orbit, "there may not be any true orbit staying near the numerical orbit". But by combining periodicity with the Chow–Palmer method we have shown that for one typical case (the method should work for other values of the parameter and starting value) the concern is unjustified: any initial segment of the infinite machine-precision trajectory can be shadowed.

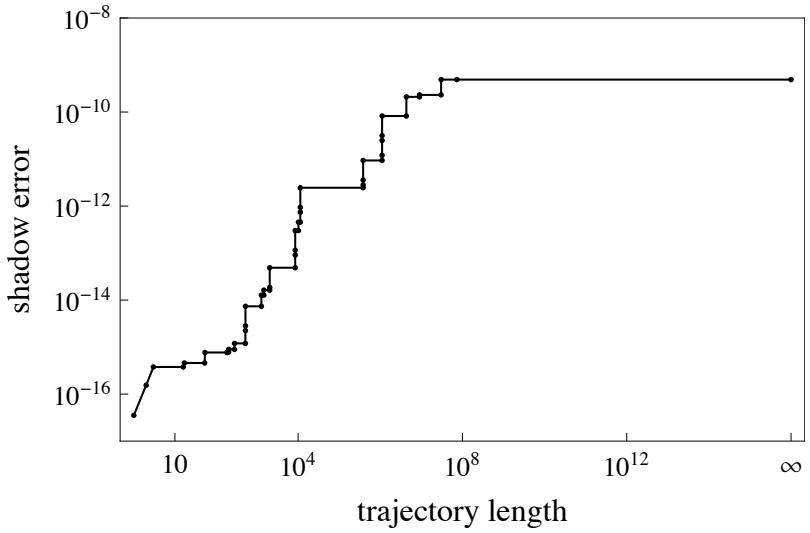


Figure 5. The shadow error for noisy trajectories Y_N of $f_{3.8}$ starting at 0.1 as N grows without bound. Beyond 10^8 the shadow error stays under 10^{-9} forever.

This technique to obtain infinite shadowing will not work for *any* pseudo-orbit of $f_{3.8}$; the periodicity of the machine-precision orbit is critical. And there is another concern in the case of f_4, called a *glitch*. A glitch in a pseudo-orbit for f_4 is when it contains a value outside of [0,1], as can happen with the sequence $\{0.5,\ 1.0 + 10^{-15},\ -4.0 \cdot 10^{-15},\ -16.0 \cdot 10^{-14}\}$. This sequence is a pseudo-orbit because each value y_{k+1} is very close to $f_4(y_k)$. Shadowing is impossible for such a pseudo-orbit because any infinitely precise real number near $1/2$ has a trajectory that stays within [0, 1]. However, Sauer [S] has proved that glitches cannot occur in machine-precision orbits of f_4 so long as the IEEE standard is used. So a natural question is whether there is a pseudo-orbit for $f_{3.8}$, not a machine-precision orbit, for which there is no meaningful shadow.

4. A billiard shadow

Problem 2 of the SIAM 100-Digit Challenge [BLWW] concerned the path of a photon of light traveling in a plane with reflective mirrors of radius 1 / 3 centered at the integer lattice points. Of course, round-off error builds up in any machine-precision computation of the photon's path; Figure 6 shows that the round-off fully contaminates the machine-precision trajectory after 18 bounces.

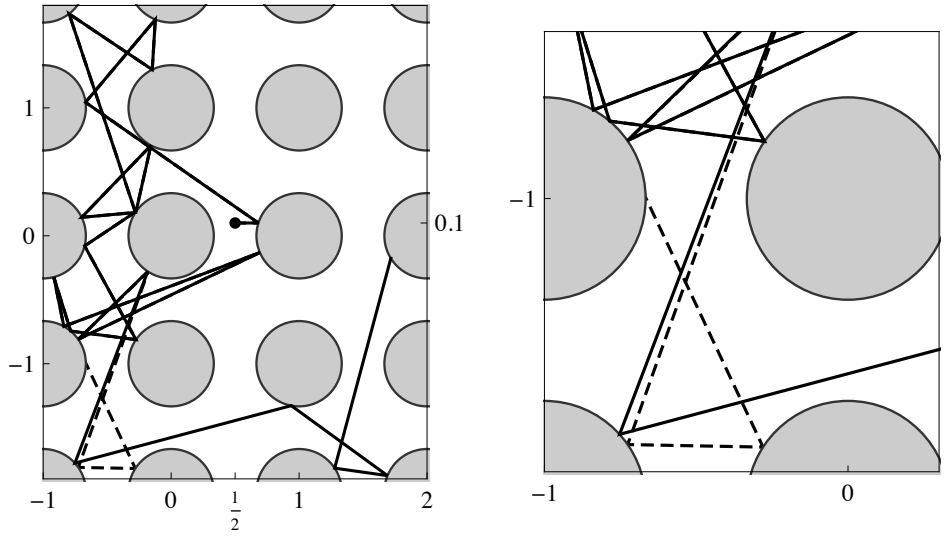


Figure 6. The machine precision trajectory (dashed) diverges from the true trajectory of (1 / 2, 1 / 10) after 17 reflections.

