Algorithms for SU(n) boson realizations and $\mathcal{D}$-functions

Ish Dhand,[1,a] Barry C. Sanders,[1,2,3,4] and Hubert de Guise[5]

[1]*Institute for Quantum Science and Technology, University of Calgary, Calgary, Alberta T2N 1N4, Canada*
[2]*Hefei National Laboratory for Physical Sciences at Microscale and Department of Modern Physics, University of Science and Technology of China, Hefei, Anhui 230026, China*
[3]*CAS Center for Excellence and Synergetic Innovation Center in Quantum Information and Quantum Physics, University of Science and Technology of China, Shanghai 201315, China*
[4]*Program in Quantum Information Science, Canadian Institute for Advanced Research, Toronto, Ontario M5G 1Z8, Canada*
[5]*Department of Physics, Lakehead University, Thunder Bay, Ontario P7B 5E1, Canada*

Boson realizations map operators and states of groups to transformations and states of bosonic systems. We devise a graph-theoretic algorithm to construct the boson realizations of the canonical SU(n) basis states, which reduce the canonical subgroup chain, for arbitrary n. The boson realizations are employed to construct $\mathcal{D}$-functions, which are the matrix elements of arbitrary irreducible representations, of SU(n) in the canonical basis. We demonstrate that our $\mathcal{D}$-function algorithm offers significant advantage over the two competing procedures, namely, factorization and exponentiation. © *2015 AIP Publishing LLC.* [http://dx.doi.org/10.1063/1.4935433]

I. INTRODUCTION

$\mathcal{D}$-functions of a group element are the entries of irreducible matrix representations (irreps) of the element. $\mathcal{D}$-functions of the special unitary group SU(2) are important in nuclear, atomic, molecular, and optical physics.[1–7] SU(1, 1) is the prototypical non-compact semi-simple Lie group, and its $\mathcal{D}$-functions appear in connection with Bogolyubov transformations, squeezing and parametric downconversion.[8,9] Methods for construction of intelligent states and the analysis of cylindrical Laguerre-Gauss beams employ $\mathcal{D}$-functions of SU(1, 1).[10,11] $\mathcal{D}$-functions of other Lie groups enable exact solutions to problems in quantum optics.[12,13]

One recent application of $\mathcal{D}$-functions of SU(n) for arbitrary n is to the BosonSampling problem, which deals with SU(n) transformations acting on indistinguishable single-photon pulse inputs.[14,15] Within the framework of BosonSampling and of multi-photon interferometry in general, $\mathcal{D}$-functions provide a deeper understanding of the permutation symmetries between the interfering photons. For instance, SU(3) $\mathcal{D}$-functions enable a symmetry-based interpretation of the action of a three-channel linear interferometer on partially distinguishable single-photon inputs.[16,17] Exploiting the permutation symmetries present in multi-photon systems reduces the cost of computing interferometer outputs in comparison to brute-force techniques.[18]

Two existing procedures for computing SU(n) $\mathcal{D}$-function are based on factorization and on exponentiation. Both procedures have drawbacks, which we describe as follows. Factorization-based methods, which compute SU(n) $\mathcal{D}$-functions in terms of $\mathcal{D}$-functions of subtransformations, are well developed for groups of low rank.[19–23] However, generalizing these algorithms to higher n requires SU($n - 1$) coupling and recoupling coefficients, which have limited availability for $n > 3$, i.e., restricted to certain subgroups of SU(3).[24–26] Hence, methods for $\mathcal{D}$-functions of higher groups are underdeveloped despite the application of their corresponding algebras to diverse problems.[27–30]

---

a)Electronic mail: ishdhand@gmail.com. URL: http://ishdhand.me/.

Another approach for computing SU(n) $\mathcal{D}$-functions involves exponentiating and composing the matrix representations of the algebra.[31,32] This approach has three hurdles. For one, this method requires knowledge of all the matrix elements of each generator to be exponentiated. Certain applications require closed-form expressions of $\mathcal{D}$-functions in terms of elements of the fundamental representation; exponentiation-based methods are infeasible for these applications because of the difficulty of exponentiating matrices analytically, especially for $n > 5$. Furthermore, if only a limited number of $\mathcal{D}$-functions are required, exponentiation is wasteful because it computes the entire set of $\mathcal{D}$-functions.

We overcome the shortcomings of these algorithms by utilizing boson realizations, which map the algebra and its carrier space to bosonic operators and spaces, respectively. Boson realizations arise naturally when considering the groups Sp($2n,\mathbb{R}$), SU(n), and some of their subgroups. For instance, SU(1, 1), SU(2), and SU(3) boson realizations are used to study degeneracies, symmetries, and dynamics in quantum systems.[33–41] A wide class of problems in theoretical physics rely on boson realizations of the symplectic group.[42–46]

Here, we aim to devise an algorithm to construct the $\mathcal{D}$-functions of arbitrary representations of SU(n) for arbitrary n. We approach the problem of limited availability of SU(n) $\mathcal{D}$-functions[47,48] by presenting (i) a mapping of the weights of an irrep to a graph, (ii) a graph-theoretic algorithm to compute boson realizations of the canonical basis states of SU(n) for arbitrary n (Algorithm 2 in Subsection IV B), and (iii) an algorithm that employs the constructed boson realizations to compute expressions for $\mathcal{D}$-functions as polynomials in the matrix elements of the defining representation (Algorithm 3 in Subsection IV C).

The rest of the paper is structured as follows. Section II includes definitions of the SU(n) operators and basis states. In Section III, we define SU(n) boson realizations and illustrate the calculations of SU(2) $\mathcal{D}$-functions using SU(2) boson realizations. Section IV details our algorithms for boson realizations of SU(n) basis states and for the SU(n) $\mathcal{D}$-functions. We discuss potential generalizations of our algorithms in Section V.

II. BACKGROUND: THE SPECIAL UNITARY GROUP AND ITS ALGEBRA

In this section, we recall the relevant properties of special-unitary group SU(n) and its algebra $\mathfrak{su}(n)$. We explain how the $\mathfrak{su}(n) \supset \mathfrak{su}(n-1) \supset \cdots \supset \mathfrak{su}(2)$ subalgebra chain is used to label the basis states of the unitary irreps of SU(n). We present the background for $n = 2$ in Subsection II A before dealing with SU(n) for arbitrary n in Subsection II B.

A. SU(2) operators and basis states

Consider the special unitary group

$$\mathrm{SU}(2) = \{V : V \in GL(2,\mathbb{C}),\, V^\dagger V = \mathbb{1},\, \det V = 1\} \tag{1}$$

of 2×2 special unitary matrices. Each element of SU(2) can be parametrized by three angles $\Omega = (\alpha, \beta, \gamma)$. The defining 2×2 representation of an element $V(\Omega)$ of SU(2) is given by

$$\mathrm{V}(\Omega) = \begin{pmatrix} \mathrm{e}^{-\frac{1}{2}\mathrm{i}(\alpha+\gamma)} \cos\frac{\beta}{2} & -\mathrm{e}^{-\frac{1}{2}\mathrm{i}(\alpha-\gamma)} \sin\frac{\beta}{2} \\ \mathrm{e}^{\frac{1}{2}\mathrm{i}(\alpha-\gamma)} \sin\frac{\beta}{2} & \mathrm{e}^{\frac{1}{2}\mathrm{i}(\alpha+\gamma)} \cos\frac{\beta}{2} \end{pmatrix}. \tag{2}$$

The Lie algebra corresponding to group SU(2) is denoted by $\mathfrak{su}(2)$ and is spanned by the operators J_x, J_y, J_z, which satisfy the angular momentum commutation relations

$$[J_x, J_y] = iJ_z, \quad [J_y, J_z] = iJ_x, \quad [J_z, J_x] = iJ_y. \tag{3}$$

We transform basis (3) of $\mathfrak{su}(2)$ to the complex combinations

$$C_{1,2} = J_x + iJ_y, \qquad C_{2,1} = J_x - iJ_y, \qquad H_1 = 2J_z, \tag{4}$$

which satisfy the commutation relations

$$[H_1, C_{1,2}] = 2C_{1,2}, \quad [H_1, C_{2,1}] = -2C_{2,1}, \quad [C_{1,2}, C_{2,1}] = H_1. \tag{5}$$

These commutation relations (5) facilitate the construction of a $(2J + 1)$-dimensional irrep with carrier space spanned by basis states $\{|J, M\rangle : -J \le M \le J\}$.[49] The integer $2M$ is the weight of the eigenstate $|J, M\rangle$ for

$$H_1|J, M\rangle = 2M|J, M\rangle. \tag{6}$$

The operators $C_{1,2}$ and $C_{2,1}$ act on eigenstates of H_1 by raising or lowering the weight $2M$ of the states

$$C_{1,2}|J, M\rangle = \sqrt{J(J + 1) - M(M + 1)}|J, M + 1\rangle, \tag{7}$$

$$C_{2,1}|J, M\rangle = \sqrt{J(J + 1) - M(M - 1)}|J, M - 1\rangle, \tag{8}$$

where $2J$ is the highest eigenvalue of H_1.

Each basis state of a finite-dimensional irrep of SU(2) is labelled by integral weight $2M \in \{-2J, -2J + 2, \ldots, 2J - 2, 2J\}$. The unique basis state $|J, J\rangle$ is called the highest-weight state (hws) and is annihilated by the action of the raising operator $C_{1,2}$. The representation is labelled by the largest eigenvalue $2J$ of H_1. Basis states of an SU(2) irrep are visualized as collections of points on a line with the location of each point related to the weight of the state. Figure 1 gives a geometrical representation of the action of $\mathfrak{su}(2)$ operators and illustrative examples of SU(2) irreps.

B. Basis states and $\mathcal{D}$-functions of SU(n) for arbitrary n

Next we consider the case of arbitrary n. The unitary group U(n) is the Lie group of $n \times n$ unitary matrices

$$\mathrm{U}(n) := \{V : V \in \mathrm{GL}(n, \mathbb{C}), V^\dagger V = \mathbb{1}\}. \tag{9}$$

The corresponding Lie algebra is denoted by $\mathfrak{u}(n)$. The complex extension of $\mathfrak{u}(n)$ is spanned by n^2 operators $\{C_{i,j} : i, j \in 1, 2, \ldots, n\}$ satisfying the canonical commutation relations

$$[C_{i,j}, C_{k,l}] = \delta_{j,k}C_{i,l} - \delta_{i,l}C_{k,j}. \tag{10}$$

The group SU(n) is the subgroup of those U(n) transformations that satisfy the additional property $\det V = 1$, i.e.,

$$\mathrm{SU}(n) := \{V : V \in \mathrm{U}(n), \det V = 1\}. \tag{11}$$

The U(n) $\mathcal{D}$-functions differ from the SU(n) $\mathcal{D}$-functions by at most a phase, and we concentrate here on the SU(n) case.

The operator $N = C_{1,1} + C_{2,2} + \cdots + C_{n,n}$ is in the centre[50] of $\mathfrak{u}(n)$. The Lie algebra $\mathfrak{su}(n)$ is obtained from $\mathfrak{u}(n)$ by eliminating the operator N. The $n - 1$ operators

$$H_i = C_{i,i} - C_{i+1,i+1} \quad \forall i \in \{1, 2, \ldots, n - 1\} \tag{12}$$

commute with each other and span the Cartan subalgebra of $\mathfrak{su}(n)$. Hence, we have the following definition of the $\mathfrak{su}(n)$ algebra.
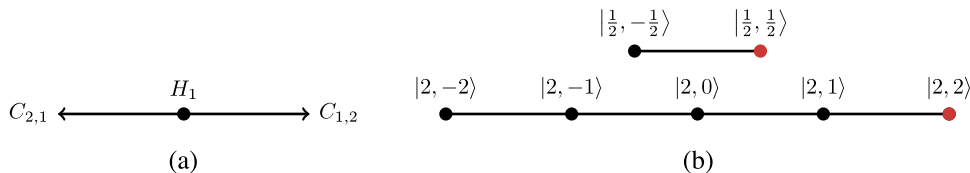

FIG. 1. (a) Generators of the $\mathfrak{su}(2)$ algebra. The action of the raising and lowering operators $C_{1,2}, C_{2,1}$ on the basis states is represented by the directed lines. The basis states are invariant under the action of the Cartan operator H_1, which is represented by the dot at the centre. (b) SU(2) irreps labelled by highest weights $2M = 1$ and $2M = 4$, respectively. The dots represent the basis states while the lines connecting the dots represent the transformation from one basis state to another by the action of the $\mathfrak{su}(2)$ raising and lowering operators. The red dot represents the hws, which is annihilated by the action of the raising operator $C_{1,2}$.

