Random Fibonacci Sequences and the Number 1.13198824... [*]

Divakar Viswanath [†]

September 30, 1997

Abstract

For the familiar Fibonacci sequence — defined by $f_1 = f_2 = 1$, and $f_n = f_{n-1} + f_{n-2}$ for $n > 2$ — f_n increases exponentially with n at a rate given by the golden ratio $(1 + \sqrt{5})/2 = 1.61803398\ldots$. But for a simple modification with both additions and subtractions — the *random* Fibonacci sequences defined by $t_1 = t_2 = 1$, and for $n > 2$, $t_n = \pm t_{n-1} \pm t_{n-2}$, where each ± sign is independent and either + or − with probability 1/2 — it is not even obvious if $|t_n|$ should increase with n. Our main result is that

$$\sqrt[n]{|t_n|} \to 1.13198824\ldots \quad \text{as} \quad n \to \infty$$

with probability 1. Finding the number 1.13198824... involves the theory of random matrix products, Stern-Brocot division of the real line, a fractal-like measure, a computer calculation, and a rounding error analysis to validate the computer calculation.

1 Introduction

1.1 Random Fibonacci Sequences

The Fibonacci numbers defined by $f_1 = f_2 = 1$ and $f_n = f_{n-1} + f_{n-2}$ for $n > 2$ are widely known. It is equally well-known that $|f_n|$ increases exponentially

[*] This work was supported in part by NSF Grant DMS-9500975CS and DOE Grant DE-FG02-94ER25199 to Lloyd N. Trefethen

[†] Department of Computer Science, Cornell University, Ithaca, NY 14853 (divakar@cs.cornell.edu)

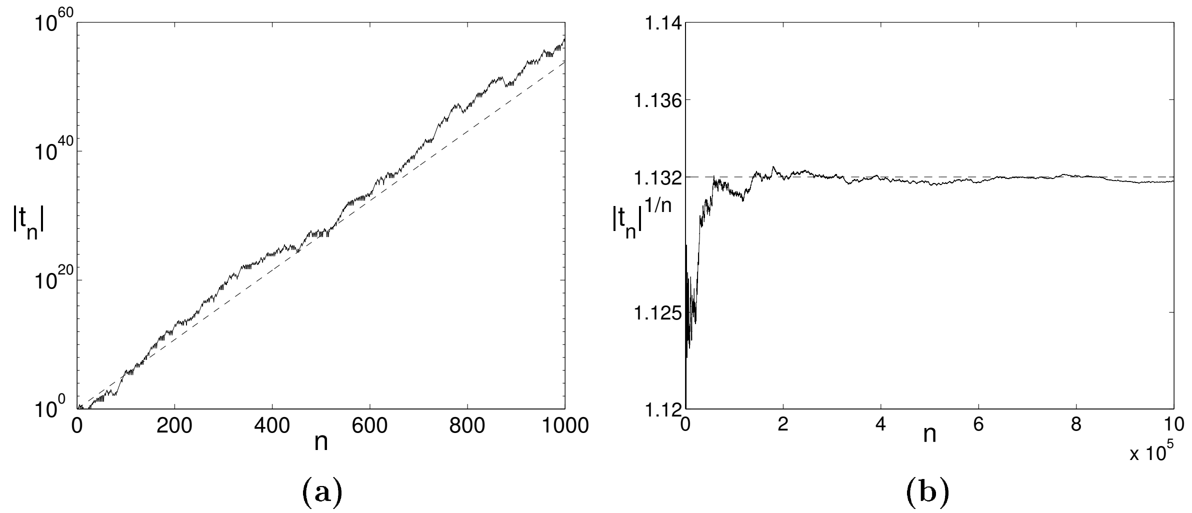


Figure 1: (a) A semilog plot of $|t_n|$ vs. n for a computer generated random Fibonacci sequence t_n showing a clear exponential trend. The dashed line is 1.132^n. (b) Plot of $\sqrt[n]{|t_n|}$ vs. n. As n increases to a million, $\sqrt[n]{|t_n|}$ seems to settle down to a constant close to 1.132.

with n at the rate $(1+\sqrt{5})/2$. Consider random Fibonacci sequences defined by the random recurrence $t_1 = 1$, $t_2 = 1$, and for $n > 2$, $t_n = \pm t_{n-1} \pm t_{n-2}$, where each $\pm$ sign is independent and either $+$ or $-$ with probability $1/2$. Do the random Fibonacci sequences level off because of the subtractions? Or do the random Fibonacci sequences increase exponentially with n like the Fibonacci sequence? If so, at what rate? The answer to these questions brings Stern-Brocot sequences, a beautiful way to divide the real number line that was first discovered in the 19th century, and fractals and random matrix products, both areas of active current research, into play. The final answer itself is obtained from a computer calculation, raising questions about computer assisted theorems and proofs.

Below are three possible runs of the random Fibonacci recurrence:

$$1, 1, -2, -3, -1, 4, -3, 7, -4, 11, -15, 4, -19, 23, -4, \ldots$$
$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 134, 223, 357, 580, \ldots$$
$$1, 1, -2, 1, 1, -2, 1, 1, -2, 1, 1, -2, 1, 1, -2, \ldots$$

The first of the runs above was randomly generated on a computer. The second run is the familiar Fibonacci sequence. The last of the three runs above is a sequence that remains bounded as $n \to \infty$. But such runs with no exponential growth occur with probability 0. For longer, typical runs see Figure 1. Numerical experiments in Figure 1 illustrate our main result (Theorem 3.2) that

$$\sqrt[n]{|t_n|} \to 1.13198824\ldots \quad \text{as} \quad n \to \infty$$

with probability 1. Thus $1.13198824\ldots$ gives the exponential rate of increase of $|t_n|$ with n for random Fibonacci sequences just as the golden ratio $(1+\sqrt{5})/2$ gives the exponential rate of increase of the Fibonacci numbers, $f_1 = f_2 = 1$ and $f_n = f_{n-1} + f_{n-2}$ for $n > 2$.

For the random Fibonacci recurrence $t_n = \pm t_{n-1} \pm t_{n-2}$ as well as the recurrence $t_n = \pm t_{n-1} + t_{n-2}$ with each $\pm$ independent and $+$ or $-$ with probability 1/2, $|t_n|$ is either $|t_{n-1}| + |t_{n-2}|$ or $\big||t_{n-1}| - |t_{n-2}|\big|$ with probability 1/2. As our interest is in $|t_n|$ vs. n as $n \to \infty$, we restrict focus to $t_n = \pm t_{n-1} + t_{n-2}$ and call it the random Fibonacci recurrence. As a result, the presentation becomes briefer, especially in Section 2.

The next step is to rewrite the random Fibonacci recurrence using matrices. In matrix form the random Fibonacci recurrence is $\left(\begin{smallmatrix} t_{n-1} \\ t_n \end{smallmatrix}\right) = \left(\begin{smallmatrix} 0 & 1 \\ 1 & \pm 1 \end{smallmatrix}\right)\left(\begin{smallmatrix} t_{n-2} \\ t_{n-1} \end{smallmatrix}\right)$, with one of the two matrices

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \tag{1.1}$$

picked independently with probability 1/2 at each step. Let μ_f denote the distribution that picks A or B with probability 1/2. Then the random matrix M_n chosen at the nth step is μ_f-distributed and independent of M_i for $i \neq n$. Moreover,

$$\begin{pmatrix} t_{n-1} \\ t_n \end{pmatrix} = M_{n-2} \ldots M_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

where $M_{n-2} \ldots M_1$ is a product of independent, identically distributed random matrices.

Known results from the theory of random matrix products imply that

$$\frac{\log \|M_n \ldots M_1\|}{n} \to \gamma_f \quad \text{as} \quad n \to \infty, \tag{1.2}$$

$$\sqrt[n]{|t_n|} \to e^{\gamma_f} \quad \text{as} \quad n \to \infty, \tag{1.3}$$

for a constant γ_f with probability 1 [6, p. 11, p. 157]. About γ_f itself known theory can only say that $\gamma_f > 0$ [6, p. 30]. Our aim is to determine γ_f or e^{γ_f} exactly. Theorem 3.2 realizes this aim by showing that $e^{\gamma_f} = 1.13198824\ldots$. In (1.2) and the rest of this paper, all norms are 2-norms, and all matrices and vectors are real and 2-dimensional. For a vector x, $\|x\|$ is its Euclidean length in the real plane. For a matrix M, $\|M\| = \sup_{x \neq 0} \frac{\|Mx\|}{\|x\|}$. However, (1.2) is true for any norm over 2-dimensional matrices and the limit γ_f is independent of the norm, because all norms over a finite dimensional vector space are equivalent.

Limit (1.2) for M_i independent but identically distributed over d-dimensional matrices has been a central concern of the theory of random matrix products. Furstenberg and Kesten [16, 1960] have shown that limit (1.2) exists under very general conditions. When it exists, that limit is usually denoted by γ and called the upper Lyapunov exponent. Furstenberg [15, 1963] has shown that when the normalizing condition $|\det M_i| = 1$ holds, as it does for μ_f, "usually" $\gamma > 0$. Furstenberg's theorem implies, for example, that $\gamma_f > 0$, and hence, that $|t_n|$ increases exponentially with n with probability 1.

In spite of the importance of the upper Lyapunov exponent γ , γ is known exactly for very few examples. Kingman, one of the pioneers of subadditive ergodic theory of which the theory of random matrix products is a special case, wrote [22, 1973]:

> Pride of place among the unsolved problems of subadditive ergodic theory must go to the calculation of the constant γ (...). In none of the applications described here is there an obvious mechanism for obtaining an exact numerical value, and indeed this usually seems to be a problem of some depth.

One of the applications Kingman refers to is the general problem of finding γ for random matrix products. For this and other applications, Kingman's problem is still unsolved. Bougerol [6, p. 33], Lima and Rahibe [26] calculate γ for some examples. The work of Chassaing, Letac and Mora [9] is closer to our determination of γ_f. But in all their examples, matrices, unlike B in (1.1), have only non-negative entries. In our opinion, the random Fibonacci recurrence is more natural than these examples. In fact, the random Fibonacci recurrence in a more general form appears as a motivating example in the very first paragraph of Furstenberg's famous paper [15].

