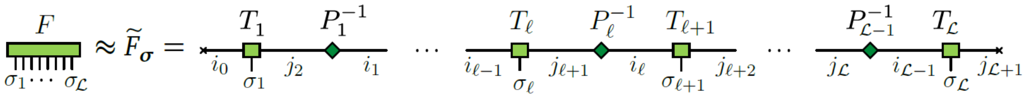
Goal: unfold given tensor to TT via repeated CI factorizations (either CI or prrLU) into 'TCI form':



(1)

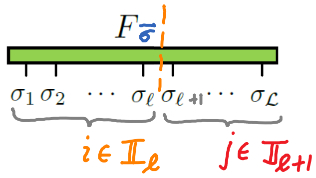
Defining characteristics of this decomposition: it is built only from one-dimensional slices of $F_{\vec{\sigma}}$

TCI algorithms use only local updates of these slices. (on which all tensor indices σ_ℓ but one are fixed)

For each CI factorization, say along bond ℓ , the tensor is viewed as a matrix, with

row index $i = (\sigma_1, \ldots, \sigma_\ell)$ with dimension(i) = d^ℓ

column index $j = (\sigma_{\ell+1}, \ldots, \sigma_{\mathcal{L}})$ with dimension(j) = $d^{\mathcal{L}-\ell}$ (2)

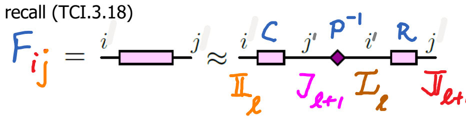


$\mathcal{I}_\ell$

CI factorization $F_{ij} \simeq \widetilde{F}_{ij} = [C_\ell]_{ij'} [P_\ell^{-1}]_{j'i'} [R_\ell]_{i'j}$

with pivot matrix $P_\ell = F(\mathcal{I}_\ell, \mathcal{J}_{\ell+1})$

recall (TCI.3.18)



(3)

external indices i, j are fixed,
internal bonds represent sums over pivot lists: $\sum_{j' \in \mathcal{J}_{\ell+1}}$ $\sum_{i' \in \mathcal{I}_\ell}$

column-pivot list row-pivot list

Note: all ingredients carry bond index ℓ.

Dimensions of i, j are generically large, 'top-down' CI factorization is impractical. Instead, use 'bottom-up' approach: start 'TCI Ansatz' (1) with small pivot matrices, increase their size via 'sweeps' until desired tolerance is reached.

To be systematic, we introduce some bookkeeping conventions:

External index σ_ℓ ($\ell \in \{1, 2, \ldots, \mathcal{L}\}$) takes d_ℓ different values from set $\mathbb{S}_\ell$ (4)

$\mathbb{I}_\ell = \mathbb{S}_1 \times \ldots \times \mathbb{S}_\ell$ = set of row multi-indices up to site ℓ ; $i \in \mathbb{I}_\ell$ has the form $i = (\sigma_1, \ldots, \sigma_\ell)$ (5a)

row multi-index = list of indices

$\mathbb{J}_{\ell+1} = \mathbb{S}_{\ell+1} \times \ldots \times \mathbb{S}_{\mathcal{L}}$ = set of column multi-indices from site ℓ upwards; $j \in \mathbb{J}_{\ell+1}$ has the form $j = (\sigma_{\ell+1}, \ldots, \sigma_{\mathcal{L}})$ (5b)

column multi-index = list of indices

$\mathbb{I}_{\mathcal{L}} = \mathbb{J}_1$ = full configuration space. A full configuration $\vec{\sigma} \in \mathbb{I}_{\mathcal{L}}$ takes the form $\vec{\sigma} = (\sigma_1, \ldots, \sigma_{\mathcal{L}})$ (6)

$$i_\ell \oplus j_{\ell+1} = (\underbrace{\sigma_1, \ldots, \sigma_\ell}_{i_\ell}, \underbrace{\sigma_{\ell+1}, \ldots, \sigma_{\mathcal{L}}}_{j_{\ell+1}})$$ = concatenation of complementary multi-indices. (7)

For each $\ell = 1, \ldots, \mathcal{L}-1$, we define a list of 'pivot rows' $\mathcal{I}_\ell \subseteq \mathbb{I}_\ell$ and list of 'pivot columns' $\mathcal{J}_{\ell+1} \subseteq \mathbb{J}_{\ell+1}$.

