

Time evolution of Matrix Product States

To cite this article: Juan José García-Ripoll 2006 *New J. Phys.* **8** 305

View the [article online](#) for updates and enhancements.

Related content

- [Out-of-equilibrium dynamics with matrix product states](#)
M L Wall and Lincoln D Carr
- [Tensor network techniques for the computation of dynamical observables in one-dimensional quantum spin systems](#)
Alexander Müller-Hermes, J Ignacio Cirac and Mari Carmen Bañuls
- [Stiffness in 1D matrix product states with periodic boundary conditions](#)
Davide Rossini, Vittorio Giovannetti and Rosario Fazio

Recent citations

- [Theoretical study of charge-transfer processes at finite temperature using a novel thermal Schrödinger equation](#)
Raffaele Borrelli
- [Scattering into one-dimensional waveguides from a coherently-driven quantum-optical system](#)
Kevin A. Fischer *et al*
- [Efficient propagation of the hierarchical equations of motion using the matrix product state method](#)
Qiang Shi *et al*



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

Time evolution of Matrix Product States

Juan José García-Ripoll

Max-Planck-Institut für Quantenoptik, Hans-Kopfermann-Str. 1,
Garching b. München, D-85748, Germany
E-mail: juanjose.garciaripoll@gmail.com

New Journal of Physics **8** (2006) 305

Received 8 October 2006

Published 6 December 2006

Online at <http://www.njp.org/>

doi:10.1088/1367-2630/8/12/305

Abstract. In this work, we develop several new simulation algorithms for one-dimensional (1D) many-body quantum mechanical systems combining the Matrix Product State variational ansatz (vMPS) with Taylor, Padé and Arnoldi approximations to the evolution operator. By comparing with previous techniques based on vMPS and Density Matrix Renormalization Group (DMRG), we demonstrate that the Arnoldi method is the best one, reaching extremely good accuracy with moderate resources. Finally, we apply this algorithm to studying how correlations are transferred from the atomic to the molecular cloud when crossing a Feshbach resonance with two-species hard-core bosons in a 1D optical lattice.

Contents

1. Introduction	2
2. MPS	3
2.1. Variational space.	3
2.2. Projector operator	5
3. MPS and DMRG	6
3.1. DMRG builds MPS	6
3.2. Targeting.	7
3.3. Targeting versus projection	7
4. Time evolution	9
4.1. Trotter decomposition	9
4.2. Runge–Kutta and Lanczos with DMRG	10
4.3. Runge–Kutta-like method with MPS	11
4.4. Padé approximations	11
4.5. Arnoldi method	12
5. Comparison	13
6. Simulation of Feshbach resonances	18
7. Conclusions	20
References	20

1. Introduction

The Density Matrix Renormalization Group (DMRG) method is a successful technique for simulating large low-dimensional quantum mechanical systems [1]. Developed for computing ground states of one-dimensional (1D) Hamiltonians, it is equivalent to a variational ansatz known as Matrix Product States (MPS) [2, 3]. This relation has been recently exploited to develop a much wider family of algorithms for simulating quantum systems, including time evolution [4, 5], finite temperature [5, 6] and excitation spectra [7]. Some of these algorithms have been translated back to the DMRG language [8]–[10] using optimizations developed in that field and introducing other techniques such as Runge–Kutta or Lanczos approximations of the evolution operator [11]–[14].

The MPS are a hierarchy of variational spaces, \mathcal{S}_D , (see equation (1)), sorted by the size of its matrices, D . MPS can efficiently represent many-body states of 1D systems, even when the Hilbert space is so big that the coefficients of a pure state on an arbitrary basis cannot be stored in any computer. While the accuracy of this representation has been proven for ground states [15], evolution of an arbitrary state changes the entanglement among its parties, and a MPS description with moderate resources (small D) might stop to be feasible. Throughout this paper, we will distinguish between the underlying mathematical structure of MPS and the variational algorithms (Matrix Product State variational ansatz (vMPS)) developed in [3, 5] and in this new study.

We will take a pragmatic approach. First of all, most algorithms in this work can compute truncation errors so that the accuracy of simulations remains controlled. Second, we are interested in simulating *physically* small problems, such as the dynamics of atoms and molecules in optical

lattices. For such problems small D are sufficient to get a qualitative and even quantitatively good description of the observables in our systems. As we will see below, the biggest problem is the accumulation of truncation errors and not always the potential accuracy of a given MPS space for representing our states.

The outline of the paper is as follows. In section 2, we briefly introduce MPS and review some of their properties. In subsection 2.2, we present the optimal projection onto a MPS space, which is the keystone of most evolution algorithms. In section 3, we introduce for completeness the DMRG algorithm, focusing on the concepts which are essential for time evolution. In particular, we concentrate on the difficulties faced when implementing DMRG simulations and how those techniques relate to vMPS. In section 4, we review almost all recently developed simulation algorithms under the common formalism based on the optimal truncation operator. Additionally, we introduce three new methods: two of them are based on Taylor and Padé expansions of the evolution operator while the other one uses an ‘Arnoldi’ basis of MPS to increase the accuracy. Section 5 is a detailed comparison of vMPS and DMRG algorithms using spin- $\frac{1}{2}$ models. Our study shows that all methods are strongly limited by truncation and rounding errors. However, among all techniques, vMPS methods and in particular our Arnoldi method perform best for fixed resources, measured by the size of matrices D or size of basis in DMRG. In the last part of our paper, section 6, we present a real-world application of the Arnoldi evolution algorithm, which is to study a model of hard-core bosonic atoms going through a Feshbach resonance. Current experiments [16]–[18] with such systems have focused on the number and stability of the formed molecules. In this study, we focus on the 1D many-body states and show that coherence is transferred from the atomic component to the molecular one, so that this procedure can be used to probe higher order correlations in the atomic cloud. Finally, in section 7, we summarize our results and open lines of research.

2. MPS

In this first section, we introduce the notion of MPS, together with some useful properties and an important operator, the projection onto a MPS space. This section is a brief review of the concepts found in [3, 5].

2.1. Variational space

The techniques in this work are designed for the study of 1D or quasi-1D quantum mechanical lattice models. If N is the number of lattice components and d the number of degrees of freedom of each site, the Hilbert space of states will have a tensor product structure, $\mathcal{H} = \mathcal{H}_1^{\otimes N}$, where $d = \dim \mathcal{H}_1$ are the degrees of freedom of a single site. We will consider two examples here: one with spin- $\frac{1}{2}$ particles, where $d = 2$ (section 5), and later on a study of atoms and molecules in an optical lattice where $d = 25$ (section 6).

Given those prerequisites, the space of MPS of size D is defined as the set of states of the form

$$\mathcal{S}_D := \left\{ \left(\text{Tr} \prod_k A_k^{s_k} \right) |s_1 \dots s_N\rangle, A_k^{s_n} \in \mathbb{C}^{D_k \times D_{k+1}} \right\}, \quad (1)$$

where $s_k = 1 \dots d$ labels the physical state of the k th lattice site. The A_k are complex matrices of dimensions that may change from site to site but are of bounded size, $D_{k-1} \times D_k \leq D^2$. Throughout the paper, we will use different notation for the matrices $\{A_k^{s_k}\}$. An index k or l will always label the site they belong to and, whenever the expression is not ambiguous, the site index will be dropped: $A^{s_k} := A_k^{s_k}$. The MPS components can also be regarded as tensors, $A_{\alpha\beta}^{s_k}$, the Greek indices denoting virtual dimensions $\alpha, \beta = 1 \dots D$. Finally, at some point, we will rearrange all values forming a vector $\vec{A}_k^t := (A_{11}^1, A_{11}^2, \dots, A_{DD}^d)$ in a complex space $\mathbb{C}^{d \times D \times D}$.

The first important property of the MPS is that they do not form a vector space. In general, a linear combination of M states with size D requires matrices of size MD to be represented. It is easy to do a constructive proof of the previous fact, but the reader may convince himself with a simple example, made of the two product states $|0\rangle^{\otimes N}$ and $|1\rangle^{\otimes N}$, which live in S_1 , and the Greenberger–Horne–Zeilinger (GHZ) state $|0\rangle^{\otimes N} + |1\rangle^{\otimes N} \in S_2$.

The previous remark leads us to another property, which is that the dimension D required to represent a state faithfully¹ is related to the amount of entanglement in the system. More precisely, that dimension is equal to the maximum Schmidt number of the state with respect to any bipartition and thus an entanglement monotone [19]. Indeed, creating entanglement forces us to use bigger and bigger MPS and this is the reason why some problems cannot be simulated efficiently using MPS.