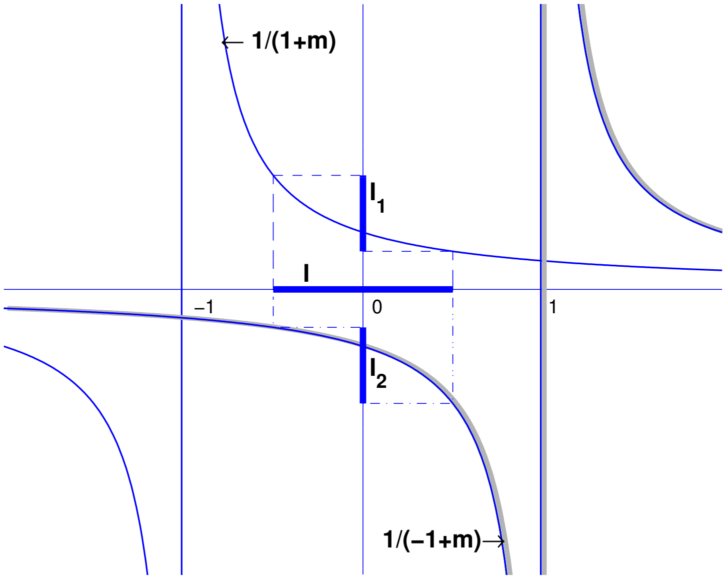


Figure 2: By (1.5), $\nu_f(I) = \nu_f(I_1)/2 + \nu_f(I_2)/2$.

1.2 Furstenberg's Formula

To determine γ_f, we use a formula from the theory of random matrix products that complements (1.2). Three things that will be defined below — the notation $\overline{x}$ for directions in the real plane R^2, $\text{amp}(\overline{x})$ which is a smooth function of $\overline{x}$ (Figure on p. 6), and $\nu_f(\overline{x})$ which is a probability measure over directions $\bar{x}$ (Figure 4) — combine to give a formula for γ_f:

$$\gamma_f = \int \text{amp}(\overline{x}) d\nu_f(\overline{x}). \tag{1.4}$$

This formula, derived by Furstenberg [6, p. 77], is the basis of our determination of γ_f.

Directions $\overline{x}$ can be parameterized using angles, $\overline{x} = \binom{\cos\theta}{\sin\theta}$ with $\theta \in (-\pi/2, \pi/2]$, or using slopes, $\overline{x} = \binom{1}{m}$ with $m \in (-\infty, \infty]$. Slopes m and angles θ are related by $m = \tan\theta$ and $\theta = \arctan m$. We use slopes in all places except Figure 4. In our notation, x is a vector in the direction $\overline{x}$, and $\overline{x}$ is the direction of the vector x for $x \neq 0$.

To define ν_f, consider the μ_f-induced random walk on directions that sends $\overline{x_0}$ to $\overline{x_1} = \overline{Ax_0}$ or to $\overline{x_1} = \overline{Bx_0}$ with probability 1/2, and then sends $\overline{x_1}$ to $\overline{x_2}$ similarly, and so on. In terms of slopes, the slope m is mapped by the random walk to $1+1/m$ or to $-1+1/m$ with probability $1/2$. The measure ν_f is the unique *invariant* probability measure over $\overline{x}$ for this random walk, i.e.,

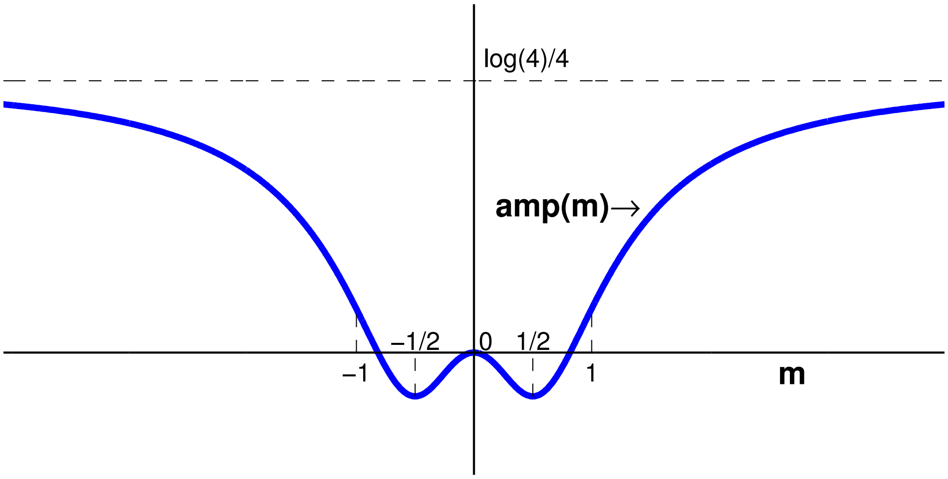
$$\nu_f(S) = \frac{1}{2}\nu_f(\overline{A^{-1}S}) + \frac{1}{2}\nu_f(\overline{B^{-1}S}),$$

where S is any Borel measurable set of directions. We also say that ν_f is μ_f-invariant. For the existence and uniqueness of ν_f, see [6, p. 10, p. 32]. It is also known that ν_f must be *continuous* [6, p. 32], i.e., $\nu_f(\{\overline{x}\}) = 0$ for any fixed direction $\overline{x}$.

Since the bijections $\overline{x} \to \overline{A^{-1}x}$ and $\overline{x} \to \overline{B^{-1}x}$ (sometimes called backward maps) map the slope m to $1/(-1+m)$ and to $1/(1+m)$ respectively, the condition for μ_f-invariance in terms of slopes is

$$\nu_f([a,b]) = \frac{1}{2}\nu_f\Big(\frac{1}{-1+[a,b]}\Big) + \frac{1}{2}\nu_f\Big(\frac{1}{1+[a,b]}\Big), \qquad (1.5)$$

where $[a,b]$ is any interval in the real line. See Figure 2.



The function $\text{amp}(\overline{x})$ defined by

$$\text{amp}(\overline{x}) = \frac{1}{2}\log\frac{\|Ax\|}{\|x\|} + \frac{1}{2}\log\frac{\|Bx\|}{\|x\|}$$

gives the average amplification in the direction $\overline{x}$ when x is multiplied by A or B with probability $1/2$. Recall that $\|\cdot\|$ was taken to be the 2-norm. In terms of slopes,

$$\begin{aligned}\operatorname{amp}(m) &= \frac{1}{4}\log\Big(\frac{m^2+(-1+m)^2}{1+m^2}\Big) + \frac{1}{4}\log\Big(\frac{m^2+(1+m)^2}{1+m^2}\Big)\\ &= \frac{1}{4}\log\Big(\frac{1+4m^4}{(1+m^2)^2}\Big).\end{aligned}$$

The figure above plots $\operatorname{amp}(m)$ vs m.

Furstenberg's formula (1.4) can now be put in a concrete form using slopes to parameterize directions $\overline{x}$:

$$\gamma_f = \int_{-\infty}^{\infty} \operatorname{amp}(m)\, d\nu_f(m) = \frac{1}{4}\int_{-\infty}^{\infty} \log\Big(\frac{1+4m^4}{(1+m^2)^2}\Big)\, d\nu_f(m). \tag{1.6}$$

To illustrate how (1.6) is used, we verify quickly that $\gamma_f > 0$. The invariance condition (1.5) applied to the set $[-\infty,-1]\cup[1,\infty]$ implies $\nu_f\big(|m|\geq 1\big) \geq 1/2$ because $1/(-1+[1,\infty]) = [0,\infty]$ and $1/(1+[-\infty,-1]) = [-\infty,0]$. Now,

$$\begin{aligned}\gamma_f &= \int_{-\infty}^{\infty} \operatorname{amp}(m)\, d\nu_f(m)\\ &> \min_{|m|<1} \operatorname{amp}(m)\ \ \nu_f\big(|m|<1\big) + \min_{|m|\geq 1} \operatorname{amp}(m)\ \ \nu_f\big(|m|\geq 1\big)\\ &= -\frac{1}{4}\log\Big(\frac{5}{4}\Big)\ \nu_f\big(|m|<1\big) + \frac{1}{4}\log\Big(\frac{5}{4}\Big)\ \nu_f\big(|m|\geq 1\big)\\ &\geq 0.\end{aligned}$$

The first inequality above is strict because ν_f must be continuous and $\operatorname{amp}(m)$ is not a constant function. Minimizing $\operatorname{amp}(m)$ over $|m|<1$ and $|m|\geq 1$ is basic calculus: the minima occur at the points $m = \pm 1/2$ and $m = \pm 1$. The final $\geq$ is by $\nu_f\big(|m|\geq 1\big) \geq 1/2$. Actually, it will be shown in Section 2 that $\nu_f\big(|m|\geq 1\big) = (\sqrt{5}-1)/2$.

1.3 Overview and Some Remarks

In Section 2, which is the heart of this paper, we construct ν_f exactly using the Stern-Brocot division of the real line. See Figures 3, 4. The computer calculation leading to $e^{\gamma_f} = 1.13198824\ldots$ and its rounding error analysis are given in the appendix. The main features of the computation are summarized in Section 3. Our main result is Theorem 3.2 ($e^{\gamma_f} = 1.13198824\ldots$) which is finally an elementary answer to an elementary question about $\sqrt[n]{|t_n|}$.

