PAPER

Momentum classification of SU(*n*) spin chains using extended Young tableaux

To cite this article: Burkhard Scharfenberger and Martin Greiter 2012 *J. Phys. A: Math. Theor.* **45** 455202

IOP PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND THEORETICAL

J. Phys. A: Math. Theor. **45** (2012) 455202 (18pp) doi:10.1088/1751-8113/45/45/455202

Momentum classification of SU(n) spin chains using extended Young tableaux

Burkhard Scharfenberger[1] **and Martin Greiter**[2]

[1] Institut für Theorie der Kondensierten Materie, KIT, Campus Süd, D-76128 Karlsruhe, Germany
[2] Lehrstuhl für Theoretische Physik 1, Fakultät für Physik und Astronomie, Julius-Maximilians-Universität, Am Hubland, D-97074 Würzburg, Germany

E-mail: burkhard@tkm.uni-karlsruhe.de

Received 2 July 2012, in final form 24 September 2012
Published 19 October 2012
Online at stacks.iop.org/JPhysA/45/455202

Abstract
Obtaining eigenvalues of permutations acting on the product space of N representations of SU(n) usually involves either diagonalizing their representation matrices on total-weight subspaces or decomposing their characters, which can be obtained from Frobenius' formula or via graphical methods using Young tableaux (YT). For products of fundamental representations of SU(n), Schuricht and one of us proposed the method of extended YT (eYT), which allows reading the eigenvalues of the cyclic permutation C_N directly off the, slightly modified, standard YT labelling an irreducible SU(n) representation. Here we generalize the method to all symmetric representations of SU(n), and show that C_N eigenvalue computation based on eYT is at least linearly faster than the standard methods mentioned.

PACS numbers: 02.20.Qs, 03.65.Fd

(Some figures may appear in colour only in the online journal)

1. Introduction

Symmetries, whether discrete or continuous, have been a central concept in physics since its earliest days, and only by making use of them, explicitly or implicitly e.g. by choosing a suitable coordinate system, can most systems be treated analytically or even numerically. This is also true in the study of spin lattice models, a vibrant field of contemporary condensed matter physics that has produced many insights into novel states of matter and manifestations of order. A spin model consists of a cluster of N spins arranged on some lattice tile, usually with periodic, but possibly other, boundary conditions, and a Hamiltonian describing the interaction of the spins with each other or with external fields. The 'spins' transform like some (irreducible) representation of SU(n), usually SU(2), and thus the Hamiltonian acts on a tensor product space whose dimension grows exponentially in N. In recent years, however, models

with higher n have also received attention [1–18], especially since cold atoms in optical lattices hold the prospect of realising SU(n) models experimentally [19–23].

The most commonly considered interaction is of the Heisenberg form $H = \sum_{i<j} J_{ij}\hat{\boldsymbol{S}}_i\hat{\boldsymbol{S}}_j$ between spins on sites i and j with a coupling constant J_{ij}. Such a Hamiltonian is inherently invariant under global SU(2) (SU(n)) rotations and conserves both $S_{\rm tot}$ and $S^{\rm z}_{\rm tot}$ (or, for general SU(n), highest total weight $\boldsymbol{w}_{\rm tot}$ and total weight $\boldsymbol{w}^{\rm z}_{\rm tot}$) respectively. Usually the J_{ij} obey some symmetry relations, often they even possess the full symmetry of the underlying lattice, implying that the Hamiltonian is conserved under all operations in $\mathscr{L}$, the (point) symmetry group of the lattice. Since in numerical studies the lattice is some finite tile containing N sites, $\mathscr{L}$ is a subgroup of S_N, the group of all permutation of N objects.

In treating such a spin lattice model, either analytically or numerically, it seems clear that one should exploit the symmetries of the problem as far as possible. Therefore a product basis, where the z-components of all individual spins provides a complete labelling of all states, while conceptually simple, is not the best choice from a performance perspective. Rather, we should use a basis labelled by the eigenvalues of a maximal commuting subset of $\mathscr{L}$ plus a number of other labels, e.g. the eigenvalues of as many further commuting permutations from S_N as are needed to provide a unique labelling.

A mathematical problem that arises in this context is to determine the eigenvalues of these labelling lattice symmetries. Changing language from SU(2) to SU(n) the general problem can be stated like this: for arbitrary N-fold product spaces $V_{\boldsymbol{v}}^{\otimes N}$ of some irreducible representation (irrep) $V_{\boldsymbol{v}}$ of SU(n), find the eigenvalues the labelling symmetries L. The $(n-1)$-dimensional vector $\boldsymbol{v}$ is the highest weight of the irrep and has the same meaning for SU(n) as spin for SU(2).

There are two traditional ways to solve this computationally. We know that the tensor product decomposes into a direct sum of irreps of SU(n):

$$V_{\boldsymbol{v}}^{\otimes N} = \bigoplus_{\boldsymbol{w}} V_{\boldsymbol{w}}^{\oplus a_{\boldsymbol{w}}}. \tag{1}$$

Say we want to know the eigenvalues for our labelling lattice symmetries on the subspace $V_{\boldsymbol{w}}^{\oplus a_{\boldsymbol{w}}}$ of all irreps $V_{\boldsymbol{w}}$ of SU(n) (i.e. $V_{\boldsymbol{w}}$ appears $a_{\boldsymbol{w}}$ times in $V_{\boldsymbol{v}}^{\otimes N}$). Then one way is to use character theory to determine the irreps of S_N contained in $V_{\boldsymbol{w}}^{\oplus a_{\boldsymbol{w}}}$. The eigenvalues of the permutations are then obtained by diagonalizing their representation matrices in these irreps, which is possible for all of them simultaneously since they commute. Alternatively, we could simply write down all product states $\phi_{\boldsymbol{w}} \in V_{\boldsymbol{v}}^{\otimes N}$, which have the total weight $\boldsymbol{w}^{\rm z}_{\rm tot} = \boldsymbol{w}$ (in the case of SU(2) this corresponds to a fixed $S^{\rm z}_{\rm tot}$ subspace), determine the representation matrices of the labelling symmetries and again diagonalize them (all simultaneously). In both cases, we need to repeat the process for those highest weight multiplets $\boldsymbol{w}' > \boldsymbol{w}$, that are contained in the subspace of the total weight $\boldsymbol{w}^{\rm z}_{\rm tot} = \boldsymbol{w}$. In the case of SU(2) for instance, it is sufficient to consider the next higher $S^{\rm z}_{\rm tot}$ subspace, i.e. $S^{\rm z}_{\rm tot} = S + 1$, and disregard all sets of eigenvalues which appear in both subspaces.

Young tableaux (YT) are a diagrammatic technique originally invented to compute various properties of irreps of the permutation group S_N [24], but they and other techniques based on them have since seen a myriad of uses in both mathematics [25, 26] and physics [27, 28]. Their usefulness is mostly related to the Schur–Weyl duality, which makes a direct connection between irreps of GL(n) and S_N contained in tensor products of some elementary GL(n) irrep. Due to this, YT also provide an elegant means of obtaining the decomposition of $V_{\boldsymbol{v}}^{\otimes N}$ into irreps of SU(n), i.e. of obtaining the $a_{\boldsymbol{w}}$ in (1).

A further use of YT was introduced in a 2007 paper [29] (see also [30]), by Schuricht and one of us: the method of extended YT (eYT). It allows the spinon content of an eigenstate of the Haldane–Shastry model for a chain of N fundamental SU(n) spins to be read off directly

from slightly modified YT. The interesting consequence we want to point out here, is that this also allows one to find the state labels $|\boldsymbol{w}_{\text{tot}}, p$ for the symmetry group of the 1D chain. This group is the cyclic group $\mathscr{C}_N$ generated by the single-site translation C_N where each eigenvalue γ of C_N can be identified with a momentum p along the chain via the equation $\gamma = \exp[\mathrm{i}p]$. To our knowledge, this is the only such method working directly with YT and as a practical consequence, this enables a significant speed-up of C_N-eigenvalue computations.

