DIMENSIONAL REDUCTIONS FOR THE COMPUTATION OF TIME–DEPENDENT QUANTUM EXPECTATIONS[*]

GIACOMO MAZZI[†] AND BENEDICT J. LEIMKUHLER[†]

Abstract. We consider dimension reduction techniques for the Liouville–von Neumann equation for the evaluation of the expectation values in a mixed quantum system. We describe several existing methods that have appeared in the literature, showing the failure of them when the system is scaled up. We introduce a new method termed direct expectation values via Chebyshev based on evaluation of a trace formula combined with a direct expansion in modified Chebyshev polynomials. This reduction is highly efficient and does not destroy any information. We demonstrate the practical application of the scheme for a nuclear spin system and compare with popular alternatives. In nuclear spin dynamics the main goal for simulations is being able to simulate a system with as many spins as possible; for this reason it is very important to have an efficient method that scales the least with respect to the number of particles. This method may be applied to autonomous quantum problems where the desired outcome of quantum simulation, rather than being a full description of the system, is only the expectation value of some observables.

1. Introduction. When dealing with the dynamics of a quantum system of particles, the motion is described by the Schrödinger equation

$$\frac{\partial |\Psi\rangle}{\partial t} = -iH|\Psi\rangle, \tag{1.1}$$

where H is the Hamiltonian and $|\Psi\rangle$ the wave function of the system, and we have chosen physical quantities such that $\hbar = 1$. When H is time independent, it is possible to have an exact formulation of the solution:

$$|\Psi(t)\rangle = e^{-iHt}|\Psi_0\rangle. \tag{1.2}$$

From a numerical point of view, finding a solution of (1.2) has proved to be a challenging task in the past decades. Many numerical methods have been proposed over the years to solve (1.2), from polynomial expansion of the exponential [1, 2, 3], to projection on Krylov subspaces [4, 5, 6].

When the wave function $|\Psi\rangle$ depends on the particle positions q and momenta p, it is very common to adopt a splitting method for the Hamiltonian, $H = T + V$ with V diagonal and T the kinetic term that is diagonal in Fourier space. With this approach the main cost is the evaluation of a fast Fourier transform (FFT) to switch between the x-grid and the p-grid; the cost of this transform is $O(N \log N)$, where N is the size of the system [7]. However, when the wave function depends on other variables, like the spin of the particles, it is not straightforward to apply a splitting method. For a general survey of these approaches see [8, 9].

[†]School of Mathematics, University of Edinburgh, Edinburgh EH9 3JZ, United Kingdom (G.Mazzi@sms.ed.ac.uk, bleimkuhler@ed.ac.uk).

It is important to remark that all of these methods suffer when the size of Ψ becomes large. In many practical cases, such as spin dynamics, where Ψ depends on or only on the spin variable, the size of Ψ grows exponentially with the number of particles of the systems. In fact, the dimension of the Hilbert space where Ψ lies in the case of a system with n spins I is $(2I + 1)^n$.

This paper is organized as follows. In section 2 we introduce the Liouville–von Neumann equation that describes the dynamics of the density matrix; we also discuss the issues that arise when attempting to solve this equation. In section 3 we describe two of the main techniques that have been applied in quantum simulation for the evaluation of the solution of the Schrödinger equation, specifically the Krylov expansion via Lanczos–Arnoldi iterations and the expansion in Chebyshev polynomials. We also present a more recent method based on a different evaluation of the Krylov subspace expansion called zero track elimination (ZTE) [10]. This method has been developed for simulations of nuclear spin dynamics within the nuclear magnetic resonance (NMR) community. We give also its first mathematical analysis of it, exploring the limitations and explaining why it typically works in the application to NMR spin simulation. In section 4 we introduce the new method we have developed: termed *direct expectation values via Chebyshev (DEC)*. In this section also the technical details of the implementation are presented; we also comment on the relationship between DEC and the kernel polynomial method (KPM) [11]. Finally, in section 5 we make a performance comparison between (a) DEC, (b) classical Krylov expansion based on the Lanczos algorithm, (c) the ZTE method, and (d) Chebyshev expansion. The sample system is a pairwise interacting nuclear spin system, which, although simple, contains the main features of a proper system the NMR community is interested in.

2. The Liouville–von Neumann equation. When the sample of interest is composed of a large number N of identical systems each described by a wave function $|\Psi\rangle$, it is common to introduce a statistical operator called the density operator ϱ defined by

$$(2.1) \qquad \varrho = \sum_{j=1}^{N} |\Psi^j\rangle\langle\Psi^j|,$$

which evolves according to the Liouville–von Neumann equation

$$(2.2) \qquad \frac{\partial \varrho}{\partial t} = -i[H, \varrho],$$

where the square brackets $[,]$ indicate the usual commutation brackets: $[H, \varrho] = H\varrho - \varrho H$. If ϱ is expanded using a (finite) approximate basis set $\{|\varphi_1\rangle \ldots |\varphi_n\rangle\}$, (2.2) may be viewed as an ordinary differential equation in a matrix argument. In the time-independent case the solution for (2.2) may be written as follows:

$$(2.3) \qquad \varrho(t) = e^{-iHt}\varrho_0 e^{iHt}.$$

It is possible and sometimes preferable to rewrite (2.3) by introduction of the Liouvillian $L = \mathrm{Id} \otimes \mathrm{H} - \mathrm{H} \otimes \mathrm{Id}$, where Id is the identity matrix, and $\otimes$ indicates the tensor product, allowing us to recast ϱ as a vector:

$$(2.4) \qquad \varrho^V(t) = e^{-iLt}\varrho_0^V,$$

where if ϱ is a matrix of size $n \times n$, ϱ^V is an n^2-dimensional vector constructed by joining together the columns of ϱ:

$$(2.5) \qquad \varrho^V = \begin{pmatrix} \varrho_{11} \\ \cdots \\ \varrho_{n1} \\ \varrho_{21} \\ \cdots \\ \cdots \\ \varrho_{nn} \end{pmatrix}.$$

For a large system and using a typical basis set, H and L will be structured sparse matrices. If the sparsity is taken into account when storing these matrices, the computational cost of (2.4) is equal to that of (2.3). At the same time most of the literature is concerned with matrix-vector operations, so the Liouvillian formulation is more suitable for comparison with other works [9]; also this formulation is better suited when dealing with relaxation phenomena [12].

