Time evolution with the density-matrix renormalization-group algorithm: A generic implementation for strongly correlated electronic systems

G. Alvarez,[1] Luis G. G. V. Dias da Silva,[2] E. Ponce,[3] and E. Dagotto[4]

[1]*Computer Science and Mathematics Division and Center for Nanophase Materials Sciences, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, USA*

[2]*Instituto de Física, Universidade de São Paulo, Caixa Postal 66318, 05315-970 São Paulo, Brazil*

[3]*Electrical and Computer Engineering and Computer Science Department, Polytechnic University of Puerto Rico, San Juan, Puerto Rico 00919, USA*

[4]*Department of Physics and Astronomy, University of Tennessee, Knoxville, Tennessee 37996, USA, and Materials Science and Technology Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, USA*

(Received 23 March 2011; revised manuscript received 17 August 2011; published 21 November 2011)

A detailed description of the time-step-targeting time evolution method within the density-matrix renormalization-group algorithm is presented. The focus of this publication is on the implementation of the algorithm and its generic application. The case of one-site excitations within a Hubbard model is analyzed as a test for the algorithm, using open chains and two-leg ladder geometries. The accuracy of the procedure in the case of the recently discussed holon-doublon photo excitations of Mott insulators is also analyzed. Performance and parallelization issues are discussed. In addition, the full open-source code is provided as supplementary material.

DOI: 10.1103/PhysRevE.84.056706 PACS number(s): 02.70.−c, 05.60.Gg

I. INTRODUCTION

The accurate calculation of time-dependent quantum observables in correlated electron systems is crucial to achieve further progress in this field of research, where the vast majority of the computational efforts in the past have mainly focused on time-independent quantities. In transition metal oxides and nanostructures, time evolution is at the core of the study of a broad array of physical phenomena such as electronic transport, optical excitations, and nonequilibrium dynamics in general. Accurate studies of time-dependent properties will advance the fields of spintronics, low-dimensional correlated systems, and possibly quantum computing as well. For a list of recent efforts on these topics by our group and a concomitant set of references, see [1–5] and references therein.

The purpose of this paper is to present an explicit implementation of the time evolution within the density-matrix renormalization-group (DMRG) method [6,7]. Knowledge of the widely discussed DMRG algorithm to compute static observables is here assumed. Readers not familiar with the method are referred to published reviews [8–11] and to the original publications [6,7] for further information. Our main focus here is to provide a detailed description of the implementation of the time-step-targeting [12] algorithm and the discussion of a few applications. We also provide full open-source codes and additional documentation to use those codes [13].

The present work builds upon considerable previous efforts by other groups. In particular, we mainly follow Manmana *et al.* [14]. The time-step targeting procedure was also reviewed in Ref. [15]. Since it would not be practical to describe in a short paragraph the considerable progress achieved in this field of research in recent years, the interested reader is encouraged to consult the aforementioned reviews along with, for example, Ref. [16] for a historical account of the development of the methods used in the present publication.

Since our aim is to discuss a generic method applicable to any Hamiltonian and lattice geometry, here we do not discuss or implement the Suzuki-Trotter method [17,18] but focus instead only on the Krylov method [19] for the time evolution, as described in Ref. [14]. Because its implementation can be isolated from the rest of the computer code, the Krylov method can be applied in a generic way to most models and geometries without changes, which is not the case for other methods, such as the Suzuki-Trotter method. The readers interested in the Suzuki-Trotter method should consult the Algorithms and Libraries for Physics Simulations (ALPS) project [20], where the time-evolving block decimation is implemented [17].

Our goal is to compute observables of the form

$$\langle\phi_1|\mathrm{e}^{iHt}A_{0,\pi(0)}A_{1,\pi(1)}\ldots A_{a-1,\pi(a-1)}\mathrm{e}^{-iHt}|\phi_2\rangle, \qquad (1)$$

where $|\phi_1\rangle$ and $|\phi_2\rangle$ denote generic quantum many-body states. This category of observables is sufficiently broad to encompass most time-dependent correlations, as represented by a number a of *local* operators $A_{0,\pi(0)}\ A_{1,\pi(1)}\ \ldots\ A_{a-1,\pi(a-1)}$ acting on sites $\pi(0),\pi(1),\pi(a-1)$ of a finite lattice, where $\pi(i)$ denotes the lattice site on which the operator $A_{i,\pi(i)}$ acts and the extra index i indicates that the operator can be different at each site.

An immediate application of this formalism and code is the study of the evolution of a system that is brought out of equilibrium by a sudden excitation. This sudden excitation can be simulated by the state of the system given by the vectors $|\phi_1\rangle$ and $|\phi_2\rangle$. Depending on the problem, sometimes it is more convenient to assume that the states remain unchanged but that it is the Hamiltonian $H(t)$ that changes with time. Another application entails the computation of time-dependent properties of systems in equilibrium, such as the Green's function $G_{ij}(t)$.

The organization of this paper is the following. Section II explains in detail the Krylov method for time evolution within the DMRG algorithm, focusing on its implementation. Section III A applies the method to the case of one-site excitations, showing a simple picture of the accuracy of the method. Section III B extends to two-leg ladder geometries the results obtained using tight-binding chains, employing holon-doublon excitations for the specific study. Performance issues are studied in Sec. IV, while Sec. V summarizes our results. The first appendix contains derivations of exact results used in the paper. The last appendix explains briefly the use of the code and points to its documentation.