The measure ν_f determined in Section 2 is shown in Figure 4. Repetition of the same structure at finer scales and an irregular appearence suggest that ν_f may be a type of fractal. We will say what we precisely mean by a fractal in Section 2, but we explain now why it should be no surprise if ν_f were a fractal. Recall that by (1.5) ν_f is the probability measure over R satisfying

$$\nu_f([a,b]) = \frac{1}{2}\nu_f\Big(\frac{1}{-1+[a,b]}\Big) + \frac{1}{2}\nu_f\Big(\frac{1}{1+[a,b]}\Big).$$

In words, transforming the measure ν_f under the two maps $m \to \pm 1 + 1/m$ and then averaging gives back ν_f. Since the slopes of the backward maps $m \to 1/(\pm 1 + m)$ vary in magnitude from 0 to ∞, not only is ν_f self-similar [32], the self-similarity equation has multiple scales. Self-similar functions, especially ones with multiple scales, usually turn out to be fractals. For example, Weierstrass's nowhere-differentiable but continuous functions, which are commonly used examples of fractal graphs, satisfy $f(x) = \lambda^{s-2}\sin(\lambda t) + \lambda f(\lambda t)$ with $1 < s < 2$, $\lambda > 1$, and λ large enough [14]. An other remarkable example is Daubechies' wavelets; the solution to

$$f(t) = \frac{1+\sqrt{3}}{4}f(2t) + \frac{3+\sqrt{3}}{4}f(2t-1) + \frac{3-\sqrt{3}}{4}f(2t-2) + \frac{1-\sqrt{3}}{4}f(2t-3))$$

has an irregular graph which is a type of fractal, yet it can be used to construct wavelets that approximate smooth functions very well [31, p. 437].

A striking aspect of Theorem 3.2 in Section 3 is that its proof depends on a computer calculation. Thus its correctness depends not only upon mathematical arguments that can be checked line by line, but upon a program that can also be checked line by line and the correct implementation of various software and hardware components of the computer system. The most famous of theorems whose proofs depend on computer calculations is the four color theorem. The first proof of the four color theorem (all planar graphs can be colored using only four colors so that no two adjacent vertices have the same color) by Appel, Haken and Koch caused controversy and aroused great interest because it relied on producing and checking 1834 graphs using 1200 hours of 1976 computer time [24] [2]. In spite of improvements (for example, the number 1834 was brought down to 1482 soon afterwards by Appel and Haken themselves), all proofs of the four color theorem still rely on the computer.

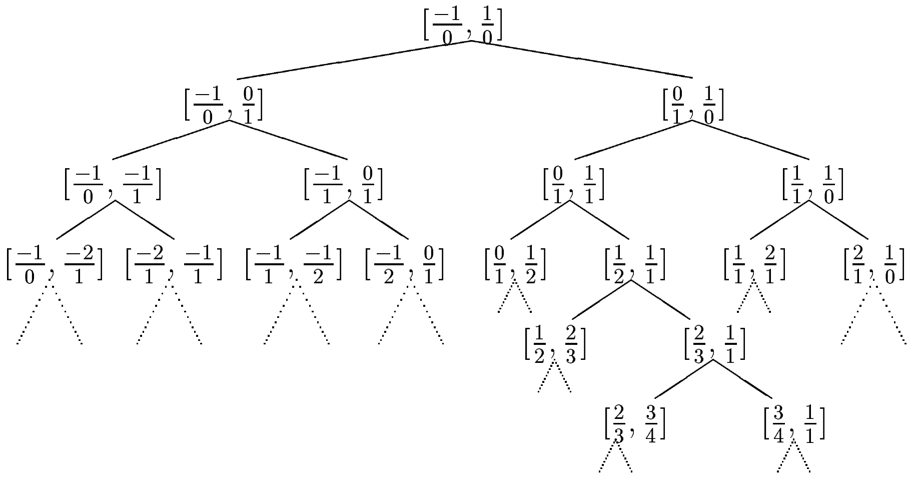
Unlike our computation, however, all computational steps in these proofs of the four color theorem are exact and do not use floating point arithmetic which is inexact owing to rounding errors. There are some claims now of theorems proved using floating point arithmetic validated by rounding error analyses. The computer assisted proof of chaos in the Lorenz equations announced by Mischaikow and Mrozek [27] is a notable example, though the full details of the error analysis are still unpublished. We will discuss the use of floating point arithmetic and other issues related to our Theorem 3.2 in Section 3.

Besides random matrix products, random Fibonacci sequences are connected to many areas of mathematics. For example, the invariant measure ν_f is also the distribution of the continued fractions

$$\pm 1 + \cfrac{1}{\pm 1 + \cfrac{1}{\pm 1 + \cdots}}$$

with each ±1 independent and either $+1$ or -1 with probability $1/2$. The matrices A and B in (1.1) can both be thought of as Möbius transformations of the complex plane; then the random matrix product and the exponential growth of $|t_n|$ in (1.2) and (1.3) would correspond to the dynamics of complex numbers acted upon by a composition of the Möbius transformations A and B [6, p. 38]. Also, the random walk on slopes $m \to 1/(\pm 1 + m)$ can be thought of as a random dynamical system [3]. These different interpretations amount merely to a change of vocabulary as far as the computation of γ_f is concerned; still each interpretation can offer a different point of view.

The study of random matrix products, initiated by Bellman [4, 1954], has led to many deep results and applications. Applications have been made to areas as diverse as Schrödinger operators, image generation, and demography [12][13][34]. Our own interest in random recurrences was aroused by their connection to random triangular matrices [35]. Furstenberg and Kesten [16, 1960], Furstenberg [15, 1963], Osseledac [28, 1968], Kingman [22, 1973], and Guivarc'h and Raugi [18, 1985] are some of the profound contributions to this area. We enthusiastically recommend the lucid, elegant and well-organized account by Bougerol [6]. For a more modern treatment, see [5]. For the basics of probability, our favorite is [7].



$[\frac{-1}{0}, \frac{1}{0}]$
$[\frac{-1}{0}, \frac{0}{1}]$
$[\frac{0}{1}, \frac{1}{0}]$
$[\frac{-1}{0}, \frac{-1}{1}]$
$[\frac{-1}{1}, \frac{0}{1}]$
$[\frac{0}{1}, \frac{1}{1}]$
$[\frac{1}{1}, \frac{1}{0}]$
$[\frac{-1}{0}, \frac{-2}{1}]$
$[\frac{-2}{1}, \frac{-1}{1}]$
$[\frac{-1}{1}, \frac{-1}{2}]$
$[\frac{-1}{2}, \frac{0}{1}]$
$[\frac{0}{1}, \frac{1}{2}]$
$[\frac{1}{2}, \frac{1}{1}]$
$[\frac{1}{1}, \frac{2}{1}]$
$[\frac{2}{1}, \frac{1}{0}]$
$[\frac{1}{2}, \frac{2}{3}]$
$[\frac{2}{3}, \frac{1}{1}]$
$[\frac{2}{3}, \frac{3}{4}]$
$[\frac{3}{4}, \frac{1}{1}]$

Figure 3: The Stern-Brocot tree; its nodes are intervals of the real line R. The division of any interval $[\frac{a}{b}, \frac{c}{d}]$, except the root, into two children is done by inserting the point $\frac{a+c}{b+d}$.

2 The Stern-Brocot Tree and Construction of the Invariant Measure ν_f

Assuming $\pm 1 \notin (a, b)$, we write the invariance condition using slopes (1.5) in a more explicit form:

$$\nu_f([a,b]) = \frac{1}{2}\nu_f\Big([\frac{1}{-1+b}, \frac{1}{-1+a}]\Big) + \frac{1}{2}\nu_f\Big([\frac{1}{1+b}, \frac{1}{1+a}]\Big). \tag{2.1}$$

Our goal in this section is to find ν_f, the unique probability measure on the real line R satisfying (2.1) for all intervals $[a, b]$ not containing ± 1. Since ν_f must be continuous, it does not matter whether we take the intervals in (2.1) to be open or closed or half-closed.

The construction of ν_f is based on the Stern-Brocot tree shown in Figure 3. The Stern-Brocot tree is an infinite binary tree that divides R recursively. Represent ∞ as $\frac{1}{0}$ and 0 as $\frac{0}{1}$, and write negative fractions with the numerator negative. Then the root of the Stern-Brocot tree is the real line $[\frac{-1}{0}, \frac{1}{0}]$. Its left and right children are $[\frac{-1}{0}, \frac{0}{1}]$ and $[\frac{0}{1}, \frac{1}{0}]$, the positive and negative halves of R. The rest of the tree is defined by dividing any node $[\frac{a}{b}, \frac{c}{d}]$ other than the root into a left child $[\frac{a}{b}, \frac{a+c}{b+d}]$ and a right child $[\frac{a+c}{b+d}, \frac{c}{d}]$. For example, the root's left child $[\frac{-1}{0}, \frac{0}{1}]$ divides into $[\frac{-1}{0}, \frac{-1}{1}]$ and $[\frac{-1}{1}, \frac{0}{1}]$.

The Stern-Brocot tree was discovered and reported independently by the mathematician Moriz Stern in 1858 and by the watchmaker Achille Brocot in 1860 [30] [8]. Unaware of its existence, we found it again while trying to construct ν_f. We summarize some basic facts about it in Lemma 2.1. The Stern-Brocot tree and its connections with continued fractions are discussed in detail by Graham, Knuth and Patashnik [17]. Their definition of the Stern-Brocot tree is slightly different from ours. We adopt their notation $a \perp b$ to say that integers a and b are relatively prime.

Lemma 2.1. (a) *The Stern-Brocot tree is symmetric about* 0 *with its right half positive and its left half negative.*

(b) *If* $[\frac{a}{b}, \frac{c}{d}]$ *is a node in the positive half of the Stern-Brocot tree, then* $bc - ad = 1$, $a \perp b$, *and* $c \perp d$.

(c) *Conversely, if* a/b *and* c/d *are non-negative rational numbers with zero and infinity represented as* $\frac{0}{1}$ *and* $\frac{1}{0}$ *respectively, and* $bc - ad = 1$ *then* $[\frac{a}{b}, \frac{c}{d}]$ *occurs as a node in the Stern-Brocot tree. Consequently, every rational number* a/b, $a \perp b$, *appears as an endpoint of a Stern-Brocot interval of finite depth.*

Proof. (a) is obvious; see Figure 3. The proof of (b) is an easy induction on the depth of the tree. (c) is a little bit less easy. Its proof is related to Euclid's algorithm for computing the greatest common divisor of two integers. See [17]. □

We adopt a labelling scheme for Stern-Brocot intervals (nodes of the Stern-Brocot tree) that differs only a bit from that in [17]. The root $[\frac{-1}{0}, \frac{1}{0}]$ has the empty label. Its left and right children $[\frac{-1}{0}, \frac{0}{1}]$ and $[\frac{0}{1}, \frac{1}{0}]$ are labelled l and r respectively. The left child of l, $[\frac{-1}{0}, \frac{-1}{1}]$, is labelled lL. The right child of lL, $[\frac{-2}{1}, \frac{-1}{1}]$, is labelled lLR, and so on. Only the first letter of a label is in small case because the division of the root is special.

