IMA Journal of Numerical Analysis (2022) **00**, 1–26
https://doi.org/DOI HERE

Lax Dynamics for Cartan Decomposition with Applications to Hamiltonian Simulation

MOODY T. CHU*
Department of Mathematics, North Carolina State University, Raleigh, 27695-8205, North Carolina, USA
*Corresponding author: mtchu@ncsu.edu

[Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year]

Simulating the time evolution of a Hamiltonian system on a classical computer is hard – The computational power required to even describe a quantum system scales exponentially with the number of its constituents, let alone integrate its equations of motion. Hamiltonian simulation on a quantum machine is a possible solution to this challenge – Assuming that a quantum system composing of spin-½ particles can be manipulated at will, then it is tenable to engineer the interaction between those particles according to the one that is to be simulated, and thus predict the value of physical quantities by simply performing the appropriate measurements on the system. Establishing a linkage between the unitary operators described mathematically as a logic solution and the unitary operators recognizable as quantum circuits for execution is therefore essential for algorithm design and circuit implementation. Most current techniques are fallible because of truncation errors or the stagnation at local solutions. This work offers an innovative avenue by tackling the Cartan decomposition with the notion of Lax dynamics. Within the integration errors that is controllable, this approach gives rise to a genuine unitary synthesis which not only is numerically feasible, but also can be utilized to gauge the quality of results produced by other means, and extend the knowledge to a wide range of applications. This paper aims at establishing the theoretic and algorithmic foundations by exploiting the geometric properties of Hamiltonian subalgebras and describing a common mechanism for deriving the Lax dynamics.

Keywords: Hamiltonian simulation; Cartan decomposition; Lax dynamics; exponential map

1. Introduction

Quantum computing, with its potential capability of transmitting information massively, swiftly, concurrently, and securely, shows great promises to the next-generation quantum-enabled science and technology [1, 21]. The range of applications is broad and far reaching, including communicating [3], sensing [23], computing [47], machine learning and big data [29, 41, 55], and other societal grand challenges, just to name a few. The current development of hardware devices is still in its infancy, capable of handling only a few qubits[1]. Even so, it is of great interest and vital importance to gain insight into how a quantum computation could be or should be conducted. This work concerns using a dynamical system to facilitate Hamiltonian simulation.

Typically, the mathematical formalism for quantum computation is expressed through finite dimensional Hilbert spaces [30, 45]. A mixed quantum state, which often is described in terms of its density matrix, is a probabilistic ensemble of pure states, i.e., the unit vectors in the corresponding

[1] As of this paper is written, Google's Sycamore quantum computer has 53 qubits. IBM's Eagle has 127 qubits. IBM plans to unveil its Osprey with 433-qubit processor, and to introduce the Condor with over 1000 qubits. In the meantime, Microsoft's Azure offers access to an array of quantum hardware from a single cloud service.

Hilbert space. Of practical importance is a multipartite quantum system involving the entanglement of several subsystems. With respect to properly selected basis the entanglement can be mathematically characterized via the Kronecker product [26, 31]. Quantum algorithms usually are described via quantum circuits which consist of series of elementary operations, called quantum gates, in line with the technological specifics of the hardware. Independent of the machine on which the algorithm will be executed, a quantum circuit is represented as a unitary operator [46]. Synthesizing the relationship between the unitary operators described as matrices and the unitary operators described as circuits is therefore essential, as it helps understand better not only how to design new algorithms, but also how to implement the circuits [17, 37, 38, 51] on a quantum machine.

The forward problem of obtaining the unitary matrix representation from a given circuit is easy. The circuit can even be simulated on a classical computer [54]. It is the inverse problem of designing the circuit from a given unitary matrix that is more complicated [15, 38], and falls in the realm of the so-called quantum compilation. What happens is that, unlike conventional digital machines, no quantum machine can accept directly a matrix as input. The central task of unitary synthesis (or quantum circuit synthesis) is to translate a quantum operation described mathematically but in an unknown form for the hardware into a sequence of elementary instructions understandable by the machine, e.g., a sequence of one and two-qubit gates. This task of unitary synthesis has broad applications such as the quantum state preparation (e.g., via the unitary coupled cluster formalism), the thermal state preparation (e.g., by the unitary embedding of mixed states), and even the execution of the fundamental quantum arithmetic logic.

More specifically, as is typical in the programming for a conventional machine, in order to cope with the complexity of a high-level quantum program, several levels of abstractions are required in the quantum compilation process. The choice of layers and the choice of transformations needed in each layer highly depend on the applications. For example, on quantum computers that are noisy with short decoherence time, the attention is at finding a quantum circuit that reduces 2-qubit gates since these are noisier than 1-qubit gates, whereas on fault-tolerant quantum computers that do error-correction, the goal is to reduce the usage of gates that require a large amount of error corrections, since they are more expensive. In all, an effective abstractions should always take the following tasks into account:

- Synthesis: Converting the mathematical formulation of the logic for solving a given problem into unitary operations that meet the qubit/gate structures/constraints of a quantum machine.
- Optimization: Developing a quantum circuit with, say, the shortest sequence of gates.
- Mapping: Translating a technology-independent quantum circuit into a technology-dependent quantum circuit.

As the memory requirement for storing the unitary matrix associated with an n-qubit system grows exponentially, i.e., $2^n \times 2^n$, any methods will be limited rapidly by the size of the problem on both classical and quantum machines. The challenge is thus to synthesize quickly and optimally any quantum operator or quantum state on only a few qubits. Being able to accomplish this task is of fundamental importance and is crucial to many applications [15, 24, 38].

The focus of this paper is on a paradigmatic problem where we want to perform the unitary synthesis of time evolution under a time-independent Hamiltonian $\mathscr{H}$, i.e., the family of unitary matrices

$$U(t) = e^{-i\mathscr{H}t}, \tag{1.1}$$

with a given Hermitian matrix $\mathscr{H}$ in $\mathbb{C}^{2^n \times 2^n}$. We shall explain the intellectual merit of studying this particular unitary synthesis and the details of our methodology in the sequel. For now, it is important

to point out first that the conventional ways of using floating-point arithmetic to compute this matrix exponential, and there are many, such as the widely used scaled and squared Padé approximation [42, 43], are not suitable for and, in fact, not implementable on quantum machines.

A host of approximation techniques have already been proposed in the literature for simulating the unitary evolution (1.1) on quantum machines, including the Trotter-Suzuki formulas, the Taylor series, random walk (qubitization), and so on [4, 27, 39]. In [4], for example, the Hamiltonian dynamics is simulated with a truncated Taylor series where the terms are expressed via linear combinations of unitary operations together with a robust form of oblivious amplitude amplification. In [27], as another example, under the structure that the layout is such that a qubit may only interact with qubits in its vicinity (the so-called lattice Hamiltonian), the time-evolution unitary is approximated by a product of small unitaries based on the Lieb–Robinson bounds and, hence, the performance is good only when the Hamiltonian is close to commuting. These algorithms are reported to have been optimized in that the circuit depth scales almost linearly in the evolution time and inverse logarithmically in the approximation error. However, it must be noted that these approaches are subject to some inherent constraints and that the approximation errors affect the simulation authenticity.

Recently, there has also been a promising approach by explicitly parameterizing a given unitary transformation over an n-qubit system in terms of 1- and 2-qubit operations [17, 22, 34, 35, 49]. The idea is based on the observation in the seminal paper [34] that, if the Cartan decomposition is properly managed, then elements of the special unitary group $\mathrm{SU}(2^n)$ can be determined, up to some local unitaries in $\mathrm{SU}(2)^{\otimes n}$, by components generated from certain Abelian subalgebras of the Lie algebra $\mathfrak{su}(2^n)$. A constructive algorithm to perform such a decomposition is not easy. The procedure proposed in [49] is recursive in nature, which requires at every step to solve numerically the zeros of a matrix polynomial obtained by a properly truncated Baker-Campbell-Hausdorff (BCH) formula. As such, though the computational concepts involved are straightforward, the truncation of the BCH formula at higher orders at every iteration so as to control the overall errors is concerning. The fact that working with the basis of the entire $\mathfrak{su}(2^n)$ requires generically exponential circuit depth for arbitrary unitaries is another serious drawback.

It is under such a background, we propose in this paper a different way to find the Cartan decomposition. Our idea is to apply the Cartan theory only to a necessary subalgebra of $\mathfrak{su}(2^n)$, in which a Lax dynamics is constructed, whose solution flow can be tracked by any available numerical ODE techniques. Our specific objectives in this paper are to describe how the subalgebra can be generated effectively on an as-needed basis, to introduce a general mechanism for deriving the Lax dynamics, and to propose the notion of continued Cartan decomposition that breaks down $\mathscr{H}$ to prepare for the unitary synthesis.

This work is motivated by our past experience with the Toda lattice [6, 7] and the isospectral flow approach for other applications [8, 9, 12], and by seeing the similarity between the setting of the general Lax dynamics [11, 13] and the Cartan decomposition. We therefore think that developing a computational framework for the unitary synthesis problem (1.1) in general is possible. We will explain the various benefits of this undertaking, outline our methodologies, and furnish some preliminary but promising empirical evidences.

This paper is organized as follows. Since some of the ideas are from across fields, we use some extended examples to help convey the points. Beginning in Section 2 with some basic information, we present a motivation for studying (1.1), the idea for working on reduced subalgebras [35], the connection to the Cartan decomposition [34, 49], and a brief outline of the Lax dynamics [11, 13], followed by a worked-out example. The general framework with greater mathematical details is delineated in Section 3. We describe a mechanism for dimension reduction by using a combinatorial approach which

checks only integer arrays and, thus, is highly efficient. We put forward a recipe for the construction of the differential system that guarantees the convergence of the Lax dynamics from the given Hamiltonian matrix to a decomposition form which can be implemented on a quantum machine. In Section 4, we discuss the idea of continued Cartan decomposition together with an example to demonstrate that not only can we do recursive factorization, as that which is achievable via conventional linear algebra techniques [17], but also that each factor is from a smaller subalgebra and is readily implementable on a quantum machine, which is hard to accomplish by the conventional means. The entire framework can be extended to general or k-local Hamiltonian. We only outline the idea in Section 5, as more investigation is needed to complete the generalization. Finally, in Section 6 we present an example with a 12-qubit system that would have been challenging for a conventional machine, but our approach with the help of existing numerical ODE integrators can accomplish the Cartan decomposition by merely checking a few integer arrays.

2. Basics

In this section, we outline some background information for building our framework. At first glance, it might seem that the subjects involved in the following discussion are too abstract, uncorrelated, and even not suitable for classical numerical computation. Nevertheless, it is precisely our point to bring in a synthesis of knowledge and understanding from different fields to the field of quantum computation.

2.1. *Application to the Schrödinger equation*

We have already mentioned that a quantum circuit acting on n-qubits is always represented by a unitary matrix and, hence, to translate between a unitary operator and a quantum circuit is imperative. We now briefly explain, among other applications, why the specific unitary synthesis in the form (1.1) is of particular interest.

Quantum mechanics postulates that the evolution of a quantum state appears as a wave function $|\psi\rangle$ whose dynamics can be described by the Schrödinger equation

$$\imath\hbar\frac{\partial}{\partial t}|\psi\rangle(\mathbf{r},t)=\left[-\frac{\hbar^2}{2m}\nabla^2+V(\mathbf{r})\right]|\psi\rangle(\mathbf{r},t), \tag{2.1}$$

where $\hbar$ is the Planck constant and the quantity inside the brackets, called the *Hamiltonian*, corresponds to the energy of the system. If we discretize $|\psi\rangle(\mathbf{r},t)$ in the spatial variable $\mathbf{r}$, e.g., particles living on a lattice [38], and denote the resulting vector as $|\Psi\rangle(t)$, then (2.1) can be expressed as

$$\imath\hbar\frac{\partial}{\partial t}|\Psi\rangle(t)=\mathscr{H}|\Psi\rangle(t), \tag{2.2}$$

where $\mathscr{H}$ is a time-independent Hermitian matrix (actually, in the current context, $\mathscr{H}$ is real-valued and symmetric). It follows that

$$|\Psi(t)\rangle=e^{\frac{-\imath\mathscr{H}t}{\hbar}}|\Psi(0)\rangle \tag{2.3}$$

and $e^{\frac{-\imath\mathscr{H}t}{\hbar}}$ is unitary. The need to calculate $e^{\frac{-\imath\mathscr{H}t}{\hbar}}$ is now apparent. Being able to execute this calculation by using quantum gates is even more paramount because the size of the matrix grows exponentially.

