Computer Physics Communications 185 (2014) 1207–1216

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Computation of extreme eigenvalues in higher dimensions using block tensor train format

S.V. Dolgov [a,b], B.N. Khoromskij [a], I.V. Oseledets [b,d], D.V. Savostyanov [b,c,*]

[a] Max-Planck Institute for Mathematics in the Sciences, Inselstrasse 22, Leipzig 04103, Germany
[b] Institute of Numerical Mathematics of Russian Academy of Sciences, Gubkina 8, Moscow 119333, Russia
[c] University of Southampton, School of Chemistry, Highfield Campus, Southampton SO17 1BJ, United Kingdom
[d] Skolkovo Institute of Science and Technology, Novaya Str. 100, 143025 Skolkovo, Moscow Region, Russia

ARTICLE INFO

Article history:
Received 12 June 2013
Received in revised form
25 November 2013
Accepted 18 December 2013
Available online 27 December 2013

Keywords:
High-dimensional problems
DMRG
MPS
Tensor train format
Low-lying eigenstates

ABSTRACT

We consider approximate computation of several minimal eigenpairs of large Hermitian matrices which come from high-dimensional problems. We use the tensor train (TT) format for vectors and matrices to overcome the curse of dimensionality and make storage and computational cost feasible. We approximate several low-lying eigenvectors simultaneously in the block version of the TT format. The computation is done by the alternating minimization of the block Rayleigh quotient sequentially for all TT cores. The proposed method combines the advances of the density matrix renormalization group (DMRG) and the variational numerical renormalization group (vNRG) methods. We compare the performance of the proposed method with several versions of the DMRG codes, and show that it may be preferable for systems with large dimension and/or mode size, or when a large number of eigenstates is sought.

1. Introduction

High-dimensional problems are notoriously difficult to solve by standard numerical techniques due to the *curse of dimensionality*—the complexity grows exponentially with the number of degrees of freedom. Such problems arise in various applications in physics, chemistry, biology and engineering, but their study in numerical linear algebra has begun quite recently.

Not many techniques are known to be capable of solving high-dimensional problems efficiently. Among the most prominent are Monte Carlo and quasi Monte Carlo methods, best N-term approximations, and advanced discretization methods such as sparse grids and radial basis functions. However, all of these methods have their own disadvantages. For example, it is difficult to achieve high accuracy using the Monte Carlo approach. In turn, sparse grid techniques require sophisticated analytical and algebraic manipulations, although still suffering (in a milder way though) from the curse of dimensionality, which makes them inapplicable for $d \gtrsim 10$.

One of the most fruitful ideas for solving high-dimensional problems is the *separation of variables*. For two variables it boils down to the celebrated Schmidt decomposition, which is known in matrix calculus as the singular value decomposition (SVD), a particular low-rank decomposition of a matrix. Various generalizations of this idea to higher dimensions have been studied, most notable are the canonical (CP) and Tucker formats, motivated by the applications in data analysis (e.g. chemometrics), see [1]. These classical formats have their drawbacks as well: the CP format is in general not stable to perturbations, and the Tucker format suffers from the curse of dimensionality. Nevertheless, in many applications the canonical representation can be computed efficiently using, e.g. *greedy algorithms* [2–4] or by a multigrid accelerated reduced higher order SVD combined with the Tucker format [5]. The Tucker approximation can be computed reliably using the SVD algorithm [6], or using a fast (but heuristic) cross interpolation algorithm [7].

Efficient methods for quantum many-body systems are based on low-parametric tensor product formats. One of the most successful approaches, the *density matrix renormalization group* (DMRG) [8,9] is an optimization technique that uses the matrix product state (MPS) representation [10,11], see the review [12]. The MPS and DMRG are described in a problem-specific language, and despite becoming the methods of choice for many applications in the solid state physics and quantum chemistry, they were unknown in numerical analysis.

* Corresponding author at: University of Southampton, School of Chemistry, Highfield Campus, Southampton SO17 1BJ, United Kingdom. Tel.: +44 7551723849.
E-mail addresses: dolgov@mis.mpg.de (S.V. Dolgov), bokh@mis.mpg.de (B.N. Khoromskij), ivan.oseledets@gmail.com (I.V. Oseledets), dmitry.savostyanov@gmail.com (D.V. Savostyanov).

