A New Era in Tensor Networks: Exploring Quantics Tensor Networks and Tensor Cross Interpolation

Hiroshi Shinaoka

Contents

A Overview of the lecture and acknowledgement

Day 1 (1.5 hours):
1. Short talk on TCI and quantics and their applications
2. CI of matrices

Day 2 (1.5 hours):
1. TCI

Day 3 (1.5 hours):
1. Applications
2. Hands-on exercises (if time permits)

This lecture is based on [1] and many fruitful discussions with the authors, especially Y. Fernández, X. Waintal, J. von Delft, and M. K. Ritter, as well as M. Stoudenmire. The lecture follows [1], but I have modified some parts to make it more accessible to beginners and more practical based on new insights from my research.

B Cross Interpolation (CI) of matrices

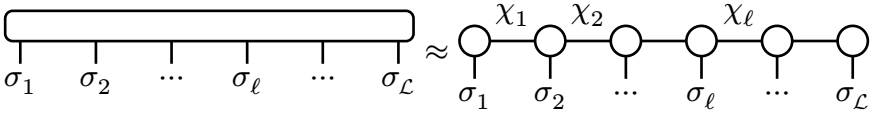
B.1 Basic linear algebra

- The rank of a matrix is the number of linearly independent rows or columns (column rank and row rank are the same).
- A rank-revealing factorization is a factorization/decomposition that reveals the rank of a matrix as follows:
  - SVD: $A = U\Sigma V^\dagger$
- Condition number of A: $\kappa(A)$ is the ratio of the largest and smallest singular values of A.
- Do not compute the inverse of an ill-conditioned matrix. BA^{-1} can be computed by solving $AX = B$ by a numerically stable solver.

B.2 MPS

Let's first review matrix product states (MPSs) before we move on to matrix CI and TCI. This is because MPS and TCI share some similar concepts. An MPS is a 1D tensor network with a chain of (core) tensors:

$$F_{\sigma_1\sigma_2\cdots\sigma_\ell\cdots\sigma_\mathcal{L}} = \approx \tag{1}$$



An MPS has a bond dimension χ that controls the number of parameters in the MPS. The core tensors are not unique and an MPS has a gauge freedom. A tensor can be decomposed into an MPS by a series of singular value decompositions (SVD).

MPS/DMRG was originally developed for quantum many-body systems, but it has been applied to various fields in science and engineering. In applied mathematics, it is called tensor train (TT) decomposition.

B.3 Cross Interpolation (CI)

Let me introduce CI as an alternative to SVD. In SVD, the left and right singular vectors are results of complex computations of the original matrix. The result of CI is more interpretable than SVD in the sense that the left and right matrices are slices of the original matrix.

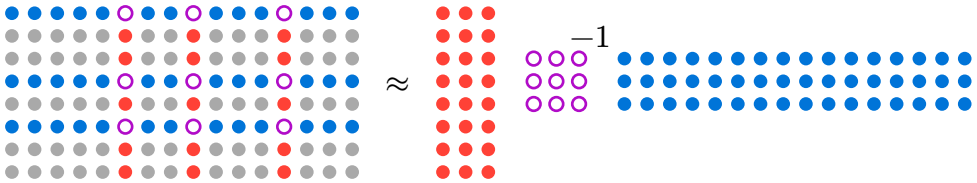
B.3.1 CI formula

Slicing $m \times n$ matrix: $A \in \mathbb{R}^{m\times n}$ or $A \in \mathbb{C}^{m\times n}$.

- Ordered set of all rows: $\mathbb{I} = \{1, ..., m\}$
- Ordered set of all columns: $\mathbb{J} = \{1, ..., n\}$
- Subset of rows (pivot list): $\mathcal{I} = \left\{i_1, ..., i_{\tilde{\chi}}\right\}$
- Subset of columns (pivot list): $\mathcal{J} = \left\{j_1, ..., j_{\tilde{\chi}}\right\}$
- Slice of A: $[A(\mathcal{I}, \mathcal{J})]_{\alpha\beta} = \{A(\mathbb{I}, \mathbb{J})\}_{i_\alpha, j_\beta}$

CI formula of rank χ for A:

$$A(\mathbb{I}, \mathbb{J}) \approx CP^{-1}R = A(\mathbb{I}, \mathcal{J})A(\mathcal{I}, \mathcal{J})^{-1}A(\mathcal{I}, \mathbb{J}) \equiv \tilde{A}(\mathbb{I}, \mathbb{J}) \tag{2}$$



If P is not singular, the CI formula satisfies the following properties:

1. $\tilde{A}(\mathcal{I}, \mathbb{J}) = A(\mathcal{I}, \mathbb{J})$: CI matches the original matrix in the selected rows.
2. $\tilde{A}(\mathbb{I}, \mathcal{J}) = A(\mathbb{I}, \mathcal{J})$: CI matches the original matrix in the selected columns.
3. If A is rank χ and we take $\tilde{\chi} = \chi$, $\tilde{A} = A$: The CI formula matches the original matrix on all the elements.

Element-wise error bound [2]

For a fixed $\tilde{\chi}$, if $\mathcal{I}^*$ and $\mathcal{J}^*$ are the pivot lists of size $\tilde{\chi}$ maximizing the volume of $A(\mathcal{I}, \mathcal{J})$ [= the absolute value of the determinant of $A(\mathcal{I}, \mathcal{J})$] as

$$\mathcal{I}^*, \mathcal{J}^* = \mathrm{argmax}_{\mathcal{I},\mathcal{J}} \ \mathrm{vol}[A(\mathcal{I}, \mathcal{J})], \tag{3}$$

then the CI formula satisfies

$$\left\| A - CP^{-1}R \right\|_\infty \leq (r+1)\sigma_{r+1}, \tag{4}$$

where σ_r is the r-th singular value of A and $\|A\|_\infty = \max_{ij}|A_{ij}|$.

Excerise: Prove property 1

Answer: $\tilde{A}(\mathcal{I}, \mathbb{J}) = \left[A(\mathbb{I}, \mathcal{J})A(\mathcal{I}, \mathcal{J})^{-1}A(\mathcal{I}, \mathbb{J})\right](\mathcal{I}, \mathbb{J}) = A(\mathcal{I}, \mathcal{J})A(\mathcal{I}, \mathcal{J})^{-1}A(\mathcal{I}, \mathbb{J}) = A(\mathcal{I}, \mathbb{J})$.

