X-Symbols for Non-Abelian Symmetries in Tensor Networks

Andreas Weichselbaum1,2, *

1Department of Condensed Matter Physics and Materials Science, Brookhaven National Laboratory, Upton, NY 11973-5000, USA
2Physics Department, Arnold Sommerfeld Center for Theoretical Physics, and Center for NanoScience, Ludwig-Maximilians-Universität, Theresienstrasse 37, 80333 Munich, Germany

(Dated: October 31, 2019)

The full exploitation of non-abelian symmetries in tensor network states (TNS) derived from a given lattice Hamiltonian is highly attractive in various aspects. From a theoretical perspective, it can offer deep insights into the entanglement structure and quantum information content of strongly correlated quantum many-body states. From a practical perspective, it allows one to push numerical efficiency by orders of magnitude. Physical expectation values based on TNS require the full contraction of a given tensor network, with the elementary ingredient being a pairwise contraction. While well-established for no or just abelian symmetries, this can become quickly extremely involved and cumbersome for general non-abelian symmetries. As shown in this work, however, the latter can be tackled in a transparent and efficient manner by introducing so-called X-symbols which deal with the underlying pairwise contraction of generalized Clebsch-Gordan tensors (CGTs). These X-symbols can be computed deterministically once and for all, and hence also be tabulated. Akin to $6j$-symbols, X-symbols are generally much smaller than their constituting CGTs. In applications, they solely affect the tensors of reduced matrix elements, and therefore, once tabulated, allow one to completely sidestep the explicit usage of CGTs, and thus to greatly increase numerical efficiency.

Tensor network states (TNS) provide a powerful natural framework for the numerical treatment of strongly correlated quantum many-body physics on lattice Hamiltonians [1–3]. Starting in one dimension (1D) with matrix product states (MPS), the numerical renormalization group (NRG, [4–6]) and the density matrix renormalization group (DMRG, [1, 2, 7]) represent powerful, non-perturbative, and accurate methods to deal with strongly correlated systems at arbitrary temperature both, statically and dynamically. An attractive extension of the 1D MPS structure was provided by the multiscale-entanglement renormalization ansatz (MERA, [8, 9]), already also with an eye on higher dimensions, even if significantly more expensive numerically. The very flexible framework of TNS for lattice Hamiltonians has already also seen a wide range of applications in two dimensions (2D) via projected entangled pair states (PEPS) [9–13] or higher [14–16]. By providing an efficient way to study entanglement in strongly correlated systems, this also generated significant interest from a quantum information perspective [17–19]. While highly efficient in 1D, the numerical cost for dealing with TNS in 2D or higher, however, grows exorbitantly even though still polynomially. Therefore the exploitation of symmetries is extremely relevant and important also on practical grounds. This includes for correlated systems in quasi-1D, i.e., for long systems of narrow width, or in the presence of multiple (symmetric) flavors [20–22].

Symmetries on all indices in a tensor network state, physical and bond indices alike, are well-defined only in generalized tree-tensor networks [23], i.e., tensor network states without loops along virtual bonds that link tensors. In the presence of loops, significant ambiguities arise. Nevertheless, on practical grounds, one typically sees that enforcing symmetries on all indices also in a TNS with loops shows clear gains in efficiency [24, 25]. An intuitive handwaving argument for this may be provided based on the interpretation of bonds as auxiliary state spaces which motivated PEPS to start with.

In the presence of non-abelian symmetries, all tensors can be decomposed into a tensor product structure of reduced matrix element tensors (RMTs) and generalized Clebsch Gordan coefficient tensors (CGTs) [22, 26] as a direct generalization of the Wigner-Eckart theorem. This results in two immediate consequences that can be used to greatly improve numerical efficiency: (i) By splitting of a CGT factor for each elementary, i.e., simple symmetry present, this allows one to (strongly) reduce the effective dimensionality $D \rightarrow D^*$ of a given index or state space by switching from a state-based description to a description based on multiplets eventually dealt with by the RMTs. (ii) The CGTs are purely related to the symmetry of a given problem. Hence much of it can be dealt with once and for all by tabulating the relevant information.

A general framework for dealing with general non-abelian symmetries in TNS was introduced in Weichselbaum (2012) [22] based on a general transparent tensor representation referred to as QSpace. Irreducible representations multiplet fusion rules together with the corresponding CGTs were tabulated there. However, contractions of CGTs were not tabulated due to the presence of outer multiplicity (OM). The prescription in [22] to deal with OM in the pairwise contraction of tensors was to always reconstruct all CGTs based on their particular instantiation in QSpace. However, from a practical point of view, this led to significant computational overhead for larger non-abelian symmetries, such as $SU(N \gtrsim 4)$. As will be shown in the present work, however, there also exists a transparent general way to deal with the problem of OM in the pairwise contraction of CGTs based the introduction of so-called X-symbols (where ‘$X$’ is simply a reference to generalized tensor multiplication, i.e., contraction). These can be computed once and for all, and thus also tabulated. X-symbols
provide an alternative approach to $6j$ symbols etc. Yet they are much more naturally suited to tensor network algorithms, since they strictly deal with the elementary operation of the contraction of a pair of tensors (and hence a pair of CGTs) on an arbitrary subset of shared indices.

Given the many reviews and detailed publications that already exist on TNS, e.g., see [2, 3, 6, 22, 27] and references therein, an elementary understanding of tensor network states is assumed in this work. With this in mind, the paper is organized as follows. Sec. I sets the stage with a focus on symmetries in TNS which strongly builds on Weichselbaum (2012) [22]. Sec. II provides conventions and preliminaries required for the rest of the paper. Sec. III then introduces X-symbols and discusses their relevance in TNS, followed by summary and outlook.

I. SYMMETRIES IN TENSOR NETWORK STATES

Tensor network states describe lattice Hamiltonians. They typically share the same lattice structure yet with nearest-neighbor bonds only (whereas the Hamiltonian may be longer ranged). Each lattice site $n$ is assigned a tensor $A_n$ that links its physical state space $|\varphi_\sigma\rangle_n$ to its connected (variationally determined) auxiliary state spaces $|\alpha_x\rangle$. In a pictorial language, the indices of a tensor are drawn as lines, also referred to as the legs of a tensor. All state spaces are indexed. E.g., the index $\sigma$ above spans the local state space of a single physical site, whereas the index $x = l,r,…$ spans specific named bonds, such as l(eft), r(ight), etc. [e.g. see (1)].

A. Arrows on all legs

The physical state spaces are orthonormal from the very outset, having $\langle \varphi_\sigma | \varphi_\sigma \rangle_n = \delta_\sigma^\sigma \delta_n^n$. For generalized tree-tensor networks (TTN) [28] which, by definition, contain no loops along any path of bond indices, cutting any auxiliary bond separates the TNS into two disconnected blocks. As a direct consequence, all auxiliary or bond state spaces can be made orthonormal. If a given TTN is an exact symmetry eigenstate globally, all bond indices can be fully symmetrized, i.e., assigned symmetry labels without increasing the bond dimension [23]. In the case of a TTN, the auxiliary bond state spaces describe well-defined orthonormal effective quantum-many-body state spaces that represent entire blocks of the system. Each such block only contains one open bond index, starting from which it necessarily stretches all the way to the open or infinite outer boundary of the physical system, and hence also of the TTN considered.

For a TTN, many-body state spaces are typically generated iteratively by adding one site after another to a block. Therefore any index or leg describes an orthonormal many-body state space that either enters a tensor as part of a tensor product space, or leaves a tensor with the interpretation of an effective combined state space. So while several lines may enter a tensor, there is always at most one line that can leave a tensor. For finite TTN simulations then, auxiliary indices all flow towards the orthogonality center (OC) [29]. This is the only tensor in an entire TTN that may have no outgoing index. It combines the orthonormal spaces of various blocks into a normalized global wave function. If multiple global states are targeted, e.g., if the global symmetry multiplet is not a scalar, then the OC also carries along an outgoing leg, namely the index that resolves the global state or multiplet.

In a pictorial description, this naturally suggests that each index (leg or line) in a tensor network (TN) is given an arrow. Correspondingly, in mathematical notation, an outgoing (incoming) index to a tensor can be written as a lowered (raised) index, which is equivalent to covariant (contravariant) index notation, respectively. A contracted, i.e., summed over index then necessarily is outgoing from one tensor and ingoing into another, consistent with Einstein summation convention that an index is summed over if it appears twice, namely as a raised and lowered index.

By having adopted the convention that the physical state space of site $n$ is denoted by $|\varphi_\sigma\rangle_n$, i.e., with lowered indices, this implies that the index (leg) in a tensor where it enters must be a raised index. Thus combining, for example, the state space of site $n$ with an effective bond state space $a_l$ (‘left’), the fused state space $a_r$ (‘right’) is given by,

$$
|a_r\rangle = \left(\sum_{l\sigma} A_{l\sigma}^r |a_l\rangle \otimes |\varphi_\sigma\rangle\right)
$$

where the summation over double indices is implicit, hence the bracket around the sum. TTNs guarantee that orthonormal state spaces, and hence also symmetries, are well-defined throughout. This is crucially important for efficient algorithms [2, 7]. These orthogonal state spaces ensure an orthogonal environment, and hence optimal conditions [30] when truncating bond dimensions to the most entangled and thus most relevant quantum-many-body states constituting a given global state. In this sense, orthogonality and hence arrows are crucially important in TTNs.

