One-site density matrix renormalization group and alternating minimum energy algorithm

Sergey V. Dolgov\(^*$\) and Dmitry V. Savostyanov\(^{†}$

December 23, 2013

Abstract

Given in the title are two algorithms to compute the extreme eigenstate of a high-dimensional Hermitian matrix using the tensor train (TT) / matrix product states (MPS) representation. Both methods empower the traditional alternating direction scheme with the auxiliary (e.g. gradient) information, which substantially improves the convergence in many difficult cases. Being conceptually close, these methods have different derivation, implementation, theoretical and practical properties. We emphasize the differences, and reproduce the numerical example to compare the performance of two algorithms.

Keywords: high–dimensional problems, DMRG, MPS, tensor train format, extreme eigenstate.

MSC: 15A18, 15A69, 65F10, 65F15, 82B28, 82B20

PACS: 02.10.Xm, 02.60.Dc, 75.10.Pq, 05.10.Cc

1 Introduction

Actual problems of science, engineering and society can be so complex, that their mathematical portrait requires more than three dimensions. Quantum world gives us a perfect example of essentially high–dimensional systems, described by a joint wavefunction (or density matrix) of all particles. A simple system of \(d\) spin-\(\frac{1}{2}\) particles is an entanglement of \(\mathcal{O}(2^d)\) possible states, and should be described by the same amount of numbers, which creates out-of-memory errors on a typical workstation for \(d \gtrsim 30\). Even with a brute force of modern supercomputers, standard numerical methods can not honestly simulate protein-size molecules (\(d \sim 10^3 \cdots 10^4\)), since the complexity and storage explode exponentially with \(d\).

To overcome this problem, known as the curse of dimensionality, we use data-sparse representations for high-dimensional vectors and matrices, and develop

\(^*$\)Max-Planck Institute for Mathematics in the Sciences, Inselstraße 22, Leipzig 04103, Germany (sergey.v.dolgov@gmail.com)

\(^{†}$University of Southampton, School of Chemistry, Highfield Campus, Southampton SO17 1BJ, United Kingdom (dmitry.savostyanov@gmail.com)
special algorithms to work with them. Proposed in 1992, the density matrix renormalization group (DMRG) algorithm \[10\] and the matrix product states (MPS) formalism \[3\] suggest to represent a wavefunction $x$ in the following tensor-product form

$$
x = \tau(x^{1}, \ldots, x^{d}) = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} x_{\alpha_1}^{(1)} \otimes x_{\alpha_1\alpha_2}^{(2)} \otimes \cdots \otimes x_{\alpha_{d-1}}^{(d)},
$$

In numerical linear algebra this format was re-discovered as the tensor train (TT) decomposition \[7,6\]. A single TT core (or site) $x^{(k)} = [x^{(k)}_{\alpha_{k-1}\alpha_k}(i_k)]$ is described by $r_{k-1} n_k r_k$ numbers, where $n_k$ denotes the number of possible states for the $k$-th particle (the mode size), and $r_k$ is the TT rank (or bond dimension). The total number of representation parameters scales as $O(dn^2r^2)$, $n \sim n_k$, $r \sim r_k$, and is feasible for computations with $d, n, r \lesssim 10^3$.

The DMRG algorithm was originally proposed to find the ground state, i.e. the minimal eigenpair of a Hermitian matrix $A$. This problem is equivalent to the minimization of the Rayleigh quotient $J_A(x) = (x, Ax)/x, x)$. Substituting $J_A(x)$ with $K_A(x) = (x, Ax) - 2\Re(x, y)$, and applying the same algorithm, we can solve linear systems $Ax = y$ with Hermitian positive definite matrix \[4\]. This framework can be extended to a broad class of problems.

Since $x$ is a huge high-dimensional vector, the solution is sought in the structured format \(\tau\) with some TT ranks $r_k$, defined a priori or chosen adaptively. The simultaneous optimization over all sites is a highly nonlinear and difficult problem. As it is usual in high-dimensional optimization, we substitute it by a sequence of partial optimizations, each over a particular (small) group of variables. For our problem, it is natural to group the variables according to the tensor format \(\tau\), e.g. optimize over the components of a single site $x^{(k)}$ at a time.