In practice, we avoid computing P^{-1} explicitly. For simplicity of notation, we assume the pivots are permutated to the first block:

$$A = \begin{pmatrix} A(\mathcal{I}_1, \mathcal{J}_1) & A(\mathcal{I}_1, \mathcal{J}_2) \\ A(\mathcal{I}_2, \mathcal{J}_1) & A(\mathcal{I}_2, \mathcal{J}_2) \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \tag{5}$$

$$\begin{aligned} \tilde{A} &= CP^{-1}R \\ &= \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} A_{11}^{-1} (A_{11} \ \ A_{12}) \\ &= \underbrace{\begin{pmatrix} \mathbb{1} \\ A_{21}A_{11}^{-1} \end{pmatrix}}_{CP^{-1}} (A_{11} \ \ A_{12}) = \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} \underbrace{(\mathbb{1} \ \ A_{11}^{-1}A_{12})}_{P^{-1}R} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{21}A_{11}^{-1}A_{12} \end{pmatrix}. \end{aligned} \tag{6}$$

$$A - \tilde{A} = \begin{pmatrix} 0 & 0 \\ 0 & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{pmatrix}. \tag{7}$$

Remarks:

- $A_{12}A_{11}^{-1}(=X)$ can be computed by QR (see Appendix B of [3]) or solving $A_{11}X = A_{12}$ by prrLU (see Section B.3.3). The latter is more stable.
- CP^{-1} and $P^{-1}R$ are like the identity matrix for for the pivot lists $\mathcal{I}_1$ and $\mathcal{J}_1$, which will be important for understanding the canonicalization of TCI (see Section C.5).

B.3.2 Pivot selection algorithms

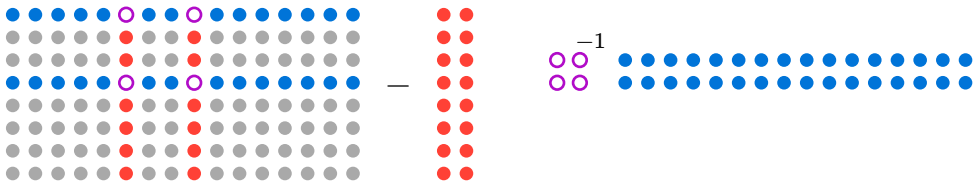
Finding the optimal pivot lists requires a combinatorial search over all possible pivot lists. There are many heuristic methods based on greedy algorithms, e.g., maxvol algorithm[4].

Toy greedy algorithm (only for educational purposes)

1. Choose the row i_1 and column j_1 that maximize $|A_{ij}|$. Set $\mathcal{I}_1 = \{i_1\}$ and $\mathcal{J}_1 = \{j_1\}$.
2. Compute the residual matrix $A - \tilde{A}_{\tilde{\chi}=1}$, where $\tilde{A}_{\tilde{\chi}=1} = A[\mathbb{I}, \mathcal{J}_1]A[\mathcal{I}_1, \mathcal{J}_1]^{-1}A[\mathcal{I}_1, \mathbb{J}]$.

$A - \tilde{A}_{\chi=1} =$ 

3. Find the row i_2 and column j_2 that maximize $|\left(A - \tilde{A}_{\tilde{\chi}=1}\right)_{ij}|$. Add i_2 and j_2 to the pivot lists as $\mathcal{I}_2 = \{i_1, i_2\}$ and $\mathcal{J}_2 = \{j_1, j_2\}$.
4. Compute the residual matrix $A - \tilde{A}_{\tilde{\chi}=2}$ and find the next pivots i_3 and j_3.

$A - \tilde{A}_{\chi=2} =$ 

5. Repeat the above steps until some stopping criterion is met: i.e., $\left\|A - \tilde{A}_{\tilde{\chi}}\right\|_\infty < \varepsilon$.

B.3.3 Partial rank-revealing (prr) LU decomposition

prrLU is a numerically stable implementation of the greedy algorithm [1]:

- performs pivot selection (mathematically the same as the above greedy algorithm),
- computes the CI formula, i.e., CP^{-1} and R in post-processing in a numerically stable manner.

In prrLU (=truncated LU decomposition with column and row pivoting), any rectangular matrix $A \in \mathbb{R}^{m\times n}$ or $A \in \mathbb{C}^{m\times n}$ can decomposed as

$$P_{\text{perm}}AQ_{\text{perm}} = \underbrace{\begin{pmatrix} L_{11} \\ L_{21} \end{pmatrix} D(U_{11} \;\; U_{12})}_{P_{\text{perm}}\tilde{A}Q_{\text{perm}}} + \delta, \tag{8}$$

where

- P_{perm} and Q_{perm} permute the $\tilde{\chi}$ pivots to the first block,
- L_{11}: lower triangular matrix of size $\tilde{\chi} \times \tilde{\chi}$ with unit diagonal,
- L_{22}: dense matrix of size $(m - \tilde{\chi}) \times \tilde{\chi}$,
- D: diagonal matrix of size $\tilde{\chi} \times \tilde{\chi}$,
- U_{11}: upper triangular matrix of size $\tilde{\chi} \times \tilde{\chi}$ with unit diagonal,
- U_{12}: dense matrix of size $\tilde{\chi} \times (n - \tilde{\chi})$,

- δ: residual matrix ($= P_{\text{perm}}\big(A - \tilde{A}\big)Q_{\text{perm}}$).

Matrices in CI can be computed from the results of prrLU [1]. For instance,

$$\begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} (A_{11})^{-1} = \begin{pmatrix} \mathbb{I}_{\tilde{\chi}} \\ L_{21}(L_{11})^{-1} \end{pmatrix}, \tag{9}$$

where L_{11} is *well-conditioned*. For the details, see Sec 3.3 of [1].

B.4 Numerical experiments

The Hilbert matrix, $H_{ij} = \frac{1}{i+j+1}$, is a well-known ill-conditioned low-rank matrix. The singular values of $N \times N$ Hilbert matrix are super-exponentially decaying at large N.

4×4 Hilbert matrix:

$$\begin{pmatrix} \frac{1}{1} & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{pmatrix}$$

Search over all pivot lists of size $\tilde{\chi} = 4$ for $N = 10$ and compare the residual of the CI formula in the maximum norm sense with that of prrLU. You will find the following:

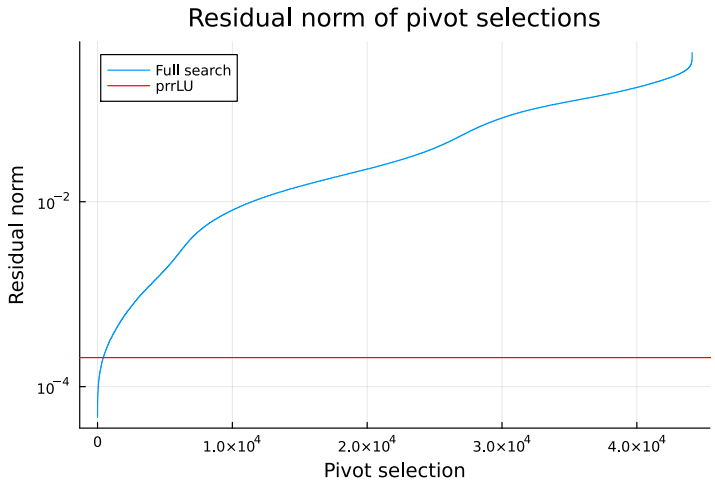


Figure 1: Residual norm of pivot selections for $N = 10$ and $\tilde{\chi} = 4$. The red line shows the residual norm of prrLU, which is close to optimal.