While the evolution of the system is described by the density matrix, the outputs we are interested in obtaining from quantum simulations are typically the expectations of observables, these being the only quantities we can compare with experiments. In the density matrix formalism the expectation value of an observable Q, associated with an operator $\hat{Q}$ is written as

$$(2.6) \qquad \langle \hat{Q}(t) \rangle = \mathrm{Trace}\{\varrho(t)\hat{Q}\}.$$

If instead of propagating $\varrho(t)$ using (2.3) we evaluate $\varrho^V(t)$ via (2.4), we simply need to recast ϱ^V into a matrix form before applying (2.6). For the rest of the paper we drop the superscript V when dealing with the vectorized form of ϱ (2.5) as we will always use the density matrix in its vectorized form.

Whereas in general quantum simulations the equation of motion is solved for a quantity ϱ that has dimension $n \times n$, the types of outputs we are generally interested in are just one-dimensional objects (2.6). In this paper we exploit this fact and design an algorithm that computes (almost) directly the evolution of the expectation value (2.6), instead of the evolution of the density matrix (2.3). This approach does not lose any information of the original system (the only errors arise due to truncation), but at the same time the method provides a powerful computational tool, with potential dramatic reduction in the computational cost, especially when dealing with large matrices. The main idea of this approach is to exploit features of a Chebyshev expansion for the matrix exponential in (2.4).

Several methods to evaluate (2.4) are discussed in a recent monograph [9]; however, our proposed DEC method is different because it does not directly evolve the density matrix. Instead it exploits the trace evaluation in (2.6), and with the computation of just one Chebyshev expansion it allows the solution of (2.6) at any time.

The idea of storing the traces has already been applied for evaluations of spectral quantities, mainly the density of states, within the so-called KPM. A key difference between our method and KPM is that in the latter method, this accumulation is not used for a time-dependent quantity, so no issues of convergence of the series at different times need to be taken into account [13].

DEC can be extremely powerful when we are only interested in the expectation values; if instead it is necessary to evaluate the evolution of the density matrix itself using (2.4), then traditional approaches, such as those presented in the next section, where the full density matrix is propagated, would be needed.

3. Existing methods. In the past decades many methods have been developed for the numerical evaluation of the matrix exponential [14]. Among them, the most succesful methods from a computational viewpoint, especially when dealing with large matrices, are those based on the evaluation of the product of the exponential and the initial vector, rather than those where the matrix exponential is explicitly calculated.

In fact, given ϱ_0, it is possible, and much less involved from a numerical point of view, to evaluate $\varrho(t) = \exp(-iLt)\varrho_0$ as in (2.4) without any explicit calculation of $\exp(-iLt)$. For this reason, especially when dealing with large sparse matrices, such methods are usually applied.

3.1. Krylov expansion via Lanczos–Arnoldi. Examples of such methods are those based on expansion of the exponential in Krylov subspace [15, 16, 17, 18, 19]. The main idea is to project (2.4) onto the subspace:

$$(3.1) \qquad K_m(L, \varrho_0) = \mathrm{span}\{\varrho_0, L\varrho_0, L^2\varrho_0, \ldots, L^m\varrho_0\}.$$

To get a suitable basis for the Krylov subspace, we may use the Lanczos algorithm [20], as L is Hermitian. The Lanczos method is an iterative method, a very desirable feature in the context of large sparse matrices. For specific details see [21].

An approximation for $\varrho(t)$ is

$$(3.2) \qquad \varrho(t) = e^{-iLt}\varrho_0 \approx \|\varrho_0\| V_m e^{-iT_m t} e_1,$$

where T_m and V_m come from the Lanczos algorithm. The Lanczos algorithm provides an orthonormal basis set V_m for the Krylov subspace $K_m(L, \varrho_0)$ via a three-term recursion [20, 22]:

$$(3.3) \qquad \beta_{j+1}q_{j+1} = Lq_j - \alpha_j q_j - \beta_j q_{j-1}, \qquad \alpha_j = (Lq_j, q_j),$$

with initial values $q_0 = 0$, $q_1 = \varrho_0$. T_m is a tridiagonal matrix of size $m \times m$, and e_1 is the first vector of the canonical basis of size n. This technique is very powerful for short time simulations, because with few iterations m it is possible to have remarkably good approximations; but for longer times larger Krylov subspaces would be needed to stay close to the real solution. On the other hand, if we do not consider enough terms in the Lanczos algorithm for longer times, (3.2) is no longer a reliable approximation.

It is possible to set a stopping criterion for the Lanczos iterations [17]; for a given t we can find m such that

$$(3.4) \qquad t[T_m]_{m+1,m}|[e^{-itT_m}]_{m,1}| \leq \varepsilon,$$

where $|[A]_{i,j}|$ indicates the absolute value of the i,jth element of the matrix A. In our experiments the best way to implement a Krylov expansion was to evaluate each step:

$$(3.5) \qquad \varrho_{n+1} = e^{-iLdt}\varrho_n \approx \|\varrho_n\| V_m^n e^{-iT_m^n t} e_1.$$

In this way with less than 10 iterations of Lanczos per step it was possible to have a fast and accurate benchmark. The obvious drawback is that no information passes from step n to step $n + 1$. However, in our numerical tests the use of a longer time step that would allow a common Krylov subset $K_m(L, \varrho)$ for more than one step ϱ_n was not preferrable as it requires more iterations of (3.3). Because of the fact that (3.4) involves the evaluation of the exponential of a tridiagonal matrix, when m is large this operation becomes a serious bottleneck for the whole simulation.

3.2. The Chebyshev expansion. Another method that has been widely applied for solving (1.1) is the expansion of the exponential into Chebyshev polynomials [1, 2, 25].

The preliminary step of this method is to rescale the matrix within the interval $[-1, 1]$, as outside this interval the Chebyshev polynomials grow exponentially, and the expansion becomes unstable; to do that we need to evaluate the two extremes of the spectrum of L.

