Cost reduction of the bond-swapping part in an anisotropic tensor renormalization group

Hideaki Oba∗

Institute for Theoretical Physics, Kanazawa University, Kanazawa 920-1192, Japan
∗E-mail: h_oba@hep.s.kanazawa-u.ac.jp

Received August 30, 2019; Revised October 19, 2019; Accepted October 19, 2019; Published January 13, 2020

The bottleneck part of an anisotropic tensor renormalization group (ATRG) is a bond-swapping part that consists of a contraction of two tensors and a partial singular value decomposition of a matrix, and their computational costs are \( O(\chi^2 d + 1) \), where \( \chi \) is the maximum bond dimension and \( d \) is the dimensionality of the system. We propose an alternative method for the bond-swapping part and it scales with \( O(\chi_{\text{max}}(d + 3, 7)) \), though the total cost of ATRG with the method remains \( O(\chi^2 d + 1) \). Moreover, the memory cost of the whole algorithm can be reduced from \( O(\chi^2 d) \) to \( O(\chi_{\text{max}}(d + 1, 6)) \). We examine ATRG with or without the proposed method in the 4D Ising model and find that the free energy density of the proposed algorithm is consistent with that of the original ATRG while the elapsed time is significantly reduced. We also compare the proposed algorithm with a higher-order tensor renormalization group (HOTRG) and find that the value of the free energy density of the proposed algorithm is lower than that of HOTRG in the fixed elapsed time.

Subject Index

1. Introduction

Tensor network algorithms are theoretically and numerically useful tools for quantum and classical many-body systems [1,2]. For example, to study a 1D quantum many-body system, one can use a density matrix renormalization group [3–5], which is based on matrix product states [6,7] and can be understood as a variational method for a wave function. For higher-dimensional cases, there are algorithms based on projected entangled pair states [8–10]. In addition to those algorithms, the tensor renormalization group (TRG) [11] is an approach for coarse-graining tensor networks and calculates the partition function of a 2D classical model. For studying two- or higher-dimensional classical systems, we can use a higher-order tensor renormalization group (HOTRG) [12]. There are series of improved algorithms for coarse graining on the basis of various philosophies [13–25].

Since the tensor network algorithms avoid the sign problem, they have received attention not only from condensed matter physics but also from high energy physics [26–42]. We hope that the lattice models suffering from the sign problem in high energy physics (such as quantum chromodynamics with finite quark density, the \( \theta \) vacuum of quantum chromodynamics, chiral gauge theory, and the super-symmetric model) can be analyzed by the tensor network algorithms. However, if one computes the physical quantities of such high-dimensional systems with the algorithms, the computational costs and the memory costs get worse and it is difficult for the algorithms to provide results with sufficient precision.

A novel algorithm, the anisotropic tensor renormalization group (ATRG), has been proposed recently [25]. It is one of the tensor network algorithms for coarse-graining tensor networks and

© The Author(s) 2020. Published by Oxford University Press on behalf of the Physical Society of Japan.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.
can be applied to an arbitrary dimensional lattice model like HOTRG. Moreover, it has a much lower computational cost than HOTRG. The bottleneck part in ATRG is the bond-swapping part, which consists of a contraction of two tensors and a partial singular value decomposition (PSVD) of a matrix. The computational cost and the memory cost of the part are \( O(\chi^{2d+1}) \) and \( O(\chi^{2d}) \) respectively, where \( \chi \) is the maximum bond dimension and \( d \) is the dimensionality of the system. If the costs of the bond-swapping part in ATRG are reduced, we can expect to obtain more accurate results than those from the original ATRG with the same elapsed time and the same memory resource.

In this paper, we propose a bond-swapping method, the computational cost of which can be reduced from \( O(\chi^{2d+1}) \) to \( O(\chi^{\text{max}(d,3,7)}) \) and the memory cost of the whole algorithm can be reduced from \( O(\chi^{2d}) \) to \( O(\chi^{\text{max}(d+1,6)}) \). Furthermore, we numerically examine the accuracy of the free energy density of the 4D Ising model and the elapsed time of ATRG using the proposed method to compare with the original ATRG and HOTRG.

The rest of this paper is organized as follows. In Sect. 2, we review the original ATRG algorithm for a 4D hypercubic system. In Sect. 3, the proposed bond-swapping method is explained. In Sect. 4, we present numerical results of the 4D Ising model for a comparison between ATRG with the proposed method and the original ATRG or HOTRG and discuss their performances. A summary and outlook are given in Sect. 5. In Appendix A, we explain a method for making tensors, \( E \) and \( F \), in ATRG. In Appendix B, we study the parameters of the randomized singular value decomposition (RSVD).