Homework Reproduce the above figure by extending the code in Listing 1.

Remark: There is still room to improve the accuracy of pivot selection. See [5] for recent attempts in this direction.

C Tensor Cross Interpolation (TCI) of tensors

Like exteding SVD to MPS, CI can be extended to higher-order tensors [1,3,6–8].

C.1 Notation

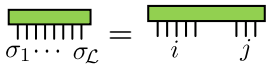
A $\mathcal{L}$-leg tensor F is defined as:

$$F_{\sigma_1\sigma_2\cdots\sigma_\ell\cdots\sigma_\mathcal{L}} = \quad (10)$$

- An external index σ_ℓ ($\ell \in \{1, 2...\mathcal{L}\}$) takes d_ℓ different values from a set $\mathbb{S}_\ell$.
- $\mathbb{I}_\ell = \mathbb{S}_1 \times \mathbb{S}_2 \times ... \times \mathbb{S}_\ell$ is the set of all row multi-indices up to site ℓ. An element $i \in \mathbb{I}_\ell$ is a row multi-index taking the form $i = (\sigma_1, \sigma_2, ..., \sigma_\ell)$.
- $\mathbb{J}_\ell = \mathbb{S}_\ell \times \mathbb{S}_{\ell+1} \times ... \times \mathbb{S}_\mathcal{L}$ is the set of all column multi-indices from site ℓ upwards. An element $j \in \mathbb{J}_\ell$ is a column multi-index taking the form $j = (\sigma_\ell, \sigma_{\ell+1}, ..., \sigma_\mathcal{L})$.
- $i_\ell \oplus j_{\ell+1} \equiv (\sigma_1, \sigma_2, ..., \sigma_\ell, \sigma_{\ell+1}, ..., \sigma_\mathcal{L})$ denotes the concatenation of complementary multi-indices.
- Short-hand notation: $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, ..., \sigma_\mathcal{L})$.

Reshape a tensor to a matrix at bond ℓ:

$$F_{\sigma_1\cdots\sigma_\ell,\sigma_{\ell+1}\cdots\sigma_\mathcal{L}} = F_{i,j}, \quad (11)$$



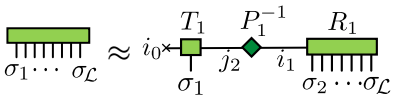
where $i_\ell = (\sigma_1, \sigma_2, ..., \sigma_\ell) \in \mathbb{I}_\ell$ and $j_{\ell+1} = (\sigma_{\ell+1}, ..., \sigma_\mathcal{L}) \in \mathbb{J}_{\ell+1}$.

C.2 TCI formula

A $\mathcal{L}$-leg tensor F can be approximated by a TT with slices of the tensor.

CI at bond 1

$$\quad (12)$$



with

- pivot lists $\mathcal{I}_1 \subseteq \mathbb{I}_1 = \mathbb{S}_1$,
- pivot lists $\mathcal{J}_2 \subseteq \mathbb{J}_2 = \mathbb{S}_2 \times \cdots \times \mathbb{S}_\mathcal{L}$,
- indices for pivots $i_1 \in \mathcal{I}_1$ and $j_2 \in \mathcal{J}_2$,
- dummy index i_0.

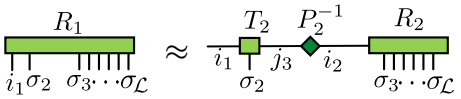
Slices of the tensor:

$$\begin{aligned} [T_1]_{i_0\sigma_1 j_2} &\equiv F_{\sigma_1 \oplus j_2}, \\ [P_1]_{i_1 j_2} &\equiv F_{i_1 \oplus j_2}, \\ [R_1]_{i_1(\sigma_2\cdots\sigma_\mathcal{L})} &\equiv F_{i_1 \oplus (\sigma_2\cdots\sigma_\mathcal{L})}. \end{aligned} \quad (13)$$

CI at bond 2

Reshape R_1 to a matrix and apply CI:

$$\quad (14)$$



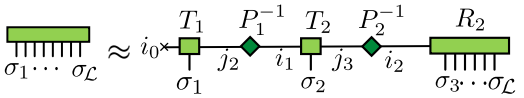
with

$$
\begin{aligned}
[T_2]_{i_1\sigma_2 j_3} &\equiv F_{i_1\oplus\sigma_2\oplus j_3},\\
[P_2]_{i_2 j_3} &\equiv F_{i_2\oplus j_3},\\
[R_2]_{i_2(\sigma_3\cdots\sigma_{\mathcal{L}})} &\equiv F_{i_2\oplus(\sigma_3\cdots\sigma_{\mathcal{L}})}.
\end{aligned}
\tag{15}
$$

and

- pivot lists $\mathcal{I}_2 \subseteq \mathcal{I}_1 \times \mathbb{S}_2$ (left nested),
- pivot lists $\mathcal{J}_3 \subseteq \mathbb{J}_3 = \mathbb{S}_3 \times \cdots \times \mathbb{S}_{\mathcal{L}}$,
- $i_2 \in \mathcal{I}_2$, $j_3 \in \mathcal{J}_3$.

By substituting Equation 14 into Equation 12, we obtain

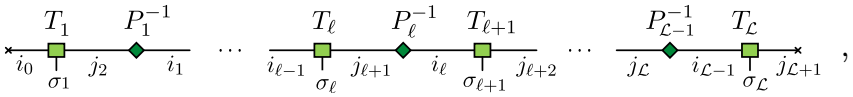


. (16)

TCI formula

By repeating the same procedure to the right end,

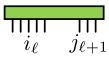
$$
\begin{aligned}
F_{\sigma_1\cdots\sigma_{\mathcal{L}}} &\approx \tilde{F}_{\sigma_1\cdots\sigma_{\mathcal{L}}} = T_1^{\sigma_1} P_1^{-1} T_2^{\sigma_2} P_2^{-1} \ldots P_{\mathcal{L}-1}^{-1} T_{\mathcal{L}}^{\sigma_{\mathcal{L}}},\\
\sigma_1\cdots\sigma_{\mathcal{L}} &\approx \tilde{F}_{\sigma_1\cdots\sigma_{\mathcal{L}}} =
\end{aligned}
\tag{17}
$$



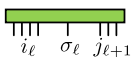
,

where $\mathcal{I}_0 = \mathcal{J}_{\ell+1} = \{()\}$ and the slices of the tensors:

$$
\begin{aligned}
(P_\ell)_{i_\ell j_{\ell+1}} &= F_{i_\ell, j_{\ell+1}} =
\end{aligned}
\tag{18}
$$



$$
[T_\ell]_{i_\ell\sigma_\ell j_{\ell+1}} = F_{i_\ell\oplus\sigma_\ell\oplus j_{\ell+1}} =
$$



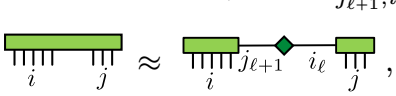
(one-dimensional slice for σ_ℓ).

