Quantum Information and Computation, Vol. 0, No. 0 (2003) 000–000
© Rinton Press

A new algorithm for producing quantum circuits using KAK decompositions

Yumi Nakajima,[a] Yasuhito Kawano, and Hiroshi Sekigawa
NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation
3-1 Morinosato-Wakamiya, Atsugi, Kanagawa 243-0198 , JAPAN

Received (received date)
Revised (revised date)

We provide a new algorithm that translates a unitary matrix into a quantum circuit according to the $G = KAK$ theorem in Lie group theory. With our algorithm, any matrix decomposition corresponding to type-**AIII** KAK decompositions can be derived according to the given Cartan involution. Our algorithm contains, as its special cases, Cosine-Sine decomposition (CSD) and Khaneja-Glaser decomposition (KGD) in the sense that it derives the same quantum circuits as the ones obtained by them if we select suitable Cartan involutions and square root matrices. The selections of Cartan involutions for computing CSD and KGD will be shown explicitly. As an example, we show explicitly that our method can automatically reproduce the well-known efficient quantum circuit for the n-qubit quantum Fourier transform.

Keywords: $G = KAK$ theorem, KGD, CSD, decomposition of the QFT

Communicated by: to be filled by the Editorial

1 Introduction

Decomposing a unitary matrix into an efficient sequence of elementary gates is a fundamental problem in designing quantum circuits. There are two types of decomposition: One is exact decomposition where an arbitrary unitary matrix is decomposed precisely into a sequence of elementary gates, such as arbitrary single-qubit rotations and the CNOT. The other involves approximate strategies by which an arbitrary unitary matrix is decomposed approximately into a sequence of a fixed set of elementary gates, as shown in Solovay-Kitaev theorem (cf. [1], Appendix 3). In this paper, we treat exact decomposition.

Cosine-Sine decomposition (CSD), which is a well-known algorithm in numerical linear algebra, was the first algorithm utilized for this purpose [2]. CSD applies the well-known algorithm for computing generalized singular value decomposition (GSVD). In CSD, we first divide an input matrix g into four square matrices and then apply SVD to each matrix. Then, we have

$$g = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} = \begin{pmatrix} u_1 & 0 \\ 0 & u_2 \end{pmatrix} \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} v_1 & 0 \\ 0 & v_2 \end{pmatrix} = U\Sigma V, \tag{1}$$

where $c = diag\,[\cos(\zeta_1), \cos(\zeta_2), \cdots, \cos(\zeta_{2^{n-1}})]$, $s = diag\,[\sin(\zeta_1), \sin(\zeta_2), \cdots, \sin(\zeta_{2^{n-1}})]$, then $g_{11} = u_1 c v_1$, $g_{12} = -u_1 s v_2$, $g_{21} = u_2 s v_1$, and $g_{22} = u_2 c v_2$ are SVD. Recursively per-

[a] nakajima@theory.brl.ntt.co.jp

forming decomposition (1) leads to a quantum circuit. The CSD-based algorithms are easy to implement on a computer because algorithms for calculating GSVD are well-known, and software libraries including GSVD are available. Some CSD-based algorithms [3–6] have been investigated with the aim of improving Barenco's result that an arbitrary $2^n \times 2^n$ unitary matrix is composed of $O(n^2 4^n)$ elementary gates [7]. And improvement to $O(4^n)$ elementary gates has been reported by Möttönen *et al.* [4] and by Shende *et al* [6].

On the other hand, Khaneja and Glaser provided another kind of decomposition [8], which was later named KGD. KGD lies within the framework of the $G = KAK$ theorem (cf. [9], Theorem 8.6) in Lie group theory. This theorem shows that an element $g \in SU(2^n)$ is decomposed into matrix products $k_1 a k_2$ for some $k_1, k_2 \in \exp(\mathfrak{k})$ and $a \in \exp(\mathfrak{h})$. Here, $\mathfrak{su}(2^n) = \mathfrak{k} \oplus \mathfrak{m}$ is a Cartan decomposition in Lie algebra $\mathfrak{su}(2^n)$, $\mathfrak{k}$ and $\mathfrak{m} = \mathfrak{k}^\perp$ are orthogonal vector spaces contained in $\mathfrak{su}(2^n)$, and $\mathfrak{h}$ is a maximal Abelian subalgebra (a Cartan subalgebra) contained in $\mathfrak{m}$ (cf. [15], §VI.2). Matrices k_1, a, and k_2 are not uniquely determined from g. They depend on the selections of the bases of $\mathfrak{k}$, $\mathfrak{m}$, and $\mathfrak{h}$; besides, they are not determined even if bases are selected. Khaneja and Glaser provided a particular selection of bases of $\mathfrak{k}$, $\mathfrak{m}$, and $\mathfrak{h}$ in Ref. [8] so that the selection matches an NMR system, and they proved that a time-optimal control on a two-qubit NMR quantum computer can be obtained from the decomposition [10]. Thus, KGD can be regarded as the $G = KAK$ theorem on the particular bases. It should be noted that KGD does not give a unique translation of the input matrix into a quantum circuit.

Bullock [11] showed that CSD can also be regarded in the framework of the $G = KAK$ theorem; i.e., CSD uses the type-**AIII** KAK decomposition with the global Cartan decomposition Θ defined as $\Theta(X) = \sigma_{1z} X \sigma_{1z}$ for $X \in SU(2^n)$, where σ_{jz} denotes that the operation defined as the Pauli matrix σ_z acts on the j-th qubit. He also introduced a method that translates matrices U, Σ, and V in (1) into $k_1 \in \exp(\mathfrak{k})$, $a \in \exp(\mathfrak{h})$, and $k_2 \in \exp(\mathfrak{k})$, respectively, where $\mathfrak{k}$ and $\mathfrak{h}$ are the ones defined in KGD. Here, KGD corresponds to $G = KAK$ decomposition with the selection of Θ defined as $\Theta(X) = \sigma_{nz} X \sigma_{nz}$. We can thus produce a KGD-based quantum circuit by combining Bullock's translation and the CSD-based algorithms.

We introduce a new algorithm that translates a $2^n \times 2^n$ unitary matrix into a quantum circuit according to the $G = KAK$ theorem. The algorithm can derive any matrix decomposition corresponding to type-**AIII** KAK decompositions for the given global Cartan involution Θ. The algorithm contains, as its special cases, both CSD and KGD in the sense that it derives the same quantum circuits as the ones calculated by them if we select suitable Cartan involutions and square root matrices. Here, we select $\Theta(X)$ as $\sigma_{1z} X \sigma_{1z}$ for CSD and as $\sigma_{nz} X \sigma_{nz}$ for KGD, where $X \in SU(2^n)$. The strategy utilized in our algorithm is related to those used in Refs. [12, 13]. However, those strategies provided methods for computing type-**AII** KAK decomposition; no translation between type-**AII** decompositions and type-**AIII** decompositions was provided. Furthermore, the method utilized in Ref. [12] is different from ours in the square root matrix calculations, i.e., methods for calculating m from m^2 (where $g = km$ is a global Cartan decomposition of the input matrix g). In the method proposed in Ref. [12], first, a square root matrix is calculated in Lie algebra level. And then it is translated into an element in Lie group level via exponential mapping. In contrast, with our method, a square root matrix is calculated directly at the Lie group level.