0010-4655/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cpc.2013.12.017

Looking for more efficient dimensionality reduction schemes, two groups in the numerical linear algebra community have independently re-discovered successful tensor formats under different names: the Tree-Tucker [13], tensor train (TT) [14], and hierarchical Tucker (HT) [15,16] formats. The equivalence of the TT and MPS format has been shortly discovered and reported in [17]. This connection is very beneficial and fruitful. The scheme of the DMRG algorithm has been applied to different problems in numerical analysis: approximate solution of linear systems [17,18], solution of eigenvalue problems [19], dynamics [20], cross interpolation [21]. Elements of the DMRG can be also found in recent algorithms for eigenproblems [22], solution of linear systems [23,24], multidimensional convolution [25–27], multidimensional Fourier transform [28], interpolation [29]. At the same time, new tensor formats have been proposed, e.g. the *quantized* tensor train (QTT) [30,31], and the QTT-Tucker [32]. Tensor product formats have been systematically applied to quantum chemistry computations [33,34,5,35–37].

Extending the work [19], in this paper we propose an algorithm to compute *several* approximate eigenpairs, e.g. *excited states*, which is a typical task in quantum physics and chemistry. Computation of several eigenvectors is equivalent to the minimization of the block Rayleigh quotient. It is natural to apply the DMRG framework, considering the index enumerating several eigenvectors as an additional dimension.

In the seminal papers by S. White [8,9], the *two-site* DMRG was already applied to the targeting of several excited states. This method reduces the large Hamiltonian to two neighboring sites, solves the two-site optimization problem, and then separates the state indices. The last step recovers the original MPS structure, and optionally adapts the TT ranks (bond dimensions) to a desired accuracy. In the two-site DMRG the index enumerating excited states enters as a *third* dimension into the reduced problem, which may lead to a larger computational cost.

The *numerical renormalization group* (NRG) [38], and its improved version *variational NRG* [39], also can be applied to target several eigenstates (the similarity of NRG and MPS was particularly emphasized in [40]). In these methods the enumerator constantly belongs to *the last site*. The block Rayleigh quotient formulated in terms of the initial vectors is then formally reduced to each *single site*. One-site problems are small, and the method benefits from faster calculation of each iteration. However the (v)NRG does not adapt the bond dimensions for targeted vectors, and simply returns them into the MPS via averaging. Therefore, the TT ranks should be properly guessed *a priori*, otherwise the convergence will not be satisfactory.

The method proposed in this paper combines the advances of DMRG and NRG: each local problem contains only one state index and the eigenpair index, which travels back and forth the tensor train during the computations. The Hamiltonian is reduced to a single site, but the optimized MPS block contains *two* dimensions—a state index and the eigenpair index. Separating the state variable and the enumerator in the same fashion as in the two-site DMRG, we adapt TT ranks, without considering the neighboring dimension. The gained adaptivity for the tensor structure empowers a fast convergence, and the overall complexity of the method is similar to the one-site DMRG. The one-site complexity is particularly important for solving high-dimensional problems with a large number of states in each site, like the high-dimensional PDEs.

The paper is organized as follows. In Section 2 definitions of the tensor train format are introduced. In Section 3 we present the algorithm, analyze its complexity, and compare it to the algorithms used in quantum physics. Sections 4–6 contain numerical experiments for the particle in a box, the Hénon–Heiles potential and the Heisenberg model, including the comparison of the computational speed with the publicly available DMRG implementations.

2. Notation, definitions and preliminaries

We consider the eigenproblem $AX = X\Lambda$, with the Hermitian matrix $A = A^*$. We are interested in B extreme eigenvalues λ_b and their eigenvectors x_b, for $b = 0, \dots, B-1$. This problem is equivalent to the minimization of the block Rayleigh quotient

$$\operatorname{trace}(X^*AX) \to \min, \quad \text{s.t. } X^*X = I, \tag{1}$$

where $X = [x_b]_{b=0}^{B-1}$ contains the orthogonal eigenvectors.