Focusing on the parameters of the contest problem, let Y_N to be the sequence of reflection points, computed in machine precision, for a photon starting at (0.5, 0.1) and moving east at speed 1. To shadow Y_N we simply minimize an objective function that measures the sum-of-squares error for the $2\,(N+1)$ numbers that form a high-precision path starting at $(1/2,\, y_0)$ with initial velocity (1, 0). To compute both low- and high-precision paths of the photon we use an elegant algorithm found by Fred Simons [BLWW, p. 35].

While it is possible to compute the Jacobian using the algebra and calculus of line-circle intersections [H], it is a little complicated because of the multidimensional nature of the state space (the reflection point and the direction of travel). It turns out that the optimization is much faster if one simply lets *Mathematica*'s minimization algorithm compute numerical approximations to the Jacobian by finite differences; that approach yields approximations to the Jacobian accurate to about half of the working precision, which is adequate. Using 3500 digits of precision we can shadow the first 500 terms of the noisy path. Figure 7 shows this shadow trajectory, which starts $3.023354735\ldots\cdot 10^{-17}$ north of $(1/2,\, 1/10)$. The trajectory is visually identical to Y_{500} as the sup-error is less than $3\cdot 10^{-15}$.

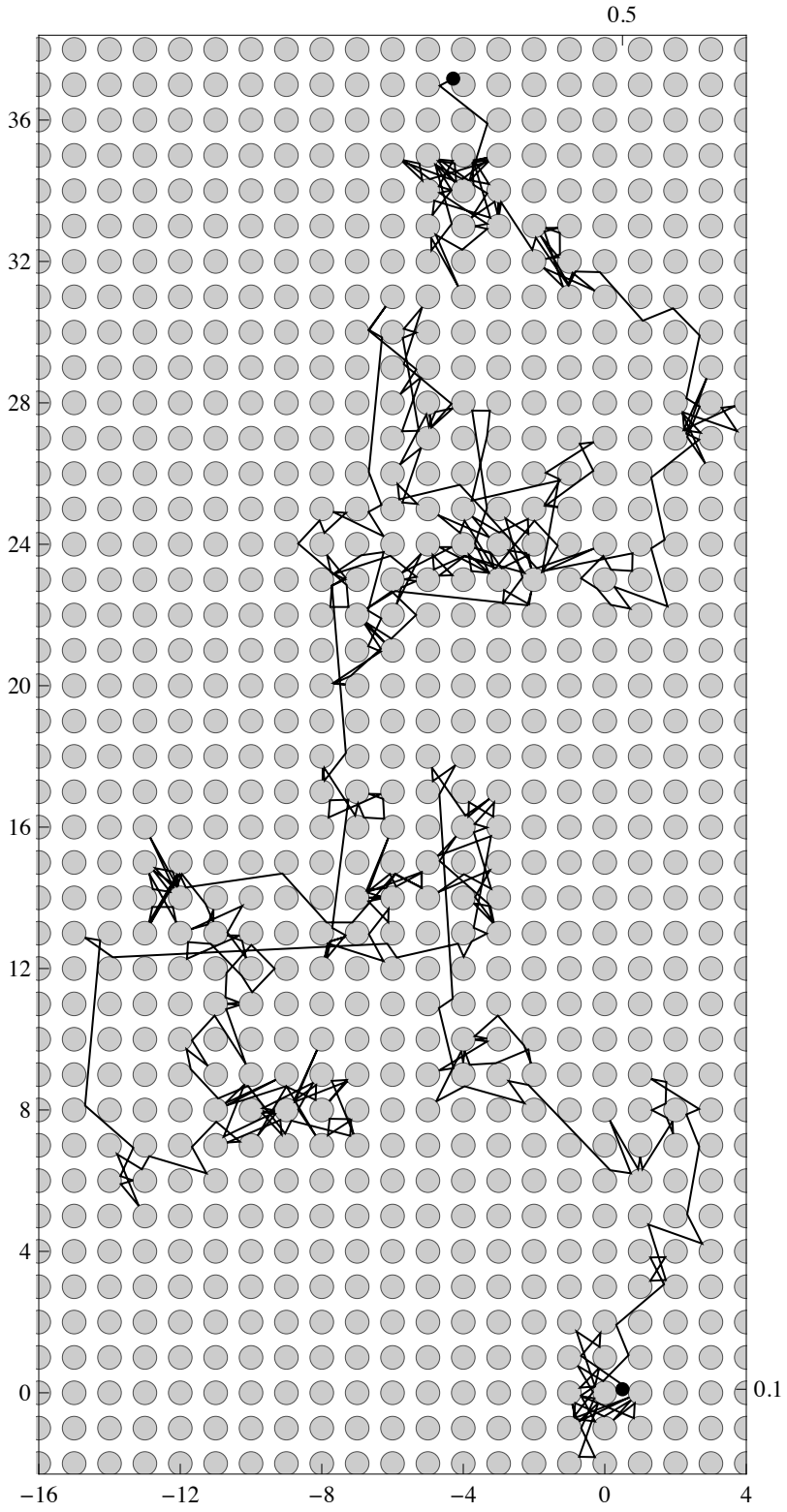