The third important property is that we can efficiently compute scalar products, distances and in general expectation values of the form $\langle \psi | O_1 \otimes O_2 \otimes \dots \otimes O_L | \phi \rangle$, where $\psi, \phi \in \mathcal{S}_D$ and $\{O_i\}_{i=1}^L$ are local operators acting on the individual qubits or components of our tensor product Hilbert space. For instance, given that we know the matrices of those states, $\{A_k^{s_k}\}$ for ψ and $\{B_k^{s_k}\}$ for ϕ , the previous expectation value is made of a product of N matrices,

$$\langle \psi | \otimes_{k=1}^L O_k | \phi \rangle = \text{Tr} \left[\prod_{k=1}^L E(O_k, A_k, B_k) \right], \quad (2)$$

where the ‘transfer matrices’ are defined as follows:

$$E(O_k, A_k, B_k) := \sum_i (A_k^{s_k})^* \otimes B_k^{s_k} \langle s_k | O_k | s_k \rangle. \quad (3)$$

Since all usual Hamiltonians and correlators can be decomposed as sum of products of local terms, the previous formulae are very useful. An important remark is that when computing equation (3) one should not directly multiply the matrices E , but cleverly contract the A and B tensors separately, so as to achieve a performance $\mathcal{O}(dD^3)$.

The last property is that expectation values, distances $\|\psi - \phi\|^2$, fidelities $|\langle \psi | \phi \rangle|$ and norms $\|\psi\|^2$, are quadratic forms with respect to each of the matrices in the MPS. Regarding the matrices of the state as elements of a complex vector, we can rewrite equation (2) for the k th site as

$$\langle \psi | \otimes_{k=1}^L O_k | \phi \rangle = \vec{A}_k^\dagger Q \vec{B}_k, \quad (4)$$

where the quadratic form Q is built as follows:

$$Q_{[s\alpha\alpha'] [s'\beta\beta']} := \langle s | O_k | s' \rangle \left[\prod_{j=\{k+1 \dots L, 1 \dots k-1\}} E(O_j, A_j, B_j) \right]_{\alpha\beta, \alpha'\beta'}, \quad (5)$$

¹ By this we mean with absolutely no error.

where $[s\alpha\beta]$ denotes grouping of indices consistent with the ordering of the elements of \vec{A} and \vec{B} . This formula is used on all algorithms, from the computation of ground states [3] to the time evolution [5]. An important optimization is to avoid computing the full matrix Q , but to use the structure in (5) together with the sparseness of the transfer matrices.

2.2. Projector operator

Even if the MPS do not form a vector space, they are embedded in a bigger Hilbert space and provided with a distance. It is therefore feasible, given an arbitrary state vector, to ask for the best approximation in terms of MPS of a fixed size. The optimal projection onto \mathcal{S}_D is a highly nonlinear operation and it will be denoted in this study by \mathcal{P}_D . Following [5]

$$\mathcal{P}_D \sum_k c_k |\phi^{(k)}\rangle := \operatorname{argmin}_{\psi \in \mathcal{S}_D} \left\| |\psi\rangle - \sum_k c_k |\phi^{(k)}\rangle \right\|^2. \quad (6)$$

If we rather want to approximate the action of an operator that can be decomposed as $U = X^{-1}Y$, we will apply a generalization of the correction vector method [1]

$$\mathcal{P}_D(X^{-1}Y|\phi\rangle) := \operatorname{argmin}_{\psi \in \mathcal{S}_D} \|X|\psi\rangle - Y|\phi\rangle\|. \quad (7)$$

This formula is simple to apply and behaves well for a singular operator X . Its use will become evident when studying time evolution algorithms in section 4.

One may quickly devise a procedure for computing the minimum of equation (7) based on the definition of distance:

$$\|X|\psi\rangle - Y|\phi\rangle\|^2 = \langle\psi|X^\dagger X|\psi\rangle - 2\Re\langle\psi|X^\dagger Y|\phi\rangle + \langle\phi|Y^\dagger Y|\phi\rangle. \quad (8)$$

All scalar products in equation (8) are quadratic forms with respect to each of the matrices in the states $|\phi\rangle$ and $|\psi\rangle$. The distance is minimized by optimizing these quadratic forms site by site, or two sites at a time,² sweeping over all lattice sites until convergence to a small value which

will be the truncation error [5].

In short the algorithm looks as follows: (i) compute some initial guess for the matrices of the optimized state ψ . (ii) Focus on the site $k = 1$. (iii) Use equations (4) and (5) to find out the quadratic form associated with equation (8) for the first matrix

$$\epsilon := \vec{A}_k^\dagger Q_{X^\dagger X} \vec{A}_k - 2\Re \vec{A}_k^\dagger Q_{X^\dagger Y} \vec{B}_k + \vec{B}_k^\dagger Q_{Y^\dagger Y} \vec{B}_k. \quad (9)$$

(iv) The stationary points of the error are given by the equation

$$Q_{X^\dagger X} \vec{A}_k = Q_{X^\dagger Y} \vec{B}_k. \quad (10)$$

Solve this equation and use the outcome as the new value of A_k . (v) Estimate the error using equation (9). If ϵ is small enough or does not improve significantly, stop. Otherwise move to another site, $k = k \pm 1$, and continue on step (iii).

² The two-sites alternative, borrowed from DMRG, has the advantage of not getting trapped in subspaces of conserved quantities that commute with both X and Y .

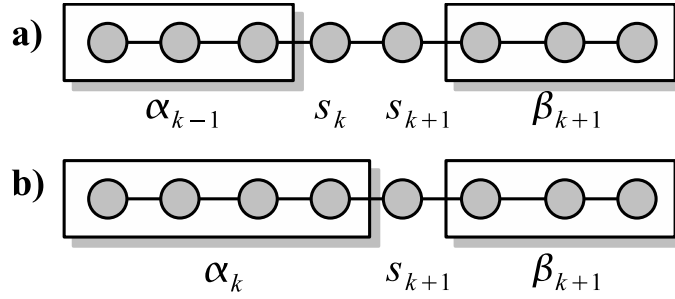


Figure 1. (a) DMRG view of a state, with the basis for the left $|\alpha_{k-1}\rangle$ and right block $|\alpha_{k+1}\rangle$, and the states of the central spins, $|s_k\rangle, |s_{k+1}\rangle$. (b) On a renormalization step, one spin is incorporated to the left block and a new basis is built, $|\alpha_k\rangle := A_{k-1,k}^{s_k} |\alpha_{k-1}\rangle |s_k\rangle$.

3. MPS and DMRG

Even though the numerical techniques for dealing with MPS variationally seem very different from those of DMRG [1], both methods are intimately connected. First of all, the DMRG produces MPS at each state of its algorithms. Secondly, particular algorithms such as the search for ground states in open boundary condition problems are equivalent in DMRG and vMPS [3]. Thirdly, other concepts, such as basis adaptation and state transformation from DMRG, are analogous to the vMPS ones, even though they are less powerful and less accurate. In this section, we will elaborate on these statements.

3.1. DMRG builds MPS

The DMRG algorithms are based on the key idea that interesting states can be expressed using a basis with a small number of vectors. Take for instance the chain in figure 1(a). Any state of this chain can be decomposed in the form

$$|\psi\rangle = \psi(\alpha_{k-1} s_k s_{k+1} \alpha_{k+1}) |\alpha_{k-1}\rangle |s_k\rangle |s_{k+1}\rangle |\alpha_{k+1}\rangle, \quad (11)$$

where we sum over repeated indices. While for describing individual spins we use all possible states, $s_k, s_{k+1} = 1 \dots d$, in DMRG the states of the left and right blocks are expressed in a finite basis, $\alpha_k, \beta_k = 1 \dots M$, where M , the number of states kept, is the basic control parameter. DMRG algorithms build those basis states recursively, by taking a smaller block, adding a site and truncating the basis of the bigger block ‘optimally’ in a way to be specified later. Thus we have the relations

$$|\alpha_k\rangle := A_{\alpha_{k-1}\alpha_k}^{s_k} |\alpha_{k-1}\rangle |s_k\rangle, \quad (12a)$$

$$|\beta_k\rangle := A_{\beta_k\beta_{k+1}}^{s_{k+1}} |s_{k+1}\rangle |\beta_{k+1}\rangle. \quad (12b)$$

It is trivial to see, substituting those equations into equation (11), that all DMRG states are MPS [2, 3].

3.2. Targeting

An important question in DMRG is how many states we have to keep for the left and right blocks and how to optimize them. The criterion is that the basis describing those blocks has to represent accurately a family of target states, $|\phi_n\rangle$. The algorithm consists of a series of sweeps over the lattice [11, 13, 14, 20] with a recipe to achieve an optimal representation. For instance, let us say that we have an approximate basis around sites k and $k + 1$ and we will improve this sweeping from left to right, to $k + 1$ and $k + 2$ (figure 1). The first step is to build the weighted density matrix of the left piece of the chain

$$\begin{aligned}\rho_L &:= \sum_n w_n \phi_n(\alpha_{k-1} s_k s_{k+1} \beta_{k+1}) \phi_n(\alpha'_{k-1} s'_k s'_{k+1} \beta_{k+1})^* |\alpha_{k-1} s_k\rangle \langle \alpha'_{k-1} s'_k| \\ &= \sum_n w_n \text{Tr}_{s_{k+1} \alpha_{k+1}} |\phi_n\rangle \langle \phi_n|,\end{aligned}\quad (13)$$

with some normalized weights, $\sum_n w_n = 1$. Secondly, take the M most significant eigenvectors of this matrix

$$\rho |\alpha_k\rangle = \lambda_k |\alpha_k\rangle, \quad \lambda_k \geq \lambda_j \quad \forall k > l; \quad k, l = 1 \dots M \times d. \quad (14)$$

These vectors become the improved new basis for the enlarged left block and are related by a transformation matrix to the basis elements of the smaller block (12a). Matrix elements of observables and of the initial and target states have to be recomputed using this isometry and one continues until the end of the lattice. A similar procedure is employed to create the vectors $|\beta_k\rangle$ recursively from right-to-left. Multiple sweeps can be performed this way.