In order to obtain extreme values we propose, as already mentioned in the literature [23], to perform a few steps of Lanczos iteration, as this provides a good approximation for the extreme eigenvalues, for small computational cost. It is known that the values we get from such a process are only approximate, and they are inside the extremes of the real spectrum of L, $\sigma(L)$. One way to overcome this issue is to slightly expand the range obtained using Lanczos iterations [11]; another approach is to use the sum of the norms of T_k and q_k obtained from the initial 4–5 iterations of (3.3) to approximate an upper bound [23].

In our numerical tests we used the MATLAB function *eigs* to compute a few eigenvalues of a given matrix starting from the two extemes of the spectrum using an ARPACK [24] routine based on the Lanczos–Arnoldi iteration. The same approach has been used elsewhere in the literature [25].

In our experiments, the values provided by this function were accurate enough to avoid any issue coming from not having rescaled L exactly within the $[-1, 1]$ interval. If we define these two values as α and β, i.e., $\beta \leq \sigma(L) \leq \alpha$, we may rewrite L as $L = (S\,\mathrm{Id} - L_s D)$, where $D = (\alpha - \beta)/2$, $S = (\alpha + \beta)/2$, and $-1 \leq \sigma(L_s) \leq 1$. We may then expand the exponential of L_s in the Chebyshev polynomials, and we arrive at the following equation for ϱ:

$$(3.6) \qquad \varrho(t) = e^{-iLt}\varrho_0 \approx e^{-itS}\left(\sum_{k=0}^{n_{\max}} c_k(t_D)T_k(L_s)\varrho_0\right),$$

with $t_D = Dt$. Both $c_k(t_D)$ and $T_k(L_s)$ can be calculated iteratively:

$$(3.7) \qquad c_k(t) = (2 - \delta_{k,0})(-i)^k J_k(t),$$

where $\delta_{k,0}$ is the Kronecker delta,

$$(3.8) \qquad T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x),$$

with initial values $T_0(x) = 1$, $T_1(x) = x$. $J_k(t)$ is the kth Bessel function of the first kind.

The Bessel functions of the first kind of integer order may be evaluated directly by using a three-term recurrence relation

$$(3.9) \qquad J_{n+1}(t) = \frac{2n}{t}J_n(t) - J_{n-1}(t).$$

It is well known that (3.9) becomes numerically unstable for $n > t$ [26]. To improve the method, we may exploit the linear nature of the iterative algorithm. It is possible to use Miller's algorithm, and to solve an inverted form of (3.9), i.e., to solve for J_{n-1} given J_n, J_{n+1} [26]. When using Miller's algorithm it is suggested to expand the number of terms (providing a sort of buffer), i.e., to start the backward iteration process from $m_{\mathrm{start}} = n + r$, where n is the actual order of the function we are

interested in and r is some small expansion. In this case we need to know already from an a priori error analysis how many iterations need to be performed to get below the threshold ε.

It is possible to prove that for the rescaled Hermitian matrix L_s, when applied to a vector of unit Euclidian norm we have [9]

$$(3.10) \qquad \|P_{m-1}(tL_s)\varrho_0 - e^{-itL_s}\varrho_0\| \leq 4\left(e^{1-(t/2m)^2}\frac{t}{2m}\right)^m \quad \text{for } m > t,$$

where $P_m(t)$ is the order m expansion in Chebyshev polynomials. This equation indicates that there is a superlinear decay of the error when $m > t$.

We may then use the relation $4(\exp\{1 - (\tau/2m)^2\}\frac{\tau}{2m})^m \leq \varepsilon$ to approximate m. From a practical point of view the usual way of applying Chebyshev is to evaluate [3, 1]

$$(3.11) \qquad \varrho_{n+1} = e^{-iLdt}\varrho_n \simeq P_{n_{\max}}(dtL)\varrho_n,$$

where

$$(3.12) \qquad P_{n_{\max}} = e^{-idtS}\sum_{k=0}^{n_{\max}} c_k(dtD)T_k(L_s).$$

$n_{\max}$ may be evaluated either from (3.10) or by directly checking the convergence (4.7).

To avoid numerical instabilities coming from the iterative formula for the Bessel functions, it is also possible to get $P_m\varrho_n$ for a given m via the Clenshaw algorithm [27, 9]:

$$(3.13) \qquad d_k = c_k\varrho_n + 2L_sd_{k+1} - d_{k+2}, \qquad k = m-1, m-2, \ldots, 0,$$

with initial values $d_{m+1} = d_m = 0$ and $P_m(dtL)\varrho_n = d_0 - d_2$.

From a computational cost point of view the performances of the two methods (Lanczos and Chebyshev) are usually comparable, although depending on the features of the problem there are cases where Lanczos [28] or Chebyshev [25] may be preferred. In our numerical tests they were practically equal when dealing with large matrices.

3.3. The Zero-Track-Elimination method. NMR is a spectroscopy technique that exploits the interaction between nuclear spins and electromagnetic fields in order to analyze the samples. The temporal evolution of such a system is described via a density matrix that has size $(2I + 1/2)^n$, where I is the spin and n the number of nuclei. The exponential growth of the size of ϱ with respect to n impedes the use of simulations when dealing with systems involving more than a few (5–6) spins. Many attempts have been made to solve this (see [29, 10] for recent approaches), even using Chebyshev polynomials [3]. These algorithms have been developed to simulate both liquid systems, where the Hamiltonian is generally time independent, and for solid–state NMR. In the latter case the Hamiltonian is time dependent due to the not averaging out of anisotropic interactions during the motion of the sample, and as previously remarked DEC is not applicable in this case.

Recently also a new method for the simulation of a large spin system called ZTE has been presented [10]. This technique is based on the idea of pruning out the elements of $\varrho(t)$ that do not belong to $K(L, \varrho_0)$. Most of the time the initial density ϱ_0 is sparse.

In order to reduce the steps needed to evolve the full system, we monitor the elements of $\varrho(t)$ that stay below a chosen threshold ξ during these first evolution

steps and introduce structural zeros based on these observations. The evolution is then performed in this reduced state space (ϱ_Z, L_Z). The idea is extremely appealing, as once the propagator for L_z is evaluated all the subsequent steps have the cost of a reduced matrix vector, and it is possible to use traditional methods to evolve ϱ_Z in the reduced system:

$$(3.14) \qquad \varrho_Z(t) = e^{-iL_z t} \varrho_Z(0).$$

The initial time length is set as the inverse of the largest Larmor frequency. The Larmor frequency is a given quantity for each element that depends on the physical property of the nucleus and is the frequency of resonance for a noninteracting spin: $\omega_j^0 = -\gamma_j B$, where $-\gamma_j$ is the gyromagnetic ratio of the nucleus and B the applied magnetic field.