Definition 1 ($\mathfrak{su}(n)$ algebra[49]). The algebra $\mathfrak{su}(n)$ is the span of the operators $\{C_{i,j}: i, j \in \{1, 2, \ldots, n\}, i \neq j\}$ and $\{H_i: H_i = C_{i,i} - C_{i+1,i+1}, i \in \{1, 2, \ldots, n-1\}\}$ where the operators $\{C_{i,j}\}$ obey the commutation relations

$$[C_{i,j}, C_{k,l}] = \delta_{j,k} C_{i,l} - \delta_{i,l} C_{k,j}. \tag{13}$$

The linearly independent (LI) $\mathfrak{su}(n)$ basis states span the carrier space of $\mathfrak{su}(n)$ representations. Each basis state is associated with a weight, which is the set of integral eigenvalues of the Cartan operators.

Definition 2 (Weight of $\mathfrak{su}(n)$ basis states[49]). The weight of a basis state is the set $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{n-1})$ of $n-1$ integral eigenvalues of the Cartan operators $\{H_1, H_2, \ldots, H_{n-1}\}$. $\mathfrak{su}(n)$ basis states have well defined weights.

Of the $n^2 - 1$ elements, $n - 1$ Cartan operators generate the maximal Abelian subalgebra of $\mathfrak{su}(n)$. The remaining operators satisfy the commutation relation

$$[H_i, C_{j,k}] = \begin{cases} \beta_{i,jk} C_{j,k}, & \forall j < k, \\ -\beta_{i,jk} C_{j,k}, & \forall j > k, \end{cases} \tag{14}$$

for Cartan operators H_i of Definition 1 and for positive integral roots $\beta_{i,jk}$. The operators $\{C_{j,k}: j < k\}$ define a set of raising operators. The remaining off-diagonal operators $\{C_{j,k}: j > k\}$ are the $\mathfrak{su}(n)$ lowering operators. Each irrep contains a unique state that has non-negative integral weights $K = (\kappa_1, \ldots, \kappa_{n-1})$ and is annihilated by all raising operators. This state is the hws of the irrep.

Definition 3 (Highest-weight state). The hws of an $\mathrm{SU}(n)$ irrep is the unique state that is annihilated according to

$$C_{i,j} |\psi_{hws}^K\rangle = 0 \quad \forall i < j, \, i, j \in \{1, 2, \ldots, n\} \tag{15}$$

by the action of all the raising operators.

The weight of the hws also labels the irrep, i.e., two irreps with the same highest weight are equivalent and two equivalent representations have the same highest weight. Hence, we label an irrep by $K = (\kappa_1, \kappa_2, \ldots, \kappa_{n-1})$ if the hws of the irrep has weight $\Lambda = K$.

Whereas in SU(2) the weight $2M$ and the representation label J are enough to uniquely identify a state in the representation, this is not so for SU(n) representations. In general, more than one SU(n) basis state of an irrep could share the same weight. For example, certain states of the $K = (2,2)$ irrep of SU(3) irrep have the same weight (Fig. 2). The number of basis states that share the same SU(n) weight $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{n-1})$ is the multiplicity $M(\Lambda)$ of the weight.[51] Hence, uniquely labelling the SU(n) basis states requires a scheme to lift the possible degeneracy of weights.

One approach to labelling the SU(n) basis states involves specifying the transformation properties under the action of the subalgebras of $\mathfrak{su}(n)$. We restrict our attention to the canonical subalgebra chain

$$\mathfrak{su}_{1,2,\ldots,n}(n) \supset \mathfrak{su}_{1,2,\ldots,n-1}(n-1) \supset \cdots \supset \mathfrak{su}_{1,2}(2), \tag{16}$$

where $\mathfrak{su}_{1,2,\ldots,m}(m)$ is the subalgebra generated by the operators $\{C_{i,j}: i, j \in \{1, 2, \ldots, m\}, i \neq j\}$ and $\{H_k: k \in \{1, 2, \ldots, m-1\}\}$. Details about the choice of subalgebra chain are presented in Appendix A. Henceforth, we drop the subscript and denote $\mathfrak{su}_{1,2,\ldots,m}(m)$ by $\mathfrak{su}(m)$.

The canonical basis comprises the eigenstates of the $\mathfrak{su}(m)$ generators for all $m \leq n$ according to the following definition.

Definition 4 (Canonical basis states). The canonical basis states of $\mathrm{SU}(n)$ irrep $K^{(n)}$ are those states

$$\left| \psi_{\Lambda^{(n)}, \ldots, \Lambda^{(3)}, \Lambda^{(2)}}^{K^{(n)}, \ldots, K^{(3)}, K^{(2)}} \right\rangle \tag{17}$$

that have well defined values of

1. *irrep labels $K^{(m)}$ for $\mathfrak{su}(m)$ algebras for all $\{m: 2 \leq m \leq n\}$ and*
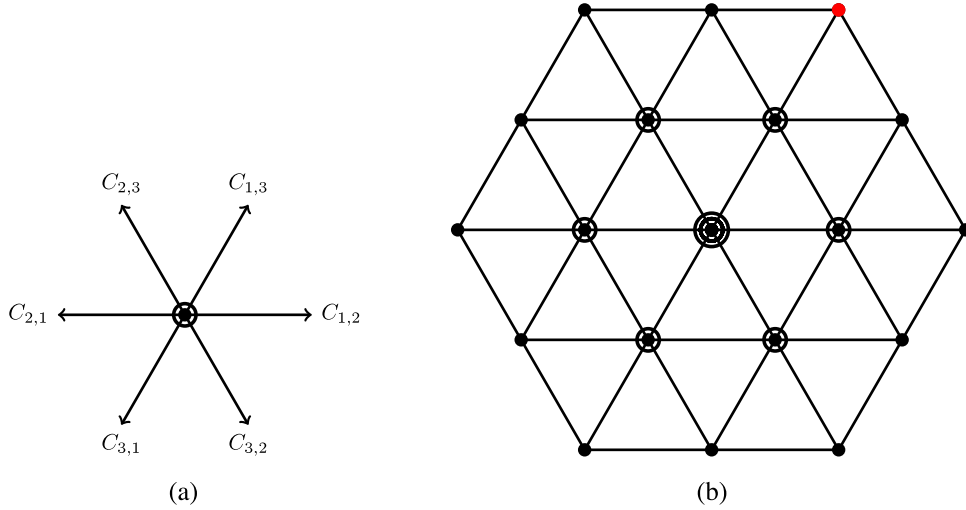
FIG. 2. (a) Generators of the $\mathfrak{su}(3)$ algebra. The action of the raising operators $\{C_{1,2}, C_{1,3}, C_{2,3}\}$ and lowering operators $\{C_{2,1}, C_{1,3}, C_{2,3}\}$ on the canonical basis states and their linear combinations is represented by the directed lines. (b) The SU(3) irrep labelled by its highest weight $(\kappa_1, \kappa_2) = (2, 2)$. The dots and circles represent the canonical basis states. The dimension of the space of states at a given vertex is the sum of the number of dots and the number of circles at the vertex, for instance, weights associated with dimension two are represented by one dot and one circle. The lines connecting the dots represent the transformation from states of one weight to those of another by the action of SU(3) raising and lowering operators. The red dot represents the highest weight of the irrep. A unique hws occupying this weight is annihilated by the action of each of the raising operator.

2. $\mathfrak{su}(m)$ weights $\Lambda^{(m)}$, i.e., eigenvalues of the Cartan operators of $\mathfrak{su}(m)$ algebras for all $\{m : 2 \leq m \leq n\}$.

Consider the example of the $(\kappa_1, \kappa_2) = (1, 1)$ irrep of SU(3). There are two basis states with the weight $(\lambda_1, \lambda_2) = (0, 0)$. We can identify these two states by specifying

1. the $\mathfrak{su}(3)$ irrep label $K^{(3)} = (\kappa_1, \kappa_2) = (1, 1)$ and the $\mathfrak{su}(2)$ irrep label $K^{(2)} = (\kappa_1) = (0)$ or $K^{(2)} = (\kappa_1) = (1)$;
2. the $\mathfrak{su}(3)$ weights $\Lambda^{(3)} = (\lambda_1, \lambda_2) = (0, 0)$ and $\mathfrak{su}(2)$ weight $\Lambda^{(2)} = (\lambda_1) = (0)$.

The connection between our labelling of canonical basis states of Definition 4 and the Gelfand-Tsetlin (GT) patterns[52] is detailed in Appendix B. The canonical basis state $\left| \psi_{\Lambda^{(n)},\dots,\Lambda^{(3)}, \Lambda^{(2)}}^{K^{(n)},\dots,K^{(3)},K^{(2)}} \right\rangle$ for which $K^{(m)} = \Lambda^{(m)}$ for all $m \in \{2, \dots, n\}$ is the highest weight of the irrep $K^{(n)}$.

The relative phases between the canonical basis states are fixed by comparing with the phase of the hws.[52] Matrix elements of the simple raising operators $C_{\ell,\ell+1}$, $\ell \in \{1, \dots, n-1\}$ are set as positive.[53] Thus, we impose the following additional constraint on the canonical basis states:

$$\left\langle \psi_{\text{hws}} \left| c_{1,2}^{p_{1,2}} c_{2,3}^{p_{2,3}} \cdots c_{n-1,n}^{p_{n-1,n}} \right| \psi_{\Lambda^{(n)},\dots,\Lambda^{(3)}, \Lambda^{(2)}}^{K^{(n)},\dots,K^{(3)},K^{(2)}} \right\rangle \geq 0, \tag{18}$$

for all canonical basis states, for positive integers $p_{\ell,\ell+1}$.

$\mathcal{D}$-functions are the matrix elements of SU(n) irreps. The rows and columns of SU(n) matrix representations are labelled by SU(n) basis states. The expressions for SU(n) $\mathcal{D}$-functions generalize those of SU(2) $\mathcal{D}$-functions (25) with M, M' replaced by suitable labels for weights and J replaced by suitable subalgebra labels.

Definition 5 ($\mathcal{D}$-functions). $\mathcal{D}$-functions of an SU(n) *transformation* $V(\Omega)$ *are*

$$\mathcal{D}_{\Lambda^{(n)},\dots,\Lambda^{(3)}, \Lambda^{(2)};\Lambda'^{(n)},\dots,\Lambda'^{(3)}, \Lambda'^{(2)}}^{K^{(n)},\dots,K^{(3)},K^{(2)};K'^{(n)},\dots,K'^{(3)},K'^{(2)}}(\Omega) := \left\langle \psi_{\Lambda^{(n)},\dots,\Lambda^{(3)}, \Lambda^{(2)}}^{K^{(n)},\dots,K^{(3)},K^{(2)}} \left| V(\Omega) \right| \psi_{\Lambda'^{(n)},\dots,\Lambda'^{(3)}, \Lambda'^{(2)}}^{K'^{(n)},\dots,K'^{(3)},K'^{(2)}} \right\rangle, \tag{19}$$

where $\Omega = \{\omega_1, \omega_2, \dots, \omega_{n^2-1}\}$ *is the set of* $n^2 - 1$ *independent angles that parameterize an* SU(n) *transformation.*[54]

Note that SU(n) $\mathcal{D}$-functions (19) are non-zero only if the left and the right states belong to the same SU(n) irrep, i.e.,

$$K^{(n)} \neq K'^{(n)} \implies \mathcal{D}^{K^{(n)},...,K^{(3)},K^{(2)};K'^{(n)},...,K'^{(3)},K'^{(2)}}_{\Lambda^{(n)},...,\Lambda^{(3)},\,\Lambda^{(2)};\Lambda'^{(n)},...,\Lambda'^{(3)},\,\Lambda'^{(2)}}(\Omega) = 0. \tag{20}$$

$\mathcal{D}$-functions of an irrep K refer to those $\mathcal{D}$-functions for which $K^{(n)} = K'^{(n)} = K$.