In this paper we generalize the method of eYT to higher representations of SU(n) and present numerical evidence that it indeed gives the correct results for the eigenvalues of C_N, as long as the representations V_σ which are coupled are *symmetric*, i.e. correspond to YT with a single row. We also find, that while it does give the correct distribution of momenta on the subspace $V_\lambda^{\oplus a_\lambda}$ of all multiplets λ, it does not assign these momenta to the individual YT in a way that would allow deducing the irreducible S_N-representation content of $V_\lambda^{\oplus a_\lambda}$. A positive result would both have been quite useful in itself and would also have given a physical meaning to individual YT, similar in spirit to the connection between YT and angular momentum states sought by McAven and Schlesinger [31]. Lastly we show how working directly with the YT of an irrep speeds up the computation of C_N eigenvalues over traditional methods by at least a linear factor N.

This paper is organized as follows. In section 2 we briefly review the method of YT for S_N and how it relates to the irreps of SU(n). In section 3, we restate the extension procedure for fundamental representations and reformulate it in a way better suited to both numerical implementation and generalization to product spaces of higher representations. We complete this task in section 4, which contains the main result, namely that the procedure appears to work for *all symmetric higher SU(n) representations*. Furthermore, we comment on the relation between the eYT method and irreps of S_N contained in the product spaces. In section 5 we compare the computational complexity of the extension procedure to that of generic methods of obtaining the eigenvalues of C_N. Finally, we summarize our results in section 6.

2. Young tableaux and SU(n)

We begin with a short summary of YT and some of their traditional uses.

A *Young diagram* or *shape* is a graphical depiction of an integer partition

$$(\lambda) = (\lambda_1, \lambda_2, \ldots, \lambda_k),\ \lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_k > 0,$$
$$|\lambda| := \sum_j \lambda_j = N, \tag{2}$$

as k left-justified rows of boxes, where row j has λ_j boxes in it. A YT on the shape λ is any filling of the boxes with integers between $1, \ldots, N$ (see figure 1(a) and 1(b)). Counting YT subject to certain building rule is what is at the heart of their application in the representation theory of both the symmetric group S_N, the group of all permutations of N distinguishable things, and SU(n), the group of special unimodular, complex $n \times n$ matrices.

S_N. The irreps of the symmetric group S_N can be labelled by integer partitions λ, $|\lambda| = N$. Originally, YT where invented to provide a graphical method of computing the character $\chi^\lambda(P)$ of an element $P \in S_N$ in the irrep λ [24]. Of special significance is the character of the identity, since it is equal the dimension of an irrep: $\chi^\lambda(\mathrm{id}) = \dim(\lambda) =: N_\lambda$. This dimension can be determined by counting all *standard YT* on the shape λ. A tableau is called standard if the numbers in its boxes are strictly increasing in both rows and columns. Figure 1(c) for instance shows all standard YT on $\lambda = (4, 2)$. There are nine of them, therefore the representation $(4, 2)$ of S_6 must be nine dimensional.

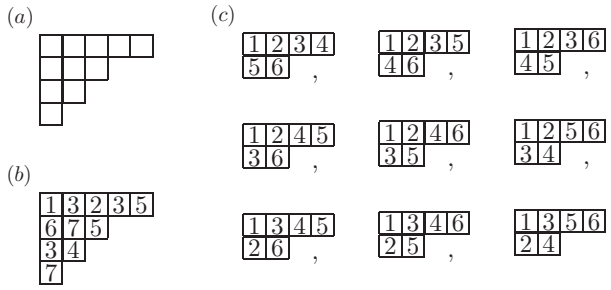

Figure 1. (*a*) The Young diagram or shape to the partition (5, 3, 2, 1). (*b*) The same diagram as a (general) Young tableaux. (*c*) All standard Young tableaux on the shape (4, 2).

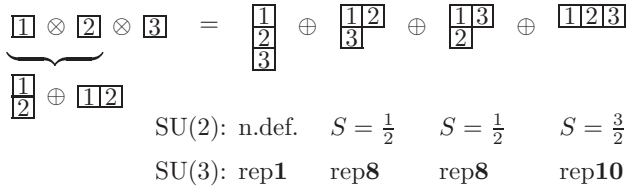

Figure 2. Building higher irreducible representations from the fundamental one via branching in the case of SU(2) and SU(3). Counting dimensions as a check, we see that the decompositions are complete in both cases: $2^3 = 2 + 2 + 4$ and $3^3 = 1 + 8 + 8 + 10$.

Special unitary group SU(n). In the context of SU(n) YT can be applied to decompose tensor products of representations. An irrep of SU(n) is characterized by its highest weight, where a weight is the ($n-1$)-dimensional vector of eigenvalues of the simultaneously diagonalizable group generators spanning the ($n-1$)-dimensional Cartan sub-algebra of su(n), the generating Lie algebra of SU(n). A weight $\boldsymbol{w} = (w_1, w_2, \ldots, w_{n-1})$ is said to be higher than a weight $\boldsymbol{w}' = (w'_1, \ldots, w'_{n-1})$ if the first nonzero entry in $\boldsymbol{w} - \boldsymbol{w}'$ is positive. In the well known case of SU(2), for instance, irreps are characterized by their spin S, which can be integer or half-integer ($S = \frac{1}{2}, 1, \frac{3}{2}, \ldots$), and the highest weight corresponds simply to the highest possible value of S^z, which is $S^z = S$. Since SU(n) is defined as a matrix group, one representation is always the group itself, its carrier vector space being $\mathbb{C}^n$. It is irreducible and of special interest, because by forming tensor products of multiple F_n and projecting onto subspaces of appropriate symmetry we can form all irreps of SU(n), This is the main consequence of the already mentioned Schur–Weyl duality.

Thus, Schur–Weyl duality in effect implies that one can use YT to decompose tensor products of F_n (or indeed higher SU(n) representations). Associating F_n with a single box Young diagram, there is a neat diagrammatic way to do this: we simply construct all N-box standard YT with no more than n rows, which can best be done using the branching rule (see e.g. [25]). The process is illustrated in figure 2. This also means there is a 1–1 correspondence between an $n-1$ dimensional highest weight $\boldsymbol{w}$ and an $(n-1)$-row shape λ, so we can from now on use shapes not only to index S_N- but also SU(n)-irreps.

It is possible to generalize this procedure to decompose the product space $V_\sigma^{\otimes N}$ of arbitrary irreps associated with the shape $\sigma = (\sigma_1, \ldots, \sigma_l)$. In the case of single-row σ (*symmetric representation*, $l = 1$), which is all we will need in this paper, this generalization is straightforward. We again use the branching rule, i.e. add boxes step by step, but now each

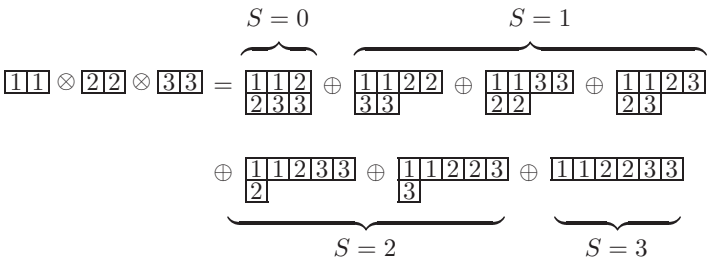

Figure 3. Decomposition of a $3\times(S=1)$ product space of SU(2).

number $j = 1, \ldots, N$ appears $|\sigma|$ times instead of only once and we have to take care not to put two boxes with the same number on top of each other (see figure 3).

As we have stated, the number N_λ of standard YT on the shape λ equals both the dimension of the irreducible S_N representation labelled with λ and the multiplicity of the irreducible SU(n) representation associated with λ (via the correspondence highest weight $\leftrightarrow$ Young diagram) tensor product $F_n^{\otimes N}$ of the fundamental representation of SU(n). This just another consequence of the Schur–Weyl duality: one can show that the subspace $V_\lambda^{N_\lambda}$ in the tensor product $\mathrm{F}_n^{\otimes n}$ always forms an irrep of the permutation group S_N equivalent to the irrep labelled by the integer partition λ.

