Electronic Transactions on Numerical Analysis.
Volume 28, pp. 190-205, 2008.
Copyright © 2008, Kent State University.
ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

QUANTUM DYNAMICAL ENTROPY AND AN ALGORITHM BY GENE GOLUB[*]

GIORGIO MANTICA[†]

In memory of Gene Golub

Abstract. The problem of computing the quantum dynamical entropy introduced by Alicki and Fannes requires the trace of the operator function $F(\Omega) = -\Omega \log \Omega$, where Ω is a non-negative, Hermitean operator. Physical significance demands that this operator be a matrix of large order. We study its properties and we derive efficient algorithms to solve this problem, also implementable on parallel machines with distributed memory. We rely on a Lanczos technique for large matrix computations developed by Gene Golub.

Key words. Quantum dynamical entropy, large matrices, Lanczos method, Montecarlo techniques

AMS subject classifications. 65F10, 37M25, 81Q50

1. Introduction.

Fundamental in the theory of classical non-linear dynamics is the so-called Kolmogorov-Sinai (K-S.) entropy, a quantity that, when positive, implies chaos [4]. One can safely say that it is the single most important piece of information that one can get on a dynamical system, and surely it is the single most employed word in all dynamics. Various techniques for its computation exists, and its link to Lyapunov exponents via Pesin's formula [13] makes it an effective tool. In quantum mechanics, on the other hand, a plurality of quantities can rightly claim to correspond to K-S. entropy, for they tend to this latter in the classical limit. None of these, though, is simply computable, nor a quantum analogue of Pesin's relation exists at the present moment. These difficulties stem ultimately from the basic fact that trajectories are not defined for quantum systems. In this paper, I study a version of "quantum entropy" due to Alicki and Fannes [1] that is based on the notion of *coherent histories*. It requires the computation of a complex Hermitean, large, non–negative full matrix Ω and of the trace of the matrix function $-\Omega \log(\Omega)$. I develop computational schemes that render viable the numerical analysis of this quantity for systems of physical interest. In this endeavor, I rely on an algorithm developed by Bai, Fahey and Golub [5] to deal with large scale matrix computation problems.

The plan of this paper is the following: in the next section I introduce the matrix Ω under investigation, with a few words on its quantum mechanical origin that also help to understand the breadth and scope of its algorithmic requirements. Readers with deeper interests in dynamics will find reference to the original literature, while numerical analysts desiring to grasp the essence of the computational problem may just focus on the linear algebra nature of the equations.

In section 3 I study Ω and its symmetries. Then, in section 4, I derive a recursion relation for computing Ω at increasing values of an integer "time" M. This has been originally developed in [3]. A deeper analysis of its properties, performed in section 5, permits us to set up a parallel algorithm for the computation of Ω at different values of M. In section 6, this idea is implemented in two algorithms for the computation of the matrix–vector product ΩW. The first algorithm runs conveniently on parallel machines with distributed memory, the second minimizes the memory storage requirements to achieve the largest possible matrix size given the finite memory space available on any single computer. These algorithms are instrumental

[*] Received February 28, 2007. Accepted for publication January 15, 2008. Recommended by M. Gutknecht. Supported by PRIN 2005 *Transport properties of classical and quantum systems*. Computation time has been provided by CINECA under the INFM-CNISM grant *Open quantum systems: transport and decoherence*.

[†] Center for Non-linear and Complex Systems, I.N.F.N. and C.N.I.S.M., Università dell'Insubria, Via Valleggio 11, 22100 Como, Italy (giorgio@uninsubria.it).

in setting up a Lanczos-Montecarlo technique for the computation of the trace of $(-\Omega \log \Omega)$ due to Golub, as discussed in section 7. Numerical results are presented in section 8 and an improvement of Golub's confidence interval estimation is presented in section 9. Finally, a Lanczos-Montecarlo technique for the direct calculation of the Jacobi matrix associated with the spectral measure of Ω is introduced and tested in section 10. The need for a numerical treatment of A-F entropy arises particularly in the field of quantum chaos and decoherence: the Conclusions briefly mention this problem and summarize the work.

2. Quantum dynamical entropy. Quantum evolution takes place in the Hilbert space $\mathcal{H}$ associated with a physical system [14]. While in most circumstances this space is infinite dimensional, we shall assume a finite dimensional reduction, of dimension N, without dealing here with the reduction problem. Therefore, we shall let $\{e_n\}_{n=0,\dots,N-1}$ be the canonical basis of $\mathcal{H} = \mathbf{C}^N$ and $(\cdot,\cdot)_{\mathcal{H}}$ be the usual scalar product in this space. Quantum evolution in $\mathcal{H}$ is effected stroboscopically by a unitary operator U: the "wave-vector" $\psi \in \mathcal{H}$ specifying the state of the system at time t is mapped into $U\psi$, the state at time $t+1$.

Although no specification of the nature of U other than it can be numerically computed is necessary here and although we shall present quite general algorithms, for sake of illustration we shall use in this paper two paradigmatic examples. The first is the operator U with matrix elements

$$U_{kl} = \frac{1}{\sqrt{N}} e^{-(\pi i l^2/N)} e^{2\pi i k l/N}, \tag{2.1}$$

that corresponds to free classical motion on the one-dimensional torus $S_1 = \mathbf{R}/\mathbf{Z}$, a stable, completely integrable system.

The second example is the so-called *quantum cat* unitary evolution operator $U^{cat} = KU$, where U has been defined in eq. (2.1) and K is the operator with matrix elements

$$K_{kl} = \frac{1}{\sqrt{N}} e^{i\pi l^2/N} \delta_{k,l}, \tag{2.2}$$

where $\delta_{k,l}$ is the Kronecker delta. The operator U^{cat} is the quantum mechanical version of the renown *Arnol'd cat map* of classical mechanics [4], a chaotic systems with positive K-S. entropy. For a derivation of this operator, its physical relevance and mostly the relations with FFT; see [7].

Clearly, given an initial state ψ, quantum evolution yields the components $\psi_n(j) := (e_n, U^j \psi)_{\mathcal{H}}$ at any future (or past) time $j \in \mathbf{Z}$. According to standard usage, the probability that the quantum system is found in state n at time j is given by the square modulus of $\psi_n(j)$. As in classical dynamics, "coarse graining" measurement can be effected, when the state vector ψ is not analyzed in all its components, but only in groups of them. Formally, if $\{P_k\}_{k=0,\dots,L}$ is a family of projection operators, so that $I = \sum_k P_k$, we can also "measure" the projection of ψ on the range of P_k, that is, compute the scalar product $(\psi, P_k\psi)_{\mathcal{H}}$.