We approach the task of constructing SU(n) $\mathcal{D}$-functions by using boson realizations of SU(n) states. In Sec. III, we define boson realizations and illustrate the construction of SU(2) $\mathcal{D}$-functions using SU(2) boson realizations.

III. BACKGROUND: BOSON REALIZATIONS OF $SU(n)$

In this section, we describe boson realizations, which map $\mathfrak{su}(n)$ operators and carrier-space states to operators and states of a system of $n - 1$ species of bosons on n sites, respectively. We first present the mapping for $n = 2$ and illustrate SU(2) $\mathcal{D}$-functions calculation using the SU(2) boson realization in Subsection III A. Boson realizations of SU(n) for arbitrary n are defined in Subsection III B.

A. SU(2) boson realizations

Commutation relations (5) of $\{C_{1,2}, C_{2,1}, H_1\}$ are reproduced by number-preserving bilinear products of creation and annihilation operators that act on a two-site bosonic system. Specifically, the $\mathfrak{su}(2)$ operators have the boson realization

$$C_{1,2} \mapsto c_{1,2} := a_1^\dagger a_2, \quad C_{2,1} \mapsto c_{2,1} := a_2^\dagger a_1, \quad H_1 \mapsto h_1 := a_1^\dagger a_1 - a_2^\dagger a_2, \tag{21}$$

where the bosonic creation and annihilation operators obey the commutation relations

$$\left[a_i, a_j^\dagger\right] = \delta_{ij}\mathbb{1}, \quad [a_i, a_j] = \left[a_i^\dagger, a_j^\dagger\right] = 0. \tag{22}$$

Here and henceforth, we use lower-case symbols for boson realizations of the respective upper-case symbols. Explicitly,

$$[h_1, c_{1,2}] = 2c_{1,2}, \qquad [h_1, c_{2,1}] = -2c_{2,1}, \qquad [c_{1,2}, c_{2,1}] = h_1. \tag{23}$$

The operators $\{c_{1,2}, c_{2,1}, h_1\}$ also span the complex extension of the $\mathfrak{su}(2)$ Lie algebra.

Boson realizations map the states in the carrier space of SU(2) to the states of a two-site bosonic system. Specifically, each basis state of the $(2J + 1)$-dimensional SU(2) irrep maps

$$|J, M\rangle \mapsto \frac{(a_1^\dagger)^{J+M}(a_2^\dagger)^{J-M}}{\sqrt{(J+M)!(J-M)!}}|0\rangle \tag{24}$$

to the state of a two-site system with $J + M$ and $J - M$ bosons in the two sites, respectively.

The $(2J + 1)$-dimensional irreps of SU(2) map to number-preserving transformations on a two-site system of $2J$ bosons in the basis of Eq. (24). The elements of these $(2J + 1) \times (2J + 1)$ matrices are the SU(2) $\mathcal{D}$-functions

$$D^J_{M'M}(\Omega) := \langle J, M'|V(\Omega)|J, M\rangle \tag{25}$$

for irrep J and row and column indices M', M. The expression for $\mathcal{D}$-functions (25) of SU(2) element $V(\Omega)$ can be calculated by noting that the creation operators transform under the action of V of Eq. (2) according to

$$a_1^\dagger \to V_{11}a_1^\dagger + V_{12}a_2^\dagger,$$
$$a_2^\dagger \to V_{21}a_1^\dagger + V_{22}a_2^\dagger, \tag{26}$$

where V is the 2×2 fundamental representation of $V(\Omega)$. The state $|J, M\rangle$ (24) thus transforms to

$$|J, M\rangle \to \frac{\left(V_{11}a_1^\dagger + V_{12}a_2^\dagger\right)^{J+M}\left(V_{21}a_1^\dagger + V_{22}a_2^\dagger\right)^{J-M}}{\sqrt{(J+M)!(J-M)!}}|0\rangle \tag{27}$$

as the vacuum state $|0\rangle$ is invariant under the action V. Using Eqs. (24) and (27), we obtain

$$D^J_{M'M}(\Omega) = \left\langle 0 \left| \frac{a_1^{J+M'} a_2^{J-M'} \left(V_{11}a_1^\dagger + V_{12}a_2^\dagger\right)^{J+M} \left(V_{21}a_1^\dagger + V_{22}a_2^\dagger\right)^{J-M}}{\sqrt{(J+M')!(J-M')!}\sqrt{(J+M)!(J-M)!}} \right| 0 \right\rangle, \tag{28}$$

which can be evaluated using the commutation relations of the creation and annihilation operators (22).[55]

In this paper, our objective is to generalize Eqs. (21) and (24) systematically from $n = 2$ to arbitrary n. In Subsection III B, we define boson realizations of operators and carrier-space states of $\mathfrak{su}(n)$. Furthermore, we construct the boson realization for the hws of arbitrary $SU(n)$ irreps.

B. SU(n) boson realizations for arbitrary n

$SU(n)$ boson realizations map $SU(n)$ states $\left|\psi^{K^{(n)},\ldots,K^{(3)},K^{(2)}}_{\Lambda^{(n)},\ldots,\Lambda^{(3)},\Lambda^{(2)}}\right\rangle$ and $\mathfrak{su}(n)$ operators to states and operators of a system of bosons on n sites. Bosons are labelled based on the site $i \in \{1,2,\ldots,n\}$ at which they are situated and by an internal degree of freedom, which is denoted by an additional subscript on the bosonic operators. The bosonic creation and annihilation operators on this system are

$$\left\{a_{i,j}^\dagger : i \in \{1,2,\ldots,n\}, j \in \{1,2,\ldots,n-1\}\right\} \quad \text{Creation}, \tag{29}$$

$$\left\{a_{k,l} : k \in \{1,2,\ldots,n\}, l \in \{1,2,\ldots,n-1\}\right\} \quad \text{Annihilation}, \tag{30}$$

where the first label in the subscript is the usual index of the site occupied by the boson. The second index refers to the internal degrees of freedom of the boson. Each boson can have at most $n-1$ possible internal states to ensure that basis states can be constructed for arbitrary irreps. In photonic experiments, this internal degree of freedom could correspond to the polarization, frequency, orbital angular momentum, or the time of arrival of photons.

The $\mathfrak{su}(n)$ operators are mapped to number-preserving bilinear products of boson creation and annihilation operators. Specifically, raising and lowering operators $C_{i,j}$ of $\mathfrak{su}(n)$ map to bosonic operators $c_{i,j}$ according to

$$C_{i,j} \mapsto c_{i,j} := \sum_{k=1}^{n-1} a_{i,k}^\dagger a_{j,k}. \tag{31}$$

Operators $\{c_{i,j}\}$ make bosons hop from site j to site i. The operators h_i are the image of the Cartan operators H_i,

$$H_i \mapsto h_i := a_i^\dagger a_i - a_{i+1}^\dagger a_{i+1}. \tag{32}$$

Operators $\{h_i\}$ count the difference in the total number of bosons at two sites and commute among themselves. As usual, we used the upper-case symbols to denote the $\mathfrak{su}(n)$ elements and the corresponding lower-case symbols for the respective boson operators.

The boson realizations of the basis states of $SU(n)$ are obtained by the action of polynomials in creation operators $\{a_{i,j}^\dagger : i \in \{1,2,\ldots,n\}, j \in \{1,2,\ldots,n-1\}\}$ on the n-site vacuum state $|0\rangle$. Each term in the polynomial is a product of

$$N_K = \kappa_1 + 2\kappa_2 + \cdots + (n-1)\kappa_{n-1} \tag{33}$$

boson creation operators for basis states in irreps $K = (\kappa_1, \kappa_2, \ldots, \kappa_{n-1})$. Therefore, an $SU(n)$ basis state is specified by the coefficient of a polynomial consisting of terms that are products of N_K creation operators.

The hws of a given $SU(n)$ irrep can be explicitly constructed in the boson realization (as polynomials in creation and annihilation operators) according to the following lemma.

Lemma 6 (Boson realization of hws[7,56]). The bosonic state

$$|\psi_{\text{hws}}^K\rangle = \det\begin{pmatrix} a_{1,1}^\dagger & \cdots & a_{1,n-1}^\dagger \\ \vdots & \ddots & \vdots \\ a_{n-1,1}^\dagger & \cdots & a_{n-1,n-1}^\dagger \end{pmatrix}^{\kappa_{n-1}} \cdots \det\begin{pmatrix} a_{1,1}^\dagger & a_{1,2}^\dagger \\ a_{2,1}^\dagger & a_{2,2}^\dagger \end{pmatrix}^{\kappa_2} \det\left(a_{1,1}^\dagger\right)^{\kappa_1} |0\rangle \tag{34}$$

is a hws for a given $\text{SU}(n)$ *irrep* $K = (\kappa_1, \kappa_2, \ldots, \kappa_{n-1})$.

One can verify that the state $|\psi_{\text{hws}}^K\rangle$ (34) is annihilated

$$c_{j,k}|\psi_{\text{hws}}^K\rangle = 0 \quad \forall j < k \tag{35}$$

by the action of any of the raising operators.

Thus, the hws of any irrep can be constructed analytically using Lemma 6. In Sec. IV, we provide an algorithm to construct each of the basis states of arbitrary $\text{SU}(n)$ irreps. Furthermore, we present an algorithm to compute expressions for $\text{SU}(n)$ $\mathcal{D}$-functions in terms of the entries of the fundamental representation.

IV. RESULTS: ALGORITHMS FOR BOSON REALIZATIONS OF SU(n) STATES AND FOR SU(n) $\mathcal{D}$-FUNCTIONS

In this section, we present three algorithms.[57] Algorithm 1 (boson-set algorithm) constructs basis sets for each weight of a given $\mathfrak{su}(n)$ irrep. Algorithm 2 (canonical-basis-states algorithm) employs the boson-set algorithm to compute expressions for the canonical basis states of a given $\text{SU}(n)$ irrep K. The states thus constructed are used by Algorithm 3 to calculate the $\mathcal{D}$-functions of a given $\text{SU}(n)$ transformation.

Algorithms 1 and 2 rely on mapping the $\text{SU}(n)$ irrep to a graph and systematically traversing the graph to obtain basis states. The vertices of the irrep graph are identified with the weights of the given irrep of $\text{SU}(n)$ and the edges with the action of the elements of the Lie algebra $\mathfrak{su}(n)$ on the states. Specifically, the irrep graph $G = (\mathcal{V}, \mathcal{E})$ of an $\text{SU}(n)$ irrep is defined as follows.

Definition 7 (Irrep graph). The bijection

$$v : \{\Lambda_1, \Lambda_2, \ldots, \Lambda_d\} \to \mathcal{V} \tag{36}$$

maps the set $\{\Lambda_1, \Lambda_2, \ldots, \Lambda_d\}$ *of the d weights in the given irrep to the vertices*

$$\mathcal{V} = \{v(\Lambda_1), v(\Lambda_2), \ldots, v(\Lambda_d)\} \tag{37}$$

of its irrep graph. Vertices $v(\Lambda_k)$ *and* $v(\Lambda_\ell)$ *are connected by an edge* $e_j = (v(\Lambda_k), v(\Lambda_\ell)) \in \mathcal{E}$ *iff* $\exists\, c_{i,j}, \Lambda_k, \Lambda_\ell$ *such that*

$$c_{i,j\neq i}|\psi_{\Lambda_k}\rangle = |\psi_{\Lambda_\ell}\rangle, \tag{38}$$

where $|\psi_{\Lambda_k}\rangle$ *and* $|\psi_{\Lambda_\ell}\rangle$ *are* $\text{SU}(n)$ *states that have weights* Λ_k *and* Λ_ℓ, *respectively. In general, states* $|\psi_{\Lambda_k}\rangle$ *and* $|\psi_{\Lambda_\ell}\rangle$ *are linear combinations of canonical basis states. Edges* $\mathcal{E}$ *together with the vertices* $\mathcal{V}$ *define the irrep graph* $G = (\mathcal{V}, \mathcal{E})$.