Also: $\mathcal{I}_0 = \mathcal{J}_{\mathcal{L}+1} = \{()\}$, with ( ) = empty tuple. $\mathcal{I}_\ell$ and $\mathcal{J}_{\ell+1}$ are lists of lists: e.g. , for $\mathcal{L} = 5$, $\mathbb{S} = \{0, 1\}$

ℓ	$\mathcal{I}_\ell$	$\mathcal{J}_{\ell+1}$
1	$\mathcal{I}_1 = ((1))$	$\mathcal{J}_2 = ((1,0,0,1))$
2	$\mathcal{I}_2 = ((1,0),(1,1))$	$\mathcal{J}_3 = ((0,0,1),(1,0,1))$
3	$\mathcal{I}_3 = ((1,1,0),(1,0,1))$	$\mathcal{J}_4 = ((0,1),(1,1))$
4	$\mathcal{I}_4 = ((1,1,0,0))$	$\mathcal{J}_5 = ((1))$

$\vec{\sigma} = i_\ell \oplus j_{\ell+1}$ configurations defined by pivot lists on the left

$(1,1,0,0,1)$

$(1,0,0,0,1), (1,0,1,0,1)$ $(1,1,0,0,1), (1,1,1,0,1)$

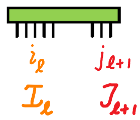
$(1,1,0,0,1), (1,1,0,1,1), (1,0,1,0,1), (1,0,1,1,1)$

$(1,1,0,0,1)$

Now define zero-, one-, and two-dimensional slices of input tensor F:

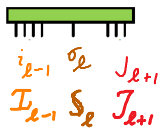
(k-dimensional slice has k free indices)

- 'pivot matrix' P_ℓ (zero-dimensional slice):

$P_\ell = F(\mathcal{I}_\ell, \mathcal{J}_{\ell+1})$ with elements $[P_\ell]_{i_\ell j_{\ell+1}} = F_{i_\ell \oplus j_{\ell+1}} =$  (9)

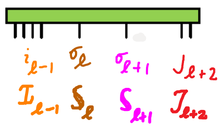
square matrix of dimension $\chi_\ell = |\mathcal{I}_\ell| = |\mathcal{J}_{\ell+1}|$

- 3-leg tensor T_ℓ (one-dimensional slice, with free index σ_ℓ):

$T_\ell = F(\mathcal{I}_{\ell-1}, \mathcal{S}_\ell, \mathcal{J}_{\ell+1})$ with elements $[T_\ell]_{i_{\ell-1}\sigma_\ell j_{\ell+1}} = F_{i_\ell \oplus (\sigma_\ell) \oplus j_{\ell+1}} =$  (10)

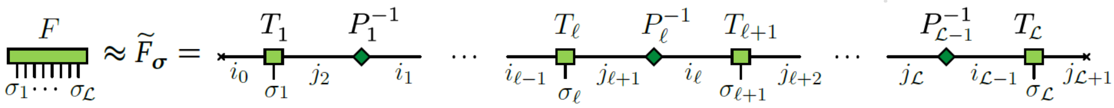
For fixed σ_ℓ, we define the matrix $[T_\ell^{\sigma_\ell}]_{i_{\ell-1} j_{\ell+1}} = [T_\ell]_{i_{\ell-1}\sigma_\ell j_{\ell+1}}$

- 4-leg tensor Π_ℓ (two-dimensional slice, with free indices σ_ℓ, $\sigma_{\ell+1}$):

$T_\ell = F(\mathcal{I}_{\ell-1}, \mathcal{S}_\ell, \mathcal{S}_{\ell+1}, \mathcal{J}_{\ell+2})$ with elements $[\Pi_\ell]_{i_{\ell-1}\sigma_\ell \sigma_{\ell+1} j_{\ell+2}} = F_{i_\ell \oplus (\sigma_\ell) \oplus (\sigma_{\ell+1}) \oplus j_{\ell+1}} =$  (11)

With these definitions, the 'TCI approximation' $\widetilde{F}$ of F is defined as

$$F_{\boldsymbol{\sigma}} \approx \widetilde{F}_{\boldsymbol{\sigma}} = T_1^{\sigma_1} P_1^{-1} \cdots T_\ell^{\sigma_\ell} P_\ell^{-1} T_{\ell+1}^{\sigma_{\ell+1}} \cdots P_{\mathcal{L}-1}^{-1} T_{\mathcal{L}}^{\sigma_{\mathcal{L}}}, \quad (12a)$$

 (12b)

with independent sums over all row multi-indices $\sum_{i_\ell \in \mathcal{I}_\ell}$ and column multi-indices $\sum_{j_{\ell+1} \in \mathcal{J}_{\ell+1}}$, for $\ell = 1, \dots, \mathcal{L}-1$

(12) defines the 'TCI form'. It is fully defined by T_ℓ and P_ℓ tensors, i.e. by slices of F. These can be constructed if

(i) one knows the pivot lists $\{\mathcal{I}_\ell, \mathcal{J}_{\ell+1} | \ell = 1, \dots, \mathcal{L}-1\}$ and

(ii) can read out / evaluate / compute the input tensor $F_{\vec{\sigma}}$ for any configuration $\vec{\sigma}$

Any tensor train can be converted <u>exactly</u> into a TCI form (see TCI.).