In more general TNSs in the presence of loops, however, it is no longer possible to associate all indices with well-defined strictly orthonormal state spaces in the strong sense this can be achieved with TTN [23]. Instead, one can adopt the much less rigorous PEPS-like interpretation that bonds host actual auxiliary state spaces where one simply imposes that these are orthonormal and adhering to the symmetries of the overall Hamiltonian. When dynamically truncating, and thus adapting auxiliary state spaces, it is no longer possible then to have a perfectly orthonormal environment. Instead, one needs to resort to optimal conditioning [30]. If one decides to exploit global symmetries, in any case, this necessitates that symmetries need to be enforced locally with each tensor, and therefore that all lines in a TNS carry arrows. For TNSs with loops then, such as a 2D PEPS, individual tensors typically also carry multiple outgoing legs. Independent of whether or not global symmetries are enforced, however, when describing specific algorithms in tensor networks, arrows on all legs, or equivalently raised and lowered indices generally represent a powerful natural concept for the underlying tensor algebra.
B. General tensor decomposition

In the presence of (non-abelian) symmetries, the tensor coefficients $A$ (without hat) of any tensor operator $\hat{A}$ (with hat) can be decomposed into a sum over tensor products of reduced matrix elements $|A||q_1\cdots q_r\rangle$ times generalized Clebsch-Gordan coefficients $C_q$. [22, 26]

$$A = \bigoplus_q |A||q_1\cdots q_r\rangle C_q .$$

(2)

The tensor $A$ can have arbitrary rank $r$ which is defined here as the number of legs (indices) attached. The sum over $q$ indicates sum over symmetry sectors. For each independent elementary (i.e., simple) symmetry considered, another CGT can be split off [22], collectively written as $C_q$ above. To be precise, in this paper, a CGT always refers to a specific elementary (semi-) simple symmetry such as SU(N), Sp(2n), or SO(n) [see also App. A], and can always also be chosen real. Yet the same way, an algorithm needs to deal with RMTs and CGTs separately, also CGTs for different symmetries can be dealt with completely separately and in parallel. Hence while, for the sake of the argument for simplicity only a single elementary non-abelian symmetry is considered, all arguments in this paper straightforwardly also generalize to the presence of multiple symmetries.

The decomposition in Eq. (2) holds for general symmetries, non-abelian and abelian symmetries alike. For abelian symmetries, however, all multiplets contain only a single state. Hence the tensorial structure of the CGTs reduces to simple numbers there. For non-abelian symmetries, on the other hand, one still also needs to systematically account for outer multiplicity as explained below. The overall tensorial structure such as rank or directions of incoming and outgoing indices, i.e., the arrow configuration, is exactly inherited by all terms, RMTs as well as CGTs.

In order to avoid excessive proliferation of indices, for the sake of readability, simplified shortcut notations are adopted as explained in the following. In particular, this also concerns the at times somewhat loose distinction between raised or lowered indices if the distinction is not explicitly important in a specific context. In Eq. (2), for example, the combined symmetry labels for all legs have simply been written as subscript $q$, even though legs may have mixed raised and lowered indices. The symmetry labels (simply also referred to as ‘$q$-labels’) for all $l + l'$ legs of a given CGT are given by the full set

$$q \equiv (q^1, \ldots, q^r, q_1^r, \ldots, q_r^r) \equiv (q_1^l \ldots q_l | q_1^r \ldots, q_r^r) .$$

(3)

Here each $q^i$ (or $q_i^r$) is a tuple of multiplet labels of fixed length that specifies the symmetry sector on a given index (vector space) $i$ for all symmetries considered. While the order of legs within the group of incoming (or outgoing) legs is important, incoming and outgoing indices can be arbitrarily interspersed, otherwise. This means that the position of raised indices relative to lowered indices is irrelevant. Therefore, e.g., all incoming indices can be listed first, as shown in Eq. (3). While the first decomposition of $q$ in Eq. (3) explicitly specifies raised and lowered indices, the equivalent last decomposition splits the group of incoming ($l$ legs) via the bar ‘|’ from the group of outgoing indices ($l'$ legs). It uses all lower indices, if relevant, otherwise. This is useful, for example, when discussing standard rank-3 CGTs $q \equiv (q_1 q_2 q_3)$ which fuse $q_1$ and $q_2$ into the combined total $q_3$. E.g., consider $(q i q_0)$ which fuses multiplet(s) $q$ with their dual(s) into the scalar representation, simply denoted as 0. This has no more raised or lowered indices whatsoever, which thus requires the last notation in Eq. (3).

Depending on the context, the combined label $q$ as a whole as in Eq. (2) may also be written as superscript, instead, with no specific meaning of the location, unless explicitly stated otherwise. Only when $q$ appears paired up as raised and lowered index, e.g., $C_q D^q$, the legs $q$ are considered traced over with arrows reverted in $D$ relative to $C$ (cf. conjugate tensors below). Decomposed $q$ labels with the same CGT always adhere to the interpretation of raised and lowered indices. The CGTs corresponding to the standard rank-3 CGTs above then, for example, correspond to $C_{q} \equiv C_{q}^{q_1 q_2}$ or $C_{q}^{q_1 q_2}$. The same index convention as in Eq. (3) also holds for the RMTs, i.e., using the notation $n \equiv (n^1, \ldots, n^r, q_1^r, \ldots, q_r^r)$ for the index into the tensor of reduced matrix elements which identifies specific symmetry multiplets in a state space decomposition [22]. By convention, in the notation of $q$ in Eq. (3) or also $n$, incoming indices are listed first. Note, however, that this is reverse to the bra-ket notation, e.g. as in Eq. (1), which places incoming states as kets, and hence needs to be read right-to-left.

C. Outer multiplicity

In the presence of non-abelian symmetries, tensors as in Eq. (2) are typically faced with the problem of outer multiplicity (OM) in their CGTs. This means that for exactly the same symmetry labels $q$ multiple ($m_q > 1$) orthogonal CGTs can arise [cf. Eq. (7) later]. Hence the CGTs acquire an additional multiplicity index $\mu = 1, \ldots, m_q$, i.e., $C_q \rightarrow C_{q,\mu}$. If a given CGT has no outer multiplicity, then $m_q = 1$ which represents a singleton dimensions and thus may be safely skipped. If a given combination of symmetry labels is not permissible from a symmetry point of view, then $m_q = 0$.

While there is no OM in the well-known SU(2) at the level of the standard Clebsch-Gordan coefficients (having rank $r = 3$), OM routinely also occurs in SU(2) for CGTs of rank $r > 3$ [see App. A]. A simple example for the emergence of OM in SU(2) for a rank-4 CGT is shown in (4a),

![Image](https://via.placeholder.com/150)

(4)

which represents the contraction of two multiplicity-free rank-3 CGTs $C$ and $C'$, over the shared intermediate multiplet $q_i = S_i$. However, typically there will exist multiple choices for the intermediate spin multiplet $S_i$. Therefore when ‘zooming out’, i.e., contracting the intermediate index, the result-
fewer rows than metric sectors with \( C \| \).

However, if there had been more than \( m \) (indexed by reason, the matrix orthogonal outer multiplicity components, there can be a \( C \) comes ‘out’ of a given CGT decomposition for OM index [green line in (4a)] also carries an arrow. Since this can be used to determine the level of OM for arbitrary symmetries and CGTs, as discussed later in Sec. III.

The presence of OM introduces an additional vector space. The CGTs in Eq. (2) then become a linear superposition in OM space, described by the normalized coefficients \( w_\mu \),

\[
A = \bigoplus_q \left( ||A||_q \otimes (w_\mu C_{q\mu'}) \right),
\]

(5)

with implicit summation of the multiplicity indices (Einstein summation convention). This coefficient matrix \( w \) can always be chosen (sub)unitary, i.e., having \( ww^\dagger = 1 \). Effectively, the role of \( w \) is such that it ‘ties together’ the otherwise plain tensor product between RMTs \( ||A||_\mu \) and CGTs \( C_q \), as graphically depicted in Eq. (4b+c). Within a given symmetry sector \( q \), \( ||A||_q \) are the reduced matrix elements that come with the CGT component \( \sum_\mu w_\mu C_{q\mu} \). In other words, for different orthogonal outer multiplicity components, there can be a completely different set of reduced matrix elements. For this reason, the matrix \( w \) in Eq. (5) can have at most as many rows (indexed by \( \mu \)) as there are columns (indexed by \( \mu \leq m_q \)). Fewer rows than \( m_q \) are allowed (the tensor \( A \) then simply does not span the full OM space), such that \( w^\dagger w \neq 1 \). However, if there had been more than \( m_q \) rows, QR or singular value (SV) decomposition of \( w = z\bar{w} \) would allow one to reduce to the effective form in Eq. (5) with \( z \) contracted onto the RMT \( ||A||_q \), and an altered, now unitary \( \bar{w} \).

The CGTs \( C_q \) can be computed once and for all and then stored (tabulated) in a central, dynamically generated database. For the storage of any tensor \( A \) then only a reference to the CGTs is required in terms of metadata. Associating \( w \) with \( C_q \), i.e., considering \( \sum_\mu w_\mu C_{q\mu'} \), in principle, one can explicitly store these individual OM components with \( ||A||_q \) for each \( \mu \), resulting in a non-unique listing of symmetry sectors \( q \) in the listing of non-zero blocks of \( A \) in Eq. (5). This was the procedure in [22]. However, instead, one can explicitly add an OM index \( \mu \) onto the RMT to the already existing indices \( n \), and contract \( w \) onto it, \( ||A||_{qn} \rightarrow \sum_\mu ||A||_{qn} w_\mu \).

Then the tensor \( A \) can be written as a unique listing over symmetry sectors \( q \) [see App. B]. While the matrix \( w \) may be eliminated this way, for practical purposes, it may be explicitly kept with the tensor decomposition in Eq. (5) nevertheless. For example, this then allows one to simply absorb operations in OM space such as orthogonal transformations into \( w \), while the possibly large CGTs and RMTs can remain unaltered.