To make things easier, without renouncing anything essential, in this paper we shall consider two orthogonal projections P_0 and P_1, on half of the Hilbert space each, like in a head/tail experiment: take $N = 2p$ and let

$$P_k = \sum_{n=kp}^{(1+k)p-1} e_n e_n^T, \quad k = 0, 1. \tag{2.3}$$

Given these premises, a "quantum history" of a vector ψ is the result of effecting the unitary quantum evolution U preceded at each time by projection on either the head or the tail half of the Hilbert space: readers familiar with the double-slit experiment might think of the motion of a particle hitting at integer times a sequence of screens with two slits. According to common usage in symbolic dynamics, Greek letters denote sequence of symbols, like $\sigma = (\sigma_0, \sigma_1, \ldots, \sigma_{M-1})$. These latter are vectors of length M, and are also called "words" in symbolic dynamics. Greek letter with subscripts, like σ_i, are therefore the components of these vectors, i.e. symbols in the alphabet (in our choice, either zero or one). With this notation, the quantum history of the vector ψ is

$$\psi_\sigma = (UP_{\sigma_{M-1}}) \cdots (UP_{\sigma_0})\psi. \tag{2.4}$$

For convenience of notation we shall put

$$U^{\sigma_j} := UP_{\sigma_j}, \quad j = 0, \ldots, M-1. \tag{2.5}$$

The "amplitude" $(\psi_\sigma, \psi_\sigma)$ should be compared with the measure of the classical phase space with symbolic dynamics given by the "word" σ. In both classical and quantum dynamics these probabilities add up to one: $\sum_\sigma (\psi_\sigma, \psi_\sigma) = 1$. In quantum mechanics, though, interference reigns and the products $(\psi_\sigma, \psi_{\sigma'})$ are non-null also when $\sigma \neq \sigma'$.

Complexity of the motion is quantified in classical dynamics by certain summations over non-null amplitudes of sequences σ, of course averaged with respect to the initial conditions. In the Alicki–Fannes (A-F) quantum formulation [3], entropy is derived by the spectrum of the decoherence matrix D with entries $D_{\sigma,\sigma'}$, defined by

$$D_{\sigma,\sigma'} := \frac{1}{N} Tr(U^{\sigma_{M-1}\dagger} \ldots U^{\sigma_0\dagger} U^{\sigma'_0} \ldots U^{\sigma'_{M-1}}), \tag{2.6}$$

where the dagger indicates the adjoint and clearly $U^\dagger = U^{-1}$, $P_k^\dagger = P_k$. Observe that D is a $2^M \times 2^M$ square matrix, Hermitian, of unit-trace and non-negative, so that the quantum A-F entropy associated with the unitary evolution U and the family of projections $\{P_k\}$ can be defined as

$$S(U, \{P_k\}) = Tr(-D \log D). \tag{2.7}$$

Entropy is therefore the trace of the function of a matrix whose entries are themselves obtained by traces of product of operators over $\mathcal{H}$. In addition, notice that in dynamics one is interested in the large–time behavior (here, large M): it is then clear that computation of S via eqs. (2.6),(2.7) is a formidable task, of exponentially increasing computational complexity. Yet, the structure revealed by eq. (2.6) permits a reduction of size independent of M.

In fact, observe that the right hand side of (2.6) can be seen as a scalar product, in the space of square matrices of size N, between the vectors $V^\sigma := U^{\sigma_0} \cdots U^{\sigma_{M-1}}$ and $V^{\sigma'} := U^{\sigma'_0} \cdots U^{\sigma'_{M-1}}$. Then, the non-null eigenvalues in the spectrum of D coincide, with their multiplicities, with those of the operator Ω acting in $\mathcal{K}$ as:

$$\Omega := \frac{1}{N} \sum_\sigma V^\sigma (V^\sigma, \cdot)_{\mathcal{K}}, \tag{2.8}$$

where the scalar product is taken in the new space of matrices, $\mathcal{K}$. Full detail will be provided in the next section. Observe here that $\mathcal{K}$ has dimensionality N^2: therefore, Ω has maximal rank smaller than that of D whenever $M \geq 2 \log_2 N$, a condition that is easily realized. On the other hand, notice that a major physical problem requires the analysis of the "classical limit" of quantum mechanics, that in turn requires N also to be large [7]. We are really facing a challenging problem.

In this paper we study computational techniques to evaluate the A-F entropy

$$S(U,\{P_k\}) = Tr(-\Omega \log \Omega). \tag{2.9}$$

In the next section we define precisely the metric in $\mathcal{K}$ and we study Ω and its symmetries.

3. The matrix Ω.

In this section, we explore the symmetries of the matrix Ω. Recall first that Ω acts in $\mathcal{K}$, the space of complex square matrices of size N. $\mathcal{K}$ is endowed with the scalar product

$$(V,W)_{\mathcal{K}} := Tr(V^{\dagger}W). \tag{3.1}$$

Ω acts in $\mathcal{K}$ as

$$\Omega W = \frac{1}{N}\sum_{\sigma} V^{\sigma}(V^{\sigma},W)_{\mathcal{K}}, \;\; W \in \mathcal{K}, \tag{3.2}$$

where the summation ranges over all binary sequences of length M, $\sigma_j \in \{0,1\}$, for $j = 0,\ldots,M-1$. In this equation we have set

$$V^{\sigma} := U^{\sigma_{M-1}} \cdots U^{\sigma_0} := (UP_{\sigma_{M-1}})\ldots(UP_{\sigma_0}). \tag{3.3}$$

U is any unitary operator over $\mathbf{C}^N$ (the dynamics) and the projection operators P_i, $i = 0,1$, have been defined in eq. (2.3).

An orthonormal basis for $\mathcal{K}$ can be constructed as

$$E_{ij} := e_i e_j^T, \;\; i,j = 1,\ldots,N. \tag{3.4}$$

Let us compute the matrix elements of Ω in this basis:

$$\begin{aligned}\Omega_{j_1,j_2,k_1,k_2} := (E_{j_1,j_2},\Omega E_{k_1,k_2})_{\mathcal{K}} &= \frac{1}{N}\sum_{\sigma}(E_{j_1,j_2},V^{\sigma})_{\mathcal{K}}(V^{\sigma},E_{k_1,k_2})_{\mathcal{K}} \\ &= \frac{1}{N}\sum_{\sigma} Tr(E^{\dagger}_{j_1,j_2}V^{\sigma})\, Tr(V^{\sigma\dagger}E_{k_1,k_2})\,. \end{aligned} \tag{3.5}$$

Traces are then explicitly written as:

$$\begin{aligned} Tr(E^{\dagger}_{j_1,j_2}V^{\sigma}) &= \sum_i (e_i,e_{j_2})_{\mathcal{H}}(e_{j_1},V^{\sigma}e_i)_{\mathcal{H}} = (e_{j_1},V^{\sigma}e_{j_2})_{\mathcal{H}}, \\ Tr(V^{\sigma^{\dagger}}E_{k_1,k_2}) &= (e_{k_2},V^{\sigma\dagger}e_{k_1})_{\mathcal{H}} = (e_{k_1},V^{\sigma}e_{k_2})^*_{\mathcal{H}}, \end{aligned} \tag{3.6}$$

where scalar products in $\mathcal{H}$ appear and where the asterisk denotes complex conjugation. Therefore, the matrix elements of V^{σ} are featured in the final formula for Ω, that reads:

$$\Omega_{j_1,j_2,k_1,k_2} = \frac{1}{N}\sum_{\sigma}(e_{j_1},V^{\sigma}e_{j_2})_{\mathcal{H}}(e_{k_1},V^{\sigma}e_{k_2})^*_{\mathcal{H}} := \frac{1}{N}\sum_{\sigma} V^{\sigma}_{j_1 j_2}(V^{\sigma}_{k_1 k_2})^*. \tag{3.7}$$