II. METHOD AND IMPLEMENTATION

A. Lanczos computation of the unitary evolution

To carry out the previously described program[1] of computing observables of the type given by Eq. (1), the first goal is to calculate $|\phi(t)\rangle \equiv \exp(-iHt)|\phi\rangle$. The Lanczos technique [21] provides a method to tridiagonalize H into $V^\dagger T V$, where T is tridiagonal and V is the matrix of Lanczos vectors. If the number of those Lanczos vectors is n_l and the Hilbert space for $|\phi\rangle$ has size n, then T is a square matrix of size $n_l \times n_l$ and V is, in general, a rectangular matrix of size $n_l \times n$.

$V^\dagger T V$ cannot be used *everywhere* as a substitution for H without inducing large errors, but if we start the Lanczos procedure [22] with the vector $|\phi\rangle$ (instead of using a random vector as is most frequently done), then we can use that substitution accurately in the multiplication $H|\phi\rangle$. However, this is not enough here, because we need to compute the exponential of H. For this purpose, it has been shown [23–25] that $|\phi(t)\rangle \approx V^\dagger \exp(iTt)V|\phi\rangle$ with an accuracy that increases as time t decreases for fixed n_l. We will assume that we have taken t small enough such that we can regard the expression above to have the accuracy of the Lanczos technique, which is usually high. In other words, if t is small enough, we will assume that using $V^\dagger \exp(iTt)V|\phi\rangle$ as a replacement for $\exp(-iHt)|\phi\rangle$ is not worse than using $V^\dagger T V|\phi\rangle$ as a replacement for $H|\phi\rangle$. This will be enough for our purposes, but for details on the scaling and bounds of the errors made in each case as a function of t and n_l, see, for example, Refs. [23] and [24].

Since we started the Lanczos recursive procedure with the vector $|\phi\rangle$, then $\sum_j V_{j',j}|\phi\rangle_j \propto \delta_{j',0}$. Finally, we need to diagonalize $T = S^\dagger D S$ into a $n_l \times n_l$ diagonal matrix D with diagonal elements $d_{l'}$. This last step is not computationally expensive, since T is a $n_l \times n_l$ matrix, as was noted before.

Denoting by $|\phi\rangle_j$ the j component of the vector $|\phi\rangle$, we arrive at

$$|\phi(t)\rangle_i = \sum_{k,l,j} V^\dagger_{i,k} S^\dagger_{k,l} e^{-id_l t} S_{l,0} V_{0,j} |\phi\rangle_j, \tag{2}$$

for small times t, where the equal sign should be understood to be valid within the accuracy of the Lanczos technique [25]. How to deal with larger times t will be explained in Sec. II C.

B. Targeting states with the DMRG algorithm

It appears that now we could use Eq. (2) to compute $|\phi_1(t)\rangle$ from some vector $|\phi_1\rangle$ and likewise $|\phi_2(t)\rangle$ starting from some vector $|\phi_2\rangle$. Then we would just apply the operators $A_{0,\pi(0)}\ A_{1,\pi(1)} \ldots A_{a-1,\pi(a-1)}$ to those states within a DMRG procedure to achieve our aim of computing Eq. (1). However, the DMRG algorithm is not immediately applicable to arbitrary states, such as $|\phi_1(t)\rangle$, and was originally developed to compute the ground state of the Hamiltonian instead.

This difficulty has been successfully overcome (see [9] and references therein) by redefining the reduced density matrix of the left block $\mathcal{L}$ as

$$\rho^{\mathcal{L}}_{\alpha,\alpha'} = \sum_{\beta\in\mathcal{R}} \sum_l \omega_l \Phi^\dagger(l)_{\alpha,\beta} \Phi(l)_{\alpha',\beta}, \tag{3}$$

where α and α' label states in the left block $\mathcal{L}$, β label those of the right block $\mathcal{R}$, and $\{\Phi(l)\}_l$ is a set of as of yet unspecified states of the superblock $\mathcal{L} \cup \mathcal{R}$. The states $\Phi(l)$ are said to be *targeted* by the DMRG algorithm. Because of their inclusion in the reduced density matrix, these states will be obtained with a precision that scales similarly to the precision of the ground state in the static formulation of the DMRG. The relevance of the weights ω_l appearing in Eq. (3) will be discussed in Sec. II C.

Which are the states $\Phi(l)$ that need to be included in Eq. (3) to compute Eq. (1)? $|\phi_1(t)\rangle$ and $|\phi_2(t)\rangle$ are certainly needed. Since observables that include the ground state are ubiquitous, the ground state of the Hamiltonian needs to be targeted as well in most cases, but additional states need to be included in order to evolve to larger times, as we will now explain.

Our implementation follows the time-step-targeting procedure of Ref. [15]. We now introduce a small time τ such that, for all $t \leqslant \tau$, Eq. (2) is accurate in the sense defined in, for example, Ref. [23]. We consider a set of n_v times $\{t_x\}_x$, $x = 0,1,\ldots,n_v - 1$, such that $t_x < t_{x+1}$, $t_0 = 0$, and $t_{n_v-1} = \tau$. For simplicity, we assume from now on that $|\phi_1\rangle = |\phi_2\rangle \equiv |\phi\rangle$ in Eq. (1). The state $|\phi\rangle$ is defined by the particular physics problem under investigation, and we will consider particular examples in Sec. III. States $|\phi(t_x)\rangle$ for each $0 \leqslant x < n_v$ can be obtained accurately from $|\phi\rangle$ by using Eq. (2) since $t_x \leqslant \tau$. To compute Eq. (1) for all $t \leqslant \tau$, we target the n_v states $|\phi(t_x)\rangle$ and the ground state $|\psi\rangle$ as well.