### 2. Anisotropic tensor renormalization group

In this section, we explain the original ATRG [25] for a 4D hypercubic system. First of all, we rewrite the partition function of the system in terms of a tensor network as

\[
Z = \text{Tr} \prod_i T_i^{(\text{init})} \left( T_i^{(SVD)} \right)^\alpha, \tag{1}
\]

where \( T_i^{(\text{init})} \) is an initial tensor, \( i \) runs all lattice sites, and Tr means summing up all the bond indices of the initial tensors. If the direction of the coarse graining is set in the \( x \)-axis\(^1\), which is the vertical direction in Fig. 1, we redefine the initial tensor as \( T_i^{(\text{init})} \left( T_i^{(SVD)} \right)^\alpha \), where \( \Omega_0 = (y_0, z_0, w_0) \) and \( \Omega_1 = (y_1, z_1, w_1) \) are bundles of the bond indices\(^2\).

Then, we execute the coarse graining of the tensor network to reduce the degrees of freedom using ATRG [25]. The whole flow of ATRG is shown in Fig. 2 where there are seven steps, (a)–(g). First, in step (a), a tensor \( T \)\(^3\) is decomposed with the singular value decomposition (SVD),

\[
T_{\Omega_0\Omega_1x_0x_1} \simeq \sum_{\alpha=1}^{\chi} U_{\{T}\Omega_0x_0^\alpha S_{\{T}\alpha\alpha} V_{\{T}\Omega_1x_1^\alpha}, \tag{2}
\]

where \( \alpha \) is a new bond for the \( x \)-axis. Then, we make four tensors, \( A \), \( B \), \( C \), and \( D \), as

---

\(^1\) Another axis, such as \( y \), \( z \), or \( w \), can be treated in a similar way.

\(^2\) \( \Omega_0 \) and \( \Omega_1 \) consist of a combination of \( y_0, z_0, w_0 \) and that of \( y_1, z_1, w_1 \) respectively. For example, \( \Omega_0 = y_0 + (z_0 - 1)\chi^{\text{init}} + (w_0 - 1)(\chi^{\text{init}})^2 \) and \( \Omega_1 = y_1 + (z_1 - 1)\chi^{\text{init}} + (w_1 - 1)\chi^{\text{init}}(\chi^{\text{init}})^2 \) if all of the bond indices run from 1 to \( \chi^{\text{init}} \). Although the bond dimensions of each of the tensor legs are different in some cases, we can make the bundles similarly.

\(^3\) \( T \) is the initial tensor at the first coarse graining.
Fig. 1. The initial tensor $T_{\text{init}}$ in the initial tensor network of a 4D system. Some red bonds have numbers; two bonds are the same bond if they have the same number. (a) We focus on two tensors in the initial tensor network. (b) If the direction of the coarse graining is set in the $x$-axis, we redefine the initial tensor as $T_{\text{init}}^\Omega_{1}\Omega_{2}\Omega_{3}\Omega_{4}$, where $\Omega_{0} = (y_{0}, z_{0}, w_{0})$ and $\Omega_{1} = (y_{1}, z_{1}, w_{1})$ are bundles of the bond indices.

$$
A_{\Omega_{0}\Omega_{1}\alpha} = U\{T\}_{\Omega_{0}\Omega_{1}\alpha},
$$

(3)

$$
B_{\Omega_{1}\Omega_{2}\alpha} = S\{T\}_{\Omega_{1}\Omega_{2}\alpha} V\{T\}_{\Omega_{2}\Omega_{3}\alpha},
$$

(4)

$$
C_{\Omega_{0}\Omega_{1}\alpha} = S\{T\}_{\Omega_{0}\Omega_{1}\alpha} U\{T\}_{\Omega_{3}\Omega_{4}\alpha},
$$

(5)

$$
D_{\Omega_{1}\Omega_{2}\alpha} = V\{T\}_{\Omega_{2}\Omega_{3}\alpha}.
$$

(6)

In step (b), the two tensors, $B$ and $C$, are contracted and a tensor $M$ is obtained by

$$
M_{\Omega_{1}\Omega_{2}\alpha\beta} = \sum_{\Omega_{1}\alpha} B_{\Omega_{1}\Omega_{2}\alpha} C_{\Omega_{2}\Omega_{3}\beta}. 
$$

(7)

Then in step (c), we decompose $M$ into two tensors, $X$ and $Y$, to change the combinations of the bonds from $\Omega_{1}\alpha$ of $B$ and $\Omega_{2}\beta$ of $C$ to $\Omega_{2}\alpha$ of $X$ and $\Omega_{1}\beta$ of $Y$ as

$$
M_{\Omega_{1}\Omega_{2}\alpha\beta} \simeq \sum_{\Omega_{1}\alpha} U\{M\}_{\Omega_{1}\Omega_{2}\alpha} S\{M\}_{\Omega_{2}\Omega_{3}\alpha} V\{M\}_{\Omega_{1}\Omega_{2}\beta},
$$

(8)

$$
X_{\Omega_{2}\alpha\xi} = \sqrt{S}\{M\}_{\Omega_{2}\Omega_{3}\alpha} U\{M\}_{\Omega_{3}\Omega_{4}\alpha},
$$