For an example we can again consider figure 2, the two YT of shape (2, 1) tell us, as mentioned, that $F_2^{\otimes 3}$ contains two doublets but also, that the states of these doublets (for each fixed S^z_{tot}) transform like a standard representation (2,1) under the action of the group S_3.

3. Extended Young tableaux

We now turn to the problem described in the introduction of computing the eigenvalues of the cyclic permutation C_N on total spin subspaces $V_\lambda^{\oplus N_\lambda}$. Let us first review the extension procedure for YT of fundamental representations introduced in [29].

Rule, original version. Let T be a standard YT of size N. By sliding them to the right where necessary, arrange all boxes of T such that in each column of the resulting *extended* tableau the numbers in the boxes are in sequence (i.e. i above $i+1$ above $i+2$ etc). This will often require leaving empty spaces between boxes. Mark each by a dot (see figures 4 and 5). To each dot i we assign a number a_i in such a way that the average of all a_i within one column equals the average of the numbers in all boxes in that column, where the a_i have integer or half-integer values with a spacing of 1 between the numbers from one column.

Haldane–Shastry model. This version of the rule betrays the origin of the procedure: it comes from the physical problem of the Haldane–Shastry spin chain [32, 33], which consists of N spins on a circle with a Heisenberg-type $J_{ij}\hat{\boldsymbol{S}}_i\hat{\boldsymbol{S}}_j$ interaction where the coupling $J_{ij} = |\eta_i - \eta_j|^{-2}$ decreases quadratically in the chord distance. In the original model the $\hat{\boldsymbol{S}}_i$ where $S = 1/2$ SU(2) spins, but it has been generalized to fundamental irreps of SU(n) [4, 5, 34, 35]. This fully integrable model [36] has a singlet ground state and the excitations are spinons, which can be thought of as delocalised domain walls (half a spin flip) in a background liquid with strong antiferromagnetic short range correlations. By interpreting each dot in the eYT as a spinon, the eYT allow obtaining the spinon content of the eigenstates (which also

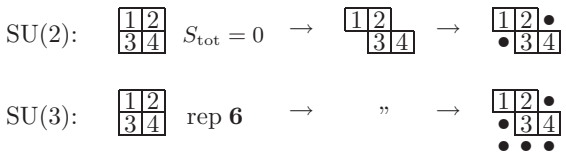

Figure 4. A simple example of the extension procedure: the lower row slides to the right, s.t. '2' is above '3'. How many dots are placed depends on the SU(n) under consideration.

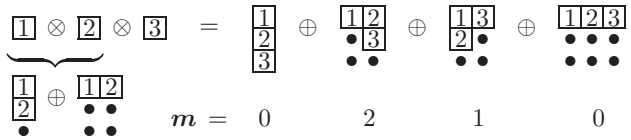

Figure 5. Building extended tableaux box-by-box. Since we consider SU(3), three rows are marked with dots. However, the momenta assigned do not depend on n.

conserve total spin and total momentum) and moreover assign each spinon a momentum number p_i connected to the a_i from above via

$$p_i = 2\pi\,(a_i - 1/2)/N. \tag{3}$$

The total momentum of a state is obtained by summing over all individual spinon momenta while the energy is essentially the sum of the squares [29]. In both quantities we need to include the constant offset p_0 and p_0^2 respectively given by

$$p_0 = \pi\,\frac{n-1}{n}N. \tag{4}$$

The original rule is a succinct formulation of the basic idea, but it is not well suited to implementation on a computer, nor does it generalize directly to higher SU(n) irreps. If one wishes to use eYT for computing eigenvalues of C_N on tensor product spaces of higher SU(n) irreps, it is better to make use of the branching rule. Let us therefore reformulate the building rule.

Rule, new version. Given a standard YT T, we start with an incomplete extended tableaux $E_1(T)$ containing only the single box labelled '1'. We then add a box labelled '2' to E_1 by looking whether in T, '2' appears in the first or second row. In the former case we add '2' to the right of '1', while in the latter case we put it below '1'. This yields an, i.g. still incomplete, 2-box extended tableaux which we call E_2. We go on building the full extended tableau $E(T)$ step-by-step. In step k, having constructed the extended tableaux E_k, we obtain the next one, E_{k+1}, in a similar way as E_2: we look up in T the row index r_{k+1} of the box '$k+1$' and compare it to the one of 'k', which is r_k. If now $r_{k+1} \leqslant r_k$, we add a new column at the right side of E_k and put the box '$k+1$' in its r_{k+1}th row. Otherwise, i.e. if $r_{k+1} > r_k$, we add '$k+1$' into same column as 'k', also in the r_{k+1}th row. The resulting extended tableaux we call $E_{k+1}(T)$. After a total of $N-1$ additions we thus arrive at the final tableaux $E(T) = E_N(T)$.

This procedure is now directly implementable and above all, generalizes to products of higher (symmetric) representations. To compute the total momentum p_T associated to the tableaux T, we combine the spinon momentum numbers a_i in each column c of the extended tableau $E(T)$ into one *column number* b_c

$$b_c = \sum_{i\in c}\left(a_i - \frac{1}{2}\right). \tag{5}$$

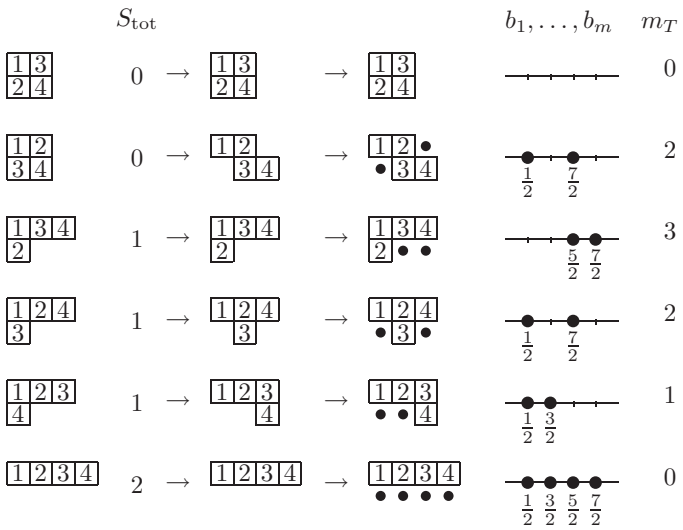

Figure 6. The complete list of extended Young tableaux for $N = 4 \times (S = 1/2)$ spins of SU(2), m_T is the integer momentum number $4p_T/2\pi$.

Written directly in terms of the average of the box-labels $\langle i \rangle_c$ and the number of boxes k_c in the column, the b_c are:

$$b_c = (n - k_c)(\langle i \rangle_c - 1/2) \tag{6}$$

the momentum p_T associated with T is then simply

$$p_T = \frac{2\pi}{N}\frac{1}{n}\left(b_0 + \sum_{c \in E(T)} b_c\right). \tag{7}$$

The momentum offset $b_0 := -(n-1)N^2/2$ is still necessary to ensure that the sum (7) is a multiple of n. Thus no matter which SU(n) a tableau T pertains to (although clearly one must have $n >$ no. of rows in T), it is always assigned the same momentum by our procedure.

In figure 6 we show as an example the extension procedure for all total spin multiplets of the tensor product space $\left(\frac{1}{2}\right)^{\otimes 4}$ of four $S = 1/2$. Since this is the tensor product of a fundamental representation, the shapes of the YT immediately tell us what irreps of S_4 the total spin multiplets belong to. It is thus easily verified, that our procedure gives the correct values. The lone quintet $S^{\rm tot} = 2$ must be fully symmetric and has therefore momentum 0, the three triplets $S^{\rm tot} = 1$ form the standard representation of S_4 (of dimension 3 and associated with the partition $(3, 1)$) while the two singlets belong to the self-conjugate irrep $(2, 2)$ (two dimensional).