Figure 7. Five hundred reflections of both the machine-precision trajectory from $(0.5, 0.1)$ and the high-precision shadow starting from about 10^{-17} north of $\left(\frac{1}{2}, \frac{1}{10}\right)$. The two paths agree to within 10^{-14}.

It is not a surprise that this billiard problem admits shadowing, but there are difficulties in deriving this from hyperbolicity considerations (see [BLWW, p. 40]). While the present computation can be viewed as evidence for the existence of good shadows for this billiard problem, it is not clear how exactly the error will grow as the noisy trajectory increases to, say, a million terms. Figure 8 shows the shadow error for the first 500 bounces; it appears that the growth is linear.

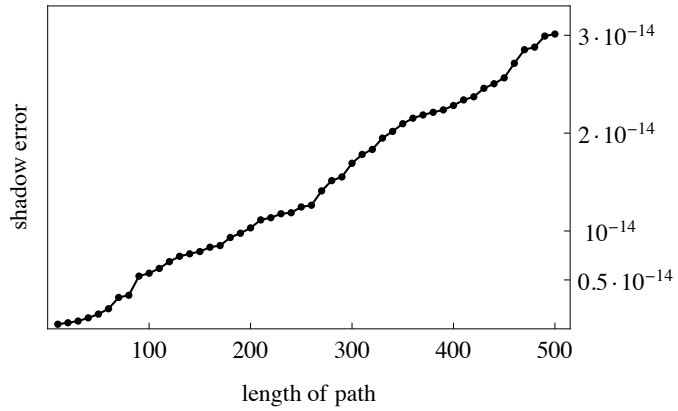


Figure 8. The shadow error for the first 500 photon reflections.

5. Shadowing chaotic differential equations

■ The Duffing differential equation

The Duffing differential equation, which arises from a forced damped oscillator with a cubic stiffness term, is: $x'' + a\,x' - b\,x + c\,x^3 = d\cos(e\,t)$. We will focus here on $x'' + \frac{15}{100}\,x' - x + x^3 = \frac{3}{10}\cos t$, which has apparently chaotic behavior for certain initial values. A typical machine-precision trajectory in the phase plane view (x, x') is shown in Figure 9, where the start is $(-1, 1)$.

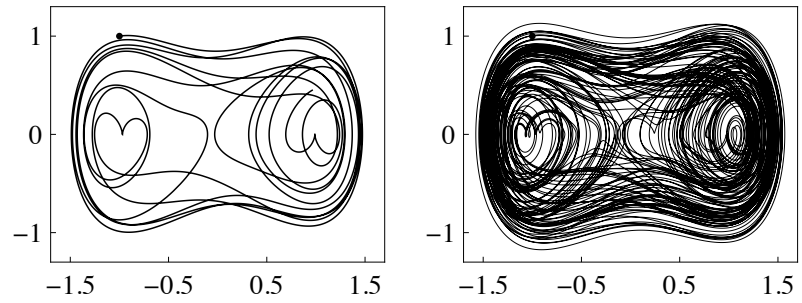


Figure 9. The solution to the Duffing equation starting from $(-1, 1)$ has a chaotic trajectory: time goes to 100 on the left and 1000 on the right.

