Report no. 00/01

Computing Lyapunov constants for random recurrences with smooth coefficients

Thomas G. Wright and Lloyd N. Trefethen
Oxford University
Numerical Analysis Group

In recent years, there has been much interest in the growth and decay rates (Lyapunov constants) of solutions to random recurrences such as the random Fibonacci sequence $x_{n+1} = \pm x_n \pm x_{n-1}$. Many of these problems involve non-smooth dynamics (nondifferentiable invariant measures), making computations hard. Here, however, we consider recurrences with smooth random coefficients and smooth invariant measures. By computing discretised invariant measures and applying Richardson extrapolation, we can compute Lyapunov constants to ten digits of accuracy. In particular, solutions to the recurrence $x_{n+1} = x_n + c_{n+1}x_{n-1}$, where the $\{c_n\}$ are independent standard normal variables, increase exponentially (almost surely) at the asymptotic rate $(1.0574735537\ldots)^n$. Solutions to the related recurrences $x_{n+1} = c_{n+1}x_n + x_{n-1}$ and $x_{n+1} = c_{n+1}x_n + d_{n+1}x_{n-1}$ (where the $\{d_n\}$ are also independent standard normal variables) increase (decrease) at the rates $(1.1149200917\ldots)^n$ and $(0.9949018837\ldots)^n$ respectively.

Key words and phrases: Fibonacci sequence; Lyapunov constant; Markov chain; random recurrence; Richardson extrapolation

Oxford University Computing Laboratory
Numerical Analysis Group
Wolfson Building
Parks Road
Oxford, England OX1 3QD
E-mail: tgw@comlab.oxford.ac.uk

February, 2000

Contents

1 Introduction

There is a literature of problems involving random recurrence relations dating as far back as the 1950s, but until recently there have been no quantitative investigations into their growth rates. The first such recent investigation was performed by Viswanath [5], who considered the recurrence

$$x_{n+1} = \pm x_n + x_{n-1}, \qquad x_0 = 1,\ x_1 = 1, \tag{1.1}$$

where the $\pm$ signs are chosen independently with equal probabilities. Initial estimates of the growth rate can be made by estimating the gradient of a log-plot of the terms from a sample of the recurrence against the iterate number, or estimating the limit of $|x_n|^{1/n}$ as $n \to \infty$. However, these methods only yield a few decimal places of accuracy before the computing time becomes prohibitive.

To improve the results Viswanath analysed the recurrence as a Markov chain. The recurrence can be written in the matrix form

$$\begin{pmatrix} x_n \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & \pm 1 \end{pmatrix} \begin{pmatrix} x_{n-1} \\ x_n \end{pmatrix}, \qquad \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

and hence as a product of random matrices:

$$\begin{pmatrix} x_n \\ x_{n+1} \end{pmatrix} = M_n M_{n-1} \cdots M_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad M_i = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \text{ or } \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix}.$$

What made Viswanath's results particularly appealing was an elegant proof of the accuracy of his calculated result—that the recurrence (1.1) grows exponentially at the rate of $(1.13198824\ldots)^n$ in the sense that $|x_n|^{1/n}$ converges almost surely (i.e., with probability 1) to the number $1.13198824\ldots$ as $n \to \infty$.

Embree and Trefethen [3] looked at the recurrence formed by including a fixed parameter β in front of one of the terms:

$$x_{n+1} = x_n \pm \beta x_{n-1}, \qquad x_0 = 1,\ x_1 = 1. \tag{1.2}$$

They showed (computationally) that by varying β it is possible to get not only different rates of growth, but also exponential decay (for $0 < \beta < \beta^*$, $\beta^* \in (0.702582, 0.702585)$). What is more, the dependence on this parameter is not smooth, but fractal.

