

Fermionic Isometric Tensor Network States in Two Dimensions

Zhehao Dai,^{1,*} Yantao Wu,^{2,1,*} Taige Wang,^{1,3} and Michael P. Zaletel^{1,3}

¹*Department of Physics, University of California, Berkeley, CA 94720, USA*

²*RIKEN iTHEMS, Wako, Saitama 351-0198, Japan*

³*Material Science Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*

(Dated: November 2, 2022)

We generalize isometric tensor network states to fermionic systems, paving the way for efficient adaptations of 1D tensor network algorithms to 2D fermionic systems. As the first application of this formalism, we developed and benchmarked a time-evolution block-decimation (TEBD) algorithm for real-time and imaginary-time evolution. The imaginary-time evolution produces ground-state energies for gapped systems, systems with a Dirac point, and systems with gapless edge mode to good accuracy. The real-time TEBD captures the chiral edge dynamics on the boundary of a Chern insulator.

Simulating quantum many-body systems efficiently is one of the most challenging goals in computational physics. For one-dimensional (1D) systems, density-matrix renormalization group (DMRG) [1, 2] gives practically exact ground states for all gapped systems and good approximations for ground states of gapless systems. The output of DMRG is a wavefunction written as a multiplication of matrices, hence the name matrix product states (MPS) [3, 4]. Two-dimensional (2D) tensor network states (TNS) are generalizations of MPS; they provide efficient representations for a large class of (if not all) ground states of local Hamiltonians [5, 6]. However, unlike in 1D, the complexity of an *exact* computation of any physical expectation value for a given 2D TNS scales exponentially with the system size. It has been a central goal to develop approximate tensor contraction methods and to generalize 1D MPS algorithms into 2D. Recently, Zaletel and Pollman proposed a more restricted ansatz, the isometric TNS (isoTNS) [7], which allows for efficient contractions in 2D. The key ingredient is the enforcement of the isometric condition in 2D, generalizing the canonical form of MPS. This ansatz opens up many possibilities, such as a DMRG algorithm in 2D [8].

In this work, we generalize the isoTNS ansatz to fermionic systems and find that the isometric structure is compatible with the fermionic structure. Broadly speaking, there are two equivalent [9] formulations of 2D fermionic tensor network states (fTNS): the virtual fermion ansatz [10–12] and the swap gate ansatz [13, 14]. To account for fermion statistics, the former attaches virtual fermion operators to each tensor in the network. The latter inserts additional fermion swap gates into the tensor network. They are both introduced to overcome the following non-locality in fermionic systems: If we naively choose an ordering of fermions c_1, \dots, c_n in 2D and write the wavefunction under the basis $c_1^{\dagger k_1} \dots c_n^{\dagger k_n} |vac\rangle$, $k_i = 0, 1$ using conventional tensor networks, the action of a *local* operator $c_i^{\dagger} c_j$ may leave a long string $\prod_{i < k < j} (-1)^k$, changing the TNS nonlocally. In both fTNS formulations, local operators change the tensors locally, which allows for a generalization of tensor network algorithms to

fermionic systems at the same leading complexity. We define *fermionic isometric tensor network states* (fisoTNS) using virtual fermions and provide alternative explanations using swap gates. We demonstrate a time-evolution block-decimation (TEBD) algorithm within the fisoTNS formalism. As an example, we solve the ground states, via imaginary-time evolution, of a trivial insulator, non-interacting and interacting Chern insulators, $p+ip$ superconductors (SC), and Dirac fermions. We also simulate the chiral dynamics at the edge of Chern insulators via real-time evolution, demonstrating that fisoTNS captures the nontrivial topological order in a finite system.

Isometric condition We first review the isometric condition for bosonic TNS. In 1D, the isometric MPS wavefunction is defined as $\psi^{k_1, \dots, k_N} = \sum_{\{i_n\}} A[1]_{i_1}^{k_1} A[2]_{i_1, i_2}^{k_2} \dots \Lambda[n_c]_{i_r}^{k_{n_c}} \dots B[N]_{i_{N-1}}^{k_N}$, where n_c is a site which we choose to be the orthogonality center (Fig. 1(a)). The isometric condition for tensors left to n_c is $\sum_{l,k} A_{lr}^{k*} A_{lr}^k = \delta_{rr'}$. For tensors right to n_c , $\sum_{r,k} B_{lr}^{k*} B_{lr}^k = \delta_{ll'}$ (Fig. 1(b)). Under this condition, the expectation value of any physical operator $O_{k,k'}$ at the center is just $\sum_{l,r,k,k'} \Lambda_{lr}^{k*} O_{k,k'} \Lambda_{lr}^{k'}$ (Fig. 1(c)). In 1D, any MPS can be put into the isometric form efficiently without increasing the bond dimension [1].

2D TNS is given by the contraction of tensors with 4 virtual legs and a physical leg (Fig 1(d)). The generalized *isometric* constraint is the following [7]. First, we assign arrows to virtual legs such that they all point toward a chosen site, which we call the *orthogonality center* (red dot in Fig 1(d)). The arrows on the physical legs always point towards the tensors. Then we require each tensor, viewed as a map from the outgoing legs to the incoming legs, to be an isometry. For example, the requirement for tensors in the lower-left quadrant is $\sum_{l,d,k} A_{lrdu}^{k*} A_{lr'du'}^k = \delta_{rr'} \delta_{uu'}$ (Fig 1(e)). To compute the expectation value at the center, we only need this single tensor (Fig 1(f)). The 1D hypersurface which all arrows in the network point towards, e.g. the red cross in Fig. 1 (d), is also special because the computation of correlation functions on it reduces to a 1D tensor contraction

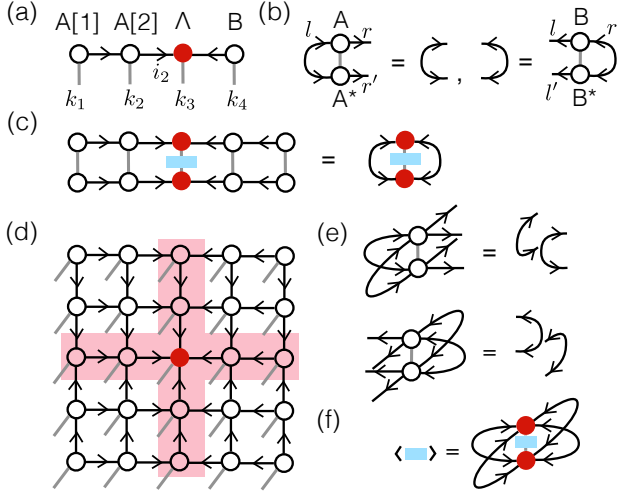


FIG. 1. Review of isometric tensor network states in 1D and 2D (a) A matrix product state in isometric form. The red dot represents the orthogonality center. (b) Isometric condition for tensors on the left (right) of the center A (B). A line without a circle on it represents identity. (c) The expectation of any physical operator (blue square) on the orthogonality center equals the contraction of the operator with the central tensor and its conjugate. (d) 2D isometric tensor network. The red dot represents the orthogonality center. On the shaded region, the orthogonality column, all arrows point in. (e) Isometric condition for tensors in the lower-left and the upper-right quadrant. (f) The same as (c) but in 2D.

on the hypersurface [7]. We will call this hypersurface the *orthogonality column* (OC).