The time length of the initial check is then

$$(3.15) \qquad \delta t = \frac{1}{t_{\text{lar}}} = \frac{2\pi}{\min_j\{|\omega_j^0|\}},$$

where ω_j^0 is the Larmor frequency of the jth spin. The following theorem given in [10] assures that it is possible to prune out from the evolution those states that remain exactly 0 during the first time steps. It is possible to prove that for a state $|l\rangle$ holds:

$$(3.16) \qquad \langle l|e^{-iLt}|\varrho_0\rangle = 0, \quad \text{for } t \in [0, \delta t] \Rightarrow \langle l|e^{-iLt}|\varrho_0\rangle = 0, \quad \text{for } t \in [0, \infty).$$

To prove (3.16) it is sufficient to expand e^{-iLt} into a Taylor expansion. In fact,

$$(3.17) \qquad \langle l|e^{-iLt}|\varrho_0\rangle = \langle l|\sum_{k=0}^{\infty} L^k \frac{(-it)^k}{k!}|\varrho_0\rangle = \sum_{k=0}^{\infty} \frac{(-it)^k}{k!}\langle l|L^k \varrho_0\rangle, \quad \text{for } t \in [0, \delta t],$$

can be true for any $t \in [0, \delta t]$ only if $\langle l|L^k \varrho_0\rangle = 0$ for all k, but this means that $\langle l|e^{-iLt}|\varrho_0\rangle = 0$, for $t \in [0, \infty)$.

However, from a practical point of view few states $\langle l|$ obey (3.16), and a much higher number of states will stay *close* to 0 during the first j time steps, where $j = \delta t/dt$. So the states $\langle l|$ that are pruned out are those for which

$$(3.18) \qquad \langle l|e^{-iLt}|\varrho_0\rangle < \epsilon, \qquad t \in [0, \delta t].$$

It is claimed [10] that the error of such an approximation is similar to what would be obtained by not considering in the Krylov expansion the contributions coming from high values of n in $L^n \varrho_0$.

The main problem of this approach is the choice of δt, i.e., the duration of the initial propagation. In fact, if we look at a full diagonalization of L we see that

$$(3.19) \qquad \varrho(t) = X e^{-iDt} X^{-1} \varrho_0,$$

so each element of $\varrho(t)$ can be written as a sum of oscillators vibrating at different $\lambda_j \in \sigma(L)$,

$$(3.20) \qquad \varrho_k(t) = \sum_{j=1}^{n} X_{[k,j]} \mu_j e^{-i\lambda_j t}, \qquad \mu_j = \sum_{l=1}^{N} X_{[j,l]} \varrho_0^l.$$

From (3.20) it is clear that to ensure that we are not pruning out the low frequencies modes we would need at least $\delta t \propto 1/\min_j\{|\lambda_j|\}$, and this could be different from the lowest Larmor frequency $\min_j\{|\omega_j^0|\}$ as this last is a quantity related to the noninteracting spins.

It is possible to have an estimate of the lowest eigenvalue of L using a technique similar to the one we used for the Chebyshev expansion (3.2). The simple choice of δt depending on λ_j rather than ω_j^0 is not enough to ensure the validity of (3.18). Within this framework we can restate (3.18) in a simpler form for a one-dimensional function and prove the following theorem.

THEOREM 3.1. *Given a function* $f(t) = \sum_{j=1}^n e^{-\lambda_j t} \mu_j$ *the condition that* $|f(t)| < \epsilon$ *for* $t \in [0, 2\pi/\min_j\{\lambda_j\}]$ *is not sufficient to ensure that* $|f(t)| < \epsilon$ *for* $t \in [0, \infty)$.

Proof. To prove Theorem 3.1 we can check that if the λ_j are not well separated, then it is possible to have a combination of similar frequencies that build up on a total frequency that is the lowest common multiplier of the initial frequencies. In the general form if for one or more ϱ_k we have the resonance condition

$$(3.21) \qquad \varrho_k(t) = \alpha e^{-i2\pi\lambda t} - \sum_j \beta^j e^{-i2\pi(\lambda + \delta^j)t},$$

and $\beta_j, \alpha > 0$ rationals such that $\alpha = \sum_j \beta_j$. $\exists$ small enough δ_j such that (s.t.) we have that $|\varrho_k(t)| < \epsilon$, $t \in [0, \delta t]$ but at the periodic maximum $t > \delta t$ s.t. $|f(t)| = \alpha + \sum_j \beta_j \gg \epsilon$. $\square$

We provide a specific counterexample for the ZTE; for instance, the function

$$(3.22) \qquad f(t) = e^{-i2\pi t} - \frac{1}{2}e^{-i2\pi(1.001)t} - \frac{1}{2}e^{-i2\pi(0.999)t}.$$

If we look at the δt evaluated looking at the lowest isolated frequency, we have that $\delta t \simeq 1$, within this time $\max|f(t)| \simeq 2 \times 10^{-5}$, but clearly the maximum amplitude of this periodic function will be 2.

In order to ensure the validity of (3.18) even with δt that depends on $\sigma(L)$ rather than the Larmor frequencies, we need to check the separation of the eigenvalues, and obviously an analysis of that kind would be as expensive as a whole simulation. While it is not clear how a resonance condition like (3.21) would arise in practice, the possibility of behavior mimicking (3.21) becomes more and more likely with an increase in system complexity.