Take now into account the product form of the operator V^{σ}, eq. (3.3) and notice that $V^{\sigma}e_{j_2}$ is null unless e_{j_2} belongs to the range of P_{σ_0}, that is, $j_2 \in I_{\sigma_0}$, the set of indices corresponding to the σ_0-th half of Hilbert space:

$$I_{\sigma_0} = [\sigma_0 p, (\sigma_0+1)p-1]. \tag{3.8}$$

For the same reason, k_2 must belong to the same set, and therefore the matrix Ω is the direct sum of two square matrices of maximal rank $p = N/2$, half of the original. We can therefore consider only one of these at a time, when computing the A-F entropy, eq. (2.9), that is additive over the two subspaces. To fix the notation, in the following we shall let implicitly $\sigma_0 = 0$, and $j_2, k_2 \in I_0$, the other case being trivially obtained. Also, with abuse of notation, but without danger of confusion, from now on $\mathcal{K}$ will denote the space spanned by $E_{i,j}$, $i = 0, \ldots, N-1$ and $j = 0, \ldots, p-1$ (recall that $N = 2p$).

Finally, inspection of eq. (3.7) also reveals the symmetry

$$\Omega_{j_1,j_2,k_1,k_2} = \Omega^*_{k_1,k_2,j_1,j_2}, \tag{3.9}$$

so that Ω is a Hermitian operator in $\mathcal{K}$.

4. Time dependence of Ω. We have noticed at the end of section 2 that the dimension of $\mathcal{K}$ does not depend on M, the “time”, that is, the length of the symbolic word σ. Yet, Ω obviously does, so that from now on we will indicate this dependence explicitly as a superscript: Ω^M will be the Ω matrix at “time” M. Now, summation over all words of length M in eq. (3.7) might lead us to believe that we are still facing an exponentially growing computational cost. For these reasons, it is important to examine in detail the “time” dependence of the problem.

Not to overburden the notation, since the scalar products of this section will all be in the space $\mathcal{H}$, we are allowed to drop the relative subscript. Let us start from the computation of $V^\sigma_{j_1j_2}$. The vector/word σ can be written as (σ', σ_{M-1}), where σ' is now a vector of length $M-1$. Accordingly, $V^\sigma = UP_{\sigma_{M-1}}V^{\sigma'}$. Inserting an identity into the definition of $V^\sigma_{j_1j_2}$ we get

$$(e_{j_1}, V^\sigma e_{j_2}) = \sum_{j_3} (e_{j_1}, UP_{\sigma_{M-1}} e_{j_3})(e_{j_3}, V^{\sigma'} e_{j_2}) = \sum_{j_3} U^{\sigma_{M-1}}_{j_1j_3} V^{\sigma'}_{j_3j_2}. \tag{4.1}$$

A quite similar equation holds for $(e_{k_1}, V^\sigma e_{k_2})$. Using these facts in eq. (3.7) we get

$$\begin{aligned} \Omega^M_{j_1,j_2,k_1,k_2} &= \frac{1}{N} \sum_{\sigma',\sigma_{M-1}} \sum_{j_3,k_3} U^{\sigma_{M-1}}_{j_1j_3} (U^{\sigma_{M-1}}_{k_1k_3})^* V^{\sigma'}_{j_3j_2} (V^{\sigma'}_{k_3k_2})^* \\ &= \sum_{\sigma_{M-1}} \sum_{j_3,k_3} U^{\sigma_{M-1}}_{j_1j_3} (U^{\sigma_{M-1}}_{k_1k_3})^* \Omega^{M-1}_{j_3,j_2,k_3,k_2}. \end{aligned} \tag{4.2}$$

Summations in this equation range from 1 to N for the indices j_3 and k_3, and on 0 and 1 for the variables σ_i. In intuitive terms, one can explain this formula in the words of a knowledgeable reviewer: the left-hand side is the overlap between two “world-lines” (or trajectories) of time-extent M, whose end-points are (j_1, j_2) and (k_1, k_2). It can be expressed as a sum over all possible ancestors at time $(M-1)$, each with their respective overlap and time-evolution.

Equation (4.2) is the basis of a recursive technique initialized by letting $M = 1$ in eq. (3.7),

$$\Omega^1_{j_1,j_2,k_1,k_2} = \frac{1}{N} \sum_{\sigma_0=0,1} V^{\sigma_0}_{j_1j_2} (V^{\sigma_0}_{k_1k_2})^*. \tag{4.3}$$

When implemented directly [3], this technique requires a storage of the order of $N^4/4$ complex quantities, while the computation of the A-F entropy

$$S(U, \{P_k\}, M) = Tr(-\Omega^M \log \Omega^M) \tag{4.4}$$

calls for the diagonalization of a square matrix of size $N^2/2$. Needless to say, it becomes soon impossible to meet these requirements as N grows. It is the purpose of the present paper to show how they can be significantly eased. In the next section we analyze the nature of the recurrence, in order to tackle the memory requirement first. We then face the entropy computation problem.

5. Partitioning the Ω^M matrix. It is now fundamental to examine in closer detail the nature of the recurrence in eq. (4.2). First of all, although somehow evocative, it does *not* mean that Ω^M is the M-th power of Ω^1, seen as an operator over $\mathcal{K}$. Nonetheless, eq. (4.2) implies a remarkable property. In fact, observe that the indices j_2 and k_2 appear unchanged at l.h.s. and r.h.s. Therefore, we define $\Omega^M(j_2, k_2)$ as the square matrix of size N, with indices j_1, k_1 and entries $\Omega^M_{j_1,j_2,k_1,k_2}$. What we have just observed can be formalized by saying that:

LEMMA 5.1 (Sectorization). *The sector matrix $\Omega^M(j_2, k_2)$ can be computed recursively according to eqs. (4.2),(4.3).*

This lemma is of paramount importance in the numerical implementation:

LEMMA 5.2 (Storage and Computational Complexity).

i) The sector matrix $\Omega^M(j_2, k_2)$ can be computed with $6MN^4$ floating point operations (f.p.o's). Its computation requires a storage of $3N^2$ complex entries (c.e's).

ii) The full matrix Ω^M can be computed with $3M(N^6/4 + N^5/2)$ f.p.o's and stored in an array of $N^4/8 + N^5/4$ c.e's. It can be computed sector by sector with a storage of $3N^2$ complex entries (c.e's).

Proof. Matrix iteration (4.2) requires $6N^2$ f.p.o's for each entry of $\Omega^M(j_2, k_2)$, so that the full (j_2, k_2) sector can be computed with $6N^4$ f.p.o's. Although there are $p^2 = N^2/4$ different sectors in Ω^M, the symmetry property, eq. (3.9), implies that only $p(p+1)/2 = N^2/8 + N/4$ of them are fully independent and can be obtained choosing $j_2 \geq k_2$. (Additional redundancy in the diagonal sectors, that is, those with $j_2 = k_2$ exists but will not be exploited, because it only provides sub-leading improvements in both computation time and storage size parameters).