(9)

$$
Y_{\Omega_{1}\beta\xi} = \sqrt{S}\{M\}_{\Omega_{1}\Omega_{2}\beta} V\{M\}_{\Omega_{2}\Omega_{3}\beta}.
$$

(10)

One may use the PSVD, such as the Arnoldi method [43] and the RSVD [44], for the decomposition of $M$ in Eq. (8). In step (d), two tensors, $E$ and $F$, are made from $A$, $X$, $Y$, and $D$ with the method explained in Appendix A. In step (e), we contract $A$, $X$, and $E$ to make $G$, and contract $D$, $Y$, and $F$
Fig. 2. The flow of the original ATRG. In step (a), a tensor $T$ is decomposed into two tensors such as $A$ and $B$ or $C$ and $D$ with the SVD. In step (b), the two tensors, $B$ and $C$, are contracted by summing up the bond $x_1$ and a tensor $M$ is obtained. In step (c), $M$ is decomposed by the PSVD into two tensors, $X$ and $Y$, to change the combinations of bonds from $\Omega_1\alpha$ of $B$ and $\Omega_2\beta$ of $C$ to $\Omega_1\beta$ of $Y$ and $\Omega_2\alpha$ of $X$. In step (d), two tensors, $E$ and $F$, are made from $A$, $X$, $Y$, and $D$ with the method explained in Appendix A. In step (e), we contract $A$, $X$, and $E$ by summing up the bonds $\alpha$, $\Omega_0$, $\Omega_1$, to make $G$, and contract $D$, $Y$, and $F$ by summing up the bonds $\beta$, $\Omega_1$, $\Omega_3$ to make $H$. In step (f), $G$ and $H$ are contracted by summing up the bond $x_1$ to obtain a coarse-grained tensor $T'$. In step (g), we reorder the bonds of $T'$ for the next coarse-graining direction, the $y$-axis.

To make $H$ as

$$G_{\Omega_0'x_0x_1} = \sum_{\alpha,\Omega_0,\Omega_2} A_{\Omega_0\alpha\Omega_2}\Omega_2 x_1 E_{\Omega_0\Omega_2\Omega_0'},$$

(11)

and

$$H_{\Omega_1'x_1x_2} = \sum_{\beta,\Omega_1,\Omega_3} D_{\Omega_1\beta\Omega_3}\Omega_3 x_1 F_{\Omega_1\Omega_3\Omega_1'},$$

(12)

where $\Omega_0' = (y_0', z_0', w_0')$ and $\Omega_1' = (y_1', z_1', w_1')$. Then, in step (f), $G$ and $H$ are contracted to obtain a coarse-grained tensor $T'$:

$$T'_{\Omega_0'\Omega_1'x_0x_1} = \sum_{x_1} G_{\Omega_0'x_0x_1} H_{\Omega_1'x_1x_2}.$$

(13)

Finally, in step (g), we reorder the bonds of $T'$ for the next coarse-graining direction, the $y$-axis, and redefine $T'$ as $T'_{\Psi_0'\Psi_1'\Psi_0'}$ in the same way as the initial tensor, where $\Psi_0' = (x_0, z_0', w_0')$ and $\Psi_1' = (x_2, z_1', w_1')$.

As explained in Ref. [25], steps (f), (g) and step (a) of the next coarse graining can be replaced by the following method, which has no truncation errors and reduces the computational cost. The flow...
Fig. 3. The flow of the improved method to make $A'$, $B'$, $C'$, and $D'$ from $G$ and $H$ [25]. Steps (a) and (b) represent steps (f), (g), and (a) in Fig. 2. In step (c), $G$ and $H$ are decomposed in the coarse-grained direction to obtain $K$ by the contraction in Eq. (16). In step (d), we execute the SVD of $K$. In step (e), $A'$, $B'$, $C'$, and $D'$ are made by the contraction in Eqs. (18)–(21). In step (f), we reorder the bonds of $A'$, $B'$, $C'$, and $D'$ for the next coarse graining.

of the improved method consists of four steps, which are shown in steps (c), (d), (e), and (f) of Fig. 3. In step (c) of Fig. 3, $G$ and $H$ are decomposed in the coarse-grained direction as

$$G_{\Omega_0'x_0x_1} = \sum_{\gamma} U_{G_{\Omega_0'x_0y}} S_{G_{y'y}} V_{G_{x_1y}}, \quad (14)$$

$$H_{\Omega_1'x_1x_2} = \sum_{\delta} U_{H_{\Omega_1'x_2\delta}} S_{H_{\delta\delta}} V_{H_{x_1\delta}}, \quad (15)$$

and $S_{G_{y'y}}$, $V_{G_{x_1y}}$, $V_{H_{x_1\delta}}$, and $S_{H_{\delta\delta}}$ are contracted to obtain $K$:

$$K_{\gamma\delta} = \sum_{x_1} S_{G_{y'y}} V_{G_{x_1y}} V_{H_{x_1\delta}} S_{H_{\delta\delta}}. \quad (16)$$