## II. CONVENTIONS AND PRELIMINARIES

### A. Conjugate tensors

The conjugate tensor \( A^\dagger \) of any given tensor \( A \) is defined by the following prescription: take the complex conjugate of all of its matrix elements, i.e., \( A^* \) (relevant for RMTs only, since CGTs are real), then revert all arrows, and take a mirror image of the tensor when depicted graphically [22]. Effectively, the latter is the generalization of the transposition part in what one means by \( A^\dagger \), e.g., if \( A \) is an operator sandwiched in between a bra-ket bracket. Taking the mirror image, actually, is a rather common graphical procedure in TNS, even if one typically does not further dwell on this. For example, when computing expectation values \( \langle \psi | A | \psi \rangle \) for a given MPS, this involves the contraction of an \( A \)-tensor as in (1) with itself, as shown in (6a),

![Diagram](image)

Here \( A^\dagger \) (empty square at the bottom) is the mirror image of \( A \) (filled square at the top) with the arrows on all lines reverted, while also taking complex conjugated matrix elements. The mirror plane is indicated by the horizontal thick yellow line. Another example is the representation of an operator \( \hat{S} \) that acts on a local state space as depicted in (6b) which, e.g., may be inserted into the vertical line labeled by \( \sigma \) in (6a). The operator \( \hat{S} \) can consist of set of operators indexed by \( \alpha \) that is irreducible w.r.t. the symmetries under consideration. This makes it an irreducible operator (irop), which thus naturally acquires the third index \( \alpha \) [22]. It can also be non-Hermitian, e.g. considering fermionic hopping, or spin raising and lowering operators. Here (6c) summarizes what one means by taking its Hermitian conjugate: matrix elements are complex conjugated \( (S \rightarrow S^*) \), top and bottom index are exchanged (tangled, yet non-crossing lines), all arrows are reverted, and a mirror image is taken (here w.r.t. to the vertical thick yellow line which flips the irop index \( \alpha \) to the left). Untangling the lines by rotating the tensor clockwise by 180 degrees which leaves the tensor unaltered per se, leads to the correct final result, indicated by \( S^\dagger \) to the right. It looks much the same as (6b), but now with reverted index \( \alpha \). The latter is important, for example, when deriving scalar operators from bilinears, as in \( \tilde{S} \cdot \tilde{S}^\dagger \), where the dot product simply translates to a contracted directed line \( \alpha \).

The above procedure of conjugating tensors also shows up systematically when dealing with fermionic PEPS [12]. In terms of index order convention, taking a mirror image implies w.r.t. the original tensor that if indices are read clockwise in \( A \) starting from the some arbitrary but fixed index, they need to be read counter-clockwise in \( A^\dagger \). Yet as pointed out earlier with Eq. (3), incoming and outgoing indices can be grouped while ensuring that the order within these groups is preserved.
As a further graphical emphasis that one is dealing with a conjugate tensor in a TN sketch, this paper frequently uses the convention already applied in Eq. (6) above that one switches from a filled tensor (box, circle, etc.) to an empty outlined object, or vice versa. As a further example, in Eq. (4)a, the conjugate tensor $C^\dagger$ with two outgoing indices was depicted with an empty circle, in contrast to the standard rank-3 CGT $C$ with two incoming indices depicted by a filled circle. In a mathematical notation, a conjugate tensor can be indicated by flipping the location of its indices, e.g., $(C_{q\mu})^\dagger \equiv C^{\mu q}$.

### B. CGT normalization convention in OM space

When considering a CGT $C_q$ for an arbitrary but fixed set of symmetry labels $q$, a convenient natural normalization convention in OM space is,

$$\text{Tr}(C_{q\mu} C_{q\mu'}) \equiv \text{Tr}(C_{q\mu} C^{\mu q'}) = \delta_{\mu}^{\mu'},$$

(7)

where $\text{Tr}(C_{q\mu} C_{q\mu'})$ stands for the full contraction (tensor trace) of $C_q$ with the conjugate of itself, only keeping OM indices open. By pairing with its conjugate tensor, this also reverts the arrow of the OM index, hence raises the index $\mu'$ in Eq. (7). The CGT normalization in Eq. (7) simply generates an orthonormal basis in the OM ‘vector’ space. This is convenient when performing a decomposition in OM space.

For rank-3 CGTs, the normalization in Eq. (7) is closely related to the normalization of Wigner-3j symbols [31]. It is different, though, from the normalization of the standard Clebsch-Gordan coefficients e.g. for SU(2) $(S_1S_2|S_3)$ that fuse $(S_1, S_2)$ into an orthonormal basis in $S_3$ (and not in OM). The latter would result in a norm in Eq. (7) that is equal to $|S_3|$, i.e., the dimension of multiplet $S_3$. Consequently, this would make the normalization dependent on the direction of arrows in a given CGT. This is rather inconvenient on general grounds, and specifically so in the TNS context, since the presence or not of OM does neither depend on the direction of arrows nor the specific order of symmetry labels. This follows from the discussion below that raising or lowering of an index on a CGT is equivalent to applying a unitary matrix on that index while also switching to the dual representation on that leg [cf. Sec. III D]. Conversely, a permutation of indices on a CGT can only induce orthogonal rotations in OM space [cf. Sec. III C] which clearly also leaves the OM dimension invariant.

Therefore based on the normalization in Eq. (7), the standard CGTs with $q = (q_1, q_2|q_3) \equiv (q_1^2q_2q_3)$ have the same normalization for any permutation of the symmetry labels together with arbitrary raising or lowering of indices. Raising and lowering of labels, however, changes the interpretation of which multiplets are fused together. By raising all indices in a CGT, for $q = (q^1q^2q^3)$, this is equivalent to fusing all three multiplets of a standard CGT into a total scalar representation denoted by $q_{tot} = 0$, i.e., $q = (q_1, q_2, q_3|0)$, and subsequently skipping the trailing singleton dimension. The resulting rank-3 CGT has all-incoming legs, i.e., with $q = (q^1q^2q^3)$ is exactly equivalent to Wigner-3j (‘3$q$’) symbols up to an overall sign. In practice, here we follow the sign convention of the standard Clebsch-Gordan coefficients, that the first non-zero matrix element in a CGT $C_q$ is chosen positive.

### C. Fully contracted CGT networks

When a tensor network is fully contracted, all indices are paired up and summed over. This holds for both, the RMTs as well as the CGTs. Here, for the sake of the argument, however, the focus is on an isolated fully contracted TNS solely comprised of CGTs. The OM for each participating CGT is assumed fixed to some arbitrary but fixed linear superpositions in OM space [e.g. see Eq. (5)], such that there is no open index left in the TNS, and the full contraction yields some number $x$. By construction, this number can be non-zero only if the CGT network is permissible from a symmetry point of view. Now if one opens up a single index in this otherwise fully contracted CGT network, say with symmetry label $q$, the result is proportional to a rank-2 CGT $C_{q\mu}^{\mu q'}$, i.e., with one incoming and one outgoing index. The only such CGT that exists is the identity matrix up to normalization. Its graphical representation is a single directed line. Therefore it follows that a CGT network that is fully contracted up to a single opened up index $i$, necessarily is proportional to the identity matrix $1|q_i|$, i.e., the size of multiplet $q_i$.

As a specific example, consider the tensor network in Eq. (7) of just one CGT fully contracted with itself for arbitrary but fixed $\mu$ and $\mu'$. Then opening up one bond index $i$ with symmetry label $q_i$, one obtains

$$\text{Tr}_{\{q_i\}}(C_{q\mu} C_{q\mu'}) = \delta_{\mu}^{\mu'}|q_i|^1,$$

(8)

The normalization is determined by the requirement that the final trace over $i$, when performed, results back in Eq. (7).

### D. Reverting arrows and 1j$\boldsymbol{\mu}$-symbols

Reverting the arrow on a given bond in a TNS changes its interpretation, as well as the interpretation of the associated tensors. As this will be useful also later in the CGT context, consider first the elementary process where the orthogonality center in an MPS is iteratively propagated from site $n \rightarrow n+1$ with associated tensors $A_n$ and $A_{n+1}$ contracted on their shared auxiliary bond [2],

$$\tilde{A}_n A_{n+1} = (A_n \tilde{X}_n) A_{n+1} \equiv \tilde{A}_n (\tilde{X}_n^{-1} \tilde{X}_n) A_{n+1},$$

(9)

where the tilde indicates the tensor that carries the OC. Before the iteration step, the OC is located on site $n$, and the bond in between sites $(n, n+1)$ describes an orthonormal effective many-body state space for the entire right block of sites $n' > n$. After the iteration step, conversely, the arrow on the bond changed its direction, and now describes the orthonormal effective many-body space for the entire left block of sites $n' \leq n$. On a procedural level, one starts with the tensor $A_n$ in Eq. (9) that carries the OC and performs QR or SV
Hence its multiplet only consists of a single state, i.e., $\tilde{A}_n = A_n \tilde{X}_n$, where $A_n$ is a new isometry, and the OC is now shifted onto the tensor $\tilde{X}_n$ located on the bond in between sites $(n, n+1)$. When contracted onto $A_{n+1}$, this makes the former isometry $A_{n+1}$ the new OC $\tilde{A}_{n+1}$. Formally, in the last equality of Eq. (9), the direction of a bond is flipped by using the Gauge freedom inherent to TNS [2]. This consists of inserting the identity $1 = X_n^{-1} X_n$ and then associating the tensors $X_n^{-1}$ and $\tilde{X}_n$ with the left and right $A$-tensors, respectively. However, as shown to left of Eq. (9), often it is not necessary to actually compute the inverse of $X_n$, as this may be ill-conditioned. Eventually, one can exactly flip the orientation of a leg in a TNS without changing the global physical state. One only changes the local perspective and interpretation by splitting off the tensor $\tilde{X}_n$ from $A_n$ and contracting it onto the neighboring tensor $A_{n+1}$.