For static problems one uses as target states, $|\phi_n\rangle$, the ground state and a number of excitations, computed with the Hamiltonian in the truncated basis. In some time evolution algorithms [11]–[14], [20] the targeting is done with respect to approximations of the time evolved state $|\psi(t)\rangle$. In this study, we have used $|\phi_k\rangle := |\psi(k\Delta t/(N_v - 1))\rangle$, where the intermediate and final states, $|\psi(\tau)\rangle$, are computed using the Hamiltonian on the truncated basis (see subsection 4.2).

3.3. Targeting versus projection

There are a number of complicated subtleties and facts that are rarely mentioned in the DMRG literature. The first one is that in most implementations of the targeting algorithm, the state itself is updated and rewritten in the new basis after each step

$$|\psi\rangle \rightarrow \psi(\alpha_{k-1} s_k s_{k+1} \beta_{k+1}) A_{\alpha_{k-1} \alpha_k}^{s_k \star} |\alpha_k\rangle |s_k\rangle |s_{k+1}\rangle |\beta_{k+2}\rangle. \quad (15)$$

However, this leads to wrong answers because the initial state $\psi(0)$ deteriorates during the algorithm, and it only works when the basis of the left and right block are already large enough. A more accurate procedure consists in keeping the initial state in the original basis, and on each step of the algorithm compute its matrix elements in the new basis. As an example, in figure 2, we plot the error of an evolution algorithm with the trivial algorithm and with the correct one, for an initial product state $|1\rangle^{\otimes L}$ in the σ_i^z basis and using a simple Hamiltonian $H = \sum_i \sigma_i^x$. In both cases, we begin with a basis of size $M = 1$ vectors per block, letting the basis grow up to $2^{L/2}$. However only the second method is capable of doing the update.

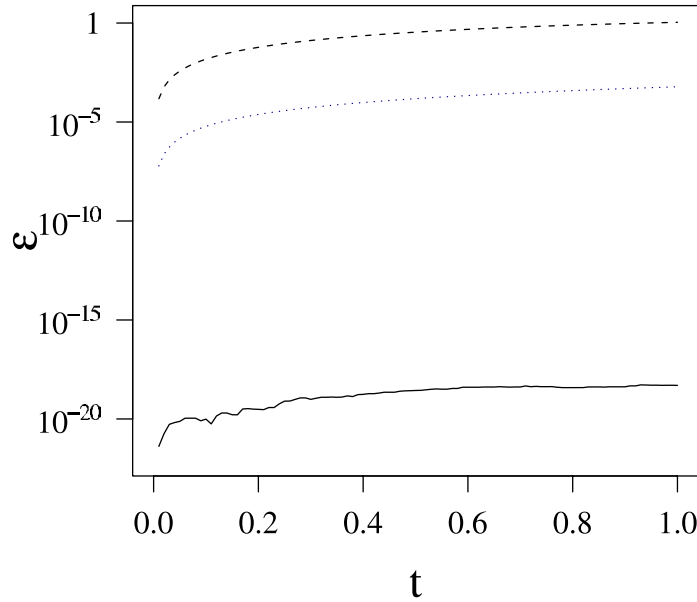


Figure 2. Error in the DMRG Runge–Kutta algorithm (see subsection 4.2), using as initial state $|\psi\rangle = |0\rangle^{\otimes N}$ and Hamiltonians $H = \sum_i \sigma_i^x$ (solid, dashed) or $H = \prod_i \sigma_i^x$ (dotted). The algorithm used either a single basis (dashed) or two sets of basis states (solid, dotted) for targeting. The dashed line shows errors due to using a single basis; the dotted line fails because the Hamiltonian does not have nonzero elements on the initial DMRG basis.

An even more important issue is that in current literature this method is formulated in an intuitive way and little is known about why it works and how accurate it is. In this respect, a close look at the matrices (12a) and (12b) created by the targeting with multiple vectors, $|\phi_i\rangle$ shows that this procedure is quite similar to the computing projection operator for some linear combination $\mathcal{P}_M(\sum_k c_k |\phi_k\rangle)$, but the steps for computing the optimal approximation are wrong if there are more than one target state.

Another consequence of being tied to the notion of ‘optimal basis’ is that the targeting procedure does not work when the operators O_n and the target states $|\phi_n\rangle$ have no elements in the initial basis. A trivial example consists of the same product state $|\psi(0)\rangle = |1\rangle^{\otimes N}$ as before, beginning with $M = 1$ states per block and evolved with $H = \prod_i \sigma_i^x$. As shown in figure 2, a conventional DMRG sweep cannot enlarge the basis in the appropriate way and the method fails to follow the evolution of the state. A vMPS algorithm, on the other hand, even if it begins with a state with $D = 1$, grows the state up to the maximal size $D = 2$ and makes absolutely no error in the simulation.

The final practical difference is that in current DMRG works [11]–[14] the powers of the Hamiltonian acting on the original state, $H^n|\phi\rangle$, are approximated by the truncation of H to the current basis. This is another potential source of errors which we cleverly avoid in our Arnoldi and Runge–Kutta methods shown below (subsection 4.3 and 4.5) and its effects are evident in some of the simulations (section 5).

The previous paragraphs can be rephrased as follows: DMRG algorithms need not only an initial input state, but also a suitable and large enough basis for doing the time evolution.

How to construct this basis is largely heuristic, and contrasts with the systematic way in which vMPS work.

4. Time evolution

Since \mathcal{S}_D is not a vector space, we cannot solve a Schrödinger equation directly on it. Our goal is rather to approximate the evolution at short times by a formula like

$$|\psi(t + \Delta t)\rangle \simeq \mathcal{P}_D[U_n(\Delta t)|\psi(t)\rangle]. \quad (16)$$

Here, \mathcal{P}_D is the optimal projection operator defined before and $U_n(\Delta t) = \exp(-iH\Delta t) + \mathcal{O}(\Delta t^n)$ is itself an approximation of n th order to the evolution operator. Even though this formulation applies qualitatively to all recently developed vMPS and DMRG algorithms, there are subtle differences that make some methods more accurate than others. The actual implementation of time evolution is thus the topic of the following subsection.

4.1. Trotter decomposition

While there had been previous works on simulating time evolution using DMRG algorithms [21]–[23], an important breakthrough happened with the techniques in [4], which was later translated to the DMRG language [8]–[10] and has since been applied to a number of interesting problems [24]–[27].

Vidal's seminal paper [4] suggested doing the time evolution with a mixture of two-qubit quantum gates and truncation operations. The idea is to split a Hamiltonian with nearest-neighbour interactions into terms acting on even and odd lattice edges

$$H = \sum_{k=1}^{L/2} h_{2k} + \sum_{k=1}^{L/2} h_{2k-1} =: H_E + H_O. \quad (17)$$

Here, h_i operates on the sites i and $i + 1$, and in our case it will correspond to a spin–spin interaction, $h_i = \cos(\theta)(s_i^x s_{i+1}^x + s_i^y s_{i+1}^y + \Delta s_i^z s_{i+1}^z) + \sin(\theta)s_i^z/2$. Finally, H_E and H_O group the interaction terms acting on even and odd pairs, h_{2k} and h_{2k+1} respectively.

The previous decomposition leads naturally to a Suzuki–Trotter decomposition of the time evolution operator, which is now approximated by a product of unitaries acting on even and odd lattice bonds:

$$e^{-iH\Delta t} \simeq e^{-iH_E\Delta t} e^{-iH_O\Delta t} = \prod_{k=1}^{L/2} e^{-ih_{2k}\Delta t} \prod_{k=1}^{L/2} e^{-ih_{2k+1}\Delta t}. \quad (18)$$

Inserting optimal truncations in between the applications of these unitaries, $U_j := \exp(-ih_j\Delta t)$, Vidal's algorithm can be recasted in the form of equation (16)

$$|\psi(t + \Delta t)\rangle := \prod_{k=1}^{L/2} \mathcal{P}_D U_{2k} \prod_{k=1}^{L/2} \mathcal{P}_D U_{2k+1} |\psi(t)\rangle. \quad (19)$$

Applying each of the unitaries U_j is a relatively costless task, and in particular, for open boundary condition problems, the combination $\mathcal{P}_D U_j$ can be done in a couple of steps which amount to contracting neighbouring matrices and performing a singular value decomposition [4, 8, 9].

In [5], we developed a variant of this method which does not insert so many truncation operators, but waits until the end

$$|\psi(t + \Delta t)\rangle := \mathcal{P}_D \prod_{k=1}^{L/2} U_{2k} \prod_{k=1}^{L/2} U_{2k+1} |\psi(t)\rangle. \quad (20)$$

This procedure has a similar computational cost, $\mathcal{O}(ND^3)$, but the solution is then expected to be optimal for a given Trotter decomposition and can be generalized to problems with periodic boundary conditions.