More than one basis state can have the same weight. The number of basis states sharing a weight Λ_i is defined as the multiplicity $M(\Lambda_i)$ of the weight. In other words, each vertex $v(\Lambda_i)$ is identified with an $M(\Lambda_i)$-dimensional space spanned by those canonical basis states that have weight Λ_i. The vertex space and vertex basis sets are defined as follows.

Definition 8 (Vertex spaces). The vertex space of $v(\Lambda_i)$ *is the span*

$$\Psi(\Lambda_i) = \text{span}\left(|\psi_{\Lambda_i}^1\rangle, |\psi_{\Lambda_i}^2\rangle, \ldots, |\psi_{\Lambda_i}^{M(\Lambda_i)}\rangle\right) \tag{39}$$

of the canonical basis states (Definition 4) that have the weight Λ_i.

The set $\left\{|\psi_{\Lambda_i}^{(1)}\rangle, |\psi_{\Lambda_i}^{(2)}\rangle, \ldots, |\psi_{\Lambda_i}^{(M(\Lambda_i))}\rangle\right\}$ of canonical basis states is not the only set that spans the vertex space $\Psi(\Lambda_i)$ of $v(\Lambda_i)$. In general, basis sets of $\Psi(\Lambda_i)$ can be defined as follows.

ALGORITHM 1. Basis-set algorithm.

Input:
- hws $|\psi_{\text{hws}}^K\rangle$ ▷ Degree N_K (33) polynomial in bosonic creation operators.
- $m \in \mathbb{Z}^+$ ▷ $|\psi_{\text{hws}}^K\rangle$ is a hws of SU(m) irrep K.

Output:
- $\{\Lambda_1, \Lambda_2, \ldots, \Lambda_d : \Lambda_i \in (\mathbb{Z}^+ \cup 0)^{m-1}\}$ ▷ List of weights in the irrep graph of K.
- d, Basis sets (42)

$$\left\{|\phi_{\Lambda_1}^{(1)}\rangle, |\phi_{\Lambda_1}^{(2)}\rangle, \ldots, |\phi_{\Lambda_1}^{(M(\Lambda_1))}\rangle\right\}, \left\{|\phi_{\Lambda_2}^{(1)}\rangle, |\phi_{\Lambda_2}^{(2)}\rangle, \ldots, |\phi_{\Lambda_2}^{(M(\Lambda_2))}\rangle\right\}, \ldots,$$
$$\left\{|\phi_{\Lambda_i}^{(1)}\rangle, |\phi_{\Lambda_i}^{(2)}\rangle, \ldots, |\phi_{\Lambda_i}^{(M(\Lambda_i))}\rangle\right\}, \ldots, \left\{|\phi_{\Lambda_d}^{(1)}\rangle, |\phi_{\Lambda_d}^{(2)}\rangle, \ldots, |\phi_{\Lambda_d}^{(M(\Lambda_d))}\rangle\right\}.$$

1: **procedure** BASISSET ($m, |\psi_{\text{hws}}^K\rangle$)
2:     Initialize empty statesList, empty weightList and currentStateQueue $\leftarrow |\psi_{\text{hws}}^K\rangle$
3:     **while** currentStateQueue is not empty **do**
4:         currentState $\leftarrow$ DEQUEUE (currentStateQueue)
5:         **for** CURRENTOPERATOR $\in$ set of $\mathfrak{su}(m)$ lowering operations **do**
6:             newState $\leftarrow$ CURRENTOPERATOR (currentState)
7:             **if** newState $\neq 0$ **then**
8:                 **if** weight of currentState is already in stateList **then**
9:                     **if** currentState is LI of stateList states with same weight **then**
10:                        independentState $\leftarrow$ NORMALIZE (newState)
11:                        Enqueue independentState in currentStateQueue
12:                        Add independentState to stateList
13:                        Add weight of independentState to weightList
14:                    **end if**   ▷ Else, do nothing.
15:                **else**
16:                    Enqueue newState in currentStateQueue
17:                    Add {weight(newState),newState} to stateList
18:                **end if**
19:            **end if**
20:        **end for**
21:    **end while**
22:    Return stateList
23: **end procedure**

Definition 9 (Vertex basis sets). The set

$$\left\{|\phi_{\Lambda_i}^{(1)}\rangle, |\phi_{\Lambda_i}^{(2)}\rangle, \ldots, |\phi_{\Lambda_i}^{(M(\Lambda_i))}\rangle\right\} \tag{40}$$

is called the basis set of a vertex $v(\Lambda_i)$ if it spans the vertex space $\Psi(\Lambda_i)$ (39) of $v(\Lambda_i)$, i.e.,

$$\text{span}\left(|\phi_{\Lambda_i}^1\rangle, |\phi_{\Lambda_i}^2\rangle, \ldots, |\phi_{\Lambda_i}^{M(\Lambda_i)}\rangle\right) = \Psi(\Lambda_i). \tag{41}$$

The states $\left\{|\phi_{\Lambda_i}^{(1)}\rangle, |\phi_{\Lambda_i}^{(2)}\rangle, \ldots, |\phi_{\Lambda_i}^{(M(\Lambda_i))}\rangle\right\}$ are linear combinations of the canonical basis states $\{|\psi_{\Lambda_i}^{(1)}\rangle,$ $|\psi_{\Lambda_i}^{(2)}\rangle, \ldots, |\psi_{\Lambda_i}^{(M(\Lambda_i))}\rangle\}$. Algorithm 1 computes basis sets of the spaces $\Psi(\Lambda_i)$ for each of the d weights Λ_i that occurs in a given irrep.

A. Basis-set algorithm (Algorithm 1)

The basis-set algorithm, which finds the basis sets for a given SU(m) irrep, is the key subroutine of our canonical-basis-state algorithm. Algorithm 1 requires inputs $|\psi_{\text{hws}}^K\rangle$ and m, where $|\psi_{\text{hws}}^K\rangle$ is a hws of the irrep K of $\mathfrak{su}(m)$ algebra. The state $|\psi_{\text{hws}}^K\rangle$ is a bosonic state, which is expressed as a summation over products of N_K (33) creation operators $\{a_{i,j}^\dagger : i \in \{1, 2, \ldots, m\}, j \in \{1, 2, \ldots, m-1\}\}$. This summation acts on the m-site bosonic vacuum state to give an N_K-boson state. The algorithm returns

ALGORITHM 2. Canonical-basis-states algorithm.

Input:
- $n \in \mathbb{Z}^+$ ▷ Algorithm constructs basis sets of $\mathfrak{su}(n)$ algebra.
- $K = (\kappa_1, \kappa_2, \ldots, \kappa_{n-1}) \in (\mathbb{Z}^+ \cup \{0\})^{n-1}$ ▷ Label of SU(n) irrep.

Output:
- $\left\{ \left(\left| \psi_{\Lambda^{(n)}, \ldots, \Lambda^{(3)}, \Lambda^{(2)}}^{K^{(n)}, \ldots, K^{(3)}, K^{(2)}} \right\rangle; K^{(n)}, \ldots, K^{(2)}; \Lambda^{(n)}, \ldots, \Lambda^{(2)} \right) \right\}$ ▷ List of all canonical basis states and weight labels in the irrep $K^{(n)} = K$.

1: **procedure** CanonicalBasisStates(n, K)
2:     Initialize empty basisStatesList, hws $\leftarrow |\psi_{\text{hws}}^{K}\rangle$
3:     SUmStates, SUnStates $\leftarrow$ BasisSet(n, hws)
4:   **while** SUnStates is not empty **do**
5:       **for** $m \in \{n, n-1, \ldots, 2\}$ **do**
6:           $\Lambda_{\max} \leftarrow \mathfrak{su}(m)$ weight with highest number of states in SUmStates.
7:           $|\psi_{\max}^{(m)}\rangle \leftarrow$ arbitrary superposition of states at $\Lambda_{\max}$ in SUmStates.
8:           Apply $\mathfrak{su}(m-1)$ raising operators on $|\psi_{\max}^{(m)}\rangle$; reach $\mathfrak{su}(m-1)$ hws $|\psi_{\text{hws}}^{(m-1)}\rangle$.
9:           $K^{(m-1)} \leftarrow$ Weight$\left(|\psi_{\text{hws}}^{(m-1)}\rangle \right)$.
10:           SUmStates $\leftarrow$ BasisSet$\left(m-1, |\psi_{\text{hws}}^{(m-1)}\rangle \right)$.
11:         **if** $m = 2$ **then**
12:             **for** All states $|\psi\rangle$ in SUmStates **do**
13:                 $\{\Lambda^{(n)}, \ldots, \Lambda^{(2)}\} \leftarrow$ Weights$(|\psi\rangle)$ ▷ $\mathfrak{su}(m)$ weights $\forall m \leq n$.
14:                 Concatenate $\left(|\psi\rangle; K^{(n)}, \ldots, K^{(2)}; \Lambda^{(n)}, \ldots, \Lambda^{(2)} \right)$ basisStatesList
15:                 Subtract SUmStates from SUnStates.
16:             **end for**
17:         **end if**   ▷ Else, do nothing.
18:       **end for**
19:     **end while**
20:     **for** All states $|\psi^{(i)}\rangle$ in statelist **do**
21:       Act $\{C_{1,2}, C_{2,3}, \ldots, C_{n-1,n}\}$ on $|\psi^{(i)}\rangle$ until hws $|\psi_{\text{hws}}^{(i)}\rangle$ is reached.
22:       $|\psi^{(i)}\rangle \leftarrow e^{i\phi^{(i)}} |\psi^{(i)}\rangle$ for $|\psi_{\text{hws}}^{(i)}\rangle = e^{i\phi^{(i)}} |\psi_{\text{hws}}^{K}\rangle$.
23:     **end for**
24:     Return basisStatesList
25: **end procedure**

---

multiple sets

$$\left\{ |\phi_{\Lambda_1}^{(1)}\rangle, |\phi_{\Lambda_1}^{(2)}\rangle, \ldots, |\phi_{\Lambda_1}^{(M(\Lambda_1))}\rangle \right\}, \left\{ |\phi_{\Lambda_2}^{(1)}\rangle, |\phi_{\Lambda_2}^{(2)}\rangle, \ldots, |\phi_{\Lambda_2}^{(M(\Lambda_2))}\rangle \right\}, \ldots,$$
$$\left\{ |\phi_{\Lambda_i}^{(1)}\rangle, |\phi_{\Lambda_i}^{(2)}\rangle, \ldots, |\phi_{\Lambda_i}^{(M(\Lambda_i))}\rangle \right\}, \ldots, \left\{ |\phi_{\Lambda_d}^{(1)}\rangle, |\phi_{\Lambda_d}^{(2)}\rangle, \ldots, |\phi_{\Lambda_d}^{(M(\Lambda_d))}\rangle \right\} \tag{42}$$

of $\mathfrak{su}(m)$ states, with each set spanning space $\Psi(\Lambda_i)$ (39) at a different vertex $v(\Lambda_i)$ in the SU(m) irrep K. The states in the output basis sets are represented as polynomials in lowering operators acting on the hws, or equivalently as polynomials in creation and annihilation operators acting on the n-site vacuum state. Figure 3 is an illustrative example of the algorithm.

A modified breadth-first search (BFS) graph algorithm[58–60] is used to traverse the irrep graph for states. As in usual BFS, we maintain a queue,[61] called currentQueue, of the states that have been constructed but whose neighbourhood is yet to be explored. The algorithm starts with the given hws in currentQueue and iteratively dequeues a state from the front of the queue. States neighbouring the dequeued state are obtained by enacting one-by-one each of the lowering operators of the algebra. The newly found states are enqueued into the rear of currentQueue, and the current state and its weight are stored.

We modify BFS to handle vertices with weight multiplicity greater than unity as follows. While traversing the irrep graph, the algorithm directly enqueues the first state that is found at each vertex. When the same vertex is explored along a different edge, i.e., by enacting different lowering operators, a different state is found in general. If the newly constructed state is LI of the states already constructed at the vertex, then the new state is enqueued into currentQueue.

ALGORITHM 3. $\mathcal{D}$-function algorithm.