The Haldane–Shastry model is valid for $\boldsymbol{S}_i \in {\rm SU}(n)$ not only for $n = 2$ and the mechanism of constructing excitations remains the same. Therefore this connection of eYT and HSM eigenstates exists not only for SU(2) but higher n as well and since eYT correctly describe the eigenstates of the Haldane–Shastry model in all cases, it provides the strongest argument in favour of the correctness of our procedure. A rigorous mathematical proof would still be desirable however.

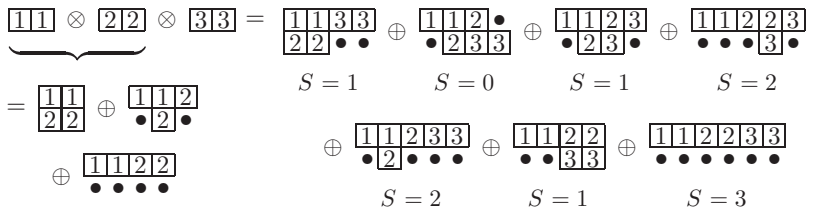

Figure 7. Higher representation eYT can be built box-wise as well. Ambiguity below which of two identical k-boxes to place a $(k+1)$-box requires an additional rule.

4. Higher representations

While the spins in the Haldane–Shastry model can transform like any fundamental SU(n) representations, not just SU(2), a straightforward generalization to higher representations runs into difficulties. One can nonetheless hope, that the *mathematical* statement remains valid for a suitable generalization of the rule, which is what we investigated. It turns out, that for products of symmetric SU(n) representations V_σ (represented by single row YT with $|(\sigma)| = \sigma$ boxes) the box-by-box approach to building extended tableaux generalizes almost directly (see figure 7).

We have to introduce only one additional condition coming from an ambiguity in where to put a box '$k+1$' if there are several eligible open columns with boxes 'k'. Given a tableau T representing a multiplet in $V_\sigma^{\otimes N}$ (i.e. each number appears $|\sigma|$ times) we require that the resulting extended tableau $E(T)$ is *minimal*, i.e. has as few dots/empty spaces as possible. The way to achieve this is to consider the boxes with number '$k+1$' *in increasing order of their row index in* T, i.e. place higher boxes first.

Consider the fourth YT from the top in figure 8. Assume we have already placed all '1's and '2's, giving us an (incomplete) extended tableaux with three columns, the first containing '1' above '2', in the second and third a '1' and a '2' respectively in the first row. If we now were to place the '3' from the lowest row first, then we have two eligible columns (the first and third). Depending on our decision, we would end up with two different extended tableaux, one with momentum $p = 0$ the other with $p = 1$. Placing the higher '2' (the one from the second row) first, there is no ambiguity, and we identify the eYT with $p = 1$ as the correct minimal one.

The only further change is a that the momentum offset b_0 acquires a factor $|\sigma|$:

$$b_0(\sigma) = -\frac{n-1}{2} N|\sigma|. \tag{8}$$

The $|\sigma|$ dependence arises naturally as is explained in appendix C.

Figures 8 and 9 show examples of the extension in the case of $|\sigma| = 2$. In the absence of a rigorous mathematical understanding *why* it works and without the physical interpretation that backs up the eYT procedure in the case of fundamental SU(n) representations, we checked the statement numerically for several SU(n) up to $n = 4$ and $N = 16$ and find it does give the correct eigenvalue distributions (see table 1).

Figure 9 illustrates a limitation of our method. As mentioned, multiplet subspaces $V_\lambda^{\oplus a_\lambda}$ in $V_\sigma^{\otimes N}$ comprise in general more than one irrep of S_N. Only for products of fundamental representations ($|\sigma| = 1$) do irreps of S_N and SU(n) coincide. The $a_{S=2}$ = six-dimensional subspace of all quintets in $(S = 1)^{\otimes 4}$ for instance (i.e. $V_{S=2}^{\oplus a_{S=2}}$) contains the three S_4 irreps (3, 1), (2, 2) and (4). If we try an ad-hoc identification of these irreps with the YTs based on the latter's structure (see figure 9), we see that the momenta assigned by our procedure and the momenta one would expect from this identification do not match: the three tableaux

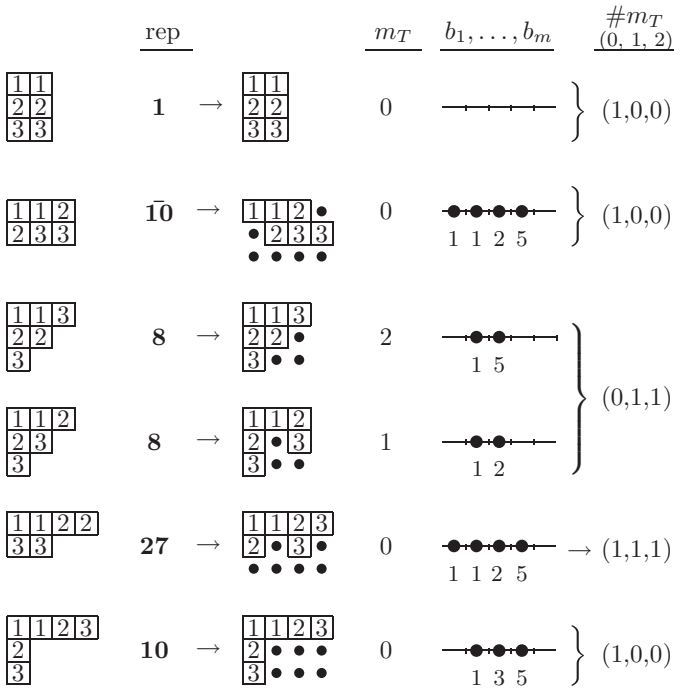

Figure 8. Here are some of the eYT we find when combining $N = 3$ rep **6** ($\hat{=}(2, 0)$) of SU(3). The last column gives the tally of momenta for the shape.

Table 1. Unitary groups SU(n) and maximal tensor powers N for which we verified the correctness of extended Young tableaux.

Group	Rep.	Shape σ	$N_{\max}$	Dim($V^{\otimes N}$)
SU(2)	$S = 1/2 \,\hat{=}(1)$	□	16	65 536
	$S = 1 \,\hat{=}(2)$	□□	14	4782 969
SU(3)	$\underline{3} \,\hat{=}(1, 0)$	□	12	531 441
	$\underline{6} \,\hat{=}(2, 0)$	□□	9	10 077 696
	$\underline{10}_3 \,\hat{=}(3, 0)$	□□□	7	10 000 000
SU(4)	$\underline{4} \,\hat{=}(1, 0, 0)$	□	10	1048 576
	$\underline{10}_4 \,\hat{=}(2, 0, 0)$	□□	7	10 000 000

we identified as belonging to irrep (3, 1) are assigned momenta 2, 3 and 0, while one would expect 1, 2 and 3.

Thus, while eYT do produce the correct frequencies of momenta for each subspace $V_{\boldsymbol{\lambda}}^{\oplus a_{\boldsymbol{\lambda}}}$ *as a whole*, they give no help in identifying the S_N irrep content of multiplet subspaces (beyond what the momentum frequencies themselves already reveal).

5. Fast tableaux generation

In this section we want to elaborate how working directly with the YT, allows a useful speed-up of C_N eigenvalue computations in the limit of large N and n.