The accuracy of either method may be increased by an order of magnitude using a different second order decomposition

$$e^{-iH\Delta t} = e^{-iH_E\Delta t/2} e^{-iH_O\Delta t} e^{-iH_E\Delta t/2} + \mathcal{O}(\Delta t^2), \quad (21)$$

but it is better to apply a Forest–Ruth formula [8, 28]

$$e^{-iH\Delta t} = e^{-iH_E\theta\Delta t/2} e^{-iH_O\theta\Delta t} e^{-iH_E(1-\theta)\Delta t/2} e^{-iH_O(1-2\theta)\Delta t} e^{-iH_E(1-\theta)\Delta t/2} e^{-iH_O\theta\Delta t} e^{-iH_E\theta\Delta t/2}, \quad (22)$$

with the constant $\theta = 1/(2 - 2^{1/3})$ and which has a Trotter error of order $\mathcal{O}(\Delta t^5)$. Note that better Suzuki–Trotter formulae can be designed but they do not provide a big improvement in the number of exponentials or accuracy [28].

4.2. Runge–Kutta and Lanczos with DMRG

After the development of the Trotter methods, in [11] we find a new algorithm in the field of DMRG. The idea now is to use not a Trotter formula, but a Runge–Kutta iteration:

$$|k_1\rangle := \Delta t H(t) |\psi(0)\rangle, \quad (23a)$$

$$|k_2\rangle := \Delta t H(t) [|\psi(0)\rangle + \frac{1}{2}|k_1\rangle], \quad (23b)$$

$$|k_3\rangle := \Delta t H(t) [|\psi(0)\rangle + \frac{1}{2}|k_2\rangle], \quad (23c)$$

$$|k_4\rangle := \Delta t H(t) [|\psi(0)\rangle + |k_3\rangle]. \quad (23d)$$

These vectors are then used to interpolate the state at other times³

$$|\psi(\frac{\Delta t}{3})\rangle \simeq |\psi(0)\rangle + \frac{1}{162}[31|k_1\rangle + 14(|k_2\rangle + |k_3\rangle) - 5|k_4\rangle], \quad (24a)$$

$$|\psi(2\frac{\Delta t}{3})\rangle \simeq |\psi(0)\rangle + \frac{1}{81}[16|k_1\rangle + 20(|k_2\rangle + |k_3\rangle) - 2|k_4\rangle], \quad (24b)$$

$$|\psi(\Delta t)\rangle \simeq |\psi(0)\rangle + \frac{1}{6}[|k_1\rangle + 2(|k_2\rangle + |k_3\rangle) + |k_4\rangle]. \quad (24c)$$

³ Notice the typo in equation (4) in [11].

These three vectors, together with $|\psi(0)\rangle$, are used to find an optimal basis using the targeting procedure explained in subsection 3.2. Once the basis is fixed, the time evolution is performed with the same formulae but a smaller time step, $\Delta t/10$. There are variants of this technique which approximate the time evolved state using a Lanczos method [29, 30] with the truncated matrix of the Hamiltonian in the DMRG basis. The different submethods differ on whether the preparation of the basis is done only using the final and initial states [14] or multiple intermediate time steps [12, 13].

4.3. Runge–Kutta-like method with MPS

Our implementation of a fourth order method with Runge–Kutta-like formulae uses several simplifications. First of all, since our Hamiltonian is constant, a Runge–Kutta expansion becomes equivalent to a fourth order Taylor expansion of the exponential

$$\begin{aligned} |\psi(\Delta t)\rangle &= \sum_{n=0}^4 \frac{1}{n!} (iH\Delta t)^n |\psi(0)\rangle + \mathcal{O}(\Delta t^5) \\ &\simeq (iH\Delta t - z_1)(iH\Delta t - z_1^*)(iH\Delta t - z_2)(iH\Delta t - z_2^*) |\psi(0)\rangle \\ &=: Y_1 Y_2 Y_3 Y_4 |\psi(0)\rangle. \end{aligned} \quad (25)$$

Here, we have rewritten the fourth order polynomial in terms of its roots, z_1 and z_2 . Using the fact that we know an efficient algorithm to compute $\mathcal{P}_D Y_i$ acting on a MPS, we can write

$$|\psi(\Delta t)\rangle := \prod_{i=1}^4 \mathcal{P}_D Y_i |\psi(0)\rangle, \quad (26)$$

which is our vMPS Runge–Kutta-like algorithm. There are multiple reasons to proceed this way. On the one hand, we do not want to approximate higher powers of the Hamiltonian using a truncated basis (see subsection 3.2). On the other hand, if we expand the exponential to all powers, there will be too many operators and the complexity of the state will increase enormously. The previous decomposition has proven to be a good compromise.

4.4. Padé approximations

Runge–Kutta and in general polynomial approximations to the exponential are not unitary. Indeed, if we look at the eigenvalues of the evolution operator in either equation (24c) or (25), we will see that they are of the form $\lambda_n = \sum_{n=0}^4 (iE_n \Delta t)^n / n!$, where E_n are the eigenvalues of the Hamiltonian H . From this equation, we see that $|\lambda_n| \neq 1$ and some eigenmodes may grow exponentially, which is another way to say that Runge–Kutta algorithms are numerically unstable.

There exist multiple implicit methods that eliminate the lack of unitarity and produce stable approximations. They receive the name ‘implicit’ because the value of the state at a later time step Δt is obtained by solving an equation or inverting an operator. We will focus on Padé approximations to the exponential

$$U_n(\Delta t) = \frac{\sum_k \alpha_k H^k}{\sum_k \beta_k H^k}, \quad (27)$$

which are computed with same order polynomials in the numerator and denominator [31]. The lowest order method is known as the Crank–Nicholson scheme, it arises from a second order discretization of the Schrödinger equation and has the well known form

$$U_{\text{CN2}}(\Delta t) = \frac{1 - iH\Delta t/2}{1 + iH\Delta t/2}. \quad (28)$$

It is easy to verify that the eigenvalues of this operator are just phases and that the total energy is a conserved quantity. The other method that we have used and which we compare in this study is a fourth order expansion

$$U_{\text{CN4}}(\Delta t) = \frac{1 - i\Delta t H/2 - (\Delta t H)^2/12}{1 + i\Delta t H/2 - (\Delta t H)^2/12}. \quad (29)$$

Applying either U_{CN2} or U_{CN4} on a matrix product state is equivalent to solving the problem in equation (7), where the operators X and Y are the denominator and the numerator in the previous quotients.

4.5. Arnoldi method

The last and most important method that we present in this paper combines many of the ideas explained before. First of all, we will use the fact that a linear combination of MPS, such as $\sum_{k=1}^{N_v} c_k |\phi_D^{(k)}\rangle$, resides in a space of bigger matrix product states, $\mathcal{S}_{N_v D}$, and it is thus a more accurate representation of the evolved state than a single vector of size D . In the language of DMRG, N_v vectors each of size D provide us with an effective basis of size $N_v D$. This optimistic estimate is only possible when the vectors are indeed linearly independent. Our choice for an optimal decomposition will be therefore a Gramm–Schmidt orthogonalization of the Krylov subspace, $\{\psi, H\psi, H^2\psi, \dots\}$, performed using MPS

$$|\phi_{k+1}\rangle \simeq \mathcal{P}_D \left(H|\phi_k\rangle - \sum_{j \leq k} \frac{\langle \phi_j | H | \phi_k \rangle}{\langle \phi_j | \phi_j \rangle} |\phi_j\rangle \right), \quad (30)$$

with initial condition $|\phi_0\rangle := |\psi(0)\rangle$. Defining the matrices $N_{ik} := \langle \phi_j | \phi_i \rangle$ and $H_{ik} := \langle \phi_j | H | \phi_i \rangle$ we compute an Arnoldi estimate of the exponential

$$|\psi(\Delta t)\rangle := \mathcal{P}_D \sum_k [e^{-i\Delta t N^{-1}H}]_{k0} |\phi_k\rangle. \quad (31)$$

This algorithm involves several types of errors all of which can be controlled. First, the error due to using only N_v basis vectors is proportional to the norm of the vector ϕ_{N_v} as in ordinary Lanczos algorithms [29, 30]. Truncation errors arising from \mathcal{P}_D can be also computed during the numerical simulation. Out of these errors, in our experience, the final truncation in equation (31) is the most critical one, since the other ones may be compensated by adding more and more vectors.

Finally, for completeness, in this study we have also implemented a Lanczos method. It differs from the previous one in that the basis is built orthogonalizing only with respect to the two previous vectors, so that equation (30) contains only three summands, as in ordinary Lanczos

iterations. One then assumes that, due to the Hermiticity of the Hamiltonian, orthogonality to the rest of the Krylov basis is preserved. Furthermore, if this is true, the effective matrices for H and N are tridiagonal and can be constructed with a simple recurrence. This method has a potential gain of $\mathcal{O}(1/N_v)$ in speed due to the simplifications in equation (30). However, as we will see later, truncation errors spoil the orthogonality of the Lanczos vectors and makes the method useless for small matrices.