Input:
- $n \in \mathbb{Z}^+$ ▷ Algorithm constructs $\mathcal{D}$-functions of SU(n) elements.
- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_{n^2-1}\} \in \mathbb{R}^{n^2-1}$ ▷ Parametrization of SU(n) transformation.
- $K^{(n)}, \ldots, K^{(2)}$ and $\Lambda^{(n)}, \ldots, \Lambda^{(2)}$ ▷ Row Label.
- $K'^{(n)}, \ldots, K'^{(2)}$ and $\Lambda'^{(n)}, \ldots, \Lambda'^{(2)}$ ▷ Column Label.

Output:
- $\mathcal{D}^{K^{(n)}, \ldots, K^{(3)}, K^{(2)}; K'^{(n)}, \ldots, K'^{(3)}, K'^{(2)}}_{\Lambda^{(n)}, \ldots, \Lambda^{(3)}, \; \Lambda^{(2)} \; ; \Lambda'^{(n)}, \ldots, \Lambda'^{(3)}, \; \Lambda'^{(2)}}(\Omega)$

1: **procedure** D($n, \Omega, K^{(n)}, \ldots, K^{(2)}, K'^{(n)}, \ldots, K'^{(2)}, \Lambda^{(n)}, \ldots, \Lambda^{(2)}, \Lambda'^{(n)}, \ldots, \Lambda'^{(2)}$)
2:      Construct $V \in GL(n, \mathbb{C})$ from parametrization Ω[54]
3:      **if** $K^{(n)} = K'^{(n)}$ **then**
4:          $\left| \psi^{K^{(n)}, \ldots, K^{(3)}, K^{(2)}}_{\Lambda^{(n)}, \ldots, \Lambda^{(3)}, \Lambda^{(2)}} \right\rangle \leftarrow$ using CANONICALBASISSTATES ($n, K(n)$).
5:          $\left| \psi^{K'^{(n)}, \ldots, K'^{(3)}, K'^{(2)}}_{\Lambda'^{(n)}, \ldots, \Lambda'^{(3)}, \Lambda'^{(2)}} \right\rangle \leftarrow$ using CANONICALBASISSTATES ($n, K'(n)$).
6:          Construct $\left\langle \psi^{K^{(n)}, \ldots, K^{(3)}, K^{(2)}}_{\Lambda^{(n)}, \ldots, \Lambda^{(3)}, \Lambda^{(2)}} \right|$ from $\left| \psi^{K^{(n)}, \ldots, K^{(3)}, K^{(2)}}_{\Lambda^{(n)}, \ldots, \Lambda^{(3)}, \Lambda^{(2)}} \right\rangle$ by complex conjugation.
7:          Construct $V(\Omega) \left| \psi^{K'^{(n)}, \ldots, K'^{(3)}, K'^{(2)}}_{\Lambda'^{(n)}, \ldots, \Lambda'^{(3)}, \Lambda'^{(2)}} \right\rangle$ using $a^\dagger_{i,k} \to \sum_j V_{i,j}(\Omega) a^\dagger_{j,k} \; \forall \, a_{i,k}, a^\dagger_{i,k}$.
8:          Return $\mathcal{D} = \left\langle \psi^{K^{(n)}, \ldots, K^{(3)}, K^{(2)}}_{\Lambda^{(n)}, \ldots, \Lambda^{(3)}, \Lambda^{(2)}} \right| V(\Omega) \left| \psi^{K'^{(n)}, \ldots, K'^{(3)}, K'^{(2)}}_{\Lambda'^{(n)}, \ldots, \Lambda'^{(3)}, \Lambda'^{(2)}} \right\rangle$
9:      **else**
10:     Return $\mathcal{D} = 0$
11:     **end if**
12: **end procedure**

The algorithm truncates when a state in currentQueue is annihilated by all of the lowering operators and there is no other state in the queue. This final state must exist because the number of LI states in a given SU(n) irrep is finite according to the following standard result in representation theory.

Lemma 10 (Dimension of an SU(n) *irrep[28]). The dimension Δ_K of the carrier space of an* SU(n) *irrep K is*
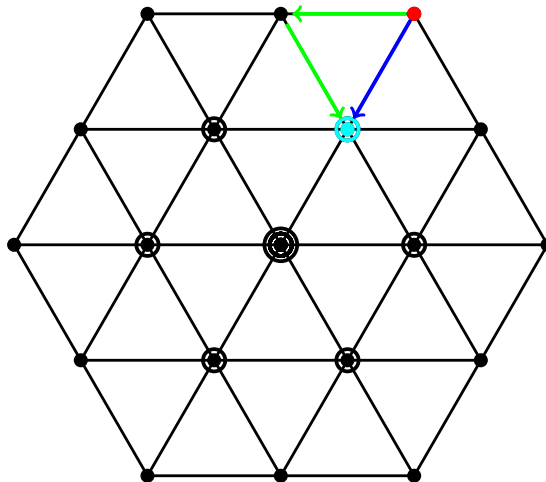

FIG. 3. The first step of the basis-set computation algorithm (Algorithm 1) illustrated for the (2, 2) irrep of SU(3), where the dimension of the space of states at a given vertex is the sum of the number of dots and the number of circles at the vertex. The algorithm constructs the hws (occupying the red vertex) using Lemma 6. The lowering operators can transform states at one vertex to states at another vertex along different paths connecting the starting and the target vertex, for instance, the two paths coloured green and blue. Lowering along the different paths to reach a target vertex will generate the same number of LI as the weight multiplicity. In our illustration, we obtain a basis set that contains two independent states at the target vertex. The algorithm traverses the irrep graph systematically until all basis sets are calculated.

$$\Delta_K := M(\Lambda_1) + M(\Lambda_2) + \cdots + M(\Lambda_d)$$
$$= (1 + \kappa_1)(1 + \kappa_2) \cdots (1 + \kappa_{n-1}) \left(1 + \frac{\kappa_1 + \kappa_2}{2}\right) \left(1 + \frac{\kappa_2 + \kappa_3}{2}\right)$$
$$\cdots \left(1 + \frac{\kappa_{n-2} + \kappa_{n-1}}{2}\right) \left(1 + \frac{\kappa_1 + \kappa_2 + \kappa_3}{3}\right) \left(1 + \frac{\kappa_2 + \kappa_3 + \kappa_4}{3}\right)$$
$$\cdots \left(1 + \frac{\kappa_{n-3} + \kappa_{n-2} + \kappa_{n-1}}{3}\right) \cdots \left(1 + \frac{\kappa_1 + \kappa_2 + \cdots + \kappa_{n-1}}{n-1}\right). \tag{43}$$

Now we prove that the basis-set algorithm terminates. The proof relies on the fact that the carrier space of SU(m) irrep is finite-dimensional (Lemma 10). The algorithm's computational cost is quantified by the number of times the lowering operators are applied on the hws or on states reached by lowering from the hws. We show that the computational cost of Algorithm 1 is linear in the dimension Δ_K of the irrep whose hws is given as input and polynomial in n.

Theorem 11 (Algorithm 1 terminates). *Suppose Algorithm 1 receives as input an hws $|\psi_{hws}^K\rangle$ of an SU(m) irrep K. Then the algorithm terminates after no more than $\Delta_K m(m-1)/2$ applications of lowering operators.*

Proof. The proof is in two parts. First, the number of states that enters currentStateQueue is bounded above by dimension Δ_K (43) of the irrep space. Second, as each state that enters currentStateQueue is acted upon by no more than $n(n-1)/2$ lowering operators, the number of lowering operations performed is less than or equal to $\Delta_K n(n-1)/2$.

We show that the number of states that enter currentStateQueue is no more than Δ_K as follows. As each currentState is a linear combination of states obtained by acting lowering operators (Line 6) on the given hws, each state that enters currentStateQueue is in the irrep labelled by the hws. Moreover, each state entering the queue is tested for linear independence (Line 9) with respect to the states already obtained. Any state that is not LI is discarded. Therefore, each enqueued state (Algorithm 1, Line 16) is in the correct irrep K and is LI of each other enqueued state. Thus, the number of states that ever enter currentStateQueue is no more than the number Δ_K of LI states in irrep K.

In each iteration of the algorithm, we act all the lowering operators on the states in currentStateQueue. The number of lowering operations is thus bounded above by the product $\Delta_K n(n-1)/2$ of the number of states that enter currentStateQueue and of the number of lowering operators in the $\mathfrak{su}(n)$ algebra. The algorithm thus terminates after no more than $\Delta_K n(n-1)/2$ applications of lowering operators. ∎

We now prove that the algorithm returns the correct output on termination. The proof requires the following lemma stating that each canonical basis state can be obtained by enacting only with the lowering operators on the hws.

Lemma 12 (Every basis-state can be reached by lowering from the hws[62]). No canonical basis-state of a given SU(n) irrep K is LI of the states obtained by lowering from the hws by the action

$$c_{i_k, j_k} \cdots c_{i_2, j_2} c_{i_1, j_1} |\psi_{\text{hws}}\rangle \quad i_\ell \le j_\ell \, \forall 1 \le \ell \le k \tag{44}$$

of $k \le \sum_i \kappa_i$ number of $\mathfrak{su}(n)$ lowering operators on the hws of the irrep.

Lemma 12 implies that each basis state can be constructed by linearly combining states obtained on lowering from the hws. Algorithm 1 leverages from the construction of Eq. (44) and from testing linear independence to construct the basis sets.

The correctness of the basis-set algorithm is proved as follows. We show that each state obtained by enacting any number of lowering operators on the hws is linearly dependent (LD) on the states returned by the algorithm. Each canonical basis state is LD on the states obtained by lowering from the hws in turn, so each canonical basis state is LD on the algorithm output. The algorithm only constructs states in the correct irrep so Algorithm 1 returns a complete basis set at each weight of the irrep on truncation.

Theorem 13 (Algorithm 1 is correct). *The sets*

$$\left\{|\phi_{\Lambda_1}^{(1)}\rangle, |\phi_{\Lambda_1}^{(2)}\rangle, \ldots, |\phi_{\Lambda_1}^{(M(\Lambda_1))}\rangle\right\}, \left\{|\phi_{\Lambda_2}^{(1)}\rangle, |\phi_{\Lambda_2}^{(2)}\rangle, \ldots, |\phi_{\Lambda_2}^{(M(\Lambda_2))}\rangle\right\}, \ldots,$$

$$\left\{|\phi_{\Lambda_i}^{(1)}\rangle, |\phi_{\Lambda_i}^{(2)}\rangle, \ldots, |\phi_{\Lambda_i}^{(M(\Lambda_i))}\rangle\right\}, \ldots, \left\{|\phi_{\Lambda_d}^{(1)}\rangle, |\phi_{\Lambda_d}^{(2)}\rangle, \ldots, |\phi_{\Lambda_d}^{(M(\Lambda_d))}\rangle\right\}$$

of states returned by Algorithm 1 span the respective vertex spaces $\Psi(\Lambda_i)$ *(39) at each vertex* Λ_i *of the given irrep K.*

Proof. We first prove by induction that each state in the form of Eq. (44) is LD on states in the algorithm output. Our induction hypothesis is that each state

$$c_{i_k,j_k} \ldots c_{i_2,j_2} c_{i_1,j_1}|\psi_{\text{hws}}\rangle, \tag{45}$$

which is obtained by acting ℓ lowering operators on the hws, is LD on the states returned by the algorithm $\forall \ell \in \mathbb{Z}^+$. The proof of the hypothesis follows from mathematical induction over ℓ.

The induction hypothesis is true for base case $k = 1$. In the first iteration, the algorithm enacts all the lowering operators on the hws (Algorithm 1, Line 6) and saves each of the obtained states. No $k = 1$ state (45) is omitted because the vertices neighbouring the hws vertex are all being explored for the first time. Hence, all the states that can be reached by lowering once from the hws are added to currentStateQueue and, eventually, to stateList.

Assume that the induction hypothesis holds for $k = \ell$, i.e., each $k = \ell$ state is LD on the states in stateList. We prove that the hypothesis holds for $k = \ell + 1$ by contradiction. Suppose there exists a state that can be reached by enacting $\ell + 1$ lowering operators on the hws but is LI of stateList. Let $|\psi\rangle = c_{i_{\ell+1},j_{\ell+1}} c_{i_\ell,j_\ell} \ldots c_{i_2,j_2} c_{i_1,j_1}|\psi_{\text{hws}}\rangle$ be such a state.