At this point, it is instructive to consider a concrete class of states $|\phi\rangle$. In a large class of problems, these states are related to the ground state $|\psi\rangle$ of the Hamiltonian by

$$|\phi\rangle = B_{b-1,\pi'(b-1)} \ldots B_{1,\pi'(1)} B_{0,\pi'(0)} |\psi\rangle, \tag{4}$$

for b *local* operators $B_{0,\pi'(0)}, B_{1,\pi'(1)}, \ldots, B_{b-1,\pi'(b-1)}$ acting on sites $\pi'(0),\pi'(1),\ldots,\pi'(b-1)$ of the superblock, where $\pi'(i)$ denotes the lattice site on which the operator $B_{i,\pi'(i)}$ acts, as explained below. The extra index i indicates that the operators B can be different on different sites. Physical examples of the operators B will be given in Sec. III.

The sites $\pi'(0)$, $\pi'(1)$, ..., $\pi'(b-1)$ in Eq. (4) [as well as sites $\pi(0)$, $\pi(1)$, ..., $\pi(a-1)$ in Eq. (1)] will be considered ordered in the way in which they appear as central sites [9] for the DMRG finite algorithm, as illustrated in Fig. 1. At a given stage of the computational procedure, if the central site of the DMRG algorithm is $\pi'(0)$, then $|\phi_0^{\rm partial}\rangle \equiv B_{0,\pi'(0)}|\psi\rangle$ can be obtained. Next, we proceed to the following site, and so on, until we reach site $\pi'(1)$ and apply $B_{1,\pi'(1)}$, *i.e.*, $|\phi_{1,0}^{\rm partial}\rangle \equiv B_{1,\pi'(1)}|\phi_0^{\rm partial}\rangle$, eventually reaching site $\pi'(b-1)$ to complete the computation of $|\phi\rangle$, given by Eq. (4). Since in cases of physical interest the operators B are either bosons or fermions,

[1] This section is inspired by handwritten notes of Schollwöck from his talk at IPAM, University of California at Los Angeles, found at [https://www.ipam.ucla.edu/publications/qs2009/qs2009_8384.pdf].

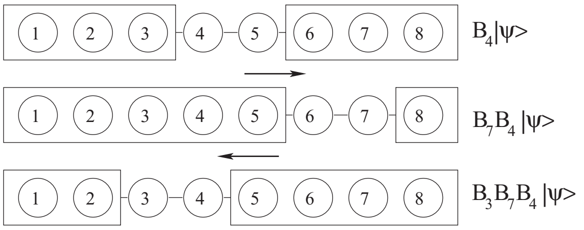

FIG. 1. Example of a state $|\phi\rangle$ given by Eq. (4). The order in which the on-site operators are applied will depend on the order in which the sites appear as "central sites." In the example above, $b = 3$, and the operators are acting on sites 3, 4, and 7. In the sweeping procedure shown, the order will be $\pi'(0) = 4, \pi'(1) = 7, \pi'(2) = 3$, that is, $|\phi\rangle = B_3 B_7 B_4 |\psi\rangle$.

a reordering is always possible due to their commutativity or anticommutativity, yielding at most a minus sign.

As the DMRG algorithm sweeps the entire lattice, the central sites change, leading to modified Hilbert spaces. Therefore, a procedure is required to "transport" the states $|\phi\rangle$ from one space to another. It is known [14] that the transformation needed to "transport" these states is the so-called *wave-function transformation*, proposed by White [26] first in the context of providing a guess for the initial Lanczos vector to speed up the algorithm but later found to be applicable to other subalgorithms.

After the state $|\phi\rangle$ in Eq. (4) is computed, the DMRG algorithm operates for a few extra steps to better converge all states $|\psi(t_x)\rangle$, $\forall t_x \leqslant \tau$. These states and all DMRG transformations can be saved to disk and later used to compute the observables in Eq. (1).

C. Evolution to arbitrary times

What happens to Eq. (1) for larger times, that is, for times $t > \tau$? Noting that $|\phi(\tau + \tau)\rangle = \exp(iH\tau)|\phi(\tau)\rangle$, we can apply Eq. (2) to $|\phi(\tau)\rangle$, which in general reads [14]

$$|\phi(t+\tau)\rangle_i = \sum_{k,k',l,l',j} V^{\dagger}_{i,k} S^{\dagger}_{k,l} e^{-id_l\tau} S_{l,0} V_{0,j} |\phi(t)\rangle_j, \qquad (5)$$

where again $|\phi(t)\rangle_j$ denotes the jth component of the vector $|\phi(t)\rangle$. Then, we proceed by *targeting* the states $\{|\phi(t_x + \tau)\rangle\}_x$ for some time until they are converged. By applying this procedure recursively, we reach arbitrary times t as we sweep the finite lattice back and forth and target $\{|\phi(t_x + t)\rangle\}_x$ in the general case.

The speed of time advancement in the algorithm is controlled by two opposing computational constraints. If we advance times too fast by applying Eq. (5) too often, then convergence might not be achieved or we might not have had the chance to visit all sites $\pi(0)$, $\pi(1)$, ... , $\pi(a-1)$ to compute Eq. (1). Conversely, advancing too slowly would increase computational cost but produce no additional data. Remember that when not advancing in time, states $|\phi(t + t_x)\rangle_x$ are *wave function transformed*, as explained in the previous section.