A Poincaré section using steps of size $2\,\pi$ shows the apparent existence of a strange attractor. Figure 10 shows such a machine-precision section of 25000 points (time runs to 157000). For other starting values between -1 and 1 one sees an essentially identical picture. Yet the existence of such an attractor has not been rigorously proved. Indeed, discussing this situation Guckenheimer and Holmes [GH, p. 91] state that, "There is a substantial theoretical question as to whether this 'strange attractor' of the Duffing equation is an artifact of the noise and is absent in the ideal deterministic system." Thus the shadowing problem is of interest here as we would like to know if a machine-precision solution is close to a true solution.

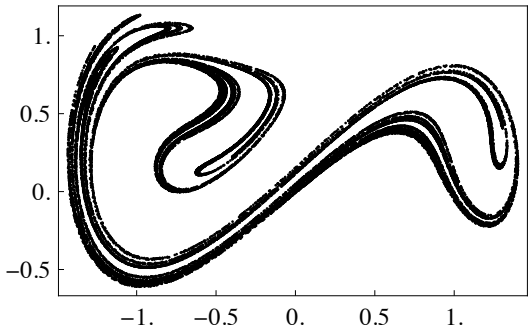


Figure 10. A Poincaré section consisting of 25000 points from a machine-precision solution out to time $25\,000 \cdot 2\,\pi$.

We can use optimization to find a Duffing shadow, but we now have a two-dimensional problem; if the noisy solution starts at $x(0) = -1$, $x'(0) = 1$, then the shadow seed should be sought near the point $(-1, 1)$.

A key point is that we can get information about the sensitivity by extending the Duffing equation to a sensitized Duffing system. Consider the following sixth-order system, where the dependence on t is suppressed.

Duffing: $$x'' + \tfrac{15}{100}\, x' - x + x^3 = \tfrac{3}{10} \cos t$$

Sensitivity with respect to $x(0)$:

$$dx_0'' + \tfrac{15}{100}\, dx_0' - dx_0\left(1 - 3\, x^2\right) = 0; \qquad dx_0(0) = 1, \quad dx_0'(0) = 0$$

Sensitivity with respect to $x'(0)$:

$$dxp_0'' + \tfrac{15}{100}\, dxp_0' - dxp_0\left(1 - 3\, x^2\right) = 0; \quad dxp_0(0) = 0, \;\; dxp_0'(0) = 1$$

This system encapsulates information about the sensitivity of the solution with respect to changes in the initial values. One views x implicitly as a function of not only t but also the two initial values $x\,(0)$ and $x'(0)$. Then dx_0 represents the partial derivative of x with respect to $x(0)$ and its equation is derived by taking the partial derivative of the Duffing equation with respect to $x(0)$.

We also discretize the problem by just looking at the solution at times 0, 1, 2, 3, … . This step-size of 1 turns out to be small enough to avoid the problem of a shadow that is good for a discrete set of points, but not for the underlying continuous function. We can solve the sensitivity system to very high precision: *Mathematica* does this by using extrapolation methods that have adapative step-size and order. And we used the technique of [KW] to verify correctness of the high-precison solutions to the specified tolerances.

Using our optimization technique with the Jacobian obtained by a high-precision solution to the sensitivity system, and bootstrapping by starting with the first 75 noisy points, and then moving forward in steps of 25 (and 15 at the end), we can find a good value for seed_{140}. It is roughly $(-1, 1) + (3.229, -4.961) \cdot 10^{-8}$ and has a sup error for the 141 points of $3.7 \cdot 10^{-6}$. Figure 11 shows how the sup-error grows when we use the shadow found on initial segments of the noisy trajectory. Using the proper working precision as one moves up can save time. Knapp and Wagon [KW] estimated the Lyapunov exponent of this Duffing equation at 0.2 and used this to estimate $0.46\,t$ for the (decimal) precision needed as a function of time t. This allows one to use lesser precision than the maximum as one works up from smaller values of t to larger.