We use $l\alpha$ or $r\alpha$ to denote the labels of Stern-Brocot intervals other than the root, with α being a possibly empty sequence of Ls and Rs. The sequence obtained by changing α's Ls to Rs and Rs to Ls is denoted $\bar{\alpha}$. For example, the reflection of the positive interval $r\alpha$ about 0 is the negative interval $l\bar{\alpha}$. The length of α is denoted by $|\alpha|$. We take the depth of $l\alpha$ or $r\alpha$ to be $1+|\alpha|$.

Lemmas 2.2 and 2.3 express the maps $m \rightarrow 1/m$ and $m \rightarrow \pm 1 + m$ succinctly for Stern-Brocot intervals. They allow us to reduce the invariance requirement (2.1) for Stern-Brocot intervals to an infinite system of linear equations (see (2.2)). That reduction is the first step in constructing ν_f.

Lemma 2.2. *The image of the interval* $[a/b, c/d]$ *under the map* $m \to 1/m$ *— the image is* $[d/c, b/a]$ *if* 0 *is not an interior point — is given by the following rules for Stern-Brocot intervals:*

$$l\alpha \to l\bar{\alpha}, \quad r\alpha \to r\bar{\alpha}.$$

Proof. We give the proof for intervals of type $r\alpha$ using induction on the depth of $r\alpha$ in the Stern-Brocot tree. The proof for intervals $l\alpha$ is similar.

The base case $r \to r$ is true because $m \in [0, \infty]$ if and only if $1/m \in [0, \infty]$.

For the inductive case, note that $[\frac{a}{b}, \frac{c}{d}]$, its left child $[\frac{a}{b}, \frac{a+c}{b+d}]$, and its right child $[\frac{a+c}{b+d}, \frac{c}{d}]$ are mapped by $m \to 1/m$ to $[\frac{d}{c}, \frac{b}{a}]$, its right child $[\frac{b+d}{a+c}, \frac{b}{a}]$, and its left child $[\frac{d}{c}, \frac{b+d}{a+c}]$ respectively. Therefore, if $r\alpha \to r\bar{\alpha}$ then $r\alpha L \to r\bar{\alpha}R$ and $r\alpha R \to r\bar{\alpha}L$. □

Unlike the inversion operation $m \to 1/m$ in the previous lemma, both the operations $m \to \pm 1 + m$ in the following lemma change the depth of Stern-Brocot intervals.

Lemma 2.3. *The image of Stern-Brocot intervals under the map* $m \to -1 + m$ *is given by the following rules:*

$$l\alpha \to lL\alpha, \quad rL\alpha \to lR\alpha, \quad rR\alpha \to r\alpha.$$

Similarly, the image of Stern-Brocot intervals under the map $m \to 1 + m$ *is given by the following rules:*

$$lL\alpha \to l\alpha, \quad lR\alpha \to rL\alpha, \quad r\alpha \to rR\alpha.$$

Proof. Similar to the previous proof. We will outline the proof for $m \to 1+m$ only.

The base cases, adding 1 to the intervals lL, lR and r, are easy to check.

For the induction, we note that $[\frac{a}{b}, \frac{c}{d}]$ is divided in the Stern-Brocot tree at the point $\frac{a+c}{b+d}$, and its map under $m \to 1+m$, $[1+\frac{a}{b}, 1+\frac{c}{d}]$, is divided in the Stern-Brocot tree at the point $1 + \frac{a+c}{b+d}$. Thus $[\frac{a}{b}, \frac{c}{d}]$, its left child, and its right child map to $[1 + \frac{a}{b}, 1 + \frac{c}{d}]$, its left child, and its right child respectively. □

By Lemma 2.3, subtraction and addition of 1 to intervals in the Stern-Brocot tree correspond to left and right rotation of the tree. Tree rotations are used to implement balanced trees in computer science [11].

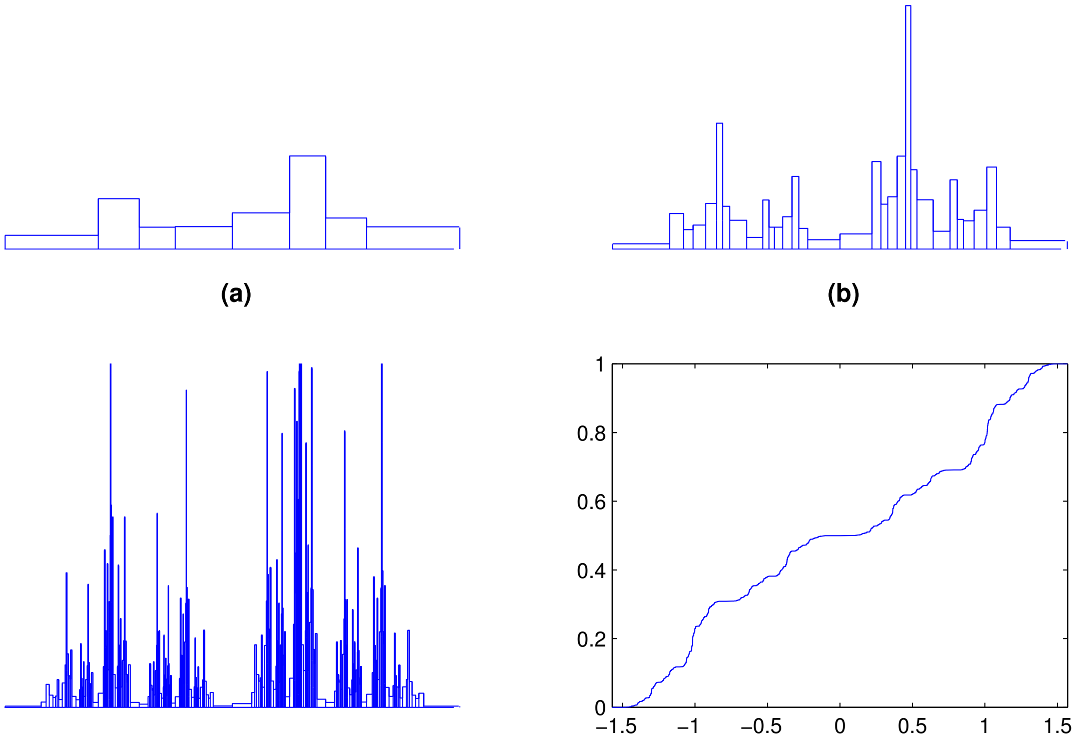
Thanks to Lemmas 2.2 and 2.3, the backward maps $m \to 1/(\pm 1 + m)$ can be performed on Stern-Brocot intervals easily. For example, $1/(1 + lLRL) = 1/lRL = lLR$. The invariance requirement (2.1) for Stern-Brocot intervals becomes an infinite set of linear equations for $\nu_f(I)$, I being any Stern-Brocot interval:

$$\begin{aligned}
\nu_f(l) &= \frac{1}{2}\nu_f(lR) + \frac{1}{2}(\nu_f(l) + \nu_f(rR)) \\
\nu_f(r) &= \frac{1}{2}(\nu_f(r) + \nu_f(lL)) + \frac{1}{2}\nu_f(rL) \\
\nu_f(lL\alpha) &= \frac{1}{2}\nu_f(l\overline{LL\alpha}) + \frac{1}{2}\nu_f(l\overline{\alpha}) \\
\nu_f(lR\alpha) &= \frac{1}{2}\nu_f(l\overline{LR\alpha}) + \frac{1}{2}\nu_f(r\overline{R\alpha}) \\
\nu_f(rL\alpha) &= \frac{1}{2}\nu_f(l\overline{R\alpha}) + \frac{1}{2}\nu_f(r\overline{RL\alpha}) \\
\nu_f(rR\alpha) &= \frac{1}{2}\nu_f(r\overline{\alpha}) + \frac{1}{2}\nu_f(r\overline{RR\alpha}).
\end{aligned} \tag{2.2}$$

We guessed the solution of (2.2). Even though the linear system (2.2) has only rational coefficients, its solution involves $\sqrt{5}$, an irrational number! Let $g = (1 + \sqrt{5})/2$. Since ν_f is a probability measure, we require that $\nu_f([-\infty, \infty]) = 1$. The solution is:

$$\begin{aligned}
\nu_f(r) &= 1/2 \\
\nu_f(r\alpha L) &= \frac{1}{1+g}\nu_f(r\alpha) \quad \text{if } |\alpha| \text{ is even} \\
&= \frac{g}{1+g}\nu_f(r\alpha) \quad \text{if } |\alpha| \text{ is odd} \\
\nu_f(r\alpha R) &= \frac{g}{1+g}\nu_f(r\alpha) \quad \text{if } |\alpha| \text{ is even} \\
&= \frac{1}{1+g}\nu_f(r\alpha) \quad \text{if } |\alpha| \text{ is odd} \\
\nu_f(l\alpha) &= \nu_f(r\overline{\alpha}).
\end{aligned} \tag{2.3}$$

For example, $\nu_f(r) = 1/2$, $\nu_f(rL) = (1+g)^{-1}/2$, $\nu_f(rLL) = g(1+g)^{-2}/2$. Since $\nu_f(l\alpha) = \nu_f(r\bar{\alpha})$ by (2.3), the measure ν_f is symmetric about 0. The same features of ν_f repeat at finer and finer scales. See Figure 4.



(a)

(b)

(c)

(d)

Figure 4: (a), (b), (c) show the measure ν_f over directions in R^2. In these figures, the interval $[0, \infty]$ is divided into 2^3, 2^5, and 2^8 Stern-Brocot intervals of the same depth, and then slopes are converted to angles in the interval $[0, \pi/2]$. The area above an interval gives its measure under ν_f. Because of symmetry, ν_f in the directions $[-\pi/2, 0]$ can be obtained by reflecting (a), (b) and (c). Some of the spikes in (c) were cut off because they were too tall. (d) is the cumulative density function for ν_f with directions parameterized using angles.