In step (d) of Fig. 3, we decompose $K$ with the SVD,

$$K_{\gamma\delta} = \sum_{\alpha} U_{K_{\gamma\alpha}} S_{K_{\alpha\alpha}} V_{K_{\delta\alpha}}, \quad (17)$$

and, in step (e) of Fig. 3, we contract the following tensors to make $A'$, $B'$, $C'$, and $D'$:

$$A'_{\Omega_0'x_0\alpha} = \sum_{\gamma} U_{K_{\gamma\alpha}} U_{G_{\Omega_0'y}} V_{G_{x_0y}}, \quad (18)$$

$$B'_{\Omega_1'x_2\alpha} = \sum_{\delta} S_{K_{\alpha\alpha}} V_{K_{x_2\delta}} U_{H_{\Omega_1'x_2\delta}}, \quad (19)$$
Fig. 4. The cost reduction of the bond-swapping part in ATRG. Steps (a) and (b) represent steps (b) and (c) of Fig. 2. In step (c), we perform the SVDs of $B$ and $C$ to make $B^s$ and $C^s$ by the contractions in Eqs. (24) and (25). Note that there is no truncation in this step and the red lines mean that the maximum bond dimensions of the indices, $\mu$ and $\nu$, are taken to be $\chi^2$. In step (d), $M^s$ is made of the contraction of $B^s$ and $C^s$ by summing up the bond $x_1$. In step (e), the combinations of the bonds are changed from $\mu \alpha$ of $B^s$ and $\beta \nu$ of $C^s$ to $\nu \alpha$ of $X^s$ and $\mu \beta$ of $Y^s$ using the PSVD. In step (f), we contract $X^s$ and $U_{\{C\}}$ by summing up the bonds $\nu$ to make $X$, and contract $Y^s$ and $U_{\{B\}}$ by summing up the bonds $\mu$ to make $Y$.

Finally, in step (f), we reorder the bonds of $A'$, $B'$, $C'$, and $D'$ for the next coarse graining in a similar way to step (g) in Fig. 2.

The total computational cost of the original ATRG is $O(\chi^{2d+1})$ and the total memory cost is $O(\chi^{2d})$. The former cost comes from step (e) in Fig. 2 and the bond-swapping part, which consists of steps (b) and (c) in Fig. 2. The latter cost arises from the bond-swapping part. Therefore, the bond-swapping part is the bottleneck of both the computational cost and the memory cost in ATRG.

3. Cost reduction of the bond-swapping part

In this section we propose a bond-swapping method instead of steps (b) and (c) in Fig. 2. The flow of the proposed method is shown in steps (c), (d), (e), and (f) of Fig. 4. The first step of the proposed bond-swapping method (step (c) of Fig. 4) is the SVDs of $B$ and $C$ as

$$B_{\Omega|\chi_1|\alpha} = \sum_{\mu} U_{\{B\}|\Omega_1|\mu} S_{\{B\}|\mu|\mu} V_{\{B\}|\chi_1|\mu},$$

$$C_{\Omega_2|\chi_1|\beta} = \sum_{\nu} U_{\{C\}|\Omega_2|\nu} S_{\{C\}|\nu|\nu} V_{\{C\}|\chi_1|\nu},$$

and defining $B^s$ and $C^s$ as

$$B^s_{\chi_1|\alpha|\mu} = S_{\{B\}|\mu|\mu} V_{\{B\}|\chi_1|\mu},$$

$$C^s_{\chi_1|\beta|\nu} = S_{\{C\}|\nu|\nu} V_{\{C\}|\chi_1|\nu}.$$
Note that there is no truncation in this step and the red lines in Fig. 4 mean that the maximum bond dimensions of the indices, \( \mu \) and \( \nu \), are taken to be \( \chi^2 \). In step (d) of Fig. 4, we perform the contraction of \( B^s \) and \( C^s \) to make \( M^s \):

\[
M_{\mu\nu\alpha\beta}^s = \sum_{x_1} B_{x_1;\alpha\mu}^s C_{x_1;\beta\nu}^s.
\]

Then, in step (e) of Fig. 4, we change the combinations of the bonds from \( \mu\alpha \) of \( B^s \) and \( \beta\nu \) of \( C^s \) to \( \nu\alpha \) of \( X^s \) and \( \mu\beta \) of \( Y^s \) as

\[
M_{\mu\nu\alpha\beta}^s \simeq \sum_{x_1} U_{\{M^s\}x_1;\alpha\mu} S_{\{M^s\}x_1;\beta\nu} V_{\{M^s\}x_1;\mu\beta},
\]

\[
X^s_{\nu\alpha x_1} = \sqrt{S_{\{M^s\}x_1;\alpha\mu} U_{\{M^s\}x_1;\beta\nu}},
\]

\[
Y^s_{\mu\beta x_1} = \sqrt{S_{\{M^s\}x_1;\alpha\mu} V_{\{M^s\}x_1;\mu\beta}},
\]

using the PSVD. Finally, in step (f) of Fig. 4, we make \( X \) and \( Y \) as

\[
X_{\Omega_2;\alpha x_1} = \sum_{\nu} X^s_{\nu\alpha x_1} U_{\{C\}x_1;\beta\nu},
\]

\[
Y_{\Omega_1;\beta x_1} = \sum_{\mu} Y^s_{\mu\beta x_1} U_{\{B\}x_1;\alpha\mu}.
\]