Fermionic tensor network states We now introduce an fTNS ansatz that explicitly attaches auxiliary fermions to virtual bonds (a variation of the ansatz in Ref.[10]).

$$|\psi\rangle = \langle \prod_{(x,y)} \hat{A}_{(x,y)} \prod_{(x,y)} \hat{V}_{(x,y)} \rangle_{\text{aux}} |vac\rangle, \quad (1)$$

$$\hat{A}_{(x,y)} = A_{lrdu}^k c_{(x,y)}^{\dagger k} \delta_{(x,y)}^D \beta_{(x,y)}^R \gamma_{(x,y)}^U \alpha_{(x,y)}^L, \quad (2)$$

$$\hat{V}_{(x,y)} = (1 + \beta_{(x,y)}^{\dagger} \alpha_{(x+1,y)}^{\dagger})(1 + \delta_{(x,y+1)}^{\dagger} \gamma_{(x,y)}^{\dagger}), \quad (3)$$

where we assume a physical Hilbert space of one complex fermion per site, $c_{(x,y)}$, and we omit indices of $\hat{A}_{(x,y)}$ for simplicity. Each tensor A_{lrdu}^k , $k \in \{0,1\}$, $l, r, d, u \in \{0, \dots, \chi - 1\}$, is required to be \mathbb{Z}_2 -graded, where the \mathbb{Z}_2 symmetry corresponds to the fermion parity conservation. Namely, for each virtual index l, r, d, u , we assign a parity label $L(l), R(r), D(d), U(u) = 0, 1$; and we require $A_{lrdu}^k = 0$ whenever $k + L + R + D + U$ is odd. On each site, we assign 4 auxiliary complex fermions $\alpha, \beta, \gamma, \delta$, on the left, right, up, and down virtual bonds respectively (Fig. 2(a)). The tensors are promoted to fermionic operators according to the rule that whenever the corresponding index has a parity label 1, an auxiliary (physical) fermion annihilation (creation) operator is attached (Eq. 2). Auxiliary fermion operators on each site

are ordered *counterclockwise*. As for usual TNS, whenever two sites are connected, the corresponding two virtual indices are contracted. Furthermore, the two auxiliary fermion annihilation operators attached to the bond are contracted with the bond operator $(1 + \beta_{(x,y)}^{\dagger} \alpha_{(x+1,y)}^{\dagger})$ for the horizontal and $(1 + \delta_{(x,y+1)}^{\dagger} \gamma_{(x,y)}^{\dagger})$ for the vertical bond. The action of the bond-operators and the site-operators on the vacuum of the physical fermions, followed by the contraction of all virtual fermion operators in their vacuum gives the desired physical state. For example, the fTNS with bond dimension one, where $\hat{A}_{(x,y)} = c_{(x,y)}^{\dagger} \beta_{(x,y)}$ for even x and $\hat{A}_{(x,y)} = c_{(x,y)}^{\dagger} \alpha_{(x,y)}$ for odd x , gives the fully filled state of one fermion per site. We may allow more than one pair of entangled fermions for each virtual bond in the definition of fTNS, but the resulting ansatz can always be rewritten in the form of Eq. 1 [15].

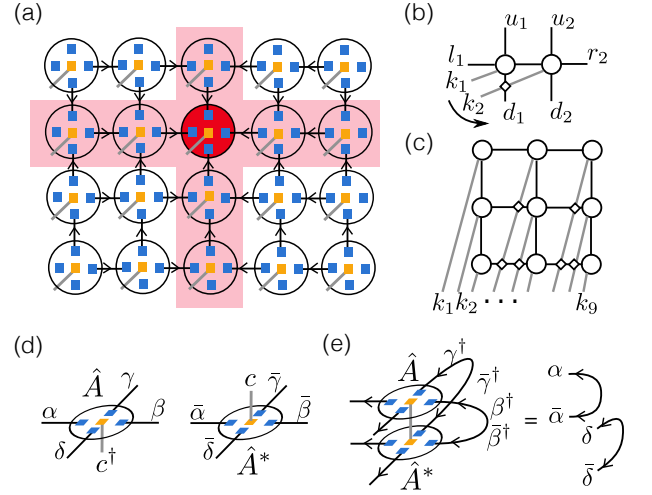


FIG. 2. (a) Fermionic isometric tensor network states in the virtual-fermion formalism. The arrows point to the orthogonality center as in the case of bosonic isoTNS. The 4 blue squares on each site represent 4 virtual complex fermions. Orange squares represent physical fermions. Each tensor is promoted into an operator by attaching the physical creation operator and the virtual fermion annihilation operator. The tensors act on the virtual-fermion state of one bell pair per bond to create the physical state. (b) To contract two fermionic tensors and order the fermion operators counterclockwise starting from k_1 , we need to do the tensor contraction with an additional swap gate $(-1)^{k_2 D_1}$ (the small square). (c) In the swap-gate convention, the wavefunction is given by the contraction of all tensors and swap gates at the crossings. (d) Fermion-attachment of the tensor A and A^* . (e) Isometric condition for fisoTNS in the upper-right quadrant.

This ansatz avoids the problem of the nonlocal string discussed before. Since each site-operator $\hat{A}_{(x,y)}$ is parity-even, the fermion annihilation (creation) operator $c_{(x,y)}$ ($c_{(x,y)}^{\dagger}$) *commutes* with all site-operators on sites other than (x,y) . Thus the action of nearest-neighbor

hopping affects only the two tensors on the corresponding sites.