Theorem 2.4. *The measure ν_f defined by* (2.3) *satisfies the invariance requirement* (2.1) *for every Stern-Brocot interval. Further, with directions parameterized by slopes, ν_f defined by* (2.3) *gives the unique μ_f-invariant probability measure over directions in the real plane R^2.*

Proof. To show that ν_f is μ_f-invariant, it is enough to show that ν_f satisfies the invariance conditions (2.2) for Stern-Brocot intervals. The reason is – ν_f is obviously a continuous measure, every rational appears in the Stern-Brocot tree at a finite depth by Lemma 2.1c, and the rationals are dense in R. For the uniqueness of ν_f, see [6, p. 31].

It is enough to prove the invariance condition for positive intervals $r\alpha$. The validity of the invariance condition for negative Stern-Brocot intervals follows from symmetry. Assume the invariance condition for the interval $rL\alpha$:

$$\nu_f(rL\alpha) = \frac{1}{2}\nu_f(l\overline{R\alpha}) + \frac{1}{2}\nu_f(r\overline{RL\alpha}).$$

Then the invariance condition for $rL\alpha L$,

$$\nu_f(rL\alpha L) = \frac{1}{2}\nu_f(l\overline{R\alpha}R) + \frac{1}{2}\nu_f(r\overline{RL\alpha}R),$$

is also true, because the three fractions

$$\frac{\nu_f(rL\alpha L)}{\nu_f(rL\alpha)}, \quad \frac{\nu_f(l\overline{R\alpha}R)}{\nu_f(l\overline{R\alpha})}, \quad \frac{\nu_f(r\overline{RL\alpha}R)}{\nu_f(r\overline{RL\alpha})},$$

are all either $g/(1+g)$ or $1/(1+g)$ according as $|\alpha|$ is even or odd. By a similar argument, if the invariance condition (2.2) holds for all positive Stern-Brocot intervals at depth $d \geq 2$, then the invariance condition holds for all positive Stern-Brocot intervals at depth $d+1$.

Therefore, it suffices to verify (2.2) for r, rL, and rR. For r, (2.2) requires,

$$\frac{1}{2} = \frac{1}{2}(\frac{1}{2} + \frac{1}{2(1+g)}) + \frac{1}{2}(\frac{g}{2(1+g)}),$$

which is obviously true. For rL, (2.2) requires,

$$\frac{1}{2(1+g)} = \frac{g}{4(1+g)} + \frac{1}{4(1+g)^2},$$

which is true because $g = (1+\sqrt{5})/2$. The invariance condition for rR can be verified similarly. Thus the invariance condition (2.2) holds for all Stern-Brocot intervals, and we can say that ν_f is the unique μ_f-invariant probability measure. □

Because of symmetry, the measure ν_f over slopes given by (2.3) is invariant even for the distribution that picks one of $\left(\begin{smallmatrix}0 & 1\\ \pm1 & \pm1\end{smallmatrix}\right)$ with probability 1/4. Moreover, Furstenberg's integral for the Lyapunov exponent γ of this distribution is also given by (1.6).

For some distributions supported on 2-dimensional matrices with non-negative entries, the infinite linear system analogous to (2.2) is triangular, or in other words, the invariance requirement for a Stern-Brocot interval involves only intervals at a lesser depth. For a typical example, choose $\left(\begin{smallmatrix}1 & 1\\ 1 & 0\end{smallmatrix}\right)$ with probability p, $0 < p < 1$, and $\left(\begin{smallmatrix}0 & 1\\ 1 & 1\end{smallmatrix}\right)$ with probability $1-p$. In this example, the invariant measure over directions parameterized by slopes is supported on $[0,\infty]$, the slope m is mapped to $1/(1+m)$ and $1+1/m$ respectively, and the ranges of those two maps ($[0,1]$ and $[1,\infty]$) are disjoint. Chassaing, Letac and Mora [9] have found the invariant measure for several 2-dimensional random matrix products that fit into this framework. All their matrices have non-negative entries. Moreover, since the linear systems for finding the invariant measure are triangular for all the examples in [9], the solution can have irrational numbers only if the original problem does.

According to historical remarks in [9], measures similar to ν_f have been studied by Denjoy, Minkowski, and de Rham. But is ν_f a fractal? To make this precise, we need the definition

$$\dim(\nu_f) = \inf\{\dim(S) \big| \nu_f \text{ is supported on } S\},$$

where $\dim(S)$ is the Hausdorff dimension of $S \subset R$. To show that ν_f is a fractal, it is necessary to prove that $0 < \dim(\nu_f) < 1$. While it is known that $0 < \dim(\nu_f)$ [6, p. 162], a proof that $\dim(\nu_f) < 1$ or that ν_f is singular with respect to the Lebesgue measure does not seem to be available though those statements are almost surely true. However, the Hausdorff dimensions of very similar measures have been determined by Kinney and Pitcher [23]. We also note Ledrappier's conjecture that $\dim(\nu_f) < 1$ [25] [6, p. 162].

3 $e^{\gamma_f} = 1.13198824\ldots$

Furstenberg's integral for γ_f (1.6) can be written as

$$\gamma_f = 2\int_0^\infty \frac{1}{4}\log\Big(\frac{1+4m^4}{(1+m^2)^2}\Big)\,d\nu_f(m)$$

because both the integrand and ν_f are symmetric about 0. In this section, we use this formula to compute γ_f with the help of a computer. Thus the determination of e^{γ_f} to be $1.13198824\ldots$ is computer assisted. We will explain later why we report this result as a theorem (Theorem 3.2), even though it is computer assisted.

Let I_j^d, $1 \leq j \leq d$, be the 2^d positive Stern-Brocot intervals at depth $d+1$. Then,

$$p_d = 2\sum_{j=1}^{2^d} \min_{m\in I_j^d} \mathrm{amp}(m)\,\nu_f(I_j^d) < \gamma_f < q_d = 2\sum_{j=1}^{2^d} \max_{m\in I_j^d} \mathrm{amp}(m)\,\nu_f(I_j^d). \quad (3.1)$$

The inequalities above are strict because $\mathrm{amp}(m)$ is not constant, and ν_f is continuous. Also, (3.1) defines p_d and q_d. Since γ_f is trapped in the intervals (p_d, q_d), and the interval length $|q_d - p_d|$ shrinks to 0 as d increases, we can find γ_f to any desired accuracy by computing p_d and q_d for large enough d.

We computed p_d and q_d with $d = 28$ on a computer using IEEE double precision arithmetic (the C program used is described in the appendix). Computations in floating point arithmetic are not exact, but when done carefully, give an answer that is close to the exact answer. If $\mathrm{fl}(e)$ denotes the number obtained by evaluating the expression e in floating point arithmetic, $\mathrm{fl}(e)$ depends both on the type of floating point arithmetic used and the algorithm used to evaluate e. Our computations using IEEE double precision arithmetic [21] and an algorithm described in the appendix gave

$$\mathrm{fl}(p_{28}) = 0.1239755981508, \quad \mathrm{fl}(q_{28}) = 0.1239755994406. \quad (3.2)$$

In hexadecimal code, the 64 bits of $\mathrm{fl}(p_d)$ and $\mathrm{fl}(q_d)$ in IEEE double precision format are *3fbfbcdd638f4d87* and *3fbfbcdd6919756d* respectively. The appendix will explain the way to reproduce our computation to get exactly these two numbers. We will now upper bound the errors $|\mathrm{fl}(p_{28}) - p_{28}|$ and $|\mathrm{fl}(q_{28}) - q_{28}|$ to realize our aim of obtaining bounds for γ_f from (3.2).

IEEE double precision arithmetic (defined by the standard IEEE-754 [21]) can represent all real numbers of binary form $(-1)^s b_0.b_1 \dots b_{52}\, 2^{e-1023}$ exactly. Here, $b_0 = 1$, the bits b_1 to b_{52} can be 1 or 0, the sign bit s can be 1 or 0, and the biased exponent e can be any integer in the range $0 < e < 2047$. The number 0 can also be represented exactly. In fact, the values $e = 0$ and $e = 2047$ are used to implement special features that we do not describe. From here on, floating point arithmetic always refers to IEEE double precision arithmetic, and floating point number refers to a number in that arithmetic. Thus if a is a real number in the range $[2^{-1022}, (1 + 2^{-1} + \cdots + 2^{-52})2^{1023}]$, a can be represented such that $\mathrm{fl}(a) = a(1 + E)$ with the relative error E satisfying $|E| < 2^{-52}$ [19, p. 42].

The IEEE standard treats $+, -, \times, \div, \surd$ as basic operations. The basic operations cannot always be performed exactly. For example, the sum of two floating point numbers may not have an exact floating point representation. However, all these basic operations are performed as if an intermediate result correct to infinite precision is coerced into a representable number by rounding. We assume the "round to nearest" mode which is the default type of rounding. Thus if a and b are floating point numbers,

$$\begin{aligned} \mathrm{fl}(a+b) &= (a+b)(1+E) \\ \mathrm{fl}(a-b) &= (a-b)(1+E) \\ \mathrm{fl}(a/b) &= (a/b)(1+E) \\ \mathrm{fl}(a \times b) &= (a \times b)(1+E) \\ \mathrm{fl}(\sqrt{a}) &= (\sqrt{a})(1+E), \end{aligned} \tag{3.3}$$

where the relative error E may depend upon a, b and the operation performed, but $|E| < 2^{-52}$. For convenience, we denote 2^{-52} by u [1]. For (3.3) to be valid, however, the operation should not overflow and produce a number that is too big to be represented, or underflow and produce a number that is too small to be represented.

The C program we give in the appendix uses a function `tlog(x)` to compute $\log x$. This becomes necessary because log is not a basic operation in the IEEE standard. However, `tlog()` is implemented so that

$$\mathrm{fl}(\log a) = \log a(1+E) \tag{3.4}$$

[1] The bounds on $|E|$ can be taken as 2^{-53} [19, p. 42], but with the current choice the relative error of Tang's log function (see (3.4)) has the same bound as that of the basic operations.

with $|E| < u$ whenever a is a positive floating point number. For the clever ideas that go into `tlog()` and the error analysis, see the original paper by Tang [33].