Focusing on NMR simulations, however, the reason why ZTE performs well in practice [10] lies in the fact that the numerical comparison with other methods is not performed on $\varrho(t)$ but on the observables (2.6). In particular, for experimental reasons the only quantity that can be compared with experiments is the free induction decay (FID) signal defined as

$$(3.23) \qquad f(t) = \text{Trace}\{\varrho(t)I_p\}.$$

$\varrho(t)$ can be written as combination of Pauli matrices, and I_p is the shift up operator: $I_p = I_x + iI_y$. The Fourier transform of $f(t)$ gives then the spectrum of the sample, where each resonant frequency is revealed by a peak. Because of the relaxation effects in the experiments the shapes of the resonance peaks are not delta function, as it would be expected from (3.20), but are smoother. To mimic this smoothness also for the simulated data, it is common practice to evaluate the Fourier transform not of (3.23) but on an exponentially decaying function $\tilde{f}(t)$:

$$(3.24) \qquad \tilde{f}(t) = e^{-\xi t}f(t),$$

where the parameter ξ comes from other fitted data [32]. The main effect of (3.24) is exactly to smooth out all the low frequency modes that will differentiate $\varrho(t)$ from $\varrho_Z(t)$.

There are, however, some drawbacks:
- For this method there is no available convergence theory.
- The performance depends strongly on the initial condition ϱ_0, and on H. As expected, in our tests the size of the reduced system could change by a factor of two depending on the number of interacting spins. The reason for this effect comes from the fact that the less sparse is L, the more nonzero states may appear within the first steps, and this will make a less effective reduction L_z.
- Another reason of the strong dependence of ZTE on the initial conditions comes from (3.15); depending on the Larmor frequencies and on the time-step size, the number of evolution steps at the beginning can become large, and as this is the most expensive part of the simulation, the influence of it on the total computational costs may become important.

4. A new approach: The direct expectation values via Chebyshev. The common point of all the methods presented in the previous section is that they involve the propagation of the matrix ϱ, and for this reason they suffer from requiring that matrix operations (or matrix-vector operations) be performed at each step of calculation. Let us recall that the only quantities that can be compared between quantum simulations and experiments are the observables; for an operator $\hat{Q}$ we have

$$(4.1) \qquad \langle \hat{Q}(t) \rangle = \text{Trace}\{\varrho(t)\hat{Q}\}.$$

The first step of DEC is to perform a Chebyshev expansion as seen in section 3.2 to get

$$(4.2) \qquad \varrho(t) = e^{-iLt}\varrho_0 \approx e^{-itS}\left(\sum_{k=0}^{n_{\max}} c_k(t_D)T_k(L_s)\varrho_0\right).$$

If we insert (4.2) into (4.1), we find

$$(4.3) \qquad \langle \hat{Q}(t) \rangle = \text{Trace}\left\{\left(e^{-itS}\sum_{k=0}^{n_{\max}} c_k(t_D)T_k(L_s)\varrho_0\right)\hat{Q}\right\}.$$

By exploiting the linearity of the trace operation, we can pull out of the trace all time-dependent parts and evaluate a priori the coefficients $T_k(L_s)$. In fact, we may rewrite (4.3) as

$$(4.4) \qquad \langle \hat{Q}(t) \rangle = e^{-itS}\sum_{k=0}^{n_{\max}} c_k(t_D)\text{Trace}\left\{T_k(L_s)\varrho_0\hat{Q}\right\}.$$

This is the key equation of the DEC method as it is possible to store an array of scalar values $\tilde{R}_k(L_s) = \text{Trace}\{(T_k(L_s)\varrho_0)\hat{Q}\}$. All the time-dependent terms are just scalar values that have to be multiplied by $\tilde{R}_k(L_s)$ to get the evolution of $\hat{Q}$ at any time:

$$(4.5) \qquad \langle \hat{Q}(t) \rangle = e^{-itS}\sum_{k=0}^{n_{\max}} c_k(tD)\tilde{R}_k(L_s).$$

If more than one observable is required, it is still possible to use DEC. The only difference with the single expectation case is that we need to store different sets of $\tilde{R}_k$, one for each operator $\hat{Q}$.

4.1. Stopping criterion. The number of terms for the polynomial expansion in (3.6) depends on a prescribed tolerance ε and on the time t_D. In other methods based on Chebyshev approximation [3], the following has been suggested as a stopping

criterion:

$$(4.6) \qquad\qquad n_{\max} \quad \text{s.t.} \quad \|c_{n_{\max}}(t_D)\| < \varepsilon.$$

Because of the zeros of the Bessel function $J(t)$, at fixed time t_D, (4.6) may hold for some n, even though the expansion has not yet reached the convergence regime; it may happen that for $n_1 > n$ we have that $c_{n_1}(t_D) > c_n(t_D)$. To avoid this effect, it is enough to use as a stopping criterion a combination of two Bessel functions; the cost of such a stopping criterion is that at most we need to perform an extra iteration step (3.7). In our numerical tests we have used the following:

$$(4.7) \qquad\qquad n_{\max} \quad \text{s.t.} \quad \sqrt{\|c_{n_{\max}-1}(t_D)\|^2 + \|c_{n_{\max}}(t_D)^2\|} < \varepsilon.$$

The total time τ plays a role here, since the larger τ the more terms (T_k, c_k) will be needed to get $|c_k|$, below the threshold ε.

4.2. Computation of the expansion. In order to optimize the number of terms we evaluate, but without having to check at each step whether we have already evaluated enough terms $T_k(L_s)$, we propose to evaluate first $\langle \hat{Q}(t) \rangle$, at the final time τ, and to store the $N_{\max}$ values of $\tilde{R}_k(L_s)$. We can prove the following theorem.

THEOREM 4.1. *Given a Chebyshev expansion for an exponential $e^{-iLt}\varrho_0$ if (4.7) holds for a given time τ and small enough ε, then (4.7) holds for any time $t \leq \tau$.*

Proof. From (3.7) it is clear that c_k depends on the Bessel functions. If we look at the asymptotic behavior of the Bessel function of the first kind, for any $k \in N$, we have that, for k fixed [30],

$$(4.8) \qquad\qquad J_k(t) \sim \frac{1}{\Gamma(k+1)} \left(\frac{t}{2}\right)^k, \qquad \lim t \to 0,$$

where $\Gamma(t)$ is the Euler-Γ and for $n \in Z$ we have that $\Gamma(n) = (n-1)!$. Equation (4.8) shows that for any $k \neq 0$, in a neighborhood of $t = 0$, $J_k(t)$ is increasing monotonically with respect to t. This behavior is maintained for the whole interval $[0, j'_k]$, where j'_k is the first zero of the derivative of $J_k(t)$. It is possible to show (see [30], equation 9.5.2), that $k \leq j'_k$; consequently we can say that if (4.7) holds for a given $n_{\max}$ at τ and $\tau \leq n_{\max}$, then we are in the monotonically increasing region for $J_{n_{\max}}(t)$ and $J_{n_{\max}+1}(t)$, see Figure 4.1. In this case, (4.7) holds also for any $t \leq \tau$. □