8. Nesting conditions

Fernandez2025, Sec. 4.2]

TCI.8

The interpolation properties of TCI Ansatz rely on nesting conditions satisfied by its pivot lists.
Below, we define these nesting conditions. Their relevance will become clear in subsequent sections.

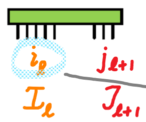
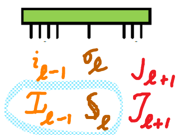
For any bond ℓ

$\mathcal{I}_\ell$ left-nested row-pivot lists

$\mathcal{I}_1 = ((1))$
$\mathcal{I}_2 = ((1,\cancel{0}),(1,\cancel{1}))$
$\mathcal{I}_3 = ((1,1,\cancel{0}),(1,0,\cancel{1}))$
$\mathcal{I}_4 = ((1,1,0,\cancel{0}))$

- $\mathcal{I}_\ell$ is nested w.r.t. to $\mathcal{I}_{\ell-1}$, denoted by $\mathcal{I}_{\ell-1} < \mathcal{I}_\ell$ if $\mathcal{I}_\ell \subseteq \mathcal{I}_{\ell-1} \times \mathbb{S}_\ell$ (1)

  Then, removing <u>last</u> index of any element of $\mathcal{I}_\ell$ yields element of $\mathcal{I}_{\ell-1}$,
  i.e. $\mathcal{I}_\ell$ 'descends from' $\mathcal{I}_{\ell-1} \times \mathbb{S}_\ell$.

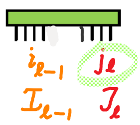
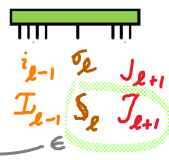
  Moreover, then P_ℓ is a slice of T_ℓ: $[P_\ell]_{i_\ell j_{\ell+1}} =$  , $[T_\ell]_{i_{\ell-1}\sigma_\ell j_{\ell+1}} =$ 

$\mathcal{J}_{\ell+1}$ right-nested column-pivot lists

$\mathcal{J}_2 = ((1,0,0,1))$
$\mathcal{J}_3 = ((0,0,1),(1,0,1))$
$\mathcal{J}_4 = ((0,1),(1,1))$
$\mathcal{J}_5 = ((1))$

- $\mathcal{J}_\ell$ is nested w.r.t. to $\mathcal{J}_{\ell+1}$, denoted by $\mathcal{J}_\ell > \mathcal{J}_{\ell+1}$ if $\mathcal{J}_\ell \subseteq \mathbb{S}_\ell \times \mathcal{J}_{\ell+1}$ (2)

  Then, removing <u>first</u> index of any element of $\mathcal{J}_\ell$ yields element of $\mathcal{J}_{\ell+1}$,
  i.e. $\mathcal{J}_\ell$ 'descends from' $\mathbb{S}_\ell \times \mathcal{J}_{\ell+1}$

  Moreover, then $P_{\ell-1}$ is a slice of T_ℓ: $[P_\ell]_{i_{\ell-1} j_\ell} =$  , $[T_\ell]_{i_{\ell-1}\sigma_\ell j_{\ell+1}} =$ 

- Pivots are 'left-nested' up to ℓ if $\quad \mathcal{I}_0 < \mathcal{I}_1 < \cdots < \mathcal{I}_\ell$ (3)
- Pivots are 'right-nested' up to ℓ if $\quad \mathcal{J}_\ell > \mathcal{J}_{\ell+1} > \cdots > \mathcal{J}_{\mathcal{L}+1}$ (4)
- Pivots are 'fully left-nested' if they are left-nested up to $\mathcal{L}-1$ $\quad \mathcal{I}_0 < \mathcal{I}_1 < \cdots < \mathcal{I}_{\mathcal{L}-1}$, (5)
- Pivots are 'fully right-nested' if they are right-nested up to 2 $\quad \mathcal{J}_2 > \mathcal{J}_{\ell+2} > \cdots > \mathcal{J}_{\mathcal{L}+1}$ (6)
- Pivots are 'fully nested' if they are fully left- and right-nested. (7)

- Pivots are nested w.r.t. T_ℓ if they are left-nested up to $\ell-1$ and right-nested up to $\ell+1$

  Then, the TCI form is exact on the one-dimensional slice T_ℓ

  $$\tilde{F}_{i_\ell \oplus (\sigma_\ell) \oplus j_{\ell+1}} \simeq [T_\ell]_{i_{\ell-1}\sigma_\ell j_{\ell+1}} = F_{i_\ell \oplus (\sigma_\ell) \oplus j_{\ell+1}}$$

  $\mathcal{I}_0 < .. < \mathcal{I}_{\ell-1}\ \mathbb{S}_\ell\ \mathcal{J}_{\ell+1} > \ \ > \mathcal{J}_{\mathcal{L}+1}$ (8)

- Pivots are nested w.r.t. Π_ℓ if they are left-nested up to $\ell-1$ and right-nested up to $\ell+2$ (9)

  $\mathcal{I}_0 < .. < \mathcal{I}_{\ell-1}\ \mathbb{S}_\ell\ \mathbb{S}_{\ell+1}\ \mathcal{J}_{\ell+2} > \ \ > \mathcal{J}_{\mathcal{L}+1}$

- If pivots are <u>fully</u> nested, TCI form is exact on every T_ℓ and P_ℓ, i.e. on all slices used to construct it, thus it is an interpolation. (10)

Properties (8) and (10) are very important. For a proof, see TCI.10.