The discussion of flipping the direction of an arrow an a given elementary CGT $C_q$ with fixed symmetry labels $q$ follows much of the same spirit. In contrast to RMTs, however, the procedure is naturally much more constrained for CGTs. For example, a CGT is already always in a canonical form [cf. higher-order SV decomposition, 14]. Even more, with \( \text{Tr}(U q q^\dagger U^\dagger q) \equiv |q| = \text{Tr}[1^{|q|}] \) and with the normalization as chosen in Eq. (10a), $U^{qq}$ must therefore be unitary (strictly speaking, orthogonal, since CGTs are always real).

The singleton dimension in the scalar representation, indicated by the dashed line to the right, will be frequently skipped in $U$ to emphasize that $U$ is a matrix, indeed. Hence the subscript $0$ has been written in brackets. Now $\text{Tr}(U^\dagger U)$ [Eq. (10b)] is proportional to a rank-3 CGT fully contracted with its conjugate. So when opening up a single index, namely the one for $q$, this must be proportional to the identity matrix [cf. Eq. (8)], and with the normalization as chosen in Eq. (10a), $U^{qq}$ must therefore be unitary. By simply contracting $U$ and $U^\dagger = U^{-1}$ onto the two neighboring tensors linked via the contraction, as indicated by the broad arrows to the left and to the right in light color in (10d), the arrow on the black line in the center now points in the opposite direction as compared to (10c). At the same time, the symmetry label on the bond flipped to its dual, i.e., $q \to \tilde{q}$. When not mentioned explicitly, the latter will be implicitly assumed whenever individual indices are lowered or raised. Since the unitary matrix $U^{qq}$ effectively only refers to a single symmetry label $q$, the unitary $U^{qq}$ is referred to as a \( 1j \)-symbol with reference to the literature on SU(2) [32, 33].

In order to show this, it is sufficient to narrow the discussion further down to a single directed line with symmetry label $q$, assuming a single elementary non-abelian symmetry for the sake of the argument without restricting the case. This line may represent, e.g., an auxiliary index in a TNS, and may be associated with the CGT $C_q \propto 1^{|q|}$. In general, now any irreducible representation (irrep) $q$ has a unique representation $\tilde{q}$. This dual shares the same multiplet dimension, i.e., $|q| = |\tilde{q}|$, and it is the only irrep that, when fused together with $q$, permits the scalar representation as an outcome, i.e., having $(q\tilde{q}|0\rangle)$ exactly once. The corresponding CGT $C_0^{qq}$ therefore never has OM, but is unique up to an overall sign convention. By definition, the scalar representation, always denoted by $q = 0$ here, is fully symmetric under symmetry operations. Hence its multiplet only consists of a single state, i.e., $|0\rangle = 1$. For example, SU(2) is self-dual, i.e., $\tilde{q} = q$ for all $q$, such that the product space of spin $q = S$ with itself (and itself only) always also yields a singlet with $S_{\text{tot}} = 0$, having $(SS|0\rangle)$. The same argument can be further extended also to SU(1/2) abelian symmetries such as charge ($N$) or spin ($S_z$). There the dual is simply given by $\tilde{q} = -q$, since $q + (-q) = 0$.

By making use of the dual representation, this allows one to define the unitary matrix $U^{qq}_{(0)} = \sqrt{|q|} \cdot C_0^{qq}$ of dimension $|q|$ for any representation $q$ as depicted in (10a).

\[ U_{(0)}^{qq} \equiv \sqrt{|q|} \times \]
A 1\textit{j}-symbol can be computed, in principle, via an irep de-
composition of \((q|q|)\). Starting from maximum weight states, how-
evertheless, the largest irreps are always generated first, with the

\(1\textit{j}\)-symbol the very last CGT to be generated. This is not prac-
tical for very large multiplets, bearing in mind that \(|q|=|\bar{q}|\).
Also, in TNS one typically fuses a given large effective state
space (bond index) with new local state spaces of small di-
mension. That is large (effective) multiplets get fused rou-
tinely with (much) smaller ones. But for most part, one can
avoid fusing two large multiplets. This is specifically impor-
tant for large symmetries such as SU\((N \geq 4)\) [cf. App. A].

Therefore an alternative route to computing \(1\textit{j}\)-symbols is
desirable. Note that \(1\textit{j}\)-symbols are only square matrices of
dimension \(|q|^2\), which is in stark contrast to fully decompose a

\(|q|^2\) dimensional vector space into irreducible representa-
tions. Moreover, \(1\textit{j}\)-symbols are (close to) anti-diagonal, i.e., very
sparse like CGTs in general. In the absence of inner multiplic-
ty (IM) [22] such as for SU\(2\), they are strictly anti-diagonal
with alternating entries \(\pm 1\).

The non-trivial part for \(1\textit{j}\)-symbols arises from the anti-
diagonal block structure due to IM which requires consistent
conventions on how to decompose IM spaces [22]. The \(1\textit{j}\)
symbol derives from a standard rank-3 CGT. When the un-
derlying symmetry already permits OM for standard rank-3
CGTs, this also implies the presence of IM, and hence block
structure in \(1\textit{j}\) symbols. The \(1\textit{j}\) symbol itself, however, is
unique otherwise up to a global sign convention which is sim-
ply inherited here from the overall sign convention on CGTs.

The approach taken then, for example, in QSpace [App. B]
to compute \(1\textit{j}\) symbols is based on the fact that the scalar mul-
tiplet \(q=0\) is destroyed by every one of the \(\alpha=1, \ldots, \tau_{\text{sym}}\)
generalized raising and lowering operators \(S^{(j)}_{\alpha}\), with \(\tau_{\text{sym}}\)
the rank of a given symmetry [cf. App. A]. Having explicit access
to the sparse generators \(S^{(j)}_{\alpha}\) in the representation \(q\) of some
given symmetry of rank \(\tau_{\text{sym}}\), one can resort to a variational
Krylov based minimization, and compute the ground state of the
sparse pseudo Hamiltonian (cost function),

\[ H^*_q = \sum_{\alpha \leq \tau_{\text{sym}}} (S^{(j)}_{\alpha} S^{(j)}_{\alpha} + S^{(j)}_{\alpha} S^{(j)}_{\alpha}^\dagger) q. \]

This is in general a well-conditioned problem with a unique
ground state (the \(1\textit{j}\)-symbol) at ‘energy’ zero and with a ‘gap’
of order 1. By construction, this ground state must be simul-
taneously maximum \(a\)nd minimum weight state, hence repre-
sents a scalar multiplet. The Krylov based minimization then
allows one to directly converge the \(1\textit{j}\)-symbols via iterative
means down to ones numerical floating point precision. The
sparse nature of the the \(1\textit{j}\)-symbols strongly limits the varia-
tional parameter space, and hence leads to fast convergence.

2. \(1\textit{j}\)-symbols via contractions

A useful application of \(1\textit{j}\)-symbols arises when computing
a ground state of a system which itself is in a global singlet

symmetry sector, i.e., the scalar representation \(q_{\text{tot}} = 0\). By
skipping this global singleton dimensions, this requires access
to simple \(1\textit{j}\)-symbols during setup. However, when sweeping
through the TNS, the OC repeatedly gets located on an aux-
iliary bond deep inside the TNS where a wide range of mul-
tiplets can be explored. Here the shifting of the OC can also
be achieved by contraction, e.g., by projecting onto identity
\(A\)-tensors (which correspond to simple \(A\)-tensor as in Eq. (1)
yet initially without truncation [22]). The tensor \(X\) that car-
ries the OC on a bond is of rank-2 with all indices incoming.
Therefore, up to normalization, all of its CGTs necessarily
must correspond to \(1\textit{j}\)-symbols. In this sense, \(1\textit{j}\)-symbols
can also be generated via contractions, e.g., when sweeping
the OC through the TNS.

E. Determination of OM dimension

Outer multiplicity of a given CGT is independent of the
direction of its legs. This is apparent from the above explicit
construction of reverting arrows which solely corresponds
to applying a specific unitary on a given leg. Therefore OM is an
intrinsic quantity of a CGT. Consequently, the OM index in a
pictorial description needs to be attached to the tensor itself
(and not to any of the leg), as already shown with the tensor
\(C\) in Eq. (4c).

The OM dimension for rank-3 CGTs can be determined via
standard fusion rules of a pair of irreducible representations.
If such a tensor product arises from building a quantum many
body state starting from the vacuum state and iterative fusion
of local state spaces, they need to be computed in full via a
standard decomposition of a pair of irreps [22]. These rank-3
CGTs also build the elementary basis and starting point for
subsequent contractions of CGTs.

Now if one encounters a CGT of rank \(r > 3\), its full OM can be
determined iteratively in a constructive way from smaller
rank tensors assuming that their OM is known. For example,
for a rank-4 CGT, the example in (4a) for SU\(2\) can general-
ized as follows: since arrow directions can be altered at will
without affecting the OM dimension (bearing in mind to also
switch to dual representations), one can build the sequential
MPS-like structure in (11a),

\[
(a) \quad C_3 \quad \begin{array}{c}
q_1 \\
q_2 \\
q_4 \\
q_3
\end{array}
(b) \quad \begin{array}{c}
q_1 \\
q_2 \\
q_4 \\
q_3
\end{array}
\]

by taking the fused multiplet \(q\) out of a CGT \(C_3\) into a tensor
product space with irep \(\mu_3\), and then picking the desired irep
\(q_4\). For fixed \(q\)’s then, given that there is no loop in (11a),
the combined OM \(\mu_{\text{tot}} = 1, \ldots, m_{\text{tot}}\) of the CGT described by
brown box is simply the product of the multiplicities of \(C_1\) and
\(C_2\). Bearing in mind, that the intermediate contracted multi-
plet \(q_4\) can vary, the total OM of the CGT with \(q = (q_1 q_2 q_3 q_4)\)
is given by,

$$m_{\text{tot}} = \sum_q m_1(q_i) \times m_2(q_i).$$ (12)

This can be shown by building an OM basis for the CGTs derived from (11a) for all $\mu_1 \leq m_1$, $\mu_2 \leq m_2$, and for all permissible $q_i$. However, these are already all orthogonal to each other, as seen from computing their overlap as in (11b). Starting with the orthogonality of the standard rank-3 tensors, the contraction in left blue box in (11b) is proportional to the identity matrix, i.e., reduces to a simple line with weight \(\propto \delta_{\mu_1,\mu_2} \delta_{q_i, q_i'}\). When repeated iteratively with the next (here last) pair of CGTs $C_1$ and $C_2'$, this directly leads to Eq. (12). As emphasized there, the multiplicities $m_1$ and $m_2$ clearly depend on the choice of $q_i$. The above procedure can be extended towards any sequence of CGTs also of higher rank, that are contracted in a linear sequence without loops. As a result this demonstrates, that the full outer multiplicity grows rapidly (exponentially) with increasing rank of a tensor.