The ansatz in Eq. 1 may seem complicated, but it does not increase the complexity of any tensor network algorithm. The \mathbb{Z}_2 symmetric tensors are routinely used for \mathbb{Z}_2 symmetric boson models; the only thing the auxiliary fermions do is to give extra minus signs when contracting tensors. For example, if we want to replace the two tensors in Fig 2(b) by a single tensor $\hat{\Theta} = \Theta_{l_1 r_2 d_1 d_2 u_1 u_2}^{k_1 k_2} c_1^{\dagger k_1} c_2^{\dagger k_2} \delta_1^{D_1} \delta_2^{D_2} \beta_2^{R_2} \gamma_2^{U_2} \gamma_1^{U_1} \alpha_1^{L_1}$ without changing the many-body state, we need $\hat{\Theta} = \sum_{r_1=1,2} \langle \hat{A}[1] \hat{A}[2] (1 + \beta_1^\dagger \alpha_2^\dagger) \rangle_{\beta_1, \alpha_2}$, where we contract the auxiliary fermion β_1 and α_2 in their vacuum. By direct calculation, (indices omitted unless necessary)

$$\Theta = A[1]A[2](-1)^{k_2 D_1}. \quad (4)$$

In fact, extra factors like $(-1)^{k_2 D_1}$ can be deduced pictorially without auxiliary fermions (Fig. 2(b)): The rule is whenever two fermion legs with parity labels I and J cross each other, an extra *swap gate*, $(-1)^{IJ}$ is applied [14]. Using swap gates, the fermion wavefunction can be written in the Fock space basis as the regular contraction of the tensors $A_{(x,y)}$ with additional swap gates applied at the crossings as we extend the physical legs outside (Fig. 2(c)). In extending the physical legs, we can freely deform them [14] without changing the wavefunction as long as the order of endpoints is fixed. The latter corresponds to the order of fermions of the Fock space basis. The general equivalence between the swap-gate method and auxiliary-fermion ansatz is discussed in [9]. A self-contained proof for our proposal is given in supplemental materials (SM) [15]. In practice, we only need to keep track of fermion signs that arise from local tensor operations instead of actually computing the wavefunction.

Isometric condition for fermions As in bosonic isoTNS, the isometric condition for fermions should make trivial the contraction between a state and its conjugate. One may guess that this is just the isometric condition at the tensor level. This is almost correct. In fact, the isometric condition we define below is equivalent to that \hat{A} is an isometry in the upper-left, lower-left, and lower-right quadrants, but $(-1)^{k_D} \hat{A}$ be an isometry in the upper-right quadrant. The asymmetry among the 4 quadrants is related to how we order fermion operators in Eq. 2.

More precisely, we *define* a set of conjugate tensor operators $\hat{A}_{(x,y)}^* = A_{l' r' d' u'}^{*k} \bar{\alpha}_{(x,y)}^{L'} \bar{\gamma}_{(x,y)}^{U'} \bar{\beta}_{(x,y)}^{R'} \bar{\delta}_{(x,y)}^{D'} c^k$, where $\bar{\alpha}_{(x,y)}, \bar{\beta}_{(x,y)}, \bar{\gamma}_{(x,y)}, \bar{\delta}_{(x,y)}$ form a set of auxiliary fermions independent from $\alpha_{(x,y)}, \dots, \delta_{(x,y)}$ (Fig. 2(d)). We define the isometric condition to be that the contraction of \hat{A}^* , \hat{A} , and the creation operators on the incoming legs give the identity tensor with auxiliary fermions attached to it. For example, in the upper-right quadrant (Fig. 2(e)),

the isometric condition is

$$\langle \hat{A}^* \hat{A} \delta_{uu'} \gamma^{\dagger U} \bar{\gamma}^{\dagger U'} \delta_{rr'} \beta^{\dagger R} \bar{\beta}^{\dagger R'} \rangle = \delta_{dd'} \bar{\delta}^{D'} \delta^D \delta_{ll'} \bar{\alpha}^{L'} \alpha^L,$$

where $\langle \cdot \rangle$ denotes contraction over $\gamma, \bar{\gamma}, \beta, \bar{\beta}$, and c . See an alternative explanation based on swap gates in SM [15]. We now state a key theorem of fisoTNS, relating the inner product of many-body states to that of tensors, which implies physical expectation values can be computed from the tensor at the center.

Theorem 1. For any two fisoTNS $|\psi\rangle$ and $|\psi'\rangle$ that have the same tensors $\hat{A}_{(x,y)}$ everywhere except at the orthogonality center, with $\hat{\Lambda}_{(x_0, y_0)}$ and $\hat{\Lambda}'_{(x_0, y_0)}$ for each respectively, $\langle \psi | \psi' \rangle = \sum_{l,r,d,u,k} \Lambda_{l,r,d,u}^{*k} \Lambda'_{l,r,d,u}^k$.

This theorem can be proved by contracting the double-layer tensors representing $\langle \psi | \psi' \rangle$ iteratively from the 4 corners to the orthogonality center. See SM [15] for details. For the computation of correlation functions on the OC, the same argument shows that tensors outside can be trivially contracted and the computation reduces to a contraction along the OC, as in the bosonic case.

Representability In terms of the ability of isoTNS to represent ground states of local Hamiltonians, as a first step, we prove that if a state admits a fisoTNS representation in the thermodynamic limit, after the action of a finite-depth local-unitary circuit it still admits a fisoTNS representation. The proof directly constructs such a representation by contracting neighboring fermionic tensors, applying gates, and splitting the tensors back to isometric tensors on each site. For details, see SM [15]. As a corollary, all topologically trivial insulators, interacting or not, have an accurate isoTNS representation.

Algorithms As a first application, we develop a TEBD algorithm for 2D lattice fermion systems (Fig. 3). We

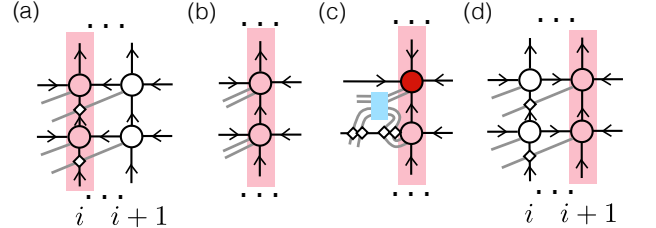


FIG. 3. TEBD in fisoTNS. (a) Column i is the OC. (b) Contraction of column i and $i+1$ into one column. (c) Apply the TEBD gates on the OC using the MPS TEBD method. (d) After the MM algorithm, column $i+1$ is now the OC.

view the 2D lattice as a collection of columns. When column i is the OC, we apply an operator $O_{i,i+1}$ supported on column i and $i+1$. To do this, we contract column i and $i+1$ into one combined column with two physical sites on each tensor and apply $O_{i,i+1}$ to it using MPS methods, truncating the column bond dimension when necessary. We do all truncation at the orthogonality center. Due to Theorem 1, maximizing the overlap of

the physical states before and after the truncation is the same as maximizing the overlap between the tensors at the center before and after the truncation. After $O_{i,i+1}$ is applied, we split the combined column back into column i and $i+1$, but with column $i+1$ now as the OC. After this, one can apply $O_{i+1,i+2}$. While the change of isometric center can be done exactly in 1D, moving the OC can only be done approximately in 2D with the *Moses Move* (MM) algorithm introduced in [7]. When the system Hamiltonian is made of two-column operators, the procedure above can be used for real and imaginary time evolution following a Trotter-Suzuki decomposition [16].