9. TCI unfolding algorithms (2-, 1-, 0-site) [Fernandez2025, Sec. 4.3-4]

TCI.9

Goal: obtain TCI approximation $\tilde{F}$ of given tensor F at specified tolerance, $\|F - \tilde{F}\|_\infty < \tau$, (1)
by finding a minimal set of suitable pivots.

maximum norm = |largest element|

Basic 2-site TCI algorithm [Fernandez2025, Sec. 4.3.1]

(1) Initialization: start with any configuration $\hat{\sigma}$
for which $F_{\hat{\sigma}} \neq 0$ and construct initial pivot lists from it:

$$\mathcal{I}_\ell = \{(\hat{\sigma}_1, \dots, \hat{\sigma}_\ell)\}, \quad \mathcal{J}_{\ell+1} = \{(\hat{\sigma}_{\ell+1}, \dots, \hat{\sigma}_L)\}$$

Example: $\hat{\sigma} = (1, 0, 1, 1)$

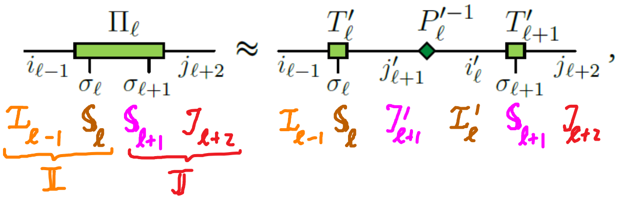
ℓ	$\mathcal{I}_\ell$	$\mathcal{J}_{\ell+1}$
0	$\mathcal{I}_0 = \{()\}$	$\mathcal{J}_1 = \{(1,0,1,1)\}$
1	$\mathcal{I}_1 = \{(1)\}$	$\mathcal{J}_2 = \{(0,1,1)\}$
2	$\mathcal{I}_2 = \{(1,0)\}$	$\mathcal{J}_3 = \{(1,1)\}$
3	$\mathcal{I}_3 = \{(1,0,1)\}$	$\mathcal{J}_4 = \{(1)\}$
4	$\mathcal{I}_4 = \{(1,0,1,1)\}$	$\mathcal{J}_5 = \{()\}$

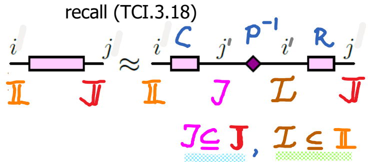
(2)

(2) Sweeping back and forth over $\ell = 1, \dots, L-1$, perform the following update at each ℓ :

Construct $\Pi_\ell \equiv F(\mathcal{I}_{\ell-1}, \mathbb{S}_\ell, \mathbb{S}_{\ell+1}, \mathcal{J}_{\ell+2})$, view it as a matrix $F(\underbrace{\mathcal{I}_{\ell-1} \times \mathbb{S}_\ell}_{\mathbb{I}}, \underbrace{\mathbb{S}_{\ell+1} \times \mathcal{J}_{\ell+2}}_{\mathbb{J}})$, prrLU-factorize it,
and use new pivot lists $\mathcal{I}'_\ell$, $\mathcal{J}'_{\ell+1}$ to update T_ℓ, P_ℓ, $T_{\ell+1}$.

$$[\Pi_\ell]_{i_{\ell-1}\sigma_\ell\sigma_{\ell+1}j_{\ell+2}} \approx [T_\ell'^{\sigma_\ell}]_{i_{\ell-1}j'_{\ell+1}} (P'_\ell)^{-1}_{j'_{\ell+1}i'_\ell} [T_{\ell+1}'^{\sigma_{\ell+1}}]_{i'_\ell j_{\ell+2}} = \tilde{\Pi}_\ell \qquad (3)$$





row indices: $\mathcal{I}'_\ell \subseteq \mathcal{I}_{\ell-1} \times \mathbb{S}_\ell = \mathbb{I}$ column indices: $\mathcal{J}'_{\ell+1} \subseteq \mathbb{S}_{\ell+1} \times \mathcal{J}_{\ell+2} = \mathbb{J}$ (4a)

nested by construction: $\mathcal{I}_{\ell-1} < \mathcal{I}'_\ell$ nested by construction: $\mathcal{J}'_{\ell+1} > \mathcal{J}_{\ell+2}$ (4b)