Although our algorithm contains CSD and derives any matrix decomposition corresponding to type-**AIII** KAK decompositions according to the given Cartan involution, the efficiency for calculating a decomposition is not sacrificed. The reason is as follows: Roughly speaking, to decompose g into $k1ak2$, the CSD-based algorithms apply SVD to four $2^{n-1} \times 2^{n-1}$ matrices (g_{11}, g_{12}, g_{21}, and g_{22} in Eq.(1)), while our algorithm applies eigenvalue decomposition to $2^n \times 2^n$ matrix. Therefore, the efficiencies for computing SVD on four $2^{n-1} \times 2^{n-1}$ matrices and for computing eigenvalue decomposition on one $2^n \times 2^n$ matrix are the same.

In addition, our algorithm might have an advantage over CSD when we would like to determine a class of quantum circuits for a given class of matrices. The reason is as follows: In CSD-based algorithms, it is difficult to formulate a class of matrices u_1, u_2, v_1, and v_2 such that relation (1) holds for a given class of input matrices. Actually, to reproduce the well-known QFT circuit by using CSD [2, 14], Tucci changes the rows and columns of the QFT matrices beforehand and makes each submatrix hold a convenient form, which can be written by the $(n-1)$-qubit QFT. It would not be possible to describe the general form of the decomposition when the input matrix does not have a convenient form like QFT. On the other hand, our algorithm does not require such a preliminary change of rows and columns. All matrices appearing through our algorithm can be described using the input matrix g, the given global Cartan involution Θ, and the eigenvalues and eigenvectors of these matrix products. This will be shown explicitly as an example of the QFT decomposition (Section 4).

The paper is organized as follows: In the following section, we cover some preliminaries about notations, the $G = KAK$ theorem, and KGD. Section 3 presents our algorithm for computing the decomposition follows from the $G = KAK$ theorem. Section 4 presents decompositions of the n-qubit QFT using our algorithm and CSD-based algorithms. We show that we can produce the well-known QFT circuit by using these matrix decompositions.

2 Preliminaries

2.1 Notations

Let σ_x, σ_y, and σ_z denote the Pauli matrices and $I^{\otimes s}$ be a $2^s \times 2^s$ identity matrix ($I = 2^1 \times 2^1$). We use $\sigma_{j\alpha}$ to denote the Pauli matrix acting on the j-th qubit; $\sigma_{j\alpha} = I^{\otimes(j-1)} \otimes \sigma_\alpha \otimes I^{\otimes(n-j)}$, ($\alpha = x, y$, or z). Let U_{CNOT} denote the standard CNOT gate, H denote a Hadamard gate, and $R_x(\zeta) = \exp(-i\zeta\sigma_x)$. All these notations follow those in Ref. [1].

2.2 $G = KAK$ theorem

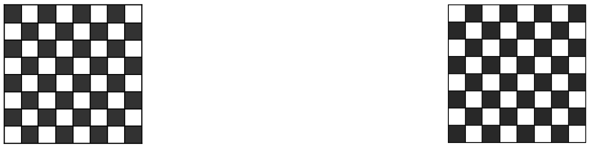
The $G = KAK$ theorem for compact groups (cf. [9], Theorem 8.6) provides a framework for decomposing $g \in SU(2^n)$ into the following matrix products:

$$g \quad = \quad k_1 a k_2, \qquad k_1, k_2 \in \exp(\mathfrak{k}), a \in \exp(\mathfrak{h}) \subset \exp(\mathfrak{m}). \tag{2}$$

Here, $\mathfrak{su}(2^n) = \mathfrak{k} \oplus \mathfrak{m}$ is a Cartan decomposition, where $\mathfrak{k}$ and $\mathfrak{m} = \mathfrak{k}^\perp$ are orthogonal vector spaces, and $\mathfrak{h}$ is a Cartan subalgebra, that is, a maximal Abelian subalgebra contained in $\mathfrak{m}$.

Let θ denote the Cartan involution of its Lie algebra $\mathfrak{su}(2^n)$; i.e., (i) $\theta^2 = I^{\otimes n}$ ($\theta \neq I^{\otimes n}$) and (ii) θ is an automorphism of the Lie algebra $\mathfrak{su}(2^n)$. And let the global Cartan involution (cf. [15], p. 362) of $SU(2^n)$ be Θ. Then $\mathfrak{k}$ and $\mathfrak{m}$ have the following property:

$$\theta(x) = \begin{cases} x & \text{if } x \in \mathfrak{k} \\ -x & \text{if } x \in \mathfrak{m} \end{cases}, \qquad \Theta(X) = \begin{cases} X & \text{if } X \in \exp(\mathfrak{k}) \\ X^\dagger & \text{if } X \in \exp(\mathfrak{m}) \end{cases}. \tag{3}$$



$\mathfrak{k}$ and $\exp(\mathfrak{k})$ $\mathfrak{m}$

Fig. 1. Patterns of an element of $\mathfrak{k}$, $\exp(\mathfrak{k})$ and $\mathfrak{m}$ for a three-qubit system, where each square represents an element of an 8×8 matrix. The white elements are always zero, and the black elements take some value that depends on the input matrix.

Three types of $\mathfrak{k}$-algebra, named **AI**, **AII**, and **AIII**, arise for $\mathfrak{su}(2^n)$. Here, **AI**, **AII**, and **AIII** correspond to $\mathfrak{k} = \mathfrak{so}(2^n)$, $\mathfrak{k} = \mathfrak{sp}(2^n)$, and $\mathfrak{s}[\mathfrak{u}(p) \oplus \mathfrak{u}(q)]$ $(p+q = 2^n)$, respectively (cf. [9], p. 518).

2.3 Khaneja-Glaser decomposition (KGD)

Khaneja and Glaser provided a particular selection of bases of $\mathfrak{k}$, $\mathfrak{m}$, and $\mathfrak{h}$ (cf. [8], Notation 3 and 5) so that the selection matches an NMR system. Notice that we use $\mathfrak{h}$ instead of $\mathfrak{h}(n)$. Here, generators of $\mathfrak{k}$, $\mathfrak{m}$, and $\mathfrak{h}$, are denoted to be tensor products of the Pauli matrices;

$$\mathfrak{k} = \operatorname{span}\{A \otimes \sigma_z/2, B \otimes I, i\sigma_{nz}/2 \mid A, B \in \mathfrak{su}(2^{n-1})\}, \tag{4}$$

$$\mathfrak{m} = \operatorname{span}\{A \otimes \sigma_x/2, B \otimes \sigma_y/2, i\sigma_{nx}/2, i\sigma_{ny}/2 \mid A, B \in \mathfrak{su}(2^{n-1})\}. \tag{5}$$

Here, generators of $\mathfrak{k}$ and $\mathfrak{m}$ have a specific operation on the last qubit; i.e., σ_z or I for generators of $\mathfrak{k}$ and σ_x or σ_y for generators of $\mathfrak{m}$. Thus, to determine $\mathfrak{k}$ and $\mathfrak{m}$, the Cartan involution θ and the global Cartan involution Θ can be chosen as follows:

$$\theta(x) = \sigma_{nz} x \sigma_{nz}, \qquad \Theta(X) = \sigma_{nz} X \sigma_{nz}. \tag{6}$$

Since $\theta(\sigma_z) = \sigma_z$, $\theta(I) = I$, $\theta(\sigma_x) = -\sigma_x$, $\theta(\sigma_y) = -\sigma_y$, we can check that the above $\mathfrak{k}$ and $\mathfrak{m}$ satisfy relation (3) when θ and Θ are chosen as (6).