As a general strategy then to avoid proliferation of OM spaces, this suggests (i) to reduce the rank of a tensor by fusing indices as far as possible in a TN algorithm. Moreover, the actual level of OM generated also depends on the specific TN calculation performed. For the largest CGTs encountered, typically a far smaller OM space is explicitly generated by contractions than theoretically possible. Hence (ii) one can refrain from insisting to build the full OM space in any circumstances encountered. Rather, one can build the OM space on demand [see App. B]. If a new OM component is encountered via contractions, it can be added once and for all to ones database. On the downside, a build up of the CGT database this way becomes dependent on the history of calculations. So one must be extremely careful to ensure consistency across independent calculations or threads that simultaneously access the same central database. This can be achieved by coordinating updates, e.g., via locking mechanisms, in order to avoid race conditions resulting in inconsistent histories.

III. CONTRACTIONS

A. Pairwise contractions and X-symbols

Contractions in any TN state, in practice, are always tackled by elementary pairwise contractions, in complete analogy to evaluating the product of multiple matrices. Hence the elementary step for contracting a TN state is the contraction of two tensors. To be specific, consider some rank-4 tensor with arbitrary but fixed index order 1, \ldots, 4, as shown in (13a),

\[
\begin{array}{cc}
\text{(a)} & \text{(b)} \\
A & \overline{B}
\end{array}
\]

In the presence of symmetries, it is redrawn schematically in the spirit of (4a,c) in (13b), depicting the decomposition of symmetry sectors into the tensor product of RMTs and CGTs. For simplicity only a single CGT is shown, while in the presence of multiple symmetries, each has its own CGT. Also with reference to Eq. (5), there is a sum over symmetry sectors while in the pictorial representation in (13b) the focus is on one arbitrary but fixed set of symmetry labels $q$ with the matrix $w$ that links the CGT to the corresponding RMT. To further simplify the following discussion, (13b) is redrawn in (13c) with indices and arrows removed, while bearing in mind that, of course, arrows and index order stay intact. Also the matrix $(w_q)_{\mu}^i$ can be fully merged with (i.e., contracted onto) $|A|_q$. Hence the matrix $w$ is not explicitly needed for the sake of the argument here, and thus is skipped. The shading of the CGT at the top of (13c), finally, indicates that the CGT itself does not need to be explicitly stored with the tensor $A$ itself, but that a reference to a central database suffices. Depending on the contraction, however, the resulting CGTs may need to be updated centrally, e.g., if a new OM component is encountered.

Now consider a pair of tensors, $A$ and $B$, as in (13) contracted on a shared set of indices (state spaces), as shown in (14a),

\[
\begin{array}{ccc}
\text{(a)} & \text{(b)} & \text{(c)} \\
A & B & \text{X}
\end{array}
\]

This is a generalized matrix multiplication, thus is symbolically written as $A \times B$. In the presence of non-abelian symmetries, they contain references to CGTs [(14b)]. Via the tensor product structure in Eq. (5), the tensorial structure such as rank and direction of in- or out-going indices is exactly the same for both, RMTs as well as CGTs. Therefore when performing a contraction of two tensors on a specified set of indices in a TNS, precisely the same contraction needs to be performed on the level of the RMTs, $|A| \times |B|$, as well as on the level of CGTs, $C \times D$. These are separate from each other, and hence can be dealt with completely independently. The contraction of the RMTs always needs to be performed explicitly, as this is part of the physical problem under investigation. The contraction of CGTs, however, is purely related to symmetries, and hence can be computed once and for all and tabulated. Now consider the contraction of some arbitrary but fixed pair of CGTs, $C \times D$ as in (14b), shown in (15a),

\[
\begin{array}{ccc}
\text{(a)} & \text{(b)} & \text{(c)} \\
C \times D & \text{X} & E \text{X}
\end{array}
\]

The contraction of two CGTs $C \times D$ necessarily yields another CGT labeled $E$ here, with its own orthonormal OM space. Assuming its OM space is complete, then the contraction of $C \times D$ can be projected by inserting the identity $E^\dagger E = 1$ in (15b). When fully contracting the conjugate $E^\dagger$ onto $C \times D$,
this results in what is referred to as an \(X\)-symbol,

\[
X^{\mu C \mu D} \equiv \text{Tr}\left[\left(\text{Tr}_{i_C}([C^{\mu C} \ast D^{\mu D}]) \ast E_{\mu E}^{\dagger}\right)\right], \quad (16)
\]

where \(C\) is contracted on legs \(i_C\) with \(D\) on legs \(i_D\), the result of which is fully contracted with \(E^{\dagger}\), while keeping OM indices open.

The \(X\)-symbol is derived from a contraction (hence ‘X’) of a pair of CGTs of arbitrary rank and with fixed symmetry labels. It has all TNS-related indices fully contracted [yellow lines(15b)]. Consequently, \(X\)-symbols always only have three open indices \(\mu_C\), \(\mu_D\), and \(\mu_E\), all of which are indices into outer multiplicity spaces (green lines). With \(E\) a CGT fixed by symmetry, it can be referenced in the final object [again suggested by the shading in (15c)]. This overall result now can be inserted back into (14c).

The pictorial representation in (14c) then exemplifies the central result of this work: in the process of a pairwise contraction of tensors, it suffices to contract the centrally stored \(X\)-symbol onto the OM indices of the corresponding pair of RMTs, thus merging and decomposing the OM spaces. Nothing else remains to be done on the level of CGTs themselves. Therefore if all \(X\)-symbol are available and up to date in the database, the CGTs themselves can be completely sidestepped. The \(X\)-symbols fully take care of the symmetry related multiplicity spaces in an efficient and general manner.

Each \(X\)-symbol needs to remember where it came from via metadata, as schematically indicated with Eq. (16). This includes references to the three participating CGTs \(C\), \(D\), and \(E\) (which also specifies all their symmetry labels, order and direction of legs), and what indices have been contracted \([i_C\) and \(i_D\) in Eq. (16)]. Moreover, one may contract the conjugate of the input tensors \(C\) or \(D\) [indicated then by \(i_C^*\) or \(i_D^*\) in Eq. (16)], hence the \(X\)-symbol also stores conjugation flags for all three CGTs. If the OM space is build successively via contractions as they occur, the \(X\)-symbol also needs to remember identifiers as to the state of CGTs such as a high-resolution time stamp of their last modification time when the \(X\)-symbol was computed. Then if any of the CGTs gets updated later along the course of a calculation, these serve as flags as to whether or not also the \(X\)-symbol needs to be updated when the same contraction is re-encountered at a later stage.

Note that if \(C\) and \(D\) already have complete OM individually, this does not at all imply that also the \(C \ast D\) will exhaust the OM space of the resulting CGT \(E\) [cf. Eq. (12)]. When (re)computing an \(X\)-symbol, if \(E\) is already present e.g. from other earlier contractions, \(C \ast D\) needs to be projected onto it. If the OM space of the current \(E\) was already complete, the projection can fully represent the result. If the OM space of \(E\) was not complete, then new OM components out of \(C \ast D\) may arise, which need to be extracted and orthonormalized via Schmidt decomposition (performed twice for numerical stability). As this extends the OM space of the CGT \(E\), finally, it needs to updated centrally.

B. Relation to \(6j\)-symbols

Consider the fully contracted tensor network of four standard rank-3 CGTs with a total of six contracted lines in (17a),

\[
\begin{align*}
\text{(a)} & \quad A & \equiv & \quad \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\text{\(q_1\)}
\text{\(q_2\)}
\text{\(q_3\)}
\text{\(q_4\)}
\end{array}
\end{array}
\end{array}
\end{array}
\end{align*}
\]

The part left of the yellow vertical marker, for example, may represent the matrix elements of an operator \(S^\dagger\) on the local site incorporated by tensor \(A\) in a matrix product state [2, 6, 22, see also Eq. (6)]. With focus on CGTs here, the above tensor network is viewed only at the level of CGTs in a particular configuration with all \(q\)-labels fixed. Then the contraction of the TN left of the yellow marker line results in a rank-3 CGT \((q_4 \ast q_2 q_4) \equiv (q_4 q_2 q_2)^\dagger\). The result of the contraction \(A \ast S^\dagger \ast A\) therefore resembles the conjugate of a standard rank-3 CGT. In order to determine the precise decomposition, it can be projected onto the CGT \((q_4 q_2 q_2)^\dagger\), marked as \(Y\) in (17a,b).