5. Comparison

We tested all algorithms by simulating the evolution of the same state under a family of spin- $\frac{1}{2}$ Hamiltonians with nearest-neighbour interactions

$$H = \sum_k [\cos(\theta)(s_k^x s_{k+1}^x + s_k^y s_{k+1}^y + \Delta s_k^z s_{k+1}^z) + \sin(\theta)s_k^z]. \quad (32)$$

As initial state we take the product $|\psi(0)\rangle \propto (|0\rangle + |1\rangle)^{\otimes L}$, where $|0\rangle$ and $|1\rangle$ are the eigenstates of s^z . By restricting ourselves to ‘small’ problems ($L \leq 20$), we can compare all algorithms with accurate solutions based on exact diagonalizations and the Lanczos algorithm [29, 30]. Notice that we measure the error in the full wavefunction, $\varepsilon := \|\psi_D(T) - U(T)\psi(0)\|^2$, and not on the expectation values of simple correlators whose exact evolution is known [14]. The outcome of some of the simulations is shown in figures 3–5, which we will discuss in the following paragraphs.

The first set of simulations was done for eight spins using matrices of size $D = 16$ and a DMRG basis of similar size. Since S_D contains all possible states, $\mathcal{P}_D = 1$, we expect no truncation errors in any of the algorithms. As shown in figure 3 for the XY model with $\theta = 0.35$ and $\Delta = 0$, for medium to long time steps most errors show the expected behaviour. Thus, the errors of the Trotter methods of second and fourth order follow the laws $\mathcal{O}(\Delta t^2)$ and $\mathcal{O}(\Delta t^4)$. Runge–Kutta, Taylor and Padé approximations have an error of $\mathcal{O}(\Delta t^2)$, and for the Arnoldi and Lanczos methods with $N_v = 4$ and 8 vectors we have a qualitative behaviour $\mathcal{O}(\Delta t^{N_v-1})$. Since the size of the matrices and of the DMRG basis is very big, all these laws are followed, irrespective of whether the implementation uses DMRG or vMPS. The only exception seems to be the DMRG Lanczos when implemented with only two target states. This method behaves more poorly than the counterpart with N_v target states given by $|\phi_k\rangle := |\psi(k\Delta t/(N_v - 1))\rangle$, $k = 0 \dots N_v - 1$.

Out of the methods that work as expected, all of them break the ideal laws at some point, acquiring an error of order $\mathcal{O}(\Delta t^{-2})$. This error is exponential in the number of steps $T/\Delta t$ and it signals the finite accuracy of the optimization algorithms, due to the limited precision of the computer. Roughly, since current computers cannot compute the norms of vectors, $\|\psi\|^2$, with a relative error better than 10^{-16} , a worst case estimate is $\epsilon = (T/\Delta t)^2 10^{-16}$, which perfectly fits these lines.

Theoretically, the performance measured in computation time and memory use of all algorithms is of order $\mathcal{O}(N_v N_H D^3/N)$, where N_H is related to the number of operators in $\{X, Y, X^\dagger X, Y^\dagger Y, \dots\}$, N is the size of the problems and the additional factor N_v only appears for Arnoldi and Lanczos methods. For periodic boundary conditions the cost increases by $\mathcal{O}(D^2)$. However, we have explicitly avoided PBC problems because DMRG cannot handle them so accurately. In practice, out of all methods, the Padé expansions are the slowest ones, while the Trotter formulae, being local, are the fastest. In between we find the Arnoldi and Lanczos

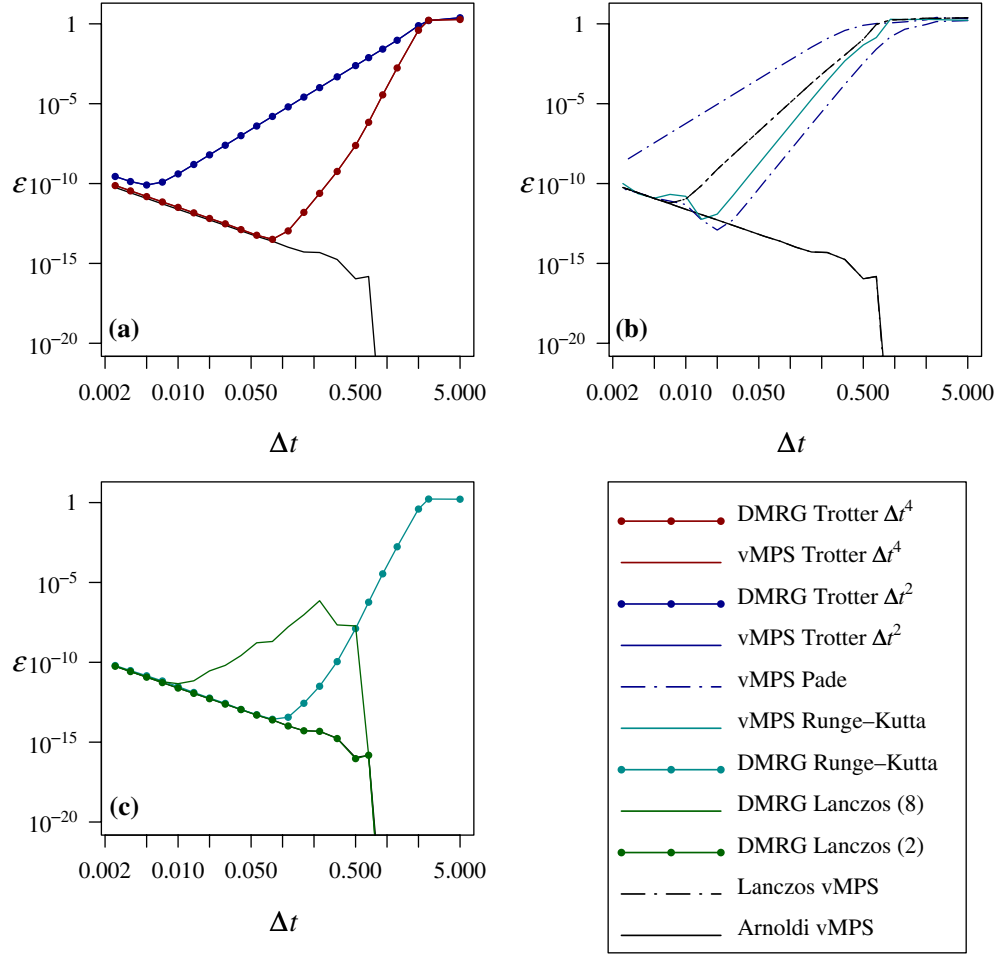


Figure 3. Error, ε , versus time step, Δt , for simulations of model (32) with eight spins, $\theta = 0.35$, $\Delta = 0$, $D = 16$ and $T = 10$. We compare (a) Trotter methods, (b) variational MPS algorithms and (c) DMRG algorithms. As a reference, all plots contain the error of the vMPS Arnoldi and Lanczos methods (solid black line).

methods, which are nevertheless competitive if we consider their accuracy and the fact that they allow for longer time steps.

In the remaining simulations we dropped the methods from subsections 4.3 and 4.4, because they have a similar computational cost and worse performance than the vMPS Arnoldi method. We keep, on the other hand, all Trotter and DMRG methods and continue our study with bigger problems in which the MPS spaces and the DMRG basis are smaller than the limit required to represent all states accurately, $D = d^{N/2}$ and $M = d^{N/2-1}$. More precisely, we choose $N = 16$ and 20 spins and try with $M, D = 32, 64, 80$ and 128.

Now that the methods are potentially inexact, our previous error laws are modified by the introduction of truncation errors. The first thing we notice in figures 4(a)–(c) is that the Trotter error of second order is rather stable, its error being the same for vMPS and DMRG. However, when we go for higher orders and increase the number of exponentials, truncation errors affect more strongly the Vidal and DMRG implementations than the vMPS one.