4.3. Efficient implementation. The cost of DEC is all in the first step. Note that the cost of the evaluation of any $T_k(L_s)$ itself is roughly equivalent to that of a matrix-matrix multiplication, as per the iteration $T_{k+1}(L_s) = 2T_k L_s - T_{k-1}$, (3.8). But what is actually needed in all our calculations is $T_k \varrho_0$. Because of the linearity of the iterative expression, we may multiply T_0 and T_1 by ϱ_0 and then use (3.8) directly on $T_k \varrho_0$. The iterated operation is then just a matrix-vector multiplication. This is the well-known reason for the good computational performances of the Chebyshev expansion for large sparse matrices.

After all the $\tilde{R}_k(L_s)$ needed have been stored, it is possible to get $\langle Q(t) \rangle$ at any time $t \in [0, \tau]$ by evaluating

$$(4.9) \qquad\qquad \langle Q(t) \rangle = e^{-iDt_s} \sum_{k=1}^{n_{\max}} c_k(t_s) \tilde{R}_k(L_s),$$

where the c_k are evaluated iteratively via (3.7), and $n_{\max}$ satisfies (4.7) for tD.

For the details of the algorithm we refer to Appendix A. As a final remark we note that if the Hamiltonian is time dependent, it is not possible to apply DEC, as it is not possible anymore to isolate the time-dependent part out of the trace.
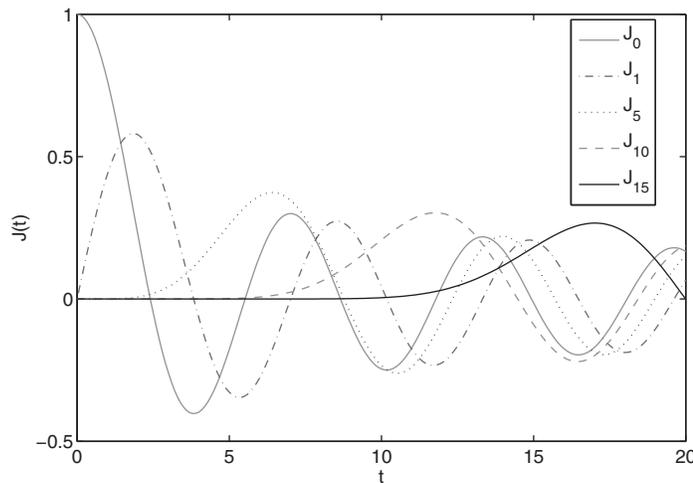
Fig. 4.1. *Example of few integer order Bessel functions of the first kind.*

4.4. Comparison of DEC with KPM. In this section we compare our DEC method with the KPM, introduced in the field of condensed matter physics as a method to compute spectral properties, such as the density of states [13, 31]. The methods appear to be similar, as both involve storing traces, but as we explain below, they are fundamentally quite different schemes. The density of states is defined, for a Hamiltonian H with spectrum E_λ, as

$$(4.10) \qquad N(E) = \sum_\lambda \delta(E - E_\lambda),$$

where $\delta(E - E_\lambda)$ is the Dirac delta.

The main idea of this method is to expand $N(E)$ into Chebyshev polynomials:

$$(4.11) \qquad N(E) = \frac{1}{\pi\sqrt{1 - x^2}} \left[\mu_0 + 2 \sum_{n=1}^{\infty} \mu_n T_n(E) \right],$$

with moments (i.e., coefficients of the expansion)

$$(4.12) \qquad \mu_n = \int_{-1}^{1} T_n(x) N(x) dx = \text{Trace}\{T_n(H)\} \simeq \lim_{N_r \to \infty} \left[\frac{1}{N_r} \sum_r \langle r | T_n(x) | r \rangle \right];$$

the μ_n values may then be evaluated once and for all and stored. For a detailed description of such a method, see, e.g., [11].

In DEC, on the other hand, which is concerned with the time-dependent evolution, the moments of the expansion in (4.2) are time dependent, but we do not need to evaluate them by performing each time the dot product of the exponential function with the polynomials T_n as the μ_n in our case are known to be the Bessel functions.

A lot of effort in the development of KPM is focussed on issues arising from having to find the μ_n (like evaluating the expansion at points x, where the original function is not differentiable); all of these techniques do not apply for DEC.

For DEC, instead, we have to compare the Bessel functions at different times to check the convergence of the series, and Theorem 4.1 has been developed for this very purpose. For this reason we store the traces of the polynomial themselves, which are time independent in the DEC case, and which may be used again at different time evaluations.

5. Numerical experiments. Nuclear spin dynamics provides a perfect example to test DEC, because the final outcome of the simulations is an observable, the FID signal, and this result is the sole important quantity, as it is the only data available from experiment.

As Hamiltonian we assumed a sum of isotropic chemical shift and the isotropic term of a pair interaction called homonuclear J-couplings, which depend on the inner product $I_j \cdot I_k$ [32]:

$$(5.1) \qquad H = -\sum_{j=1}^{n} \omega_j^0 I_j^z + \sum_{j,l=1}^{n} J_{jl} I_j \cdot I_l.$$

For the initial density matrix we set $\varrho_0 = -I_y$, that is, the result of the application of a so-called x-pulse to a sample already under the effect of a strong constant magnetic field along the z direction [32]. This is the usual initial condition when the acquisition of the signal starts.

An illustration of the structure of the Liouvillian matrix is presented in Figure 5.1. The sparsity depends on the number of interactions among the spins. In most cases the J-coupling interaction matrix J is relatively sparse. In our numerical test a strong coupling system, where J is dense and each spin interacts with every other spin, was simulated.

Because of the fact that our implementation involves only matrix-vector multiplication, techniques developed for both structured and unstructured sparse matrices may be exploited.