The TN in (17a) contains six contracted indices, and therefore six symmetry sectors \(q = \{q_1, q_2, q_2, q_3, q_3, q_4\}\). They connect four standard rank-3 CGTs, resulting in a fully contracted TN. Overall, therefore (17a) represents a ‘\(6j\)’ symbol, or in the SU(2) context, the well-known \(6j\)-symbol. Contrary to the case of SU(2), however, which has no OM at the rank-3 level, for general symmetries, each of the four participating CGTs in (17a) can carry outer multiplicity. Therefore the resulting ‘\(6j\) symbol has four open OM indices for a general non-abelian symmetry, i.e., represents a rank-4 tensor purely in terms of OM indices.

Now the contraction in (17a) can be performed pairwise. The order of contractions is somewhat arbitrary, where the one chosen in (17b) is \(((A \ast S^\dagger) \ast A^\dagger) \ast Y\). Each of the pairwise contractions of CGTs can make use of X-symbols. The first contraction \(A \ast S^\dagger\), say, makes use of the X-symbol \(x_1\). The resulting CGT contracted with \(A^\dagger\) makes use of the X-symbol \(x_2\), the result of which when contracted with \(Y\) makes use of \(x_3\). For the pairwise contraction of full tensors which include RMTs and CGTs, the X-symbols actually need to be contracted onto the RMTs as discussed with (14c).

In that sense, the TN in (17b) shows a TN in terms of the RMTs that come with the CGTs in (17a). Now the sequence of X-symbols depicted in (17b; green lines) can be isolated, while keeping OM indices open. With (15c) then, the resulting MPS-like sequence of X-symbols in (17c) exactly corresponds to the contraction of the CGTs in (17a). Since the TN in (17a) is fully contracted, the CGT describing the overall result is a rank-0 CGT, which itself clearly has no outer multiplicity. Consequently, the sequence in (17c) stops with a singleton dimension indicated by a dashed line with label ‘0’. It follows therefore from the above constructive approach, that any fully contracted TN built from CGTs can be decomposed into a linear sequence of contractions based on X-
symbols. In this sense, X-symbols are equally general as $3n-j$ symbols, in that any $3n-j$ symbol can be computed from them. Yet X-symbols are much more naturally suited to tensor network algorithms, in that they provide a general prescription for the elementary operation of a pairwise contraction of two tensors.

C. Permutations

The indices of any tensor in a TNS need to be chosen in some arbitrary but fixed order. The precise choice of order within the TNS is typically a matter of convention, but of no further concern, otherwise. Permutations, if performed correspond to resorting of matrix elements according to the new index order. The same also holds for CGTs. Therefore, by convention, it suffices to only tabulate sorted CGTs which have their symmetry labels (q-labels) sorted within the set of incoming or outgoing indices, e.g., in a lexicographical style. Any reference to a specific CGT then includes a reference to a sorted CGT together with a permutation $p$ describing the actual index order. The adopted sign convention is that the stored sorted CGT starts with a positive coefficient. The permuted references adhere to this original sorted tensor up to the permutation only, i.e., there is no further sign change. This approach of making use of sorted CGTs, i.e., with sorted q-labels, allows one to significantly reduce redundancy of the CGTs that need to be stored, or subsequently, also contracted.

The above is a well-defined prescription for CGTs that have different q-labels on all of their legs. However, subtleties arise if symmetry labels are degenerate, i.e., when precisely the same representation occurs on more than one leg within the group of either incoming or outgoing indices. Such a permuted references adhere to this original sorted tensor up to the permutation only, i.e., there is no further sign change. This approach of making use of sorted CGTs, i.e., with sorted q-labels, allows one to significantly reduce redundancy of the CGTs that need to be stored, or subsequently, also contracted.

Since the full tensor-product decomposition of the fusion of two large multiplets can become prohibitive for large symmetries in terms of computational cost [cf. App. A], the question arises, to what extent specific (standard) rank-3 CGTs can be obtained by other means.

Now being interested in some specific CGT $(q_1 q_2 | q_3)$, in a TNS setting, the typical situation is such that at least one of the legs corresponds to either the local state space of a physical site or to the irreps according to which practically relevant irreducible operators transform. All of these are typically considered to be multiplets of small dimension. Therefore, for TN simulations, in practice, one of the irreps $q_i$ in a rank-3 CGTs, with $i = 1, 3$ can be considered small. Frequently, it may even refer to the defining representation or its dual in which case these may be referred to as primitive CGTs [34, 35]. One simple strategy to compute such rank-3 CGTs with their full OM, is to exploit the freedom that arrows can be reverted at will (while also switching to dual irreps). Hence the multiplets $q_i$ can be sorted according to their dimension $|q_i|$. Then the full tensor-product decomposition may be performed by taking the tensor product of the smallest two irreps, and subsequently reverting arrows as needed, making use of 1j-symbols.

On the other hand, new CGTs are also routinely generated via contractions. A possible strategy thus concerns the explicit generation of new standard rank-3 CGTs, i.e., with two incoming and one outgoing index, via a recursive scheme based on contractions for the sake of numerical efficiency [34, 36, 37]. To start with, consider the standard rank-3 CGT in (18a).

\[
\begin{align*}
&(a) \quad q_1 \quad q_2 \quad q_3 \\
&(b) \quad q_1' \quad q_2' \quad q_3 \\
&(c) \quad q_1 \quad q_2' \quad q_3 \\
&(d) \quad q_2 \\
&(e) \quad q_1' \quad q_2' \quad q_3 \\
&(f) \quad q_1' \quad q_2' \quad q_3
\end{align*}
\]

Clearly, one is free to trivially add a disconnected line e.g. in the defining representation, e.g., along indices 1 and 3 as indicated by the blue line in (18b) with the defining representation labeled as ‘1’ [e.g., one box in a SU(N) Young tableau]. This additional line then may be fused with both, $q_1$ and $q_3$ [(18c)], giving rise (also) to larger multiplets $q_1'$ and $q_3$. The contracted result in (18c) again is a standard rank-3 CGT, but now in $(q_1', q_2' | q_3')$.

The simple procedure above therefore demonstrates that a standard rank-3 CGT in some larger irreps [depicted by the thicker lines in (18d)] can be decomposed via a contraction of three smaller CGTs in a triangular configuration with thinner lines (smaller irreps) [(18e)]. This also includes primitive CGTs, i.e., CGTs that contain the defining irrep on one of their legs [34, 35]. As an aside, note that the triangular configuration represents the minimal TN of rank-3 CGTs that, when contracted, gives rise to another rank-3 CGT.

The simple argument with (18a-c) can be reformulated into
a more general strategy on how to compute standard CGTs recursively for some specific set of larger irreps from CGTs in smaller irreps, based on (18d-f). Suppose one is interested in computing the CGT with larger irreps \((q'_1, q'_2, q'_3)\) in (18d) with \(q'_1, q'_2 > 1\), meaning that \(q'_2\) and \(q'_3\) are not the defining representation (thicker lines). Then one is free to split off the defining irep, such that \((q_1, 1| q'_1)\) and \((q_3, 1| q'_3)\) with \(q_1 < q'_1\) and \(q_3 < q'_3\), in the sense that \(q_1\) and \(q_3\) are smaller either in terms of dimension or, for SU(N), in terms of boxes in a Young tableau. Typically, there will be multiple choices for \(q_1\) and \(q_3\), not all of which may lead to an overall permissible symmetry configuration on the entire triangle considered. Hence one needs to search for permissible triangles of CGTs as in (18e) that have overall smaller irreps in the above sense. Here (18e) resembles (18c), in that the upper line in the triangle is the defining irep. In general, however, the defining irep can reside on any of the sides \(a, b,\) or \(c\) of the triangle in (18f). By reverting the arrow on either \(q_1\) or \(q_2\) (as always, while switching to the corresponding dual representation) permits to have different arrow configuration also on the central triangle. Hence no arrows are shown there.

The strategy above to compute standard rank-3 CGTs for specific \(q\)-labels via a recursive approach is general for non-abelian symmetries from its outline. However, a priori, there is no guarantee that this generates the full OM of the targeted CGT in (18d). While the contraction of the first pair of CGTs in the triangle leads to an enhanced OM, the contraction of the third CGT will typically strongly reduce OM due to the presence of the triangular loop. Overall, therefore multiple triangular arrangements e.g. also with different permissible intermediate smaller multiplets \(q_1, q_3\) may have to be computed to exhaust the full OM of the CGT in (18d). For SU(2), the situation is much simpler. Since it is OM free at the level of rank-3 CGTs, any non-zero triangular configuration suffices, to obtain a valid decomposition of the larger CGT in (18d). For general non-abelian symmetries, however, proving completeness of the resulting OM based on the contraction itself is far less obvious due to the presence of the loop. Therefore it remains an open question in this scheme how one can guarantee that the OM space of the resulting CGT in (18f) is fully explored [38] via a fixed systematic triangular decomposition scheme.

Recursive schemes similar to the one above have been proposed for 3\(j\) and 6\(j\) symbols [34–37]. The general strategy is to iteratively compute larger symbols (here CGTs), e.g., via a triangular decomposition including primitive symbols. However, the primitive symbols need to be computed by other means. While closed expressions exist for these in the case of SU(2), this is not the case for general non-abelian symmetries.

In any case, explicit generation of all encountered CGTs becomes prohibitive for symmetries with large rank \(r\), since typical multiplets grow exponentially with \(r\), in practice like \(\lesssim 10^r\) [see App. A]. Whether these tensors are generated recursively or by other means, eventually always will hit a hard wall. Now given that the X-symbols introduced in this paper represent fully contracted CGT networks up to OM indices, this also makes them susceptible to recursive build-up schemes. This is an attractive route, since X-symbols are typically much smaller than the CGTs involved. For the purposes of this paper, however, this is left as an outlook.