For the number of qubits $n \geq 3$, $\mathfrak{k}$ and $\mathfrak{m}$ have specific patterns, as shown in Fig. 1, because all generators defined in (4) and (5) have these patterns. Note that in contrast to an element of $\exp(\mathfrak{k})$ taking the same pattern as an element of $\mathfrak{k}$, an element of $\exp(\mathfrak{m})$ does not take the same pattern as $\mathfrak{m}$. This property enables us to apply the KAK decomposition recursively, as shown in Fig. 2.

3 Our algorithm

3.1 Basic ideas

We provide a new constructive algorithm that computes a decomposition based on the $G = KAK$ theorem. Here, we choose a Cartan subalgebra $\widetilde{\mathfrak{h}}$, which is different from the $\mathfrak{h}$ used in KGD. Since Cartan subalgebras are Abelian, they can translate each other by $\widetilde{\mathfrak{h}} = Ad_{T \in \exp(\mathfrak{k})}(\mathfrak{h})$. Here, T is fixed for given $\widetilde{\mathfrak{h}}$ and $\mathfrak{h}$. It should be noted that $G = KAK$ decomposition for the fixed input $g \in SU(2^n)$ is not unique. We provide the following theorem.

Theorem 1 *Let $g \in SU(2^n)$ be the input matrix. If g has a global Cartan decomposition $g = km$ ($k \in \exp(\mathfrak{k}), m \in \exp(\mathfrak{m})$), then m^2 is uniquely determined by $m^2 = \Theta(g^\dagger)g$.*

Proof. From (3), $\Theta(g^\dagger)g = \Theta(m^\dagger k^\dagger)km = \Theta(m^\dagger)\,\Theta(k^\dagger)km = mk^\dagger km = m^2$. $\square$

Theorem 1 shows that the fixed global Cartan involution Θ only determines m^2. Therefore, arbitrariness remains in the selection of m, and also k. Furthermore, a in (2) has also arbitrariness because it follows from a decomposition of m, $m = \widetilde{k}^\dagger a \widetilde{k}$, where $\widetilde{k} \in \exp(\mathfrak{k})$. (See, [15], §AII.3). Using Theorem 1, we compute the decomposition in (2) as follows:

1. Compute $m^2 = \Theta(g^\dagger)g$.
2. Decompose $m^2 = pbp^\dagger$ such that $p \in \exp(\mathfrak{k})$ and $b \in \exp(\widetilde{\mathfrak{h}})$.
   Such decomposition always exists because $m^2 \in \exp(\mathfrak{m})$ (cf. [15], Proposition 7.29). We compute p using eigenvectors of m^2. We show examples in Section 3.2.
3. Find y such that $y^2 = b$ and $y \in \exp(\widetilde{\mathfrak{h}})$.
   The y can be computed by replacing the diagonal blocks of b, when we choose a suitable $\widetilde{\mathfrak{h}}$. We show examples of selections of $\widetilde{\mathfrak{h}}$ in Section 3.2.
4. Compute $m = pyp^\dagger$. Here, $m \in \exp(\mathfrak{m})$ because $\Theta(m) = \Theta(p)\Theta(y)\Theta(p^\dagger) = py^\dagger p^\dagger = m^\dagger$.
5. Compute $k = gm^\dagger$. Then, k always satisfies $k \in \exp(\mathfrak{k})$ because $(m^2)^\dagger = g^\dagger\Theta(g)$ and $\Theta(gm^\dagger) = \Theta(g)\Theta(m^\dagger) = g(m^2)^\dagger m = gm^\dagger$.

Steps 2–4 provide a method for computing the square root of a matrix to find m from m^2. After these procedures, we obtain

$$g \;=\; kpyp^\dagger \;=\; \widetilde{k}yp^\dagger. \tag{7}$$

Here, $\Theta(\widetilde{k}) = \widetilde{k}$, $\Theta(y) = y^\dagger$, and $\Theta(p^\dagger) = p^\dagger$, so that the decomposition follows the $G = KAK$ theorem. All matrices that appear through the algorithm can be described using g and Θ. This enables us to show that algorithm can automatically reproduce the well-known QFT circuit (See, Section 4.2).

3.2 Methods for performing steps 2 and 3 based on KGD

Above, we provided a method for computing $G = KAK$ decomposition. We did not provide a concrete method for computing p, b, and y in steps 2 and 3 in our algorithm. To show examples of such concrete methods, we fixed Θ and $\widetilde{\mathfrak{h}}$. Here, we treat examples that compute KGD. We choose Θ as in (6) and show two methods that compute p, b, y for particular selections of $\widetilde{\mathfrak{h}}$; i.e., $\widetilde{\mathfrak{h}_1} = \mathrm{span}\,\{|j\rangle\langle j| \otimes i\sigma_x \,|\, j = 0, \cdots, 2^{n-1}-1\}$ and $\widetilde{\mathfrak{h}}_2 = \mathrm{span}\{|j\rangle\langle j| \otimes i(\sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y) \,|\, j = 0, \cdots, 2^{n-2}-1\}$. We should note that the second selection is not always possible because it demands that all eigenvalues appearing in m^2 in step 1 should be duplicated twice.

3.2.1 *Example 1*

Let Θ be as in (6), and $\widetilde{\mathfrak{h}_1}$ as span $\{|j\rangle\langle j| \otimes i\sigma_x \mid j = 0, \cdots, 2^{n-1}-1\}$. Then, we compute p, b, and y in steps 2 and 3 as follows:

(i) Compute eigenvalue decomposition of m^2.
Let $m^2 = \widetilde{p}d\widetilde{p}^\dagger$ be eigenvalue decomposition and $\mu_1, \cdots, \mu_N$ be the columns of $\widetilde{p}$, where $N = 2^n$. Then, d is a diagonal matrix in which diagonals have eigenvalues of m^2 and all μ_j's are eigenvectors of m^2 and mutually orthogonal.