1.1 Markov chain analysis

In order to treat a random recurrence efficiently as a Markov chain, one must first normalise the two-dimensional space of states $(x_n, x_{n+1})^T$ to one dimension. This can be done by looking at the angle between consecutive pairs of terms of the recurrence ($\bar{x} = (\cos\theta, \sin\theta)^T, \theta \in [-\pi/2, \pi/2]$), or equivalently the slope ($\bar{x} = (1, m)^T, m = \tan\theta$). One must now determine the proportion (density) of time the recurrence spends in the state corresponding to each particular angle θ (or slope m), $p(\theta)$, which should be

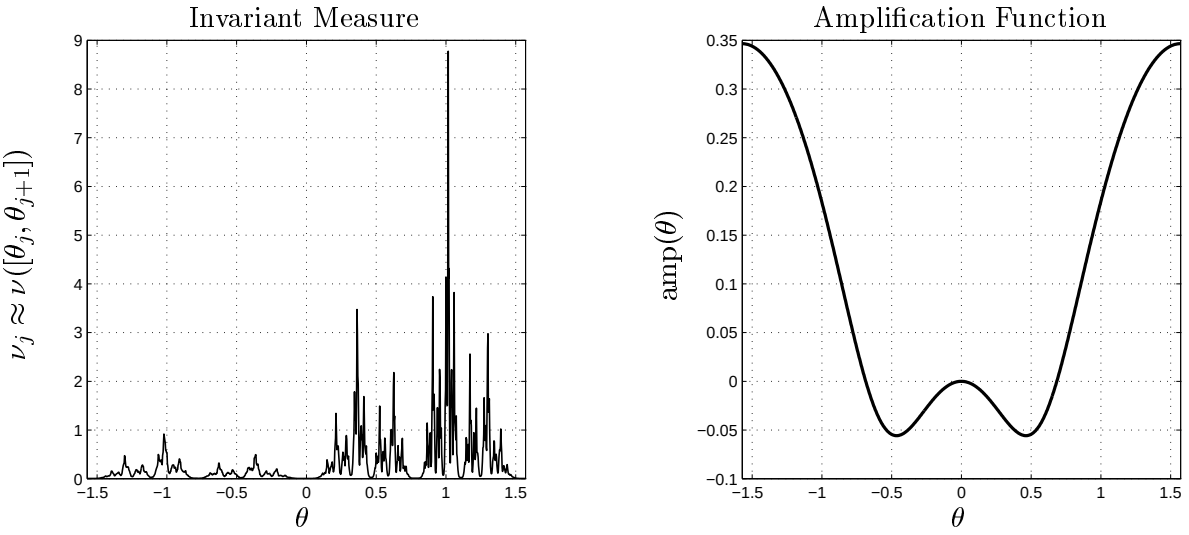

Figure 1: Discrete approximations to the invariant measure and amplification function for (1.1) using $N = 1024$ points. The invariant measure is not smooth; this histogram does not have a limit as $N \to \infty$.

invariant with the steps of the recurrence. To calculate this density, one needs in general to look for an invariant measure ν that corresponds to the integral of $p(\theta)$, as p defined pointwise does not always exist. For an interval $I = [\theta_1, \theta_2]$, $\nu(I)$ is defined to be the proportion of time that the Markov process spends in I as $n \to \infty$. This can then be used to calculate the Lyapunov constant through the following equation due to Furstenberg [1]

$$\log(\sigma) = \int \mathrm{amp}(\bar{x})\ \nu(\mathrm{d}\bar{x}),$$

where $\mathrm{amp}(\bar{x})$ is the average of the log of the amplification in the direction of the state corresponding to $\bar{x} = (1, m)^T$, and σ is the Lyapunov constant. The amplitude function arises because the normalisation to one dimension ignores the information about the size of the pair of terms being considered.

Looking at Figure 1, one can see that the invariant measure ν arising from (1.1) is far from smooth, and this is still the case for higher resolutions. This implies that in order to obtain accurate approximations to ν a very large system of equations is needed, with dimension of the order of millions. The associated matrix is fortunately very sparse, but the large dimension still limits the accuracy which can be obtained using this method.

2 Smooth Random Coefficients

As just indicated, the invariant measures resulting from the recurrences studied by Viswanath, Embree and Trefethen have all been fractal, implying that large systems of equations are needed to obtain reasonable approximations. However, if independent normally distributed coefficients $\{c_n\}$ are included in front of one of the terms of the

recurrence, the invariant measure becomes piecewise smooth (Figure 2), which means it can be accurately approximated using a much coarser mesh. The first equation we look at has a single normally distributed coefficient in front of the second term:

$$x_{n+1} = x_n + c_{n+1}x_{n-1}, \qquad x_0 = 1,\ x_1 = 1. \tag{2.1}$$

2.1 Calculation of the invariant measure

The numerical approximation of ν is now achieved using the following method. First, the domain $[-\frac{\pi}{2}, \frac{\pi}{2}]$ is split into sub-intervals $[e_j, e_{j+1}]$, and 'state j' is defined as the jth interval. Next we define a matrix A whose entries are as follows:

$$a_{i,j} = \mathrm{P}(\text{if started in state } i, \text{ one step of (2.1) goes to state } j),$$

where P denotes probability. While the matrices needed to solve (1.1) and (1.2) are very sparse, the matrix A here is dense. The discrete invariant measure ν is the limit of $x^T A^n$ as $n \to \infty$ for some non-negative row vector of initial probabilities x^T with sum 1. This is also the left eigenvector of A corresponding to the largest eigenvalue, $\lambda = 1$.

To make the computations simpler, all the angles from the discretisation of the domain $[-\frac{\pi}{2}, \frac{\pi}{2}]$ are converted to slopes (by taking the tangent) before any calculations are performed. To calculate the entries of A, consider the transformation matrix T which would be needed to take state i to state j. It is of the following form, where C is a normally distributed random variable, $\bar{x}_i = (1, m_i)^T$ represents state i and $m_j = (e_j + e_{j+1})/2$ (taking the midpoint of the interval as its representative):

$$T\bar{x}_i = \begin{pmatrix} 0 & 1 \\ C & 1 \end{pmatrix} \begin{pmatrix} 1 \\ m_i \end{pmatrix} = \begin{pmatrix} m_i \\ C + m_i \end{pmatrix} \equiv \begin{pmatrix} 1 \\ C/m_i + 1 \end{pmatrix} \text{(when normalised)}.$$

The question now is, what is the probability that C above takes a value which transforms $\bar{x}_i$ to the interval $I_j = [e_j, e_{j+1}]$ (which contains all states $\bar{x}_j = (1, m_j)^T$ with $e_j \le m_j \le e_{j+1}$)? The probability required is then the probability that C satisfies $e_j \le C/m_i + 1 \le e_{j+1}$ which is simply the probability that a suitably shifted and scaled normal random variable lies in the interval $[e_j, e_{j+1}]$:

$$a_{i,j} = \mathrm{P}\left(e_j \le \frac{C}{m_i} + 1 \le e_{j+1}\right).$$

The value of each of the entries of A can now easily be calculated using the normal distribution function and the invariant measure can be found by computing the left eigenvector via power iteration. As this only involves vector-matrix products with the matrix A, it is an efficient method for this case, especially for the larger matrices (convergence is achieved in fewer than 60 iterations in nearly all cases).

It is interesting to note the singularity in the invariant measure at $\theta = \pi/4$ (Figure 2), the angle corresponding to the condition $x_{n+1} = x_n$, i.e., $c_{n+1}x_{n-1} = 0$. This is due to the fact that $|c_{n+1}x_{n-1}| = O(\epsilon)$ with probability $O(\epsilon \log \epsilon)$, not $O(\epsilon)$, as $\epsilon \to 0$, as is easily explained by an exercise of calculus.