For comparison of computational costs we tested this method with an increasing number of spin particles using different methods to evaluate the exponential. In particular, to examine the error we compared DEC with the *expm* function of MATLAB, which uses a scaling and squaring algorithm with Padé approximation. In this way we evaluate once for all $U = e^{-iLdt}$ where dt is the step size of the simulation, and then at each time step we propagate ϱ:

$$(5.2) \qquad \varrho_{n+1} = U\varrho_n.$$

It is well known that in terms of computational costs this simplistic approach performs poorly, so we compared DEC also with the methods presented in this paper.
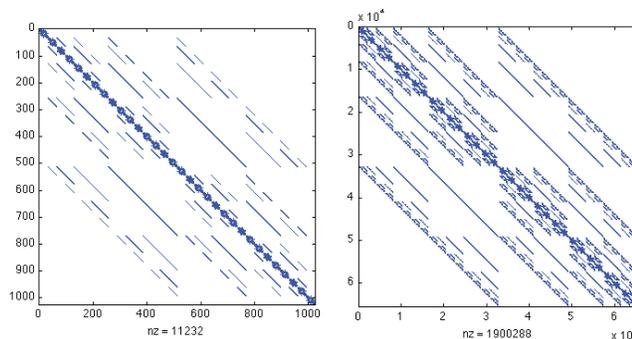

FIG. 5.1. *Left: Sparse structure of the Liouvillian* ($n = 1024, nz = 11232$) *for a system of* 5 *spins. Right: Structure of the Liouvillian* ($n = 65536, nz = 1900288$) *for a system of* 8 *spins. Both figures refer to strong coupling systems.*
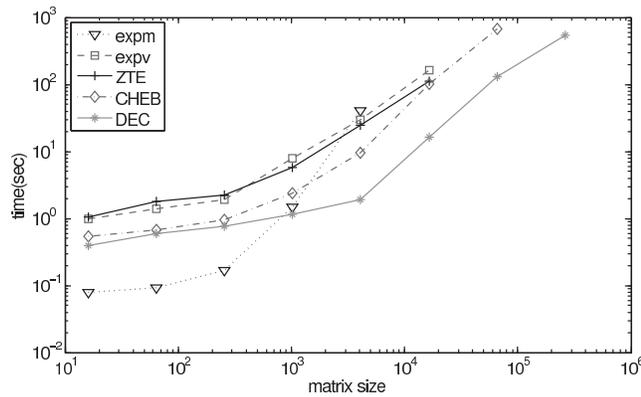
Fig. 5.2. *Logarithmic comparison of computational costs for Lanczos, ZTE, and DEC, with* $dt = 0.1$, $N = 1000$.

Table 5.1
Comparison of computational costs, CPU time in seconds, for $dt = 0.1$, $N = 1000$.

n spin	Matrix size	expm	expv	Chebyshev	ZTE	Reduced[1]	DEC
2	16	0.08	0.99	0.54	1.06	8	0.39
3	64	0.1	1.41	0.68	1.79	30	0.60
4	256	0.17	1.88	0.95	2.26	112	0.76
5	1024	1.5	7.87	2.40	5.80	420	1.14
6	4096	40.84	29.78	9.60	24.51	1584	1.93
7	16384		165.06	102.92	112.85	6006	16.11
8	65536			677.22			129.93
9	262144						542.98

[1] Size of ϱ for the reduced system.

The other three alternate methods we compared DEC with are the ZTE [10], Krylov expansion via Lanczos [33], and the Chebyshev expansion.

For the Lanczos method we have used the function *expv* of the package EXPOKIT [33] written in MATLAB.

5.1. Summary of results. The error-to-cost (measured in CPU time) diagrams are shown in Figure 5.2 and summarized in Table 5.1 for all the methods described. All the numerical tests have been performed on a Dell PowerEdge 1950 with 4GB RAM and a DualCore Intel processor running in 32bit mode. The language used is MATLAB. To avoid instabilities coming from the evaluation of the Bessel functions in these numerical tests, we set the tolerance to be $\varepsilon = 1e - 7$.

It is clear that DEC is more than an order of magnitude more efficient than the alternatives. DEC performs at its best for short time simulations (i.e., when the total time τ is small), so that we do not need to evaluate a large number of T_k, and when at the same time the use of small time step dt is required, as the cost for any step after the first is negligible.

For instance, while for all the other methods the cost of a 1000 step simulation with $dt = 0.1$ is approximately half the cost of a simulation of 1000 steps with $dt = 0.01$, for DEC there is a gain of almost an order of magnitude; see Figure 5.3.

6. Conclusion. In this article we have presented a new method for simulation of an observable in a mixed quantum system. By expanding the exponential of the Hamiltonian in Chebyshev polynomials, and exploiting the trace operation performed when evaluating the expectation value of an observable, it is possible to reduce the
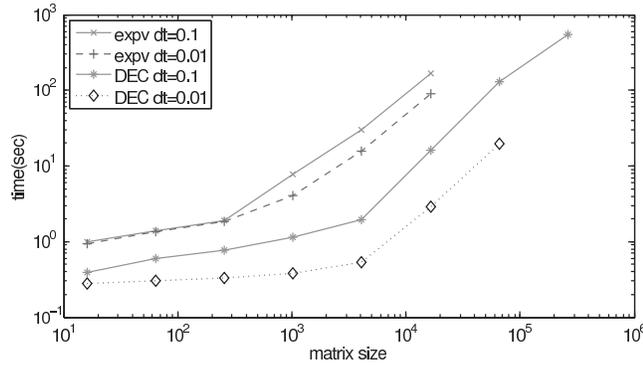
FIG. 5.3. *Logarithmic comparison of computational costs for Lanczos and DEC when simulating for the same number of total steps N but with different step size dt. $N = 1000$ in both the cases.*

evolution of any observable to a one-dimensional function that can be evaluated directly.

We also presented an optimal algorithm to perform such a calculation and showed how this new method can easily compete in terms of computational costs with a variety of model reduction approaches, while maintaining the approximation errors below a chosen threshold.

Moreover, we have discussed the ZTE method in some detail, demonstrating why it cannot be reliable in every case and also clarifying why it performs well in the setting of solid state NMR simulation.