A historical note on this calculation is worth mentioning. The abstract of the influential paper by Lloyd [38] in Science consists of one sentence only, "Feynman's 1982 conjecture, that quantum computers can be programmed to simulate any local quantum system, is shown to be correct," but

the impact of such an affirmation is profound. Feynman's idea was that if we could build a quantum simulator at our disposal, composed of spin-½ particles that we could manipulate at will, then we would be able to engineer the interaction between those particles according to the one we want to simulate, and thus predict the value of physical quantities by simply performing the appropriate measurements on the quantum simulator [15]. The proof in [38] was via the notion that even evolving in small time steps of (2.2) would allow efficient simulation of any many-body quantum Hamiltonian containing few-particle interactions. Therefore, the application of the unitary synthesis in the form (1.1) for the Schrödinger equation alone is already of considerable significance.

2.2. *Dimension reduction on Lie subalgebra*

Some of the most important operators in a single-qubit system are the Pauli matrices,

$$X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; \quad Y := \begin{bmatrix} 0 & -\imath \\ \imath & 0 \end{bmatrix}; \quad Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2.4}$$

Each Pauli matrix is readily an observable describing the spin of a spin-½ particle. Pauli matrices are Hermitian and unitary. When exponentiated, the Pauli matrices give rise to rotation matrices around the three orthogonal axis in 3-dimensional space. Together with the identity matrix I, an element in the set $\{X,Y,Z,I\}^{\otimes n}$, i.e., a tensor product of n matrices selected from the set $\{X,Y,Z,I\}$, is referred to as a Pauli string. Elements in $\{X,Y,Z,I\}^{\otimes n}$ are mutually orthogonal with respect to the Frobenius inner product over $\mathbb{C}^{2^n\times 2^n}$, so they span all possible $2^n \times 2^n$ Hermitian matrices. With a multiplication by the imaginary number $\imath$ to the Pauli strings, we have the Lie algebra of the unitary group $\mathrm{U}(2^n)$, which consist of skew-Hermitian matrices with the Lie bracket given by the commutator. For mathematical rigor, the subsequent discussion requires the assumption of semisimplicity, but $\mathrm{U}(2^n)$ is not. Instead, we shall work with the special unitary group $\mathrm{SU}(2^n)$ and its Lie algebra $\mathfrak{su}(2^n)$. Recall that $\mathfrak{su}(2^n)$ consists of skew-Hermitian matrices with trace zero and has real dimension 4^n-1 because the element $\imath I^{\otimes n}$ is removed from the basis. In practice, if $I^{\otimes n}$ ever appears as a term in the makeup of $\mathscr{H}$, its contribution amounts to only a scalar multiplication which can easily be handled.

While any given Hamiltonian $\mathscr{H}$ can be expressed as a linear combination of Pauli strings, the trouble is that the matrix exponential of $-\imath\mathscr{H}$ in the summation form cannot easily be simulated on a quantum circuit. The simplest reason to see this infeasibility is that the BCH formula or the Zassenhaus formula [52] will involve infinite terms or products in the expansion. This is where the work proposed in this paper might come in handy.

Most earlier works for the unitary synthesis (1.1) of a given Hamiltonian $\mathscr{H}$ require using the entire $\mathfrak{su}(2^n)$ for the approximation [4, 22, 27, 34, 39, 49]. Among other issues, the implementation will require long circuit depth which may cause loss of fidelity due to noise presented in current technology. A close inspection of the Zassenhaus formula implies that only nested commutators of the individual terms in $\mathscr{H}$ will appear in the ultimate exponents [5, 35]. Therefore, for a fixed $\mathscr{H}$, it is feasible to work with the subalgebra $\mathfrak{g}(\imath\mathscr{H}) \subset \mathfrak{su}(2^n)$ which is defined to be the closure of all possible Lie brackets generated from individual terms in $\mathscr{H}$. Under some suitable conditions the dimension of $\mathfrak{g}(\imath\mathscr{H})$ can be far less than 4^n-1, as we will see in our examples. This gives rise to the basic notion of dimension reduction.

Given a subset V of an arbitrary Lie algebra, there is no general way to characterize the closure $\mathfrak{g}(V)$. Some discussions on the dimensionality for some special subsets V can be found in [25, 35, 48]. However, if we use Pauli strings as the basis, the following lemma implies that such a task is possible.

Lemma 1 *Let B_ℓ, $\ell = 1,\ldots,4^n$, denote elements in $\{X,Y,Z,I\}^{\otimes n}$ multiplied by ι.*

1. *If the commutator $[B_i,B_j]$ of two distinct Pauli strings B_i,B_j is not zero, then $[B_i,B_j] = cB_k$ for some $k \neq i$ or j, and c is either 2 or -2.*
2. *If $[B_i,B_j] = cB_k$, then $[B_j,B_k] = cB_i$ and $[B_k,B_i] = cB_j$.*
3. *The proportional constants c depends on the makeup of the Pauli strings.*

Proof : Observe first that the Pauli matrices satisfy the relationships

$$XX = YY = ZZ = -\iota XYZ = I. \tag{2.5}$$

These are sufficient to establish other cyclic relationships such as $XY = \iota Z = -YX$, $YZ = \iota X = -ZY$, $ZX = \iota Y = XZ$, and so on. It follows that the commutator $[B_i,B_j]$ can eventually be simplified to either zero or a scalar multiple of an element $B_k \in \{X,Y,Z,I\}^{\otimes n}$. Observe that

$$B_iB_j = \begin{cases} -I, & \text{if } i = j, \\ -B_jB_i, & \text{if } [B_i,B_j] \neq 0. \end{cases}$$

Therefore, the scalar multiple c must be 2 or -2. Suppose that we have $[B_i,B_j] = cB_k$. Then

$$[B_j,B_k] \quad = \quad \frac{1}{c}[B_j,[B_i,B_j]] = \frac{1}{c}(2B_jB_iB_j + 2B_i) = \frac{4}{c}B_i.$$

A similar argument can be applied to $[B_k,B_i]$. □

The practical question is how to search to determine $\mathfrak{g}(\iota\mathscr{H})$ in the most efficient way. Given $\mathscr{H}$ which is a finite sum of Pauli strings, an exhaustive search by checking through all possible brackets $[A,B]$ and continuing to check $[[A,B],C]$ and so on of all terms until no more elements can be added to $\mathfrak{g}(\iota\mathscr{H})$ is possible, but such a task becomes too demanding when the dimension is high. The first specific goal of this work is to develop a mechanism that identifies and executes elements in $\mathfrak{g}(\iota\mathscr{H})$ by checking only a few integer indices on an as-needed basis. Technical details are given in Section 3.1.

2.3. *Connection to Cartan decomposition*

The notion of Cartan decomposition of a semisimple Lie algebra plays an important role in the theoretical study of structure and representation in the Lie theory [53]. It is interesting that this abstract mathematics now finds applications in quantum simulation. This connection is nothing new in the literature [19, 28, 34, 49]. What is new is to find the Cartan decomposition for the subalgebra $\mathfrak{g}(\iota\mathscr{H})$ by means of the Lax dynamics, which is the central theme of this work. For completion we state two fundamental theorems concerning the Cartan decomposition [22, 34]. The results hold for any general semisimple Lie algebra, but we apply them to the subalgebra $\mathfrak{g}(\iota\mathscr{H})$ only.

Given a real semisimple Lie algebra $\mathfrak{g}$, a Lie algebra automorphism $\theta : \mathfrak{g} \to \mathfrak{g}$ is called an involution if its square is equal to the identity. The Cartan decomposition can be characterized via the notion of involution.

Theorem 2 *Let θ be an involution on a semisimple Lie algebra $\mathfrak{g}$. Let $\mathfrak{k}$ and $\mathfrak{p}$ denote the eigenspaces corresponding to the eigenvalues +1 and -1 of θ, respectively. Then $\mathfrak{g}$ can be decomposed as a direct sum*

$$\mathfrak{g} = \mathfrak{k} \oplus \mathfrak{p}, \tag{2.6}$$

with the properties that

$$[\mathfrak{k},\mathfrak{k}] \subseteq \mathfrak{k},\ [\mathfrak{k},\mathfrak{p}] \subseteq \mathfrak{p},\ \textit{and}\ [\mathfrak{p},\mathfrak{p}] \subseteq \mathfrak{k}. \tag{2.7}$$

Conversely, any decomposition of $\mathfrak{g}$ with properties (2.7) defines an involution.

Obviously, $\mathfrak{k}$ is a Lie subalgebra by itself, whereas any subalgebra within $\mathfrak{p}$ is necessarily commutative. Let $\mathfrak{h} \subset \mathfrak{p}$ be any of the maximal subalgebras contained in $\mathfrak{p}$ and denote

$$\mathfrak{p} = \widetilde{\mathfrak{p}} \oplus \mathfrak{h}. \tag{2.8}$$

The pair $(\mathfrak{k},\mathfrak{p})$ is often referred to as a Cartan pair of $\mathfrak{g}$, and the subalgebra $\mathfrak{h}$ as a *Cartan subalgebra*.

The next theorem asserts that this Cartan subalgebra $\mathfrak{h}$ serves as the fundamental stratum of $\mathfrak{p}$ because everything else in $\mathfrak{p}$ can be obtained from a rotation (conjugation) of $\mathfrak{h}$. It is this structure that catches our attention to connect it to the Lax dynamics.

Theorem 3 *Let $\mathfrak{K}$ be the Lie subgroup corresponding to the Lie subalgebra $\mathfrak{k}$, i.e., $\mathfrak{K} = e^{\mathfrak{k}}$. For a fixed $K \in \mathfrak{K}$, let $\mathrm{Ad}_K : \mathfrak{g} \to \mathfrak{g}$ denote the conjugation action, i.e., $\mathrm{Ad}_K(g) = KgK^{-1}$ for $g \in \mathfrak{g}$. Then*

1. $\mathfrak{p} = \bigcup_{K\in\mathfrak{K}} \mathrm{Ad}_K(\mathfrak{h})$.
2. *If $\widehat{\mathfrak{h}}$ is another Cartan subalgebra in $\mathfrak{p}$, then $\widehat{\mathfrak{h}} = \mathrm{Ad}_K(\mathfrak{h})$ for some $K \in \mathfrak{K}$.*

With regard to our unitary synthesis problem (1.1), suppose that the task of forming $\mathfrak{g}(\iota\mathscr{H}) \subset \mathfrak{su}(2^n)$ is done. Suppose that the involution

$$\theta_1(g) := -g^{\top} \tag{2.9}$$

is used, where we stress that g might be complex-valued but we take only the transpose, not the conjugate transpose, of $g \in \mathfrak{g}(\iota\mathscr{H})$. By Theorem 2, we can construct the decomposition

$$\mathfrak{g}(\iota\mathscr{H}) = \mathfrak{k} \otimes (\widetilde{\mathfrak{p}} \otimes \mathfrak{h}), \tag{2.10}$$

where elements in $\mathfrak{k}$ are both skew-Hermitian and skew-symmetric and, hence, must be real-valued, while elements in $\mathfrak{p}$ are both skew-Hermitian and symmetric and, hence, its entries must be pure imaginary. Since $\mathscr{H}$ in the context of (1.1) is real-valued and symmetric, $\iota\mathscr{H} \in \mathfrak{p}$. By Theorem 3, there exist $\kappa \in \mathfrak{k}$ and $\eta \in \mathfrak{h}$ such that

$$-\iota\mathscr{H} = e^{\kappa}\eta e^{-\kappa}. \tag{2.11}$$

It follows that the unitary synthesis can be realized from

$$e^{-\iota\mathscr{H}t} = e^{\kappa}e^{\eta t}e^{-\kappa}. \tag{2.12}$$

Such a fundamental factorization is significant because the commutativity of $\mathfrak{h}$ implies that the exponential $e^{\eta t}$ on the right hand side of (2.12) is readily implementable on quantum circuits. We shall deal with the exponential e^{κ} similarly, which will be discussed separately in Section 3.3, and there are several ways to approximate e^{κ}. If we can control the precision in computing $\kappa \in \mathfrak{k}$ and $\eta \in \mathfrak{h}$, then the product on the right hand side of (2.12) is a successful synthesis of $e^{-\iota\mathscr{H}t}$.