In the introduction, we already mentioned briefly two traditional computational methods for obtaining the eigenvalues of the cyclic subgroup generator C_N. They are character theory and diagonalization of the matrix of C_N on total weight representations. Both begin by writing

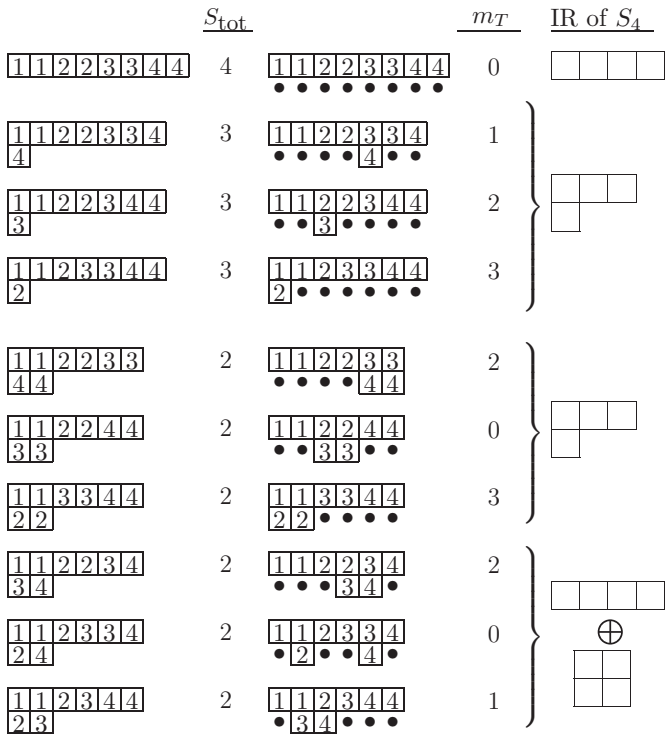

Figure 9. The complete decomposition of $V = S = 1^{\otimes 4}$ for $S_{\mathrm{tot}} \geqslant 2$ and momenta assigned by our method. The right column shows the ad-hoc identification of YT with the irreps of S_4 as described in the text.

down a product-state basis $\mathscr{B}_{\boldsymbol{w}}$ of a total weight subspace $\boldsymbol{w}^{\mathrm{z}}_{\mathrm{tot}} = \boldsymbol{w}$. In the case of $(S = 1)^{\otimes 4}$ of SU(2) for instance this would be the ten dimensional space spanned by

$$C_4^i \,|1, 1, 1, -1\rangle \,, C_4^j \,|1, 1, 0, 0\rangle \,, C_4^k \,|1, 0, 1, 0\rangle \tag{9}$$

where $i, j = 0, \ldots, 3$, $k = 0, 1$ and the cyclic permutation C_4 is applied to a state in the natural way. Clearly, these states form the basis of a representation of S_4.

Character theory. There are in fact two methods based on group characters: one working with the characters of S_N and another, simpler one, using the characters of $\mathscr{C}_N$.

The former, mentioned here for the sake of completeness, obtains the S_N character $\boldsymbol{\chi}_{\boldsymbol{w}}$ of the total weight representation spanned by $\mathscr{B}_{\boldsymbol{w}}$ (referred to it simply as *the* representation $\mathscr{B}_{\boldsymbol{w}}$ from now on), i.e. we compute the trace of the representation matrix of one element from each conjugacy class in S_N and then decompose this compound character using the formula $a_\lambda = \frac{1}{N!} \sum_{[P]} |[P]| \chi_\lambda(P) \chi_{\boldsymbol{w}}(P)$, where the sum runs over all conjugacy classes $[P] \subset S_N$ where P is some representative of the class and $|[P]|$ is its cardinality. The eigenvalues of C_N follow directly, since each irrep λ comes with a fixed set of eigenvalues.

One arrives at more efficient way of using characters by realising that $\mathscr{B}_{\boldsymbol{w}}$ is also a representation of $\mathscr{C}_N$, which means we can apply character decomposition directly to the representation matrices of $C_N, C_N^2, \ldots, C_N^N$. We can thus compute the multiplicity f_m of a momentum number m via the group characters of $\mathscr{C}_N$:

$$f_m = \frac{1}{N} \sum_{k=1}^{N} \exp\left[\frac{2\pi\,\mathrm{i}}{N}\, mk\right] \,\mathrm{Tr}\, C_N^k \tag{10}$$

where $m = 0, \ldots, N-1$ labels the irreps and $k = 1, \ldots, N$ the classes in $\mathscr{C}_N$ and $\mathrm{Tr}\, C_N^k$ is the trace of the $n_{\boldsymbol{w}} \times n_{\boldsymbol{w}}$ representation matrix of the kth power of C_N.

This is both faster than the full character decomposition and does not assume prior knowledge of all the irreducible S_N characters (which would in practice have to be computed too). We do however have to generate all the powers C_N^k of C_N, which takes (at least) $O(N\, n_{\boldsymbol{w}})$ steps.

We are not done yet however, for remember that the total weight representation $\mathscr{B}_{\boldsymbol{w}}$ contains not only the irrep $\boldsymbol{w}$ but also some with highest weight $\boldsymbol{w}' > \boldsymbol{w}$, which we need to sift out. The SU(2) case is straightforward: we simply run the procedure twice, once for $S^{\mathrm{z}}_{\mathrm{tot}} = S$ and once for $S^{\mathrm{z}}_{\mathrm{tot}} = S+1$ and then subtract the S_N-momentum tally of the latter from that of the former. The general case requires more work however: first, we need to know the positive integers $c_{\boldsymbol{w}\boldsymbol{w}'}$ recording how many states a representation $\boldsymbol{w}'$ contributes to $\boldsymbol{w}$. The fact that only $\boldsymbol{w}' > \boldsymbol{w}$ contribute and $\boldsymbol{w}$ is contained exactly once means that viewed as a matrix, (c) will be upper triangular with only 1s on the main diagonal. To obtain the tally of all momenta for $\boldsymbol{w}^{\mathrm{tot}}$, we then need to take linear combinations of a certain number $h_{\boldsymbol{w}}$ of rows of this matrix, such that, all contributions of higher multiplets are cancelled.

Thus, total asymptotic complexity is

$$\mathrm{C}_{\mathrm{char},C_N} = \mathrm{O}\,(N\, h_{\boldsymbol{w}}\, n_{\boldsymbol{w}})\,. \tag{11}$$

This is however still not as good as the conceptually simple diagonalization we will turn to next.

Diagonalization. Diagonalization is straightforward: we write out the representation matrix of C_N in the basis $\mathscr{B}_{\boldsymbol{w}}$ and diagonalize it. In general, diagonalization is of (time) complexity $\mathrm{O}(m^3)$ for an $m \times m$ matrix, but since we are dealing with permutation matrices (in each row and column all entries are zero except for exactly one '1'), $\mathrm{O}\,(m)$ steps suffice. Like with the previous character methods, we will also obtain C_N eigenvalues belonging to SU(n) irreps of higher highest weight which can be got rid of in the same way, incurring the same $h_{\boldsymbol{w}}$ factor.

In the end therefore, diagonalization is faster than the based character methods and if we assume that the representation matrix of C_N can be written down in $O(n_{\boldsymbol{w}})$ steps the final time and space requirements are

$$\mathrm{C}_{\mathrm{diag}} = \mathrm{O}\,(h_{\boldsymbol{w}}\, n_{\boldsymbol{w}})\,. \tag{12}$$

Extension. The extension procedure on the other hand works directly with the N_λ YT on a shape λ, assigning each a momentum number $m(T) = 0, \ldots, N-1$. The key to making it superior to the other methods, is that it is possible to combine YT creation, extension and momentum computation efficiently into one procedure.

Let us first consider YT generation: one should exploit the branching property somehow, but a naive ansatz building up tableaux by adding box after box starting from scratch for each tableaux will require $\mathrm{O}\,(N\, N_\lambda)$, which will only be marginally better than diagonalization in the most interesting cases (the states with low total highest weight, e.g. SU(n) singlets) and due to the more intricate nature of the algorithms involved probably turn out to be somewhat slower in the less interesting ones (states with total highest weight close to the completely symmetric one).

We can achieve a complexity of $\mathrm{O}\,(N_\lambda)$ however, if we store the branching information in a suitable way: the branching graph (BG). It encodes the relations between a shape λ and all (valid) shapes $\mu \subset \lambda$ derivable from it by repeated regular removal of elementary shapes σ in the form of a directed graph, where σ was the shape associated with the SU(n) irrep from which we build our product space.