By construction, $\mathcal{I}_2 \subseteq \mathcal{I}_1 \times \mathbb{S}_2$, $\mathcal{I}_3 \subseteq \mathcal{I}_2 \times \mathbb{S}_3$, $\cdots$, $\mathcal{I}_{\mathcal{L}-1} \subseteq \mathcal{I}_{\mathcal{L}-2} \times \mathbb{S}_{\mathcal{L}-1}$ (*left nested up to* $\mathcal{L}-1$, see Section C.3).

We can construct pivot lists in an arbitrary way. For any pivot lists, Equation 17 yields a TT as long as the pivot matrices P_ℓ are invertible. Interpolation properties and accuracy of the approximation depend on the choice of the pivot lists.

Comparison with matrix CI

$$
\begin{aligned}
F_{ij} &\approx C_{ij_{\ell+1}} \left(P_\ell^{-1}\right)_{j_{\ell+1}, i_\ell} R_{i_\ell j},\\
&\approx
\end{aligned}
\tag{19}
$$



,

where $i \in \mathbb{I}_\ell$, $j \in \mathbb{J}_\ell$, the pivot lists $\mathcal{I}_\ell \subseteq \mathbb{I}_\ell$ and $\mathcal{J}_{\ell+1} \subseteq \mathbb{J}_{\ell+1}$:

$$
(P_\ell)_{i_\ell j_{\ell+1}} = F_{i_\ell, j_{\ell+1}}, C_{ij_{\ell+1}} = F_{i\oplus j_{\ell+1}}, R_{i_\ell, j} = F_{i_\ell\oplus j}. \tag{20}
$$

C.3 Nesting conditions and interpolation properties

- Left nested up to ℓ: $\mathcal{I}_0 < \mathcal{I}_1 < \cdots < \mathcal{I}_\ell$.
- Right nested up to ℓ: $\mathcal{J}_\ell > \mathcal{J}_{\ell+1} > \cdots > \mathcal{J}_{\mathcal{L}+1}$.
- Fully nested: $\mathcal{I}_0 < \mathcal{I}_1 < \cdots < \mathcal{I}_{\mathcal{L}-1}$ and $\mathcal{J}_2 > \mathcal{J}_3 > \cdots > \mathcal{J}_{\mathcal{L}+1}$.

If the pivots are left-nested up to $\ell-1$ and right-nested up to $\ell+1$ (nested w.r.t. T_ℓ),

$$\tilde{F}_{i\oplus\sigma\oplus j} = [T_\ell]_{i\sigma j} = F_{i\oplus\sigma\oplus j} \quad \forall i \in \mathcal{J}_{\ell-1}, \sigma \in \mathbb{S}_\ell, j \in \mathcal{J}_{\ell+1}. \tag{21}$$

Pivot-wise sufficient condition

For a given element $[T_\ell]_{i\sigma j}$, the TCI formula interpolates it perfectly if all the following conditions are satisfied:

- $\forall \ell' \leq \ell$, $\boldsymbol{\sigma}_{1:\ell'} \in \mathcal{I}_{\ell'}$,
- $\forall \ell' \geq \ell+1$, $\boldsymbol{\sigma}_{\ell':\mathcal{L}} \in \mathcal{J}_{\ell'}$,

where $\boldsymbol{\sigma} = i \oplus \sigma \oplus j$ and $\boldsymbol{\sigma}_{a:b}$ is the subvector of $\boldsymbol{\sigma}$ from a to b. This statement can be proved by extending Appendix A.3 of [1].

C.4 Basic learning algorithms

We show hueristic algorithms to find good pivots, which are used in our implementation of **reset mode** and **full pivot search** described in [1].

Requirements The target tensor can be evaluated at any multi-index.

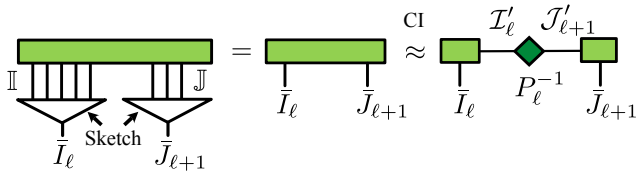
Tasks For a given tolerance ε in the maximum norm sense, find pivot lists at all bonds to approximate the target tensor without reading the whole tensor.

C.4.1 General 2-site update strategy

This idea is not written explicitly in [1] .

We find good pivot lists at all bonds:

1. Find $\hat{\boldsymbol{\sigma}}$ with $F(\hat{\boldsymbol{\sigma}}) \neq 0$, and construct initial pivots: $\mathcal{I}_\ell = \{(\widehat{\sigma_1}, \widehat{\sigma_2}, ..., \widehat{\sigma_\ell})\}$, $\mathcal{J}_{\ell+1} = \{(\widehat{\sigma_{\ell+1}}, \widehat{\sigma_{\ell+2}}, ..., \widehat{\sigma_\mathcal{L}})\}$ for all ℓ.
2. At each ℓ, we update $\mathcal{I}_\ell$ and $\mathcal{J}_{\ell+1}$ by the following steps.
   1. Generate candidate $\overline{\mathcal{I}}_\ell$ and $\overline{\mathcal{J}}_{\ell+1}$. Their sizes are larger than $\mathcal{I}_\ell$ and $\mathcal{J}_{\ell+1}$ by a factor of $O(1)$.
   2. Sketch the unfolded matrix $F(\mathbb{I}_\ell, \mathbb{J}_{\ell+1})$ as $F\big(\overline{\mathcal{I}}_\ell, \overline{\mathcal{J}}_{\ell+1}\big)$.
   3. Decompose the sketched matrix by prrLU to tolerance ε (or a slightly smaller for safety[1]) and select the new pivot lists $\mathcal{I}_{\ell'} \subseteq \overline{\mathcal{I}}_\ell$ and $\mathcal{I}'_{\ell+1} \subseteq \overline{\mathcal{J}}_{\ell+1}$.
   4. Replace $\mathcal{I}_\ell$ and $\mathcal{J}_{\ell+1}$ with the new ones $\mathcal{I}'_\ell$ and $\mathcal{J}'_{\ell+1}$.