If old pivots are discarded, i.e. if $\mathcal{I}_\ell \not\subseteq \mathcal{I}'_\ell$ or $\mathcal{J}_{\ell+1} \not\subseteq \mathcal{J}'_{\ell+1}$, that may break previously existing nesting conditions:

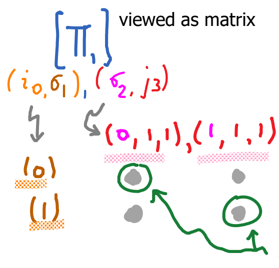
even if $\mathcal{I}_\ell < \mathcal{I}_{\ell+1}$ it may happen that $\mathcal{I}'_\ell \not< \mathcal{I}_{\ell+1}$, and even if $\mathcal{J}_\ell > \mathcal{J}_{\ell+1}$ it may happen that $\mathcal{J}_\ell \not> \mathcal{J}'_{\ell+1}$ (5)

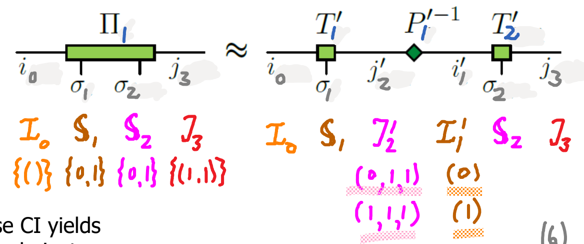
Remedy: if full nesting is desired, it can be restored at the end using 1-site TCI algorithm (see TCI.11).

(3) Iterate step (2) until specified tolerance or specified maximum bond dimension is reached.

Example: for $\Pi_{\ell=1}$, with pivot lists from (2):

$\mathcal{I}_0 = \{()\}$	$\mathcal{J}_1 = \{(1,0,1,1)\}$
$\mathcal{I}_1 = \{(1)\}$	$\mathcal{J}_2 = \{(0,1,1)\}$
$\mathcal{I}'_1 = \{(0),(1)\}$	$\mathcal{J}'_2 = \{(0,1,1),(1,1,1)\}$
$\mathcal{I}_2 = \{(1,0)\}$	$\mathcal{J}_3 = \{(1,1)\}$
$\mathcal{I}_3 = \{(1,0,1)\}$	$\mathcal{J}_4 = \{(1)\}$
$\mathcal{I}_4 = \{(1,0,1,1)\}$	$\mathcal{J}_5 = \{()\}$





(6)

Nesting properties: $\mathcal{I}_0 < \mathcal{I}'_1 < \mathcal{I}_2$, $\mathcal{J}_1 > \mathcal{J}'_2 > \mathcal{J}_3$ (7)

by construction; because no row pivots were discarded; by construction; because no column pivots were discarded

Next, for $\Pi_{\ell=2}$, with pivot lists from (6):

$\mathcal{I}_0 = \{()\}$ $\quad \mathcal{J}_1 = \{(1,0,1,1)\}$

$\mathcal{I}_1 = \{(0),(1)\}$ $\quad \mathcal{J}_2 = \{(0,1,1),(1,1,1)\}$

$\mathcal{I}_2 = \{(1,0)\}$ $\quad \mathcal{J}_3 = \{(1,1)\}$

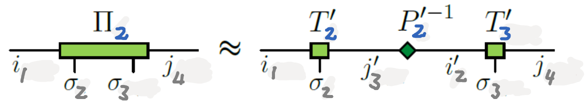
$\mathcal{I}'_2 = \{(0,0),(0,1)\}$ $\quad \mathcal{J}'_3 = \{(0,1),(1,1)\}$

$\mathcal{I}_3 = \{(1,0,1)\}$ $\quad \mathcal{J}_4 = \{(1)\}$

$\mathcal{I}_4 = \{(1,0,1,1)\}$ $\quad \mathcal{J}_5 = \{()\}$

$[\Pi_1]$ viewed as matrix

$$\begin{array}{c|cc} (i_1,\sigma_2)\downarrow \;\; (\sigma_3,j_4)\rightarrow & (0,1) & (1,1) \\ \hline (0,0) & \bullet & \bullet \\ (0,1) & \bullet & \bullet \\ (1,0) & \bullet & \bullet \\ (1,1) & \bullet & \bullet \end{array}$$



$\mathcal{I}_1 \quad \mathcal{S}_2 \quad \mathcal{S}_3 \quad \mathcal{J}_4$ $\qquad\qquad$ $\mathcal{I}_1 \quad \mathcal{S}_2 \quad \mathcal{J}'_3 \quad \mathcal{I}'_2 \quad \mathcal{S}_3 \quad \mathcal{J}_4$