2.4. *New realization of Lax dynamics*

The notion of Cartan decomposition generalizes the well-studied polar decomposition, *QR* decomposition, and singular value decomposition over the general matrices. It has been established that these different matrix decompositions are related to a variety of differential systems known as the Lax dynamics [6, 8, 10, 11]. We now employ the Lax dynamics to help calculate the Cartan decomposition by numerical ODE techniques.

The decomposition (2.11) resembles the classical spectral decomposition of the matrix $-\iota\mathscr{H}$, where e^{κ} is unitary and η has the same spectrum as $-\iota\mathscr{H}$. The difference is that η is not necessarily diagonal. We do know one additional property for elements in the Cartan subalgebra $\mathfrak{h}$, namely, commutable symmetric matrices can be diagonalized simultaneously. However, finding the eigenvectors for $\eta \in \mathfrak{su}(2^n)$ for its diagonalization is as hard as finding the spectral decomposition of $-\iota\mathscr{H}$, which is precisely what is prohibited in the first place. Being able to perform the decomposition (2.11) without invoking spectral decomposition and, in fact, we only need to work with $\dim(\mathfrak{p})$ variables, should be considered as a valuable contribution of this work.

To convey the idea, we first recollect some background information. Let $\mu, \nu : \mathfrak{gl}(N) \to \mathfrak{gl}(N)$ denote two linear operators on the general linear space $\mathfrak{gl}(N)$ of dimension N such that any element $X \in \mathfrak{gl}(N)$ can be expressed as

$$\mu(X) + \nu(X) = X. \tag{2.13}$$

The splitting of $\mathfrak{gl}(N)$ in (2.13) need not even be a direct sum. Different choices of μ and ν lead to different dynamics, including the so-called Toda lattice, the SVD flow, and so on. See a list in our paper [10]. Let the dot $\dot{}$ denote the differential operator $\frac{d}{dt}$ with respect to the parameter t. Consider the initial value problem

$$\dot{X}(t) := [X(t), \mu(X(t))], \quad X(0) := X_0, \tag{2.14}$$

referred to as a general *Lax dynamical system*. Consider also the two associated systems:

$$\dot{g}_1(t) := g_1(t)\mu(X(t)), \quad g_1(0) := I, \tag{2.15}$$

and

$$\dot{g}_2(t) := \nu(X(t))g_2(t), \quad g_2(0) := I, \tag{2.16}$$

referred to as the parameter dynamical systems. In the paper [13], we have already established the following facts which can be generalized to a wider range of applications [12].

Theorem 4 *For any t within the interval of existence, the solutions $X(t)$, $g_1(t)$, and $g_2(t)$ of the systems (2.14), (2.15), and (2.16), respectively, are related to each other by the following three properties:*

1. *(Similarity Property)*
$$X(t) = g_1(t)^{-1} X_0 g_1(t) = g_2(t) X_0 g_2(t)^{-1}. \tag{2.17}$$
2. *(Decomposition Property)*
$$e^{tX_0} = g_1(t)g_2(t). \tag{2.18}$$
3. *(Reversal Property)*
$$e^{tX(t)} = g_2(t)g_1(t). \tag{2.19}$$

There is nothing particular about using the general linear space $\mathfrak{gl}(N)$. These properties remain true over the subalgebra $\mathfrak{g}(\iota\mathscr{H})$. We just need be specific in the choice of μ for the Cartan decomposition.

By rewriting (2.17) as

$$X_0 = g_1(t)X(t)g_1(t)^{-1}, \tag{2.20}$$

we see its analogy to (2.11) if we interpret $X_0 = -\iota\mathscr{H}$ and $g_1(t) = e^{k(t)}$. Our idea is to design an isospectral flow $X(t)$ which is governed by a differential system in the form of (2.14) and is restricted to the subspace $\mathfrak{p}$ such that, starting with $X(0) = -\iota\mathscr{H}$, $X(t)$ converges to an element in $\mathfrak{h}$. In the meantime, the associated flow $g_1(t)$ converges to an element in $e^{\mathfrak{k}}$. In this way, both η and κ needed in the factorization (2.11) are found. Of course, since we already have Pauli strings in place, we do not work with matrices $X(t)$ or $g_1(t)$ directly. Rather, we work with their combination coefficients in the basis of $\mathfrak{g}(\iota\mathscr{H})$.

To keep $X(t) \in \mathfrak{p}$, by the property (2.7), it is necessary that $\mu(X(t)) \in \mathfrak{k}$. The question is how to characterize this map $\mu : \mathfrak{p} \to \mathfrak{k}$ to achieve our goal. In the next section, we shall demonstrate that such a flow can be constructed with guaranteed convergence. A more general mechanism for constructing μ will be detailed in Section 3.2.

2.5. *Example of Lax dynamics*

It might be informative to work out the case $n = 2$ to demonstrate how the above abstract notions can really be pieced together for practical usage. For clarity, we display all data, but keep in mind that the first goal of this work is to avoid producing this entire set of data, only to generate them on an as-needed basis.

There will be $4^n = 16$ Pauli strings. For convenience, we shall identify a Pauli string multiplied by ι by using its ordinal index, i.e., $B_\ell \Rrightarrow \ell$, after some proper enumeration strategy (see Section 3.1). The commutator table of $[B_i, B_j]$ is as follows:

$$\begin{pmatrix}
0 & -15 & 14 & 0 & -12 & 0 & 0 & -9 & 8 & 0 & 0 & 5 & 0 & -3 & 2 & 0 \\
15 & 0 & -13 & 0 & 0 & -12 & 0 & -10 & 0 & 8 & 0 & 6 & 3 & 0 & -1 & 0 \\
-14 & 13 & 0 & 0 & 0 & 0 & -12 & -11 & 0 & 0 & 8 & 7 & -2 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -9 & -10 & -11 & -12 & 5 & 6 & 7 & 8 & 0 & 0 & 0 & 0 \\
12 & 0 & 0 & 9 & 0 & -15 & 14 & 0 & -4 & 0 & 0 & -1 & 0 & -7 & 6 & 0 \\
0 & 12 & 0 & 10 & 15 & 0 & -13 & 0 & 0 & -4 & 0 & -2 & 7 & 0 & -5 & 0 \\
0 & 0 & 12 & 11 & -14 & 13 & 0 & 0 & 0 & 0 & -4 & -3 & -6 & 5 & 0 & 0 \\
9 & 10 & 11 & 12 & 0 & 0 & 0 & 0 & -1 & -2 & -3 & -4 & 0 & 0 & 0 & 0 \\
-8 & 0 & 0 & -5 & 4 & 0 & 0 & 1 & 0 & -15 & 14 & 0 & 0 & -11 & 10 & 0 \\
0 & -8 & 0 & -6 & 0 & 4 & 0 & 2 & 15 & 0 & -13 & 0 & 11 & 0 & -9 & 0 \\
0 & 0 & -8 & -7 & 0 & 0 & 4 & 3 & -14 & 13 & 0 & 0 & -10 & 9 & 0 & 0 \\
-5 & -6 & -7 & -8 & 1 & 2 & 3 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -3 & 2 & 0 & 0 & -7 & 6 & 0 & 0 & -11 & 10 & 0 & 0 & -15 & 14 & 0 \\
3 & 0 & -1 & 0 & 7 & 0 & -5 & 0 & 11 & 0 & -9 & 0 & 15 & 0 & -13 & 0 \\
-2 & 1 & 0 & 0 & -6 & 5 & 0 & 0 & -10 & 9 & 0 & 0 & -14 & 13 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

For example, the $(1,2)$ entry reads that $[B_1, B_2] = -2B_{15}$, where the proportional constant 2 which, by Lemma 1 is universal, is suppressed, but we keep the signs. Suppose now that the target $\iota\mathscr{H}$ is an element in $\mathrm{span}\{3,6,11,15\}$. Then, upon checking through all possible nested commutators, we obtain

$$\mathfrak{g}(\iota\mathscr{H}) = \mathrm{span}\{3,5,6,8,11,15\},$$

indicating that we only need to work with a subspace of dimension 6, a reduction from $\dim(\mathfrak{su}(2^2)) = 15$. We further find that the splitting asserted in Theorem 2 is given by

$$\mathfrak{k} = \mathrm{span}\{5,8\}, \quad \mathfrak{p} = \mathrm{span}\{3,6,11,15\}, \quad \widetilde{\mathfrak{p}} = \mathrm{span}\{3,15\}, \quad \mathfrak{h} = \mathrm{span}\{6,11\}.$$

Express a flow $p(t) \in \mathfrak{p}$ in terms of the basis as

$$p(t) = \alpha_1(t)B_3 + \alpha_2(t)B_{15} + \beta_1(t)B_6 + \beta_2(t)B_{11},$$

where the real-valued scalar functions $\alpha_1(t), \alpha_2(t), \beta_1(t), \beta_2(t)$ are yet to be determined by the Lax dynamics. Define also

$$\mu(p(t)) := \frac{1}{2}\alpha_2(t)B_5 + \frac{1}{2}\alpha_1(t)B_8.$$

The mathematical reason of such a choice of $\mu(p(t))$ will be described in Section 3.2. Note that $\mu(p(t)) \in \mathfrak{k}$. The corresponding Lax dynamics (2.14) for $p(t)$ can be expressed in terms of the coefficients $\alpha_i(t), \beta_i(t)$, $t = 1, 2$. Upon simplification, we find that the coefficients are governed by the differential equations

$$\begin{cases} \dot{\alpha}_1 &= \alpha_1\beta_2, \\ \dot{\alpha}_2 &= \alpha_2\beta_1, \end{cases} \qquad \begin{cases} \dot{\beta}_1 &= -\alpha_2^2, \\ \dot{\beta}_2 &= -\alpha_1^2. \end{cases}$$

By (2.17), $p(t)$ is norm-preserved, so all its coefficients are bounded. Since $\beta_1(t)$ and $\beta_2(t)$ are descending, it follows that $\alpha_1(t)$ and $\alpha_2(t)$ must converge to zero, while $\beta_1(t)$ and $\beta_2(t)$ converge to some values. Thus, $p(t)$ converges to an element η in $\mathfrak{h}$. In the meantime, note that $[B_5, B_8] = 0$. Therefore, if we write $g_1(t) = e^{\gamma_1(t)B_5 + \gamma_2(t)B_8}$, then by using (2.15) we find that

$$\begin{cases} \dot{\gamma}_1 &= \alpha_2, \\ \dot{\gamma}_2 &= \alpha_1, \end{cases}$$

which determines a limit point $\kappa \in \mathfrak{k}$. Since everything is now expressible in terms of implementable Pauli matrices, the unitary matrix $e^{-\iota\mathscr{H}t}$ is ready to be simulated on a quantum circuit.

We stress that to characterize the flow $p(t)$, we only need to work with its scalar combination coefficients. The Pauli strings play only in the interpretation, but not in the calculation. The differential equations for the parameters are nonlinear in general, but can be integrated numerically (this example is simple that an analytic solution can be obtained), and involved only $\dim(\mathfrak{p})$ many variables. It is true in this example, as is true in general, that the vector fields for these coefficient flows are always made of homogeneous polynomials of degree 2 and that the coefficients of monomials in the polynomials are approximately of equal magnitudes, suggesting that the numerical integration is not difficult [40, 50].

3. General framework

In this section, we give more technical details about the general framework. We divide the discussion to subsections to address each of the goals specified earlier.

3.1. *Generating* $\mathfrak{g}(\iota\mathscr{H})$

Even as simple as the case $n = 2$ in the example above, we have seen the benefit of working with the lower dimensional subalgebra $\mathfrak{g}(\iota\mathscr{H})$. Once $\mathfrak{g}(\iota\mathscr{H})$ is known, the splitting (2.7) is an immediate consequence by merely using the involution to separate its eigenspace, e.g., skew-symmetric versus symmetric. Therefore, effectively generating $\mathfrak{g}(\iota\mathscr{H})$ for a given $\mathscr{H}$ is the first imperative task.