By applying the above method to ATRG, we can reduce the computational cost from \( O(\chi^{2d+1}) \) to \( O(\chi^{\max(d+3,7)}) \) in the swapping part and the memory cost of the whole algorithm from \( O(\chi^{2d}) \) to \( O(\chi^{\max(d+1,6)}) \). Note that there is no gain when we apply the proposed bond-swapping method to ATRG for 2- and 3D systems \((d = 2, 3)\).

### 4. Numerical results

In this section, we present the results of the 4D hypercubic Ising model with the periodic boundary condition to examine the effectiveness of the proposed bond-swapping method. We use the Ising model, which has 1024 spin sites, and set the temperature to 6.68, which is close to the critical point of the Monte Carlo result [45]. We use the RSVD as the PSVD part, and the parameters of the RSVD iterations and oversamples are taken to be \( \chi \) and 2\( \chi \) respectively. A detailed study of these parameters is given in Appendix B.

First, we show the comparison between ATRG using the proposed method and the original ATRG. The free energy densities obtained by the two algorithms are shown in Fig. 5. From this figure, we can see that the result of the proposed algorithm is consistent with that of the original ATRG. This consistency would be because the rank of \( M \) in the original ATRG is \( \chi^3 \) and coincides with the rank of \( M^s \) in the proposed algorithm. Next, Fig. 6 shows the total elapsed times of the two algorithms and the time of the proposed algorithm is significantly reduced from that of the original ATRG for larger \( \chi \). From Figs. 5 and 6, we conclude that the performance of ATRG can be improved without the loss of the accuracy of the free energy density. Figure 7 shows the details of the elapsed times.
Fig. 5. The free energy density of the 4D Ising model as a function of $\chi$ on a $1024^4$ lattice. The temperature is set to 6.68. The result of the proposed algorithm is consistent with that of the original ATRG.

Fig. 6. The total elapsed time as a function of $\chi$. The time of the proposed algorithm is significantly reduced from that of the original ATRG for larger $\chi$.

Fig. 7. Details of the total elapsed time at $\chi = 16$. The bottleneck part in the original ATRG is the bond-swapping part, while step (e) in Fig. 2 turns out to be the most time-consuming part in the proposed algorithm.
The free energy density of the 4D Ising model as a function of the elapsed time on a $1024^4$ lattice with the temperature set to 6.68. The result of the proposed algorithm can reach lower than that of HOTRG in the same elapsed time when the maximum bond dimensions of the two algorithms are increasing.

of the two algorithms at $\chi = 16$. We can see that the speeding up of ATRG is attained by a cost reduction of the bond-swapping part and step (e) in Fig. 2, which contains $O(\chi^9)$ contractions, turns out to be the most time-consuming part in the proposed algorithm.

Next, we compare the results of the proposed algorithm and HOTRG. The maximum bond dimensions in HOTRG are set to the integers in $3 \leq \chi \leq 7$. Figure 8 shows the relation between the free energy densities and the elapsed times of the two algorithms. The elapsed times are monotonically increasing when the bond dimensions are growing in this figure. We assume that the larger the maximum bond dimension $\chi$ is, the higher the accuracy of the free energy density becomes. Since the free energy densities of the proposed algorithm and HOTRG are monotonically decreasing as the maximum bond dimensions are increasing in Fig. 8, we can consider that the lower the free energy density is, the higher the accuracy of that becomes. From this viewpoint, the proposed algorithm can obtain a more accurate free energy density than HOTRG in the same elapsed time when the maximum bond dimensions of the two algorithms are increasing.

We use a machine that has 132 GB for the memory and Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60 GHz for the CPU. The programs for the algorithms are written with Python 3.7.1; we use numpy.tensordot in NumPy 1.15.4 for the tensor contractions, scipy.linalg.svd in SciPy 1.1.0 for the SVD of the tensors, and sklearn.utils.extmath.randomized_svd in scikit-learn 0.20.1 for the RSVD of the tensors.