3. Iterate the above steps until convergence, i.e., the sizes of the pivot lists stop growing.
4. Construct a TT by evaluating Equation 17, where TP^{-1} or $P^{-1}T$ can be computed by solving linear systems[2].

Remarks:

- In TCI, it is convinient to measure the error in terms of maximum norm.
- The algorithm selects good pivots from the candidate pivot lists and does not necessarily preserved the previous pivot lists (**reset mode** in [1])[3].

[1] This is because the residual error in the matrix CI is not equal to the error of the TCI formula in general. These two errors match if $\mathcal{I}_0 < \mathcal{I}_1 < \cdots < \mathcal{I}_{\ell-1} < \overline{\mathcal{I}}_\ell$ and $\overline{\mathcal{J}}_{\ell+1} > \mathcal{J}_{\ell+2} > \cdots > \overline{\mathcal{J}}_{\mathcal{L}+1}$. Refer to Sec. 4.3.1.

[2] Since the pivot matrices are ill-conditioned, we need to use a numerically stable solver. We observed that prrLU (LU with column and row pivoting) is more robust than QR.

- In the abovementioned algorithm, we select pivots to minimize the residual error of TCI in terms of maximum norm. Alternatively, we can minimize the integration error of TCI (Sec. 4.3.7 of [1]).

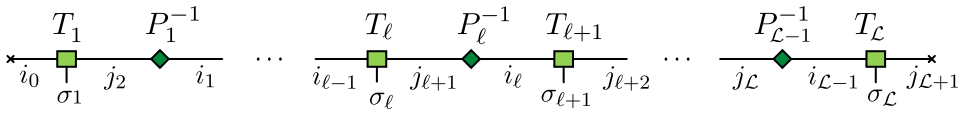
C.4.2 Sweeps with 2-site updates

We perform back and forth sweeps for $\ell = 1, 2, ..., \mathcal{L}-1$. At each ℓ, we construct candidates $\overline{\mathcal{I}}_\ell$ and $\overline{\mathcal{J}}_{\ell+1}$ from pivot lists at neighboring bonds:

- $\overline{\mathcal{I}}_\ell = \mathcal{I}_{\ell-1} \times \mathbb{S}_\ell \cup \mathcal{I}_\ell$,
- $\overline{\mathcal{J}}_{\ell+1} = \mathbb{S}_{\ell+1} \times \mathcal{J}_{\ell+2} \cup \mathcal{J}_{\ell+1}$,

where we ensures that the candidate pivot lists includes the current pivot lists. But, this breaks the nesting conditions: $\mathcal{I}_{\ell-1} < \mathcal{I}_\ell$ and $\mathcal{J}_{\ell+1} > \mathcal{J}_{\ell+2}$ after the update[4].

We explore the full local space of $\mathbb{S}_\ell$ and $\mathbb{S}_{\ell+1}$ in the enviroment of $\mathcal{I}_{\ell-1}$ and $\mathcal{J}_{\ell+2}$:



The size of the candidate pivot lists is approximately $d\tilde{\chi} + \tilde{\chi}$, where d is the dimension of external indices of the tensor and $\tilde{\chi}$ is the number of current pivots.

Then, the new pivot lists are selected by prrLU from the candidate pivot lists. During sweeps, good pivots may propagate between neighboring bonds[5].

Computational cost:
- The number of function evaluations: $O(\mathcal{L}d^2\tilde{\chi}^2)$
- The computational time of prrLU: $O(\mathcal{L}d^2\tilde{\chi}^3)$

In *block rook search* mode (see Sec. 3.3.2 of [1]), the scalings are $O(\mathcal{L}d\tilde{\chi}^2)$ and $O(\mathcal{L}d\tilde{\chi}^3)$, respectively.

C.4.3 Global pivot insertion

Local updates may fail to find good pivots and miss important features of the tensor. A very simple example is a tensor F has nonzero elements only at $\boldsymbol{\sigma} = (0, 0, \cdots, 0)$ and $\boldsymbol{\sigma} = (1, 1, \cdots, 1)$ (length $\mathcal{L} > 2$). If the intial pivots are initialized with $\hat{\boldsymbol{\sigma}} = (0, 0, \cdots, 0)$, the 2-site updates fail to reach $\hat{\boldsymbol{\sigma}} = (1, 1, \cdots, 1)$. This is analogous to ergodicity problems in Markov Monte Carlo methods.

During learning, we can a multi-index $\hat{\boldsymbol{\sigma}}$ (*global pivot*) at all bonds:

$$\begin{aligned} \mathcal{I}'_\ell &= \{(\widehat{\sigma_1}, \widehat{\sigma_2}, ..., \hat{\sigma}_\ell)\} \cup \mathcal{I}_\ell, \\ \mathcal{J}'_{\ell+1} &= \{(\widehat{\sigma_{\ell+1}}, \widehat{\sigma_{\ell+2}}, ..., \widehat{\sigma_\mathcal{L}})\} \cup \mathcal{J}_{\ell+1} \end{aligned} \tag{22}$$

for all ℓ.

[3] An alternative to **reset mode** is **accumulative mode** where we never remove the previous pivot lists and keep adding new pivots to the pivot lists. Accumulative mode requires to read a smaller number of tensor elements. But it is less numerically stable because the pivot matrices are worse conditioned.

[4] In Sec. 4.3 of [1], a variant without including the current pivot lists is mainly discussed. In this case, the slice of the tensor on the candidate pivots is a 4-leg tensor, $\Pi = F(\overline{\mathcal{I}}_\ell, \overline{\mathcal{J}}_{\ell+1})$. But, we observed that including the current pivot lists is more important than preserving the nesting condition.

[5] It is shown that one can updates all bonds $\ell = 1, 2, \cdots, \mathcal{L}-1$ in parallel at one iteration step [8]. In this case, the candidate pivot lists are passed between neighboring bonds during iterations.

Inserting a global pivot extends the exploration of the local space more than the inserted pivot itself. More specifically, we explore the configuration space consisting of the cross product of the new row and existing columns, and the new column and existing rows (like *genetic algorithm*).

After inserting a global pivot, some of the pivot matrices are non-singular or non-invertible, which will be fixed by removing linearly dependent (redundant) pivots by prrLU of these pivot matrices with a machine-precision tolerance (see Sec. 4.4.2 of [1]). Although this may break the nesting conditions, we observed that the TCI formula interpolates the added global pivot perfectly (no math proof yet).

Figure 2 shows the effect of a global pivot insertion for the abovementioned example.