For fermions, two complications arise: the \mathbb{Z}_2 symmetry and the fermion signs in the tensor operations. For the \mathbb{Z}_2 symmetry, we follow the standard procedure of dealing with abelian symmetries in TNS [17], which allows a straightforward generalization of the MM to the symmetric case [15]. The fermion signs necessary for TEBD are *all* shown in Fig. 3 as swap gates. For example, the sign in Fig. 3 (a) follows from the tensor multiplication rule in Fig. 2 (b), written as Eq. 4.

Results To benchmark the TEBD algorithm, we study four prototypical fermionic systems on a square lattice: a trivial sublattice band insulator (Insulator), a $\phi = \pi$ Hofstadter model filled up to the Dirac point (Dirac), a $\phi = 2\pi/3$ Hofstadter model with two of the three bands filled, i.e. total Chern number = 1 (Chern), and a mean-field $p + ip$ superconductor ($p + ip$ SC),

$$H = \sum_{\langle ij \rangle} t_{ij} c_i^\dagger c_j + \eta_{ij} c_i^\dagger c_j^\dagger + \text{h.c.} + U \hat{n}_i \hat{n}_j - \sum_i \mu_i c_i^\dagger c_i \quad (5)$$

In all systems, $|t_{ij}| = 1$, and in $p + ip$ SC, $|\eta_{ij}| = 2$. The phases of t_{ij} (η_{ij}) set the flux (pairing symmetry) of the Hofstadter model ($p + ip$ SC). All systems are free with $U = 0$, except for the Chern band where we also explore the interacting case with $U = 0.5$. We set $\mu_i = \pm 1$ for the two sublattices for the insulator, and $\mu_i = -2$ in the $p + ip$ SC. For the Chern band (Dirac), we set a uniform μ_i such that the system is $2/3$ filled (half-filled).

We perform imaginary-time TEBD to find the ground state energies to compare with the exact values. To benchmark the Chern band with finite interaction, we use U(1)-conserved DMRG with bond dimension $\chi = 2400$. In Fig. 4, we show the error in ground state energy density at various bond dimensions χ and Trotter time $d\tau$. For fixed χ , there is a system-dependent optimal $d\tau$, below which the MM error $\sim d\tau^{-1}$ dominates, and above which the Trotter-splitting error $\sim d\tau^2$ dominates [7].

We further benchmark the real-time dynamics obtained by the TEBD algorithm. We start with the ground state of the Chern insulator, $|\text{GS}\rangle$, obtained as described above, add an electron at the corner, $|\text{ext}\rangle = c^\dagger |\text{GS}\rangle$, and then evolve the excited state for time t , $|t\rangle = e^{-iHt} |\text{ext}\rangle$. For Chern bands, adding an electron at the corner is expected to excite mostly the *chiral* edge mode. We show

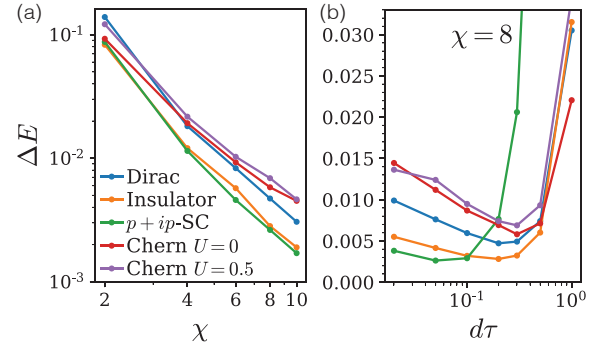


FIG. 4. Error in ground state energy density. (a) is presented with the optimal $d\tau$. The system size is 9×9 for Chern and 10×10 for Dirac, Insulator, and $p + ip$ SC.

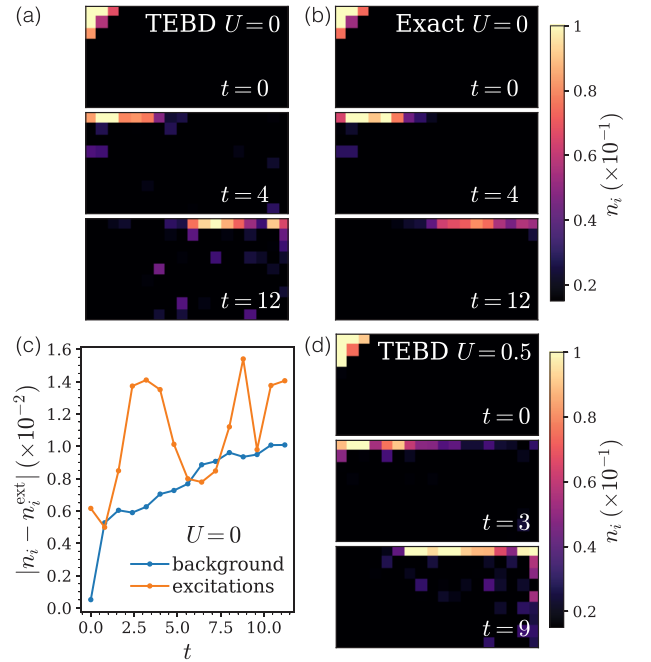


FIG. 5. Real-time evolution of the $\phi = 2\pi/3$ Hofstadter model starting with a single-site excitation at the top-left corner of a 9×18 system. (a-b, d) Number density n_i relative to the ground state. Different time slices are taken for the free and interacting case to accommodate the renormalized velocity due to interaction. (c) Average errors in number density when $U = 0$. The background curve averages over all sites, and the excitation curve averages over sites that support high density due to the excitation.

the number density of each site relative to the ground state $n_i = \langle t | \hat{n}_i | t \rangle - \langle \text{GS} | \hat{n}_i | \text{GS} \rangle$ in Fig. 5. For both the free and the interacting case, the excitation propagates along the edge without leaking into the bulk. Comparing Fig. 5 (a) and (b), the TEBD algorithm with $dt = 0.2$ reproduces not only the velocity of the edge mode but also subtle features of the density profile (e.g. at $t = 4$).