5. Summary

We propose a bond-swapping method, the computational cost of which can be reduced from $O(\chi^{2d+1})$ to $O(\chi^{\max(d+3,7)})$. Moreover, the memory cost of the whole algorithm can be reduced from $O(\chi^{2d})$ to $O(\chi^{\max(d+1,6)})$ with the method. We examine ATRG using the method with the 4D hypercubic Ising model and find that the free energy density obtained by the algorithm is consistent with that obtained by the original ATRG while the elapsed time is significantly reduced. Moreover, we compare the proposed algorithm with HOTRG in terms of the free energy density of the Ising model and the elapsed time. Then, we find that the proposed algorithm can obtain a more accurate free energy density than HOTRG in the same elapsed time for large maximum bond dimensions.
For future work, step (e) in Fig. 2, which has the $O(\chi^{2d+1})$ contractions, is the bottleneck of the proposed algorithm. To speed up this part, we will apply parallel computing to the bottleneck contractions. Furthermore, we will analyze the physical quantities of the 4D Ising model with large maximum bond dimensions to compare with other results [45,46].

Acknowledgements

The author would like to thank Shinji Takeda for insightful feedback.

Appendix A. Making $E$ and $F$

In this appendix, we explain a method for making the tensors, $E$ and $F$, in step (d) of Fig. 2. Figure A.1 shows the flow of the method to make the tensors for the $y$-axis. Before starting the method for the $y$-axis, we reorder the bonds of $A$ and $X$ as

$$A_{y_0y_0} = A_{\Omega_{y0y0}}, \quad (A.1)$$
$$X_{y_2y_2} = X_{\Omega_{y2y2}}, \quad (A.2)$$

where the superscript $(y)$ represents the $y$-axis direction and $\Psi_0 = (x_0, z_0, w_0)$ and $\Psi_2 = (x_2, z_2, w_2)$. In step (a) of Fig. A.1, the tensors and their conjugate tensors are contracted by summing up the $\Psi$ as

$$L_{y_0y_2} = \sum_{\Psi_0} A_{y_0y_0} (A_{y_0y_0})^*, \quad (A.3)$$
$$W_{y_2y_2} = \sum_{\Psi_2} X_{y_2y_2} (X_{y_2y_2})^*. \quad (A.4)$$

In step (b) of Fig. A.1, we contract $L$ and $W$ by summing up the bonds $\alpha$ and $\tilde{\alpha}$:

$$P_{y_0y_2} = \sum_{\alpha, \tilde{\alpha}} L_{y_0y_2} W_{\alpha, \tilde{\alpha}}. \quad (A.5)$$

In step (c) of Fig. A.1, we perform the SVD or the eigenvalue decomposition of $P$ and the truncation to retain $\chi$ largest singular values,

$$P_{y_0y_2} \simeq \sum_{\eta=1}^{\chi} U_{y_0y_2} \hat{\eta}^* S_{y_0y_2} \eta^* U^*_{y_0y_2} \eta. \quad (A.6)$$

where $U_{y_0y_2}$ is an isometry made by the left singular vectors that corresponds to the $\chi$ largest singular values. We can also obtain $Q$ and $U_{y_0y_2}$ from $Y$ and $D$ in a similar way to $P$ and $U_{y_0y_2}$. Then, in step (d) of Fig. A.1, the isometries, $U_{y_0y_2}$, $U_{y_0y_2}$, $U_{y_0y_2}$, and $U_{y_0y_2}$, are inserted between $A^{(y)}$ and $X^{(y)}$. To make $E^{(y)}$ and $F^{(y)}$ explicitly, $U_{y_0y_2}$ and $U_{y_0y_2}$ are contracted as

$$N_{y_{\tilde{y}1}} = \sum_{y_0y_1} U_{y_0y_1} U_{y_0y_{\tilde{y}1}}. \quad (A.7)$$

in step (e) of Fig. A.1. Next, in step (f) of Fig. A.1, we carry out the SVD of $N$,

$$N_{y_\beta} = \sum_{y_{\alpha}} U_{y_\beta} \tilde{S}_{y_\alpha} V_{y_{\alpha}y_\beta}. \quad (A.8)$$
Fig. A.1. The flow of making $E$ and $F$ for the $y$-axis in step (d) of Fig. 2. In step (a), we contract a tensor, $A^y(X^y)$, and its conjugate to make a tensor, $L(W)$. In step (b), $L$ and $W$ are contracted by summing up the bonds $\alpha$ and $\tilde{\alpha}$. In step (c), we perform the SVD or the eigenvalue decomposition of $P$ and the truncation to retain $\chi$ largest singular values. Steps (a), (b), and (c) show an example of obtaining $U\{P\}$ and we can also obtain $Q$ and $U\{Q\}$ from $Y$ and $D$ similarly. In step (d), the isometries, $U^\dagger\{P\}$, $U\{P\}$, $U\{Q\}$, and $U^\dagger\{Q\}$, are inserted between $A^yX^y$ and $Y$ and $D$ similarly. In step (e), $U\{P\}$ and $U\{Q\}$ are contracted by summing up the bonds $\tilde{y}_0$ and $\tilde{y}_1$ to make $N$. In step (f), we decompose $N$ with the SVD and make $E^y$ and $F^y$ by the contractions in Eqs. (A.9) and (A.10) respectively.