$\{(0),(1)\}\ \{0,1\}\ \{0,1\}\ \{(1)\}$ $\qquad\qquad$ $\mathcal{J}'_3$: $(0,1)$, $(1,1)$; $\mathcal{I}'_2$: $(0,0)$, $(0,1)$ $\qquad$ (8)

suppose CI yields two good pivots

Nesting properties: $\mathcal{I}_1 < \mathcal{I}'_2 \not< \mathcal{I}_3$, $\qquad \mathcal{J}_2 > \mathcal{J}'_3 > \mathcal{J}_4$ $\qquad$ (9)

($\mathcal{I}_1 < \mathcal{I}'_2$) by construction; ($\mathcal{I}'_2 \not< \mathcal{I}_3$) because row pivots <u>were</u> discarded; ($\mathcal{J}_2 > \mathcal{J}'_3$) by construction; ($\mathcal{J}'_3 > \mathcal{J}_4$) because no column pivots were discarded

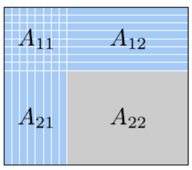
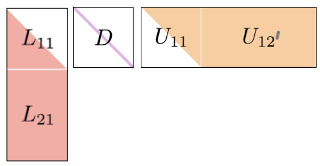
Important fact: if pivots are nested w.r.t. Π_ℓ (left-nested up to $\ell-1$ and right-nested up to $\ell+2$), then

$$\boxed{[\Pi_\ell - \tilde{\Pi}_\ell]_{i_{\ell-1}\sigma_\ell\sigma_{\ell+1}j_{\ell+2}} = [F_\ell - \tilde{F}_\ell]_{i_{\ell-1}\sigma_\ell\sigma_{\ell+1}j_{\ell+2}}}$$

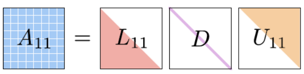
for all $\sigma_\ell, \sigma_{\ell+1} \in \mathcal{S}_\ell \times \mathcal{S}_{\ell+1}$ (proof: see TCI.10) $\qquad$ (6)

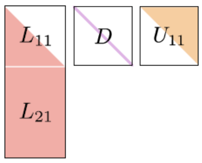
Thus, error of approximating Π_ℓ by $\tilde{\Pi}_\ell$ is also error, on this 2-dimensional slice, of approximating $F_{\vec\sigma}$ by $\tilde{F}_{\vec\sigma}$.
The above algorithm chooses pivots in order to minimize this error.

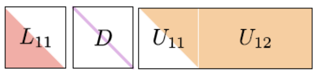
When factorizing $\Pi_\ell \simeq T_\ell P_\ell T_{\ell+1}$ via prrLU, the ingredients on the right are constructed as follows:
cf. (TCI.6.14-16)

$\Pi_\ell =$  $\simeq \tilde{\Pi}_\ell = T_\ell (P_\ell)^{-1} T_{\ell+1} = LDU = \begin{pmatrix} P_\ell & L_{11}DU_{12} \\ L_{21}DU_{11} & L_{21}DU_{12} \end{pmatrix} =$  $\qquad$ (7)

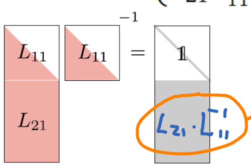
$P_\ell = L_{11}DU_{11}$ $\qquad T_\ell = \begin{pmatrix} L_{11}DU_{11} \\ L_{21}DU_{11} \end{pmatrix}$ $\qquad T_{\ell+1} = \begin{pmatrix} L_{11}DU_{11} & L_{11}DU_{12} \end{pmatrix}$ $\qquad$ (8)

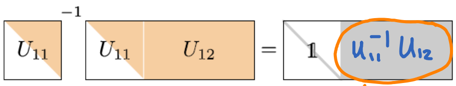






$T_\ell P_\ell^{-1} = \begin{pmatrix} \mathbb{1} \\ L_{21}L_{11}^{-1} \end{pmatrix}$ $\qquad P_\ell^{-1} T_{\ell+1} = \begin{pmatrix} \mathbb{1} & U_{11}^{-1}U_{12} \end{pmatrix}$ $\qquad$ (9)





This is what one computes in practice!
Since L_{11} and U_{11} are lower- or upper triangular, their inverses can be computed in <u>stable</u> manner using forward/backward substitution

<u>Pivot update method</u> $\qquad$ 'reset' vs. 'accumulative'

'reset' mode: replace all pivots in $\mathcal{I}_\ell, \mathcal{J}_{\ell+1}$ by all pivots in $\mathcal{I}'_\ell, \mathcal{J}'_{\ell+1}$ (as described above)
pro: can discard bad pivots; $\qquad$ con: can break nesting conditions

'accumulative' mode: don't discard pivots from $\mathcal{I}_\ell, \mathcal{J}_{\ell+1}$, just add new ones from $\mathcal{I}'_\ell, \mathcal{J}'_{\ell+1}$, typically one at a time.
pro: satisfies nesting conditions; $\qquad$ con: cannot discard bad pivots.