We assume that the problem has a tensor-product structure, i.e. all eigenvectors can be associated with d-dimensional tensors. Specifically, the elements of a vector $x = [x(i)]_{i=1}^{N}$ can be enumerated with d *mode indices* $i_1, \dots, i_d$ by a linear map $i = \overline{i_1 \dots i_d}$. The mode indices run through $i_k = 1, \dots, n_k$, where n_k are referred to as the *mode sizes* for $k = 1, \dots, d$. Naturally, $N = n_1 \dots n_d$, and if all mode sizes are of the same order $n_k \sim n$, the number of unknowns grows exponentially with the dimension, $N \sim n^d$. To make the problem tractable, we use the *tensor train* (TT) format [14], defined as follows,

$$\begin{aligned} x(i) &= x(\overline{i_1 \dots i_d}) = X^{(1)}(i_1) \dots X^{(d)}(i_d) \\ &= \sum_{\alpha_1 \dots \alpha_{d-1}} X^{(1)}_{\alpha_1}(i_1) \dots X^{(k-1)}_{\alpha_{k-2},\alpha_{k-1}}(i_{k-1}) X^{(k)}_{\alpha_{k-1},\alpha_k}(i_k) \\ &\quad \times X^{(k+1)}_{\alpha_k,\alpha_{k+1}}(i_{k+1}) \dots X^{(d)}_{\alpha_{d-1}}(i_d). \end{aligned} \tag{2}$$

Here and later we write equations in the *elementwise notation*, i.e. assume that they hold for all possible values of all free indices. The summation runs over all possible values of all *auxiliary* (or bond) indices $\alpha_k = 1, \dots, r_k$, where numbers $r_1, \dots, r_{d-1}$ are referred to as the *tensor train ranks* (TT-ranks), which are known as *bond dimensions* in the MPS/DMRG community. Each $X^{(k)}(i_k)$ is an $r_{k-1} \times r_k$ matrix, i.e. each entry of a vector $x = [x(i)]$ is represented by a product of d matrices in the right-hand side. The three-dimensional arrays $X^{(k)} = [X^{(k)}_{\alpha_{k-1},\alpha_k}(i_k)]$ of size $r_{k-1} \times n_k \times r_k$ are referred to as the *TT-cores*. The tensor train format is a *linear tensor network*, and can be illustrated as a graph, see Fig. 1.

In this paper we represent all computed eigenvectors *simultaneously* by the *block* tensor train format.

Definition 1 (*Block TT-format*)**.** The vectors $X = [x_b]_{b=0}^{B-1}$ are said to be in the block-TT format, if

$$\begin{aligned} x_b(i) &= x_b(\overline{i_1 \dots i_d}) \\ &= X^{(1)}(i_1) \dots X^{(p-1)}(i_{p-1}) \hat{X}^{(p)}(i_p, b) X^{(p+1)}(i_{p+1}) \dots X^{(d)}(i_d) \end{aligned} \tag{3}$$

for any $p = 1, \dots, d$.

The choice of the mode p which *carries* the enumerator b is not fixed—we will move it back and forth during the optimization. When the position p is chosen, it means that the matrix $\hat{X}^{(p)}(i_p, b)$ is additionally parametrized by the index b. The 'block' TT-core $\hat{X}^{(p)} = [\hat{X}^{(p)}_{\alpha_{p-1},\alpha_p}(i_p, b)]$ is now a tensor with four indices.

Following [41], we define the *interfaces* $X^{<k}$ of size $n_1 \dots n_{k-1} \times r_{k-1}$ and $X^{>k}$ of size $r_k \times n_{k+1} \dots n_d$ as follows

$$\begin{aligned} &X^{<k}(\overline{i_1 i_2 \dots i_{k-1}}, \beta_{k-1}) \\ &\quad = \sum_{\alpha_1 \dots \alpha_{k-2}} X^{(1)}_{\alpha_1}(i_1) X^{(2)}_{\alpha_1\alpha_2}(i_2) \dots X^{(k-1)}_{\alpha_{k-2},\beta_{k-1}}(i_{k-1}), \\ &X^{>k}(\beta_k, \overline{i_{k+1} \dots i_{d-1} i_d}) \\ &\quad = \sum_{\alpha_{k+1} \dots \alpha_{d-1}} X^{(k+1)}_{\beta_k,\alpha_{k+1}}(i_{k+1}) \dots X^{(d-1)}_{\alpha_{d-2},\alpha_{d-1}}(i_{d-1}) X^{(d)}_{\alpha_{d-1}}(i_d). \end{aligned} \tag{4}$$