The TT format is linear in each site, i.e. $x = \tau(x^{1}, \ldots, x^{d}) = X_{\alpha k}^{(k)} x^{(k)}$, where $X_{\alpha k}$ is the $(n_1, \ldots, n_d) \times (r_{k-1} n_k r_k)$ frame matrix, which linearly maps the elements of $x^{(k)}$ to the full vector $x$. This turns every partial optimization into a local problem.

Figure 1: Sequence of low-dimensional optimizations in subspaces $X_1, X_2, X_3, \ldots$ (left), and $X_{1,2}, X_{2,3}, X_{3,4}, \ldots$ (right)
of the same type, as the original one,
\[ x^{(k)} = \arg \min_{x^{(k)}} J_A(x^{(1)}, \ldots, x^{(d)}) = \arg \min_{x^{(k)}} J_A(x^{(k)}), \tag{2} \]

where \( A_k = X^{(k)}_{\neq k} A X^{(k)}_{\neq k} \) is the \( (r_{k-1} n_k r_{k}) \times (r_{k-1} n_k r_{k}) \) reduced matrix, which inherits the properties of \( A \), i.e. is Hermitian. Since the frame matrix \( X^{(k)}_{\neq k} \) has a structured TT representation (which is the same as \( \mathbb{I} \) with \( x^{(k)} \) substituted by the identity matrix), the reduced matrix \( A_k \) can be assembled avoiding the exponential costs. Finally, introducing simple orthogonality conditions for all sites but \( x^{(k)} \), we can make the whole matrix \( X^{(k)}_{\neq k} \) orthogonal [9]. As a consequence, \( A_k \) becomes better conditioned than \( A \), and the reduced functional writes \( J_{A_k}(x) = (x, A_k x)/(x, x) \) for the ground state problem, and \( K_{A_k}(x) = (x, A_k x) - 2\mathbb{R}(x, X^{(k)}_{\neq k} y) \) for the linear system. Each optimization (2) is now a classical problem of a tractable size, that can be solved by classical algorithms of numerical linear algebra.

Each local step (2) finds \( \min_{x \in X_k} J_A(x) \), where the subspace \( X_k = \text{span} X^{(k)}_{\neq k} \) is of dimension \( r_{k-1} n_k r_{k} \), see Fig. I (left). Here and later by \( \text{span} X \) we denote the subspace of columns of a matrix \( X \). If TT ranks are fixed, the local convergence of such scheme can be analysed using standard methods of multivariate analysis [5]. However, in numerical practice the tensor ranks of the solution are not known in advance, and fixed-rank optimization with wrong ranks would not be efficient. The DMRG scheme with variable TT ranks is more advantageous, but the theoretical analysis is even more difficult.

When we allow TT ranks to grow, the dimensions of subspaces \( X_k \) grow as well, and we can use different strategies to expand the subspaces. Originally, the one-site DMRG scheme (DMRG1) increased the rank \( r_k \) by adding (random) orthogonal vectors to \( X_k \), but this algorithm often got stuck far from the ground state. The problem was solved using \textit{two sites} instead of one in the optimization step [10]. The \textit{two-site} DMRG algorithm (DMRG2) merges blocks \( x^{(k)} \) and \( x^{(k+1)} \), and solves the local optimization problem in \( X_{k,k+1} = \text{span} X^{(k)}_{\neq[k,k+1]} \), see Fig. I (right). Here \( X^{(k)}_{\neq[k,k+1]} \) is the \( (n_1 \ldots n_d) \times (r_{k-1} n_k r_{k+1} r_{k+1}) \) matrix, which has the same TT representation as \( \mathbb{I} \) with blocks \( x^{(k)} \) and \( x^{(k+1)} \) replaced by the identities. The DMRG2 converges remarkably well (and is in fact a method of choice) for 1D systems with short–range interactions, but the cost is approximately \( n \) times larger than in the DMRG1. For systems with long-range interactions two neighboring sites do not provide sufficient information, and DMRG2 can stagnate as well. To simulate such systems faster and more accurately, better methods to choose search subspaces are required.