To our knowledge, furnishing an *a priori* estimate for the dimension of $\mathfrak{g}(\iota\mathscr{H})$ for a general $\mathscr{H}$ is still an open question. For some specifically structure quantum models, it is sometimes possible to

exploit the structure to give an analytic description of $\mathfrak{g}(\imath\mathscr{H})$. For instance, the Heisenberg model assumes that the n spin-½ particles interacts with only the nearest neighbors. The corresponding Hamiltonian appears in the pattern

$$\mathscr{H} = -\frac{1}{2}\sum_{j=1}^{n-1}(J_X\sigma_j^X * \sigma_{j+1}^X + J_Y\sigma_j^Y * \sigma_{j+1}^Y + J_Z\sigma_j^Z * \sigma_{j+1}^Z + \gamma\sum_{j=1}^{n}\sigma_j^Z), \tag{3.1}$$

where $\sigma_j^A := I^{\otimes j-1}\otimes A\otimes I^{\otimes n-j}$ and J_A is the coupling constant with A standing for any of the Pauli matrices X, Y, or Z, γ denotes an transverse interfering magnetic field, and $*$ denotes the matrix multiplication. In this case, it can be argued that $\dim(\mathfrak{g}(\imath\mathscr{H})) = 4^n - 4$, so there is not much gain in using $\mathfrak{g}(\imath\mathscr{H})$ for the Heisenberg model [35]. This case-by-case closed-form analysis is interesting, but we look for some practical ways to generate $\mathfrak{g}(\imath\mathscr{H})$ for any given general $\mathscr{H}$. We also are aware of some powerful software packages, e.g., Maple's LIEALGEBRA, which are capable of creating Lie algebras from a variety of sources and calculating Cartan decompositions [2]. These are helpful tools for initial investigations, but suffer from the curse of dimensionality. To overcome these issues, we propose to employ some basic tools to cast this construction of $\mathfrak{g}(\imath\mathscr{H})$ as a combinatorial problem.

Denote X,Y,Z,I by the Fraktur numerics $1,2,3,4$, which will be used both as an integer index and as a symbol. Every element in $\{X,Y,Z,I\}^{\otimes n}$ has a unique n-digit ID

$$\mathfrak{d}_1\otimes\mathfrak{d}_2\otimes\ldots\otimes\mathfrak{d}_n \Rrightarrow \mathfrak{d}_1\mathfrak{d}_2\ldots\mathfrak{d}_n, \quad \mathfrak{d}_j\in\{1,2,3,4\},$$

which can be translated into a unique ordinal number ℓ with

$$\ell := \sum_{j=1}^{n} 4^{j-1}(\mathfrak{d}_j - 1) + 1, \tag{3.2}$$

and vice versa. This conversion enables us to identify the Pauli string B_ℓ by the single integer ℓ. We have already used this notion in Section 2.5. As another example, for the case $n = 5$, we have

$$X\otimes Y\otimes Z\otimes I\otimes Z \Rrightarrow 12343 \equiv B_{741}.$$

Our goal is to avoid generating the commutator table for all B_ℓ as we did in Section 2.5. Indeed, we just need the closure under the commutator of terms involved in $\imath\mathscr{H}$. From the relationships (2.5), the matrix-to-matrix multiplication table can be summarized as in Table 1.

TABLE 1 *Matrix-to-matrix multiplication table of* $\{X,Y,Z,I\}$.

$*$	1	2	3	4
1	4	$\imath 3$	$-\imath 2$	1
2	$-\imath 3$	4	$\imath 1$	2
3	$\imath 2$	$-\imath 1$	4	3
4	1	2	3	4

If we write the Pauli string B_i in its n-digit ID as $B_i = \mathfrak{d}_{i1}\ldots\mathfrak{d}_{in}$, where $\mathfrak{d}_{ik}\in\{1,2,3,4\}$ for $k = 1,\ldots,n$, then the commutator $[B_i,B_j]$ is given by

$$\begin{aligned}[B_i,B_j] &= (\mathfrak{d}_{i1}\ldots\mathfrak{d}_{in})*(\mathfrak{d}_{j1}\ldots\mathfrak{d}_{jn})-(\mathfrak{d}_{j1}\ldots\mathfrak{d}_{jn})*(\mathfrak{d}_{i1}\ldots\mathfrak{d}_{in})\\ &= (\mathfrak{d}_{i1}*\mathfrak{d}_{j1})\ldots(\mathfrak{d}_{in}*\mathfrak{d}_{jn})-(\mathfrak{d}_{j1}*\mathfrak{d}_{i1})\ldots(\mathfrak{d}_{jn}*\mathfrak{d}_{in}),\end{aligned}$$

where, for clarity, we have carefully distinguished the matrix multiplication $*$ from the Kronecker product $\otimes$ whose writing is suppressed. Each of the products $\mathfrak{d}_{ik}*\mathfrak{d}_{jk}$, $k=1,\ldots,n$, can be looked up from Table 1. Furthermore, note that $\mathfrak{d}_{ik}*\mathfrak{d}_{jk}$ and $\mathfrak{d}_{jk}*\mathfrak{d}_{ik}$ differ by at most a negative sign, so the two terms on the right side are essentially the same. We have either $[B_i,B_j]=0$ or $[B_i,B_j]=2(\mathfrak{d}_{i1}*\mathfrak{d}_{j1})\ldots(\mathfrak{d}_{in}*\mathfrak{d}_{j1})$. The n-digit ID of the bracket $[B_i,B_j]$ is thus completely determined. There is no need to execute any real matrix or tensor multiplications at all. All we need is just a few index retrievals or swaps. Once we know the terms involved in $\mathscr{H}$, together with Lemma 1, we can generate the subalgebra $\mathfrak{g}(\iota\mathscr{H})$ effectively by just checking the membership through the associated ordinal numbers. The immediate benefit of this approach is that we can handle large n, which has caused the exponential growth in the size of $\mathfrak{su}(2^n)$. A general-purpose code that implements the procedure described above has been implemented for this paper and can be furnished to interested readers.

3.2. *Characterizing* $\mu(p(t))$

The key ingredients in the Cartan decomposition approach for the Hamiltonian simulation of a given $\mathscr{H}$ are the matrices $\kappa\in\mathfrak{k}$ and $\eta\in\mathfrak{h}$ satisfying (2.11). Different from other approaches in the literature that compute only their approximations, we propose using the Lax dynamics to pinpoint the values of κ and η to high precision. We have seen a special example in Section 2.5. Now we describe a general mechanism. We should point out that there is more than one way to design this mechanism. It is yet to be further investigated on whether some might converge faster than others. The following discussions hold for any Cartan decomposition, but we limit our attention to the subalgebra $\mathfrak{g}(\iota\mathscr{H})$.

Lemma 5 *Let* $\mathfrak{g}$ *be a semisimple Lie algebra and* $\mathfrak{g}=\mathfrak{k}\oplus\mathfrak{p}$ *be its Cartan decomposition. Suppose* $\mathfrak{p}=\widetilde{\mathfrak{p}}\oplus\mathfrak{h}$ *with* $\mathfrak{h}$ *as a nonempty Cartan subalgebra. Then* $\dim(\mathfrak{k})=\dim(\widetilde{\mathfrak{p}})$.

Proof: By the stratification relationship in Theorem 3, we see that $\dim(\mathfrak{p})=\dim(\mathfrak{k})+\dim(\mathfrak{h})$. □

We may therefore denote the basis in each subspace by the Pauli strings multiplied by ι as

$$\widetilde{\mathfrak{p}}=\text{span}\{\widetilde{p}_1,\ldots\widetilde{p}_r\};\quad \mathfrak{h}=\text{span}\{h_1,\ldots h_s\};\quad \mathfrak{k}=\text{span}\{k_1,\ldots k_r\},$$

respectively. A flow $p(t)\in\mathfrak{p}$ is necessarily of the form

$$p(t)=\sum_{i=1}^{r}\alpha_i(t)\widetilde{p}_i+\sum_{j=1}^{s}\beta_j(t)h_j, \tag{3.3}$$

where $\alpha_i(t)$ and $\beta_j(t)$ are to be determined. We want the vector field

$$\dot{p}(t)=[p(t),\mu(p(t))] \tag{3.4}$$

to stay inside $\mathfrak{p}$. It is necessary to choose $\mu(p(t))\in\mathfrak{k}$. Express $\mu(p(t))$ in terms of the basis of $\mathfrak{k}$ as

$$\mu(p(t))=\sum_{\ell=1}^{r}z_\ell(t)k_\ell. \tag{3.5}$$

Then, we have two ways to convey the derivative of $\dot{p}(t)$ in the Lax dynamics:

$$\dot{p}(t) = [p(t), \mu(p(t))] = \begin{cases} \sum_{i=1}^{r} \dot{\alpha}_i(t)\widetilde{p}_i + \sum_{j=1}^{s} \dot{\beta}_j(t) h_j, \\ \sum_{\ell=1}^{r}\sum_{i=1}^{r} \alpha_i(t) z_\ell(t)[\widetilde{p}_i, k_\ell] + \sum_{j=1}^{s}\sum_{\ell=1}^{r} \beta_j(t) z_\ell(t)[h_j, k_\ell]. \end{cases} \tag{3.6}$$

Since the Pauli strings are mutually orthogonal, we can obtain the differential equations for $\dot{\alpha}_i$ and $\dot{\beta}_j$ by comparing the two equivalent expressions in (3.6) with respect to the basis. Toward that end, we have to separate the terms in the double summations into those belonging to $\widetilde{\mathfrak{p}}$ and those to $\mathfrak{h}$. We do know that $[\widetilde{p}_i, k_\ell] \in \mathfrak{p}$, but it is hard to tell whether it falls in $\widetilde{\mathfrak{p}}$ or $\mathfrak{h}$. It is the second double summation that provides the clue on how to select z_ℓ, $\ell = 1, \ldots r$, to ensure that the $p(t)$ converges to an element in $\mathfrak{h}$ as t goes to the infinity. The idea is based on the following observation.

Lemma 6 *Suppose that a semisimple Lie algebra $\mathfrak{g}$ has a Cartan decomposition $\mathfrak{g} = \mathfrak{k} \oplus \mathfrak{p}$ and that $\mathfrak{p}$ has a splitting $\mathfrak{p} = \widetilde{\mathfrak{p}} \oplus \mathfrak{h}$ with $\mathfrak{h}$ as a maximal subalgebra. Then for all $h \in \mathfrak{h}$ and $k \in \mathfrak{k}$, it is true that $[h,k] \in \widetilde{\mathfrak{p}}$.*

Proof : Since $\mathfrak{h} \subset \mathfrak{p}$, by (2.7), we can write

$$[h,k] = \widehat{p} + \widehat{h},$$

for some $\widehat{p} \in \widetilde{\mathfrak{p}}$ and $\widehat{h} \in \mathfrak{h}$. It follows that

$$\langle \widehat{h}, \widehat{h} \rangle = \langle hk - kh, \widehat{h} \rangle = \langle k, h^*\widehat{h} - \widehat{h}h^* \rangle = 0,$$

where the last equality is because h and $\widehat{h}$ commute. This proves that $[h,k] = \widehat{p} \in \widetilde{\mathfrak{p}}$. □

For $j = 1, \ldots, s$, define

$$\mathfrak{H}_j := \{\ell | [h_j, k_\ell] \neq 0\}.$$

It should be easy to see that

$$\mathfrak{k} = \bigcup_{j=1}^{s} \mathfrak{H}_j. \tag{3.7}$$

For each $\ell_j \in \mathfrak{H}_j$, by Lemma 6, there exists an integer $1 \leq i_{\ell_j} \leq r$ such that $[h_j, k_{\ell_j}] = c_{i_{\ell_j}} \widetilde{p}_{i_{\ell_j}}$ with $c_{i_{\ell_j}} = \pm 2$. By Lemma 1,

$$[\widetilde{p}_{i_{\ell_j}}, k_{\ell_j}] = -c_{i_{\ell_j}} h_j. \tag{3.8}$$

We now rely on these indices i_{ℓ_j}, $\ell_j \in \mathfrak{H}_j$, $j = 1, \ldots, s$, to define the flow $\mu(p(t))$ in (3.5) segment by segment. Without causing ambiguity, when j is fixed, we shall drop the reference to j in ℓ_j as ℓ.