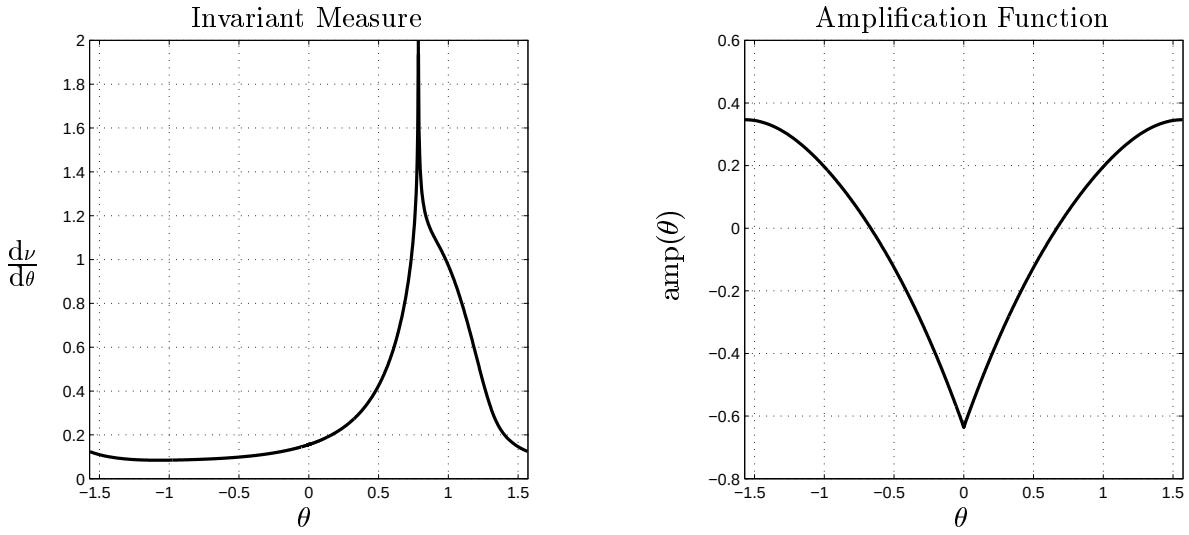

Figure 2: Invariant measure and amplification function for (2.1). (As in Figures 3 and 4, these curves are based on discrete approximations with $N = 1024$ points, but are believed to be correct to plotting accuracy for the limit $N \to \infty$.) The invariant measure is now smooth, apart from a singularity at the point $\theta = \pi/4$, where the density is infinite.

2.2 Calculation of the amplitude function and Furstenberg's integral

We have just dealt with the approximation of the invariant measure, but we still need to approximate the amplitude function before we can use Furstenberg's integral to calculate the Lyapunov constant. If we treat the normal distribution approximately as a discrete distribution (using the discretisation $[\tan(e_j), \tan(e_{j+1})]$ based on the discretisation $[e_j, e_{j+1}]$ for the invariant measure we have already) the amplification can be written as

$$\mathrm{amp}(\bar{x}_i) \approx \sum_{j=0}^{N} \mathrm{P}(\tan(e_j) \leq C \leq \tan(e_{j+1})) \log\left(\frac{\|T_j \bar{x}_i\|}{\|\bar{x}_i\|}\right),$$

where T_j is the transformation matrix associated with the interval $[e_j, e_{j+1}]$ and C a normally distributed random variable. (That is to say that the amplification associated with state i is approximately equal to the amplification which occurs on transformation to other states averaged over the probabilities of going to those other states.)

We take $T_j = \begin{pmatrix} 0 & 1 \\ \tan(m_j) & 1 \end{pmatrix}$, where m_j is the midpoint of the corresponding interval. However, the obvious choice $\bar{x}_i = (1, \tan(m_i))^T$ does not give the best results (convergence to the Lyapunov constant is only linear). Instead we first take $\bar{x}_i$ to be $(1, \tan(e_i))^T$, then $(1, \tan(e_{i+1}))^T$. Calculating Furstenberg's integral by summing rectangles, we evaluate the estimate for each of the strips $[e_j, e_{j+1}]$ using both of the above approximations for $\bar{x}_i$, summing the minimum values to get a lower bound for the integral and summing the maximum values to get an upper bound. Finally we take the

average of the lower and upper bounds as our estimate for the Lyapunov constant.

2.3 Calculating the Lyapunov constant

By calculating a sequence of approximations to the Lyapunov constant and halving the step size each time, it is possible to use Richardson extrapolation (see, e.g., [2]) to extrapolate the results and greatly improve the accuracy (Table 1). The calculation involved in the Richardson extrapolation itself is of negligible cost. Using the method outlined above we get quadratic convergence of our estimate to the Lyapunov constant, which the extrapolation then improves. Looking at the tableau indicates that this method gives the growth rate of (2.1) as 1.057473553704 to 12 decimal places. The calculation using a matrix of dimension 2^{12} takes under ten minutes on a Sun Ultra 5 workstation, making the total time to calculate the final results less than a quarter of an hour.

Table 1: Richardson extrapolation tableau for (2.1). The first column contains the calculated values for different mesh sizes ($h = 1/N$), and the extrapolated results become more accurate towards the bottom right. Apparently correct digits are underlined.