The error in number density is shown in Fig. 5 (c).

Conclusion We generalized isoTNS to fermionic systems, which allows for efficient adaptations of 1D MPS algorithms to 2D fermionic systems. The 2D TEBD algorithm we developed as the first application produces ground states and real-time dynamics of low-lying states with good accuracy. As an example, it captures the chiral edge dynamics on the boundary of a Chern insulator. Numerical results suggested that the accuracy for ground-state energies improves as a power law of the bond dimension for an insulator, a model with gapless Dirac fermion, and models with a chiral edge mode. Future directions include the study of the representing power of fisoTNS and the exploration of strongly correlated electronic states with fisoTNS in 2D. Most interestingly, the fisoTNS representation should tell us something about the nature of the system’s multipartite entanglement, which we will explore in the future.

Acknowledgement. We thank Johannes Hauschild and Peter Lunts for helpful discussions. YW acknowledges financial support from the iTHEMS fellowship. This work is partly supported by the Gordon and Betty Moore foundation. TW and MZ were supported by the U.S. DOE, Office of Science, Office of Basic Energy Sciences, Materials Sciences and Engineering Division, under Contract No. DE-AC02-05CH11231, within the van der Waals Heterostructures Program (KCWF16). The code is implemented based on the TenPy TNS code base [17].

* These two authors contributed equally

- [1] U. Schollwöck, *Rev. Mod. Phys.* **77**, 259 (2005).
- [2] S. R. White, *Phys. Rev. Lett.* **69**, 2863 (1992).
- [3] M. Fannes, B. Nachtergaele, and R. Werner, *EPL (Europhysics Letters)* **10**, 633 (1989).
- [4] A. Klümper, A. Schadschneider, and J. Zittartz, *EPL (Europhysics Letters)* **24**, 293 (1993).
- [5] F. Verstraete and J. I. Cirac, “Renormalization algorithms for quantum-many body systems in two and higher dimensions,” (2004).
- [6] J. I. Cirac, D. Pérez-García, N. Schuch, and F. Verstraete, *Rev. Mod. Phys.* **93**, 045003 (2021).
- [7] M. P. Zaletel and F. Pollmann, *Phys. Rev. Lett.* **124**, 037201 (2020).
- [8] S.-H. Lin, M. Zaletel, and F. Pollmann, arXiv preprint arXiv:2112.08394 (2021).
- [9] R. Orús, *The European Physical Journal B* **87**, 1 (2014).
- [10] C. V. Kraus, N. Schuch, F. Verstraete, and J. I. Cirac, *Phys. Rev. A* **81**, 052338 (2010).
- [11] T. Barthel, C. Pineda, and J. Eisert, *Phys. Rev. A* **80**, 042333 (2009).
- [12] I. Pizorn and F. Verstraete, *Phys. Rev. B* **81**, 245110 (2010).
- [13] Q.-Q. Shi, S.-H. Li, J.-H. Zhao, and H.-Q. Zhou, arXiv preprint arXiv:0907.5520 (2009).
- [14] P. Corboz, R. Orús, B. Bauer, and G. Vidal, *Phys. Rev. B* **81**, 165104 (2010).

- [15] Z. Dai, Y. Wu, T. Wang, and M. Zaletel, (2022).
- [16] N. Hatano and M. Suzuki, “Finding exponential product formulas of higher orders,” in *Quantum Annealing and Other Optimization Methods*, edited by A. Das and B. K. Chakrabarti (Springer Berlin Heidelberg, Berlin, Heidelberg, 2005) pp. 37–68.
- [17] J. Hauschild and F. Pollmann, *SciPost Physics Lecture Notes*, 005 (2018).

Replacing multiple entangled fermion pairs by a single pair

Consider the following ansatz where Site 1 and Site 2 share n bonds $i_1 \dots i_n$, with parity label $I_1 \dots I_n$ (Fig. 6).

$$|\psi\rangle = \langle \hat{A}[1] \hat{A}[2] \dots (1 + \beta_{1,1}^\dagger \alpha_{2,1}^\dagger) \dots (1 + \beta_{1,n}^\dagger \alpha_{2,n}^\dagger) \rangle_{\text{aux}} |vac\rangle \quad (6)$$

$$\hat{A}[1] = A[1]_{l_1 i_1 \dots i_n d_1 u_1}^{k_1} c_1^{\dagger k_1} \delta_1^{D_1} \beta_{1,n}^{I_n} \dots \beta_{1,1}^{I_1} \gamma_1^{U_1} \alpha_1^{L_1}, \quad (7)$$

$$\hat{A}[2] = A[2]_{i_1 \dots i_n r_2 d_2 u_2}^k c_2^{\dagger k_2} \delta_2^{D_2} \beta_2^{R_2} \gamma_2^{U_2} \alpha_{2,1}^{I_1} \dots \alpha_{2,n}^{I_n}, \quad (8)$$

$$(9)$$

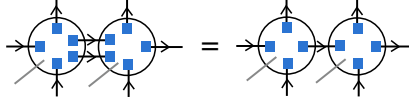


FIG. 6. LHS: Site 1 (left) and Site 2 (right) connected by 2 fermionic bonds with indices i_1 (top) and i_2 (bottom). RHS: Site 1 and Site 2 share a single bond with the index i

We claim that replacing the n bonds with a single bond labeled by i (the collection of indices $i_1 \dots i_n$) and a parity label $I = I_1 + \dots + I_n$ gives the same physical state:

$$|\psi\rangle = \langle \hat{A}'[1] \hat{A}'[2] \dots (1 + \beta_1^\dagger \alpha_2^\dagger) \rangle_{\text{aux}} |vac\rangle \quad (10)$$

The new tensors are simply defined as $\hat{A}'[1] = A[1]_{l_1 i d_1 u_1}^{k_1} c_1^{\dagger k_1} \delta_1^{D_1} \beta_1^I \gamma_1^{U_1} \alpha_1^{L_1} = A[1]_{l_1 i_1 \dots i_n d_1 u_1}^{k_1} c_1^{\dagger k_1} \delta_1^{D_1} \beta_{1,n}^{I_n} \gamma_1^{U_1} \alpha_1^{L_1}$, $\hat{A}'[2] = A[2]_{i r_2 d_2 u_2}^{k_2} c_2^{\dagger k_2} \delta_2^{D_2} \beta_2^{R_2} \gamma_2^{U_2} \alpha_2^I$. This equivalence follows from the fact that β_1^I (α_2^I) has the same fermion parity as $\beta_{1,n}^{I_n} \dots \beta_{1,1}^{I_1}$ ($\alpha_{1,1}^{I_1} \dots \alpha_{1,n}^{I_n}$) and they give the same result after contracting with the corresponding creation operators

$$\langle \alpha_2^I \beta_1^I (1 + \beta_1^\dagger \alpha_2^\dagger) \rangle_{\beta_1, \alpha_2} = 1, \quad \forall I \quad (11)$$

$$\langle \alpha_{2,1}^{I_1} \dots \alpha_{2,n}^{I_n} \beta_{1,n}^{I_n} \dots \beta_{1,1}^{I_1} (1 + \beta_{1,1}^\dagger \alpha_{2,1}^\dagger) \dots (1 + \beta_{1,n}^\dagger \alpha_{2,n}^\dagger) \rangle_{\beta_{1,1}, \alpha_{2,1}, \dots, \beta_{1,n}, \alpha_{2,n}} = 1, \quad \forall I_1, \dots, I_n \quad (12)$$

Note that this result relies on the counterclockwise ordering convention of fermion operators.