Lemma 7 *Starting with h_s, for each $\ell \in \mathfrak{H}_s$, choose the integer i_ℓ as above. Define the coefficients*

$$z_\ell(t) := \frac{c_{i_\ell}}{4} \alpha_{i_\ell}(t), \quad \ell \in \mathfrak{H}_s. \tag{3.9}$$

Then

$$\alpha_{i_\ell}(t) \to 0, \quad \textit{for all } \ell \in \mathfrak{H}_s. \tag{3.10}$$

Proof: By the way the integer i_ℓ is chosen and the fact (3.8), we find from (3.6) that

$$\dot{\beta}_s(t) = -\sum_{\ell\in\mathfrak{H}_s} c_{i_\ell}\alpha_{i_\ell}(t)z_\ell(t) = -\sum_{\ell\in\mathfrak{H}_s}\alpha_{i_\ell}^2(t). \tag{3.11}$$

Therefore, $\beta_s(t)$ is strictly decreasing unless $\alpha_{i_\ell}(t)$ becomes zero for all $\ell\in\mathfrak{H}_s$. Since $\beta_s(t)$ is bounded, it must be the case as $t\to\infty$. It also follows that $\beta_s(t)$ converges to a limit point. □

It is possible that there exists some $\ell\notin\mathfrak{H}_s$, i.e., $[h_s,k_\ell]=0$. Then we move to examine the set $\mathfrak{H}_{s-1}$ and for $\ell\in\mathfrak{H}_{s-1}$ define for the remaining $z_\ell(t)$ accordingly. We repeat the process for every $h_j\in\mathfrak{h}$ until every $z_\ell(t)$ is defined. In this way, segment by segment $\alpha_{i_\ell}(t)\to 0$ for $\ell\in\mathfrak{H}_j$ while $\beta_j(t)$ converges to a limit point. It follows that the limit point of $p(t)$, i.e.,

$$\eta := \lim_{t\to\infty} p(t) = \sum_{j=1}^{s}(\lim_{t\to\infty}\beta_j(t))h_j \in \mathfrak{h} \tag{3.12}$$

exists. In this way, we have secured a point in the subalgebra $\mathfrak{h}$ which is isospectral to $-\iota\mathscr{H}$ without knowing the spectrum of $-\iota\mathscr{H}$. Note that we are not trying to diagonalize $-\iota\mathscr{H}$ as a conventional Toda lattice will do. Rather, we use the flow to locate a point $\eta\in\mathfrak{h}$ which is not necessarily diagonal, but is surely quantum implementable. We do think that this innovative application of the Lax dynamics (2.14) is interesting.

Usually the dimension s of the maximal Abelian subalgebra $\mathfrak{h}$ is relatively low. The sign of c_{i_ℓ} can quickly be determined by using the techniques described in Section 3.1. In our preliminary experiment thus far, we have found that this strategy of construction works fairly well.

The reason we start with examining h_s and work backward for other h_ℓ is only for the convenience of explaining our ideas. There is a lot of room for further investigation. For example, is there a preferable ordering, say, starting with the largest set $\mathfrak{H}_j$, that will help improve the convergent behavior? Also, the current choice $z_\ell(t)$ is solely for enforcing the nonnegativity in the summation of (3.11). Will other choices, such as using a high degree odd polynomial in $\alpha_{i_\ell}(t)$, help speed up the convergence? Thus far, by our construction, the vector field involves homogeneous polynomials of degree 2 with coefficients of almost equal module. This particular quadratic form might help justify that the differential system is not stiff. Any further investigation of these questions will likely help improve our framework.

3.3. *Simulating* $e^{k(t)}$

The conjugation action Ad_K in Theorem 3 can be interpreted as a rotation of the subalgebra $\mathfrak{h}$. As such, many approaches proposed in the literature have been resorting to the notion of optimization. One such formulation is of the form [20, 34, 49]

$$\min_{h\in\mathfrak{h},K\in\mathfrak{K}} \|KhK^{-1}-\iota\mathscr{H}\|_F, \tag{3.13}$$

which is a special type of the so-called weighted orthogonal Procrustes problem [14]. Because this is a nonlinear programming problem, there is a great danger of being trapped in a local solution which does not warrant an equality in (2.11). In contrast, our idea outlined above uses the Lax dynamics to construct an isospectral flow that finds the absolute optimal $\eta\in\mathfrak{h}$ without using any information of K. With the state-of-the-art numerical ODE techniques, the flow can be followed closely with precision up to the user-controlled integration tolerance. Now we outline some ideas for finding the corresponding unitary matrix K.

3.3.1. Derivative of exponential map

To maintain the similarity property (2.20), $g_1(t)$ must satisfy the initial value problem (2.15). To fit in Theorem 3, we also want $g_1(t)$ to be of the form $g_1(t) = e^{k(t)}$ for some

$$k(t) = \sum_{\ell=1}^{r} \gamma_\ell(t) k_\ell \in \mathfrak{k}. \tag{3.14}$$

We characterize the coefficients $\gamma_\ell(t)$, $\ell = 1, \ldots, r$, and their limiting behavior as follows.

Theorem 8 *Let $\Theta = [\theta_{\ell j}] \in \mathbb{R}^{r\times r}$ be the matrix representation of the linear map $\mathrm{ad}_{k(t)} : \mathfrak{k} \to \mathfrak{k}$, where $\mathrm{ad}_k(\Delta) := [k, \Delta]$, that is,*

$$\mathrm{ad}_k(k_j) = \sum_{\ell=1}^{r} \theta_{\ell j} k_\ell. \tag{3.15}$$

Then the column vector of coefficients $\gamma := [\gamma_1, \ldots, \gamma_r]^\top$ in (3.14) satisfy the linear system of equations

$$\underbrace{\Big(\sum_{j=0}^{\infty} \frac{(-\Theta)^j}{(j+1)!}\Big)}_{\Omega} \dot{\gamma} = \mathbf{z}, \tag{3.16}$$

where entries of $\mathbf{z} := [z_1, \ldots z_r]^\top$ are given by (3.9).

Proof : Recall the fact that the derivative of exponential map is given by the dexp formula [32]

$$\frac{de^{k(t)}}{dt} = e^{k(t)} \Big(\sum_{j=0}^{\infty} \frac{(-1)^j}{(j+1)!} (\mathrm{ad}_{k(t)})^j\Big) \dot{k}(t). \tag{3.17}$$

Comparing with the differential system (2.15), we find the relationship that

$$\mu(p(t)) \quad = \quad \Big(\sum_{j=0}^{\infty} \frac{(-1)^j}{(j+1)!} (\mathrm{ad}_k)^j\Big) \dot{k}(t) = \dot{k} - \frac{1}{2!}[k, \dot{k}] + \frac{1}{3!}[k, [k, \dot{k}]] - \ldots. \tag{3.18}$$

Introduce a formal row vector $\omega := [k_1, \ldots, k_r]$ of matrices. Then

$$(\mathrm{ad}_k)^j(\dot{k}) \quad = \quad \omega \Theta^j \dot{\gamma}, \quad j = 0, 1, 2, \ldots.$$

Upon comparing the coefficients, we see that (3.18) is equivalent to (3.16). □

The infinite sum can be simplified to

$$\Omega = \begin{cases} I, & \text{if } \Theta \equiv 0, \\ (I - e^{-\Theta})\Theta^{-1}, & \text{if } \Theta \text{ is invertible.} \end{cases} \tag{3.19}$$

Note that Θ depends on $\gamma_1, \ldots, \gamma_r$, so Ω^{-1} in general is a nonlinear function of the variable γ. However, by the fact in (3.16) that $\dot{\gamma}$ depends linearly in $\mathbf{z}$ and $\mathbf{z}(t) \to 0$, we still can conclude that a limit point κ in the sense

$$\kappa := \sum_{\ell=1}^{r} (\lim_{t\to\infty} \gamma_\ell(t)) k_\ell, \tag{3.20}$$

exists and can be calculated. In the case that elements in $\mathfrak{k}$ commute, it is easy to see that $\dot{\gamma}_\ell(t) = z_\ell(t)$, $\ell = 1, \ldots, r$, which is exactly the case $\Theta \equiv 0$.

By now, we have developed two dynamical systems for finding matrices η and κ, respectively, needed for the Cartan decomposition of a given $\mathscr{H}$. Accurate up to the user-specified tolerance, this ODE approach is much more theoretically assuring and computationally economical than solving the optimization problem (3.13).

3.3.2. Example for computing Ω

It might be interesting to demonstrate a different way to compute Ω. Consider the case that $n = 2$ and $\iota\mathscr{H} \in \text{span}\{4, 11, 13, 15\}$. Then,

$$\mathfrak{g}(\iota\mathscr{H}) = \text{span}\{4,5,6,7,9,10,11,13,14,15\}$$

which is larger than the example given in Section 2.5. The splitting asserted in Theorem 2 is given by

$$\mathfrak{k} = \text{span}\{5,7,10,14\}, \quad \mathfrak{p} = \text{span}\{4,6,9,11,13,15\}, \quad \widetilde{\mathfrak{p}} = \text{span}\{4,6,11,15\}, \quad \mathfrak{h} = \text{span}\{9,13\}.$$

Note that, in contrast to the example in Section 2.5, elements in $\mathfrak{k}$ do not commute this time. Following (3.15), we obtain that

$$-\Theta = 2\begin{bmatrix} 0 & \gamma_4 & 0 & -\gamma_2 \\ -\gamma_4 & 0 & 0 & \gamma_1 \\ 0 & 0 & 0 & 0 \\ \gamma_2 & -\gamma_1 & 0 & 0 \end{bmatrix},$$

which is always skew-symmetric. Note that $\text{span}\{5,7,14\}$ is itself the maximal proper subalgebra of $\mathfrak{k}$, while the singleton B_{10} commutes with every element in $\mathfrak{k}$. It follows that

$$\begin{aligned} (-\Theta)^2 &= 4\begin{bmatrix} -\gamma_2{}^2 - \gamma_4{}^2 & \gamma_1\gamma_2 & 0 & \gamma_1\gamma_4 \\ \gamma_1\gamma_2 & -\gamma_1{}^2 - \gamma_4{}^2 & 0 & \gamma_2\gamma_4 \\ 0 & 0 & 0 & 0 \\ \gamma_1\gamma_4 & \gamma_2\gamma_4 & 0 & -\gamma_1{}^2 - \gamma_2{}^2 \end{bmatrix}, \\ (-\Theta)^3 &= -4\rho(-\Theta), \end{aligned}$$

with $\rho := \gamma_1^2 + \gamma_2^2 + \gamma_4^2$. This establishes the recursion relationship that

$$\begin{aligned} (-\Theta)^{2j+1} &= -4\rho(-\Theta)^{2j-1} = \ldots = (-4\rho)^j(-\Theta), \\ (-\Theta)^{2j+2} &= -4\rho(-\Theta)^{2j} = \ldots = (-4\rho)^j(-\Theta)^2, \quad j = 0,1,2,\ldots. \end{aligned}$$

Therefore, the coefficient matrix Ω in (3.16) is given by

$$\begin{aligned} \Omega = \sum_{j=0}^{\infty} \frac{(-\Theta)^j}{(j+1)!} &= I + \sum_{j=0}^{\infty} \frac{(-\Theta)^{2j+1}}{(2j+2)!} + \sum_{j=0}^{\infty} \frac{(-\Theta)^{2j+2}}{(2j+3)!} \\ &= I - (\sum_{j=0}^{\infty} \frac{(-4\rho)^j}{(2j+2)!})\Theta + (\sum_{j=0}^{\infty} \frac{(-4\rho)^j}{(2j+3)!})\Theta^2 \\ &= I - \frac{1-\cos\left(2\sqrt{\rho}\right)}{4\rho}\Theta + \frac{2\sqrt{\rho} - \sin\left(2\sqrt{\rho}\right)}{8\rho\sqrt{\rho}}\Theta^2. \end{aligned}$$

It can be checked that determinant is

$$\det(\Omega) = \frac{1-\cos\left(2\sqrt{\rho}\right)}{2\rho}.$$

Therefore, unless $\sqrt{\rho}$ is an integer multiple of π, the linear system (3.16) is uniquely solvable. In fact, it is not difficult to prove that the recursion relationship exemplified above holds for any subalgebra of dimension three.