We now explain the choice of the weights [15] that appear in Eq. (3). Assume n to be the total number of states to be targeted, including the ground state. To give them more prominence, we have chosen a weight of 2Ω for the ground state and for the $\Phi(l)$ vectors at the beginning and end of the τ interval. We have chosen a weight of Ω for the rest. Then $2\Omega \times 3 + \Omega(n-3) = 1$ implies $\Omega = 1/(n+3)$. The algorithm does not appear much dependent on the choice of weights. However, irrespective of what the choice actually is, all mentioned states must have nonzero weights to avoid loss of precision for one or more states.

D. Overview of the implementation

The DMRG++ code was introduced in Refs. [27] and [28]. The extension of the code to handle the time evolution and computation of observables of the type represented by Eq. (1) was carried out with minimal refactoring. A *targeting* interface was introduced, with two concrete classes, GroundStateTargetting, and TimeStepTargetting. The first handles the usual case and is used even in the presence of time evolution during the so-called infinite DMRG algorithm and during the finite algorithm before encountering the first site $\pi'(0)$ in Eq. (4).

A call to target.evolve(...) handles (i) the computation of the vectors $\{|\phi(t + t_x)\rangle\}_x$ as needed and (ii) their time evolution or, depending on the stage of the algorithm, their *wave-function transformation*. When the target object belongs to the TimeStepTargetting class, the actual implementation of these tasks is performed by the member function evolve(...). When the target object is of class GroundStateTargetting, the evolve(...) function is empty. The call to this function is always done immediately after obtaining the ground state $|\psi\rangle$ for that particular step of the DMRG algorithm. File TimeStepTargetting.h is documented in place using literate programming [29]. Further details about how to run the DMRG++ code and how to specify its input file are given in Ref. A.

III. EXAMPLES

A. One-site excitations

To test the accuracy of the time-dependent DMRG approach explained in the previous section, we consider first the following problem. Consider the tight-binding model $H_0 = \sum_{i,j,\sigma} t_{ij} c^{\dagger}_{i\sigma} c_{j\sigma}$, with t_{ij} a symmetric matrix, with the observable [31] we wish to calculate being

$$X_{ij\uparrow}(t) = \langle\psi| c^{\dagger}_{i\uparrow} e^{iH_0 t} n_{j\uparrow} e^{-iH_0 t} c_{i\uparrow} |\psi\rangle, \qquad (6)$$

where $|\psi\rangle$ is the ground state of H_0. (We keep the usual notation t_{ij} for the matrix of hopping integrals in the context of a tight binding model [32], even though t is also used to denote time here.)

This is equivalent to taking $b = 1$, $B = c_{\uparrow}$, and $\pi'(0) = i$ in Eq. (4) and $a = 1$, $\pi(0) = j$, and $A = n_{\uparrow}$ in Eq. (1). The physical interpretation for $X_{ij\uparrow}(t)$ is then clear: It provides the time-dependent expectation value of the charge density $\langle n_{j\uparrow}\rangle(t)$ at site j over a state that, at time $t = 0$, is defined by creating a "hole-like" excitation in state $|\psi\rangle$ at site i. We assume that site i has been specified and is kept fixed throughout this discussion.

$X_{ij}(t)$ can be expressed in terms of the eigenvectors and eigenvalues of t_{ij}. For a half-filled lattice, we have $X_{ij}(t) = R_i R_j - |T_{ij}(t)|^2$, where R_i and T_{ij} are given in Appendix. Then, $X_{ij}(t)$ can be calculated numerically and compared to

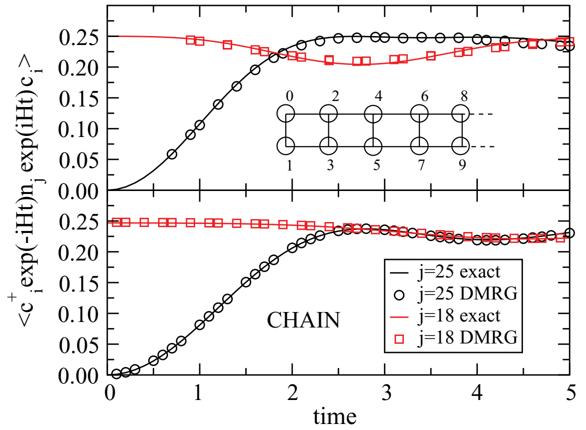

FIG. 2. (Color online) Observable $X_{ij\uparrow}(t)$, Eq. (6), for Hamiltonian H_0, with $i = 25$ and j as indicated, on two geometries: (a) a 25×2 ladder and (b) a 50-site chain. Circles and squares represent DMRG results, and solid lines show exact results. Inset: Labeling of sites on a two-leg ladder. The one-site excitation (creation of a "hole") was applied on site $i = 25$. Chain sites are labeled from left to right, starting at 0.

DMRG results for this model on a 50-site chain and on a 25×2 ladder; see Fig. 2.

On the chain, the number of states kept for the DMRG algorithm was set to $m = 200$, which was found to give good accuracy for the ground-state energy. On the ladder, $m = 400$ was used, which is a typical [33] m value to achieve good accuracy for the ground-state energy and static properties. For instance, in both cases, DMRG gives $X_{ii}(t = 0) = 0$ and $X_{ij}(t = 0) = 1/4$ for $i \neq j$, as expected. As shown in Fig. 2, the use of these values of m yields an accurate time evolution.

A full lattice sweep is done before advancing in time, which equals $n/2 - 1$ DMRG iterations, with n the total number of sites. As discussed in Refs. [14] and [15], care must be taken in choosing the time step and the number of retained states m during time evolution in order to avoid the "runaway" of the results after a certain "runaway time" t_R. This is especially critical for the ladder: We find that $m = 400$ and a time step of 0.1 results in $t_R > 4$.