Both modes have runtimes $\mathcal{O}(\chi^3)$.
Accumulative: $\mathcal{O}(\chi^2)$ per update, needs χ updates to reach rank χ.
Reset: $\mathcal{O}(\chi^3)$ per update, but typically converges within a few updates, independent of χ

<u>Pivot search</u> One can use full search, rook search, or block rook search.
Scaling with d: $\mathcal{O}(d^2)$, $\mathcal{O}(d)$, $\mathcal{O}(n_{\text{rook}}\, d)$
(works well in combination with reset mode)

<u>Adding global pivots</u>

In addition to using pivots found via 2-site TCI algorithm, it may be useful to add 'global' pivots based on 'outside' information, such as:
- knowledge of configurations where $F_{\vec\sigma}$ is very large;
- doing TCI on a tensor F_2 that is very similar to a tensor F_1 whose TCI unfolding $\tilde F_1$ is already known

Strategy for adding global pivots:

- split each $\vec\sigma$ as $\vec\sigma = i_\ell \oplus j_{\ell+1}$ for all $\ell = 1, \dots, \mathcal{L}-1$,
- add these i_ℓ, $j_{\ell+1}$ to the pivot lists $\mathcal{I}_\ell$, $\mathcal{J}_{\ell+1}$
- do prrLU on all pivot matrices P_ℓ to discard any spurious pivots
- perform a few sweeps using 2-site TCI in reset mode to stabilize the pivot lists.

<u>Ergodicity</u>

TCI is based on exploration of configuration space, so can encounter ergodicity issues -- remaining stuck in subpart of configuration space and not visiting other relevant parts.
If one notices such issues, initialize pivot search with suitably chosen global pivots: Examples:
- Very sparse tensors, where TCI might miss some nonzero entries;
  remedy: add global pivots for a list of nonzero entries.
- Tensors with discrete symmetries, where exploration may get stuck in one symmetry sector;
  remedy: add one global pivot per symmetry sector.
- Multivariate functions with very narrow peaks;
  remedy: add global pivots corresponding to peak maxima.

<u>Error estimation: bare vs. environment</u>

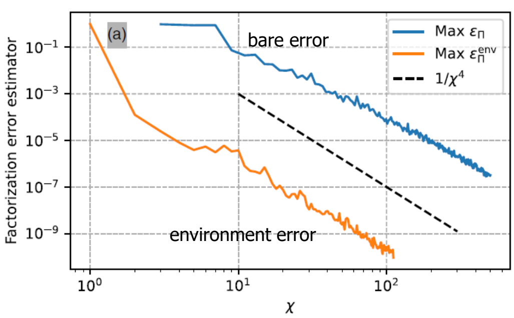
CI-decomposition on Π_ℓ [cf. (3) above] minimizes the 'bare error', $\left| [\Pi_\ell - \tilde\Pi_\ell]_{i_{\ell-1}\sigma_\ell\sigma_{\ell+1}j_{\ell+2}} \right|$, [cf. (6) above].

Alternatively option: minimize the 'environment error': $\left| L_{i_{\ell-1}} R_{j_{\ell+2}} \left[\Pi_\ell - \tilde\Pi_\ell\right]_{i_{\ell-1}\sigma_\ell\sigma_{\ell+1}j_{\ell+2}} \right|$ (10)

with left- and right environments

$$L_{i_{\ell-1}} = \sum_{\sigma_1,\dots,\sigma_{\ell-1}} [T_1^{\sigma_1} P_1^{-1} \cdots T_{\ell-1}^{\sigma_{\ell-1}} P_{\ell-1}^{-1}]_{1 i_{\ell-1}}, \qquad R_{j_{\ell+2}} = \sum_{\sigma_{\ell+2},\dots,\sigma_{\mathcal{L}}} [P_{\ell+1}^{-1} T_{\ell+2}^{\sigma_{\ell+2}} \cdots P_{\mathcal{L}-1}^{-1} T_{\mathcal{L}}^{\sigma_{\mathcal{L}}}]_{j_{\ell+2} 1} \qquad (11)$$

Minimizing environment error aims to find best approximation of 'integrated' tensor $\sum_{\vec\sigma} F_{\vec\sigma}$, summed over all indices. This is useful for computing integrals involving integrands with long tails.
Example: see Fig. 7 of [Fernandez2022].