E. Summary

This work introduces X-symbols for the efficient treatment of pair-wise contractions in tensor networks in the presence of non-abelian symmetries. Once computed from CGTs and tabulated, they permit to completely sidestep the explicit usage of CGTs at a latter stage, as they are contracted onto the multiplicity indices of the involved RMTs. X-symbols represent a general framework that is also trivially applicable to abelian symmetries. Much of this paper is a summary of significant extensions that have been implemented and already tested thoroughly in the QSpace v3 tensor library [see App. B]. In this sense, the present paper provides a concise, polished, and proven version of the underlying concepts.

ACKNOWLEDGMENTS

This work has greatly benefitted from a few intensive discussions on non-abelian symmetries in general while visiting Anthony Kennedy (University of Edinburgh, UK). I also acknowledge general discussions and much appreciated support from the group of Jan von Delft in Munich specifically including Seung-Sup Lee, Katharina Stadler, Benedikt Bruognolo, Bin-Bin Chen, Jheng-Wei Li, and Arne Alex; furthermore Wei Li (University of Beihang, China), Wang Yilin (Brookhaven National Lab, US), Thomas Quella (University of Melbourne, Australia), László Borda and Gergely Zaránd (Budapest University, Hungary). This work acknowledges financial support by the Heisenberg fellowship (DFG WE4819/2-1), as well as DFG WE4819/1-1 and DOE DE-SC0012704 without temporal overlap.

[1] Steven R. White, “Density matrix formulation for quantum renormalization groups,” Phys. Rev. Lett. 69, 2863–2866 (1992).
[2] Ulrich Schollwöck, “The density-matrix renormalization group in the age of matrix product states,” Ann. Phys. 326, 96–192 (2011).
[3] Román Orús, “Advances on tensor network theory: symmetries, fermions, entanglement, and holography,” The European Physical Journal B 87, 280 (2014), 10.1140/epjb/e2014-50502-9.
[4] Kenneth G. Wilson, “The renormalization group: Critical phenomena and the Kondo problem,” Rev. Mod. Phys. 47, 773–840 (1975).
[5] Ralf Bulla, Theo A. Costi, and Thomas Pruschke, “Numerical renormalization group method for quantum impurity systems,” Rev. Mod. Phys. 80, 395–450 (2008).
[6] Andreas Weichselbaum, “Tensor networks and the numerical renormalization group,” Phys. Rev. B 86, 245124 (2012).
[7] U. Schollwöck, “The density-matrix renormalization group,” Rev. Mod. Phys. 77, 259–315 (2005).
[8] G. Vidal, “Entanglement renormalization,” Phys. Rev. Lett. 99, 220405 (2007).
[9] J Ignacio Cirac and Frank Verstraete, “Renormalization and tensor product states in spin chains and lattices,” J. Phys. A: Math. Theor. 42, 504004 (2009).
[10] F. Verstraete and J. I. Cirac, “Renormalization algorithms for quantum-many body systems in two and higher dimensions,” arXiv:cond-mat/0407066v1 [cond-mat.str-el] (2004), arXiv:cond-mat/0407066.
[11] Christina V. Kraus, Norbert Schuch, Frank Verstraete, and J. Ignacio Cirac, “Renormalization group algorithms for the entanglement entropy,” Phys. Rev. Lett. 99, 050501 (2007).
[12] Philippe Corboz, Román Orús, Bela Bauer, and Guifré Vidal, “Simulation of strongly correlated fermions in two spatial dimensions with fermionic projected entangled-pair states,” Phys. Rev. B 81, 052338 (2010).
[13] Ho N. Phien, Johann A. Bengua, Hoang D. Tuan, Philippe Corboz, and Román Orús, “Infinite projected entangled pair states algorithm improved: Fast full update and gauge fixing,” Phys. Rev. B 92, 035142 (2015).
[14] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, “Coarse-graining renormalization by higher-order singular value decomposition,” Phys. Rev. B 86, 045139 (2012).
[15] Artur García-Sáez and José I. Latorre, “Renormalization group contraction of tensor networks in three dimensions,” Phys. Rev. B 87, 085130 (2013).
[16] Shi-Ju Ran, Angelo Piga, Cheng Peng, Gang Su, and Maciej Lewenstein, “Few-body systems capture many-body physics: Tensor network approach,” Phys. Rev. B 96, 155120 (2017).
[17] S. Anders, M. B. Plenio, W. Dür, F. Verstraete, and H.-J. Briegel, “Ground-state approximation for strongly interacting spin systems in arbitrary spatial dimension,” Phys. Rev. Lett. 97, 107206 (2006).
[18] Norbert Schuch, Ignacio Cirac, and David Pérez-García, “PEPS as ground states: Degeneracy and topology,” Annals of Physics 325, 2153 – 2192 (2010).
[19] J. Eisert, M. Cramer, and M. B. Plenio, “Colloquium: Area laws for the entanglement entropy,” Rev. Mod. Phys. 82, 277–306 (2010).
[20] T. A. Costi, L. Bergqvist, A. Weichselbaum, J. von Delft, T. Micklitz, A. Rosch, P. Mavropoulos, P. H. Dederichs, F. Mallet, L. Saminadayar, and C. Bäuerle, “Kondo decoherence: Finding the right spin model for iron impurities in gold and silver,” Phys. Rev. Lett. 102, 056802 (2009).
[21] Steven R. White and D. J. Scalapino, “Pairing on striped t-t′-J lattices,” Phys. Rev. B 79, 220504 (2009).
[22] Andreas Weichselbaum, “Non-abelian symmetries in tensor networks: A quantum symmetry space approach,” Annals of Physics 327, 2972 – 3047 (2012).
[23] Sukhwinder Singh and Guifré Vidal, “Global symmetries in tensor network states: Symmetric tensors versus minimal bond dimension,” Phys. Rev. B 88, 115147 (2013).
[24] B. Bauer, P. Corboz, R. Orús, and M. Troyer, “Implementing global abelian symmetries in projected entangled-pair state algorithms,” Phys. Rev. B 83, 125106 (2011).
[25] Tao Liu, Wei Li, Andreas Weichselbaum, Jan von Delft, and Gang Su, “Simplex valence-bond crystal in the spin-1 Kagome Heisenberg antiferromagnet,” Phys. Rev. B (R) 91, 060403 (2015).
[26] Sukhwinder Singh, Robert N. C. Pfeifer, and Guifré Vidal, “Tensor network decompositions in the presence of a global symmetries,” Phys. Rev. A 82, 050301 (2010).
[27] Román Orús, “Tensor networks for complex quantum systems,” Nature Reviews Physics 1, 538–550 (2019).
[28] V. Murg, F. Verstraete, O. Legeza, and R. M. Noack, “Simulating strongly correlated quantum systems with tree tensor networks,” Phys. Rev. B 82, 205105 (2010).
[29] E. M. Stoudenmire and Steven R. White, “Minimally entangled typical thermal state algorithms,” New Journal of Physics 12, 055026 (2010).
[30] Michael Lubusch, J. Ignacio Cirac, and Mari-Carmen Bañuls, “Algorithms for finite projected entangled pair states,” Phys. Rev. B 90, 064425 (2014).
[31] Albert Messiah, Quantum Mechanics (English translation from French) (John Wiley and Sons, 1966).
[32] J.-R. Derome and W. T. Sharp, “Racah Algebra for an Arbitrary Group,” Journal of Mathematical Physics 6, 1584–1590 (1965).
[33] P. H. Butler, “Coupling coefficients and tensor operators for chains of groups,” Trans. Roy. Soc. A (London) 277, pp. 545–585 (1975).
[34] P. H. Butler and B. G. Wybourne, “Calculation of j and jm symbols for arbitrary compact groups. I. Methodology,” Int. J. Quantum Chem. 10, 581–598 (1976).
[35] B GScarle and P H Butler, “Calculation of primitive 6-j symbols,” Journal of Physics A: Mathematical and General 21, 3041–3050 (1988).
[36] Klaus Schulten and Roy G. Gordon, “Exact recursive evaluation of 3j and 6jcoefficients for quanummechanical coupling of angular momenta,” Journal of Mathematical Physics 16, 1961–1970 (1975), https://doi.org/10.1063/1.522426.
[37] James H. Luscombe and Marshall Luban, “Simplified recursive algorithm for wigner 3j and 6j symbols,” Phys. Rev. E 57, 7274–7277 (1998).
[38] J. S. Prakash and H. S. Sharatchandra, “A calculus for su(3) leading to an algebraic formula for the clebschgordan coefficients,” Journal of Mathematical Physics 37, 6530–6569 (1996), https://doi.org/10.1063/1.531750.
[39] Andreas Weichselbaum and Jan von Delft, “Sum-rule conserving spectral functions from the numerical renormalization group,” Phys. Rev. Lett. 99, 076402 (2007).
[40] Bin-Bin Chen, Lei Chen, Ziyu Chen, Wei Li, and Andreas Weichselbaum, “Exponential thermal tensor network approach for quantum lattice models,” Phys. Rev. X 8, 031082 (2018).
[41] A. Weichselbaum, S. Capponi, P. Lecheminant, A. M. Tsvelik, and A. M. Läuchli, “Unified phase diagram of antiferromagnetic su(n) spin ladders,” Phys. Rev. B 98, 085104 (2018).
[42] ITensor Library (version 2.0.11) http://itensor.org.
Appendix A: Symmetry rank and typical multiplet dimension

The rank $\tau$ of a symmetry is an intrinsic property of a given simple non-abelian symmetry that needs to be differentiated from the rank $r$ of tensors. The Lie algebra of a simple non-abelian symmetry possesses at most $\tau$ simultaneously commuting generators which thus defines its rank. This Cartan subalgebra $S^\alpha_\alpha$ with $\alpha = 1, \ldots, \tau$ can be simultaneously diagonalized, and hence gives rise to an $\tau$-dimensional label, or weight space [22]. The Cartan subalgebra is complemented by generalized raising and lowering operators $S^{(\dagger)}_\alpha$ with $\alpha = 1, \ldots, \tau$ which correspond to the simple roots of the Lie algebra.