The two matrices V^σ can be conjunctly stored in a square N by N matrix with complex entries. In fact, we have seen that $V^\sigma e_{j_2}$ is null unless $j_2 \in I_{\sigma_0}$. In addition, the matrix iteration (4.2) requires only two memory arrays of size N^2. □

REMARK 5.3. Notice that the previous lemmas unveil the possibility of parting the computation of Ω^M over parallel machines with distributed memory.

6. Computing the matrix-vector product $\Omega^M W$. Of particular relevance for the Lanczos technique that we shall outline in section 7 is the computation of the matrix-vector product $\Omega^M W$, where W is a vector in $\mathcal{K}$. When not necessary, in this section we shall drop the index M. The heart of our technique is an application of Lemma 5.1.

Algorithm 1: Computation of the (j_2, k_2) sector of the product $\overline{W} = \Omega W$.

```
• for $j_1 = 1, \ldots, N$
    – compute the sector product $\overline{W}_{j_1}(j_2, k_2) = \sum_{k_1} \Omega^M_{j_1,k_1}(j_2, k_2) W_{k_1,k_2}$
    – accumulate into the result vector $\overline{W}$: $\overline{W}_{j_1,j_2} \to \overline{W}_{j_1,j_2} + \overline{W}_{j_1}(j_2, k_2)$
• end
```

Algorithm 1 requires a storage of N^2 c.e's for W and $\overline{W}$, and N^2 c.e's for the sector of the matrix Ω^M. Computational complexity is $2N^2$ f.p.o's.

Algorithm 2: Computation of the matrix-vector product $\overline{W} = \Omega W$.

```
• for $j_2 = 1, \ldots, N/2$ and $k_2 = 1, \ldots, N/2$, $j_2 \leq k_2$
    – execute Algorithm 1 and the conjugate Algorithm 1' stemming from the symmetry property (3.9)
    – accumulate the results into the vector $\overline{W}_{j_1,j_2}$
• end
```

It is readily seen that if we want to use Algorithm 2 (a serial application of Algorithm 1) for the computation of the full matrix-vector product ΩW, the computational complexity is of $N^4/2 + N^3$ f.p.o's. In addition, we need to store the $N^2/8 + N/4$ independent sectors of Ω, action that requires a full storage of $N^4/8 + N^3/4$ c.e's: this is a serious memory limitation that may limit significantly the Hilbert dimension N that can be numerically simulated. We have devised two procedures to overcome this limitation.

Algorithm 3: Computation of the matrix-vector product $\overline{W} = \Omega W$ on parallel computers with distributed memory and P processors $\pi_1, \ldots, \pi_P$.

```
1. order the set of labels $(j_2, k_2)$ with $j_2 \geq k_2$, and distribute them among the $P$ processors as equally as possible
2. transmit the input vector $W$ to all processors
3. in each processor $\pi_l$
     • for each pair $(j_2, k_2)$ assigned to processor $\pi_l$
         – execute Algorithm 1 and the conjugate Algorithm 1' stemming from the symmetry property (3.9)
         – accumulate the results into the vector $\overline{W}^l_{j_1,j_2}$
     • end
4. accumulate the vectors $\overline{W}^l_{j_1,j_2}$ produced in all processors $\pi_l$, $l = 1, \ldots, P$, into the result vector $\overline{W}_{j_1,j_2}$.
```

Memory requirement of Algorithm 3 is $N^4/8P + N^3/4P + bN^2$ c.e's on each processor and computational complexity is $N^4/2 + N^3$ f.p.o's, that can be executed in a real time proportional to $(N^4/2 + N^3)/P$ seconds. We assume that the sectors of Ω^M had been previously computed and stored in each processor. Notice that processors communication—usually, a time-demanding operation—is limited to steps 2 and 4 and consists of the total transmission of N^2 c.e's. This is also the sole significant transmission also when computing the matrix Ω^M: Lemma 5.1 is actually the statement of a parallelization property.

On the other hand, with or without parallel computers, one can drastically diminish the memory requirement, at the expense of increasing computation time:

Algorithm 4: Computation of the matrix-vector product $\overline{W} = \Omega^M W$ with minimal storage requirement.

```
• for each label $(j_2, k_2)$ with $j_2 \geq k_2$
    – compute the sector matrix $\Omega^M(j_2, k_2)$
    – execute Algorithm 1 and the conjugate Algorithm 1' stemming from the symmetry property (3.9)
    – accumulate the results into the vector $\overline{W}_{j_1,j_2}$
• end
```

Algorithm 4 attains the minimal memory requirement of $6N^2$ c.e's. Computation time is augmented by the need of computing the matrix Ω^M, that brings the total computational complexity to grow like $\frac{3}{4}MN^6 + \frac{3}{2}MN^5$. This may become significant in the Lanczos algorithm that we shall describe in the next section.

7. Computation of the entropy: The algorithm of Golub et al.

Computation of the spectrum of Ω by full-matrix techniques is not viable as N grows. Yet, we are interested not so much in the spectrum *per se*, as in the entropy function (4.4). Therefore, in view of the results of sections 5 and 6, the Lanczos' technique developed by Golub et al. [5] becomes an interesting possibility. In this section, we sketch the details that permit the application of Golub's algorithms 1 and 2 of reference [5] without major modifications. We shall refer to them as G1 and G2.

In essence, G1 is based on the construction of the tridiagonal representation of Ω in the Krylov basis [9] associated with a randomly chosen initial vector $W_i \in \mathcal{K}$. For complex Ω, differing from Golub's case, we choose the entries of W in the set of complex numbers $\{\pm 1, \pm i\}$ with equal probability [6]. W_i is then conveniently normalized. Ω is Hermitian and its tridiagonal representation is real.

This tridiagonal matrix is then easily diagonalized, and the entropy

$$S_i := (W_i, -\Omega \log(\Omega) W_i)$$

is estimated via Gaussian summation. In addition, since the spectrum of Ω is contained in the set $[0, 1]$, Gauss-Radau formulae can also be employed. Since $f(x) = -x \log(x)$ is the integrand for the computation of the entropy, it is readily found that the derivatives of f satisfy the relations $f^{(2n)}(x) < 0$ and $f^{(2n+1)}(x) > 0$ for all $x > 0$ and $n \geq 1$, so that Gaussian summation and Gauss-Radau with a prescribed eigenvalue at $x = 1$ both provide upper bounds to the quantity $(W_i, f(\Omega)W_i)_{\mathcal{K}}$, while Gauss-Radau with prescribed eigenvalue at zero yields a lower bound. In the following, we shall indicate with S_i^l the lower bound obtained with the Gauss-Radau formula with prescribed eigenvalue at zero, and with S_i^u the upper bound obtained by the usual Gauss formula.

The Monte-Carlo algorithm G2 consists in taking a statistical average over a large number of realizations of the random vector W_i, $i = 1, \ldots, I$, of the values provided by the algorithm G1. The predicted value for S is then the mean of the average upper and lower bounds, and a confidence interval is derived on the basis of Hoeffding's inequality. We shall come back to this statistical estimate later.