The \textit{gradient direction} is central in the theory of optimization methods, and many algorithms use the gradient or its approximate surrogates. In 2005, S. White proposed the \textit{corrected one-site} DMRG algorithm (DMRG1c), which adds auxiliary direction to improve the convergence and reduce the computational cost, see [11] and [9] Sec 6.3 for more details. In this paper we compare the DMRG1c with the \textit{alternating minimum energy} (AMEn) algorithm. The AMEn algorithm was recently proposed in [12] for the solution of linear equations, and the version for the ground state problem appears immediately when we choose \( J_A(x) \) as a target function. In the next section we compare the ideas and implementation aspects of both methods and explain the motivation behind AMEn from numerical linear algebra perspective. In Sec. 3 we reproduce a numerical experiment of S. White from [11], and demonstrate that AMEn can solve it better than DMRG1c.
2 Comparison of methods

Both DMRG1c and AMEn combine the local optimization \( \mathcal{O} \) with the step that injects the auxiliary information. Both algorithms are local, i.e., modify only one block \( x^{(k)} \) at a time (cf. the non-local “ALS\((t + z)\)” algorithm in [1]). Both methods sequentially cycle over TT blocks \( (1, 2, \ldots, d, d-1, \ldots) \). In the following we assume that \( \mathcal{O} \) was just solved for \( x^{(k)} \), and consider the step that corrects \( x^{(k)} \) before the optimization passes to the next block \( x^{(k+1)} \). This step does not change the vector \( x = \tau(x^{(1)}, \ldots, x^{(d)}) \) for AMEn, or perturbs it slightly (for DMRG1c), and therefore has a minor direct effect on \( J_A(x) \).

It is crucial how exactly the block \( x^{(k)} \) is modified, and which vectors end up in span \( X_{\neq k+1} \), that improves the convergence of \( J_A(x) \) to its global minimum.

2.1 Which vector is targeted: \( p = Ax \) vs. \( z = Ax - J_A(x)x \)

Following the power iteration method, the DMRG1c algorithm of S. White targets in addition to the solution \( x \) the first Krylov vector \( p = Ax \). The AMEn algorithm uses the gradient direction \( z = Ax - J_A(x)x \). In exact arithmetics this makes no difference, since span\(\{x, p\} = \text{span}\{x, z\}\). In practical computations both \( p \) and \( z \) are perturbed by inevitable machine rounding errors, perturbations associated with the use of tensor format, and additional errors which appear when a surrogate formula (like [11, Eq. (14)]) is used to speed up the computations. The DMRG1c algorithm is derived from perturbation arguments, valid in the vicinity of the minimum of \( J_A(x) \). When \( x \) approaches the ground state, the angle between \( x \) and \( p = Ax \) vanishes, and any perturbation in \( Ax \) yields a random new direction. This creates a certain gap between the theory supporting the DMRG1c, and the practice.

Following the steepest descent algorithm, the AMEn uses orthogonal vectors \( z \) and \( x \), and span\(\{x, z\} \) is much more stable to perturbations of \( z \). (In general, the Krylov vectors \( \{x, Ax, A^2x, \ldots\} \) form an extremely unstable basis, and orthogonalization is crucial.) The steepest descent algorithm with \( z \) substituted by \( \tilde{z} \) converges as long as \( (\tilde{z}, z) > 0 \). For the linear systems this fact is elegantly proven in [5], and the convergence rate of perturbed method is estimated. An eigenvalue counterpart follows similarly, and the rate of convergence in span\(\{x, \tilde{z}\} \) can be estimated from the spectral range of \( A \). This makes the approach implemented in the AMEn algorithm preferable both theoretically and in practice.