The proof of Lemma 3.1 is given in the appendix.

Lemma 3.1. *Assume that* (3.3) *and* (3.4) *hold with* $0 < u < 1/10$ *for the floating point arithmetic used. Then for the algorithm to compute the sums* p_d *and* q_d *described in the appendix,*

$$\begin{aligned} |\text{fl}(p_d) - p_d| &< \frac{\log 4}{4}(e^{u(d+1)} - 1) + \frac{33}{4}ue^{u(d+1)}, \\ |\text{fl}(q_d) - q_d| &< \frac{\log 4}{4}(e^{u(d+1)} - 1) + \frac{33}{4}ue^{u(d+1)}. \end{aligned}$$

In the theorem below, by $1.13198824\ldots$ we mean a number in the interval $[1.13198824, 1.13198825)$.

Theorem 3.2. (a) *The constant* γ_f *lies in the interval*

$$(0.1239755980, 0.1239755995).$$

(b) $e^{\gamma_f} = 1.13198824\ldots.$

(c) *As* $n \to \infty$*,*

$$\sqrt[n]{|t_n|} \to 1.13198824\ldots$$

with probability 1.

Proof. In the computation leading to $\text{fl}(p_{28})$ and $\text{fl}(q_{28})$, there are no overflows or underflows, and hence, (3.3) and (3.4) are always true. Therefore, we can use $u = 2^{-52}$ and $d = 28$ in Lemma 3.1 to get

$$|\text{fl}(p_{28}) - p_{28}| < 10^{-14}, \ \ |\text{fl}(q_{28}) - q_{28}| < 10^{-14}.$$

Now the values of $\text{fl}(p_{28})$ and $\text{fl}(q_{28})$ in (3.2) imply (a). (b) is implied by (a). In fact, we can also say that the digit of e^{γ_f} after the last 4 in (b) must be an 8 or a 9. (c) follows from earlier remarks. □

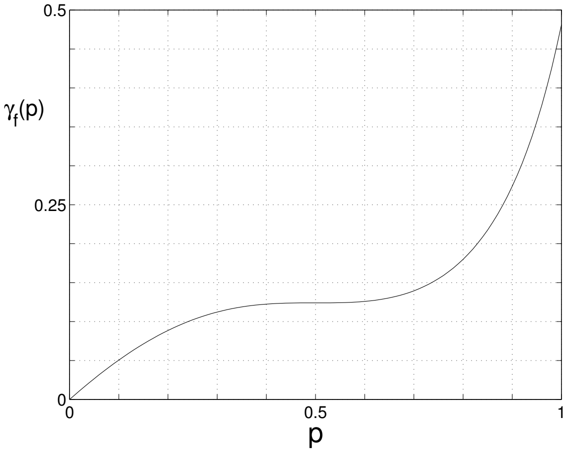


Figure 5: The Lyapunov exponent $\gamma_f(p)$ vs. p. $\gamma_f(p)$ is determined by numerically approximating the correct invariant distribution for the given p. For a description of the numerical method, sometimes called Ulam's method, see [20].

Theorem 3.2 above is the main result of this paper. We arrived at Theorem 3.2 using Lemma 3.1 and rounding error analysis. An alternative is to use interval arithmetic to validate the computation [1]. Instead of rounding the computations to the nearest floating point number, interval arithmetic carefully rounds the various stages of the computation either upwards or downwards to compute a lower bound for p_d and an upper bound for q_d. As a result, were we to use interval arithmetic there would be no need for rounding error analysis. A disadvantage would be that the manipulation of rounding modes necessary for implementing interval arithmetic would make it significantly more expensive on most computers. Our approach exposes the ideas behind floating point arithmetic and shows that floating point arithmetic is rigorous too. Besides, the rounding error analysis as summarized by Lemma 3.1 gives a clear idea of the error due to rounding. This tells us, for example, that the rounding errors $|\mathrm{fl}(p_{28}) - p_{28}|$ and $|\mathrm{fl}(q_{28}) - q_{28}|$, which are both less than 10^{-14}, are much smaller than the discretization error $|p_{28} - q_{28}|$, which is about 10^{-8}.

Since the proof of Theorem 3.2 relies on a computer calculation, the validity of the proof requires some comment. The construction of ν_f in Section 2, the program and the rounding error analysis given in the appendix can all be checked line by line. However, Theorem 3.2 still assumes the correct implementation of various software and hardware components including the standard IEEE–754. We did the computation on two entirely different systems — SUN's Sparc server 670 MP, and Intel's i686 with the Linux operating system. In both cases, the results were exactly the same as given in (3.2); the hex codes for $\mathrm{fl}(p_d)$ and $\mathrm{fl}(q_d)$ matched the hex codes given below (3.2). As it is very unlikely that two systems with such different architectures may have the same bug, we feel that the correctness of Theorem 3.2 should, at worst, be doubted no more than that of tedious and intricate proofs that can be checked line by line. Though the use of floating point arithmetic to prove a theorem may be unusual, the proof of Theorem 3.2 is only as depedent on the correctness of the computer system as, say, the proof of the four-color theorem; in other words, assuming the implementation of IEEE arithmetic to be correct is just like assuming the implementation of a memory-to-register copy instruction to be correct.

Besides, all components of a computer system, like mathematical proofs, can be checked in careful line by line detail, and this is done many times during and after their implementation. However, experience has shown that some bugs can defy even the most careful scrutiny. A great deal of research has gone into developing systems to verify that hardware and software implementations meet their specification [10]. But those systems are still not mature enough to be used in situations such as ours.

To conclude, we ask: Is there a short analytic description of γ_f? The fractal quality of γ_f suggests no. But let $\gamma_f(p)$ be the Lyapunov exponent of the obvious generalization $t_1 = t_2 = 1$, and for $n \geq 2$, $t_n = \pm t_{n-1} \pm t_{n-2}$ with each $\pm$ sign independent and either $+$ with probability p or $-$ with probability $1-p$. Unfortunately, the techniques described in this paper for $\gamma_f(1/2)$ do not seem to generalize easily to $\gamma_f(p)$, $0 < p < 1$. A beautiful result of Peres [29] implies that $\gamma_f(p)$ is a real analytic function of p. See Figure 5. The analyticity of $\gamma_f(p)$ vs. p seems to increase the possibility that there might be a short analytic description of γ_f.

A Appendix : Rounding Error Analysis

The main steps in the computation of p_d and q_d are the computation of $\nu_f(I_j^d)$, where I_j^d, $1 \leq j \leq 2^d$, are the 2^d positive Stern-Brocot intervals of depth $d+1$; the minimization and maximization of $\mathrm{amp}(m)$ over I_j^d; and the summation over $1 \le j \le 2^d$ as in the defining equation (3.1). We describe some aspects of the computation and then give a rounding error analysis to prove Lemma 3.1. A C program for computing p_d and q_d for $d = 28$ is given at the end of this section so that our computation can be reproduced; its perusal in not necessary for reading this section.

Lemma 2.2 implies that the denominators of the 2^d positive Stern-Brocot intervals of depth $d+1$ occur in an order that is the reverse of the order of the numerators. For example, the positive Stern-Brocot intervals of depth 4 are defined by divisions at the points $\frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1}, \frac{3}{2}, \frac{2}{1}, \frac{3}{1}, \frac{1}{0}$, the numerators for that depth occur in the order $0, 1, 1, 2, 1, 3, 2, 3, 1$, and the denominators occur in the reverse order $1, 3, 2, 3, 1, 2, 1, 1, 0$. We use this fact to avoid storing the denominators of the Stern-Brocot divisions. The numerators are stored in the array `num[]` by the C program.

To compute p_d and q_d, we use (3.1) in the following form:

$$\begin{aligned} p_d &= \sum_{j=1}^{2^d} \min_{m \in I_j^d} \Bigl(\log \frac{1+4m^4}{(1+m^2)^2} \Bigr) \frac{\nu_f(I_j^d)}{2}, \\ q_d &= \sum_{j=1}^{2^d} \max_{m \in I_j^d} \Bigl(\log \frac{1+4m^4}{(1+m^2)^2} \Bigr) \frac{\nu_f(I_j^d)}{2}. \end{aligned} \tag{A.1}$$

By (2.3), $\nu_f(I_j^d)/2$ is one of the $d+1$ numbers $g^{d-i}(1+g)^{-d}/4$, $0 \le i \le d$, where $g = (1+\sqrt{5})/2$. The array `table[]` in the C program is initialized after precomputing these $d+1$ numbers to very high accuracy in the symbolic algebra system *Mathematica* so that `table[i]`$= (g^{d-i}(1+g)^{-d}/4)(1+E)$ with the relative error E satisfying $|E| < u$. The index i into `table[]` for getting $\nu_f(I_j^d)/2$ is obtained by taking the binary representation of j, flipping all the odd bits if d is even and all the even bits if d is odd with the least significant bit taken as an even bit, and then counting the number of 1s; correctness of this procedure can be proved easily using induction.

The minimization or the maximization of $4\,\mathrm{amp}(m)$ over I_j^d in (A.1) are easy to do. Since $\mathrm{amp}(m)$ has its only local minimum for $m \ge 0$ at $m = 1/2$ (see Figure on page 6), both the minimum and the maximum are at the endpoints of I_j^d.

The summations in (A.1) are performed pairwise, not left to right. The pairwise summation of 2^d numbers is done by dividing the 2^d numbers into 2^{d-1} pairs of adjacent numbers, adding each pair to get 2^{d-1} numbers, and then reducing the 2^{d-1} numbers to 2^{d-2} numbers similarly, and so on until a single number is obtained. Rounding error analysis leads to smaller upper bounds on $|\mathrm{fl}(p_d) - p_d|$ and $|\mathrm{fl}(q_d) - q_d|$ for pairwise summation than for term-by-term left to right summation [19, p. 92]. The bounds for left to right summation are not small enough to give e^{γ_f} correctly to the 8 decimal digits shown in Theorem 3.2.

Lemmas A.1 and A.2 help simplify the proof of Lemma 3.1.