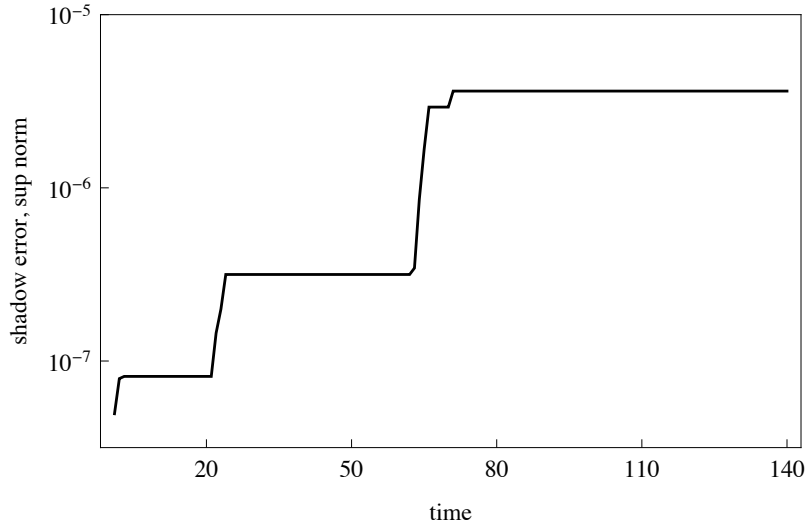


Figure 11. A log-log plot of the error growth when shadowing the Duffing equation.

The shadow provides a little evidence that the machine-precision attractor is a true feature of the infinitely precise system. For more evidence we can look at a true orbit starting at a point other than $(-1, 1)$. One difficulty is that for a true computation one needs both high working precision and many steps in the numerical differential equation solver. Figure 12 shows a small such section computed by starting at $(1/5, 1/5)$ and going to time 160 using 80 digits of precision. The 25 points are superimposed on the machine-precision section from $(-1, 1)$ and they are consistent with the assertion that the observed attractor really exists. In this and the shadowing work one must be certain that the precison used for the high-precision Duffing computation is adequate to guarantee correctness for the time interval used; this was checked using the perturbation technique described in [KW].

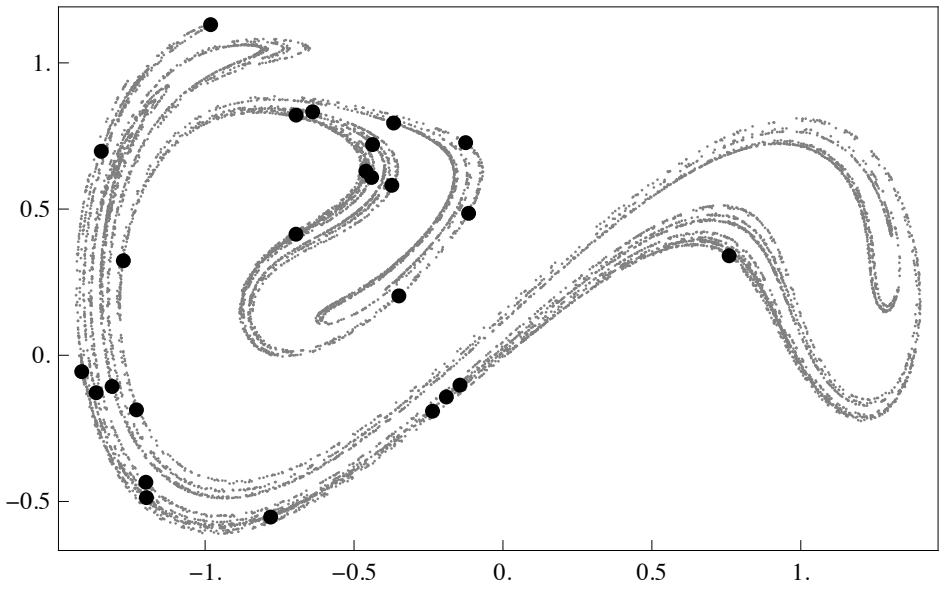
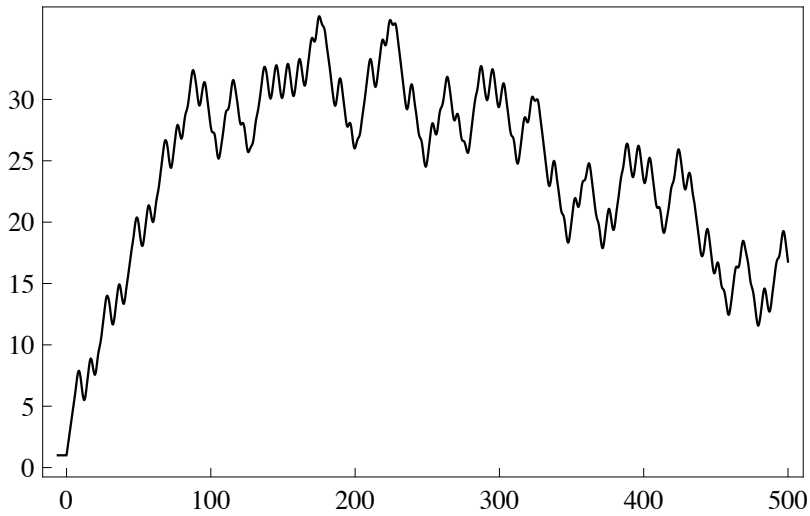


Figure 12. The apparent attractor computed using machine precision and a small (25 points) but true Poincaré section of a solution starting at $(1/5, 1/5)$, computed with high precision.