B. Holon-doublon

Currently there is considerable interest in studying the feasibility of a new class of materials—the Mott insulators—for their possible use in photovoltaic devices and oxide electronics in general. The crucial question under study is whether charge excitations in the Mott insulator will be able to properly transfer the charge into the metallic contacts, thus establishing a steady-state photocurrent. Answering this question will require computation of the out-of-equilibrium dynamics and the time evolution of the excitonic excitations produced by the absorption of light by the material.

The electron and hole created by light absorption are modeled by the state [34]

$$|\phi\rangle \equiv c_{i\sigma}(1 - n_{i\bar{\sigma}})c^\dagger_{j\sigma'} n_{j\bar{\sigma}'}|\psi\rangle, \tag{7}$$

where $|\psi\rangle$ is the ground state, σ and σ' are spin indices, and $\bar{\sigma} = 1 - \sigma$ denotes the spin opposite to σ. A sum over σ and σ' is assumed in the equation above. This is equivalent to taking $b = 2$, $B_0 = c^\dagger_\sigma n_{\bar{\sigma}'}$, $B1 = c_\sigma(1 - n_{\bar{\sigma}})$, $\pi'(0) = j$, and $\pi'(1) = i$ in Eq. (4). We assume that the sites i and j of the lattice have been specified and will remain fixed throughout this discussion.

To model a Mott insulator, we consider the Hubbard Hamiltonian [35–37]

$$\hat{H} = \hat{H}_0 + U \sum_i \hat{n}_{i\uparrow}\hat{n}_{i\downarrow}, \tag{8}$$

where the notation is as in Ref. [34], and we drop the hat from the operators from now on. The hopping matrix t corresponds either to an open chain or a two-leg ladder in the studies below.

1. *Density*

The time-dependent density at site p of state Eq. (7) is

$$O_{j,i,p,\uparrow}(t) = \frac{\langle\phi|e^{iHt} n_{p\uparrow} e^{-iHt}|\phi\rangle}{\langle\phi|\phi\rangle}, \tag{9}$$

which amounts to taking $a = 1$, $A = n_\uparrow$, and $\pi(0) = p$ in Eq. (1). Consider $O_{j,i,p}(t) = \sum_\sigma O_{j,i,p,\sigma}(t)$. In the case of $U = 0$ and half-filling we have

$$O_{j,i,p}(0) \equiv O_{j,i,p,\uparrow}(0) + O_{j,i,p,\downarrow}(0) = \begin{cases} 0 & \text{if} \quad p = i \\ 2 & \text{if} \quad p = j \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

The observable we test in this section is $\langle\Psi_e|n_p|\Psi_e\rangle$, which has a similar physical interpretation as $X_{ij}(t)$ [Eq. (6)] in the holon-doublon case. Results for $U = 0$ and $U = 10$ are shown in Fig. 3. At $t = 0$, the values of $\langle\Psi_e|n_p|\Psi_e\rangle$ given by Eq. (10) hold true in the $U \neq 0$ case at half-filling. The time evolution for interacting and noninteracting cases are quite distinct, however, as in the case of the chain (see, e.g., Ref. [38]).

Readers might want to know why we emphasize the noninteracting $U = 0$ case. One obvious advantage of the $U = 0$ case is that we can test the Krylov method, and indirectly the accuracy of the DMRG, against exact results. In the case of the ladder geometry, the $U = 0$ benchmark is particularly useful for large system sizes as it guides the choice of time-evolution parameters in the DMRG runs: for a 50-site ladder, for instance, using $m = 400$ and a time step equal to 0.1, small deviations from the exact calculations can appear for $t > 3.5$ on certain sites. In addition, the U term, at least when on site, is not a major source of efficiency problems for the DMRG algorithm.

To test our results for $U \neq 0$, we have compared them to the Suzuki-Trotter method (not shown). We have also computed the small time expansion, and this is shown in Fig. 4.

2. *Double occupation*

The double occupation of state Eq. (7) is [34]

$$N_d(j,i,p,t) \equiv \frac{\langle\phi|e^{iHt} n_{p\uparrow} n_{p\downarrow} e^{-iHt}|\phi\rangle}{\langle\phi|\phi\rangle} \tag{11}$$

and amounts to taking $a = 1$, $A = n_\uparrow n_\downarrow$, and $\pi(0) = p$ in Eq. (1).

Summarizing the *operator* equations obtained in Sec. III B 1, $n_{i\uparrow}A_{ij} = 0$, and $\bar{n}_{j\uparrow}A_{ij} = 0$, where $A_{ij} = c_{i\sigma}$