Equivalence of the swap gate ansatz and the virtual fermion ansatz

We provide a self-contained introduction to the swap-gate ansatz and show the equivalence to the virtual-fermion ansatz by explicitly contracting the virtual fermions.

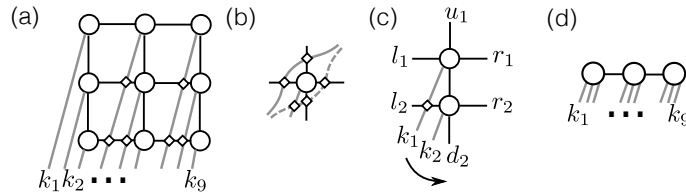


FIG. 7. (a) Representation of the wavefunction in the swap gate convention. The wavefunction is given by the contraction of all tensors and swap gates. (b) Deforming the physical legs give does not affect the wavefunction. (c) The contraction of two tensors. (d) The result of contracting the three tensors in each column.

The swap-gate ansatz is defined by 2 rules. First, each tensor is \mathbb{Z}_2 -graded and has zero total parity: each virtual index has a parity label 0 or 1, and the tensor element is zero unless the corresponding labels on all legs add to an

even number. Second, whenever two lines with parity I and J cross each other, a swap gate $(-1)^{IJ}$ is added. The wavefunction ψ^{k_1, \dots, k_N} in the basis of $c_1^{\dagger k_1} c_2^{\dagger k_2} \dots c_N^{\dagger k_N} |vac\rangle$ is defined as the contraction of all tensors and swap gates as shown in Fig. 7(a). In Fig. 7(a), the order of the endpoints of physical lines from left to right corresponds to the fermion order of the Fock-space basis. If we were to swap the endpoints of two physical fermion lines, for example, k_2 and k_3 , we would add a swap-gate $(-1)^{k_2 k_3}$. It is easy to see the resulting wavefunction is the wavefunction of the same state written in a different basis $c_1^{\dagger k_1} c_3^{\dagger k_3} c_2^{\dagger k_2} \dots c_N^{\dagger k_N} |vac\rangle$. On the other hand, Deforming the physical lines without changing the order of endpoints does not change the wavefunction. For example, in Fig. 7(b), the physical line k_1 may pass through the tensor A_{lrdu}^k (parity label k, L, R, D, U) in two ways (the solid line and the dashed line). For the solid line, there are 2 swap gates giving $(-1)^{k_1(L+U)}$; for the dashed line, there are 3 swap gates giving $(-1)^{k_1(k+D+R)}$. Since the tensor has even parity, the tensor elements are zero unless $L+U = k+D+R \pmod{2}$. Thus, the two choices of swap gates produce the same wavefunction.

Now we show the virtual-fermion ansatz we discussed in the main text produces the same wavefunction by explicitly contracting all virtual fermions. We first contract all tensors in each column, and then contract the resulting tensors of different columns. As shown in Fig. 7(c), we can replace the top tensor ($A[1]$) and the bottom tensor ($A[2]$) with a single tensor (Θ , fermion operators ordered counterclockwise starting from physical legs). Using the virtual-fermion ansatz, we have

$$\hat{\Theta} = \sum_{r=1}^{l_2} \langle A[2]_{l_2 r_2 d_2 i}^{k_2} c_2^{\dagger k_2} \delta_2^{D_2} \beta_2^{R_2} \gamma_2^I \alpha_2^{L_2} A[1]_{l_1 r_1 i u_1}^{k_1} c_1^{\dagger k_1} \delta_1^I \beta_1^{R_1} \gamma_1^{U_1} \alpha_1^{L_1} (1 + \delta_1^{\dagger} \gamma_2^{\dagger}) \rangle_{\delta_1, \gamma_2} \quad (13)$$

$$\hat{\Theta} \equiv \Theta_{l_1 l_2 r_1 r_2 d_2 u_1}^{k_1 k_2} c_1^{\dagger k_1} c_2^{\dagger k_2} \delta_2^{D_2} \beta_2^{R_2} \beta_1^{R_1} \gamma_1^{U_1} \alpha_1^{L_1} \alpha_2^{L_2} \quad (14)$$

We bring the RHS of Eq. 13 to the form of 14 in 3 steps: bringing $c_1^{\dagger k_1}$ to the left, moving $\alpha_2^{L_2}$ to the right, and contracting $\gamma_2^I \delta_1^I$ with the creation operator. In the second step, $\alpha_2^{L_2}$ is swapped with the underlined operators, with total parity k_1 , hence the sign $(-1)^{k_1 L_2}$ in Fig. 7. We contract tensors in a column from the top to the bottom. Each physical leg needs to pass through all horizontal legs on the left of the column below it, hence the minus signs indicated by the swap gates in Fig. 7(a). After the vertical contractions, we get a tensor for each column (Fig. 7(d)). It is easy to see that contracting them does not give any extra sign.

The argument above shows the equivalence of the virtual-fermion ansatz and the swap-gate ansatz in the case that the physical legs are extended outside as in Fig. 7(a). Using the consistency relations discussed earlier in this section, this equivalence stands no matter how we extend the physical legs outside.

Alternative interpretation of the isometric condition in the swap-gate convention

In the swap-gate convention, it's important to fix an order of the legs for tensors representing the state $|\psi\rangle$. We choose to the order in Fig. 8(a). Counting counterclockwise from the physical leg, the order is ‘physical, down, right, up, left’, the same as the order of fermion operators in the virtual-fermion ansatz. The order of legs for tensors representing the conjugate $\langle\psi|$ must be the opposite as the order for tensors representing $|\psi\rangle$ (Fig. 8(b)). This is in analogy of the reverse of orders when taking the conjugate of products of operators.