2.2 What is approximated: subspace \( \tilde{P} \) vs. vector \( \tilde{z} \)

The computation of full vectors \( p = Ax \) and \( z = Ax - J_A(x)x \) is not possible due to their exponentially large size. Since \( x \) and \( A \) are both in TT format, we can avoid the curse of dimension and represent \( p = Ax \) and \( z = Ax - J_A(x)x \) by the TT format. However, the TT ranks of \( Ax \) can be as large as product of TT ranks of \( A \) and \( x \), which makes the calculations difficult.

To reduce these costs, S. White suggests in the DMRG1c the following scheme. The TT format [1] is divided in two parts: left blocks (number 1, \ldots, k) are referred
to as system, and right blocks \((k + 1, \ldots, d)\) as environment. The TT format for the matrix \(A\) is written accordingly,

\[
A = \sum_{\gamma_1 \cdots \gamma_{d-1}} A_{\gamma_1}^{(1)} \otimes \cdots \otimes A_{\gamma_{k-1} \gamma_k}^{(k)} \otimes A_{\gamma_k \gamma_{k+1}}^{(k+1)} \otimes \cdots \otimes A_{\gamma_{d-1}}^{(d)},
\]

or shortly \(A = \sum_{\gamma} A_{\gamma}^\varsigma \otimes A_{\gamma}^\varsigma\). Similarly, Eq. (1) reduces to \(x = \sum_{\alpha} x_{\alpha}^\varsigma \otimes x_{\alpha}^\varsigma\). The targeting of \(p = Ax\) is substituted by targeting of all \(p_{\gamma} = (A_{\gamma}^\varsigma \otimes I)x\).

Although in general \(p \notin \cup_{\gamma} \text{span}\, p_{\gamma}\), it can be argued that the set \(\{p_{\gamma}\}\) contains a sufficient subspace information. To show this, we write

\[
p = \sum_{\alpha, \gamma} (A_{\gamma}^\varsigma x_{\alpha}^\varsigma) \otimes (A_{\gamma}^\varsigma x_{\alpha}^\varsigma), \quad p_{\gamma} = \sum_{\alpha} (A_{\gamma}^\varsigma x_{\alpha}^\varsigma) \otimes x_{\alpha}^\varsigma,
\]

and consider vectors \(p\) and \(p_{\gamma}\) as system-by-environment matrices \(P\) and \(P_{\gamma}\) of size \((n_1 \ldots n_k) \times (n_{k+1} \ldots n_d)\). Now span \(P \subset \cup_{\gamma} \text{span}\, P_{\gamma} = \bar{P}\), where \(A_{\gamma}^\varsigma\) contains the coefficients of the required linear combination — in the exact arithmetics the system-related components of \(p\) belong to \(\bar{P}\). Each \(p_{\gamma}\) is easier to compute than \(p\), because it does not depend on the environment part \(A_{\gamma}^\varsigma\).

The total dimension of \(\bar{P}\) grows in each step, and to keep TT ranks and storage moderate, we have to truncate it. The approximation step in the DMRG1c replaces \(\bar{P}\) with a subspace \(\bar{P}\) of a smaller dimension, using a classical singular value decomposition (SVD), or Schmidt decomposition technique. The dominant subspace \(\bar{P}\) is spanned by the first singular vectors of the matrix \(X \sqrt{\alpha_1} P_1 \sqrt{\alpha_2} P_2 \ldots\), where all target vectors are concatenated with empirically chosen weighting coefficients \(\alpha_\gamma\). The method assumes that the vector \(p = Ax\) is likely to belong to \(\bar{P}\).

This assumption makes perfect sense if \(p\) is a random sample from \(\bar{P}\) — for a random \(u\), \(Xu\) is more likely to end up in the dominant subspace of \(X\). However, the target vector \(p = Ax\) does not belong to \(\bar{P}\) in general, for any choice of weights \(\sqrt{\alpha_\gamma}\). The reason is that \(p\) depends crucially on \(A_{\gamma}^\varsigma\), whereas this information is dropped for the sake of faster computations in \(p_{\gamma}\) and hence \(P\) and \(\bar{P}\). Selecting \(A_{\gamma}^\varsigma\) in \(4\), we may come across any vector in \(P\), even the smallest singular vector. That is, for each choice of \(\sqrt{\alpha_\gamma}\) and \(x\) there is a counterexample of a Hamiltonian, for which the slightest truncation of \(P\) loses the system-related part of the target vector \(p = Ax\).