Appendix A. We provide here a detailed description of the algorithm. T_k^M indicates the matrix of size n built from the vector T_k^V as in (2.5) *inputs: L (an $n \times n$ Hermitian matrix, ϱ_0 vector of dimension n*

outputs: expectation value $f(t)$ evaluated at $(j\Delta t)$, $j = 1, \ldots, N$.

1. *evaluate α,β via Lanczos s.t. $\alpha \leq \sigma(L) \leq \beta$*
2. *scale L and get L_s*
3. *evaluate $T_0^V = \mathrm{Id}\varrho_0$, $T_1^V = L_s\varrho_0$*
4. **while** $\|c_k\| < \varepsilon$
5. $c_k = (2 - \delta_{k,0})(-i)^k J_k(\tau S)$, $\tau = $ *total time*
6. $T_{k+1}^V = 2L_s T_k^V - T_{k-1}^V$
7.   *store $\tilde{R}_k = \mathrm{Trace}\{(T_k^M)\hat{Q}\}$*
8. **end**
9. **for** $j = 1 : \mathrm{N}$
10.   *reevaluate the c_k at different time $t = jdt$*
11. $f(j) = \sum_{k=1}^{n_{\max}^j} c_k \tilde{R}_k$
12. **end**

Acknowledgments. G.M. is very grateful to Arieh Iserles for useful suggestions at the start of this work. We are also thankful to the anonymous referees for their helpful comments.

REFERENCES

[1] H. TAL–EZER AND R. KOSLOFF, *An accurate and efficient scheme for propagating the time-dependent Schrödinger equation*, J. Chem. Phys., 81 (1984), pp. 3967–3971.
[2] A. VIJAY AND H. METIU, *A polynomial expansion of the quantum propagator, the Green's function and the spectral density operator*, J. Chem. Phys., 116 (2001), pp. 60–68.

[3] M. Veshtort and R. G. Griffin, *SPINEVOLUTION: A powerful tool for the simulation of solid and liquid state NMR experiments*, J. Magn. Reson., 178 (2006), pp. 248–282.

[4] G. Moro and J. H. Freed, *Calculation of ESR spectra and related Fokker–Plank forms by the use of the Lanczos algorithm*, J. Chem. Phys., 74 (1981), pp. 3757–3773.

[5] T. J. Park and J. C. Light, *Unitary quantum time evolution by iterative Lanczos reduction*, J. Chem. Phys., 85 (1986), pp. 5870–5876.

[6] K. Kormann, S. Holmgren, and H. O. Karlsson, *Accurate time propagation for the Schrödinger equation with an explicitly time-dependent Hamiltonian*, J. Chem. Phys., 128 (2008), pp. 184101–184112.

[7] R. H. Hardin and F. D. Tappert, *Applications of the split-step Fourier method to the numerical solution of nonlinear and variable coefficient wave equations*, SIAM Rev. Chronicles, 15 (1973), p. 423.

[8] C. Leforestier et al., *A comparison of different propagation schemes for the time-dependent Schrödinger equation*, J. Comput. Phys., 94 (1991), pp. 59–80.

[9] C. Lubich, *From quantum to classical molecular dynamics: Reduced models and numerical analysis*, Zurich Lectures in Advanced Mathematics, Zurich, European Mathematical Society, 2008.

[10] I. Kuprov, *Polynomially scaling spin dynamics* II: *Further state-space compression using subspace techniques and zero track elimination*, J. Magn. Reson., 195 (2008), pp. 45–51.

[11] A. Weisse, G. Wellein, A. Alvermann, and H. Fehske, *The kernel polynomial method*, Rev. Modern Phys., 78 (2006), pp. 275–306.

[12] H. P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems*, Oxford University Press, London, 2002.

[13] H. Röder and R. N. Silver, *Kernel polynomial method for a nonorthogonal electronic-structure calculation of amorphous diamond*, Phys. Rev. B, 55 (1997), pp. 15382–15385.

[14] C. Moler and C. F. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty–five years later*, SIAM Rev., 45 (2003), pp. 3–49.

[15] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 29 (1992), pp. 209–228.

[16] C. Bekas, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, *Computing charge densities with partially reorthogonalized Lanczos*, Comput. Phys. Comm., 171 (2005), pp. 175–186.

[17] M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925.

[18] J. C. Schulze, P. J. Schmid, and J. L. Sesterhenn, *Exponential time integration using Krylov subspaces*, Internat. J. Numer. Methods Fluids, 60 (2009), pp. 591–609.

[19] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008.

[20] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, 45 (1950), pp. 255–282.

[21] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore, MD, 1996.

[22] J. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*, Birkhäuser, Boston, 1985.

[23] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, *Self-consistent–field calculations using Chebyshev-filtered subspace iteration*, J. Comput. Phys., 219 (2006), pp. 172–184.

[24] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK USERS GUIDE: Solution of Large Scale Eigenvalue Problem by Implicit Restarted Arnoldi Methods*, SIAM, Philadelphia, 1997.

[25] L. Bergamaschi, M. Caliari, and M. Vianello, *Efficient computation of the exponential operator for discrete* 2d *advection-diffusion equations*, Numer. Linear Algebra Appl., 10 (2003), pp. 271–289.

[26] F. W. J. Olver and D. J. Sookne, *Note on backward recurrence algorithms*, Math. Comp., 26 (1972), pp. 941–947.

[27] C. W. Clenshaw, *Chebyshev Series for Mathematical Functions*, Mathematical Tables, 5, H.M. Stationery Office, London, 1962.

[28] M. Hochbruck and C. Lubich, *Exponential Integrators for Quantum-Classical Molecular Dynamics*, BIT Numer. Math., 39 (1999), pp. 620–645.

[29] M. Bak, J. T. Rasmussen, and N. C. Nielsen, *SIMPSON: A general simulation program for solid-state NMR spectroscopy*, J. Magn. Reson., 147 (2000), pp. 296–330.

[30] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1964.

[31] Y. Saad, J. R. Chelikowsky, and S. M. Shontz, *Numerical methods for electronic structure calculations of materials*, SIAM Rev., 52 (2010), pp. 3–54.

[32] M. H. Levitt, *Spin Dynamics*, Wiley, Chichester, 2001.

[33] R. B. Sidje, *EXPOKIT: A software package for computing matrix exponentials*, ACM Trans. Math. Software, 24 (1998), pp. 130–156.