(ii) Normalize all μ_j's by

$$\nu_{2j-1} = \frac{\mu_j + \sigma_{nz}\mu_j}{\|\mu_j + \sigma_{nz}\mu_j\|}, \qquad \nu_{2j} = \frac{\mu_j - \sigma_{nz}\mu_j}{\|\mu_j - \sigma_{nz}\mu_j\|}. \tag{8}$$

(iii) For all ν_j's that are associated with imaginary eigenvalues,

(a) let W_1, W_2, and W_3 be sets of vectors such that

$$W_1 = \{\nu_j \mid \sigma_{nz}\nu_j = \nu_j\}, \quad W_2 = \{\nu_j \mid \sigma_{nz}\nu_j = -\nu_j\}, \quad W_3 = \{\nu_j \mid \sigma_{nz}\nu_j \neq \pm\nu_j\}.$$

(b) For each $w \in W_3$, compute

$$\nu^+ = \frac{w + \sigma_{nz}w}{\|w + \sigma_{nz}w\|}, \qquad \nu^- = \frac{w - \sigma_{nz}w}{\|w - \sigma_{nz}w\|}. \tag{9}$$

Here, $\|\cdot\|$ denotes the length of a vector. Then,

- if all elements in W_1 and u^+ are linearly independent, then $W_1 = W_1 \cup \{\nu^+\}$;
- if all elements in W_2 and u^- are linearly independent, then $W_2 = W_2 \cup \{\nu^-\}$.

(iv) Repeat steps (a) and (b) for all μ_j's that are associated with positive real eigenvalues.

(v) Repeat steps (a) and (b) for all μ_j's that are associated with negative real eigenvalues.

(vi) Let $p = (\upsilon_1, \upsilon_2, \cdots, \upsilon_N)$, where $\upsilon_{2j-1} \in W_1$ and $\upsilon_{2j} \in W_2$, for $j = 1, \cdots, N/2$.
The computation procedure follows from Appendix A. Since $\sigma_{nz}\upsilon_{2j-1} = \upsilon_{2j-1}$ and $\sigma_{nz}\upsilon_{2j} = -\upsilon_{2j}$, we can easily check that $\Theta(p) = p$. Then, $b = p^\dagger m^2 p$ satisfy $\Theta(b) = b^\dagger$ and $b \in \exp(\widetilde{\mathfrak{h}})$, where $b = \sum_{j=0}^{2^{n-1}} |j\rangle\langle j| \otimes R_x(2\zeta_j)$ $(0 \leq \zeta_j < \pi)$.

(vii) Compute y by replacing all $R_x(2\zeta_j)$ appears in b with $R_x(\zeta_j)$.
Since $\Theta(R_x(2\zeta_j)) = R_x^\dagger(2\zeta_j)$, $R_x^2(\zeta_j) = R_x(2\zeta_j)$, then y satisfies $y^2 = b$ and $y \in \exp(\widetilde{\mathfrak{h}_1})$.

In step (vi), one may notice that, when $R_x(\pi)$ appears in b, then we can use $R_y(\pi/2)$ instead of $R_x(\pi/2)$ as a replacement rule.

Since $\widetilde{k}$ and $p^\dagger \in$ are elements of $\exp(\mathfrak{k})$ that has the specific pattern as shown in Fig. 1, they have the following decomposition:

$$\widetilde{k} = g_1^{(0)} \otimes |0\rangle\langle 0| + g_1^{(1)} \otimes |1\rangle\langle 1|, \qquad p^\dagger = g_2^{(0)} \otimes |0\rangle\langle 0| + g_2^{(1)} \otimes |1\rangle\langle 1|, \tag{10}$$

where $g_1^{(j)}, g_2^{(j)} \in SU(2^{n-1})$ for $\widetilde{k}, p^\dagger \in SU(2^n)$ ($j = 0$ or 1). Here, $g_1^{(0)}$ and $g_2^{(0)}$ are composed of nonzero elements (black squares in Fig. 1) of odd rows, and $g_1^{(1)}$ and $g_2^{(1)}$ are composed of nonzero elements of even rows.

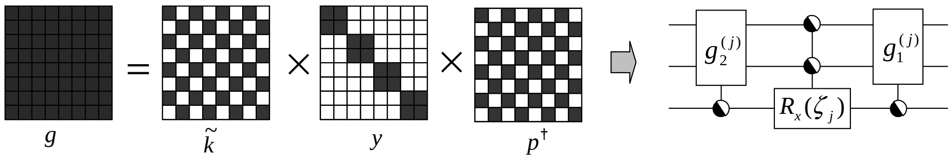


Fig. 2. Image of a decomposition when we choose Θ as in (6) and a Cartan subalgebra $\mathfrak{h}_1$ as span $\{|j\rangle\langle j| \otimes i\sigma_x \mid j = 0, \cdots, 2^{n-1} - 1\}$:
The matrices uses the same notation as in Fig. 1 to represent the properties. In the quantum circuit, the symbol of the control qubit represents the uniformly controlled rotations [5,16]. $g_\ell^{(j-1)} \in SU(4)$ $(\ell, j = 1 \text{ or } 2)$ are applied selectively; that is, $g_1^{(0)}$ and $g_2^{(0)}$ are applied when the third qubit is $|0\rangle$, whereas $g_1^{(1)}$ and $g_2^{(1)}$ are applied when it is $|1\rangle$.

Fig. 2 shows the image of a decomposition in (2) for a three-qubit system; that is, we choose Θ as in (6) and a Cartan subalgebra as $\mathfrak{h}_1$. Applying the decomposition in (7) recursively to elements $g_1^{(j)}, g_2^{(j)} \in SU(2^{n-1})$ (j=0 or 1), we obtain a sequence of uniformly controlled rotations like in Fig. 13 in Ref. [5], except that R_x is used instead of R_y in our case. The full decomposition of these uniformly controlled rotations into elementary gates has been provided by Möttönen *et al.* [5, 16]. Also, if we change the order of qubits and apply the quantum Multiplexor decomposition to $\widetilde{k}$ and $p^\dagger$ in (2), the produced circuit is the same as that in Fig. 2 in Ref. [6], except that rotation R_y is used instead of R_x. Therefore, the number of elementary gates needed to compose $g \in SU(2^n)$ in our method is $O(4^n)$, which is the same as in Refs. [4, 6].

3.2.2 Example 2

Here, we show another example of methods for computing p, b, and y in Section 3.1. Here, we choose $\widetilde{\mathfrak{h}}_2 = \text{span}\,\{|j\rangle\langle j| \otimes i(\sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y) \mid j = 0, \cdots, 2^{n-1} - 1\}$. The Cartan involution Θ is the same as in (6). The decomposition of this type is chosen as an example of a decomposition of the QFT. In this case, we only change steps (vi) and (vii) in Section 3.2.1 as follows:

(vi) Let $p = (\upsilon_1, \upsilon_2, \cdots, \upsilon_N)$, for $j = 1, \cdots, N/4$,

- $\nu_{4j-3} \in W_1$ and it is associated with positive eigenvalues,
- $\nu_{4j-2} \in W_2$ and it is associated with negative eigenvalues,
- $\nu_{4j-1} \in W_1$ and it is associated with negative eigenvalues,
- $\nu_{4j} \in W_2$ and it is associated with positive eigenvalues.

Then, p also satisfies $\Theta(p) = p$, and $b = p^\dagger m^2 p$ is a block-diagonal matrix, in which the diagonals are constructed from the 4×4 matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\zeta_j) & i\sin(2\zeta_j) & 0 \\ 0 & i\sin(2\zeta_j) & \cos(2\zeta_j) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \exp(i\zeta_j(\sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y)).$$

Here, the middle part of the above matrix is $R_x(2\zeta_j)$.