We can use the same way for another direction, and $E$ and $F$ consist of tensors of directions other than the coarse-graining one, such as

\begin{align}
E_{y02\kappa}^{(y)} &= \sum_\eta \sqrt{S_{[N]\kappa\kappa}} \ U_{[N]\eta\kappa}^* \ U_{[P]\eta y_0 y_2}^, \quad (A.9) \\
F_{y13\kappa}^{(y)} &= \sum_\theta \sqrt{S_{[N]\kappa\kappa}} \ V_{[N]\theta\kappa}^* \ U_{[Q]\theta y_1 y_3}^, \quad (A.10)
\end{align}

and the contractions for making $E^y$ and $F^y$ as
Fig. B.1. The relative errors of the free energy densities of the Ising model on a 1024^4 lattice with the original ATRG and the proposed algorithm, where $\chi = 10, 14$, the temperatures are set to 6.65 and 6.68 ($t = 6.65, 6.68$), and then the number of the RSVD iterations is taken to be $\chi$. The oversampling parameter $n$ is changed from 1 to $2\chi$. We find that the results of the original ATRG and the proposed algorithm show similar behavior at $\chi = 10$; this is large enough to take $n \geq 20$ at $\chi = 10, 14$.

\[ E = E^{(y)} \otimes E^{(z)} \otimes E^{(w)} \]  
\[ F = F^{(y)} \otimes F^{(z)} \otimes F^{(w)}. \] (A.11)  
(A.12)

To complete this step for making $E$ and $F$, $4(d-1)$ times $O(\chi^{d+3})$ contractions, $2(d-1)$ times $O(\chi^6)$ contractions, and $2(d-1)$ times $O(\chi^6)$ SVDs are needed. Then, the bottleneck of this part is the $O(\chi^{d+3})$ contraction.

Appendix B. Study of RSVD parameters

In this appendix, we study the parameters of the RSVD in the proposed algorithm and the original ATRG. The purpose of this study is to see how large the parameters of each algorithm should be taken to converge to the free energy density obtained by an ATRG algorithm that uses the truncation of the full SVD instead of the RSVD.

To investigate the convergence for the original ATRG, we use the Ising model on a 1024^4 lattice and define the relative error of the free energy density as

\[ \delta_f = \left| \frac{f_{n,RSVD} - f_{SVD}}{f_{SVD}} \right|, \] (B.1)

where $f_{SVD}$ is the free energy density obtained by ATRG, which applies the truncation of the full SVD to $M$, and $f_{n,RSVD}$ is the one using the original ATRG, which applies the RSVD of $n$ oversamples and $\chi$ iterations to the matrix. In the case of the proposed algorithm, we can also define $\delta_f$ similarly. Note that if the number of the RSVD iterations is fixed to $\chi$, the computational cost of the bond-swapping part in the original ATRG becomes $O(\chi^{10})$, and then the leading cost of the algorithm gets worse from $O(\chi^9)$. On the other hand, the cost of the part in the proposed algorithm becomes $O(\chi^8)$ and it is not the leading one in the whole algorithm (see Fig. 7). Figure B.1 shows $\delta_f$ as the function of $n$ with $\chi = 10, 14$ and the temperatures are taken to be 6.65 and 6.68 ($t = 6.65, 6.68$). From Fig. B.1, we can see that the results of the original ATRG and the proposed algorithm show similar behavior at $\chi = 10$; this is large enough to take $n \geq 20$ at $\chi = 10, 14$.

The cost of these SVDs can be reduced to $O(\chi^5)$ if one uses the PSVDs.
behavior at $\chi = 10$; this is large enough to take $n \geq 20$ at $\chi = 10$, 14. Therefore, we use $\chi$ and $2\chi$ for the numbers of the RSVD iterations and oversamples respectively for the numerical results shown in Sect. 4. We believe that this is large enough for the RSVD to converge to the full SVD results when $\chi \geq 10$. Finally we remark that when degenerate singular values are truncated, the results of the RSVD may not converge to those of the full SVD even if the parameters are taken to be sufficiently large.