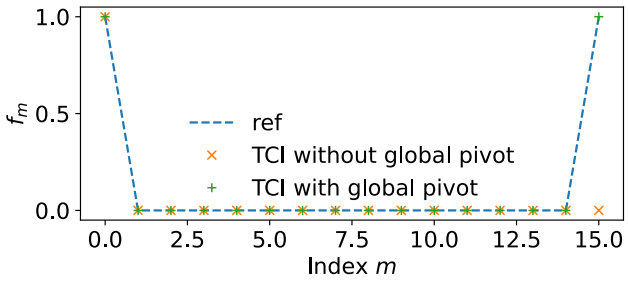


Figure 2: Global pivot insertion. Taken from [1].

We can find such good global pivots by some prior knowledge of the tensor, i.e., its symmetry, or by a global search to find the best pivot with a large interpolation error. Reference [9] showed that the ergodicity problem can be (partially) circumvented by alternating between local and global updates with automatic greedy selection of global pivots.

C.4.4 Error estimates of TCI formula

Two main sources of errors:

1. The algorithm fails to find important features of the tensor.
2. Breaking the nesting conditions does not guarantee the error reported by matrix CIs matches the actual error of the TCI formula (see Sec. 4.3.1 of [1]).

From experience, the first source is more important than the second one. It is high recommended to start with a good initial guess, and mix local and global updates.

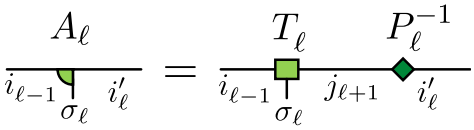
C.5 Canonical forms of TCI

In this section, we discuss the canonical forms of TCI based on nesting conditions, in better connection with the canonical forms of MPS.

Left nested

$\mathcal{I}_{\ell-1} < \mathcal{I}_\ell \Leftrightarrow \mathcal{I}_\ell \subseteq \mathcal{I}_{\ell-1} \times \mathbb{S}_\ell$:

$$A_{i_{\ell-1},i_\ell} = \sum_{j_{\ell+1}} \left[T_\ell^{\sigma_\ell}\right]_{i_{\ell-1},j_{\ell+1}} \left(P_\ell^{-1}\right)_{j_{\ell+1},i'_\ell} \tag{23}$$



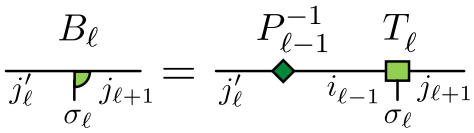
The tensor A_ℓ can be regarded as a *tall* matrix with rows $i_{\ell-1} \oplus \sigma_\ell$ and columns i_ℓ. The projection of A^ℓ to the rows $i_{\ell-1} \oplus \sigma_\ell \in \mathcal{I}_\ell$ is the identity matrix (cf. Equation 6):

$$A_\ell = \begin{pmatrix} T'_\ell \\ T''_\ell \end{pmatrix} (P_\ell)^{-1} = \begin{pmatrix} \mathbb{I} \\ T''_\ell P_\ell^{-1} \end{pmatrix}, \tag{24}$$

where the rows of the first block T'_ℓ satisfy $i_{\ell-1} \oplus \sigma_\ell \in \mathcal{I}_\ell$.

Right nested

$\mathcal{J}_\ell > \mathcal{J}_{\ell+1} \Leftrightarrow \Leftrightarrow \mathcal{J}_\ell \subseteq \mathbb{S}_\ell \times \mathcal{J}_{\ell+1}$:

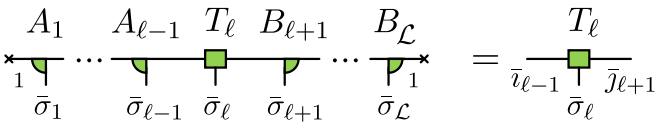


The projection to the columns $j_{\ell'} \oplus \sigma_\ell \in \mathcal{J}_{\ell+1}$ is the identity matrix:

$$B_\ell = \left(P_\ell^{-1}\right)\left(T'_\ell \;\; T''_\ell\right) = \left(\mathbb{I} \;\; P_\ell^{-1} T''_\ell\right). \tag{25}$$

Nested with respect to T_ℓ

A TCI is nested with respect to T_ℓ if $\mathcal{I}_0 < \mathcal{I}_1 < \cdots < \mathcal{I}_\ell$ and $\mathcal{J}_{\ell+1} > \cdots > \mathcal{J}_{\mathcal{L}+1}$:



The TCI formula interpolates the slice T_ℓ (see Equation 21). This is analogous to the mixed canonical form of MPS.

C.6 Other important topics

- Converting a general MPS/TT to a fully nested TCI using prrLU (Sec. 4.5.1 of [1])
- Addition of MPSs, zip-up contraction of MPSs and MPOs using prrLU (Sec. 4.7 of [1])
- A 0-site TCI update of P_ℓ removes redundant pivots by prrLU to make the pivot matrix invertible (Sec. 4.4.2 of [1]). This will be useful after insertion of global pivots.

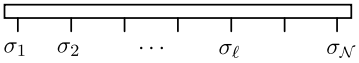
D Applications TCI and quantics representation

D.1 Tensorising functions

D.1.1 Natural representation

Function with $\mathcal{N}$ variables ($\boldsymbol{x} \in \mathcal{R}^{\mathcal{N}}$):

$$(\boldsymbol{x}(\boldsymbol{\sigma})) = f(x_1(\sigma_1), \cdots, x_{\mathcal{L}}(\sigma_{\mathcal{N}})) = \tag{26}$$



Discrete grid:

$$x_1 \to \{x_1(1), \cdots, x_1(d_1)\}, \cdots, x_{\mathcal{N}} \to \{x_{\mathcal{N}}(1), \cdots, x_{\mathcal{N}}(d_{\mathcal{N}})\} \tag{27}$$

Here, d_ℓ is the size of the discretization grid for the ℓ-th variable.

Drawback: If there are coexisting local structures and vastly different length scales, we need to increase d_ℓ to resolve the local structures.

D.1.2 Quantics representation

Effective for coexisting vastly different length scales [10,11].

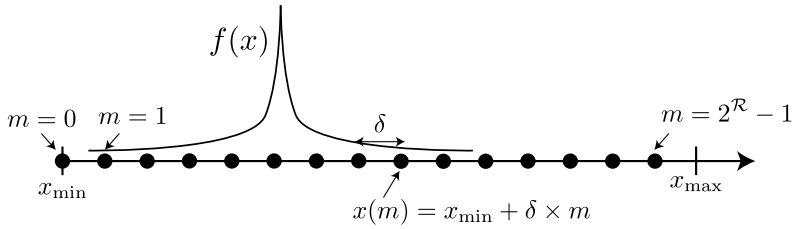
Binary conding

$$i = a_1 \times^{\mathcal{R}-1} + a_2 \times 2^{\mathcal{R}-2} + \cdots + a_{\mathcal{R}} \times 2^0 = (a_1 a_2 \cdots a_{\mathcal{R}})_2, \tag{28}$$

where $a_r \in \{0, 1\}$. The range of the integers that can be represented increases exponentially with the number of bits $\mathcal{R}$.