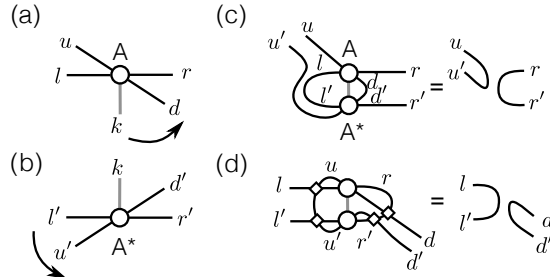


FIG. 8. (a) The legs of tensors representing $|\psi\rangle$ are ordered counterclockwise according to ‘physical, down, right, up, left’. (b) The legs of tensors representing $\langle\psi|$ have the opposite order. (c) The isometric condition for fTNS in the lower-left quadrant. (d) The isometric condition for fTNS in the upper-right quadrant.

With this convention, we can give a pictorial definition of the isometric condition for fermionic tensor networks.

The isometric condition is shown in Fig. 8(c) for tensors in the lower-left quadrant and in Fig. 8(d) for tensors in the upper-right quadrant. In the lower-left quadrant, if contracting the left, physical, and down legs gives delta functions in the right and up legs, the network can be contracted trivially from the lower-left corner to the central cross. Since there is no cross of the legs, the isometric condition is simply $\sum_{k,l,d} A_{lrdu}^k A_{lr'du'}^{*k} = \delta_{uu'} \delta_{rr'}$, namely A is an isometry.

On the other hand, in the upper-right quadrant, we need to contract tensors from the upper-right to the lower-left. The contraction of the up and right legs leads to 4 swap gates, hence the isometric condition $\sum_{k,r,u} (-1)^{u(L+L')+r(D+D')} A_{lrdu}^k A_{lr'du'}^{*k} = \delta_{ll'} \delta_{dd'}$. This condition is equivalent to that $(-1)^{UL+RD} A$ is an isometry. We can equivalently write this condition as $(-1)^{UD+RD} A$ is an isometry, by changing L into $D+U+k+R$ (total parity even) and using the fact that $(-1)^{U(k+U+R)}$ is a unitary transformation on the incoming legs (diagonal matrix with ± 1 diagonal elements). Finally, we can simplify the isometric condition to that $(-1)^{kD} A$ is an isometry by changing $U+R$ to $k+D+L$ and using the fact that $(-1)^{D(D+L)}$ is a unitary on the outgoing legs.

A similar argument shows that the isometric condition in the lower-right and upper-left quadrants is also that A is an isometry from outgoing legs to incoming legs.

Proof of Theorem 1

The tensor network representations of $|\psi\rangle$ and $\langle\psi|$ are

$$|\psi'\rangle = \langle \hat{\Lambda}'_{(x_0, y_0)} \prod_{(x,y) \neq (x_0, y_0)} \hat{A}_{(x,y)} \prod_{(x,y)} \hat{V}_{(x,y)} \rangle_{\text{aux}} |vac\rangle, \quad (15)$$

$$\langle\psi| = \langle vac| \langle \hat{\Lambda}^*_{(x_0, y_0)} \prod_{(x,y) \neq (x_0, y_0)} \hat{A}^*_{(x,y)} \prod_{(x,y)} \hat{V}^*_{(x,y)} \rangle_{\text{aux}}, \quad (16)$$

$$\hat{V}^*_{(x,y)} \equiv (1 + \bar{\alpha}^\dagger_{(x+1,y)} \bar{\beta}^\dagger_{(x,y)}) (1 + \bar{\gamma}^\dagger_{(x,y)} \bar{\delta}^\dagger_{(x,y+1)}) \quad (17)$$

To get Eq. 16 we took the conjugate of Eq. 1 and made a particle-hole transformation of the auxiliary fermions. Consider contracting the fermionic tensors and the bond operators in the expression of $\langle\psi|\psi'\rangle$ from the lower-left corner toward the orthogonality center. The tensor at the lower-left corner has trivial left and down legs, $l = L = 0, d = D = 0$; the definition of the isometric condition guarantees that contracting the tensor with its conjugate (summing over the physical index) gives $\delta_{uu'} \bar{\gamma}_1^{U'} \gamma_1^U \delta_{rr'} \bar{\beta}_1^R \beta_1^R$. Then contracting $\delta_{rr'} \bar{\beta}_1^R \beta_1^R$ with the bond operator on its right gives $\langle \delta_{rr'} \bar{\beta}_1^R \beta_1^R (1 + \beta_1^\dagger \alpha_2^\dagger) (1 + \bar{\alpha}_1^\dagger \bar{\beta}_2^\dagger) \rangle_{\beta_1, \bar{\beta}_1} = \delta_{l,l'} \alpha_2^{L'} \bar{\alpha}_2^{L'}$, which contracts with the second tensor on the bottom row and its conjugate. We get the desired result by continuing this process iteratively from the 4 corners to the center. \square

Finite-depth-local-unitary circuits preserve fisoTNS

We prove that if a fermionic state has a finite-bond-dimension fisoTNS representation in the thermodynamic limit, it can still be represented to any accuracy by a finite-bond-dimension fisoTNS after the action of a finite-depth-local-unitary (FDLU) circuit.

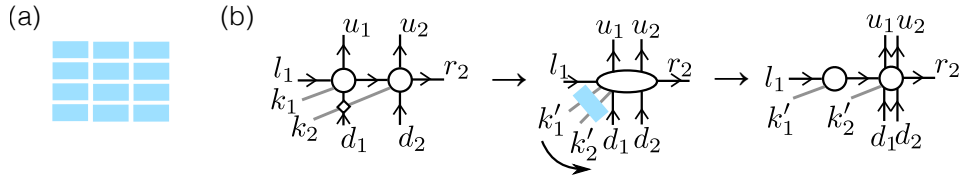


FIG. 9. (a) One layer of local 2-site unitary gates. (b) The process of getting the fisoTNS representation of the final state in the lower-left quadrant

Since any FDLU can be deformed into finite-depth 2-site unitaries to any desired accuracy, it suffices to derive a fisoTNS representation after a single layer of 2-site unitaries (Fig. 9). We demonstrate it in the lower-left quadrant.