We ran algorithm G2 endowed with algorithm 3 of the previous section for matrix-vector multiplication on a cluster of parallel computers with MPI communication protocol. The dimension of the Hilbert space was $N = 2^7$ (corresponding to Ω^M of size 2^{13}), M ranged from 1 to 30, the dimension of the Jacobi matrix was six (seven for Gauss-Radau) and the number of trial vectors, I, was 1000. In this paper, we show data obtained with 14 processors, also to underline the fact that our algorithm is not bound to work with a number of processors equal to a power of two. Figure 7.1 displays the time T_Ω (in real seconds) spent by each processor in the matrix computation part, eq. (4.2), in each iteration from M to $M + 1$ and the time T_L required by the Lanczos algorithm, again at each iteration.

8. The algorithm of Golub et al.: Results.

We can now start by showing results obtained for the quantum cat evolution U^{cat}. Figure 8.1 displays a sample of $I = 1000$ realizations of the algorithm G1 with $M = 11$ and $N = 2^7$ and six Gaussian points. Upper values S_i^u and lower values S_i^l are almost coincident for the same sample item (the largest difference $\delta := \max\{S_i^u - S_i^l, i = 1, \ldots, I\}$ is about $3.7\ 10^{-3}$), while different samples feature a much larger range, of amplitude 0.59917. In keeping with Golub's notation this value is $U_{max} - L_{min}$, where $U_{max} := \max\{S_i^u, i = 1, \ldots, I\}$ and $L_{min} := \min\{S_i^l, i = 1, \ldots, I\}$.

In Table 8.1 we report these data for M ranging from 1 to 14. According to algorithm G2, it is then possible to extract from this table statistical estimates of $S(U, \{P_k\}, M)$. Before doing that, though, we observe that δ is always several orders of magnitude smaller that $U_{max} - L_{min}$. Moreover, we want to further analyse the sample data S_i^u, or S_i^l for that matter.

9. Experimental statistical analysis and improved probabilistic bounds.

We have observed at the end of the previous section that the sample variations $U_{max} - L_{min}$ are orders of magnitude larger than the upper-lower bound differences and indeed we have reasons to believe this to be the general case. Then, it is not worth spending computer time to compute the Gauss-Radau formula: we shall now consider uniquely the Gaussian summation.

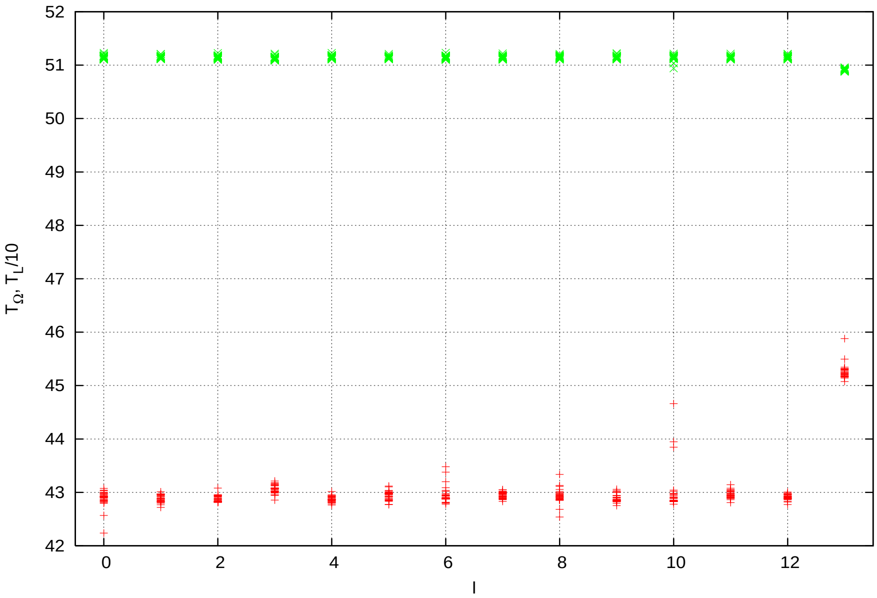

FIG. 7.1. *Computation time versus processor index l $(l = 0, \ldots, 13)$ of algorithm G2 and G3. Units are real seconds. T_Ω (red +'s) is the time required for updating the Ω^M matrix with $N = 2^7$, while T_L (green x's, value divided by ten for graphical convenience) is the time required by the Lanczos algorithm G1 repeated for $I = 1000$ random initial vectors W_i. All values for $M = 1, \ldots, 30$ are reported.*

TABLE 8.1

Statistical data extracted from $I = 1000$ samples of the quantum cat evolution with $N = 2^7$ and six Gaussian points. Symbols are defined in the text.

M	δ	U_{max}	L_{min}	$U_{max} - L_{min}$
1	0.365320e-03	0.128128e-01	0.364302e+01	0.363021e+01
2	0.925000e-04	0.674307e-01	0.347163e+01	0.340420e+01
3	0.522670e-03	0.297215e+00	0.315047e+01	0.285326e+01
4	0.107000e-04	0.806697e+00	0.360842e+01	0.280172e+01
5	0.960000e-05	0.111395e+01	0.377458e+01	0.266063e+01
6	0.720000e-05	0.168792e+01	0.335986e+01	0.167194e+01
7	0.620000e-05	0.202631e+01	0.364741e+01	0.162110e+01
8	0.690000e-05	0.243792e+01	0.373945e+01	0.130153e+01
9	0.730000e-05	0.305548e+01	0.397821e+01	0.922735e+00
10	0.997000e-04	0.347071e+01	0.415136e+01	0.680642e+00
11	0.137570e-02	0.379457e+01	0.439375e+01	0.599170e+00
12	0.593880e-02	0.416238e+01	0.458430e+01	0.421922e+00
13	0.375400e-02	0.446022e+01	0.474224e+01	0.282027e+00
14	0.323500e-03	0.457285e+01	0.483109e+01	0.258243e+00

Accordingly, our final formula for the entropy will be

$$S(U, \{P_k\}, M) \sim \bar{S} := \frac{1}{I} \sum_{i=1}^{I} S_i^u, \tag{9.1}$$

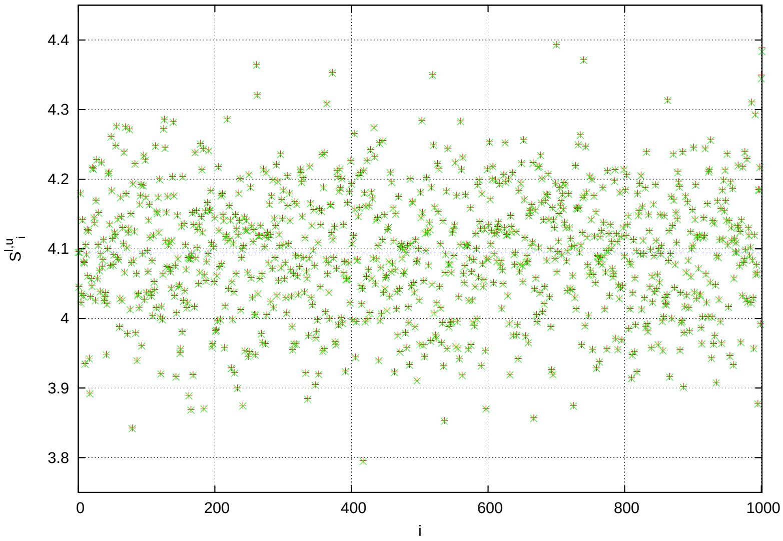

FIG. 8.1. *Sample entropies lower bounds S^l_i (green x's) and upper bounds S^u_i (red +'s) for $M = 11$ and six Gaussian points versus sample number i. Data sets are almost coincident, on the scale of the figure. The horizontal dotted line is the sample average.*

where S^u_i indicates the Gaussian summation result for $S_i := (W_i, -\Omega^M \log(\Omega^M)W_i)$. We now turn to the problem of deriving a confidence interval for $S(U, \{P_k\}, M)$.