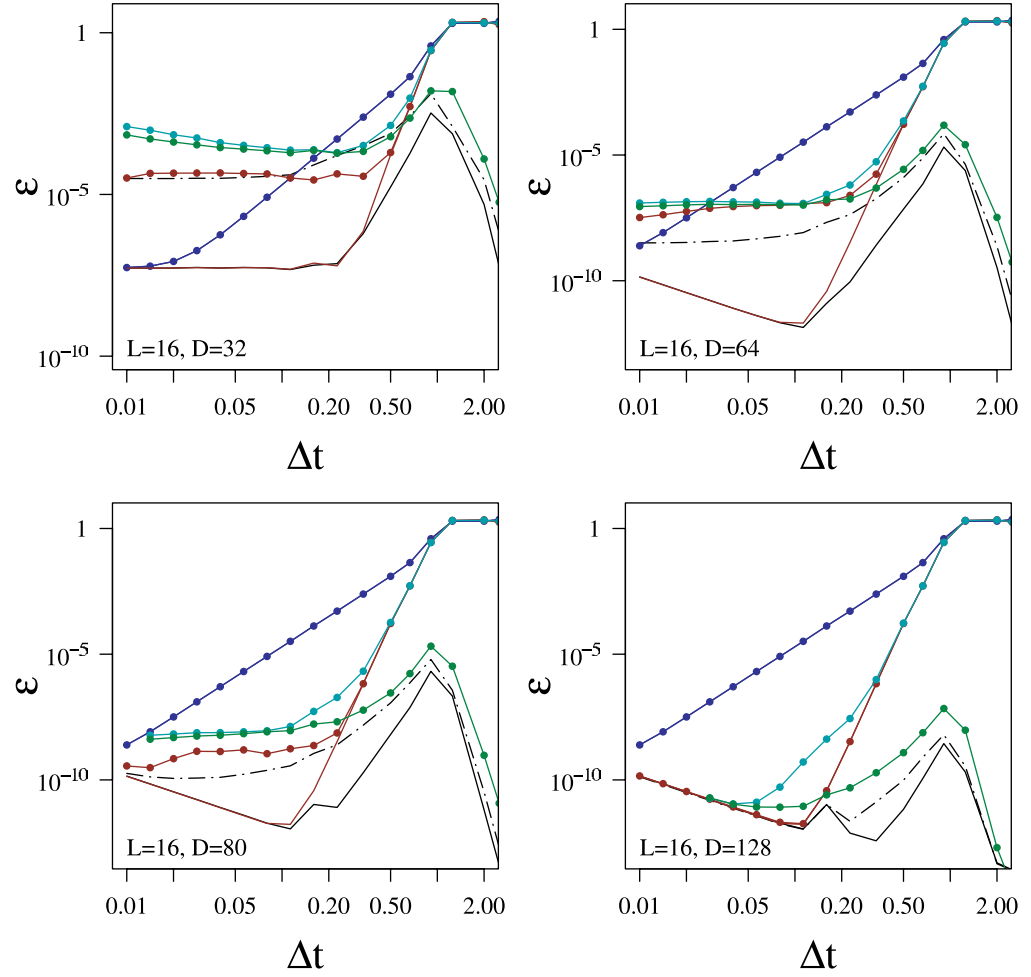


Figure 4. Wavefunction error versus time step, Δt , for simulations of model (32) with 16 spins, $\theta = 0.35$, $\Delta = 0$ and $T = 10$. In (a)–(d), we plot the outcome for different MPS sizes or DMRG basis, denoted by D . The association between methods and line types is that of figure 3.

This shows the difference between making local truncations after each bond unitary (DMRG and Vidal) versus delaying truncations until the end and using the optimal projection [5].

Another important conclusion is that all DMRG methods are very sensitive to truncation errors. As shown in figures 4(a)–(c), there is not a very big difference in accuracy between the DMRG Runge–Kutta and the Lanczos implementation, and both methods are not more accurate than the DMRG Forest–Ruth formula. The main reason why higher order methods from subsection 4.2 do not improve the results is due to estimating the evolution with the matrix of the Hamiltonian on the truncated DMRG basis. Another reason is that under truncation the Lanczos recurrence no longer produces an orthogonal set of vectors.

To prove those statements we compared with a Lanczos method implemented with MPS as explained in subsection 4.5. This method does indeed have a better accuracy than the DMRG ones, which can be attributed to the use of the full Hamiltonian. The errors of the Lanczos are however larger than those of the Arnoldi method for a similar D and we have checked that

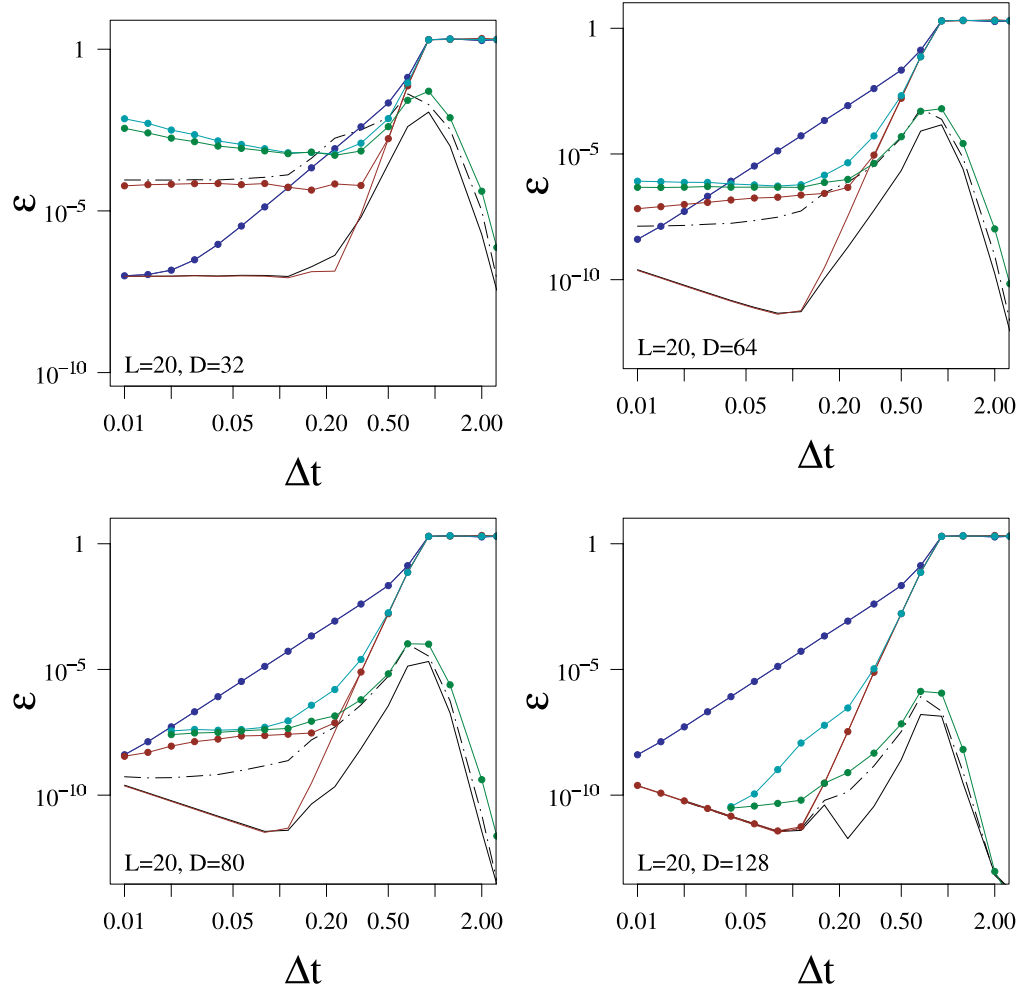


Figure 5. Same as in figure 3 but for $N = 20$ spins.

under truncation the vectors of the Lanczos basis are not truly orthogonal. These small errors accumulate for shorter time steps, and only the Arnoldi method can correct them.

As for the Arnoldi method, it has the greatest accuracy and seems to be stable as D becomes small. For very small matrices the error remains constant as we decrease the time step, but this is only because there is a lower bound in the approximation error given by $\|\mathcal{P}_D|\psi(t)\rangle - |\psi(t)\rangle\|^2$. Figure 6(a) shows how the errors in the Arnoldi method are correlated to the errors made when approximating the exact solution with a MPS of fixed size. This plot illustrates the fact that all errors in this method are due to the final truncation.

Summing up, one should use the method that allows for the longest time steps and the least number of truncations (or applications of \mathcal{P}_D) and mathematical operations. All methods have an optimal time step which is a compromise between the errors in U_n and the rounding and truncation errors made on each step. Regarding performance and accuracy, the two winning methods are variational MPS algorithms using either the fourth order Forest–Ruth decomposition or the Arnoldi basis. The last method, however, has two advantages. One is that it can deal with non-local interactions and the second one is its potential for parallelizability, roughly $\mathcal{O}(N_v N_H / L)$, which most other presented algorithms lack. This can make it competitive with, for instance,

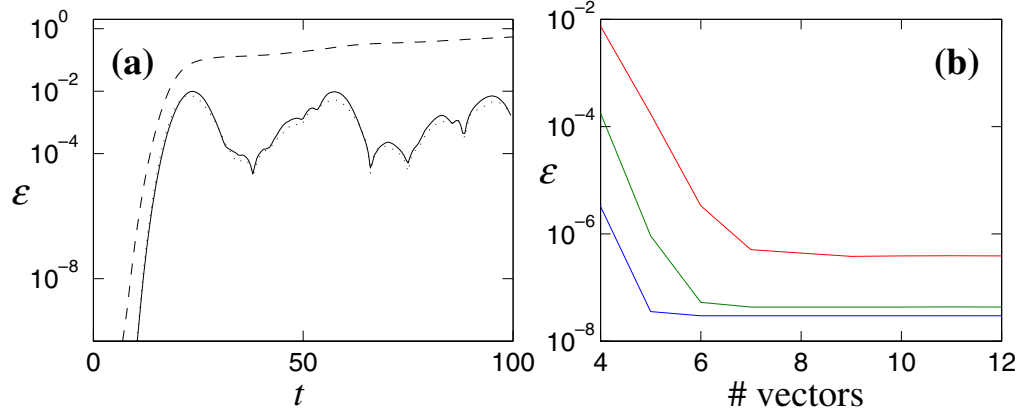


Figure 6. Simulations of $L = 16$ spins with $\theta = 0.35$, $\Delta = 0$. (a) Minimal truncation error $\varepsilon = \|(1 - \mathcal{P}_{64}) \exp(-iHt)|\psi(0)\rangle\|^2$ of the time evolved state and accumulated error made when simulating this Schrödinger equation using the Arnoldi method with 8 vectors, $D = 64$ and $\Delta t = 0.16$ (dashed). (b) Similar as before, errors in the Arnoldi method for varying number of vectors, $T = 10$ and $\Delta t = 0.04, 0.16, 0.32$ (bottom to top).