N	Calculated (h^2)	h^3	h^4	h^5
2^5	1.05816787788315			
2^6	1.05764737322614	1.05747387167380		
2^7	1.05751699793470	1.05747353950423	1.05747349205143	
2^8	1.05748441341861	1.05747355191325	1.05747355368596	1.05747355779493
2^9	1.05747626847468	1.05747355349337	1.05747355371910	1.05747355372131
2^{10}	1.05747423237837	1.05747355367960	1.05747355370620	1.05747355370534
2^{11}	1.05747372337080	1.05747355370161	1.05747355370476	1.05747355370466
2^{12}	1.05747359612089	1.05747355370426	1.05747355370463	1.05747355370463
	h^6	h^7	h^8	h^9
2^5				
2^6				
2^7				
2^8				
2^9	1.05747355358990			
2^{10}	1.05747355370483	1.05747355370665		
2^{11}	1.05747355370464	1.05747355370464	1.05747355370462	
2^{12}	1.05747355370462	1.05747355370462	1.05747355370462	1.05747355370462

3 Moving the Normal Coefficient

What happens if we make a small modification to the recurrence already considered, by placing the normal coefficient in front of the first term instead of the second? We now

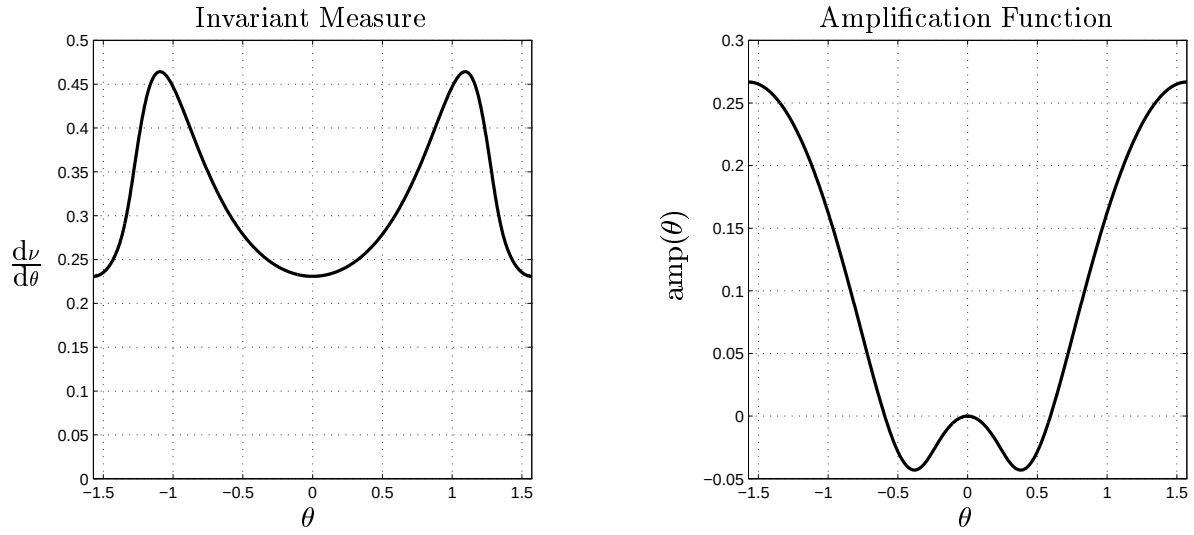

Figure 3: Invariant measure and amplification function for (3.1). Both are symmetric, a fact which could be used to make the algorithm for calculating the invariant measure more efficient.

have

$$x_{n+1} = c_{n+1}x_n + x_{n-1}, \qquad x_0 = 1,\ x_1 = 1. \tag{3.1}$$

In principle, we can use exactly the same method as we did for (2.1), the only differences being in the definition of the transformation matrices T_j (which are now of the form $\begin{pmatrix} 0 & 1 \\ 1 & C \end{pmatrix}$ where C is again a normally distributed random variable). However, splitting the domain $[-\pi/2, \pi/2]$ into equally sized intervals does not give the cleanest data for extrapolation; the data still converge quadratically, but the extrapolation does not improve the accuracy as well as it did for (2.1).

As the invariant measure is smooth about the origin (Figure 3), we tried using Chebyshev points (see, e.g., [4]) instead of evenly spaced points—making the intervals near the ends of the domain shorter than those in the centre. Unfortunately, this makes the situation worse, because the wide intervals near the middle of the domain contain a relatively large proportion of the probability density of the normal distribution. This means that large peaks can occur within rows of the matrix A, leading to oscillations at the ends of the computed approximation to the invariant measure.