Clearly, S^u_i is a realization of a random variable of mean μ and finite variance η^2: therefore, the sample average $\bar{S}$ is itself a random variable, with the same mean, and standard deviation $\eta/\sqrt{I}$. In addition, because of the central limit theorem, the distribution of $\bar{S}$ tends to the normal, when I tends to infinity, and we might think of using this fact to improve the Hoeffding's bounds.

In the case at hand, moreover, the *individual* sample values S_i appears to be approximately normally distributed, the more so the larger the value of M: this is apparent in Figure 9.1, where we compare the Gaussian distribution function $F(z) := \frac{1}{2}(\text{erf}(z)+1)$ with the experimental distribution functions F of the standardized random variables $z := (S_i - \mu)/\eta$. All quantities (including μ and η) are estimated from the $I = 1000$ samples of the previous section, and various values of M are reported.

Therefore, we can safely assume that the sample means $\bar{S}$ are normally distributed to high precision, and we can derive a confidence interval of probability p according to the standard values $z_{(1+p)/2}$ of common usage in statistics:

$$\bar{S} \pm z_{(1+p)/2}\frac{\eta}{\sqrt{I}} := \bar{S} \pm Z^G_p \frac{1}{\sqrt{I}}\,. \tag{9.2}$$

Confidence intervals derived via Hoeffding's inequality have the same form (9.2), where $Z^G_p = \eta z_{(1+p)/2}$ is replaced by Z^H_p:

$$Z^H_p := (U_{max} - L_{min})\sqrt{\frac{1}{2}\log(\frac{2}{1-p})}; \tag{9.3}$$

the usual Chebyshev inequality yields in turn eq. (9.2) with Z^T_p given by

$$Z^T_p := \frac{\eta}{\sqrt{1-p}}. \tag{9.4}$$

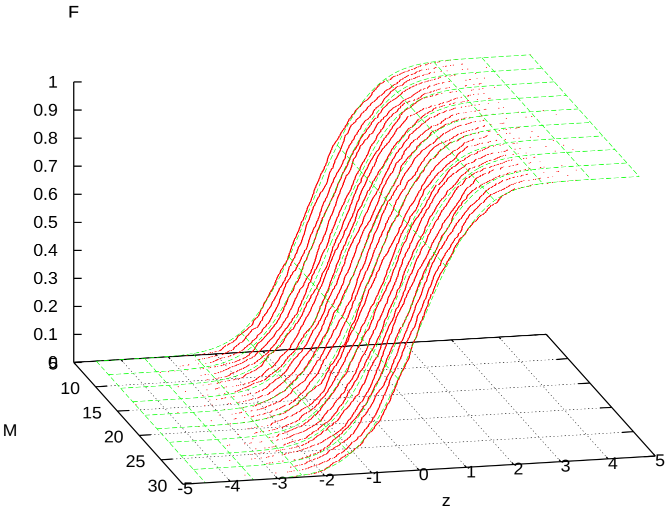

FIG. 9.1. *Probabilistic distribution functions F of the sample entropies S_i^u, standardized to zero mean and unit variance, versus z and M (red dots). Green lines: distribution function of the normal random variable $F(z)$ (independent of M).*

TABLE 9.1
Confidence values Z_p with $p = 0.99$ for the same case of Table 8.1.

M	Z_p^H	Z_p^T	Z_p^G
1	0.590861e+01	0.458289e+01	0.106781e+01
2	0.554075e+01	0.535940e+01	0.124874e+01
3	0.464403e+01	0.470522e+01	0.109632e+01
4	0.456015e+01	0.451669e+01	0.105239e+01
5	0.433051e+01	0.366327e+01	0.853543e+00
6	0.272129e+01	0.296183e+01	0.690107e+00
7	0.263854e+01	0.250148e+01	0.582845e+00
8	0.211840e+01	0.195064e+01	0.454500e+00
9	0.150187e+01	0.153929e+01	0.358653e+00
10	0.110783e+01	0.117137e+01	0.272929e+00
11	0.975224e+00	0.893929e+00	0.208285e+00
12	0.686731e+00	0.660134e+00	0.153811e+00
13	0.459033e+00	0.457901e+00	0.106691e+00
14	0.420323e+00	0.336077e+00	0.783060e-01

In Table 9.1 we report the Z_p values with $p = 0.99$ for the same case of Table 8.1. We observe that while Chebyshev's and Hoeffding's inequalities give comparable results (the former being indeed better than the latter in most cases) the normal estimate is superior (that is, narrower) by a factor of about four at this value of p. In terms of computer time, this means a most significant reduction of a factor of 16 in I, the number of Lanczos' evaluations needed to attain the same accuracy.

10. The Jacobi matrix of the spectral measure of Ω. In this section we propose an alternative to the algorithm G2 just presented and utilized. According to the latter, a Jacobi matrix is computed for each random vector W_i, $i = 1, \ldots, I$. We want now to compute a single Jacobi matrix for the estimation of the entropy function $S(U, \{P_k\}, M)$. In this section, Ω will be a shorthand notation for Ω^M.

Technically, the Jacobi matrices computed in the algorithm G2 correspond to the spectral measures ν^i defined as follows: let Ψ_j be the eigenvectors of Ω and λ_j the associated eigenvalues. For any $x \in \mathbf{R}$ let δ_x be the atomic measure at x. Then, ν^i is the measure

$$\nu^i := \sum_j \delta_{\lambda_j} |(W_i, \Psi_j)_{\mathcal{K}}|^2. \tag{10.1}$$

In physics, this measure is called the "local density of states". Entropy, in turn, is the integral of the function $f(x) = -x \log(x)$ with respect to ν, the spectral measure of Ω, called the "density of states" in the physics literature:

$$\nu := \sum_j \delta_{\lambda_j}. \tag{10.2}$$

It is therefore the Jacobi matrix of ν that we need to compute.

To do this, recall that Ω is an operator from $\mathcal{K}$ to itself. Introduce the linear space $\mathcal{L}$ of such operators, endowed with a scalar product just as done in $\mathcal{K}$: for any $\Theta, \Xi \in \mathcal{L}$

$$(\Theta, \Xi)_{\mathcal{L}} := Tr(\Theta^\dagger \Xi), \tag{10.3}$$

where obviously the trace is taken in the new space: for any $\Theta \in \mathcal{L}$

$$Tr(\Theta) := \sum_{k,l} (E_{k,l}, \Theta E_{k,l})_{\mathcal{K}}, \tag{10.4}$$

being $E_{k,l}, k = 0, \ldots, N-1,\ l = 0, \ldots, N/2-1$ the basis vectors of $\mathcal{K}$, given by eq. (3.4) and by the remark at the end of section 3.

