Fast algorithms for classical specifications of stabiliser states and Clifford gates

Nadish de Silva[1], Wilfred Salmon[2], and Ming Yin[1]

[1]Department of Mathematics, Simon Fraser University
[2]Department of Applied Mathematics and Theoretical Physics, University of Cambridge

Abstract

The stabiliser formalism plays a central role in quantum computing, error correction, and fault-tolerance. Stabiliser states are used to encode computational basis states. Clifford gates are those which can be easily performed fault-tolerantly in the most common error correction schemes. Their mathematical properties are the subject of significant research interest.

Conversions between and verifications of different specifications of stabiliser states and Clifford gates are important components of many classical algorithms in quantum information, e.g. for gate synthesis, circuit optimisation, and simulating quantum circuits. These core functions are also used in the numerical experiments critical to formulating and testing mathematical conjectures on the stabiliser formalism.

We develop novel mathematical insights concerning stabiliser states and Clifford gates that significantly clarify their descriptions. We then utilise these to provide ten new fast algorithms which offer asymptotic advantages over any existing implementations. We show how to rapidly verify that a vector is a stabiliser state, and interconvert between its specification as amplitudes, a quadratic form, and a check matrix. These methods are leveraged to rapidly check if a given unitary matrix is a Clifford gate and to interconvert between the matrix of a Clifford gate and its compact specification as a stabiliser tableau.

For example, we extract the stabiliser tableau of a Clifford gate matrix with N^2 entries in $O(N \log N)$ time. Remarkably, it is not necessary to read all the elements of a Clifford matrix to extract its stabiliser tableau. This is an asymptotic speedup over the best-known method that is superexponential in the number of qubits.

We provide example implementations of our algorithms in `Python`.

1 Overview

Stabiliser states on n-qubits admit three convenient descriptions: as a vector of amplitudes, as a check matrix, and as a triple containing an affine subspace of $\mathbb{Z}_2^n$, a quadratic form, and a linear map. Clifford gates admit two convenient descriptions: as a unitary matrix or as a stabiliser tableau, i.e. a set of $2n$ Pauli gates representing the images of basic Pauli gates under conjugation.

We develop novel mathematical theory concerning stabiliser states and Clifford gates. We use these developments to give much faster algorithms for verifying that a vector or matrix is a stabiliser state or a Clifford gate. We also give much faster algorithms for interconverting between their descriptions. As an illustration, we give a superexponential speedup over the stabiliser tableau extraction function in Google's stabiliser circuit simulator [4, 7].

Each of these different descriptions are the most suitable one to use in different settings. For example, the stabiliser tableau form of a Clifford is used in Gottesman-Knill-type simulations. However, applying a Clifford gate to nonstabiliser states requires its full matrix. Therefore, one major application of our work will be to give speedups to core functions used in the classical simulations of quantum circuits which play a key role in algorithm and hardware development. Our methods are also useful for circuit and gate synthesis; for example, recent work of Kliuchnikov and Schonnenbeck [5, p. 7] uses our work to accelerate gate synthesis of Clifford isometries. Further applications are detailed in Section 1.2 of the full paper: `https://arxiv.org/abs/2311.10357`.

Both our mathematical insights and our resulting algorithms will aid theoretical investigations involving the stabiliser formalism by providing conceptual clarity and enabling faster and larger numerical experiments.

Python implementations of our algorithms can be found at `https://github.com/ndesilva/stabiliser-tools`. This code has been highly optimised using additional tricks and insights beyond the scope of our paper and employs `Numba` to precompile its functions into fast machine code.

2 Background

An n-qubit stabiliser state requires $N = 2^n$ amplitudes to describe as a state vector. They also admit two far more compact descriptions that require a number of bits that is polynomial in n. The first is given by its check matrix which collates the Pauli generators of its stabiliser group. The second is in terms of quadratic forms over $\mathbb{Z}_2$:

Theorem 1 (Dehaene-De Moor [3], 2003)**.** *Every stabiliser state* $|s\rangle$ *(up to phase and normalisation) is specified by a triple* $(\mathcal{A}, Q, \ell)$ *where* $\mathcal{A} \subset \mathbb{Z}_2^n$ *is the affine subspace* $V + \vec{z_0}$ *for* $V \subset \mathbb{Z}_2^n$ *a vector subspace and* $\vec{z_0} \in \mathbb{Z}_2^n$*,* $Q : V \to \mathbb{Z}_2$ *is a quadratic form, and* $\ell : V \to \mathbb{Z}_2$ *is a linear map:*

$$|s\rangle \propto \sum_{\vec{z} \in V} (-1)^{Q(\vec{z})} i^{\ell(\vec{z})} |\vec{z} + \vec{z_0}\rangle \,. \tag{1}$$

A Clifford gate C is specified by the N^2 complex entries of its matrix. It is also defined up to phase by the data of its *conjugate tuple* [2, Definition 3.6] of Pauli gates $((CZ_1C^*, CX_1C^*), \ldots, (CZ_nC^*, CX_nC^*))$. Each Pauli gate of the conjugate tuple can be specified by $2n + 1$ bits.

The conjugate tuple specification of a Clifford gate is thus vastly more compact ($4n^2 + 2n$ bits vs. N^2 complex numbers) and can be used more easily to e.g. act on stabiliser states in check matrix form or conjugate Pauli gates expressed as vectors in $\mathbb{Z}_2^{2n+1}$. It is well known in the literature as the *stabiliser tableau* of the Clifford gate.

3 Summary of results

Stabiliser states (up to phase) admit three distinct convenient descriptions:

(S1) as a complex vector of amplitudes: an element of $\mathbb{C}^N$,

(S2) as a k-dimensional affine subspace, a linear map, and a quadratic form: an element of $(\mathbb{Z}_2^n)^{k+1} \times \mathbb{Z}_2^n \times \mathbb{Z}_2^{n(n+1)}$,

(S3) as a check matrix: an element of $(\mathbb{Z}_2^{2n+1})^n$.