3.3.3. Block diagonal Θ

The above example actually sheds light on another way to facilitate the calculation of Θ. Suppose that the subalgebra $\mathfrak{k}$ can be partitioned as

$$\mathfrak{k} = (\bigoplus_{i=1}^{s} \mathfrak{k}_i) \oplus \mathfrak{r}, \tag{3.21}$$

where each $\mathfrak{k}_i$ is a maximal, proper and non-commutative subalgebra of $\mathfrak{k}$ and $\mathfrak{r}$ is the remainder. It is possible that $\mathfrak{k}_i$ or $\mathfrak{r}$ is empty. By the definition, it must be that

$$[\mathfrak{r},\mathfrak{r}] = 0,\ [\mathfrak{k}_i,\mathfrak{r}] = 0 \text{ for each } i, \text{ and } [\mathfrak{k}_i,\mathfrak{k}_j] = 0 \text{ for all } i \neq j. \tag{3.22}$$

Therefore, upon rearranging the basis if necessary, we may assume without loss of generality that Θ is a block diagonal matrix. In fact, the block corresponds to $\mathfrak{r}$ is identically zero. Write

$$\Theta = \operatorname{diag}\{\Theta_1,\ldots,\Theta_s,0\}. \tag{3.23}$$

Then the calculation of Ω in (3.19) can be limited to each Θ_i, $i = 1,\ldots,s$. Correspondingly, if we group the terms involved in $k(t)$ defined in (3.14) according to

$$k(t) = \sum_{i=1}^{s} \sum_{k_{\ell_i}\in\mathfrak{k}_i} \gamma_{\ell_i}(t) k_{\ell_i} + \sum_{k_{r_j}\in\mathfrak{r}} \gamma_{r_j}(t) k_{r_j}, \tag{3.24}$$

then each of these groups $\{\dot{\gamma}_{\ell_i}\}$, $k_{\ell_i} \in \mathfrak{k}_i$, and $\{\gamma_{r_j}\}$, $k_{r_j} \in \mathfrak{r}$, can be calculated by using the corresponding blocks in Θ.

4. Continued Cartan decomposition

Our initial goal is to establish the relationship

$$-\iota\mathscr{H} \equiv e^{k(t)} p(t) e^{-k(t)} \to e^{\kappa}\eta e^{-\kappa} \quad \text{as } t\to\infty. \tag{4.1}$$

Because elements in $\mathfrak{h}$ commute, the summation involved in η through the definition (3.12) can be handled directly by the mixture of quantum gates. The summation involved in κ according to (3.20), nonetheless, makes the quantum simulation of e^{κ} as hard as that of $e^{-\iota\mathscr{H}t}$. The reason is that, again, we can handle the exponential of each individual term k_ℓ on a quantum circuit, but not an algebraic combination such as the whole κ. Therefore, it is necessary to further decompose the subalgebra $\mathfrak{k}$, which leads to the notion of continued Cartan decomposition.

Since $\mathfrak{k}$ forms a subalgebra by itself, one possible approach is to apply the same theory we have developed for $\mathfrak{g}(\iota\mathscr{H})$ to the subalgebra $\mathfrak{k}$, but with a different involution. The change of the involution

is necessary because $\mathfrak{k}$ is already invariant under the old involution θ_1 used in (2.9). A new involution will have the effect of defining a new notion of symmetry and skew-symmetry. The point is to continue breaking down e^{κ} to the product form (2.12) by the process we have just described for decomposing $e^{-\iota\mathscr{H}t}$ until we have just singletons of Pauli strings in hand. By then, a quantum circuit can be constructed.

This continued Cartan decomposition is of great mathematical interest in itself. It is similar to the classical notion that every element in the special unitary group $\mathrm{SU}(2^n)$ can be recursively factorized [18, 22, 34] by using, say, the elementary Givens rotations [17]. See also [36, 44] for the idea of recursively applying generalized polar decompositions in such a way that any element of the Lie algebra is written as a sum of elements whose corresponding exponentials are easier to compute. However, there are two fundamental differences in practicality. First, we limit our decomposition to Pauli strings only which are readily quantum implementable. Second, we conduct the continued decomposition on subalgebras contained in $\mathfrak{g}(\iota\mathscr{H})$ and each time we work on some smaller subalgebras. This successive reduction of sizes is analogous to the idea of divide-and-conquer used in, say, the Fast Fourier Transform. For example, in Section 2.5 we see the case of a quick reduction from $\dim(\mathfrak{su}(2^2)) = 15$ to $\dim(\mathfrak{g}(\iota\mathscr{H}) = 6$, and then to $\dim(\mathfrak{k}) = 2$ and $\dim(\mathfrak{h}) = 2$. A theoretical understanding on how quickly the dimensions are reduced through this nested decomposition for $\mathfrak{su}(2^n)$ where n is large is a question of high interest, but we do not have room to address it in this paper. It is worth remarking that ultimately the factors in the decomposition consists of Pauli strings within the subalgebra $\mathfrak{g}(\iota\mathscr{H})$ only.

4.1. *Choice of involution*

The beauty of the Cartan decomposition is its generality – that the splitting in Theorems 2 exists (but varies) for any given involution and that the conjugation in Theorem 3 holds for any Cartan decomposition. Furthermore, our Lax dynamics relies only on the inclusion relationship (2.7). Other than the fact that the basis is made of Pauli strings which are readily quantum implementable, the innate structure of $\mathfrak{k}$ and $\mathfrak{p}$ is fundamentally immaterial. Therefore, per given involution θ, and with the resulting $\mathfrak{k}$ and $\mathfrak{h}$, we can develop a corresponding Lax dynamical system with the property that $p(t) \to \eta \in \mathfrak{h}$ and $k(t) \to \kappa \in \mathfrak{k}$. What we have demonstrated in Sections 3.2 and 3.3 represents only one choice of θ in the form (2.9), and shows the feasibility and applicability of our ideas.

Suppose that $\varphi : \mathfrak{su}(2^n) \to \mathfrak{su}(2^n)$ is an automorphism. Then

$$\theta_2 = \varphi\theta_1\varphi^{-1}. \tag{4.2}$$

defines a new involution with respect to which the notion of skew-symmetry and symmetry will have a new meaning [19, Sec. 2.2.2]. That is,

$$\theta_2(g) = \pm g \Longleftrightarrow \varphi\theta_1(g) = \pm g\varphi \Longleftrightarrow g^{\top} = \mp\varphi^{-1}g\varphi \Longleftrightarrow \begin{cases} \varphi\text{-skew-symmetry,} \\ \varphi\text{-symmetry.} \end{cases} \tag{4.3}$$

This idea can be applied to $\mathfrak{g}(\iota\mathscr{H})$ as well as to the Lie subalgebra $\mathfrak{k}$ of $\mathfrak{g}(\iota\mathscr{H})$.

4.2. *Example of continued Cartan decomposition*

We work out an example in details to illuminate the essential steps. Consider the case $n = 3$, so $\{X,Y,Z,I\}^{\otimes 3}$ has 64 Pauli strings. Suppose $\iota\mathscr{H} \in \mathrm{span}\{1,4,6,7,11,12,13\}$. The subalgebra

$$\mathfrak{g}(\iota\mathscr{H}) = \mathrm{span}\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63\}$$

is of dimension 30 only. Using θ_1, we find that

$$\begin{aligned}\mathfrak{k} &= \text{span}\{2,5,7,8,10,14,50,53,55,56,58,62\},\\ \mathfrak{p} &= \text{span}\{1,3,4,6,9,11,12,13,15,49,51,52,54,57,59,60,61,63\}.\end{aligned}$$

There are actually five Cartan subalgebras within $\mathfrak{p}$, spanned by the bases

$$\{1,4,13,49,52,61\},\{3,4,15,51,52,63\},\{3,6,9,51,54,57\},\{1,6,11,49,54,59\},\{11,12,15,59,60,63\},$$

respectively. By Theorem 3, each can be reached by some suitable conjugations of others. We can choose any as $\mathfrak{h}$ for the Lax dynamics to construction the decomposition (4.1). Depending on which subalgebra is used, the resulting limit point of the dynamics will be different. What matters is that the exponentials of limit points of all choices are quantum implementable. We conjecture that the answer is negative, but it will be interesting to investigate whether one particular selection is computationally superior to the other.

Suppose that η and κ have been found. The emphasis in this example is to further decompose e^{κ} to make it quantum implementable. We find that $\mathfrak{k}$ contains two maximal, proper and non-commutative subalgebras

$$\mathfrak{k}_1 = \text{span}\{2,8,10,50,56,58\}, \quad \mathfrak{k}_2 = \text{span}\{5,7,14,53,55,62\}.$$

Therefore, by writing $\kappa = \kappa_1 + \kappa_2$ with $\kappa_i \in \mathfrak{k}_i$, $i = 1,2$, we already have

$$e^{\kappa} = e^{\kappa_1} e^{\kappa_2}. \tag{4.4}$$

It suffices to work with each subalgebra separately. Suppose that we use the automorphism

$$\varphi := \begin{bmatrix} I_4 & 0 \\ 0 & -I_4 \end{bmatrix},$$

where I_4 stands for the 4×4 identity matrix, to define θ_2. Based on Theorem 2, we find new Cartan decompositions

$$\mathfrak{k}_1 = \underbrace{\text{span}\{8,56\}}_{\mathfrak{k}^{(1)}} \oplus \underbrace{\text{span}\{10,58\}}_{\widetilde{\mathfrak{p}}^{(1)}} \oplus \underbrace{\text{span}\{2,50\}}_{\mathfrak{h}^{(1)}}, \quad \mathfrak{k}_2 = \underbrace{\text{span}\{7,55\}}_{\mathfrak{k}^{(2)}} \oplus \underbrace{\text{span}\{14,62\}}_{\widetilde{\mathfrak{p}}^{(2)}} \oplus \underbrace{\text{span}\{5,53\}}_{\mathfrak{h}^{(2)}},$$

where $\mathfrak{k}^{(i)}$, $\mathfrak{p}^{(i)} := \widetilde{\mathfrak{p}}^{(i)} \oplus \mathfrak{h}^{(i)}$, $i = 1,2$, correspond to the decompositions in (2.6) and (2.8) for the subalgebras $\mathfrak{k}_1$ and $\mathfrak{k}_2$, respectively. It turns out that each of these subsets commutes with itself. Additionally, they enjoy these transitive relations through the Lie bracket:

$$[\mathfrak{k}^{(i)}, \widetilde{\mathfrak{p}}^{(i)}] \subseteq \mathfrak{h}^{(i)}, \quad [\widetilde{\mathfrak{p}}^{(i)}, \mathfrak{h}^{(i)}] \subseteq \mathfrak{k}^{(i)}, \quad [\mathfrak{h}^{(i)}, \mathfrak{k}^{(i)}] \subseteq \widetilde{\mathfrak{p}}^{(i)}, \quad i = 1,2. \tag{4.5}$$

Even so, we still cannot apply Theorem 3 to decompose $\kappa_i \in \mathfrak{k}_i$ immediately because κ_i might not reside entirely in $\mathfrak{p}^{(i)}$, i.e., some components from $\mathfrak{k}^{(i)}$ might contribute to the makeup of κ_i. To remedy this, we modify the Lax dynamics (3.4) as follows.