(vii) Compute y by replacing $R_x(2\zeta_j)$ in b with $R_x(\zeta_j)$.

4 Decompositions of the QFT using $G = KAK$ decomposition

In this section, we show that we can automatically reproduce the well known QFT circuit using our method. All the matrices that appear through our algorithm can be described using the input matrix g and Θ. In contrast, it is difficult to describe all the matrices that appear through the CSD-based algorithm because, as shown in (1), the input matrix g has to be divided into four square matrices and SVD has to be applied to each partitioned matrix. Furthermore, we have to choose a suitable decomposition for each partitioned matrix g_{11}, g_{12}, g_{21}, and g_{22} such that Eq. (1) holds, which makes it difficult to formulate U, Σ, and V in (1).

Fortunately, the n-qubit QFT is a very special matrix that has the following property: If we permute the order of qubits, then each partitioned matrix can be described using $(n-1)$-qubit QFT. Using the feature, we provide a decomposition of the QFT by CSD-based algorithm. This is shown in Section 4.3.

4.1 Notation

The QFT on n qubits, F_n, is a $2^n \times 2^n$ matrix such that

$$F_n \quad = \quad \left(\frac{1}{\sqrt{2^n}} \omega_n^{(j-1)(\ell-1)} \right)_{j\ell}, \qquad \text{where } \omega_n = \exp\left(\frac{2\pi i}{2^n}\right). \tag{11}$$

We define Q_n as a $2^n \times 2^n$ permutation matrix: $Q_n = \chi^n_{n-1} \cdots \chi^n_2 \chi^n_1$, where χ^k_j is the SWAP gate applied to the j-th and the k-th qubits. Let $H_1 = H \otimes I^{\otimes(n-1)}$, then (11) is written as

$$F_n \quad = \quad \frac{1}{\sqrt{2}} \begin{pmatrix} F_{n-1} & \Omega_{n-1} F_{n-1} \\ F_{n-1} & -\Omega_{n-1} F_{n-1} \end{pmatrix} Q_n \quad = \quad H_1 D_n (I \otimes F_{n-1}) Q_n, \tag{12}$$

where

$$D_n = \begin{pmatrix} I^{\otimes(n-1)} & 0 \\ 0 & \Omega_{n-1} \end{pmatrix}, \qquad \Omega_{n-1} = diag\,(1, \omega_n, \cdots, \omega_n^{2^{n-1}-1}).$$

This notation follows from Section 4.6.4 in Ref. [17].

4.2 Decomposition of the QFT by our method

Following Section 3.1, we compute a decomposition as follows:

1. Compute $m^2 = \Theta(F_n^\dagger) F_n$.
   Let S be $(I \otimes F_{n-1}) Q_n$. Since $\Theta(H_1) = H_1$, $\Theta(D_n) = D_n$, and $\sigma_{1z} S = S \sigma_{nz}$, we have $m^2 = S^\dagger \sigma_{1z} \sigma_{nz} S$. All column vectors of $S^\dagger$ are then eigenvectors of m^2 because $\sigma_{1z}\sigma_{nz}$ is a diagonal matrix in which diagonal elements are eigenvalues of m^2.

2. Decompose $m^2 = pbp^\dagger$ such that $\Theta(p) = p$ and $\Theta(b) = b^\dagger$.
   We define $p = S^\dagger Q_n$ (This selection was done so that p satisfies $\Theta(p) = p$ and follows Section 3.2.2), then $b = p^\dagger m^2 p = Q_n^\dagger \sigma_{1z} \sigma_{nz} Q_n = I^{\otimes(n-2)} \otimes diag(1, -1, -1, 1)$.

3. Choose y such that $\Theta(y) = y^\dagger$. Following the step (vi) in Section 3.2.2, we have

$$y \;=\; I^{\otimes(n-2)} \otimes \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \;=\; I^{\otimes(n-2)} \otimes \exp(\pi(\sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y)/4).$$

This is obtained by replacing each 4×4 diagonal block of v, $b = diag(1,\,-1,\,-1,\,1)$, with $y = \exp(\pi(\sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y)/4)$. We can easily check that $\Theta(y) = y^\dagger$ and $y^2 = b$, because $\Theta(y) = y^\dagger$ and $y^2 = b$.

4. Compute $m \;=\; pyp^\dagger$.

5. Compute $k \;=\; gm^\dagger \;=\; F_n m^\dagger$.

Then, $\widetilde{k} \;=\; kp \;=\; (F_n p y^\dagger p^\dagger) p = F_n p y^\dagger$, so we have the following decomposition:

$$F_n \;=\; \widetilde{k} y p^\dagger \;=\; (H_1 D_n Q_n y^\dagger) y (Q_n^\dagger S) \;=\; H_1 D_n S \;= H_1 D_n (I \otimes F_{n-1}) Q_n. \tag{13}$$

We apply a similar decomposition to F_j, for $j \;=\; n-1, n-2, \cdots, 2$. Next, we show a decomposition of D_n. D_n is controlled-Ω_j (where $j \;=\; n-1, n-2, \cdots, 2$), so it suffices to consider the decomposition of Ω_j. Since $\Omega_j \in \exp(\mathfrak{k})$ (it follows from $\Theta(\Omega_j) = \Omega_j$), we apply the decomposition in (10) to Ω_j. Consider Ω_3 as an example, then we have $\Omega_3 = g_1^{(0)} \otimes |0\rangle\langle 0| + g_1^{(1)} \otimes |1\rangle\langle 1|$, where $g_1^{(0)} = diag\,(1, \omega^2, \omega^4, \omega^6)$ and $g_1^{(1)} = diag\,(\omega, \omega^3, \omega^5, \omega^7) = \omega \cdot diag\,(1, \omega^2, \omega^4, \omega^6)$. Then, we have $\Omega_3 = diag\,(1, \omega^2, \omega^4, \omega^6) \otimes diag\,(1, \omega)$. Similarly, since $diag(1, \omega^2, \omega^4, \omega^6)$ is also an element of $\exp(\mathfrak{k})$, it is decomposed into $diag(1, \omega^4) \otimes diag(1, \omega^2)$. Therefore, Ω_{n-1} is composed of $n-1$ single-qubit rotations as follows:

$$\Omega_{n-1} \;=\; \begin{pmatrix} 1 & 0 \\ 0 & \omega_n^{2^{n-1}-1} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 & 0 \\ 0 & \omega_n^{2^{j-1}} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 & 0 \\ 0 & \omega_n^2 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & \omega_n \end{pmatrix}. \tag{14}$$

The circuit obtained from the above decomposition is shown in Fig. 3. We apply a similar decomposition to F_j (for $j = n-1, n-2, \cdots, 2$). Finally, we have a full QFT decomposition composed of n Hadamard gates, $\frac{1}{2}n(n-1)$ controlled-rotations, and $\lfloor \frac{n}{2} \rfloor$ SWAP gates. Here, SWAP gates that appeared in a sequence of permutations $Q_n Q_{n-1} \cdots Q_2$ were optimized. It is known that a controlled-rotations can be implemented by three single-qubit rotation and two CNOTs, so that the number of elementary gates that appear in Fig. 3 is $O(n^2)$.