The AMEn approximates \(z = Ax - J_A(x)x\) into its own TT format using any compression tool. Either the SVD-based technique, which computes the approximation \(\tilde{z} \approx z\) up to any prescribed tolerance \(\varepsilon\), or a faster (but heuristic) alternating least squares (ALS) method may be used. In any case, we may generate an approximation \(\tilde{z}\) with a desired accuracy, which guarantees the convergence of the steepest descent method with the imperfect direction \(\tilde{z}\). This fact provides the theoretical bounds for the global convergence rate of the whole AMEn scheme, similarly to \[2\].

### 2.3 How the new direction is used: averaging vs. enrichment

The last but not the least detail is how exactly the information about the auxiliary direction is injected in the algorithm. To show this in isolation from the other dissimilarities outlined above, we assume that in both methods we target in addition
to \(x\) only one vector \(s\). To simplify the presentation we also consider the \(d = 2\) case, and write \(x = \sum_{\alpha=1}^{r_x} x^{\alpha} x^{\alpha}\), and \(s = \sum_{\beta=1}^{r_s} s^{\beta} s^{\beta}\), where “\(<\)” and “\(>\)” denote the first and the second blocks, respectively.

The DMRG1c averages the subspaces \(X = [x_1^< \ldots x_{K}^<]\) and \(S = [s_1^< \ldots s_{K}^<]\) by computing the dominant subspace span \(U\) of the Gram matrix as follows, \(G = XX^* + \alpha SS^* \approx UU^*\), where \(U = [u_1^< \ldots u_{K}^<]\). As shown in previous subsection, this procedure does not guarantee that \(x\) or \(s\) ends up in span\([U \otimes I]\), unless span \(U = \text{span}\{X S\}\). The TT core \(x^<\) is replaced by the vectors of \(U\), that introduces a \(O(\sqrt{d})\) perturbation to \(x\) and probably worsen \(J_A(x)\). It is clear though that \(x\) should vanish when we approach the exact solution, but the general recipe is not known.

The AMEn avoids the outlined difficulties by merging \(U = [X S]\) and zero-padding the second block. Values of \(x\) and \(J_A(x)\) are preserved, no rescaling is required, and both \([x, s] \in \text{span}\{U \otimes I\}\). The downside is that we choose \(r_u = r_x + r_s\) each time we expand the subspaces. However, when we use the approximate gradient direction \(s = \tilde{z} \approx z = Ax - J_A(x)x\) the low-rank \(\tilde{z}\) usually suffice, e.g. with \(r_s \approx r_x/2\). We can also truncate the TT-ranks at the end of each iteration and control the perturbation to \(J_A(x)\).

### 3 Numerical example

Following S. White \cite{11}, we consider the spin-1 periodic Heisenberg chain,

\[ A = H_1 \cdot H_2 + H_2 \cdot H_3 + \ldots + H_{d-1} \cdot H_d + H_d \cdot H_1, \]

\[ H_i = (H_i^x, H_i^y, H_i^z)^T, \quad H_i \cdot H_1 = H_i^x H_1^x + H_i^y H_1^y + H_i^z H_1^z, \]

\[ H_i^{[x,y,z]} = I \otimes \cdots \otimes I \otimes S_{[x,y,z]} \otimes I \otimes \cdots \otimes I, \quad S \text{ in position } i, \]

where \(S_{[x,y,z]}\) are the \(3 \times 3\) Pauli matrices for spin-1 particles. The number of spins \(d\) is set to \(100\), i.e. the wavefunction belongs to the \(3^{100}\)-dimensional Hilbert space. This example is particularly illustrating, since the mismatch between the linear TT model (1) and the cycle structure of (5) complicates the problem — the solution has large TT ranks, and both the one- and two-site DMRG converge slowly.