Consider now the state $|\varphi\rangle = c_{i_\ell,j_\ell} \ldots c_{i_2,j_2} c_{i_1,j_1}|\psi_{\text{hws}}\rangle$ obtained by enacting one less lowering operation from the hws, i.e., $|\psi\rangle = c_{i_{\ell+1},j_{\ell+1}}|\varphi\rangle$. We have assumed that the induction hypothesis holds for $k = \ell$. Therefore, $|\varphi\rangle$ is LD on the states constructed by a algorithm. In other words,

$$|\varphi\rangle = \sum_{j=1}^{J} a_j|\phi_j\rangle \tag{46}$$

is LD on the stateList elements $\{|\phi_j\rangle : j \in \{1, 2, \ldots, J\}\}$ for complex numbers a_j.

The algorithm enacts the lowering operator $c_{i_{\ell+1},j_{\ell+1}}$ on each $|\phi_j\rangle$ and the resulting states are either stored in stateList or are LD on elements in stateList. Therefore, the elements of the set $\{c_{i_{\ell+1},j_{\ell+1}}|\phi_j\rangle : j \in \{1, 2, \ldots, J\}\}$ are LD on the elements of stateList. Hence, the element $c_{i_{\ell+1},j_{\ell+1}}|\varphi\rangle$ is also LD on the elements of stateList. This dependence contradicts the supposition that $|\psi\rangle = c_{i_{\ell+1},j_{\ell+1}}|\varphi\rangle$ is LI of stateList, thereby proving the induction hypothesis for $k = \ell + 1$.

The induction hypothesis is true for $\ell = 1$ and is shown to hold for $k = \ell + 1$ if it holds for $k = \ell$. Thus, our induction hypothesis is true for all $\ell \in \mathbb{Z}^+$. Every state obtained of irrep K obtained by lowering from the hws is LD on the basis sets that are returned by the algorithm.

We know from Lemma 12 that each canonical basis state is LD on the states obtained by lowering. Hence, each canonical basis state is LD on the states obtained at the output of the algorithm. Therefore, the state returned by the algorithm span the space of irrep K states, and the output basis sets span the set of all basis states of the given irrep K. ∎

We have proved that Algorithm 1 terminates and that it returns the correct basis sets on termination. Now we present our algorithm for the construction of the canonical basis states. Furthermore, we prove the correctness and termination of the canonical-basis-states algorithm.

B. Canonical-basis-states algorithm (Algorithm 2)

The algorithm for constructing the canonical basis-states of SU(n) requires inputs $n \in \mathbb{Z}^+$ and the irrep label K. The algorithm returns expressions for all the canonical basis states in the given irrep. Figure 4 illustrates SU(3) basis-state construction using our algorithm. Algorithm 2 details the step-by-step construction of the canonical basis states.
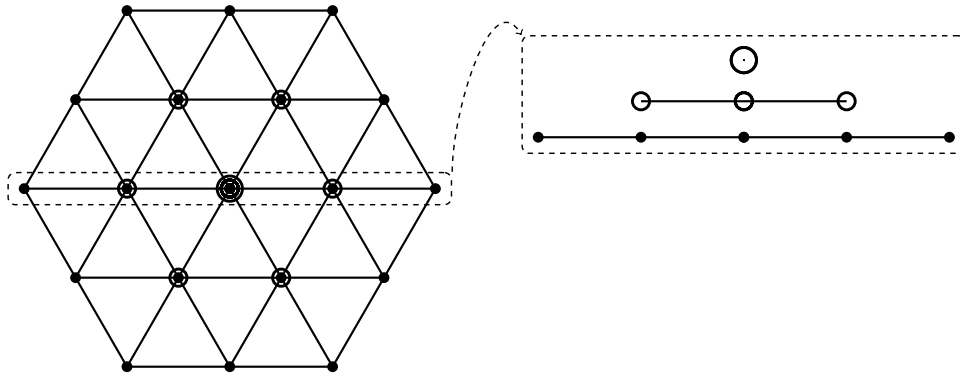
FIG. 4. Diagrammatic representation of the main algorithm for $n = 3$. The dots and circles represent the canonical basis states. The dimension of the space of states at a given vertex is the sum of the number of dots and the number of circles at the vertex, for instance, weights associated with dimension two are represented by one dot and one circle. The lines connecting the dots represent the transformation from states of one weight to those of another by the action of SU(3) raising and lowering operators. We use Algorithm 1 to construct basis sets for each vertex in the SU(n) irrep graph. Once the basis sets for the SU(n) irreps are computed, the algorithm enacts the $\mathfrak{su}(n-1)$ raising operators on the $(n-1)$-dimensional sub-irreps to find the $\mathfrak{su}(n-1)$ hws. Then, the algorithm starts with the $\mathfrak{su}(n-1)$ hws and employs the basis-set construction (Algorithm 1) to find all the states in the $\mathfrak{su}(n-1)$ irrep labelled by the hws. The states thus obtained are subtracted from the set of $\mathfrak{su}(n)$ states. A new state is chosen from the weight of highest multiplicity and the process repeated until all the $\mathfrak{su}(n-1)$ irreps are found.

The canonical-basis-states algorithm proceeds by partitioning $\mathfrak{su}(n)$ basis sets into $\mathfrak{su}(m)$ basis sets for progressively smaller m over $n-1$ stages. In the first stage, the algorithm employs Lemma 6 to construct the hws of the given irrep K (Algorithm 2, Line 2). Algorithm 1 is then used to construct the basis sets of the SU(n) irrep of the constructed hws (Line 3).

By the $(n-m)$-th stage, the algorithm has partitioned the entire $\mathfrak{su}(n)$ space into basis states of the SU($m+1$) irreps. In this stage, each of the $\mathfrak{su}(m+1)$ basis sets is partitioned into $\mathfrak{su}(m)$ basis sets by using $\mathfrak{su}(m)$ operators. The algorithm searches each SU($m+1$) irrep graph for the vertex that has the highest multiplicity.

An arbitrary linear combination of the basis states at this vertex is chosen. The algorithm then enacts all the raising operators in the $\mathfrak{su}(m)$ subalgebra on this linear combination until the action of each of the raising operators annihilates the state. The state thus obtained is the hws of an SU(m) irrep, whose label $K^{(m)}$ can be calculated by enacting the Cartan operators on the state.

Next the algorithm performs the basis-set construction algorithm on the $\mathfrak{su}(m)$ hws employing only the $\mathfrak{su}(m)$ lowering operators. This procedure gives us sets of basis states that belong to the SU(m) irrep $K^{(m)}$. The irrep $K^{(m)}$ basis sets are stored and are then subtracted from the SU($m+1$) states. The algorithm iteratively (i) starts from the highest multiplicity vertex of SU($m+1$) irrep graphs, (ii) constructs a hws by raising, (iii) stores the basis sets of SU(m) irreps corresponding to this hws, and (iv) subtracts them from SU($m+1$) states until all the states in the $\mathfrak{su}(m+1)$ are partitioned.

At the end of $n-1$ stages, we have a list of basis sets of the SU($n-1$) irreps. We iteratively perform the process of finding basis sets for smaller subgroups until we reach SU(2) basis sets, which are known to have unit multiplicity. Hence, the algorithm returns the basis states that are eigenvectors of the Cartan operators of all SU(m): $m \leq n$ groups.

The relative phases between the basis states are fixed by imposing Eq. (18). Each of the constructed basis states is acted upon by the simple raising operators $\{C_{1,2}, C_{2,3}, \ldots, C_{n-1,n}\}$ until the hws is reached. The phase of this hws obtained by raising is required to be the same for all basis states. Our algorithm multiplies each of the basis states by a phase factor (Line 22) to impose the phase convention Eq. (18) and returns the set of canonical basis states.

Now we prove that the canonical basis states algorithm terminates. The proof of termination uses the facts that the number of basis states is equal to the dimension Δ_K of the irrep and that each basis state is added to currentStateQueue no more than once.

Theorem 14 (Algorithm 2 terminates). *Algorithm 2 terminates after the action of no more than* $\Delta_K n(n-1)^2/2$ *lowering operators.*

Proof. In each of the $n-1$ stages of Algorithm 2, the states that are added to currentStateQueue are LI of each other because of the conditions imposed in the algorithm. There are no more LI states in the given SU(n) irrep than the dimension Δ_K of the irrep space. Thus, the total number of states that are added to currentQueue in each of the $n-1$ stages is no more than Δ_K. No more that $n(n-1)/2$ lowering operators are applied on the states that enter currentQueue. Thus, each stage terminates after the application of $\Delta_K n(n-1)/2$ lowering operations. Furthermore, the algorithm terminates after $n-1$ stages and the application of no more than $\Delta_K n(n-1)^2/2$ lowering operations. ∎

Finally, we prove that the canonical-basis-states algorithm returns the correct output when it terminates.

Theorem 15 (Algorithm 2 is correct). *The* SU(n) *states*

$$\left| \psi^{K^{(z)},\ldots,K^{(3)},K^{(2)}}_{\Lambda^{(n)},\ldots,\Lambda^{(3)},\,\Lambda^{(2)}} \right\rangle \tag{47}$$

yielded by Algorithm 2 are the canonical states of Definition 4.

Proof. The theorem holds if the states yielded by Algorithm 2 have well defined weights and have well defined irrep labels. First, we show that the weight of each state in the output of the algorithm is well defined. Each state in the output is obtained either by enacting lowering operators on the hws or by taking linear combinations of states that have the same weight. Linear combination of states with the same weights has well defined weights themselves. Thus, all the output states have well defined weights for SU(m) irreps for all $2 \le m \le n$.

We prove that the states have well defined SU(m) irrep label separately for $m = n$ and for $2 \le m \le n-1$. The correctness of the $\mathfrak{su}(m)$ hws follows from Lemma 6. Every state in the output is a linear combination of states obtained by lowering from the constructed $\mathfrak{su}(m)$ hws. Thus, every state is in the correct SU(n) irrep $K^{(n)}$.

The algorithm (Line 8) enacts raising operators on linear combinations of $\mathfrak{su}(m+1)$ basis states at one weight until each of the raising operators annihilates the raised state. The $\mathfrak{su}(m)$ states thus obtained are legitimate $\mathfrak{su}(m)$ hws's or possibly linear combinations of $\mathfrak{su}(m)$ hws's by construction. The uniqueness of the hws is guaranteed by the existence of the canonical basis.[62] Each of the canonical basis states is obtained by lowering from these $\mathfrak{su}(m)$ hws's using $\mathfrak{su}(m)$ lowering operators and thus have well defined irrep labels for all $2 \le m \le n-1$.

We have shown that the states yielded by the algorithm have well defined values of irrep labels $K^{(m)}$ for $\mathfrak{su}(m)$ algebras for all $\{m : 2 \le m \le n\}$ and of $\mathfrak{su}(m)$ weights $\Lambda^{(m)}$ for all $\{m : 2 \le m \le n\}$. Thus, these states are the canonical SU(n) basis states. This completes the proof of correctness of Algorithm 2. ∎

We have proved that Algorithm 2 terminates and returns the canonical basis states on termination. The states constructed by the canonical-basis-states algorithm are employed to compute arbitrary SU(n) $\mathcal{D}$-functions using an algorithm presented in Subsection IV C.