4.3 Decomposition of the QFT by CSD

If we apply $Q_n^\dagger$ to the input matrix F_n beforehand, then we have the following decomposition:

$$\begin{aligned} F_n Q_n^\dagger \;&=\; U \Sigma V \\ &=\; \begin{pmatrix} I^{\otimes n-1} & 0 \\ 0 & I^{\otimes n-1} \end{pmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} I^{\otimes(n-1)} & -I^{\otimes(n-1)} \\ I^{\otimes(n-1)} & I^{\otimes(n-1)} \end{pmatrix} \cdot \begin{pmatrix} F_{n-1} & 0 \\ 0 & -\Omega_{n-1} F_{n-1} \end{pmatrix}, \quad (15) \\ &=\; (H\sigma_z \otimes I^{\otimes(n-1)})(\sigma_z \otimes I^{\otimes(n-1)}) D_n (I \otimes F_{n-1}) \;=\; H_1 S. \quad (16) \end{aligned}$$

We can easily check that (15) satisfies the definition of CSD, where all θ_j's appear in c and s in (1) are $\pi/4$. Although we have to apply $Q_j^\dagger$ beforehand when we apply CSD recursively to F_j ($j = 1, \cdots, n-1$), we obtain the well-known QFT circuit. Using the feature, Tucci reproduced the well-known QFT circuit using a CSD-based algorithm [2, 14].

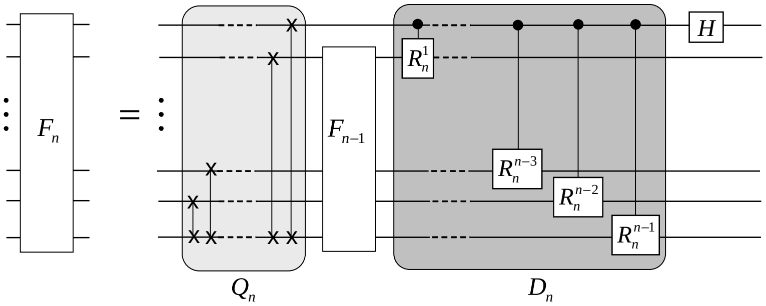


Fig. 3. Decomposition of the QFT by our algorithm. Here, $R_n^j = diag\,(1, \omega_n^j)$, i.e., a single-qubit rotation, and Q_n is composed of $n-1$ SWAP gates.

4.4 Decomposition of the QFT by the quantum Shannon decomposition (QSD)

The QSD [6] is a method that combines CSD and the quantum multiplexor decomposition. First, we compute CSD for the input matrix and then apply the quantum multiplexor decomposition (cf. [6], Theorem 12),

$$\begin{pmatrix} u_1 & 0 \\ 0 & u_2 \end{pmatrix} = \begin{pmatrix} v & 0 \\ 0 & v \end{pmatrix} \begin{pmatrix} d & 0 \\ 0 & d^\dagger \end{pmatrix} \begin{pmatrix} w & 0 \\ 0 & w \end{pmatrix}, \tag{17}$$

to U and V in (1). Here, $u_1 u_2^\dagger = v d^2 v^\dagger$ and $w = d v^\dagger u_2$. In the QFT, we apply the decomposition in (17) to V because U is an identity matrix in (15). Then, we have $v = I^{\otimes(n-1)}$ and $d = \sqrt{-\Omega_{n-1}^\dagger}$, and $w = \sqrt{-\Omega_{n-1}} F_{n-1}$ in (17). Here, $\sqrt{-\Omega_{n-1}}$ is a $2^{n-1} \times 2^{n-1}$ diagonal matrix whose (j,j)-th component is $i\omega_{n+1}^{j-1}$, $(j = 1, \cdots, 2^{n-1})$. Therefore, the decomposition of QFT by the QSD is as follows:

$$F_n Q_n^\dagger = \exp(i\sigma_y \otimes \delta_2) \exp(-i\sigma_z \otimes \delta_3)(I \otimes v_4), \tag{18}$$

where $v_4 = \sqrt{-\Omega_{n-1}} F_{n-1}$ and δ_2 and δ_3 are $2^{n-1} \times 2^{n-1}$ diagonal matrices. Each element of δ_2 is $\pi/2^{n-1}$ and each (j,j)-th element of δ_3 is $-j\pi/2^{n+1}$. Furthermore, $\sqrt{\Omega_{n-1}}$ is composed of $n-1$ single-qubit rotations as follows:

$$\sqrt{-\Omega_{n-1}} = i \begin{pmatrix} 1 & 0 \\ 0 & \omega_{n+1}^{2^{n-1}} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 & 0 \\ 0 & \omega_{n+1}^{2^{j-1}} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 & 0 \\ 0 & \omega_{n+1}^{2} \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & \omega_{n+1} \end{pmatrix}.$$

Here, (18) is also equal to the well-known QFT decomposition (12) after optimization as shown in Fig. 4. To obtain the decomposition in (18), note that we have to apply Q_j $(j = 0, \cdots, n)$ beforehand for F_j.

5 Conclusion

We introduced a new algorithm for computing any type-**AIII** KAK decomposition according to the given global Cartan involution Θ. Recursively performing the decomposition leads us to a quantum circuit composed of uniformly controlled rotations. The algorithm can derive any matrix decomposition corresponding to the type-**AIII** KAK decomposition, and it contains

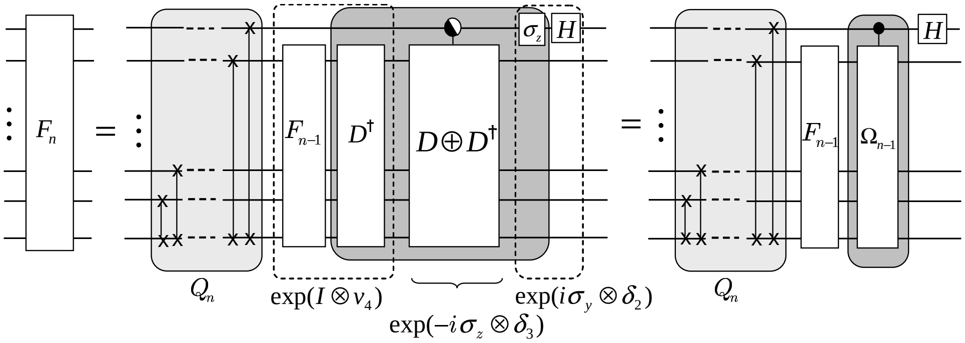


Fig. 4. Decomposition of the QFT by the QSD. Here, we move $Q_n^\dagger$ to the right-hand-side by inverting it. The dark-gray block in the middle circuit can be simplified. We show the simplified block by the same color in the rightmost circuit. Here, $D = \sqrt{-\Omega_{n-1}}$.

CSD and KGD as its special cases. This is because our algorithm contains arbitrariness in selecting the Cartan subalgebra $\mathfrak{h}$ and a square root matrix m for the given Cartan involution Θ, where m is a matrix derived from a global Cartan decomposition $g = km$. We also showed two methods for computing a square root matrix.