Define the sequence of polynomials $p_n(\Omega)$ of degree n in $\mathcal{L}$ initialized by $p_{-1}(\Omega) = 0$, $p_0(\Omega) = \mathbf{I}$ (0 and $\mathbf{I}$ being the null operator and the identity in $\mathcal{L}$) that satisfy the three-term relation

$$\Omega p_n(\Omega) = b_{n+1} p_{n+1}(\Omega) + a_n p_n(\Omega) + b_n p_{n-1}(\Omega), \tag{10.5}$$

with real coefficients a_n, $b_n \geq 0$, $n = 0, \ldots$. The definition is unique if we enforce that these polynomials (each of which is an operator in $\mathcal{L}$) be orthogonal with respect to the scalar product (10.3) and normalized for $n > 0$:

$$(p_n(\Omega), p_m(\Omega))_{\mathcal{L}} = \delta_{n,m}. \tag{10.6}$$

Of course, these are *not* the orthogonal polynomials of a measure over $\mathcal{L}$: as a matter of fact, the coefficients a_n and b_n do depend on Ω. Yet, they serve our scope:

LEMMA 10.1. *The coefficients a_n and b_n, $n = 0, \ldots,$ are the entries of the Jacobi matrix of the measure ν associated with the Hermitian matrix Ω.*

Proof. We start by observing that

$$\int d\nu = N^2/2 = (p_0(\Omega), p_0(\Omega))_{\mathcal{L}} = Tr(\mathbf{I}) = \nu_0,$$

and that

$$a_0 := (p_0(\Omega), \Omega p_0(\Omega))_{\mathcal{L}} = Tr(\Omega) = \sum_j \lambda_j = \nu_1.$$

ν_0 and ν_1 are the first two moments of ν. Furthermore, it is easily seen that the coefficients a_n and b_n are constructed as

$$a_n = (p_n(\Omega), \Omega p_n(\Omega))_{\mathcal{L}} = Tr(p_n(\Omega)^\dagger \Omega p_n(\Omega)), \tag{10.7}$$

and

$$b_{n+1}^2 = ((\Omega - a_n)p_n(\Omega) - b_n p_{n-1}(\Omega), (\Omega - a_n)p_n(\Omega) - b_n p_{n-1}(\Omega))_{\mathcal{L}}. \tag{10.8}$$

and that $Tr(g(\Omega)) = \sum_j g(\lambda_j) = \int g(x) d\nu(x)$ for any continuous function g. Therefore, $p_n(x)$ computed via eq. (10.5) with x in place of Ω and the coefficients a_n and b_n derived as above, is the n-th orthogonal polynomial of ν. □

A Lanczos algorithm for the computation of this Jacobi matrix follows directly from the previous lemma and can be easily set up, at least in principle. Yet, this algorithm faces two main computational difficulties. Firstly, it requires computation of the traces (10.7) and (10.8). Secondly, it requires the storage of three Lanczos vectors $p_{n+1}(\Omega), p_n(\Omega), p_{n-1}(\Omega)$, each of which of size $N^4/4$.

The first difficulty can be overcome by the same statistical idea applied in G2: rather than computing traces as summations over all the $N^2/2$ vectors $E_{k,l}$, choose a fixed set of random vectors $\{W_i, i = 1, \ldots, I\}$, and estimate the trace in eq. (10.3) as

$$Tr_{\mathcal{L}}(\Theta) \simeq \frac{1}{I} \sum_i (W_i, \Theta W_i)_{\mathcal{K}}. \tag{10.9}$$

The second difficulty can be avoided by noticing that in so doing computation of the traces (10.7) and (10.8) only requires the vectors $p_n(\Omega)W_i$, that can be obtained via repeated actions of the matrix Ω in the recursion relation (10.5). Therefore, the resulting algorithm ends up to be a variation of G2:

Algorithm 5: Lanczos-Monte Carlo computation of the Jacobi matrix of ν.

```
• set $p_0 = I$, $b_0 = 0$
• for $j = 0, \ldots, J$
    1. set $a = 0$
        – for $i = 1, \ldots, I$
            ∗ generate the random vector $W_i$ and the three-term sequence $p_n W_i$ for
              $n = 0, \ldots, j$, as well as the vector $\Omega p_j W_i$
            ∗ compute the scalar product $(p_j W_i, \Omega p_j W_i)_{\mathcal{K}}$
            ∗ accumulate the result into the variable $a$
        – end
    2. set $a_j = a/I$
    3. set $b = 0$
        – for $i = 1, \ldots, I$
            ∗ generate the random vector $W_i$ and the three-term sequence $p_n W_i$ for
              $n = 0, \ldots, j$, as well as the vector $X_i = (\Omega - a_j)p_j W_i - b_j p_{j-1} W_i$
            ∗ compute the scalar product $(X_i, X_i)_{\mathcal{K}}$
            ∗ accumulate the results into the variable $b$
        – end
    4. set $b_{j+1} = \sqrt{b/I}$
• end
```

Of course, all the usual precautions to be observed in Lanczos algorithms apply here, like early terminations for small b at step 4. Once the Jacobi matrix of ν has been obtained, it can be easily diagonalized, and Gaussian integration performed. The advantage of this technique is that diagonalization is performed only once, and it can be effected at every value of J. In Figure 10.1 we show the estimated entropy versus M and J. We notice that good results can be obtained already at $J = 2$. Computation complexity can improve upon that of G2 if the vectors $p_j(\Omega)W_i$ can be stored in memory for three consecutive values of j.

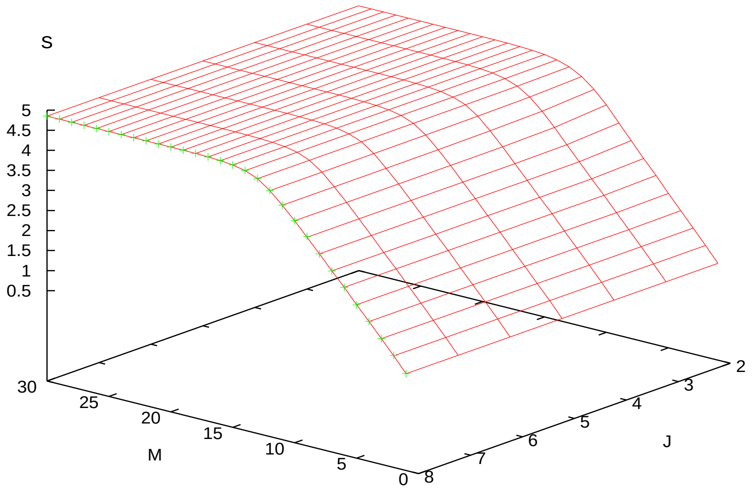

FIG. 10.1. *A-F entropy S for the quantum cat evolution versus M and Jacobi size J (red lines). Here, $N = 2^7$. Sample is size $I = 1000$. The green symbols at $J = 8$ report the data obtained with the Algorithm G2. Perfect accordance is observed.*

11. Conclusions. Quantum chaos, two examples. We can now finally enjoy the display of the A-F entropies $S(U, \{P_k\}, M)$ versus M as confidence intervals for the two paradigmatic examples introduced in section 2. Notice however that plots displayed refer to the contribution of the $\sigma_0 = 0$ sector of the matrix (see eq. (3.8) and the related discussion), that turns out to correspond exactly to one half of the total value. Figure 11.1 displays, versus M, the $p = 0.99$ confidence intervals that, thanks to the normal estimates of section 9, are smaller than symbol size. In the former case, the cat map, we observe a linear initial increase of $S(U, \{P_k\}, M)$, followed by saturation to the maximum attainable value $S^{max} = \log(N)$. This termination is induced mathematically from the finiteness of the Hilbert space of the system, and physically by the effect of quantum interference. In the second case, a sublinear increase, also with a saturation, is observed.