We shall concentrate on the subalgebra $\mathfrak{k}_1$ only. A similar argument applies to $\mathfrak{k}_2$. Consider the solution flow $x(t) \in \mathfrak{k}_1$ to the initial value problem

$$\dot{x}(t) = [x(t), \nu(x(t))], \quad x(0) = \kappa_1, \tag{4.6}$$

and the flow $\varepsilon(t) \in e^{\mathfrak{k}^{(1)}}$ defined by

$$\dot{\varepsilon}(t) = \varepsilon(t)\nu(x(t)), \quad \varepsilon(0) = I, \tag{4.7}$$

with

$$\nu(x(t)) = \zeta_1(t)B_8 + \zeta_2(t)B_{56} \in \mathfrak{k}^{(1)}$$

whose coefficients $\zeta_1(t)$ and $\zeta_2(t)$ are to be determined. By Theorem 4, the similarity relationship

$$\kappa_1 = \varepsilon(t)x(t)\varepsilon(t)^{-1}$$

still holds. Since know only that $x(t) \in \mathfrak{k}_1$, we write

$$x(t) = \alpha_1(t)B_{10} + \alpha_2(t)B_{58} + \beta_1(t)B_2 + \beta_2(t)B_{50} + \delta_1(t)B_8 + \delta_2(t)B_{56}.$$

Upon comparison, the coefficients of $x(t)$ must satisfy the differential equations

$$\left\{ \begin{array}{rcl} \dot{\alpha}_1 &=& -2\beta_1\zeta_2 - 2\beta_2\zeta_1, \\ \dot{\alpha}_2 &=& -2\beta_1\zeta_1 - 2\beta_2\zeta_2, \end{array} \right. \quad \left\{ \begin{array}{rcl} \dot{\beta}_1 &=& 2\alpha_1\zeta_2 + 2\alpha_2\zeta_1, \\ \dot{\beta}_2 &=& 2\alpha_1\zeta_1 + 2\alpha_2\zeta_2. \end{array} \right. \quad \left\{ \begin{array}{rcl} \dot{\delta}_1 &=& 0, \\ \dot{\delta}_2 &=& 0. \end{array} \right.$$

Therefore, if we choose

$$\left\{ \begin{array}{rcl} \zeta_1 &:=& -\alpha_1, \\ \zeta_2 &:=& -\alpha_2, \end{array} \right.$$

then, by the boundedness of $\beta_2(t)$, it must be such that $\alpha_1(t)$ and $\alpha_2(t)$ converge to zero. That is, the components of $x(t)$ involving $\widetilde{\mathfrak{p}}^{(1)}$ are gradually diminished. Together with the fact that $\beta_1(t)$ and $\beta_2(t)$ converge to some fixed values, we conclude that $x(t)$ converges to a point $\widetilde{\kappa}_1 \in \mathfrak{h}^{(1)} \oplus \mathfrak{k}^{(1)}$. By the fact that $\delta_1(t)$ and $\delta_2(t)$ stay invariant, we can write $\widetilde{\kappa}_1 := \eta^{(1)} + \mathscr{P}_{\mathfrak{k}^{(1)}}\kappa_1$ for some $\eta^{(1)} \in \mathfrak{h}^{(1)}$ and $\mathscr{P}_{\mathfrak{k}^{(1)}}\kappa_1$ is merely the projection of κ_1 onto $\mathfrak{k}^{(1)}$. Furthermore, if we write

$$\varepsilon(t) = e^{\gamma_1(t)B_8 + \gamma_2(t)B_{56}},$$

then by the commutativity of B_8 and B_{56} we have

$$\left\{ \begin{array}{rcl} \dot{\gamma}_1 &=& \zeta_1, \\ \dot{\gamma}_2 &:=& \zeta_2, \end{array} \right.$$

and $\varepsilon(t)$ converges to a limit point, say, $e^{\widehat{\kappa}_1}$. By now, we see that

$$e^{\kappa_1} = e^{\gamma_1(t)B_8 + \gamma_2(t)B_{56}} e^{x(t)} e^{-\gamma_1(t)B_8 - \gamma_2(t)B_{56}} \to e^{\widehat{\kappa}_1} e^{\widetilde{\kappa}_1} e^{-\widehat{\kappa}_1}. \tag{4.8}$$

The accomplishment we have advanced in (4.8) is that the matrix exponential e^{κ_1} has been transformed into the product of three matrix exponentials, in which $e^{\widehat{\kappa}_1}$ and its inverse are readily quantum

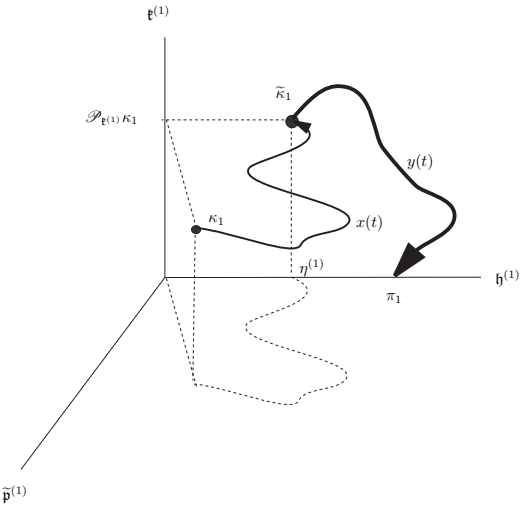

FIG. 1. Continued Cartan decomposition, where the initial point κ_1 is first transited by the path $x(t) \in \mathfrak{k}_1$ (thin curve) to $\widetilde{\kappa}_1$, and then transited by the path $y(t) \in \mathfrak{k}^{(1)} \oplus \mathfrak{h}^{(1)}$ (thick curve) to $\pi_1 \in \mathfrak{h}^{(1)}$.

implementable and $\widetilde{\kappa}_1 \in \mathfrak{h}^{(1)} \oplus \mathfrak{k}^{(1)}$. This dynamics is depicted by the black curve in the diagram sketched in Figure 1. We still need to decompose $e^{\widetilde{\kappa}_1}$.

To deal with $e^{\widetilde{\kappa}_1}$, we exploit the transitive properties (4.5) by restricting a third flow $y(t)$ to the subspace $\mathfrak{h}^{(1)} \oplus \mathfrak{k}^{(1)}$. Define the Lax dynamics via

$$\dot{y}(t) = [y(t), \rho(y(t))], \quad y(0) = \widetilde{\kappa}_1, \tag{4.9}$$

and the flow $\varepsilon(t) \in e^{\widetilde{\mathfrak{p}}^{(1)}}$ via

$$\dot{\varepsilon}(t) = \varepsilon(t)\rho(y(t)), \quad \varepsilon(0) = I, \tag{4.10}$$

with

$$\rho(y(t)) = \xi_1(t)B_{10} + \xi_2(t)B_{58} \in \widetilde{\mathfrak{p}}^{(1)}$$

By going through the same argument as that for (4.8), we can choose $\xi_1(t)$ and $\xi_2(t)$ properly, and conclude finally that there exist matrices $\varsigma_1 \in \mathfrak{k}^{(1)}$ and $\pi_1 \in \mathfrak{h}^{(1)}$ such that

$$e^{\kappa_1} \to e^{\widehat{\kappa}_1} e^{\widetilde{\kappa}_1} e^{-\widehat{\kappa}_1} \to e^{\widehat{\kappa}_1} e^{\varsigma_1} e^{\pi_1} e^{-\varsigma_1} e^{-\widehat{\kappa}_1}. \tag{4.11}$$

By now, every matrix exponential is quantum implementable.

The details above exemplify the feasibility of continued Cartan decomposition. It is not a coincidence that our idea works in this example. The general notion is that the Lax dynamics can be used to "annihilate" an undesired portion in the sum of Pauli strings by converting it to a "partial" spectral decomposition, even under the situation such as (4.6) where $x(t)$ does not fit Theorem 3 exactly. By executing this annihilation procedure systematically we can break down the unitary matrix (1.1) to make it implementable on a quantum machine. Different from the classical Givens rotations that annihilate one entry of the matrix at a time, our method annihilate a segment of subspace at a time. To analyze all procedures rigorously to establish a comprehensive theory for this continued factorization, including the effect of different choices of involutions, will have to be discussed in a separate paper.

5. Applications to general or k-local Hamiltonians

Using the involution $\theta_1(g) = -g^\top$ to perform the Cartan decomposition results in the structure that $\mathfrak{k}$ and $\mathfrak{p}$ contain skew-symmetric and symmetric matrices, respectively. If the given Hamiltonian $\mathscr{H}$ happens to be real and symmetric (as is the case in the Schrödinger equation), then $\imath\mathscr{H} \in \mathfrak{p}$ and Theorem 3 fits in perfectly. It become curious to explore other scenarios such as $\mathscr{H}$ is a general Hermitian matrix, but not symmetric, or $\mathscr{H}$ is symmetric, but sparse, e.g., the Heisenberg model or k-local Hamiltonian. The first scenario is perhaps merely a mathematical inquiry, but the other models have practical significance because in most quantum systems the interaction is only with neighboring particles. We briefly outline some ideas about how the above theory could be modified to tackle these scenarios. A thorough general theory is still under development.

The example discussed in Section 4.2 with the dynamical system (4.6) is actually a case where the flow $x(t)$ and the target matrix κ_1 are not symmetric. We have demonstrated in (4.11) that a sequence of Lax dynamical systems can still be exploited to factorize the matrix exponential e^{κ_1}. The key to success is the same notion of annihilation we have mentioned earlier – After choosing some properly selected involutions to divide the subalgebra $\mathfrak{g}(\kappa_1)$ into three parts, $\mathfrak{k}^{(1)}$, $\widetilde{\mathfrak{p}}^{(1)}$, and $\mathfrak{h}^{(1)}$, use $\mathfrak{k}^{(1)}$ to construct a dynamics to annihilate the components of $\widetilde{\mathfrak{p}}^{(1)}$ in $x(t)$ while maintaining the conjugation relationship (2.17). If necessary, repeat this procedure multiple times to annihilate other components such as that demonstrated in Figure 1. The idea has been applied successfully in several other settings [10, 11, 13].

With regard to the case that $\mathscr{H}$ is sparse, observe that in our representation of $\mathfrak{g}(\imath\mathscr{H})$ we identify each basis matrix by an integer ID, and never physically generate the matrices. Because we only work with the coefficients of the basis, i.e., $\alpha_i(t)$, $\beta_j(t)$ and so on, the specific structure of the underlying Pauli strings is not essential at all. We do think that the above-mentioned approach is applicable, regardless of the sparsity of $\mathscr{H}$.

The idea of using dynamical systems to prepare the Hamiltonian simulation on a quantum machine is still at a rudimentary stage. Any further investigation to see how our theory can be generalized will be interesting. It will be a useful tool for quantum simulation, if the procedure that can be automated to generate the decomposition for arbitrarily given Hamiltonian $\mathscr{H}$.

6. Example of large scale problem in small breadth calculation

In this section, we apply the algorithm described in this paper to a 12-qubit Hamiltonian simulation problem which, if done on a conventional machine, will amount to a large scale problem. By this example, we wish to demonstrate the efficiency of our algorithm for achieving the Cartan decomposition via checking a few integer arrays. We can make our code available for interested readers.

Consider the Heisenberg model (3.1) with $n = 12$ and $J_Z = 0$. The corresponding Hamiltonian $\mathscr{H}$ therefore consists of 34 terms in its summation. The underlying Lie algebra $\mathfrak{su}(2^{12})$ has real dimension $4^{12} - 1 = 16777215$. Each basis element B_ℓ in the Lie algebra is of size $2^{12} \times 2^{12} = 4096 \times 4096$. However, as we have argued in Section 3, it suffices to represent each B_ℓ by a 12-digit ID which is further translated via (3.2) into a unique ordinal number ℓ. Since we can easily convert ℓ back to its 12-digit ID, this single integer ℓ provides every bit information of B_ℓ. There is no need to generate the commutator table of all $[B_i, B_j]$ in its entirety, which would be an enormous matrix of size $(4^{12} - 1) \times (4^{12} - 1)$. Instead, using the technique described in Section 3.1, we can quickly find the basis of the subalgebra $\mathfrak{g}(\imath\mathscr{H})$ and, indeed, $\dim(\mathfrak{g}(\imath\mathscr{H})) = 276$ which is less than 0.00165% of the dimension of $\mathfrak{su}(2^{12})$. This is a significant reduction of the work when comparing with the conventional approaches for the unitary synthesis of $\imath\mathscr{H}$. In Figure 2, we observer a fractal pattern in the commutator table of $\mathfrak{g}(\imath\mathscr{H})$.

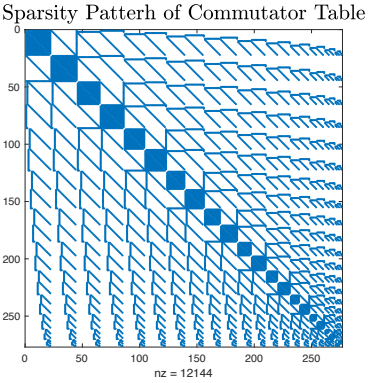

FIG. 2. Nonzero Lie brackets among elements of $\mathfrak{g}(\iota\mathscr{H})$.