Although the correctness of our algorithm depends on Lie group theory, the main methods involved are eigenvalue decomposition and a simple replacement rule. Thus, we can compute a decomposition without knowledge of Lie group theory.

As an example, we showed that our method automatically reproduces the well-known QFT circuit for arbitrary n-qubits. When using CSD-based algorithms, we have to apply some permutations beforehand in order to reproduce the circuit. The same technique can not always be used to describe the canonical form of the decomposition for a given matrix because matrices do not always have a convenient form like QFT. Our algorithm might be useful in showing the effectiveness of $G = KAK$ matrix decompositions for other particular input matrices, because all matrices appearing through our algorithm can be described using a given input matrix g and a given Cartan involution Θ.

References

1. M. Nielsen, I. Chuang, *Quantum Computation and Quantum Information*, (Cambridge University Press, Cambridge, England, 2000).
2. R. Tucci, "A rudimentary quantum compiler", quant-ph/9902062.
3. K. Svore, "Compiling quantum circuits into elementary unitary operations", QIP2004, Waterloo, Canada (2004).
4. M. Möttönen, J. Vartiainen, V. Bergholm, and M. Salomaa, "Quantum circuits for general multiqubit gates", Phys. Rev. Lett. **93**, 130502 (2004) quant-ph/0404089.
5. M. Möttönen, and J. Vartiainen "Decompositions of general quantum gates", quant-ph/0504100.
6. V. Shende, S. Bullock, and I. Markov, "Synthesis of quantum logic circuits", quant-ph/0406176, to appear in IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems.
7. A. Barenco *et al.*, "Elementary gates for quantum computation", Phys. Rev. A **52**, pp. 3457–3467 (1995) quant-ph/9503016.
8. N. Khaneja and S. Glaser, "Cartan decomposition of $SU(2^n)$, constructive controllability of spin systems and universal quantum computing", Chem. Phys. **267**, pp. 11–23 (2002)

quant-ph/0010100.

9. S. Helgason, *Differential geometry, Lie groups, and Symmetric spaces*, (Academic Press, 1978).
10. N. Khaneja, R. Blockett, and S. Glaser, “Time optimal control in spin systems”, Phys. Rev. A **63**, 032308 (2001) quant-ph/0006114.
11. S. Bullock, “Note on the Khaneja Glaser decomposition”, Quantum Inf. Comput. **4**, pp. 396–400 (2004) quant-ph/0403141.
12. S. Bullock, G. Brennen, and D. P. O’Leary, “Time reversal and n-qubit canonical decomposition”, J. Math. Phys. **46**, 062105 (2005) quant-ph/0402051.
13. S. Bullock and I. Markov, “Arbitrary two-qubit computation in 23 elementary gates”, Phys. Rev. A **68**, 012318 (2003)
14. R. Tucci, “Quantum fast Fourier transform viewed as a special case of recursive application of Cosine-Sine decomposition”, quant-ph/0411097.
15. A. W. Knapp,*Lie groups beyond an introduction*, 2nd ed., (Birkhäuser, 2005).
16. M. Möttönen, J. Vartiainen, V. Bergholm, and M. Salomaa, “Transformation of quantum states using uniformly controlled rotations”, Quantum Inf. Comput. **5**, pp. 467–473 (2005) quant-ph/0407010.
17. G. Golub and C. Van Loan, *Matrix computations*, 3rd ed., (The Johns Hopkins University Press, 1996).

Appendix A Properties of eigenvalues of $m^2 = \Theta(G^\dagger)G$

In Section 3.2.1, we show that b is a block-diagonal matrix, each block of which is $R_x(2\zeta)$, where $\widetilde{\mathfrak{h}} = \mathrm{span}\{|j\rangle\langle j| \otimes i\sigma_x \mid j = 0, \cdots, 2^{n-1} - 1\} \subset \mathfrak{m}$. This appendix provide two lemmas and proofs for the eigenvalues of $m^2 = \Theta(G^\dagger)G$, where $G \in SU(2^n)$ to prove that. Let $X = m^2 = \Theta(G^\dagger)G$.

Lemma A.1 (Properties of complex eigenvalues of X)

1. *The number of the complex eigenvalues of X, with multiplicity counted, is even. They are of the form α_1, $\overline{\alpha_1}$, α_2, $\overline{\alpha_2}$, $\ldots$, α_t, $\overline{\alpha_t}$, repeated with multiplicity.*

2. *There exists unit length vectors u_1, $\ldots$, u_{2t} that are mutually orthogonal and satisfy $\sigma_{nz}u_{2j-1} = u_{2j-1}$ and $\sigma_{nz}u_{2j} = -u_{2j}$. Furthermore, there exists ζ_1, $\ldots$, $\zeta_t \in \mathbb{R}$ satisfying $Xu_{2j-1} = \cos(\zeta_j)u_{2j-1} - i\sin(\zeta_j)u_{2j}$ and $Xu_{2j} = -i\sin(\zeta_j)u_{2j-1} + \cos(\zeta_j)u_{2j}$.*

Proof. Let α_1, $\ldots$, α_{2^n} be the eigenvalues of X, repeated with multiplicity. Then, except for the order, the elements of the sequence $\overline{\alpha_1}$, $\ldots$, $\overline{\alpha_{2^n}}$ are equal to α_1, $\ldots$, α_{2^n} because $\Theta(X) = X^\dagger$. Therefore, the number of the complex eigenvalues, counted with multiplicity, is even, and we denote it as $2t$. We can write the complex eigenvalues as α_1, $\overline{\alpha_1}$, $\ldots, \alpha_t$, $\overline{\alpha_t}$. That is, the first statement holds.

Let u be an eigenvector of X corresponding to the eigenvalue α, then $\sigma_{nz}u$ is an eigenvector of X corresponding to the eigenvalue $\overline{\alpha}$ because $X(\sigma_{nz}u) = \sigma_{nz}X^\dagger u = \overline{\alpha_j}(\sigma_{nz}u)$. Now, let W be the eigenspace corresponding to a complex eigenvalue α and β_1, $\ldots$, β_r be an orthonormal basis of W. Then, let $\sigma_{nz}\beta_1, \ldots, \sigma_{nz}\beta_r$ be an orthonormal basis of the eigenspace corresponding to the eigenvalue $\overline{\alpha}$. Therefore, $u_{2j-1} = \beta_j + \sigma_{nz}\beta_j$ and $u_{2j} = \beta_j - \sigma_{nz}\beta_j$ for $j = 1, \cdots, r$, span the eigenspace of X corresponding to the eigenvalues α and $\overline{\alpha}$. Furthermore, they are eigenvectors of σ_{nz} because $\sigma_{nz}u_{2j-1} = u_{2j-1}$ and $\sigma_{nz}u_{2j} = -u_{2j}$. On the other hand, the following relations hold:

$$\begin{aligned} Xu_{2j-1} &= \alpha\beta_j + \overline{\alpha}\sigma_{nz}\beta_j = \frac{\alpha+\overline{\alpha}}{2}u_{2j-1} + \frac{\alpha-\overline{\alpha}}{2}u_{2j} = \mathrm{Re}(\alpha)u_{2j-1} + i\mathrm{Im}(\alpha)u_{2j}, \\ Xu_{2j} &= \alpha\beta_j - \overline{\alpha}\sigma_{nz}\beta_j = \frac{\alpha-\overline{\alpha}}{2}u_{2j-1} + \frac{\alpha+\overline{\alpha}}{2}u_{2j} = i\mathrm{Im}(\alpha)u_{2j-1} + \mathrm{Re}(\alpha)u_{2j}. \end{aligned}$$