Quantics representation (one variable)

Discretize a function $f(x), x \in [x_{\min}, x_{\max}]$ on an equally spaced grid (size $M = 2^{\mathcal{R}}$, spacing δ):

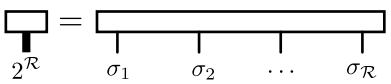


A one-leg tensor of size $M = 2^{\mathcal{R}}$:

$$f(x(m)) = \tag{29}$$

By using $m = (\sigma_1 \cdots \sigma_{\mathcal{R}})_2$, we can represent the function as a $\mathcal{R}$-leg tensor of size $2 \times \cdots \times 2$:

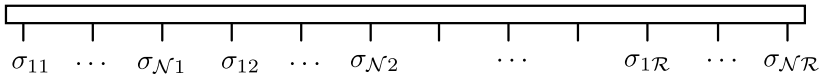
$$f(x(m)) = =$$



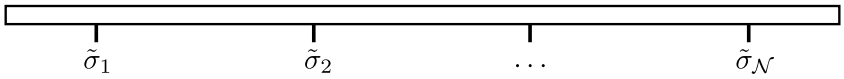
Quantics representation ($\mathcal{N}$ variables)

Use binary representations of the variables ($n = 1, \cdots, \mathcal{N}$): $m_n = (\sigma_{n1} \cdots \sigma_{n\mathcal{R}})_2$:

Interleaved representation:



Fused representation:

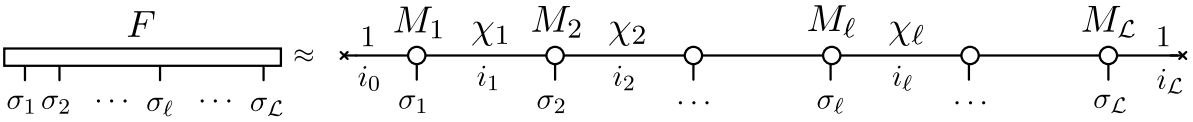


The bond dimension will strongly depend on the order of the indices (see below).

D.2 TT unfolding

Natural representation

Unfold (decompose) a $\mathcal{N}$-variable function into a TT:
$f(x_1, ..., x_\mathcal{N}) \approx M_1(x_1)M_2(x_2)...M_\mathcal{N}(x_\mathcal{N})$,



If the TT is low-rank, we can perform superfast summation:

$$\int d^\mathcal{N}\boldsymbol{x} f(\boldsymbol{x}) \approx \int dx_1 M_1(x_1) \int dx_2 M_2(x_2) ... \int dx_\mathcal{N} M_\mathcal{N}(x_\mathcal{N}). \tag{30}$$

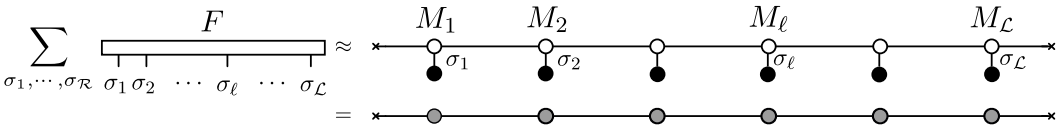
In practice, we use a Gaussian quadrature along each variable, i.e., discretize each variable into d_ℓ Gauss-Legendre nodes.

Quantics representation

TT unfolding corresponds to the separation across different scales.:

Integration by Riemann sum:

$$\int dx f(x) \approx 2^{-\mathcal{R}} \left(\sum_{\sigma_1} M(\sigma_1) \right) \cdots \left(\sum_{\sigma_\mathcal{R}} M(\sigma_\mathcal{R}) \right) + O(2^{-2\mathcal{R}}) \tag{31}$$



The block filled circles are vectors $\left(\frac{1}{2}, \frac{1}{2}\right)$. Linear computational cost with respect to $\mathcal{R}$: $O(\chi^2\mathcal{R})$, and exponentially small discretization error.

Good cases for quantics representation:

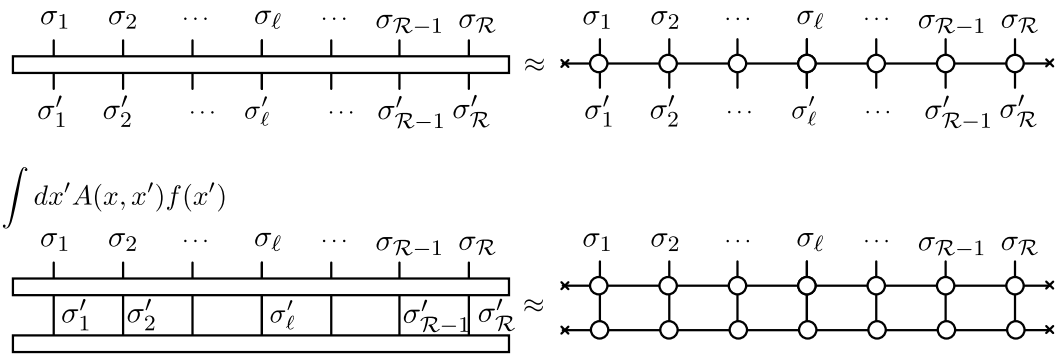
- Polynomials and exponential functions: $e^{ax} = e^{a\sum_{\ell=1}^{\mathcal{R}} \sigma_\ell 2^{-\ell}} = \prod_{\ell=1}^{\mathcal{R}} e^{a2^{-\ell}}$.
- Identity matrix: $\delta_{xy} = \delta_{\sigma_1\sigma_1'} ... \delta_{\sigma_\mathcal{R}\sigma_\mathcal{R}'}$.

Bad cases for quantics representation:

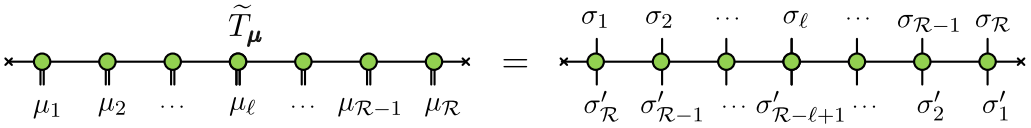
- $f(x, y) = 1$ if $x^2 + y^2 \leq 1$ otherwise 0. The bond dimension brows exponentially with $\mathcal{R}$.

In general, discontinuity or sharp features on a hyperplace is OK, but not on a hypercurve.

Convolution with tensor Train Operator (TTO):



Quantum Fourier Transform (QFT) MPO has a small entanglement entropy [12,13]:



σ_ℓ and σ'_ℓ are bits for the variables before and after the QFT, respectively. The order of the indices is reversed after the QFT. The bond dimension of the QFT MPO is up to 10–20 even for machine precision irrespectively of $\mathcal{R}$.