Using the fact that $(A\otimes B)^\top = A^\top \otimes B^\top$, we can also effectively determine whether an element $g\in\mathfrak{g}(\iota\mathscr{H})$ is symmetric or skew-symmetric with respect to the involution θ_1 in (2.9) by simply counting how many 2's, i.e., the Pauli matrix Y, are in its 12-digit ID. This task is almost trivial. As a result, we find the Cartan decomposition (2.10) of $\mathfrak{g}(\iota\mathscr{H})$ with $\dim(\mathfrak{k}) = \dim(\widetilde{\mathfrak{p}}) = 132$ and $\dim(\mathfrak{h}) = 12$. It is interesting to note that among the 144 basis elements of $\mathfrak{p}$, there are 111 maximal subalgebras, each of dimension 12, that can be used to form $\mathfrak{h}$. The differential system (3.4) proposed in Section 3.2 defines a 144-dimensional solution flow $p(t)$ which is guaranteed to converge to a point η in the 12-dimensional subalgebra $\mathfrak{h}$. Note that we never form $p(t)$ per se, but merely work on its 144 real-valued coefficients. Even so, we can control the precision of η via the numerical ODE integrator.

To calculate $k(t)$, we use the technique described in Section 3.3 by computing Θ which can be obtained from the the relationship (3.15) over $\mathfrak{k}$, whereas the commutators can be obtained by looking up from Table 1. It turns out that the algebra $\mathfrak{k}$ also contains rich structure therein. First, using our index searching technique, we find that $\mathfrak{k}$ contains 85 subalgebras, each of which is of dimension 12. Of course, these subalgebras overlap. Second, there are only two maximal, proper and non-commutative subalgebras each of which is of dimension 66. We can exploit the block diagonal structure of Θ. In this way, we formulate a flow $k(t)$ from the differential equation (3.16), which leads to a limit point κ in the sense of (3.20). By now, we have achieve the decomposition (4.1). We can write $\kappa = \kappa_1 + \kappa_2$ with $\kappa_i \in \mathfrak{k}_i$ and exact information of $\mathfrak{k}_i$, $i = 1,2$.

In fact, similar to $\mathfrak{k}$, each $\mathfrak{k}_i$ contains 43 subalgebras, each of which is of dimension 6 and is commutative, suggesting also its rich structure. By choosing appropriate involutions (4.2), we apply the continued Cartan decomposition, as is described in Section 4, to each κ_i, $i = 1,2$. It might be necessary to repeat the continued Cartan decomposition multiple times. There is a lot of open questions about choosing the involutions, including how the choice affects the dimensionality in the resulting Cartan decomposition and how many times are needed to carry out the continued Cartan decomposition. Ultimately, the unitary matrix $e^{-\iota\mathscr{H}}$ is decomposed as the product of pieces each of which is implementable on a quantum machine. A lengthy detail of the computational data might not be suitable for this presentation, but we stress that other than solving the differential systems that requires using existing numerical integrators, most of the bookkeeping tasks are handled by using our system of ordinal numbers.

7. Conclusion

The computational power required to describe a quantum system scales exponentially with the number of its constituents. To merely describe the most general (pure) quantum state for n spin-½ particles, we have to store 2^n coefficients, not to mention that we will have to operate on these coefficients in order to productively describe a physical quantity. Quantum simulation promises to use a number of qubits similar to the number of particles in the original system to mimic the physical phenomenon. The overall time needed for the simulation would not grow exponentially with the number of particles. The essential task for this simulation is to approximate each time step by a sequence of quantum gates.

Most existing methods for unitary synthesis rely on approximation, some of which are elegant in theory but suffer from truncation errors or being only a local solution. In contrast, we develop a framework which calls for an eclectic mix of techniques from the Cartan decomposition and Lax dynamics, together with a combinatorial type representation of Pauli strings. Our techniques prepare numerically the essential components needed to synthesize the final unitary operator effectively and more precisely. In contrast to other existing methods, the most important attribute of our approach is that we can quantify the exact factors up to integration error which is easy to control by existing numerical techniques. This combination of techniques is innovative. The expected result should be useful to the field.

Acknowledgement

The author wishes to thank Thomas Steckmann and the anonymous referees for bringing to his attention the report [35] and the papers [32, 44], respectively.

Funding

This research was supported in part by the National Science Foundation under grant DMS-1912816.

References

1. S. Aaronson. *Quantum computing since Democritus*. Cambridge University Press, Cambridge, 2013.
2. I. M. Anderson. Cartan involutions and cartan decompositions of a semi-simple Lie algebra. *Tutorials on... in 1 hour or less*, Paper 5, 2005.
3. C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70:1895–1899, Mar 1993.
4. D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Phys. Rev. Lett.*, 114:090502, Mar 2015.
5. F. Casas, A. Murua, and M. Nadinic. Efficient computation of the Zassenhaus formula. *Comput. Phys. Commun.*, 183(11):2386–2391, 2012.
6. M. T. Chu. The generalized Toda flow, the QR algorithm and the center manifold theory. *SIAM J. Algebraic Discrete Methods*, 5(2):187–201, 1984.
7. M. T. Chu. Asymptotic analysis of Toda lattice on diagonalizable matrices. *Nonlinear Anal.*, 9(2):193–201, 1985.
8. M. T. Chu. A differential equation approach to the singular value decomposition of bidiagonal matrices. *Linear Algebra Appl.*, 80:71–79, 1986.
9. M. T. Chu. Matrix differential equations: a continuous realization process for linear algebra problems. *Nonlinear Anal.*, 18(12):1125–1146, 1992.

10. M. T. Chu. A list of matrix flows with applications. In *Hamiltonian and gradient flows, algorithms and control*, volume 3 of *Fields Inst. Commun.*, pages 87–97. Amer. Math. Soc., Providence, RI, 1994.
11. M. T. Chu. Linear algebra algorithms as dynamical systems. *Acta Numer.*, 17:1–86, 2008.
12. M. T. Chu and G. H. Golub. *Inverse eigenvalue problems: theory, algorithms, and applications*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2005.
13. M. T. Chu and L. K. Norris. Isospectral flows and abstract matrix factorizations. *SIAM J. Numer. Anal.*, 25(6):1383–1391, 1988.
14. M. T. Chu and N. T. Trendafilov. On a differential equation approach to the weighted orthogonal Procrustes problem. *Statistics and Computing*, 8(2):125–133, 1998.
15. J. I. Cirac and P. Zoller. Goals and opportunities in quantum simulation. *Nature Physics*, 8(4):264–266, Apr. 2012.
16. L. Clinton, J. Bausch, and T. Cubitt. Hamiltonian simulation algorithms for near-term quantum hardware. *Nature Communications*, 12(1):4989, 2021.
17. G. Cybenko. Reducing quantum computations to elementary unitary operations. *Computing in Science Engineering*, 3(2):27–32, 2001.
18. M. Dağlı, D. D'Alessandro, and J. D. H. Smith. A general framework for recursive decompositions of unitary quantum evolutions. *Journal of Physics A: Mathematical and Theoretical*, 41(15):155302, 2008.
19. M. Dağli. *Lie algebra decompositions with applications to quantum dynamics*. PhD dissertation, Iowa State University, 2008. Retrospective Theses and Dissertations, 15721.
20. T. G. de Brugière, M. Baboulin, B. Valiron, and C. Allouche. Synthesizing quantum circuits via numerical optimization. In *Lecture Notes in Computer Science*, pages 3–16. Springer International Publishing, 2019.
21. D. P. DiVincenzo. Quantum computation. *Science*, 270(5234):255–261, 1995.
22. B. Drury and P. Love. Constructive quantum Shannon decomposition from Cartan involutions. *Journal of Physics A: Mathematical and Theoretical*, 41(39):395305, sep 2008.
23. A. K. Ekert. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.*, 67:661–663, Aug 1991.
24. R. P. Feynman. Simulating physics with computers. *Internat. J. Theoret. Phys.*, 21(6-7):467–488, 1981/82. Physics of computation, Part II (Dedham, Mass., 1981).
25. A. Giambruno, D. La Mattina, and V. M. Petrogradsky. Matrix algebras of polynomial codimension growth. *Israel J. Math.*, 158:367–378, 2007.
26. O. Gühne and G. Tóth. Entanglement detection. *Phys. Rep.*, 474(1-6):1–75, 2009.
27. J. Haah, M. B. Hastings, R. Kothari, and G. H. Low. Quantum algorithm for simulating real time evolution of lattice Hamiltonians. *SIAM J. Computing*, 0(0):FOCS18–250–284, 2018.
28. T. Hawkins. *Emergence of the theory of Lie groups*. Sources and Studies in the History of Mathematics and Physical Sciences. Springer-Verlag, New York, 2000. An essay in the history of mathematics 1869–1926.
29. M. Hayashi. *Quantum information theory*. Graduate Texts in Physics. Springer-Verlag, Berlin, second edition, 2017. Mathematical foundation.
30. F. Hiai and D. Petz. *Introduction to matrix analysis and applications*. Universitext. Springer, Cham; Hindustan Book Agency, New Delhi, 2014.
31. R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. Quantum entanglement. *Rev. Mod. Phys.*, 81:865–942, Jun 2009.
32. A. Iserles, H. Z. Munthe-Kaas, S. P. Nø rsett, and A. Zanna. Lie-group methods. *Acta Numer.*, 9:215–365, 2000.
33. S. Jin, X. Li, and N. Liu. Quantum simulation in the semi-classical regime. *Quantum*, 6:739, June 2022.
34. N. Khaneja and S. J. Glaser. Cartan decomposition of $SU(2^n)$ and control of spin systems. *Chemical Physics*, 267(1):11–23, 2001.
35. E. Kökcü, T. Steckmann, J. K. Freericks, E. F. Dumitrescu, and A. F. Kemper. Fixed depth hamiltonian simulation via Cartan decomposition. preprint, North Carolina State University, 2021. arXiv:2104.00728.
36. S. Krogstad, H. Z. Munthe-Kaas, and A. Zanna. Generalized polar coordinates on Lie groups and numerical integrators. *Numer. Math.*, 114(1):161–187, 2009.

37. C.-C. Lin, A. Chakrabarti, and N. K. Jha. Qlib: Quantum module library. *J. Emerg. Technol. Comput. Syst.*, 11(1), Oct. 2014.
38. S. Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.
39. G. H. Low and I. L. Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.*, 118:010501, Jan 2017.
40. R. I. McLachlan and G. R. W. Quispel. Explicit geometric integration of polynomial vector fields. *BIT*, 44(3):515–538, 2004.
41. M. Melucci. *Introduction to information retrieval and quantum mechanics*, volume 35 of *The Information Retrieval Series*. Springer, Heidelberg, 2015.
42. C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Rev.*, 20(4):801–836, 1978.
43. C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.*, 45(1):3–49, 2003.
44. H. Z. Munthe-Kaas, G. R. W. Quispel, and A. Zanna. Generalized polar decompositions on Lie groups with involutive automorphisms. *Found. Comput. Math.*, 1(3):297–324, 2001.
45. M. Nakahara and T. Ohmi. *Quantum computing*. CRC Press, Boca Raton, FL, 2008. From linear algebra to physical realizations.
46. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
47. R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86:5188–5191, May 2001.
48. D. Riley and H. Usefi. Lie algebras with finite Gelfand-Kirillov dimension. *Proc. Amer. Math. Soc.*, 133(6):1569–1572, 2005.
49. H. N. Sá Earp and J. K. Pachos. A constructive algorithm for the Cartan decomposition of $\mathrm{SU}(2^N)$. *J. Math. Phys.*, 46(8):082108, 11, 2005.
50. J. M. Sanz-Serna and M. P. Calvo. *Numerical Hamiltonian problems*, volume 7 of *Applied Mathematics and Mathematical Computation*. Chapman & Hall, London, 1994.
51. V. Shende, S. Bullock, and I. Markov. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, 2006.
52. M. Suzuki. On the convergence of exponential operators—the Zassenhaus formula, BCH formula and systematic approximants. *Comm. Math. Phys.*, 57(3):193–200, 1977.
53. A. Čap and J. Slovák. *Parabolic geometries I, Background and general theory*, volume 154 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2009.
54. G. Vidal. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.*, 91:147902, Oct 2003.
55. M. M. Wilde. *Quantum information theory*. Cambridge University Press, Cambridge, second edition, 2017.