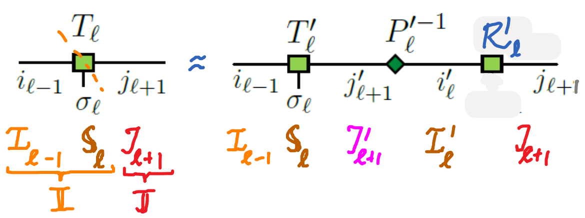


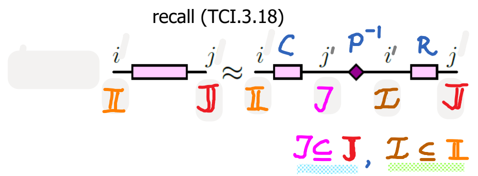
1-site TCI algorithm [Fernandez2025, Sec. 4.4.1]

Useful for (i) compressing a given TCI to smaller rank; (ii) restoring full nesting; (iii) improve pivots at lower computational cost than 2-site TCI. Limitation: cannot increase bond dimension.

Input: a TT in TCI form. General strategy: Sweep back and forth, compressing T_ℓ using prrLU.

Forward sweep: view $T_\ell \equiv F(\mathcal{I}_{\ell-1}, \mathbb{S}_\ell, \mathcal{J}_{\ell+1})$ as a matrix $F(\underbrace{\mathcal{I}_{\ell-1} \times \mathbb{S}_\ell}_{\mathbb{I}}, \underbrace{\mathcal{J}_{\ell+1}}_{\mathbb{J}})$, prrLU-factorize it, and use new pivot lists $\mathcal{I}'_\ell, \mathcal{J}'_{\ell+1}$ to update T_ℓ, P_ℓ.





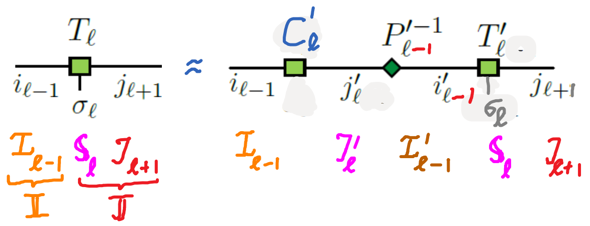
(12)

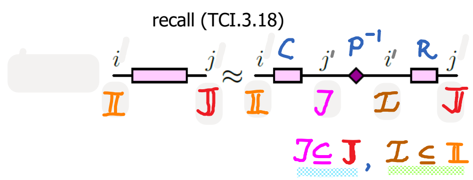
row indices: $\mathcal{I}'_\ell \subseteq \mathcal{I}_{\ell-1} \times \mathbb{S}_\ell = \mathbb{I}$ column indices: $\mathcal{J}'_{\ell+1} \subseteq \mathcal{J}_{\ell+1} = \mathbb{J}$ (13)

nested by construction: $\mathcal{I}_{\ell-1} < \mathcal{I}'_\ell$

After complete forward sweep, pivots are <u>fully left-nested</u>: $\mathcal{I}_0 < \cdots < \mathcal{I}_{\mathcal{L}-1}$

Backward sweep: view $T_\ell \equiv F(\mathcal{I}_{\ell-1}, \mathbb{S}_\ell, \mathcal{J}_{\ell+1})$ as a matrix $F(\underbrace{\mathcal{I}_{\ell-1}}_{\mathbb{I}}, \underbrace{\mathbb{S}_\ell \times \mathcal{J}_{\ell+1}}_{\mathbb{J}})$, prrLU-factorize it, and use new pivot lists $\mathcal{I}'_{\ell-1}, \mathcal{J}'_\ell$ to update $P_{\ell-1}, T_\ell$.





(14)

row indices: $\mathcal{I}'_{\ell-1} \subseteq \mathcal{I}_{\ell-1} = \mathbb{I}$ column indices: $\mathcal{J}'_\ell \subseteq \mathbb{S}_\ell \times \mathcal{J}_{\ell+1} = \mathbb{J}$ (15)

nested by construction: $\mathcal{J}'_\ell > \mathcal{J}_{\ell+1}$

After complete backward sweep, pivots are <u>fully right-nested</u>: $\mathcal{J}_2 > \cdots > \mathcal{J}_{\mathcal{L}+1}$

Backward sweeping may break left-nesting, because taking the subset $\mathcal{I}'_{\ell-1} \subseteq \mathcal{I}_{\ell-1}$ may discard pivots from $\mathcal{I}_{\ell-1}$. To achieve full nesting, do one more forward sweep at same tolerance. This preserves right-nesting, since all bond dimensions already meet the tolerance, so last forward sweep removes no pivots from $\mathcal{J}_{\ell+1}$ for $\ell = 1, .., \mathcal{L}-1$

0-site TCI algorithm [Fernandez2025, Sec. 4.4.2]

Input: given TT in TCI format. Sweep through pivot matrices P_ℓ, prrLU decomposing each to yield updated pivot lists $\mathcal{I}'_\ell, \mathcal{J}'_{\ell+1}$ that replace $\mathcal{I}_\ell, \mathcal{J}_{\ell+1}$. Main usage: improve conditioning of P_ℓ by removing 'spurious' pivots. Breaks nesting conditions.