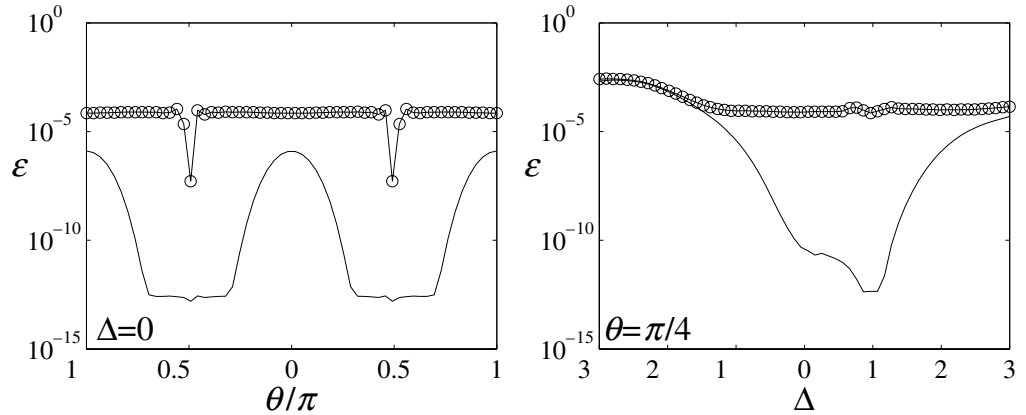


Figure 7. Errors for various other spin $s = 1/2$ models as parametrized in equation (32). We have used a second order vMPS Trotter method (circles) and an Arnoldi method (solid), with 16 spins, $T = 5$ and $D = 32$.

increasing the size of the matrices in the Forest–Ruth method. Regarding DMRG methods, we find that they give comparable results only for a big basis. When truncation errors pop in, their behaviour is less predictable and it does not seem worth going with more elaborate algorithms (Lanczos, Runge–Kutta) versus an ordinary Trotter formula.

The previous results have been confirmed by a systematic scanning of all possible Hamiltonians in equation (32). A selection is shown in figure 7. The Arnoldi method is shown to be accurate, even for gapless problems. When the Arnoldi method fails it is due to truncation errors. In those situations the evolved state cannot be accurately represented by MPS (and for that matter also not by DMRG), simply because $\|(1 - \mathcal{P}_D)|\psi(t)\rangle\|^2$ is finite and large

(figure 3(d)). Increasing the number of Arnoldi vectors will also not help (see figure 6(b)) for the same reason.

6. Simulation of Feshbach resonances

As a real-world application, we have used the vMPS Trotter and Arnoldi methods to simulate the conversion of bosonic atoms into molecules, when confined in an optical lattices and moving through a Feshbach resonance [16]–[18]. The goal is to study how correlation properties are transferred from the atoms into the molecules and how this dynamics is affected by atom motion and conversion efficiency.

The effective model combines the soft-core Bose–Hubbard model used to describe the Tonks gas experiments [32], with a coupling to a molecular state [33]

$$H = -J \sum_{\langle i,j \rangle, \sigma} a_{i\sigma}^\dagger a_{j\sigma} + \sum_{i, \sigma, \sigma'} \frac{U_{\sigma, \sigma'}}{2} a_{i\sigma}^\dagger a_{i\sigma'}^\dagger a_{i\sigma} a_{i\sigma'} + \sum_i \frac{U_{mm}}{2} b_i^\dagger b_i^\dagger b_i b_i + \sum_i [(E_m + U_m n_i^{(a)}) n_i^{(m)} + \Omega [b_i^\dagger a_{i\uparrow} a_{i\downarrow} + \text{H.c.}]]. \quad (33)$$

Here, $a_{i\uparrow}$, $a_{i\downarrow}$ and b_i are bosonic operators for atoms in two internal states and the molecule; $n^{(a)}$ and $n_i^{(m)}$ are the total number of atoms and of molecules on each site, and we have the usual two-level coupling with Rabi frequency Ω and detuning $\Delta := E_m - U_m$. For simplicity, we will assume that atoms and molecules interact strongly among themselves ($U_{\uparrow\uparrow}$, $U_{\downarrow\downarrow}$, $U_{mm} \rightarrow \infty$), so that we can treat them as hard-core, $a_{i\sigma}^2$, b_i^2 , $a_{i\sigma} b_i = 0$. Also since molecules are heavier, we have neglected their tunnelling amplitude, although that could be easily included.

As the energy of the molecular state is shifted from $E_m \gg U_m$ down to $E_m \ll U_m$, the ground state of equation (33) changes from a pair of coupled of Tonks gases, to a purely molecular insulator. We want to study the dynamics of this crossover as E_m is ramped slowly from one phase to the other.

The simplest situation corresponds to no hopping: isolated atoms experience no dynamics, while sites with two atoms may produce a molecule. The molecular and atomic correlations at the end of the process are directly related to two-body correlations in the initial state [34, 35],

$$\langle m_k^\dagger m_k \rangle_{t=T} \sim \langle n_{k\uparrow} n_{k\downarrow} \rangle_{t=0}, \quad (34)$$

$$\langle a_k^\dagger a_k \rangle_{t=T} \sim \langle a_k^\dagger a_k \rangle_{t=0} - \langle n_{k\uparrow} n_{k\downarrow} \rangle_{t=0}. \quad (35)$$

Therefore, this process can thus be used as a tool to probe quantum correlations between atoms. Studying the two-level system $\{a_{k\uparrow}^\dagger a_{k\downarrow}^\dagger |0\rangle, b_k^\dagger |0\rangle\}$, we conclude that for this process to work with a 90% efficiency, the ramping time should be larger than $T \sim 1.5/\Omega$ (see figure 8(b)).

We have simulated numerically the ramping of small lattices, $L = 10$ –32 sites, with an initial number of atoms $N_{\uparrow, \downarrow} = L/2, 3L/4$. The value of the molecular coupling has been fixed to $\Omega = 1$ and the interaction has been ramped according to $E_m = U_{\uparrow\downarrow} + 4\Omega(1 - 2t/T)$ using the ideal ramptime $T = 1.5/\Omega$. We have used two particular values of the inter-species interaction, $U_{\uparrow\downarrow}/\Omega = 0, 2$, and scanned different values of the hopping $J/\Omega \in [0, 0.4]$. The initial condition was always the ground state of the model with these values of $U_{\uparrow\downarrow}/J$ and no coupling. These states contain the correlations that we want to measure.

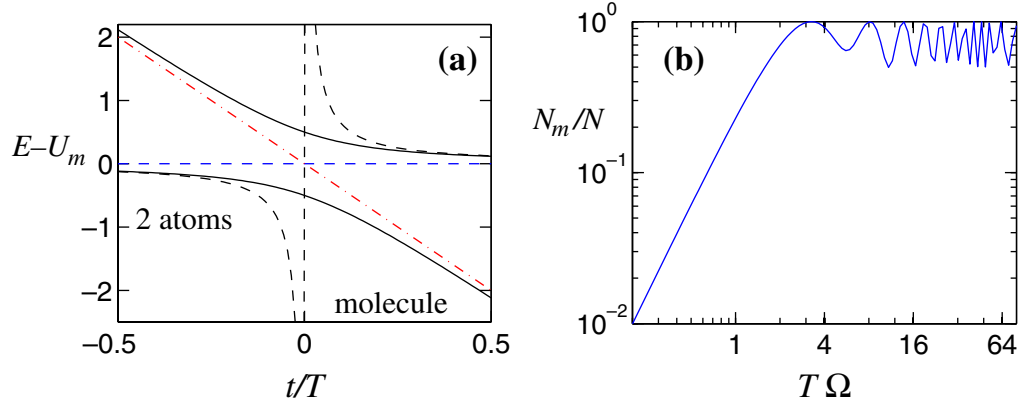


Figure 8. Dynamics of a site with two atoms when the energy of the molecules is ramped linearly: $E_m = U_{\uparrow\downarrow} + 4\Omega(1 - 2t/T)$. We plot (a) the instantaneous energy levels (solid), and (b) the fraction of atoms converted into molecules.