■ The Ikeda delay differential equation

The Ikeda delay equation, $x'(t) = \sin(x(t-\tau))$, where τ is the delay constant, is very simple but yet has chaotic dynamic behavior: We consider teh case where the initial history function $x(t)$ for $t \le 0$ is a constant x_0. Our noisy trajecotory uses the specific value $x_0 = 1$. The interest here is that such a simple equation without the delay cannot exhibit such behavior. For more about this equation see [Sp]; see [K] for a dynamic demonstration with varying parameters. Figure 13 shows a machine precision solution out to time 500.



Estimates of the Lyapunov exponent lead to the conclusion that 70 digits of precision is enough to get correct solutions out to time 500; this allows our optimization technique to obtain a shadow out to time 500. *Mathematica* solves delay differential equations using the method of steps; when high precision is used, extrapolation methods are used as the base method. The sup error is under 0.000025. As with the Duffing shadow, a key step is feeding the sensitivity to the optimization algorithm. In this case the sensitivity is computed by solving, in high precision, the following system of two delay equations.

$$x'(t) = \sin(x(t-\tau)) \text{ and } s'(t) = s(t-\tau)\cos(x(t-\tau))$$

with initial condition of x_0 on x and 1 on s for negative t. Here x_0 is the independent variable in the optimization; that is, the optimization starts at $x_0 = 1$ and then varies that to minimize the norm of the difference between the high-precision and noisy solutions, discretized by time-step 1. The final value for seed_{500} is 1.000000031055798402067144036827603063219528209017577540987344675779120 4, which shadows our machine-precision trajectory to time 500.

6. Conclusion

Our work shows that one can in many cases get the explicit shadowing trajectories for a few hundred, or even a few thousand, terms of a noisy trajectory. High-precision is the key and computers are now fast enough to run numerical algorithms using a thousand or more digits of working precision. *Mathematica*'s environment, which includes the ability to use high-precision reals not only in standard arithmetic operations, but also throughout the numerical optimization routines, makes our approach possible with minimal programming effort. There are some subtleties though, such as the need to extend machine-precision reals to high-precision reals so that the former can be used in high-precision computations that involve the latter.

References

[BLWW] F. Bornemann, D. Laurie, S. Wagon, and J. Waldvogel, *The SIAM 100-Digit Challenge: A Study in High-Accuracy Numerical Computing*, SIAM, Philadelphia, 2004.

[CP] S.-N. Chow and K. J. Palmer, On the numerical cpmpuitation of orbits of dynamical systems: The one-dimensional case, *Journal of Dynamics and Differential Equations* **3** (1991) 361–379.

[CV] S.-N. Chow and E. S. Van Vleck, Shadowing of lattice maps, in *Chaotic Numerics*, vol. 172 in Contemporary Mathematics, Amer. Math. Soc., Providence, R.I., 1994.

[GH] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Springer, New York, 1983.

[H] H. Hakonsen, *Shadowing Chaos via Optimization*, Honors thesis, Macalester College, 2009.

[HYG] S. M. Hammel, J. A. Yorke, and C. Grebogi, Do numerical orbits of chaotic dynamical processes represent true orbits?, *Journal of Complexity,* **3** (1987) 136–145.

[K] R. Knapp, Ikeda delay differential equation, from the Wolfram Demonstrations Project, http://demonstrations.wolfram.com/IkedaDelayDifferentialEquation, 2008.

[KW] R. Knapp and S. Wagon, Orbits worth betting on!, *C•ODE•E Newsletter* (Consortium for Ordinary Differential Equations Experiments) Winter 1996, 8–13.

[S] T. Sauer, Computer arithmetic and sensitivity of natural measure, *Journal of Difference Equations and Applications* **11** (2005) 669–676.

[Sp] J. C. Sprott, A simple chaotic delay differential equation, *Physics Letters A,* **366** (2007) 397–402.

[W] S. Wagon, *Mathematica in Action*, third ed., Springer, New York, 2009.