Therefore, put $\zeta = -\arg(\alpha)$. Then $\cos(\zeta) = \mathrm{Re}(\alpha)$ and $\sin(\zeta) = -\mathrm{Im}(\alpha)$. That is,

$$Xu_{2j-1} = \cos(\zeta)u_{2j-1} - i\sin(\zeta)u_{2j}, \qquad Xu_{2j} = -i\sin(\zeta)u_{2j-1} + \cos(\zeta)u_{2j}.$$

Similar arguments hold for the other complex eigenvalues. □

Lemma A.2 (Properties of real eigenvalues of X) *Real eigenvalues of X are ± 1.*

1. *The multiplicity of the eigenvalue 1 is even. There exists an orthonormal basis u_{2t+j} ($j = 1, \cdots, 2\mu$) of the eigenspace corresponding to the eigenvalue 1 that satisfies the $\sigma_{nz}u_{2t+2j-1} = u_{2t+2j-1}$ and $\sigma_{nz}u_{2t+2j} = -u_{2t+2j}$.*

2. *The multiplicity of the eigenvalue -1 is even. There exists an orthonormal basis $u_{2t+2\mu+j}$ ($j = 1, \cdots, 2\nu$) of the eigenspace corresponding to the eigenvalue -1 that satisfies $\sigma_{nz}u_{2t+2\mu+2j-1} = u_{2t+2\mu+2j-1}$ and $\sigma_{nz}u_{2t+2\mu+2j} = -u_{2t+2\mu+2j}$.*

Proof. Lemma A.1 implies that the product of all the complex eigenvalues is 1, and thus the product of all the real eigenvalues is 1. Therefore, the real eigenvalues of X are 1 or -1, thus both of the multiplicities of the eigenvalues 1 and -1 are even. Let W_1 and W_2 be the eigenspaces of X corresponding to the eigenvalues of 1 and -1, respectively. Then $\sigma_{nz}W_1 \subset W_1$ and $\sigma_{nz}W_2 \subset W_2$ hold because $\Theta(X) = X^\dagger$. Put $W = W_1 \oplus W_2$. Then $W^\perp$is the direct sum of the eigenspaces for the complex eigenvalues of X. The trace of $\sigma_{nz}|_W$ is 0 because Lemma A.1 implies that the trace of $\sigma_{nz}|_{W^\perp}$ is 0, and this implies that the multiplicities of the eigenvalues of 1 and -1 of $\sigma_{nz}|_W$ are equal. We write $W_1 = W_{11} \oplus W_{12}$ and $W_2 = W_{21} \oplus W_{22}$, where W_{11} and W_{21} are the eigenspaces of σ_{nz} corresponding to the eigenvalue 1, and W_{12} and W_{22} are the eigenspaces of σ_{nz} corresponding to the eigenvalue -1. Let $\dim W_{ij}$ be d_{ij}. Then we have $d_{11} + d_{21} = d_{12} + d_{22}$.

We can make similar arguments for$X' = \Theta(G)G^\dagger$. We write $W'_1 = W'_{11} \oplus W'_{12}$ and $W'_2 = W'_{21} \oplus W'_{22}$, where W'_{11} and W'_{21} are the eigenspaces of σ_{nz} corresponding to the eigenvalue 1, and W'_{12} and W'_{22} are the eigenspaces of σ_{nz} corresponding to the eigenvalue -1. Let $\dim W'_{ij}$ be d'_{ij}. Then we have $d'_{11} + d'_{21} = d'_{12} + d'_{22}$.

From Lemma A.3 below, we have $d_{11} = d'_{11}$, $d_{12} = d'_{12}$, $d_{21} = d'_{22}$, and $d_{22} = d'_{21}$. Then, we have $d_{11} = d_{12}$ and $d_{21} = d_{22}$. Therefore, the statements of the lemma hold. □

Lemma A.3 *Let $X = \Theta(G^\dagger)G$ and $X' = \Theta(G)G^\dagger$.*

1. *If $Xu = u$ and $\sigma_{nz}u = u$, then $X'(Gu) = Gu$ and $\sigma_{nz}(Gu) = Gu$.*

2. *If $Xu = u$ and $\sigma_{nz}u = -u$, then $X'(Gu) = Gu$ and $\sigma_{nz}(Gu) = -Gu$.*

3. *If $Xu = -u$ and $\sigma_{nz}u = u$, then $X'(Gu) = -Gu$ and $\sigma_{nz}(Gu) = -Gu$.*

4. If $Xu = -u$ and $\sigma_{nz}u = -u$, then $X'(Gu) = -Gu$ and $\sigma_{nz}(Gu) = Gu$.

Proof. First, we prove the statements for the eigenvalues and eigenvectors of X'. Since $G^\dagger X' G = \Theta(X)$ holds, we have $(X'(Gu), Gu) = (G^\dagger X' Gu, u) = (\Theta(X)u, u) = (X^\dagger u, u) = (u, Xu)$. Thus, the equation $Xu = \epsilon u$, where $\epsilon = \pm 1$, implies $(X'(Gu), Gu) = \epsilon(u, u)$. On the other hand, the Cauchy-Schwarz inequality implies $|(X'(Gu), Gu)| \leq \|X'(Gu)\| \cdot \|Gu\|$. Since the right-hand side is equal to $\|Gu\|^2 = \|u\|^2 = (u, u) = |(X'(Gu), Gu)|$, the equality $|(X'(Gu), Gu)| = \|X'(Gu)\| \cdot \|Gu\|$ holds. Therefore, we have $X'(Gu) = \alpha(Gu)$ for some $\alpha \in \mathbb{C}$; that is, Gu is an eigenvector of X' corresponding to the eigenvalue α. The equality $\alpha = \epsilon$ follows $\epsilon(u, u) = (X'(Gu), Gu) = (\alpha Gu, Gu) = \alpha(Gu, Gu) = \alpha(u, u)$.

To prove the statements for the eigenvalues and eigenvectors of σ_{nz}, we use the following equations:

$$(\sigma_{nz}(Gu), Gu) = (G^\dagger \sigma_{nz} Gu, u) = (\sigma_{nz} Xu, u) = (Xu, \sigma_{nz}u).$$

The equations $Xu = \epsilon u$ and $\sigma_{nz}u = \epsilon' u$, where ϵ, $\epsilon' = \pm 1$, imply $(\sigma_{nz}(Gu), Gu) = \epsilon\epsilon'(u, u)$. From similar arguments for the eigenvalues and eigenvectors of X', we have $\sigma_{nz}(Gu) = \epsilon\epsilon'(Gu)$. □