Physical analysis takes off from this point [7, 3]. In the renown problem of quantum classical correspondence, this logarithmic time barrier [8] could be beaten following the prescriptions of decoherence [10]: the present work aims at developing analytical and numerical techniques to address this problem rigorously. It is our conviction that the numerical techniques presented in this paper will open the way to investigations up to now impossible with conventional algorithms [2].

In conclusion, I have presented in this paper a series of techniques that will render viable the computation of the A.F. entropy for systems of physical interest. These techniques rest on the algorithmic properties of the Ω^M matrices introduced in the original work [3] and here systematically investigated, and on a blend of parallel computing algorithms and the Lanczos' technique of Golub. In addition, I have shown how the normal property of the distribution of statistical samples permits to largely improve the statistical bounds provided in [5], allowing a significant reduction in computer time. Finally, an algorithm for the direct computation of the Jacobi matrix associated to the spectral measure of Ω^M has been presented. Its performance, in comparison with the previous algorithms, will be the object of further investigation. Outside the present problem, this last algorithm might have relevance in the study of singular continuous measures and of their Fourier transforms [11, 12].

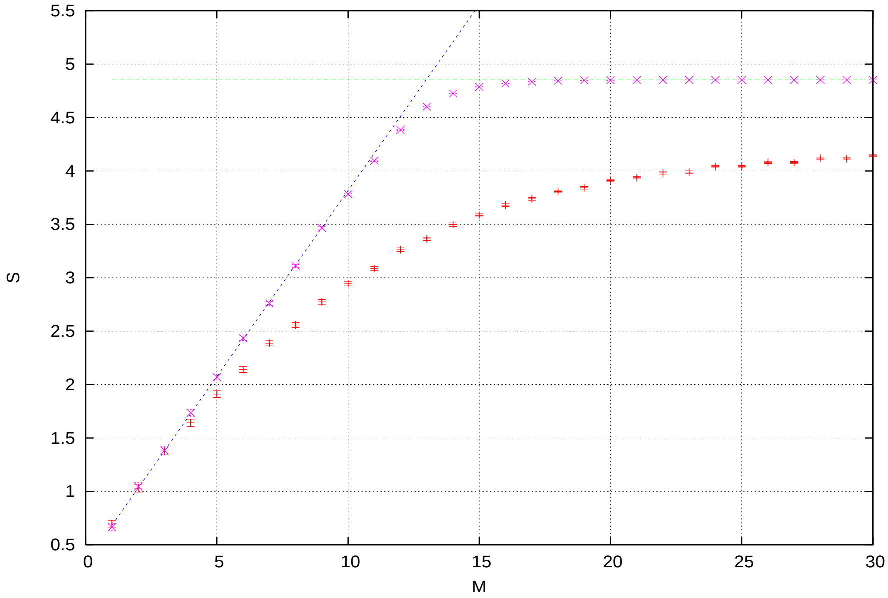

FIG. 11.1. *Confidence intervals for the entropy S with $p = 0.99$ for the quantum cat evolution (magenta x's) and the free rotation (red +'s) versus M. Here, $N = 2^7$, and six Gaussian points have been used. Sample size $I = 1000$ is such that the confidence interval, although plotted, is smaller than symbol size. The horizontal line is at $S = \log(2^7)$ and the inclined line has slope $s = .3466 \simeq \log(2)/2$. Notice that data displayed in these plots are for the $\sigma_0 = 0$ sector of the matrix Ω: see eq. (3.8).*

Acknowledgments. This research benefitted from discussions and advice by Gene Golub. The author warmly remembers his enthusiasm, dedication and kindness. The author also wishes to thank Luca Paredi for invaluable assistance in taming a reluctant processor cluster.

REFERENCES

[1] R. ALICKI AND M. FANNES, *Defining quantum dynamical entropy* , Lett. Math. Phys., 32 (1994), pp. 75–82.

[2] R. ALICKI, A. LOZINSKI, P. PAKONSKI, AND K. ZYCZKOWSKI, *Quantum dynamical entropy and decoherence rate*, J. Phys. A: Math. Gen., 37 (2004), pp. 5157–5172.

[3] R. ALICKI, D. MAKOWIEC, AND W. MIKLASZEWSKI, *Quantum chaos in terms of entropy for a periodically kicked top*, Phys. Rev. Lett., 77 (1996), pp. 838–841.

[4] V. I. ARNOL'D AND A. AVEZ, *Ergodic Problems of Classical Mechanics*, W. A. Benjamin, Inc., New York, 1968.

[5] Z. Bai, M. Fahey and G. H. Golub, *Some large-scale matrix computation problems*, J. Comput. Appl. Math., 74 (1996), pp. 71–89.

[6] S. Dong and K. Liu, *Stochastic estimations with Z_2 noise*, Phys. Lett. B, 328 (1994), pp. 130–136.

[7] J. Ford, G. Mantica and G. Ristow, *The Arnol'd cat: Failure of the correspondence principle*, Phys. D, 50 (1991), pp. 493–520.

[8] J. Ford and G. Mantica, *Does quantum mechanics obey the correspondence principle? Is it complete?* Am. J. Phys., 60 (1992) pp. 1086–1098.

[9] G. H. Golub and G. Meurant, *Matrices, moments and quadrature*, Tech. Report SCCM-93-07, Stanford University, 1993.

[10] A. R. Kolovsky, *Quantum coherence, evolution of the Wigner function, and transition from quantum to classical dynamics*, Chaos, 6 (1996), pp. 534–542.

[11] G.Mantica, *Fourier-Bessel functions of singular continuous measures and their many asymptotics*, Electron. Trans. Numer. Anal., 25 (2006), pp. 409–430.
http://etna.math.kent.edu/vol.25.2006/pp409-430.dir/pp409-430.html

[12] G. Mantica *Fourier transforms of orthogonal polynomials of singular continuous spectral measures*, in Applications and computation of orthogonal polynomials (Oberwolfach, 1998), Internat. Ser. Numer. Math., 131, Birkhäuser, Basel, 1999, pp. 153–163.

[13] Ya. B. Pesin, *Characteristic Lyapunov exponents, and smooth ergodic theory*, Russian Math. Surveys, 32 (1977), pp. 55–114.

[14] M. C. Reed and B. Simon, *Methods of Modern Mathematical Physics II. Fourier Analysis, Self-Adjointness*, Academic Press, New York, 1975.