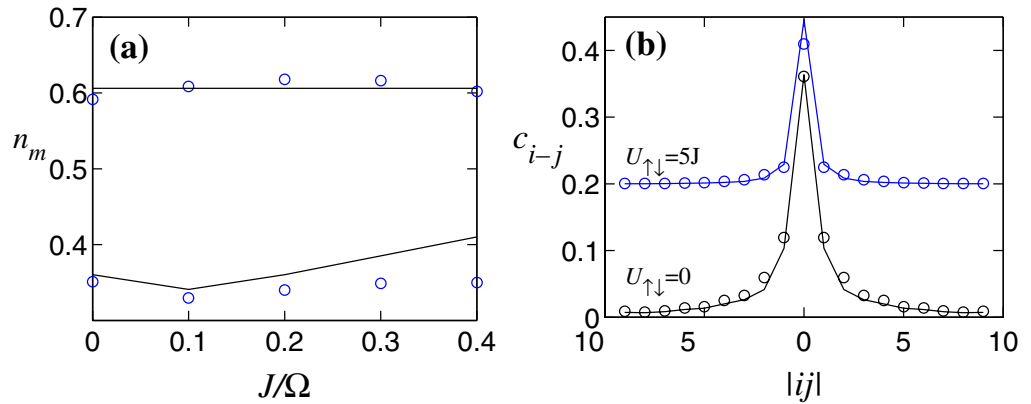


Figure 9. (a) Fraction of atoms converted into molecules versus adimensionalized hopping amplitude, for $U_{\uparrow\downarrow} = 0$ and $U_{\uparrow\downarrow} = 1$, from top to bottom. Circles show the outcome of the numerical experiment, while solid lines contain the ideal fraction (34). The case $J = 0$ uses an initial condition $J = 0.1\Omega$ and then switches off tunnelling before ramping. (b) Correlations of the molecular state, $\langle m_i^\dagger m_j \rangle_{t=T}$ (circles) after the ramp, and those of the initial atomic state, $\langle a_{i\downarrow}^\dagger a_{i\uparrow}^\dagger a_{j\uparrow} a_{j\downarrow} \rangle_{t=0}$ (solid line). The plot for $U_{\uparrow\downarrow} = 5J$ has been shifted up by 0.2.

The main conclusion is that, as shown in figure 9, indeed the correlations of the molecules are almost those of the initial state of the atoms (34), even for $J = 0.4\Omega$ when the process has not been adiabatic. An intuitive explanation is that hopping is strongly suppressed as we approach the resonance, due to the mixing between atomic states, which can hop, and molecular states, which are slower. We can say that the molecules thus pin the atoms and *measure* them. This explanation is supported by a perturbation analysis at $J \ll \Omega$, where one finds that a small molecular contamination slows the atoms on the lattice. This analysis breaks down, however, for $J \sim \Omega$, the regime in which the numerical simulations are required.

7. Conclusions

We have performed a rather exhaustive comparison of different methods for simulating the evolution of big, 1D quantum systems [4, 5], [8]–[14] with three other methods developed in this study. We find the vMPS methods to be optimal both in accuracy and performance within the formulae of similar order. All procedures are substantially affected by truncation and rounding errors, and to fight the latter we must choose large integration time steps. However, the only algorithm which succeeds for very large time steps is an Arnoldi method developed in this study. Finally, this algorithm can be applied to problems with long-range interactions.

Using this algorithm, we have simulated the dynamics of cold atoms in a 1D optical lattice when crossing a Feshbach resonance. The main conclusion is that with rather fast ramp times it is possible to map the correlations of the atomic cloud (two Tonks gases in this case) and use this as a measuring tool in current experiments. This result connects with similar theoretical predictions for fermions in [34, 35]. Simple generalizations of this study will allow us in the future to analyse losses and creation of strongly correlated states with the help of the molecular component.

As a possible outlook, we envision the possibility of developing new algorithms in which the state is approximated by a linear combination of MPS at all times. This should be more efficient than increasing the size of the matrices, and could support distributed computations in a cluster.

References

- [1] Peschel I, Wang X and Kaulke M (ed) 1999 *Density-Matrix Renormalization* (Berlin: Springer)
- [2] Ostlund S and Rommer S 1995 Thermodynamic limit of density matrix renormalization *Phys. Rev. Lett.* **75** 3537–40
- [3] Verstraete F, Porras D and Cirac J I 2004 Density matrix renormalization group and periodic boundary conditions: a quantum information perspective *Phys. Rev. Lett.* **93** 227205
- [4] Vidal G 2004 Efficient simulation of one-dimensional quantum many-body systems *Phys. Rev. Lett.* **93** 040502
- [5] Verstraete F, García-Ripoll J J and Cirac J I 2004 Matrix product density operators: simulation of finite-temperature and dissipative systems *Phys. Rev. Lett.* **93** 207204
- [6] Zwolak M and Vidal G 2004 Mixed-state dynamics in one-dimensional quantum lattice systems: a time-dependent superoperator renormalization algorithm *Phys. Rev. Lett.* **93** 207205
- [7] Porras D, Verstraete F and Cirac J I 2006 Renormalization algorithm for the calculation of spectra of interacting quantum systems *Phys. Rev. B* **73** 014410
- [8] White S R 1993 Density-matrix algorithms for quantum renormalization groups *Phys. Rev. B* **48** 10345–56
- [9] Daley A J, Kollath C, Schollwöck U and Vidal G 2004 Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces *J. Stat. Mech.: Theor. Exp.* **P04005**
- [10] Gobert D, Kollath C, Schollwöck U and Schütz G 2005 Real-time dynamics in spin-1/2 chains with adaptive time-dependent DMRG *Phys. Rev. E* **71** 036102
- [11] Feiguin A E and White S R 2005 Time-step targeting methods for real-time dynamics using the density matrix renormalization group *Phys. Rev. B* **72** 020404(R)
- [12] Schmitteckert P 2004 Nonequilibrium electron transport using the density matrix renormalization group method *Phys. Rev. B* **70** 121302(R)
- [13] Manmana S R, Muramatsu A and Noack R M 2005 Time evolution of one-dimensional quantum many body systems *AIP Conf. Proc.* **789** 269–78
- [14] Noack R M and Manmana S R 2005 Collapse and revival starting from a Luttinger liquid *AIP Conf. Proc.* **816** 198–203

- [15] Verstraete F and Cirac J I 2006 Matrix product states represent ground states faithfully *Phys. Rev. B* **73** 094423
- [16] Thalhammer G, Winkler K, Lang F, Schmid S, Grimm R and Denschlag J H 2006 Long-lived Feshbach molecules in a three-dimensional optical lattice *Phys. Rev. Lett.* **96** 050402
- [17] Stoferle T, Moritz H, Gunter K, Kohl M and Esslinger T 2006 Molecules of fermionic atoms in an optical lattice *Phys. Rev. Lett.* **96** 030401
- [18] Volz T, Syassen N, Bauer D M, Hansis E, Durr S and Rempe G 2006 A Mott state of molecules *Preprint cond-mat/0605184*
- [19] Vidal G 2003 Efficient classical simulation of slightly entangled quantum computations *Phys. Rev. Lett.* **91** 147902
- [20] Schollwöck U and White S R 2006 Methods for time dependence in DMRG *Effective Models for Low-dimensional Strongly Correlated Systems* ed G G G Batrouni and D Poilblanc (New York: AIP)
- [21] Cazalilla M A and Marston J B 2002 Time-dependent density-matrix renormalization group: a systematic method for the study of quantum many-body out-of-equilibrium systems *Phys. Rev. Lett.* **88** 256403
- [22] Cazalilla M A and Marston J B 2003 Cazalilla and marston reply *Phys. Rev. Lett.* **91** 049702
- [23] Luo H G, Xiang T and Wang X Q 2003 Comment on ‘time-dependent density-matrix renormalization group: a systematic method for the study of quantum many-body out-of-equilibrium systems’ *Phys. Rev. Lett.* **91** 049701
- [24] Winkler K, Thalhammer G, Lang F, Grimm R, Hecker Denschlag J, Daley A J, Kantian A, Büchler H P and Zoller P 2006 Repulsively bound atom pairs in an optical lattice *Nature* **441** 853–6
- [25] Kollath C, Schollwöck U and Zwirger W 2005 Spin-charge separation in cold Fermi gases: a real time analysis *Phys. Rev. Lett.* **95** 176401
- [26] Micheli A, Daley A J, Jaksch D and Zoller P 2004 Single atom transistor in a 1d optical lattice *Phys. Rev. Lett.* **93** 140408
- [27] Clark S R and Jaksch D 2004 Dynamics of the superfluid to Mott-insulator transition in one dimension *Phys. Rev. A: At. Mol. Opt. Phys.* **70** 043612
- [28] Omelyan I P, Mryglod I M and Folk R 2002 Optimized Forest–Ruth- and Suzuki-like algorithms for integration of motion in many-body systems *Comput. Phys. Commun.* **146** 188–202
- [29] Noack R M and Manmana S R 2005 Diagonalization- and numerical renormalization-group-based methods for interacting quantum systems *AIP Conf. Proc.* **789** 93–163
- [30] Hochbruck M and Lubich C 1996 On Krylov subspace approximations to the matrix exponential operator *SIAM J. Numer. Anal.* **34** 1911–25
- [31] Moler C and Van Loan C 2003 Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later *SIAM Rev.* **45** 3–49
- [32] Paredes B, Widera A, Murg V, Mandel O, Fölling S, Cirac I, Shlyapnikov G V, Hänsch T W and Bloch I 2004 Tonks-Girardeau gas of ultracold atoms in an optical lattice *Nature* **429** 277–81
- [33] Dickerscheid D B M, AlKhawaja U, van Oosten D and Stoof H T C 2005 Feshbach resonances in an optical lattice *Phys. Rev. A* **71** 043604
- [34] Altman E and Vishwanath A 2005 Dynamic projection on Feshbach molecules: a probe of pairing and phase fluctuations *Phys. Rev. Lett.* **95** 110404
- [35] Barankov R A and Levitov L S 2005 Dynamical projection of atoms to Feshbach molecules at strong coupling *Preprint cond-mat/0506323*