References

[1] J. I. Cirac and F. Verstraete, J. Phys. A: Math. Theor. 42, 504004 (2009).
[2] R. Orús, Ann. Phys. 349, 117 (2014).
[3] S. R. White, Phys. Rev. Lett. 69, 2863 (1992).
[4] S. R. White, Phys. Rev. B 48, 10345 (1993).
[5] U. Schollwöck, Rev. Mod. Phys. 77, 259 (2005).
[6] M. Fannes, B. Nachtergaele, and R. F. Werner, Commun. Math. Phys. 144, 443 (1992).
[7] S. Östlund and S. Rommer, Phys. Rev. Lett. 75, 3537 (1995).
[8] F. Verstraete and J. I. Cirac, arXiv:cond-mat/0407066 [cond-mat.str-el] [Search INSPIRE].
[9] F. Verstraete, V. Murg, and J. I. Cirac, Adv. Phys. 57, 143 (2008).
[10] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, Phys. Rev. Lett. 101, 250602 (2008).
[11] M. Levin and C. P. Nave, Phys. Rev. Lett. 99, 120601 (2007).
[12] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, Phys. Rev. B 86, 045139 (2012).
[13] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang, Phys. Rev. Lett. 103, 160601 (2009).
[14] Z.-C. Gu and X.-G. Wen, Phys. Rev. B 80, 155131 (2009).
[15] G. Evenbly and G. Vidal, Phys. Rev. Lett. 115, 180405 (2015).
[16] G. Evenbly, Phys. Rev. B 95, 045117 (2017).
[17] S. Yang, Z.-C. Gu, and X.-G. Wen, Phys. Rev. Lett. 118, 110504 (2017).
[18] M. Bal, M. Mariën, J. Haegeman, and F. Verstraete, Phys. Rev. Lett. 118, 250602 (2017).
[19] S. Morita, R. Igarashi, H.-H. Zhao, and N. Kawashima, Phys. Rev. E 97, 033310 (2018).
[20] M. Hauru, C. Delcamp, and S. Mizera, Phys. Rev. B 97, 045111 (2018).
[21] K. Harada, Phys. Rev. B 97, 045124 (2018).
[22] G. Evenbly, Phys. Rev. B 98, 085155 (2018).
[23] Y. Nakamura, H. Oba, and S. Takeda, Phys. Rev. B 99, 155101 (2019).
[24] S. Iino, S. Morita, and N. Kawashima, Phys. Rev. B 100, 035449 (2019).
[25] D. Adachi, T. Okubo, and S. Todo, arXiv:1906.02007 [cond-mat.stat-mech] [Search INSPIRE].
[26] M. C. Bañuls, K. Cichy, J. I. Cirac, and K. Jansen, J. High Energy Phys. 1311, 158 (2013).
[27] S. Kühn, E. Zohar, J. I. Cirac, and M. C. Bañuls, J. High Energy Phys. 1507, 130 (2015).
[28] M. C. Bañuls, K. Cichy, J. I. Cirac, K. Jansen, and H. Saito, Phys. Rev. D 92, 034519 (2015).
[29] M. C. Bañuls, K. Cichy, K. Jansen, and H. Saito, Phys. Rev. D 93, 094512 (2016).
[30] M. C. Bañuls, K. Cichy, J. I. Cirac, K. Jansen, and S. Kühn, Phys. Rev. Lett. 118, 071601 (2017).
[31] M. C. Bañuls, K. Cichy, J. I. Cirac, K. Jansen, and S. Kühn, Phys. Rev. X 7, 041046 (2017).
[32] Y. Shimizu, Mod. Phys. Lett. A 27, 1250035 (2012).
[33] Y. Shimizu and Y. Kuramashi, Phys. Rev. D 90, 014508 (2014).
[34] Y. Shimizu and Y. Kuramashi, Phys. Rev. D 90, 074503 (2014).
[35] Y. Shimizu and Y. Kuramashi, Phys. Rev. D 97, 034502 (2018).
[36] A. Denbleyker, Y. Liu, Y. Meurice, M. P. Qin, T. Xiang, Z. Y. Xie, J. F. Yu, and H. Zou, Phys. Rev. D 89, 016008 (2014).
[37] J. F. Yu, Z. Y. Xie, Y. Meurice, Y. Liu, A. Denbleyker, H. Zou, M. P. Qin, J. Chen, and T. Xiang, Phys. Rev. E 89, 013308 (2014).
[38] Y. Meurice, A. Bazavov, S. W. Tsai, J. F. Unmuth-Yockey, L. P. Yang, and J. Zhang, PoS LATTICE 2016, 325 (2016).
[39] S. Takeda and Y. Yoshimura, Prog. Theor. Exp. Phys. 2015, 043B01 (2015).
[40] H. Kawauchi and S. Takeda, Phys. Rev. D 93, 114503 (2016).
[41] R. Sakai, S. Takeda, and Y. Yoshimura, Prog. Theor. Exp. Phys. 2017, 063B07 (2017).
[42] D. Kadoh, Y. Kuramashi, Y. Nakamura, R. Sakai, S. Takeda, and Y. Yoshimura, J. High Energy Phys. 1803, 141 (2018).
[43] W. E. Arnoldi, Q. Appl. Math. 9, 17 (1951).
[44] N. Halko, P. G. Martinsson, and J. A. Tropp, SIAM Rev. 53, 217 (2011).
[45] P. H. Lundow and K. Markström, Phys. Rev. E 80, 031104 (2009).
[46] S. Akiyama, Y. Kuramashi, T. Yamashita, and Y. Yoshimura, Phys. Rev. D 100, 054510 (2019).