To do so, we combine the 2 tensors on the neighboring sites and act it by the corresponding unitary gate.

$$\hat{\Theta} = \langle A[1]c_1^{\dagger k_1} \delta_1^{D_1} \beta_1^I \gamma_1^{U_1} \alpha_1^{L_1} A[2] \underline{c_2^{\dagger k_2} \delta_2^{D_2} \beta_2^{R_2} \gamma_2^{U_2} \alpha_2^I} (1 + \beta_1^{\dagger} \alpha_2^{\dagger}) \rangle_{\beta_1, \alpha_2} \quad (18)$$

$$\hat{\Theta} \equiv \Theta_{l_1 r_2 d_1 d_2 u_1 u_2}^{k_1 k_2} c_1^{\dagger k_1} c_2^{\dagger k_2} \delta_1^{D_1} \delta_2^{D_2} \beta_2^{R_2} \gamma_2^{U_2} \gamma_1^{U_1} \alpha_1^{L_1} \quad (19)$$

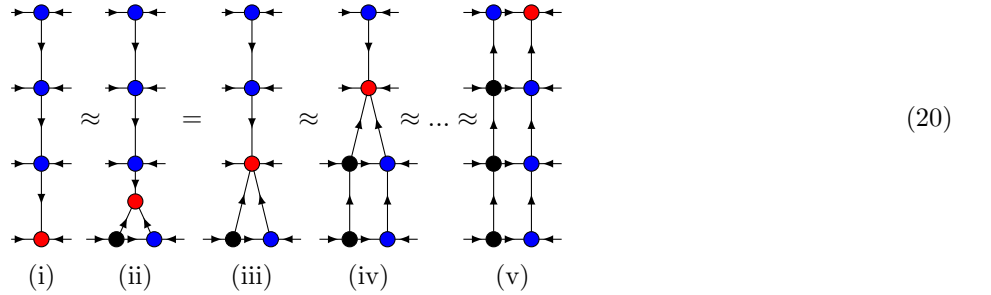
To derive an expression for tensor Θ , we place the underlined operators between δ_1 and β_1 , contract $\alpha_2^I \beta_1^I$ with the creation operator, and then swap $\delta_1^{D_1}$ and $c_2^{k_2}$. Since the underlined operators have even total parity, moving it does not give a minus sign; the only step that may produce a minus sign is the swap. Thus we have $\Theta = A[1]A[2](-1)^{k_2 D_1}$. Note that if $A[1]$ and $A[2]$ are isometries with respect to the arrows in Fig. 9, The contraction $A[1]A[2]$ is the composition of two isometries, which is still an isometry. Θ is also an isometry since the extra factor $(-1)^{k_2 D_1}$ is a unitary transformation on the incoming legs. For the same reason, the physical gate acting on the leg k_1 and k_2 , $\Theta' = U\Theta$, also preserves the isometric structure.

Then we split the tensor into two isometric tensors as illustrated in Fig. 9(b). The internal bond is just the combination of k'_1 and l_1 . The left tensor is the identity matrix from the outgoing leg to the incoming legs, with trivial up and down legs. The right tensor is Θ' .

With minor changes regarding the corresponding definition of isometric form, the same procedure applies to other quadrants and the central cross.

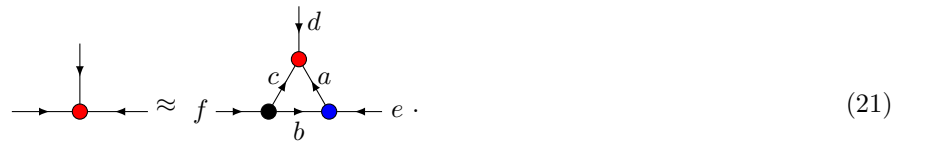
Moses Move for symmetric TNS

The MM algorithm for isoTNS without symmetry is described in [7]. We repeat it here. The MM algorithm performs the following decomposition of the orthogonality column (OC) into a left-canonical isometric column and a new OC:



where the physical legs of the isoTNS are grouped either with the left or the right virtual index. The MM algorithm effectively “unzips” the original OC by iteratively splitting the (red) 5-leg tensor at the orthogonality center into 3 smaller tensors.

Because of the isometric form, each splitting step in the MM can be reduced to the following local tripartite decomposition:



where we view the five-leg red tensor in (iii) of Eq. 20 as a tripartite tensor by grouping its lower left and lower right index respectively with its left and right index. Typically, one needs to place a limit on the maximum internal bond dimension of the decomposition, which generally disallows an exact equality in Eq. 21. This decomposition can be done as follows.

1. One first groups index d and e and then performs an SVD decomposition on the grouped matrix $\Psi_{f:de} = U s V^\dagger$. The index between U and $s V^\dagger$, arising from the SVD decomposition, is split into indices b and c . The permutational freedom in splitting the indices will be fixed by the disentangler u in step 2. The bond dimensions of b and c are truncated when necessary.

2. Note that the SVD in step 1 has a unitary freedom, u , acting on the indices b and c : $UsV^\dagger = Uuu^\dagger sV^\dagger$. To make use of this freedom, the tensor $u^\dagger sV^\dagger$ is reshaped into a matrix Θ with its b and e indices grouped as the column index and the c and d grouped as the row index. u is chosen so that the Renyi- α entropy of the matrix Θ is minimized, for a certain Renyi index α . We choose $\alpha = 1/2$. The disentangler, u , is optimized in order to minimize the error of the SVD truncation in step 3. The black tensor is then equal to the reshaped Uu .
3. Perform an SVD on Θ : $\Theta = U's'V'^\dagger$. The blue tensor is then equal to the reshaped V'^\dagger after truncation, and the red tensor is equal to the truncated $U's'$.

In the presence of \mathbb{Z}_2 symmetry, all operations described above respect the charge structure naturally, except that in step 1, when the index between U and sV^\dagger is split into indices b and c , a charge structure on b and c has to be prescribed. Let n_0 and n_1 respectively be the size of the charge-0 and charge-1 block on the bond between U and sV^\dagger before the splitting. Then we have to prescribe the size of the charge-0 and charge-1 block on both bond b and c . We denote them as n_0^b, n_1^b, n_0^c , and n_1^c . They have to satisfy the constraint

$$n_0^b n_0^c + n_1^b n_1^c \leq n_0 \quad (22)$$

$$n_0^b n_1^c + n_1^b n_0^c \leq n_1 \quad (23)$$

The charge block of size $n_0 n_1 - (n_0^b + n_1^b)(n_0^c + n_1^c)$ will need to be truncated. We choose the n s such that the size of the truncated block is the smallest. When multiple sets of n s tie, we choose the set for which the standard deviation of the list $[n_0^b, n_1^b, n_0^c, n_1^c]$ is the least, i.e. we choose the evenest charge block sizes. In the code, $[n_0^b, n_1^b, n_0^c, n_1^c]$ is chosen via explicit enumeration and comparison. The computational cost for this is negligible.