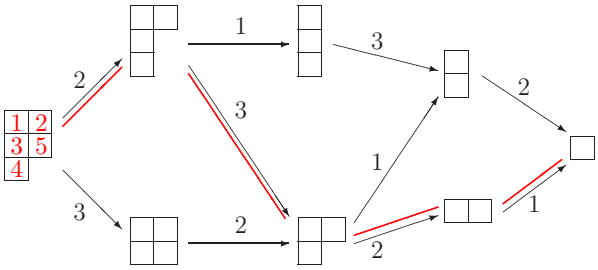

Figure 10. The standard YT branching graph of the shape $\lambda = (2, 2, 1)$. For all depth j the arrow-labels denote into which line the index $N - j + 1$ is to be put. Thus, the paths through the BG correspond 1–1 to all standard YT. An example of such a path and the YT it corresponds to is shown in red.

The nodes of the graph are the shapes $\mu \subseteq \lambda$ (with λ being the root) and a labelled edge $(\mu \to \nu; l)$ goes from shape μ to ν if and only if the latter can be obtained from the former by a regular removal of one elementary shape σ. A regular removal is the inverse of a regular addition, which is defined as the addition of $|\sigma|$ boxes such that the resulting tableau is valid. The label l will be a list of length $|\sigma|$, recording into which row we put the first, second, third,..., $|\sigma|$th box. An example of a BG for the standard YT on shape $(2, 2, 1)$ is depicted in figure 10. Since $|\sigma| = 1$, the label consists of a single row-index only.

As long as σ is a single-row tableau, as we always assume here, there will be at most one edge between nodes. However, the BG is also defined for tableaux built from multi-row-σ (non-symmetric SU(n) representations), but there it can happen that more than one edge leads from one node to the another (they will differ in their label however). Irrespective of the elementary tableau σ, each node in a BG can be assigned a depth, i.e. a unique distance from the root, and it also holds that all BGs have a unique lowest node (leaf) given by the elementary shape σ itself.

Computing the BG of a compound shape λ with k rows for some elementary shape σ (where $|\lambda| = N|\sigma|$) requires $\mathrm{O}\big(\binom{k+|\sigma|}{k} D_\lambda\big)$ steps, where D^σ_λ is the number of shapes μ obtainable from λ by regular removal of σ. It can be estimated by (see appendix B)

$$D^\sigma_\lambda \leqslant \sum_{\lambda_1 \geqslant j_1 \geqslant \ldots \geqslant j_k \geqslant 0} 1 = \binom{\lambda_1 + k}{\lambda_1}. \tag{13}$$

The leading contribution is $D^\sigma_\lambda = \mathrm{O}(N^k)$ (because the first row $\lambda_1 < |\sigma|N$) and thus we find that BG generation takes

$$\mathrm{C_{BG}} = \mathrm{O}(k^{|\sigma|} N^k) \tag{14}$$

time. We should point out that this is in general not polynomial, as it might appear at first glance. Since k is not independent of N, for e.g. a square shape $N = k^2$ we indeed have $\mathrm{C_{BG}} = \mathrm{O}(\sqrt{N}^{|\sigma|} \exp[\sqrt{N}])$. We will still profit from using the BG however, because even in these cases N_λ grows much faster still and thus dominates the total complexity of computing the momenta (for more details see appendix B).

One can now use the efficient graph iteration described in appendix A to traverse all paths through the BG, simultaneously building up the eYT as we go. It is necessary to compute this both at once, because a modularized approach of extracting the paths first and then translating them one by one into eYT incurs an additional $\mathrm{O}(N)$ time factor coming from the fact that each path is of length $N - 1$.

The total asymptotic complexity (in both *time* and *memory*) achievable is therefore indeed determined purely by the number of YT on λ

$$\mathrm{C}_{\mathrm{extendedYT}} = \mathrm{O}\,(N_\lambda)\,. \tag{15}$$

How much is this superior to diagonalization? The biggest differences occur for low-weight SU(n) representations (e.g. singlets), and for these, the number of all multiplets N_λ grows slower than n_λ, the size of the corresponding total weight space.

Take for instance N spin $S = 1/2$ (N even): there are $\binom{N}{N/2}$ $S_z^{\mathrm{tot}} = 0$ states but only

$$\binom{N}{N/2} - \binom{N}{N/2-1} = \frac{2}{N+2}\binom{N}{N/2}$$

$S^{\mathrm{tot}} = 0$ singlets.

In addition, the other methods incur the factor $h_{\boldsymbol{w}(\lambda)}$ because they need to repeated for higher weights, as described above. This factor, while trivial for SU(2), becomes increasingly important for larger n .

6. Conclusion

We have demonstrated how the extended Young tableaux (eYT) method of calculating the eigenvalues of C_N, the generator of the cyclic subgroup $\mathscr{C}_N \subset S_N$, can be used not only for product spaces of fundamental SU(n) representations (associated with a single-box Young diagram), but for those of higher ones as well, if they are symmetric, i.e. correspond to single-row diagrams.

Furthermore, since eYT derive directly from the YT on a shape λ, it is possible by exploiting the branching rule to speed up the computation of C_N eigenvalues on $V_\lambda^{\otimes a_\lambda}$ to an asymptotic complexity O (N_λ) in time and memory.

Acknowledgment

BS was supported by the Landesgraduiertenförderung Baden-Württemberg.

Appendix A. Efficient graph iteration

The branching graph (BG) is the key data-structure for implementing fast Young tableaux (YT) generation, but the form presented above is not yet sufficient to allow an efficient iteration over all paths through it. We need to both add some additional information to the it and use suitable data-structures to guide the iteration itself.

To illustrate: taking the BG as it is, we could for instance perform a depth-first iteration: we use a size $N-1$ (= max. depth) array $c[.]$ to record to which child we descend to from the node $\mu[c](d)$ at the depth d. In each step we then descend one level further down the graph until we reach the leaf and there, having found a new path from root to leaf, we add it to our result and backtrack to the closest node where we can descend in a different direction. If we are only interested in the eYT corresponding to the path we can built it as we descend down to the leaf, and store it instead of the path.

The problem is however, that in all this we descend and backtrack step by step through the graph, which will take on average O (N) steps and this brings the total complexity up to O ($N N_\lambda$).

To do better, we perform a pre-computation before the iteration itself adding the following information to each node μ: from μ we follow the *leftmost path* (lmp) to the leaf and, while

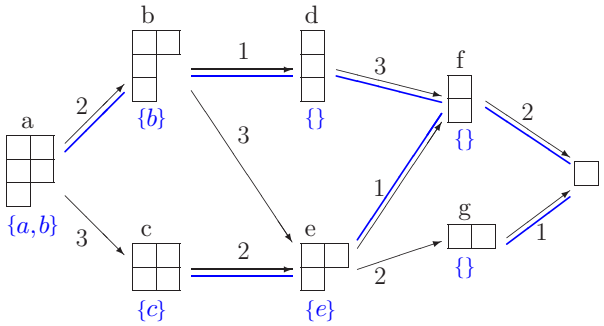

Figure A1. To enable efficient graph iteration, we need to augment the basic branching graph with additional information (shown in blue): to each node we attach the potential backtracking-positions, i.e. all nodes with more than one child, lying on the *leftmost path* descending from that node to the leaf.

we descend, push on a *stack* K_μ all the nodes with more than one child, because only these will be potential candidates to backtrack to (see figure A1). We also save T_μ the 'incomplete' eYT containing all numbers $N - \mathrm{d}(\mu), N - \mathrm{d}(\mu) - 1, \ldots, 1$ where $d(\mu)$ is the depth of node μ.

Precomputation of the triple K_μ, l_μ, T_μ for all nodes of a BG for shape λ built from elementary shapes σ takes

$$\mathrm{C}_{\mathrm{pre}} = \mathrm{O}\,(N)\,\mathrm{O}\left(D_\lambda^\sigma\right) = \mathrm{O}(N^{k+1}) \tag{A.1}$$

where we used that the function D_λ^σ counting the number of shapes $\mu \subset \lambda$ obtainable from λ by regular removal of elementary tableaux σ is bounded from above by N^k (see appendix B). We see it is only linearly more demanding (in N) than computing the basic BG.