Lemma A.1. *Assume* $0 < f_1(u) < 1 + e_1 < g_1(u)$ *and* $0 < f_2(u) < 1 + e_2 < g_2(u)$.

(a) *If* $a > 0$, $b > 0$, *and* $a(1 + e_1) + b(1 + e_2) = (a + b)(1 + E)$, *then* $\min(f_1(u), f_2(u)) < 1 + E < max(g_1(u), g_2(u))$.

(b) *If* $1 + E = (1 + e_1)(1 + e_2)$, *then* $f_1(u)f_2(u) < 1 + E < g_1(u)g_2(u)$.

(c) *If* $1 + E = (1 + e_1)/(1 + e_2)$, *then* $f_1(u)/g_2(u) < 1 + E < g_1(u)/f_2(u)$.

Proof. To prove (a), note that $1 + E$ is the weighted mean of $1 + e_1$ and $1 + e_2$. (b) and (c) are trivial. □

Consider the computation $\mathrm{fl}(m^2)$:

$$\mathrm{fl}(m^2) = \mathrm{fl}(m)\,\mathrm{fl}(m)(1 + e') = m^2(1 + e')(1 + e'')^2,$$

where e'' is the relative error in representing m, and e' is the relative error caused by rounding the multiplication. By (3.3) and remarks in the paragraph preceding it, $1 - u < 1 + e', 1 + e'' < 1 + u$. Lemma A.1b allows us to gather the factors $1 + e'$ and $(1 + e'')^2$ together and write

$$\mathrm{fl}(m^2) = m^2(1 + e_0), \tag{A.2}$$

with $(1 - u)^3 < 1 + E < (1 + u)^3$.

Consider the computation $\mathrm{fl}(1 + m^2)$:

$$\mathrm{fl}(1 + m^2) = (1 + \mathrm{fl}(m^2))(1 + e''') = (1 + m^2(1 + e')(1 + e'')^2)(1 + e'''),$$

where e''' is the relative error in the addition $1 + m^2$, and e'', e' are, as before, the relative errors in representing m and the multiplication $m \times m$ respectively. As it was with $1 + e'$ and $1 + e''$, $1 - u < 1 + e''' < 1 + u$ by (3.3), and we can use Lemma A.1a to pull $(1+e')(1+e'')^2$ out of the sum $1+m^2$, and Lemma A.1b to multiply $(1+e')(1+e'')^2(1+e''')$ to get

$$\mathrm{fl}(1+m^2) = (1+m^2)(1+e_0'), \tag{A.3}$$

with $(1-u)^4 < 1+e_0' < (1+u)^4$.

Thus Lemma A.1 allows us to pull factors like $(1+e_i)$ out of sums (Lemma A.1a), or to multiply them together (Lemma A.1b), or to divide between them (Lemma A.1c). Rounding error analyses of simple computations, like the analyses of $\mathrm{fl}(m^2)$ and $\mathrm{fl}(1+m^2)$ given above, feature three steps. First, relative errors e_i caused by rounding are assigned to all the basic operations. Second, $1+e_i$ are bounded using (3.3) or (3.4). Third, factors like $(1+e_i)$ are gathered together using Lemma A.1. In the proof of Lemma 3.1, we always spell out the first step in detail, but sometimes omit details for the second and third steps.

The inequalities in Lemma A.2 below are used in the proof of Lemma 3.1.

Lemma A.2. (a) *If* $0 < u < 1/4$, $\log \frac{1+u}{1-u} < 3u$.

(b) $(1+\alpha)^d < e^{\alpha d}$ *for* $\alpha > 0$ *and* d *a positive integer.*

Proof. It is easy to prove (a) by expanding $\log((1+u)/(1-u))$ in a series. (b) can be proved by comparing the binomial expansion of $(1+\alpha)^d$ with the series expansion of $e^{\alpha d}$. □

The summations in the proof below are all over $1 \le j \le 2^d$.

Proof of Lemma 3.1. We will prove the upper bound only for $|\mathrm{fl}(p_d) - p_d|$. The proof for $|\mathrm{fl}(q_d) - q_d|$ is similar.

Firstly, consider the computation of $4\,\mathrm{amp}(m) = \log \frac{1+4m^4}{(1+m^2)^2}$:

$$\mathrm{fl}\Big(\log \frac{1+4m^4}{(1+m^2)^2}\Big) = \log\Big(\frac{(1+4m^4(1+e_0)^2(1+e_1)(1+e_2))(1+e_3)}{(1+m^2)^2(1+e_0')^2(1+e_4)}(1+e_5)\Big)$$
$$(1+e_6),$$

where e_0 and e_0' are the relative errors in $\mathrm{fl}(m^2)$ and $\mathrm{fl}(1+m^2)$ as in (A.2) and (A.3) respectively, e_1, e_2 are the relative errors of the two multiplications $(4\times m^2)\times m^2$, e_3 of the addition $1+4m^4$, e_4 of the multiplication $(1+m^2)\times(1+m^2)$, e_5 of the division $(1+4m^4)/(1+m^2)^2$, and e_6 of taking the log. By assumptions (3.3) and (3.4), $1-u < 1+e_i < 1+u$ for $1 \le i \le 6$. Lemma A.1 gives

$$\mathrm{fl}\Big(\log\frac{1+4m^4}{(1+m^2)^2}\Big) = \Big(\log\frac{1+4m^4}{(1+m^2)^2}\Big)(1+E_1)+E_2, \tag{A.4}$$

with $1-u < 1+E_1 < 1+u$ and $|E_2| < (1+u)\log((1+u)^{10}(1-u)^{-9})$. A weaker, but simpler, bound is $|E_2| < 10(1+u)\log((1+u)/(1-u))$. Now, the assumption $u < 1/10$ implies $10(1+u) < 11$, which together with Lemma A.2b, gives the simple bound $|E_2| < 33u$.

Secondly, recall that $\nu_f(I_j^d)/2$ is obtained by precomputing $g^{d-i}(1+g)^{-d}/4$ to high precision. Therefore,

$$\mathrm{fl}(\nu_f(I_j^d)/2) = \frac{\nu_f(I_j^d)}{2}(1+E_3), \tag{A.5}$$

with $|E_3| < u$.

Finally, consider the pairwise summation to compute p_d. Let m_j be the endpoint of I_j^d where $\mathrm{amp}(m)$ is minimum. Then,

$$\mathrm{fl}(p_d) = \sum\Big(\log\frac{1+4m_j^4}{(1+m_j^2)^2}(1+E_1^j)+E_2^j\Big)\Big(\frac{\nu_f(I_j^d)}{2}(1+E_3^j)\Big)(1+E_4^j)$$

where E_1^j and E_2^j are the relative errors in computing $\log((1+4m_j^4)(1+m_j^2)^{-2})$, and therefore, are bounded like E_1 and E_2 in (A.4); E_3^j is the relative error in computing $\nu_f(I_j^d)/2$ and is bounded like E_3 in (A.5); and the factors $1+E_4^j$ take up the errors in the pairwise summation. By Higham [19, p. 91], E_4^j can be chosen so that $(1-u)^d < 1+E_4^j < (1+u)^d$. Lemma A.1 gives

$$\mathrm{fl}(p_d) = \frac{1}{2}\sum\log\frac{1+4m_j^4}{(1+m_j^2)^2}\nu_f(I_j^d)(1+E_a^j)+\frac{1}{2}\sum\nu_f(I_j^d)E_b^j \tag{A.6}$$

with $(1-u)^{d+2} < 1+E_a^j < (1+u)^{d+2}$ and $|E_b^j| < 33u(1+u)^{d+1}$.

Bounding $|\mathrm{fl}(p_d)-p_d|$ is now a simple matter:

$$\begin{aligned}|\mathrm{fl}(p_d)-p_d| &< \frac{1}{2}\sum\Big|\log\frac{1+4m_j^4}{(1+m_j^2)^2}\Big|\nu_f(I_j^d)|E_a^j-1|+\frac{1}{2}\sum\nu_f(I_j^d)|E_b^j| \\ &< \frac{\log 4}{4}((1+u)^{d+2}-1)+\frac{33}{4}u(1+u)^{d+1} \\ &< \frac{\log 4}{4}(e^{u(d+2)}-1)+\frac{33}{4}ue^{u(d+1)}.\end{aligned}$$

The second inequality above uses $\sum \nu_f(I_j^d) = 1/2$, $|\log \frac{1+4m^4}{(1+m^2)^2}| < \log 4$, $|E_a^j - 1| < (1+u)^{d+2} - 1$, and $|E_b^j| < 33u(1+u)^{d+1}$. The bound on $|E_a^j - 1|$ can be derived easily from $(1-u)^{d+2} < 1 + E_a^j < (1+u)^{d+2}$. The final inequality follows from Lemma A.2b. □

Upper bounding $|\mathrm{fl}(q_d) - q_d|$ involves a small, additional detail. For the rightmost positive Stern-Brocot interval I_j^d, $\mathrm{amp}(m)$ is maximum at $m = \infty$. This causes no difficulty, however, because $\log((1+4m^4)/(1+m^2)^2)$ is taken as $\log 4$ at $m = \infty$ by the computation, and as a result, the bounds in (A.4) still hold.