To remedy this we used two sets of Chebyshev points, making intervals both at the ends of the domain and around the origin shorter than others. This yielded much more fruitful results after the extrapolation (Table 2), giving the growth rate of (3.1) as 1.11492009172 to 11 decimal places.

Table 2: Richardson extrapolation tableau for (3.1). Again the first column converges quadratically, although unevenly spaced points were needed to get the cleanest extrapolation.

N	Calculated (h^2)	h^3	h^4	h^5
2^5	1.11608599692250			
2^6	1.11521046593091	1.11491862226705		
2^7	1.11499256139770	1.11491992655329	1.11492011287990	
2^8	1.11493818203185	1.11492005557657	1.11492007400847	1.11492007141704
2^9	1.11492461117473	1.11492008755569	1.11492009212413	1.11492009333184
2^{10}	1.11492122124624	1.11492009127008	1.11492009180071	1.11492009177915
2^{11}	1.11492037406672	1.11492009167354	1.11492009173118	1.11492009172655
2^{12}	1.11492016230357	1.11492009171585	1.11492009172190	1.11492009172128
	h^6	h^7	h^8	h^9
2^5				
2^6				
2^7				
2^8				
2^9	1.11492009403877			
2^{10}	1.11492009172906	1.11492009169240		
2^{11}	1.11492009172485	1.11492009172478	1.11492009172504	
2^{12}	1.11492009172111	1.11492009172105	1.11492009172102	1.11492009172100

4 Including a Second Normal Coefficient

Until now, the running time of the algorithms considered has been quadratic in the number of intervals used in the discretisations, and all the results above can be calculated in a quarter of an hour on a Sun Ultra 5 workstation. The limiting factor is the amount of memory needed to store the full matrix A (128MB for $N = 2^{12}$).

We now look at another similar recurrence where the situation is slightly different:

$$x_{n+1} = c_{n+1}x_n + d_{n+1}x_{n-1}, \qquad x_0 = 1,\ x_1 = 1, \tag{4.1}$$

(c_n and d_n are both normal coefficients). The calculation of the invariant measure proceeds exactly as before, but now two discretisations are needed for the calculation of the amplitude function, one for each normal coefficient:

$$\mathrm{amp}(\bar{x}) = \sum_{j=0}^{N}\sum_{k=0}^{N} \mathrm{P}(e_j \le C \le e_{j+1})\mathrm{P}(e_k \le D \le e_{k+1}) \log\left(\frac{\|T_{j,k}\bar{x}\|}{\|\bar{x}\|}\right),$$

where C and D are normally distributed random variables.

This double sum means that the limiting factor is now time instead of memory as the running time of the algorithm is cubic in the number of intervals used in the