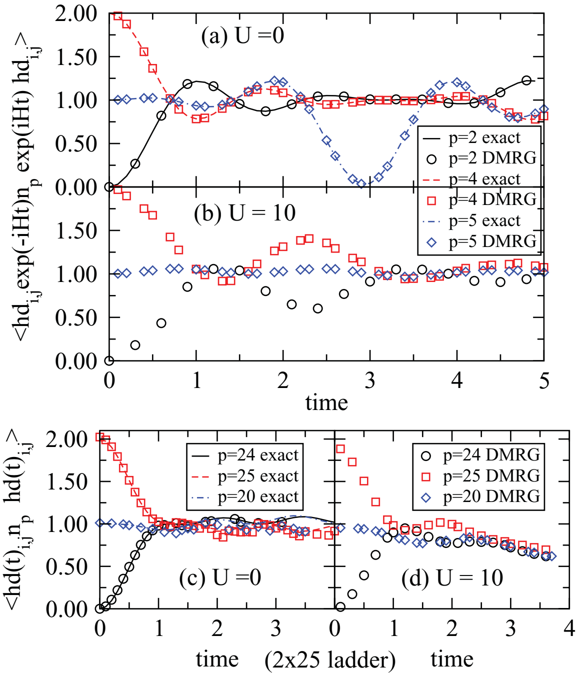

FIG. 3. (Color online) (a), (b) Density at site p of the holon-doublon state Eq. (7) as a function of time for (a) $U = 0$ and (b) $U = 10$. A 2×4 ladder with $t_x = 1$ and $t_y = 0.5$ was used, containing four up and four down electrons. The holon-doublon operator was applied at $i = 2$, $j = 4$. (c), (d) Same as for panels (a) and (b) but now on a 2×25 ladder with 50 electrons. In this case, the holon-doublon operator was applied at $i = 25$, $j = 24$. In the vertical axes labels, $\mathrm{hd}_{ij}(t)$ denotes the state obtained by acting with a holon-doublon operator at sites i and j over the ground state and evolving that state to time t.

$(1 - n_{i\bar{\sigma}})c^\dagger_{j\sigma'}n_{j\bar{\sigma}'}$, $\bar{n} = 1 - n$, and these equations also hold if we replace ↑ by ↓. Then $N_d(j,i,i,t = 0)\langle\phi|\phi\rangle = \langle A^\dagger_{ij} n_{i\uparrow} n_{i\downarrow} A_{ij}\rangle = 0$, and

$$\begin{aligned} N_d(j,i,j,t = 0)\langle\phi|\phi\rangle &= \langle A^\dagger_{ij} n_{j\uparrow} n_{j\downarrow} A_{ij}\rangle \\ &= \langle A^\dagger_{ij} n_{j\uparrow}(1 - \bar{n}_{j\downarrow}) A_{ij}\rangle \\ &= \langle A^\dagger_{ij} n_{j\uparrow} A_{ij}\rangle \equiv O_{j,i,j,\uparrow}\langle\phi|\phi\rangle. \end{aligned}$$

DMRG results for $U = 0$ and $U = 10$ are shown in Fig. 5. Also shown are exact results for $U = 0$. At $t = 0$, the double occupation at the doublon ($p = j$) and holon ($p = i$) sites are, respectively, $N_d(j,i,p = j,t = 0) = 1$ and $N_d(j,i,p = i,t = 0) = 0$ for both interacting and noninteracting cases. At the doublon site, the double occupation has a characteristic oscillating decay caused by the dynamics of the holon-doublon pair within the system, also observed for the chain case [38].

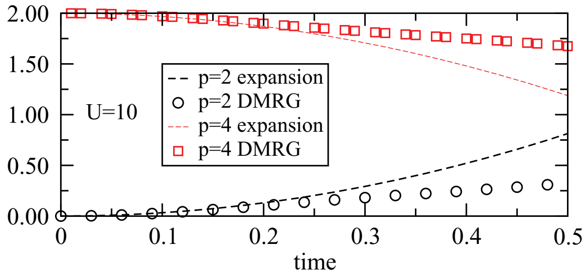

FIG. 4. (Color online) Comparison of $O_{j,i,p}(t)$ computed with DMRG (circles and squares) and with a $t \to 0$ expansion. For the latter $O_{j,i,p}(t) = O_{j,i,p}(0) + a_{j,i,p}t^2 + O(t^4)$, where $a_{j,i,p} \equiv \langle\phi|Hn_pH - H^2n_p|\phi\rangle$. Same parameters as in Fig. 3(b).

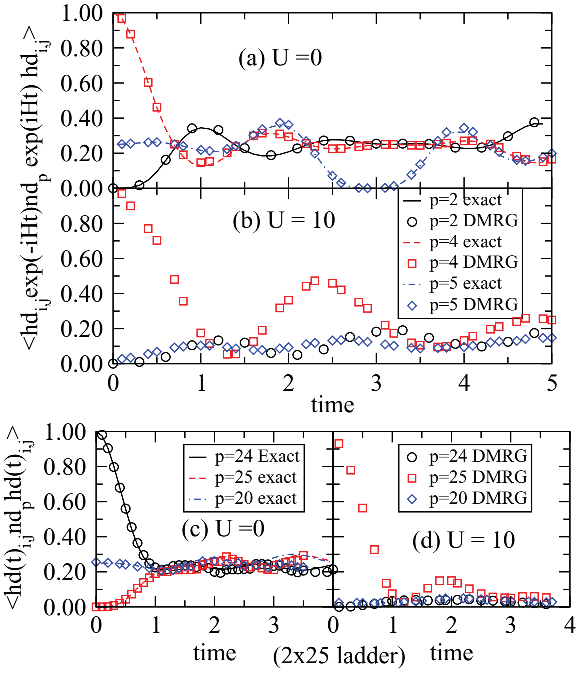

FIG. 5. (Color online) (a), (b) Double occupation at site p of the holon-doublon state Eq. (7) as a function of time for (a) $U = 0$ and (b) $U = 10$. A 2×4 ladder with $t_x = 1$ and $t_y = 0.5$ was used, containing four up and four down electrons. The holon-doublon operator was applied at $i = 2$, $j = 4$. (c), (d) Same as in panels (a) and (b) but now on a 2×25 ladder with 50 electrons. In this case, the holon-doublon operator was applied at $i = 25$, $j = 24$. In the vertical axes labels, $\mathrm{hd}_{ij}(t)$ denotes the state obtained by acting with a holon-doublon operator at sites i and j over the ground state and evolving that state to time t.