In practice, then the typical multiplet dimension encountered grows exponentially with the rank $\tau$ as in $|g| \sim 10^\tau$ [22]. The essential reason for this is that multiplets explore an $\tau$-dimensional volume of label space derived from $S^\alpha_\alpha$. This makes large symmetries, such as SU(N) with $t_{\text{SU}(N)} = N - 1$ for $N \gtrsim 4$ computationally challenging. In contrast, the well-known SU(2) is a comparatively very simple yet elementary non-abelian symmetry, in that it corresponds to a symmetry of rank $t_{\text{SU}(2)} = 1$. Therefore a single label suffices (i.e., the spin $S$). For the same reason, there is no inner multiplicity (IM) for SU(2), and also no outer multiplicity (OM) for standard Clebsch Gordan coefficient tensors (rank-3 CGTs).

Appendix B: QSpace v3

The concepts for non-abelian symmetries presented in this paper have been implemented over the past few years in the tensor library QSpace [22]. They were thoroughly tested in a range of papers [6, 22, 25, 39–41] with many more applications via student and other research projects. QSpace was introduced in 2006 [in hindsight as version 1 (v1)] on the level of arbitrary combinations of abelian symmetries. This is effectively the present state of the iTensor Library [42]. QSpace v1 presented a convenient and fast framework for its first strong application in Weichselbaum and von Delft (2007) [39]. QSpace v2 was also able to handle non-abelian symmetries on a generic level, by introducing an additional tensor layer for generalized Clebsch Gordan coefficient tensors (CGTs). It built a database for irreducible representations (R-store), as well as standard rank-3 CGTs (C-store), i.e., with two arrows in and one out. The C-store also stored a listing of the branching rules out of each tensor product decomposition.

However, in QSpace v2 the OM resolved CGTs were explicitly attached to each tensor in full together with the RMTs. Individual entries (fixed symmetry sectors) contained linear superpositions in outer multiplicity, i.e., CGTs of type $\sum_{q'\tau} \epsilon_{\mu
u\rho} C_{q'\tau}$ [cf. discussion after Sec. 5]. This turned out cumbersome since CGTs were re-contracted in every QSpace tensor contraction. Specifically for larger symmetries, e.g., for SU($N \gtrsim 4$), this quickly shifted the dominant numerical cost from the actual physical calculation w.r.t. the RMTs $\|A\|_q$ to the treatment of the CGTs $C_q$.

Therefore for QSpace v3 (developed and thoroughly tested since 2015), not just standard Clebsch-Gordan coefficients of rank-3, but all CGTs of arbitrary rank are computed on demand once and for all, properly orthonormalized [cf. Sec. II B], and stored in the C-store. The QSpace tensor no longer carries the full CGTs but only a reference. The X-symbols for pairwise contractions are also computed on demand once and for all, and stored in an additional database (X-store).

In QSpace, databases are generally built on demand, except for the very elementary initialization when a symmetry is used for the very first time. Preemptive calculation of all possible objects, e.g., up to some prespecified multiplet dimension quickly proliferates to hundreds of thousand of entries, otherwise, even though redundancy in storage has been minimized to a large degree by using sorted CGTs, i.e., sorted w.r.t. their $q$-labels, etc. In the sense that it is impossible to build a complete database for non-abelian symmetries that permit an infinite number of irreps, it is mandatory eventually in any case, to build entries on demand. While it would not really matter for SU(2) or SU(3), since all objects are (re-)computed quickly there, for larger non-abelian symmetries, the concepts of *on demand and once for all* become crucially important for numerical performance. For example, note that starting with SU($N \geq 4$), the typical size of individual multiplets in CGTs quickly surpasses the number of multiplets in RMTs used in a calculation. For example, in DMRG simulations one barely exceeds an effective dimension $D^* \lesssim 8,000$ within the space of RMTs. However, while typical multiplets for SU(N) already reach dimensions up to $\lesssim 10^{N-1} = 1,000$ for SU(4), the largest generated multiplets there already reach $\lesssim 10^N = 10,000$ [cf. App. A].

Now building ones database on demand, quickly makes it dependent on the history of a calculation. Even if CGTs are computed with full OM, there can be an arbitrary orthogonal rotation in OM space. In the absence of OM, this reduces to a simple sign convention. However, for example, in SU(3), the standard rank-3 CGT for $(qq'q)$ with $q = (nn)$ [using Dynkin labels; 22] with $n \geq 0$ has OM $m = n + 1$, and hence can already be made arbitrarily large. Given that larger-rank CGTs are computed on demand via contractions, their specific rotation in OM space necessarily will depend on the specific tensors contracted first, and hence on the history of the calculation. There exists no (at least to author) known convention that naturally and fully fixes the basis of an OM decomposition in all circumstances for CGTs of arbitrary rank.

Standard rank-3 CGTs can be generated in two ways: (i) via explicit tensor product decomposition (‘standard Clebsch Gordan coefficients’), or (ii) via contractions including tensors with rank $r > 3$. (i) is the clearly preferred option, since it generates the full OM at once and in a deterministic and thus well-defined manner. However, this can become prohibitive for large symmetries when requiring e.g. the fusion of two large multiplets both of which can occur on the bonds of a tensor network state, but may be far too large to actually occur as a multiplet at the level of a physical site in the lattice Hamiltonian of interest. In this sense, there is (necessarily) a cutoff in dimension: if a tensor product decomposition is required, e.g., when building a product state space in TNS setting, it needs to be performed anyway. But if one needs to compute a con-
traction that result in a rank-3 CGT, one may opt to perform a full tensor-product decomposition first, and then project the result of the contraction onto it. Or, alternatively, one may be satisfied just with the result of the contraction itself. If the resulting CGT already exists, the result is projected onto it, thus possibly extending the existing OM space. Either way may result in a different basis in OM space. It is crucially important then, that one strictly ensures consistency across ones calculations.

Now given a history dependent database that is accessed and maintained centrally, this implies when running multiple jobs at the same time or when parallelizing within a single job that threads need to be coordinated. That is threads may have to wait, if another thread is currently in the process of updating the same object in the C-store (via contraction or tensor-product decomposition) or the X-store (if a contraction between a new pair of CGTs needs to be performed, or if a new OM component was encountered such that the derived X-symbol needs to be updated). This coordination can be enforced on the level of the database (e.g. locks on affected objects) and also in memory in between different threads (thread locks).

In summary, QSpace v3 consists of three databases (referred to as ‘RCX store’ as a whole)

- R-store for irep representations generated in a calculation; these explicitly include a full basis decomposition in terms of weight labels, a sparse representation of the diagonal Cartan subalgebra (generalized $S_z$ matrices) as well as of the simple roots of the Lie algebra (the generalized raising and lowering operators) under consideration [cf. App. A].

- C-store for storage of all CGTs of arbitrary rank $r \geq 2$. The CGTs are stored in sparse format with typical average sparsity $\gtrsim 10^{-3}$. The sparse format necessarily requires a framework for sparse tensors of arbitrary rank which has been coded from scratch into QSpace v2. The C-store also stores the fusion rules out of full tensor product decompositions, as well as all 1j-symbols, which simply represent a special case of CGTs, namely $(q\bar{q}|0)$.

- X-store for the X-symbols that derive from any encountered pairwise CGT contraction that are not trivially zero due to non-permissible combinations of symmetry labels (e.g. when a CGT contraction were to result in a non-diagonal rank-2 CGTs, or a rank-3 CGTs that was not listed in an earlier full tensor product decom-

The data in the R- and C-store is computed in better than double precision (roughly quad), since the entire RCX store is built iteratively along a TN calculation starting from the very elementary defining representation (and its dual, for convenience). This guards against accumulated error and ensures that all entries are numerically exact in double precision. The X-symbols are computed from CGTs in the C-store, but can eventually be cast into plain double precision as they are contracted onto RMTs anyway.

The C-store for larger symmetries is extremely heterogeneous, as it contains tensors that represent scalars, all the way up to individual CGTs that [e.g. for rank-4 CGTs in SU(4) quickly] require 1TB of space or larger. The X-store contains by far most of the entries. Many contractions are known to be trivially zero since the symmetry labels of the resulting CGT are not permissible from a symmetry point of view, and hence can be excluded from the X-store. Still, e.g., by not insisting that OM spaces are complete, the X-store also contains many X-symbols that are actually zero, meaning that the pairwise contractions of two CGTs results in a CGT of finite dimensions, yet with (Frobenius) norm resembling numerical noise.

When running multiple jobs, it is convenient to maintain a central global RCX-store that has strictly read-only access (except for times when it is updated manually) which contains the bulk of all symmetry related data. In addition, a differential store that is local to each job, allow each job to compute and store whatever is needed in addition. Since the latter is decoupled from other running jobs, at least at this level interference between different simulations leading to possible inconsistencies is avoided. Once an RCX-store is complete for a given calculation, only meta-data is read from the R- and C-store (such as branching rules in tensor products or irep dimensions). For contractions, only the required X-symbols in the X-store need to be read once, with no need to explicitly load the full CGTs from the C-store. The X-symbols are typically much smaller than the involved CGTs, much like 6j symbols.

Since any tensor is stored as a QSpace with the data comprised as the tensor product in Eq. (2), in principle, it has access to all matrix elements in the full state space. This makes QSpace tensors versatile [22] in that all of the elementary tensor operations are allowed that one is used to when performing calculations without symmetries, as long as they do not explicitly break a symmetry. For example, it is very difficult (since inconsistent) to represent a finite magnetic field $BS_z$ if the calculation was initialized with SU(2) spin symmetry. A representation of $S_z$, would require to break up CGTs into specific components which when preserving symmetries, however, are considered inseparable units.