Exercise: QTT representation of exponential function

Show that $f(x) = e^x$ for $x \in [0, 1]$ has a QTT representation with bond dimension 1.

Exercise: QTT representation of identity matrix

Find the QTT representation of $f(x, y) = \delta_{xy}$ with bond dimension 1, where $x, y = 0, 1, \cdots, 2^{\mathcal{R}} - 1$.

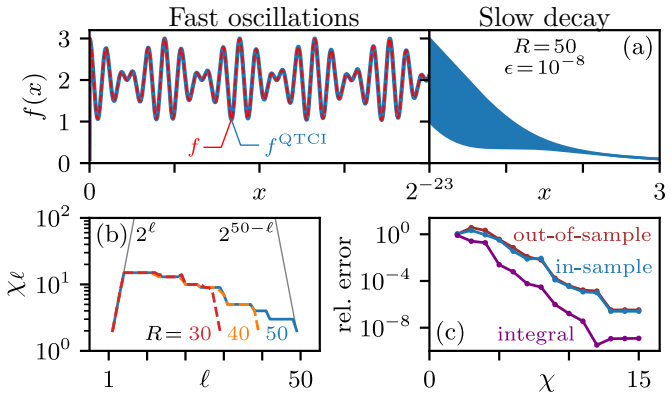
D.3 Combination of quantics and TCI (QTCI)

From [14].

Highly oscillatory function with slowly decaying envelope:

$$f(x) = \cos\left(\frac{x}{B}\right)\cos\left(\frac{x}{4\sqrt{5}B}\right)e^{-x^2} + 2e^{-x} \quad (B = 2^{-30}),$$
$$\int_0^{\ln(20)} dx f(x) \approx \frac{19}{10}. \tag{32}$$

The following is the results of QTCI with 8706 samples (1 sample per 59,000 oscillations):


Fast oscillations
Slow decay
$f(x)$
$R=50$
$\epsilon=10^{-8}$
(a)
f
f^{QTCI}
x
2^{-23}
(b)
2^{ℓ}
$2^{50-\ell}$
χ_ℓ
$R=30$ 40 50
ℓ
rel. error
out-of-sample
in-sample
(c)
integral
χ

E Future research directions

- Highly parallelizable algorithms for TCI
  - Integration
  - MPO-MPO contraction
- Extension to tree tensor networks
- Combination with neural networks

Bibliography

[1] Y. N. Fernández et al., Learning tensor networks with tensor cross interpolation: new algorithms and libraries, Arxiv Preprint Arxiv:2407.02454 (2024)

[2] S. Goreinov and E. Tyrtyshnikov, The maximal-volume concept in approximation by low-rank matrices, 47 (2001)

[3] Y. N. Fernández, M. Jeannin, P. T. Dumitrescu, T. Kloss, J. Kaye, O. Parcollet, and X. Waintal, Learning feynman diagrams with tensor trains, Physical Review X **12**, 41018 (2022)

[4] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, How to find a good submatrix, Matrix Methods: Theory, Algorithms and Applications 247 (2010)

[5] M. Fornace and M. Lindsey, Column and row subset selection using nuclear scores: algorithms and theory for Nyström approximation, CUR decomposition, and graph Laplacian reduction, (2024)

[6] I. Oseledets and E. Tyrtyshnikov, TT-cross approximation for multidimensional arrays, Linear Algebra Appl. **432**, 70 (2010)

[7] I. V. Oseledets, Tensor-Train Decomposition, SIAM J. Sci. Comput. **33**, 2295 (2011)

[8] S. Dolgov and D. Savostyanov, Parallel cross interpolation for high-precision calculation of high-dimensional integrals, Comput. Phys. Commun. **246**, 106869 (2020)

[9] H. Ishida, N. Okada, S. Hoshino, and H. Shinaoka, Low-rank quantics tensor train representations of Feynman diagrams for multiorbital electron-phonon models, (2024)

[10] I. Oseledets, *Approximation of Matrices with Logarithmic Number of Parameters*, in *Doklady Mathematics*, Vol. 80 (2009), pp. 653–654

[11] B. N. Khoromskij, O (d log N)-quantics approximation of N-d tensors in high-dimensional numerical modeling, Constructive Approximation **34**, 257 (2011)

[12] J. Chen, E. M. Stoudenmire, and S. R. White, Quantum Fourier Transform Has Small Entanglement, PRX Quantum **4**, 40318 (2023)

[13] H. Shinaoka, M. Wallerberger, Y. Murakami, K. Nogaki, R. Sakurai, P. Werner, and A. Kauch, Multiscale Space-Time Ansatz for Correlation Functions of Quantum Systems Based on Quantics Tensor Trains, Physical Review X **13**, 21015 (2023)

[14] M. K. Ritter, Y. N. Fernández, M. Wallerberger, J. von Delft, H. Shinaoka, and X. Waintal, Quantics Tensor Cross Interpolation for High-Resolution Parsimonious Representations of Multivariate Functions, Physical Review Letters **132**, 56501 (2024)

```julia
using LinearAlgebra, Combinatorics
import TensorCrossInterpolation as TCI # for prrLU

# Function to create a Hilbert matrix
function hilbert_matrix(N)
    return [1 / (i + j - 1) for i in 1:N, j in 1:N]
end

# Function to compute the residual matrix using CI approximation
function ci_residual(A, I, J)
    tildechi = length(I)
    if tildechi != length(J)
        error("Pivot lists I and J must have the same length")
    end

    # Select pivot submatrix
    A_IJ = A[I, J]  # Pivot submatrix (tildechi×tildechi)
    C = A[:, J]     # (N×tildechi)
    R = A[I, :]     # (tildechi×N)

    # Compute determinant of A_IJ
    det_A_IJ = abs(det(A_IJ))

    # Solve the linear system instead of computing the inverse
    A_CI = C * (A_IJ \ R)  # Using backslash operator for solving linear equations

    # Residual matrix
    Residual = A - A_CI
    residual_norm = norm(Residual, Inf)

    return Residual, residual_norm, det_A_IJ
end

# Example execution: N=5, tildechi=3
N = 10
tildechi = 4

A = hilbert_matrix(N)

prrlu_result = TCI.rrlu(A; maxrank=tildechi)

@assert prrlu_result.npivot == tildechi
I_lu = prrlu_result.rowpermutation[1:tildechi]
J_lu = prrlu_result.colpermutation[1:tildechi]

residual_lu = ci_residual(A, I_lu, J_lu)[2]
```

Listing 1: Code to compute the residual of prrLU for $N = 10$ and $\tilde{\chi} = 4$ using prrLU.