IV. COMPUTATIONAL EFFICIENCY AND CONCURRENCY

As in the static DMRG algorithm, the most computationally intensive task of the time-step targeting DMRG algorithm is the computation of Hamiltonian connections between the system and environment blocks. The difference is that now the lattice needs to be swept painstakingly to advance to larger times. The scaling, however, is linear with the number of finite sweeps, as long as the truncation m remains constant.

Sometimes it is better to stipulate the discarded weight instead of keeping m constant. This can easily be achieved from the input file, as described in Ref. [30].

This expensive task of building Hamiltonian connections between system and environment blocks can be parallelized [39–43]. Our implementation uses PTHREADS, a shared memory approach.[2] In percentage, the computation speed-up is similar to the static DMRG case, and a discussion of the strong scaling can be found in Ref. [28]. In terms of wall-clock time, the speed-up is larger due to the time-step-targeting DMRG algorithm taking more time than the static version.

The computation of target states could be parallelized easily, but whether serial or parallel, it is too fast to have substantial impact on the CPU times of production runs.

The measurement of observables is a different matter. DMRG++ computes observables *postprocessing*; that is, the main code saves all DMRG transformations, permutations, and quantum states to disk and a second *observer* code reads the data from disk and computes observables as needed. We argue that *postprocessing* is more advantageous than *in situ processing*, whether for the static or the time-dependent DMRG algorithm.

First, a single run of the main code enables computation of all observables. If, instead, one needed to make a decision on which observables to compute when running the main code, one would risk computing too much or too little. In the former case, computational resources and wall-clock time would be wasted. In the latter case, the main run would have to be repeated, leading to vast redundancies because the observations are not computationally intensive compared to the main code.

Moreover, by computing observables *postprocessing*, we decouple the code and enable scalable parallel computations. For example, one-point observables of the form $\langle\phi_1(t)|A_i|\phi_2(t)\rangle$ are parallelized over i, and two-point correlations, such as $\langle\phi_1(t)|A_i B_j|\phi_2(t)\rangle$, are parallelized over j and cached over i. If N is the number of sites of the lattice, the parallelization scales linearly up to almost N; the scaling is good but not perfect due to initialization costs [44].

V. SUMMARY

This paper explained in detail the implementation of the Krylov method for the real-time evolution within the DMRG algorithm, using time-step targeting [14,15]. We applied the method to a simple case of one-site excitations [31] and found the method to be accurate. For the case of the holon-doublon excitation, we have extended to two-leg ladders the previous results obtained in chains. Our analysis has shown that the method is accurate as long as the underlying DMRG algorithm is accurate. Since Mott insulators are under study for its possible applicability to solar cells, the present results pave the way for their continued study, now on more complex (but still quasi-one-dimensional) geometries, such as ladders.

We described computational tricks that can help decrease the run time. For example, we mentioned that shared memory parallelization with a few CPU cores can cut times by a factor of 2 or more. Parallelization works in the same way for time-dependent DMRG as it does for static DMRG, but it helps more in the former case, due to runs taking longer. We also argued in favor of the *postprocessing* of observables to speed up production runs and increase computational efficiency.

Our implementation, DMRG++, is free and open source. It emphasizes generic programming using C++ templates, a friendly user interface, and as few software dependencies as possible. DMRG++ makes writing new models and geometries easy and fast by using a generic DMRG engine.

ACKNOWLEDGMENTS

This work was supported by the Center for Nanophase Materials Sciences, sponsored by the Scientific User Facilities Division, Basic Energy Sciences, US Department of Energy (DOE), under contract with UT-Battelle. This research used resources of the National Center for Computational Sciences and the OIC at Oak Ridge National Laboratory. E.D. is supported by the US DOE, Office of Basic Energy Sciences, Materials Sciences and Engineering Division. G.A. acknowledges support from the DOE early career research program. L.G.D.S. also acknowledges support from Brazilian agencies FAPESP and CNPq.

APPENDIX: ONE-SITE EXCITATION IN THE NONINTERACTING CASE

Let U be the matrix that diagonalizes H_0, so that $c^\dagger_{i\uparrow} = \sum_\lambda U^*_{i,\lambda,\uparrow} u_{\lambda,\uparrow}$ and $u_{\lambda,\uparrow}$ are the diagonal operators. Let E_λ be the eigenvalues of H_0.

For the rest of this appendix, we omit ↑. After some algebra and omitting the sums over duplicated indices, we obtain

$$X_{ij}(t) = U^*_{i,\lambda} U^*_{j,\xi} U_{j,\xi'} U_{i,\lambda'} \langle u^\dagger_\lambda \mathrm{e}^{iH_0 t} u^\dagger_\xi u_\xi \mathrm{e}^{-iH_0 t} u_{\lambda'}\rangle. \quad \text{(A1)}$$

The ground state of H_0 is made up of $N_\uparrow$ filled levels, up to the Fermi energy, and particle-hole excitations are eigenstates of H (or, conversely, the excited states of H_0 are particle-hole excitations). Then, the λ'th level of $|\phi\rangle$ is occupied, and $u_{\lambda'}|\phi\rangle$ is an eigenstate of H with a hole at λ'. By applying this reasoning multiple times, the final result is $X_{ij}(t) = R_i R_j - |T_{ij}(t)|^2$, where $R_i = \sum'_\lambda |U_{i,\lambda}|^2$, and $T_{ij}(t) = \sum'_\lambda U^*_{i,\lambda} U_{j,\lambda} \exp(-iE_\lambda t)$, where the prime over the summation means sum only over occupied states λ.