A program to compute p_d and q_d is given on page 27 so that the computation leading to (3.2) can be easily reproduced. The program uses up 1.1 gigabytes of memory. It can be written using only a small amount of memory, but then it would be slower. For finding logs, we used the version of Tang's algorithm [33] that does not precompute and store $1/F$ for $F = 1 + j2^{-7}$, $0 \leq j \leq 128$. Though we do not give the code here because it is machine dependent, the guidelines given in [33] are enough to reproduce that log function (called `tlog()` in the program) exactly.

```c
#include <stdlib.h>
#include <stdio.h>
#define D 28
#define N 268435456
#define NRT 16384
unsigned int filter = 0xAAAAAAAA;


#define bitcount(x,b)   \
{b = 0;                 \
 for( ; x!=0; x&=(x-1)) \
   b++;                 \
}
double tlog(double);
double sum(double *, int);

static double table[D+1] = {
  3.51792099313013395856e-7,
  2.17419474349120812252e-7,
  1.34372624963892583604e-7,
  8.30468493852282286483e-8,
  5.13257755786643549553e-8,
  3.17210738065638736930e-8,
  1.96047017721004812623e-8,
  1.21163720344633924307e-8,
  7.48832973763708883155e-9,
  4.62804229682630359918e-9,
  2.86028744081078523237e-9,
  1.76775485601551836682e-9,
  1.09253258479526686555e-9,
  6.75222271220251501272e-10,
  4.17310313575015364275e-10,
  2.57911957645236136997e-10,
  1.59398355929779227278e-10,
  9.85136017154569097184e-11,
  6.08847542143223175599e-11,
  3.76288475011345921584e-11,
  2.32559067131877254014e-11,
  1.43729407879468667570e-11,
  8.88296592524085864439e-12,
  5.48997486270600811265e-12,
  3.39299106253485053174e-12,
  2.09698380017115758091e-12,
  1.29600726236369295083e-12,
  8.00976537807464630088e-13,
  4.95030724556228320737e-13};




void main()
{
 int i,j,*num;
 double lower,upper,larray1[NRT],larray2[NRT],
                    uarray1[NRT],uarray2[NRT];
 unsigned int *lptr, *uptr;

 num = (int *)malloc(sizeof(int)*(N+1));
 num[0] = 1; num[1]=1;
 for(i=2;i<N;i=i+2){
   num[i] = num[i/2];
   num[i+1] = num[i/2]+num[i/2+1];}
 num[N] = 1;

 for(i=0; i<NRT; i++){
   unsigned int k,b,x; double m, m2, m2p1,
                     left, right, measure;

   k = i*NRT; m =(double)num[k]/(double)num[N-k];
   m2 = m*m; m2p1 = m2+ 1.0;
   left = tlog((1+4*m2*m2)/(m2p1*m2p1));

   if (i < NRT/4)
     for(j=0; j<NRT; j++){
       k = i*NRT+j;
       m = (double)num[k+1]/(double)num[N-k-1];
       m2 = m*m;
       m2p1 = 1 + m2;
       right = tlog((1+4*m2*m2)/(m2p1*m2p1));
       x = k^filter;
       bitcount(x,b);
       measure = table[b];
       larray1[j] = measure*right; uarray1[j] = measure*left;
       left = right;}
   else if(i < NRT-1)
     for(j=0;j<NRT;j++){
       k = i*NRT+j;
       m = (double)num[k+1]/(double)num[N-k-1];
       m2 = m*m;
       m2p1 = 1 + m2;
       right = tlog((1+4*m2*m2)/(m2p1*m2p1));
       x = k^filter;
       bitcount(x,b);
       measure = table[b];
       larray1[j] = measure*left; uarray1[j] = measure*right;
       left = right;}
   else /* i == NRT-1 */
     for(j=0; j<NRT;j++){
       k = i*NRT+j;
       if(j==NRT-1)
         right = tlog(4.0);
       else{
         m = (double)num[k+1]/(double)num[N-k-1];
         m2 = m*m;
         m2p1 = 1 + m2;
         right = tlog((1+4*m2*m2)/(m2p1*m2p1));}
       x = k^filter;
       bitcount(x,b);
       measure = table[b];
       larray1[j] = measure*left; uarray1[j] = measure*right;
       left = right;}
   larray2[i] = sum(larray1,NRT); uarray2[i] = sum(uarray1,NRT);}

   lower = sum(larray2,NRT);
   upper = sum(uarray2,NRT);

   lptr = (unsigned int *)(&lower);
   uptr = (unsigned int *)(&upper);
   printf("(l,r)= (%.17E, %.17E)\n",lower, upper);
   printf("(l,u) in hex = (%x %x, %x %x)\n",*lptr,*(lptr+1),*uptr, *(uptr+1));
}


/* sums a list, length being a power of 2 */
double sum(double *list, int length)
{
 int i,step;


 for(step = 1; step < length; step = 2*step)
   for(i=0; i < length; i += 2*step)
     list[i]+= list[i+step];

 return list[0];
}
```

B Acknowledgements

First of all, my thanks to Prof. Nick Trefethen. Much help, guidance, and prodding from him at every stage made this paper possible. I thank P. Diaconis, R. Strichartz, H. Kesten, J. Hubbard, D. Knuth, T. Liggett, G. Strang, A. Stuart, J. Hickey, S. Ravikumar, S. Strogatz, and D. Kozen for discussions and pointers to the literature. I am indebted to Jason Hickey who not only let me use his Linux machine, but coaxed it into accepting my memory-hogging program.

References

[1] G. Alefeld and J. Herzberger, *Introduction to Interval Computations*, Academic Press, New York, 1983.

[2] L. Appel and W. Haken, The solution of the four-color map problem, *Scientific American 237(4)* (oct. 1977), 108-121.

[3] L. Arnold and H. Crauel, Random dynamical systems, in *Lyapunov Exponents*, eds. L. Arnold, H. Crauel, J.-P. Eckmann, Lecture Notes in Math. 1486, Springer-Verlag, Berlin, 1991, 1-22.

[4] R. Bellman, Limit theorem for non-commutative operations, *I. Duke Math. J. 21* (1954), 491-500.

[5] M.A. Berger, *An Introduction to Probability and Stochastic Processes*, Springer-Verlag, Berlin, 1993.

[6] P. Bougerol and J. Lacroix, *Random Products of Matrices with Applications to Infinite-dimensional Schrödinger Operators*, Birkhauser, Basel, 1984.

[7] L. Breiman, *Probability*, SIAM, Philadelphia, 1992.

[8] A. Brocot, Calcul des rouages par approximation nouvelle méthode, *Revue Chronometrique 6* (1860), 186-194.

[9] P. Chassaing, G. Letac and M. Mora, Brocot sequences and random walks on $SL_2(R)$, in *Probability measures on groups 7*, ed. H. Heyer, Lecture Notes in Math. 1064, Springer-Verlag, Berlin, 1984, 36-48.

[10] E. Clarke and J. Wing, Formal methods: state of the art and future directions, *ACM Computing Surveys 28(4)* (1996), 626-643.

[11] T.H. Cormen, C.E. Leiserson and R.L. Rivest, *Introduction to Algorithms*, MIT press, Cambridge, Massachusetts, (1990).

[12] A. Crisanti, G. Paladin and A. Vulpiani, *Products of Random Matrices in Statistical Physics*, Springer-Verlag, Berlin, 1992.

[13] P. Diaconis and M. Shahshahani, Products of random matrices and computer image generation, in *Random Matrices and their Applications*, eds. J.E. Cohen, H. Kesten and C.M. Newman, American Mathematical Society, Providence, 1986, 173-182.

[14] K. Falconer, *Fractal Geometry, Mathematical Foundations and Applications*, John Wiley and Sons, New York, 1990.

[15] H. Furstenberg, Non-commuting random products, *Trans. Amer. Math. Soc. 108* (1963), 377-428.

[16] H. Furstenberg and H. Kesten, Products of random matrices, *Ann. Math. Stat. 31* (1960), 457-469.

[17] R. Graham, D. Knuth and O. Patashnik, *Concrete Mathematics*, Addison-Wesley, Reading, Massachusetts, 1994.

[18] Y. Guivarc'h and A. Raugi, Frontière de Furstenberg, propiétés de contraction et théoèmes de convergence, *Zeit. fur Wahrsch. und Verw. Gebiete. 69* (1985), 187-242.

[19] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[20] F.Y. Hunt and W.M. Miller, On the approximation of invariant measures, *J. Stat. Phys. 66(1/2)* (1992), 535-548.

[21] *IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Standard 754-1985*, Institute of Electrical and Electronics Engineers, New York, 1985. Reprinted in *SIGPLAN Notices 22(2)* (1987), 9-25.

[22] J.F.C. Kingman, Subadditive ergodic theory, *Ann. Prob. 1* (1973), 883-909.

[23] J.R. Kinney and T.S. Pitcher, The dimension of some sets defined in terms of f-expansions, *Zeit. für Wahr. und Verw. Gebeite 4* (1966), 293-315.

[24] V. Krishnamurthy, The four color theorem, Appendix IV in F. Harary, *Graph Theory*, Indian student edition, Narosa/Addison-Wesley, New Delhi, 1988.

[25] F. Ledrappier, Quelques propriétés des exposants caractéristiques, in *Ecole d'été de Saint-Flour 12-1982*, ed P.L. Hennequin, Lecture Notes in Math. 1097, Springer-Verlag, Berlin, 1984.

[26] R. Lima and M Rahibe, Exact Lyapunov exponent for infinite products of random matrices, *J Phys. A: Math. Gen. 27* (1994), 3427-3437.

[27] K. Mischaikow and M. Mrozek, Chaos in the Lorenz equations: a computer assisted proof, *Bull. Amer. Math. Soc. (N.S.) 33* (1995), 66-72.

[28] V.I. Osseledac, A multiplicative ergodic theorem, *Trans. Moscow Math. Soc. 19* (1968), 197-231.

[29] Y. Peres, Analytic dependence of Lyapunov exponents on transition probabilities, in *Lyapunov Exponents*, ed L. Arnold, Lecture Notes in Math. 1486, Springer-Verlag, Berlin, 1986, 64-80.

[30] M.A. Stern, Ueber eine zahlentheoretische Funktion, *J. fur die reine und angewandte Mathematik 55* (1858), 193-220.

[31] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Cambridge, Massachusetts, 1996.

[32] R.S. Strichartz, A. Taylor and T. Zhang, Densities of self-similar measures on the line, *Experimental Mathematics 4(2)* (1995), 101-128

[33] P.T.P. Tang, Table-driven implementation of the logarithm function for IEEE floating-point arithmetic, *ACM Trans. Math. Soft. 16(4)* (1990), 378-400.

[34] S. Tuljapurkar, Demographic applications of random matrix products, in *Random Matrices and their Applications*, eds. J.E. Cohen, H. Kesten and C.M. Newman, American Mathematical Society, Providence, 1986, 319-326.

[35] D. Viswanath and L.N. Trefethen, Condition Numbers of Random Triangular Matrices, *SIAM J. Matrix Anal. Applics.*, to appear.