C. $\mathcal{D}$-function algorithm

Our task is to construct the $\mathcal{D}$-function

$$\mathcal{D}^{K^{(n)},\ldots,K^{(3)},K^{(2)};K'^{(n)},\ldots,K'^{(3)},K'^{(2)}}_{\Lambda^{(n)},\ldots,\Lambda^{(3)},\,\Lambda^{(2)};\Lambda'^{(n)},\ldots,\Lambda'^{(3)},\,\Lambda'^{(2)}}(\Omega) \tag{48}$$

for given labels $\{K^{(m)}\}, \{\Lambda^{(m)}\}, \{K'^{(m)}\}, \{\Lambda'^{(m)}\}$ of the SU(n) element $V(\Omega)$ given by the parametrization Ω. $\mathcal{D}$-function (48) is computed as the inner product between the state

$$\left| \psi^{K^{(n)},\ldots,K^{(3)},K^{(2)}}_{\Lambda^{(n)},\ldots,\Lambda^{(3)},\,\Lambda^{(2)}} \right\rangle \tag{49}$$

of Eq. (51) and the transformed state

$$V(\Omega)\left|\psi^{K'^{(n)},\ldots,K'^{(3)},K'^{(2)}}_{\Lambda'^{(n)},\ldots,\Lambda'^{(3)},\Lambda'^{(2)}}\right\rangle. \tag{50}$$

Algorithm 3 constructs the fundamental representation, i.e., the $n \times n$ matrix, V_{ij} of the SU(n) element $V(\Omega)$.[54] Then, the expressions for the basis states

$$\left|\psi^{K^{(n)},\ldots,K^{(3)},K^{(2)}}_{\Lambda^{(n)},\ldots,\Lambda^{(3)},\Lambda^{(2)}}\right\rangle, \left|\psi^{K'^{(n)},\ldots,K'^{(3)},K'^{(2)}}_{\Lambda'^{(n)},\ldots,\Lambda'^{(3)},\Lambda'^{(2)}}\right\rangle \tag{51}$$

corresponding to the given labels are computed using the canonical-basis-states algorithm. The basis states thus obtained are expressed as summations over products of creation and annihilation operators. $V(\Omega)$ acts on the boson realization by transforming each boson independently according to

$$a^{\dagger}_{i,j} \to a^{\dagger\prime}_{i,j} = \sum_k V_{ik}(\Omega)a^{\dagger}_{k,j}, \tag{52}$$

where $\{V_{ik}(\Omega)\}$ are the matrix elements of the $n \times n$ representation of $V(\Omega)$. The algorithm transforms the second basis state of Eq. (51) under the action of $V(\Omega)$ by replacing each of the creation and annihilation operators of the state according to Eq. (52).

The $\mathcal{D}$-function is evaluated as the inner product using commutation relations (22) or equivalently by using the Wick's theorem.[63] The correctness and termination of Algorithm 3 follows directly from Theorems 14 and 15, which completes our algorithms for the computation of boson realizations of SU(n) states and of $\mathcal{D}$-functions.

V. CONCLUSION

In summary, we have devised an algorithm to compute expressions for boson realizations of the canonical basis states of SU(n) irreps. Boson realizations are ideally suited for analyzing the physics of single photons, providing a tractable interpretation to basis states as multiphoton states, and to transformations on these states as optical transformations. Furthermore, we have devised an algorithm to compute expressions for SU(n) $\mathcal{D}$-functions in terms of elements of the fundamental representation of the group. Our algorithm offers significant advantage over competing algorithms to construct $\mathcal{D}$-functions. Furthermore, our $\mathcal{D}$-function algorithm lays the groundwork for generalizing the analysis of optical interferometry beyond the three-photon level.[16–18]

This work is the first known application of graph-theoretic algorithms to SU(n) representation theory. We overcome the problem of SU(n) weight multiplicity greater than unity by modifying the breadth-first graph-search algorithm. Our procedure for generating a basis set can be extended to subgroups of SU(n). In particular, the boson realization of the hws of O($2k$) and O($2k + 1$) irreps can be constructed along the lines of Lemma 6.[64–66] Graph-search algorithms can be employed to construct O(n) basis states and $\mathcal{D}$-functions if the problem of labelling O(n) basis states can be overcome. Our approach opens the possibility of exploiting the diverse graph-algorithms toolkit for solving problems in representation theory of Lie groups.

APPENDIX A: CHOICE OF SUBALGEBRA CHAIN

Our algorithms construct canonical basis states that reduce subalgebra chain (16). Other $\mathfrak{su}(n) \supset \mathfrak{su}(n-1) \supset \cdots \supset \mathfrak{su}(2)$ subalgebra chains are possible and our algorithm can be generalized to construct canonical basis states that reduce other chains, as we discuss in this appendix.

Each $\mathfrak{su}(m)$ subalgebra of $\mathfrak{su}(n)$, $m < n$ is specified by the sets of raising, lowering, and Cartan operators that generate it. For a given sequence

$$I^{(m)} = \left\{ i_1^{(m)}, i_2^{(m)}, \ldots, i_m^{(m)} \right\} \tag{A1}$$

of m increasing integers, we can define the corresponding set of raising, lowering, and Cartan operators

$$\left\{ C_{i_1,i_2}, C_{i_1,i_3}, \ldots, C_{i_1,i_m}, C_{i_2,i_3}, \ldots, C_{i_2,i_m}, \ldots, C_{i_{m-1},i_m} \right\} \qquad \text{(Raising)}, \tag{A2}$$

$$\left\{ C_{i_2,i_1}, C_{i_3,i_1}, \ldots, C_{i_m,i_1}, C_{i_3,i_2}, \ldots, C_{i_m,i_2}, \ldots, C_{i_m,i_{m-1}} \right\} \qquad \text{(Lowering)}, \tag{A3}$$

$$\left\{ C_{i_2,i_2} - C_{i_1,i_1}, C_{i_3,i_3} - C_{i_2,i_2}, \ldots, C_{i_m,i_m} - C_{i_{m-1},i_{m-1}} \right\} \qquad \text{(Cartan)} \tag{A4}$$

that generate the algebra. Thus, each $\mathfrak{su}(n) \supset \mathfrak{su}(n-1) \supset \cdots \supset \mathfrak{su}(2)$ subalgebra chain is uniquely specified by the ordered sequences $I^{(m)}$: $m < n$ of integers, where

$$
\begin{aligned}
I^{(n-1)} &= \{ i_1^{(n-1)}, i_2^{(n-1)}, \ldots, i_{n-1}^{(n-1)} \} \subset \{1, 2, \ldots, n\}, \\
I^{(n-2)} &= \{ i_1^{(n-2)}, i_2^{(n-2)}, \ldots, i_{n-1}^{(n-2)} \} \subset I^{(n-1)}, \\
&\cdots \\
I^{(m-1)} &= \{ i_1^{(m-1)}, i_2^{(m-1)}, \ldots, i_m^{(m-1)} \} \subset I^{(m)}, \\
&\cdots \\
I^{(2)} &= \{ i_1^{(2)}, i_2^{(2)} \} \subset I^{(3)}.
\end{aligned}
\tag{A5}
$$

Consider the example of $\mathfrak{su}(2)$ subalgebras of $\mathfrak{su}(3)$. The three subsets

$$\{ C_{1,2}, C_{2,1}, C_{1,1} - C_{2,2} \}, \tag{A6}$$

$$\{ C_{1,3}, C_{3,1}, C_{1,1} - C_{3,3} \}, \tag{A7}$$

$$\{ C_{2,3}, C_{3,2}, C_{2,2} - C_{3,3} \} \tag{A8}$$

of the generators $\{ C_{i,j} : i, j \in \{1, 2, 3\} \}$ of $\mathfrak{su}(3)$ generate three distinct $\mathfrak{su}(2)$ algebras. Each of the three subsets (A6)-(A8) can be labelled with a two-element subset of the $\{1, 2, 3\}$ and can be employed to define canonical basis states of SU(n). For instance, consider $(\lambda, \kappa) = (1, 1)$ irrep of SU(3). The weight $(\lambda_2, \lambda_1) = (0, 0)$ is associated with a two-dimensional space. We can identify two basis states of this space by specifying the following:

1. choice of $\mathfrak{su}(2)$ algebra. For instance $I^{(2)} = \{1, 2\}$, which corresponds to the algebra generated by $\{ C_{1,2}, C_{2,1}, C_{1,1} - C_{2,2} \}$,
2. $\mathfrak{su}_{1,2,3}(3)$ irreps label: $K^{(3)} = (1, 1)$, $\mathfrak{su}_{(1,2)}(2)$ irreps label: $K^{(2)} = (0)$ and (1) for the two basis states,
3. $\mathfrak{su}_{1,2,3}(3)$ weights: $(0, 0)$, $\mathfrak{su}_{(1,2)}(2)$ weights: (0).

Another basis set of the $\Lambda = (0, 0)$ space of $\mathfrak{su}(3)$ irrep $K = (1, 1)$ is specified by choosing a different $\mathfrak{su}(2)$ subalgebra as follows.

1. choice of $\mathfrak{su}(2)$ algebra. For instance $I^{(2)} = \{1, 3\}$, which corresponds to the algebra generated by $\{ C_{1,3}, C_{3,1}, C_{1,1} - C_{3,3} \}$,
2. $\mathfrak{su}_{1,2,3}(3)$ irreps label: $K^{(3)} = (1, 1)$, $\mathfrak{su}_{(1,3)}(2)$ irreps label: $K^{(2)} = (0)$ and (1) for the two basis states,
3. $\mathfrak{su}_{1,2,3}(3)$ weights: $(0, 0)$, $\mathfrak{su}_{(1,3)}(2)$ weights: (0).

Thus, different choices of subalgebra chain give us different basis states.

In the main text, we have chosen subalgebra chain (16). Our algorithms can be modified to account for other choices of subalgebra chain by choosing a different set of lowering operators in the basis-set subroutine. Thus, our algorithms can be used to construct states and $\mathcal{D}$-functions in any of the bases that reduce $\mathfrak{su}(m)$ subalgebra chains.

APPENDIX B: CONNECTION TO GELFAND-TSETLIN BASIS

In this appendix, we detail the mapping between our $SU(n)$ basis states and the canonical GT basis. The GT basis identifies each $SU(n)$ irrep with a sequence of n numbers

$$S_n = (m_{1,n}, \ldots, m_{n,n}), \tag{B1}$$

$$m_{k,n} \geq m_{k+1,n} \; \forall 1 \leq k \leq n-1, \tag{B2}$$

where the first label in the subscript is the sequence index and the second label identifies the algebra. The carrier space of every $\mathfrak{su}(m)$ subalgebra is composed of disjoint $\mathfrak{su}(m-1)$ carrier spaces

$$\{(m_{1,n-1}, \ldots, m_{n-1,n-1})\} \tag{B3}$$

that obey the betweenness condition

$$m_{k,n} \geq m_{k,n-1} \geq m_{k+1,n}. \tag{B4}$$

Thus, each $\mathfrak{su}(n)$ basis state $|M\rangle$ can be labelled by the GT pattern

$$|M\rangle \equiv \begin{pmatrix} m_{1,N} & m_{2,N} & \ldots & & m_{N,N} \\ & m_{1,N-1} & \ldots & m_{N-1,N-1} & \\ & & \ddots & \iddots & \\ & & m_{1,2} & m_{2,2} & \\ & & & m_{1,1} & \end{pmatrix}, \tag{B5}$$

where

$$m_{k,\ell} \geq m_{k,n-1} \geq m_{k+1,\ell}, \; 1 \leq k < \ell \leq n. \tag{B6}$$

The canonical basis states are eigenstates of Cartan operators $\{H_i\}$ (12) as detailed in the following lemma.

Lemma 16 (Connection to Gelfand-Tsetlin basis[67]). The canonical basis states are connected to the GT basis according to

$$\left| \psi^{K^{(z)}, \ldots, K^{(3)}, K^{(2)}}_{\Lambda^{(n)}, \ldots, \Lambda^{(3)}, \Lambda^{(2)}} \right\rangle = \begin{pmatrix} m_{1,N} & m_{2,N} & \ldots & & m_{N,N} \\ & m_{1,N-1} & \ldots & m_{N-1,N-1} & \\ & & \ddots & \iddots & \\ & & m_{1,2} & m_{2,2} & \\ & & & m_{1,1} & \end{pmatrix}. \tag{B7}$$

Every state $|M\rangle$ in the GT-labeling scheme is a simultaneous eigenstate of all $\mathfrak{su}(n)$ Cartan operators,

$$H_\ell |M\rangle = \lambda_\ell^M |M\rangle, \quad (1 \leq \ell \leq N-1), \tag{B8}$$

with eigenvalues

$$\lambda_\ell = \sum_{k=1}^{\ell} m_{k,\ell} - \frac{1}{2} \left(\sum_{k=1}^{\ell+1} m_{k,\ell+1} + \sum_{k=1}^{\ell-1} m_{k,\ell-1} \right), \; 1 \leq \ell \leq N-1. \tag{B9}$$

Thus, the canonical basis states of Def. 4 is uniquely mapped to the GT basis.