The way how the TT ranks are chosen during the algorithm is also very important. We first adopt the rank selection strategy from \cite{11}, and compare the DMRG2, the DMRG1c and the AMEn algorithms. The results are shown in Fig. 2 (top left), which overlays \cite{11} Fig. 3 with the AMEn behavior. In Fig. 2 (top right) the convergence of \(\lambda = J_A(x)\) to the reference value \(\lambda_s = -140.14840390392\) (computed in \cite{11} by the DMRG1c with TT ranks 4000) is given w.r.t. the cumulative CPU time.

We see that both DMRG methods correctly reproduce the experiment from \cite{11}: the two-site DMRG stagnates at a high error level, while the corrected DMRG converges significantly faster. The AMEn method manifests practically the same efficiency. Since it searches in a larger subspace, it is even more accurate w.r.t. iterations, but becomes slightly slower during the optimization of inner TT blocks. However, letting it to increase the ranks (each fourth iteration) yields sharper error decays.

To free the algorithm from tuning parameter, we prefer to choose the ranks adaptively to the desired accuracy. With this we also avoid artificial rank limita-
Figure 2: Error in the eigenvalue vs. iteration (left) and CPU time (right). Methods: AMEn [1, 2], DMRG1c [11], DMRG2 [10]. Top: parameters depend on iteration as shown on top of left figure (ranks and $\log_{10}(1/a)$, resp.). Bottom: $a = 10^{-4}$, ranks depend on accuracies: $\epsilon = 10^{-3}$ (solid lines), $\epsilon = 10^{-4}$ (dashed lines).

We see that when ranks are chosen adaptively, the AMEn rapidly becomes faster than the other algorithms. Even the DMRG1c stagnates relatively early, since the correction $p_\gamma(4)$ contaminates the dominant basis of the ground state. Moreover, since the $\epsilon$-truncation eliminates the correction if $a \lesssim \epsilon^2$, it is worthless to decrease the scale $a$ (cf. Fig 2 top left). Both the adaptivity and speed speak in favour of such truncation: the same accuracy levels are achieved several times faster than in the fixed-rank experiment (e.g. 10 vs. 100 sec. for $\lambda - \lambda_\ast \approx 10^{-2}$ and $\epsilon = 10^{-3}$). Larger time spent by AMEn in the latter iterations is compensated by a significantly better accuracy, which is close to the optimal level $O(\epsilon^2)$.

Finally, the AMEn is applicable to a wider class of problems, and is free from heuristic parameters.
References

[1] S. V. Dolgov and D. V. Savostyanov, Alternating minimal energy methods for linear systems in higher dimensions. Part I: SPD systems, arXiv preprint 1301.6068, 2013.

[2] ———, Alternating minimal energy methods for linear systems in higher dimensions. Part II: Faster algorithm and application to nonsymmetric systems, arXiv preprint 1304.1222, 2013.

[3] M. Fannes, B. Nachtergaele, and R. Werner, Finitely correlated states on quantum spin chains, Communications in Mathematical Physics, 144 (1992), pp. 443–490.

[4] E. Jeckelmann, Dynamical density–matrix renormalization–group method, Phys Rev B, 66 (2002), p. 045114.

[5] H. Munthe-Kaas, The convergence rate of inexact preconditioned steepest descent algorithm for solving linear systems, Numerical Analysis Report NA-87-04, Stanford University, 1987.

[6] I. V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317.

[7] I. V. Oseledets and E. E. Tyrtyshnikov, Breaking the curse of dimensionality, or how to use SVD in many dimensions, SIAM J. Sci. Comput., 31 (2009), pp. 3744–3759.

[8] T. Rohwedder and A. Uschmajew, Local convergence of alternating schemes for optimization of convex problems in the TT format, SIAM J. Num. Anal., 51 (2013), pp. 1134–1162.

[9] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Annals of Physics, 326 (2011), pp. 96–192.

[10] S. R. White, Density matrix formulation for quantum renormalization groups, Phys. Rev. Lett., 69 (1992), pp. 2863–2866.

[11] S. R. White, Density matrix renormalization group algorithms with a single center site, Phys. Rev. B, 72 (2005), p. 180403.