[2] PTHREADS or POSIX THREADS is a standardized C language threads programming interface, specified by the IEEE POSIX standard.

[1] F. Heidrich-Meisner, I. González, K. A. Al-Hassanieh, A. E. Feiguin, M. J. Rozenberg, and E. Dagotto, Phys. Rev. B **82**, 205110 (2010).

[2] L. G. G. V. Dias da Silva, M. L. Tiago, S. E. Ulloa, F. A. Reboredo, and E. Dagotto, Phys. Rev. B **80**, 155443 (2009).

[3] F. Heidrich-Meisner, S. R. Manmana, M. Rigol, A. Muramatsu, A. E. Feiguin, and E. Dagotto, Phys. Rev. A **80**, 041603 (2009).

[4] F. Heidrich-Meisner, A. E. Feiguin, and E. Dagotto, Phys. Rev. B **79**, 235336 (2009).

[5] F. Heidrich-Meisner, G. Martins, C. Büsser, K. Al-Hassanieh, A. Feiguin, G. Chiappe, E. Anda, and E. Dagotto, Eur. Phys. J. B **527**, 67 (2009).

[6] S. R. White, Phys. Rev. B **48**, 10345 (1993).

[7] S. White, Phys. Rev. B **48**, 345 (1993).

[8] U. Schollwöck, Ann. Phys. **96**, 326 (2011).

[9] U. Schollwöck, Rev. Mod. Phys. **77**, 259 (2005).

[10] K. Hallberg, Adv. Phys. **55**, 477 (2006).

[11] J. Rodriguez-Laguna, e-print arXiv:cond-mat/0207340.

[12] A. E. Feiguin and S. R. White, Phys. Rev. B **72**, 020404 (2005).

[13] G. Alvarez, L. G. G. V. D. da Silva, E. Ponce, and E. Dagotto, [http://www.ornl.gov/∼gz1/dmrgPlusPlus/].

[14] S. R. Manmana, A. Muramatsu, and R. M. Noack, AIP Conf. Proc. **789**, 269 (2005).

[15] U. Schollwöeck and S. White, in *Effective models for low-dimensional strongly correlated systems*, edited by G. G. Batrouni and D. Poilblanc (AIP, Melville, NY, 2006), p. 155.

[16] U. Schollwöck, J. Phys. Soc. Jpn. **74**, 246 (2005).

[17] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal, J. Stat. Mech.: Theory Exp. (2004) P04005.

[18] S. R. White and A. E. Feiguin, Phys. Rev. Lett. **93**, 076401 (2004).

[19] A. Krylov, Izvestija AN SSSR, Otdel. mat. i estest. nauk **VII**, 491 (1931).

[20] ALPS Collaboration, B. Bauer *et al.*, J. Stat. Mech. (2011) P05001.

[21] C. Lanczos, J. Res. Nat. Bur. Stand. **45**, 255 (1950).

[22] E. Dagotto, Rev. Mod. Phys. **66**, 763 (1994).

[23] M. Hochbruck and C. Lubich, BIT **39**, 620 (1999).

[24] M. Hochbruck and C. Lubich, SIAM J. Numer. Anal. **34**, 1911 (1997).

[25] Y. Saad, (ed.), *Iterative Methods for Sparse Linear Systems* (PWS, Philadelphia, 2003).

[26] S. R. White, Phys. Rev. Lett. **77**, 3633 (1996).

[27] G. Alvarez, Comput. Phys. Commun. **180**, 1572 (2009).

[28] G. Alvarez, e-print arXiv:1003.1919.

[29] D. E. Knuth, *Literate programming* (Center for the Study of Language and Information, Stanford, CA, 1992).

[30] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.84.056706 for a description and usage of the computer code.

[31] C. Kollath, U. Schollwöck, and W. Zwerger, Phys. Rev. Lett. **95**, 176401 (2005).

[32] J. C. Slater and G. F. Koster, Phys. Rev. **94**, 1498 (1954).

[33] R. Noack, S. White, and D. Scalapino, in *Computer Simulations in Condensed Matter Physics VII*, edited by D. Landau, K. Mon, and H. Schüttler (Spinger Verlag, Heidelberg, 1994).

[34] K. A. Al-Hassanieh, F. A. Reboredo, A. E. Feiguin, I. Gonzalez, and E. Dagotto, Phys. Rev. Lett. **100**, 166403 (2008).

[35] J. Hubbard, Proc. R. Soc. London Ser. A **276**, 238 (1963).

[36] J. Hubbard, Proc. R. Soc. London Ser. A **281**, 401 (1964).

[37] J. Hubbard, Proc. R. Soc. London Ser. A **281**, 401 (1964).

[38] L. G. G. V. Dias da Silva, K. A. Al-Hassanieh, A. E. Feiguin, F. A. Reboredo, and E. Dagotto, Phys. Rev. B **81**, 125113 (2010).

[39] G. Hager, E. Jeckelmann, H. Fehske, and G. Wellein, J. Comput. Phys. **194**, 795 (2004).

[40] G. K.-L. Chan, J. Chem. Phys. **120**, 3172 (2004).

[41] Y. Kurashige and T. Yanai, J. Chem. Phys. **130**, 234114 (2009).

[42] S. Yamada, M. Okumura, and M. Machida, J. Phys. Soc. Jpn. **78**, 094004 (2009).

[43] J. Rincón, D. J. García, and K. Hallberg, Comput. Phys. Commun. **181**, 1346 (2010).

[44] G. Amdahl, in *AFIPS Conference Proceedings* (Academic Press, Waltham, Massachusetts, 1967), Vol. 30, pp. 483–485