Furthermore, the weights λ_ℓ are also mapped via the boson realizations to differences in number of bosons at sites ℓ and $\ell+1$. Hence, the difference

$$\nu_{\ell+1} - \nu_\ell = \sum_{k=1}^{l} m_{k,\ell} - \frac{1}{2} \left(\sum_{k=1}^{\ell+1} m_{k,\ell+1} + \sum_{k=1}^{\ell-1} m_{k,\ell-1} \right), \; 1 \leq \ell \leq N-1 \tag{B10}$$

in the number of bosons at sites $\ell + 1$ and ℓ of the boson realization of a basis state is also connected to its GT pattern. Once we recall the total number of bosons in the system is $\nu_1 + \nu_2 + \cdots + \nu_n = N_k$, one can then invert the differences and recover ν_p in term of the $m_{k,\ell-1}$. Thus, the canonical GT basis states are connected to our SU(n) basis states.

[1] D. A. Varšalovič, A. N. Moskalev, and V. K. Kersonskij, *Quantum Theory of Angular Momentum* (World Scientific, Singapore, 1989).
[2] M. Rose, *Elementary Theory of Angular Momentum* (Dover, New York, 1995).
[3] A. R. Edmonds, *Angular Momentum in Quantum Mechanics* (Princeton University Press, Princeton, 1996).
[4] G. Racah, "Theory of complex spectra. III," Phys. Rev. **63**, 367–382 (1943).
[5] M. Jacob and G. Wick, "On the general theory of collisions for particles with spin," Ann. Phys. **7**, 404 (1959).
[6] K. Alder, A. Bohr, T. Huus, B. Mottelson, and A. Winther, "Study of nuclear structure by electromagnetic excitation with accelerated ions," Rev. Mod. Phys. **28**, 432–542 (1956).
[7] M. Moshinsky, "Wigner coefficients for the SU$_3$ group and some applications," Rev. Mod. Phys. **34**, 813–828 (1962).
[8] H. Ui, "Clebsch-Gordan formulas of the SU(1, 1) group," Prog. Theor. Phys. **44**, 689–702 (1970).
[9] B. Yurke, S. McCall, and J. Klauder, "SU(2) and SU(1, 1) interferometers," Phys. Rev. A **33**, 4033–4054 (1986).
[10] P. Joanis, D. H. Mahler, and H. de Guise, "𝔰𝔲(1, 1) intelligent states," J. Phys. A: Math. Gen. **43**, 385304 (2010).
[11] E. Karimi, R. W. Boyd, P. de la Hoz, H. de Guise, J. Řeháček, Z. Hradil, A. Aiello, G. Leuchs, and L. L. Sánchez-Soto, "Radial quantum number of Laguerre-Gauss modes," Phys. Rev. A **89**, 063813 (2014).
[12] J. M. Cerveró and J. D. Lejarreta, "Generalized two-mode harmonic oscillator: Dynamical group and squeezed states," J. Phys. A: Math. Gen. **29**, 7545–7560 (1996).
[13] A. Wünsche, "Symplectic groups in quantum optics," J. Opt. B: Quantum Semiclassical Opt. **2**, 73–80 (2000).
[14] S. Aaronson and A. Arkhipov, "The computational complexity of linear optics," Theor. Comput. **9**, 143–252 (2013).
[15] S. Aaronson and A. Arkhipov, "Bosonsampling is far from uniform," J. Quantum Inf. Comput. **14**, 1383–1423 (2014).
[16] S.-H. Tan, Y. Y. Gao, H. de Guise, and B. C. Sanders, "SU(3) quantum interferometry with single-photon input pulses," Phys. Rev. Lett. **110**, 113603 (2013).
[17] M. Tillmann, S.-H. Tan, S. E. Stoeckl, B. C. Sanders, H. de Guise, R. Heilmann, S. Nolte, A. Szameit, and P. Walther, "Generalized multi-photon quantum interference," Phys. Rev. X **5**(4), 041015 (2015); e-print arXiv:1403.3433 [quant-ph] (2014).
[18] H. de Guise, S.-H. Tan, I. P. Poulin, and B. C. Sanders, "Coincidence landscapes for three-channel linear optical networks," Phys. Rev. A **89**, 063819 (2014).
[19] N. Vilenkine, *Special Functions and the Theory of Group Representations* (American Mathematical Society, Providence, 1968).
[20] W. Miller, *Lie Theory and Special Functions* (Academic Press, New York, 1968).
[21] J. D. Talman, "Special functions: A group theoretic approach," in *Boojums all the Way Through* (Cambridge University Press, 1990), p. 309.
[22] E. Chacón and M. Moshinsky, "Representations of finite U$_3$ transformations," Phys. Lett. **23**, 567–569 (1966).
[23] D. J. Rowe, B. C. Sanders, and H. de Guise, "Representations of the Weyl group and Wigner functions for SU(3)," J. Math. Phys. **40**, 3604 (1999).
[24] J. P. Draayer, "Wigner and Racah coefficients for SU(3)," J. Math. Phys. **14**, 1904 (1973).
[25] D. J. Millener, "A note on recoupling coefficients for SU(3)," J. Math. Phys. **19**, 1513 (1978).
[26] D. J. Rowe and C. Bahri, "Clebsch-Gordan coefficients of SU(3) in SU(2) and SO(3) bases," J. Math. Phys. **41**, 6544 (2000).
[27] M. A. B. Beg and H. Ruegg, "A set of harmonic functions for the group SU(3)," J. Math. Phys. **6**, 677 (1965).
[28] R. Slansky, "Group theory for unified model building," Phys. Rep. **79**, 1–128 (1981).
[29] H. Georgi, *Lie Algebras in Particle Physics* (Westview Press, Boulder, 1999).
[30] D. J. Rowe, *Fundamentals of Nuclear Models: Foundational Models* (World Scientific, Singapore, 2010).
[31] J. F. Cornwell, *Group Theory in Physics: An Introduction* (Academic Press, San Diego, 1997).
[32] R. Gilmore, *Lie Groups, Lie Algebras, and Some of Their Applications*, Dover Books on Mathematics (Dover Publications, Mineola, New York, 2012).
[33] J. M. Jauch and E. L. Hill, "On the problem of degeneracy in quantum mechanics," Phys. Rev. **57**, 641–645 (1940).
[34] J. Schwinger, "On angular momentum," USAEC Report NYO-3071, 1952.
[35] G. A. Baker, "Degeneracy of the n-dimensional, isotropic, harmonic oscillator," Phys. Rev. **103**, 1119–1120 (1956).
[36] J. P. Elliott, "Collective motion in the nuclear shell model. I. Classification schemes for states of mixed configurations," Proc. R. Soc. A **245**, 128–145 (1958).
[37] D. M. Fradkin, "Three-dimensional isotropic harmonic oscillator and SU(3)," Am. J. Phys. **33**, 207 (1965).
[38] F. Iachello, *The Interacting Boson Model* (Cambridge University Press, Cambridge, 1987).
[39] A. Klein and E. Marshalek, "Boson realizations of Lie algebras with applications to nuclear physics," Rev. Mod. Phys. **63**, 375–558 (1991).
[40] A. Kuriyama, J. da Providência, and M. Yamamura, "A possible description of many-body system composed of four kinds of boson operators: In relation to the su(2)- and the su(1, 1)-algebraic model," Prog. Theor. Phys. **103**, 305–319 (2000).
[41] S. D. Bartlett, D. A. Rice, B. C. Sanders, J. Daboul, and H. de Guise, "Unitary transformations for testing Bell inequalities," Phys. Rev. A **63**, 042310 (2001).
[42] R. C. Hwa and J. Nuyts, "Group embedding for the harmonic oscillator," Phys. Rev. **145**, 1188–1195 (1966).
[43] P. Kramer and M. Moshinsky, "Group theory of harmonic oscillators (III). States with permutational symmetry," Nucl. Phys. **82**, 241–274 (1966).
[44] M. Moshinsky and C. Quesne, "Linear canonical transformations and their unitary representations," J. Math. Phys. **12**, 1772–1780 (1971).

[45] C. Quesne and M. Moshinsky, "Canonical transformations and matrix elements," J. Math. Phys. **12**, 1780–1783 (1971).

[46] D. J. Rowe, "Coherent state theory of the noncompact symplectic group," J. Math. Phys. **25**, 2662 (1984).

[47] H. Weyl, *The Theory of Groups and Quantum Mechanics* (Dover, New York, 1950).

[48] S. Chaturvedi, G. Marmo, N. Mukunda, R. Simon, and A. Zampini, "The Schwinger representation of a group: Concept and applications," Rev. Math. Phys. **18**, 887–912 (2006).

[49] D. Littlewood, *The Theory of Group Characters and Matrix Representations of Groups*, AMS Chelsea Publishing Series (Clarendon Press, Gloucestershire, 1950).

[50] The centre of an algebra $\mathfrak{u}$ comprises those elements x of $\mathfrak{u}$ such that $xu = ux$ for all $u \in \mathfrak{u}$.

[51] B. Kostant, "A formula for the multiplicity of a weight," Trans. Am. Math. Soc. **93**, 53 (1959).

[52] I. Gelfand and M. Tsetlin, *Finite-Dimensional Representations of the Group of Unimodular Matrices* (Springer-Verlag, 1988), pp. 653–656 (in English); Dokl. Akad. Nauk SSSR (N.S.) **71**, 825–828 (1950) (in Russian).

[53] A. O. Barut and R. Raczka, "The explicit construction of finite-dimensional irreducible representations," in *Theory of Group Representations and Applications* (World Scientific, 1986), Vol. 2.

[54] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator," Phys. Rev. Lett. **73**, 58–61 (1994).

[55] A useful computational shortcut involves the map

$$a_k^\dagger \to x_k, \quad a_\ell \to \frac{\partial}{\partial x_\ell}, \quad k, \ell \in \{1, 2\},$$

which preserves the boson commutation relations. The above map in this reference transforms the vector $|J, M\rangle$ (24) into a formal polynomial and the corresponding dual vector $\langle J, M|$ into a linear differential operator in the dummy variables x_1, x_2. $\mathcal{D}$-function (28) is thus evaluated as the action of a linear differential operator on a polynomial in x_1, x_2.

[56] M. Moshinsky, "The harmonic oscillator and supermultiplet theory," Nucl. Phys. **31**, 384–405 (1962).

[57] In Algorithms 1-3, we denote operations in capital case and SMALLCAPS font. Variables are denoted by roman font and are in lower case.

[58] E. F. Moore, "The shortest path through a maze," in *Proceedings of the International Symposium on the Theory of Switching* (Harvard University Press, 1961), pp. 285–292.

[59] C. Y. Lee, "An algorithm for path connections and its applications," IEEE Trans. Electron. Comput. **EC-10**, 346–365 (1961).

[60] D. E. Knuth, *The Art of Computer Programming: Sorting and Searching* (Pearson Education, New York, 1998), Vol. 3.

[61] A queue[60] is a first-in-first-out data structure whose entries are maintained in order. The two operations allowed on a queue are enqueue, i.e., the addition of entries to the rear and dequeue, which is the removal of entries from the front of the queue. Both the enqueue and dequeue operations require constant, i.e., O(1) time.

[62] J. E. Humphreys, "Weight and maximal vectors," in *Introduction to Lie Algebras and Representation Theory* (Springer, New York, 1972), Vol. 9.

[63] N. N. Bogolyubov and D. V. Shirkov, *Introduction to the Theory of Quantized Fields*, Interscience Monographs in Physics and Astronomy Vol. 3 (John Wiley, New York, 1959).

[64] S. C. Pang, "Lowering and raising operators for the orthogonal group in the chain O(n) ⊃ O(n − 1) ⊃ · · · , and their graphs," J. Math. Phys. **8**, 1233 (1967).

[65] M. K. F. Wong, "Representations of the orthogonal group. II. Polynomial bases for the irreducible representations of the orthogonal group," J. Math. Phys. **10**, 1065 (1969).

[66] M. A. Lohe, "The boson calculus for the orthogonal and symplectic groups," J. Math. Phys. **12**, 1882 (1971).

[67] A. Alex, M. Kalus, A. Huckleberry, and J. von Delft, "A numerical algorithm for the explicit calculation of SU(N) and SL(N,C) Clebsch-Gordan coefficients," J. Math. Phys. **52**, 023507 (2011).