Let us now sketch an iteration process which uses this precomputed information. The ingredients are first the array $c[.]$, already known from the naive iteration above and still needed to keep track of where we have descended to from the node $\mu[c](d)$ lying at a depth d. Furthermore we introduce a stack K (the backtrack-stack) which will at all times contain the depths of those nodes on our current path, where we could descend in a different direction, i.e. those which have more than one child *and* have not yet been exhausted ($c[d] <$ no. of children of node $\mu[c](d)$). Flow control requires only a single 'while'-loop which is repeated as long as K is nonempty.

The loop performs the following steps.

0. Assume we enter the loop with $c[.]$ initialized and a complete eYT E (from initialization or the previous pass).
1. First, retrieve the uppermost element (depth) from K (removing it in the process).
2. If that element is, say, j, increment $c[j]$ by one, reset $c[i] = 1$ for all $i > j$ and descend to $\nu := \mu[c](j+1))$ (the next, still unvisited child of node $\mu[c](j)$).
3. Update stack K: if $c[j] <$ no. of children of $\mu[c](j) \rightarrow$ push j back onto K.
4. In any case: push all nodes (depths) indicated in the backtrack-list of node ν onto K (the blue lists in figure A1).
5. Obtain next eYT: drop the indices $1, \ldots, j$ from E and join the remainder with the (incomplete) YT T_ν which was added to node ν during the pre-computation.

For standard YT, where the number of boxes $|\sigma|$ in the elementary tableaux is equal to one, joining two parts of an eYT can be done in a single step: we need only check wether

row(j) < row($j+1$). If so, we merge the leftmost column of the remainder with the rightmost column of T_v. If not, we simply concatenate. If we are considering tableaux with $|\sigma| > 1$, joining requires more steps, but can always be achieved in O ($|\sigma|$) time.

In all, the algorithm sketched above needs O (1) (or O ($|\sigma|$) i.g.) steps to generate one path/extended Young tableaux, and thus the full iteration requires O ($|\sigma| N_\lambda$) time and O ($N N_\lambda |\sigma|$) memory if we store the complete list of extended tableaux. If we only keep the momenta, O (N_λ) memory will suffice.

Appendix B. Branching graph size

The time required to generate the basic BG for a shape λ built up from N elementary tableaux σ as well as augmenting it in preparation for efficient iteration is determined mostly by its size, i.e. the number of its nodes. This in turn is just D_λ^σ, defined as

$$D_\lambda^\sigma = \text{no. of diagrams } \mu \text{ with } \mu \subset \lambda \text{ and } \mu \text{ obtained from } \lambda \text{ by regular removal of one or more shapes } \sigma. \tag{B.1}$$

Our goal is now to find a good estimate for D_λ^σ.

Defining $D_\lambda := D_\lambda^{\sigma=(1)}$, we can use it as an upper bound on D_λ^σ, as the additional σ-dependent constraints in (B.1) serve only to *decrease* the number μ that are compatible.

But D_λ is easily expressed as the multiple sum

$$D_\lambda = \sum_{j_1 \leqslant \lambda_1} \sum_{j_2 \leqslant \mathrm{Min}(\lambda_2, j_1)} \cdots \sum_{j_k \leqslant \mathrm{Min}(\lambda_k, j_{k-1})} 1 \tag{B.2}$$

where k is the number of rows in λ. This can be estimated from above by forgetting about $\mathrm{Min}(\lambda_i, j_{i-1})$ and bounding j_i just by j_{i-1} instead

$$D_\lambda \leqslant \sum_{j_k \leqslant j_{k-1} \leqslant \ldots \leqslant j_1 \leqslant \lambda_1} 1 = \binom{\lambda_1 + k}{\lambda_1}. \tag{B.3}$$

Two instances are of particular interest (set $\sigma = 1$).

- $\lambda_1 = N, k = 1$ in which case the estimate gives almost the exact result ($D_\lambda = N$ compared to $\binom{N+1}{N} = N + 1$).
- $\lambda_1 = N/k =: m, k > 1, k|N$ (rectangle shape) where the above estimate gives the exact result, as we will show in the following.

Building the BG for a rectangular shape removing in turn $1, 2, \ldots, s, \ldots, N-1$ boxes is equivalent to building (box by box) shapes with no more than k rows and m columns. Without restrictions, the number of shapes with N boxes is simply $p(N)$, the number of integer partitions of N. With the restrictions, we must instead use $p_{<k;m}(n)$, the number of integer partitions using at most m summands of size $\leqslant k$. Thus we obtain the exact BG size for a rectangular shape $\lambda = (m, \ldots, m)$, if we sum this over all steps $s = 1, \ldots, N$:

$$D_{(m,\ldots,m)} = \sum_{s=1}^{N} p_{\leqslant k;m}(s). \tag{B.4}$$

However, a little thought reveals that this and the sum (B.3) are, in fact, the same, proving that in this case the bound (B.2) is tight and the number of nodes is exactly given by $(m+k)!/k!m!$.

What is the asymptotic complexity in terms of N? If $k \ll \lambda_1 \approx N$ (or vice versa), clearly $(\lambda_1 + k)!/k!\lambda_1! \leqslant (N + k)!/k!N! = \mathrm{O}(N^k)$ and therefore polynomial in N. However, if

Table C1. The four possible parity combinations of $\sum_c \langle i\rangle_c$ and N. All lead to an integer value for the total momentum (C.2).

$\sum_c \langle i\rangle_c$	N	$\Rightarrow$	c_T	Total
Half-integer	even	$\Rightarrow$	odd	Integer
”	odd	$\Rightarrow$	even	”
Integer	even	$\Rightarrow$	even	”
”	odd	$\Rightarrow$	odd	”

$k \approx \sqrt{N}$ (e.g. shapes of square or triangular form like $(k, k-1, \ldots, 1)$), then our upper bound is

$$D_\lambda = \mathrm{O}\left(\frac{\Gamma(2\sqrt{N})}{\Gamma^2(\sqrt{(N)})}\right) = \mathrm{O}(\exp[\sqrt{N}])$$

and since we have shown that it is tight in the case $\lambda_1 = N/k$, we see that there are indeed shapes for which computing the BG is of nearly exponential complexity. But exactly these shapes also have the highest number of YT, growing like O (exp[N]), i.e. fully exponential. Therefore computing the BG is always worthwhile, as it is in all cases much less costly than generating the YT from it.

For many other combinations of N and k (B.2) overestimates the size of the BG considerably. Take $\lambda = (N-k+1, 1, \ldots, 1)$. Assuming $N-k+1 > k$ the true value is $D_\lambda = (\lambda_1 - 1)^2 + (\lambda_1 - 1)(\lambda_1 - k + 1) \approx 2\lambda_1^2 - k\lambda_1 = \mathrm{O}\left(\lambda_1^2\right)$ independent of k (as long as it remains smaller than λ_1) while (B.2) yields $\mathrm{O}\left(\lambda_1^k\right)$.

Appendix C. Momentum offset b_0

Given a tableaux T built from N elementary tableaux σ we may interpret it as pertaining to the product space $V_\sigma^{\otimes N}$ of any SU(n) where n is at least as large as the number of rows in T. We want to show here, that the momentum assigned to T via the sum (7) is independent of this interpretation, i.e. independent of n:

$$\sum_{c\in E(T)} b_c = \sum_c (n-k_c)\left(\langle i\rangle_c - \frac{1}{2}\right) = n\left(\sum_c \langle i\rangle_c - \frac{1}{2}c_T\right) - \frac{1}{2}|\sigma|N^2 \quad \text{(C.1)}$$

where we defined c_T as the number of columns of $E(T)$ and used the relations $c_T = \sum_c 1$, $\sum_c k_c = |\sigma|N$ and $\sum_c k_c \langle i\rangle_c = |\sigma|N(N+1)/2$. As a reminder, k_c is the number of boxes in column c of the extended tableaux $E(T)$ and $|\sigma|N$ is just the total number of boxes in $E(T)$ (and therefore also in T).