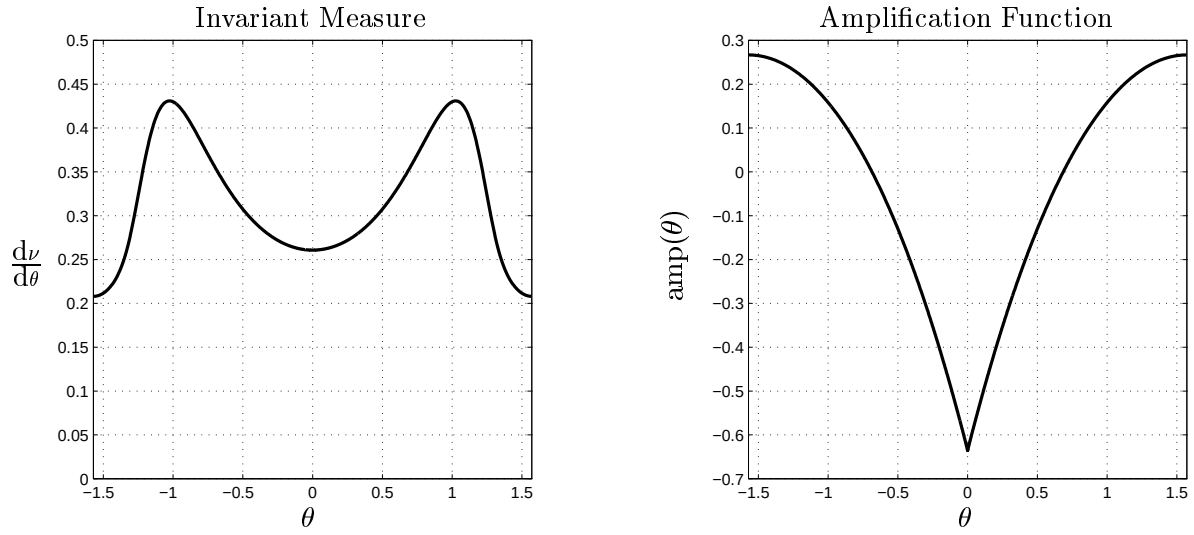

Figure 4: Invariant measure and amplification function for (4.1). As with (3.1), both functions are symmetric. This time the reason is obvious: each normally distributed coefficient in the equation is symmetric about the origin, which means that each term can take either sign with equal probability.

discretisation. (The amplification must be computed at N points, each of which takes $O(N^2)$ operations.) To get results to the same accuracy as the other recurrences now takes approximately 24 hours of computation. The invariant measure and amplification functions are shown in Figure 4 and Richardson extrapolation improves the accuracy in the same manner as before, giving Table 3. This shows the growth rate of the recurrence (4.1) to be 0.994901883717 to 12 decimal places. In other words, solutions to this recurrence decay very gradually as $n \to \infty$.

5 Comments on the Numerical Procedures

In order to obtain the results given here, we had to try several variations on the method—changing the spacing of the mesh points and using different representatives for the intervals that arise from the discretisation (for example using the midpoint or one of the endpoints)—before obtaining data that were clean enough to extrapolate well; these problems are not trivial. At the same time, we make no claim that our method is close to optimal; indeed it would probably be possible to devise methods with exponential convergence, and ones with algebraic convergence that permit extrapolation with h^2, h^4, h^6, ... rather than h^2, h^3, h^4, ...

6 Conclusion

By discretising the Markov chain and applying Richardson extrapolation, we have calculated the Lyapunov constants for the three basic random recurrences (2.1), (3.1) and

Table 3: Richardson extrapolation tableau for (4.1). Again the first column (calculated data) converges quadratically and the extrapolation is very effective.

N	Calculated (h^2)	h^3	h^4	h^5
2^5	0.99638571885308			
2^6	0.99527440306349	0.99490396446696		
2^7	0.99499519959550	0.99490213177284	0.99490186995939	
2^8	0.99492523939416	0.99490191932704	0.99490188897765	0.99490189024553
2^9	0.99490772619873	0.99490188846693	0.99490188405834	0.99490188373038
2^{10}	0.99490334479760	0.99490188433055	0.99490188373964	0.99490188371839
2^{11}	0.99490224904613	0.99490188379565	0.99490188371923	0.99490188371787
2^{12}	0.99490197505727	0.99490188372764	0.99490188371793	0.99490188371784
	h^6	h^7	h^8	h^9
2^5				
2^6				
2^7				
2^8				
2^9	0.99490188352022			
2^{10}	0.99490188371801	0.99490188372115		
2^{11}	0.99490188371785	0.99490188371785	0.99490188371783	
2^{12}	0.99490188371784	0.99490188371784	0.99490188371784	0.99490188371784

(4.1) with normal coefficients, apparently obtaining eleven or twelve digits of accuracy in each case. Similar methods should be applicable to other random recurrences with smooth coefficients (for example with coefficients drawn from the uniform distribution on $(-1, 1)$).

The use of extrapolation gives an additional 4 or 5 accurate digits over the accuracy of the data calculated. This is equivalent to using a matrix of size 2^{17} with the direct method, which would need approximately 130GB of memory and take 50 hours for (2.1) and (3.1), and would take about 50 years for (4.1).

References

[1] P. Bougerol and J. Lacroix, 1984. *Random Products of Matrices with Applications to Infinite-Dimensional Schrödinger Operators.* Birkhäuser, Basel.

[2] G. Dahlquist and A. Björck, 1974. *Numerical Methods.* Prentice–Hall, Englewood Cliffs, NJ.

[3] M. Embree and L.N. Trefethen, 1999. Growth and decay of random fibonacci sequences. *Proc. R. Soc. Lond. A*, **455**: 2471–2485.

[4] L.N. Trefethen, 2000. *Spectral Methods in* MATLAB. SIAM.

[5] D. Viswanath, to appear. Random fibonacci sequences and the number 1.13198824... *Math. Comp.*