We see, that if we add the offset momentum number $b_0 = -(n-1)N^2/2$ we arrive at

$$b_0 + \sum_{c\in E(T)} b_c = n\left(\sum_c \langle i\rangle_c - \frac{1}{2}(c_T + |\sigma|N^2)\right) \quad \text{(C.2)}$$

and thus n cancels when computing the momentum $p_T = 2\pi/Nn(\sum_c b_c + b_0)$.

What still needs to be checked is whether the quantity in parenthesis in (C.2) is always an integer. To see that this is indeed the case, we need to analyse the relationship between N, c_T and $\sum_c \langle i\rangle_c$. Since in all columns of $E(T)$, the boxes are in sequence, $\sum_c \langle i\rangle_c$ is always either integer or half-integer. In fact we can express each summand as $\langle i\rangle_c = j_c + (k_c - 1)/2$, where j_c is number in the uppermost box. Therefore, $\langle i\rangle_c$ is half-integer, if and only if there is an even number of boxes in column c. Now assume $\sum_c \langle i\rangle_c$ is half-integer. This means, we

must have an *odd* number of columns with an even number of boxes. If now N is even, there is an even number of boxes left to be distributed over rows with an odd number of boxes in them. This means that this number of (odd-box-number) columns must be even. Thus in this case $c_T/2$ is half-integer while $N^2/2$ is integer and in total sum (C.2) is of the form $n\times$integer. It is not hard to see that in the other three cases this holds as well (see table C1).

References

[1] Sutherlandn B 1975 Model for a multicomponent quantum system *Phys. Rev.* B **12** 3795–805

[2] Affleck I 1988 Critical behaviour of SU(n) quantum chains and topological nonlinear σ-models *Nucl. Phys.* B **305** 582–96

[3] Affleck I, Arovas D P, Marston J B and Rabson D A 1991 Su(2n) quantum antiferromagnets with exact c-breaking ground states *Nucl. Phys.* B **366** 467–506

[4] Kawakami N 1992 Asymptotic Bethe-ansatz solution of multicomponent quantum systems with $1/r^2$ long-range interaction *Phys. Rev.* B **46** 1005–14

[5] Kawakami N 1992 Su(N) generalization of the Gutzwiller–Jastrow wave function and its critical properties in one dimension *Phys. Rev.* B **46** 3191–4

[6] Ha Z N C and Haldane F D M 1993 Squeezed strings and Yangian symmetry of the Heisenberg chain with long-range interaction *Phys. Rev.* B **47** 12459–69

[7] Shen S-Q 2001 Generalized valence bond state and solvable models for spin-$\frac{1}{2}$ systems with orbital degeneracy *Phys. Rev.* B **64** 132411

[8] Honerkamp C and Hofstetter W 2004 Ultracold fermions and the SU(n) Hubbard model *Phys. Rev. Lett.* **92** 170403

[9] Damerau J and Klümper A 2006 Nonlinear integral equations for the thermodynamics of the *sl*-symmetric Uimin–Sutherland model *J. Stat. Mech.* P12014

[10] Greiter M and Rachel S 2007 Valence bond solids for SU(n) spin chains: exact models, spinon confinement, and the Haldane gap *Phys. Rev.* B **75** 184441

[11] Pankov S, Moessner R and Sondhi S L 2007 Resonating singlet valence plaquettes *Phys. Rev.* B **76** 104436

[12] Arovas D P 2008 Simplex solid states of SU(N) quantum antiferromagnets *Phys. Rev.* B **77** 104404

[13] Katsura H, Hirano T and Korepin V E 2008 Entanglement in an SU(n) valence-bond-solid state *J. Phys. A: Math. Theor.* **41** 135304

[14] Xu C and Congjun W 2008 Resonating plaquette phases in su(4) Heisenberg antiferromagnet *Phys. Rev.* B **77** 134449

[15] Manmana S R, Hazzard K R A, Chen G, Feiguin A E and Rey A M 2011 SU(n) magnetism in chains of ultracold alkaline-earth-metal atoms: Mott transitions and quantum correlations *Phys. Rev.* A **84** 043601

[16] Hermele M and Gurarie V 2011 Topological liquids and valence cluster states in two-dimensional su(n) magnets *Phys. Rev.* B **84** 174441

[17] Corboz P, Läuchli A M, Penc K, Troyer M and Mila F 2011 Simultaneous dimerization and SU(4) symmetry breaking of 4-color fermions on the square lattice *Phys. Rev. Lett.* **107** 215301

[18] Bauer B, Corboz P, Läuchli A M, Messio L, Penc K, Troyer M and Mila F 2012 Three-sublattice order in the su(3) Heisenberg model on the square and triangular lattice *Phys. Rev.* B **85** 125116

[19] Garcia-Ripoll J J, Martin-Delgado M A and Cirac J I 2004 Implementation of spin Hamiltonians in optical lattices *Phys. Rev. Lett.* **93** 250405

[20] Hermele M, Gurarie V and Rey A M 2009 Mott insulators of ultracold fermionic alkaline earth atoms: underconstrained magnetism and chiral spin liquid *Phys. Rev. Lett.* **103** 135301

[21] Azaria P, Capponi S and Lecheminant P 2009 Three-component fermi gas in a one-dimensional optical lattice *Phys. Rev.* A **80** 041604

[22] Xu C 2010 Liquids in multiorbital SU(n) magnets made up of ultracold alkaline-earth atoms *Phys. Rev.* B **81** 144431

[23] Gorshkov A V, Hermele M, Gurarie V, Xu C, Julienne P S, Ye J, Zoller P, Demler E, Lukin M D and Rey A M 2010 Two-orbital SU(N) magnetism with ultracold alkaline-earth atoms *Nature Phys.* **6** 289–95

[24] Young A 1900 On quantitative substitutional analysis *Proc. Lond. Math. Soc.* **33** 97–146

[25] Hamermesh M 1989 *Group Theory and Its Application to Physical Problems (Dover Books on Physics)* 1st edn (New York: Dover)

[26] Fulton W and Harris J 1991 *Representation Theory: a First Course (Graduate Texts in Mathematics* vol 129*)* (New York: Springer)

[27] Wu W and Zhang Q 1992 An efficient algorithm for evaluating the standard Yamanouchi–Young orthogonal representation with two column Young tableaux for symmetric groups *J. Phys. A: Math. Gen.* **25** 3737–47
[28] Yu Y, Palting P and Chiu Y-N 1996 Young operator methods for fermion systems *Theor. Chim. Acta* **94** 125–41
[29] Greiter M and Schuricht D 2007 Many-spinon states and the secret significance of Young tableaux *Phys. Rev. Lett.* **98** 237202
[30] Greiter M 2011 *Mapping of Parent Hamiltonians* (Berlin: Springer)
[31] McAven L F and Schlesinger M 2001 The identification of Young tableaux with angular momentum states *J. Phys. A: Math. Gen.* **34** 8333–43
[32] Haldane F D M 1988 Exact Jastrow–Gutzwiller resonant-valence-bond ground state of the spin-$1/2$ antiferromagnetic Heisenberg chain with $1/r^2$ exchange *Phys. Rev. Lett.* **60** 635–8
[33] Shastry B S 1988 Exact solution of an $s = 1/2$ Heisenberg antiferromagnetic chain with long-ranged interactions *Phys. Rev. Lett.* **60** 639–42
[34] Ha Z N C and Haldane F D M 1992 Models with inverse-square exchange *Phys. Rev.* B **46** 9359–68
[35] Schuricht D and Greiter M 2006 Coloron excitations of the SU(3) Haldane–Shastry model *Phys. Rev.* B **73** 235105
[36] Talstra J C and Haldane F D M 1995 Integrals of motion of the Haldane–Shastry model *J. Phys. A: Math. Gen.* **28** 2369–77