We will refer to second description as the *quadratic form triple* of a stabiliser state for brevity. We have also described two ways of specifying a Clifford gate (up to phase):

(C1) as a complex matrix C: an element of $\mathbb{C}^{N \times N}$,

(C2) as a conjugate tuple of Pauli gates $U_i = CZ_iC^*, V_i = CX_iC^*$: an element of $((\mathbb{Z}_2^{2n+1})^2)^n$,

In this work, we describe ten novel algorithms for interconverting between the above descriptions and for verifying that a candidate description is valid. In the table below, we compare the worst-case asymptotic complexity of our algorithms to the state-of-the-art [4, 6]. The guide to interpreting these tables immediately follows them.

In all nontrivial cases, we offer complexity advantages of a factor of at least n; this is highly consequential given that proposed error-correcting systems can require millions of qubits [1]. In some cases, we give exponential or higher improvements.

	(S1)	**(S2)**	**(S3)**
(S1)	$Nn^2 \to Nn$	$? \to Nn$	$Nn^2 \to Nn$
(S2)	$Nn^2 \to Nn$	n^2	$Nn^2 \to n^3$
(S3)	$N^4/Nn^2 \to Nn$	$? \to n^3$	n^3

	(C1)	**(C2)**
(C1)	$N^2n^2 \to N^2n$	$N^2n^2 \to Nn$
(C2)	$N^2n^2 \to N^2n$	n^3

- Grey cells correspond to simple tasks, all related to the verification of compact descriptions, for which there exists an obvious method that is nearly optimally fast.

- Diagonal entries of tables correspond to algorithms for verifying candidate descriptions. For example, the (**(S1)**,**(S1)**)-entry in the top-left of the table describes the problem of taking as input a vector in $\mathbb{C}^N$ and deciding whether it represents a valid stabiliser state.

  Off-diagonal entries of tables correspond to algorithms for converting from one valid description to another. For example, the (**(S1)**,**(S3)**)-entry in the top-right of the table describes the problem of taking as input a vector in $\mathbb{C}^N$ that represents a valid stabiliser state and gives as output its check matrix.

- An entry of the form $X \to Y$ indicates that the best currently-known technique requires time $\Omega(X)$ whereas our methods require only $O(Y)$ time.

  An entry of the form $X_1/X_2 \to Y$ indicates that the best currently-known technique that is guaranteed to succeed requires time $\Omega(X_1)$ and the best currently-known technique that succeeds with high probability requires time $\Omega(X_2)$ whereas our methods require only $O(Y)$ time.

- An entry of the form $? \to Y$ indicates that there is neither an existing implementation nor an obvious method for the interconversion task in question; we give a method that requires time $O(Y)$.

We further note that worst-case asymptotic complexity is a very coarse metric that does not fully capture the scope of an algorithm's advantage; it ignores differences in prefactors and lower order terms in their runtime costs. Establishing an asymptotic difference is sufficient, however, to guarantee that one algorithm will outperform another by an arbitrary factor for sufficiently large inputs.

In practice, our algorithms exhibit excellent performance for any number of qubits. We achieve this by working directly with the mathematical description in question.

In two cases, we give conversion algorithms (**(S1)**,**(S3)** to **(S2)**) for which no alternative implementations exist.

4 Methods

While the various descriptions of stabiliser states and Clifford gates are all common knowledge in the quantum information community, the precise relationships between them have, in many cases, remained obscure. At the core of our fast algorithms are insights that conceptually clarify these relationships. Bringing their connections to the fore simplifies conversion tasks and massively reduces redundant calculations.

Our nine novel algorithms are built up from four core conversions that employ newfound connections. Our other new conversion algorithms utilise the connections between descriptions developed below and by composing our new techniques. Our verification algorithms check for a successful conversion supplemented with the minimal additional checks made necessary by the fact that we do not assume our input description is valid.

Extracting a quadratic form triple from amplitudes. We show how to quickly extract the affine subspace defining the support of a stabiliser state by proving that an ordered vector space over $\mathbb{Z}_2$ has a basis defined by vectors enumerated by powers of 2. We then extract a quadratic form by solving an inhomogenous linear system.

Extracting a check matrix from a quadratic form triple. The relationship between the descriptions of a stabiliser state as a check matrix and as a quadratic form triple has not been previously made clear in the literature. We establish some required relations between these two descriptions. With these in hand, we find that from a row-reduced check matrix, one can essentially read off its corresponding quadratic form triple data.

Extracting a quadratic form triple from a check matrix. Using the insights developed in the previous method, we can essentially reverse the process to find that the quadratic form triple of a stabiliser state can be read off from its check matrix by constructing and solving an underdetermined linear system.

Extracting the stabiliser tableau of a Clifford gate matrix. Here, we use the fact that the first column of a Clifford gate matrix is a stabiliser state whose stabiliser group generators we can rapidly extract using the above method. These give us one half of the stabiliser tableau. We show that the second half can be extracted from the relative phases of $n-1$ of the remaining columns. This requires inspecting only a few elements beyond the first column.

References

[1] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, et al. Quantum algorithms: A survey of applications and end-to-end complexities. *arXiv preprint arXiv:2310.03011*, 2023. 3

[2] N. de Silva. Efficient quantum gate teleportation in higher dimensions. *Proceedings of the Royal Society A*, 477(2251):20200865, 2021. 2

[3] J. Dehaene and B. De Moor. Clifford group, stabilizer states, and linear and quadratic operations over GF(2). *Physical Review A*, 68(4):042318, 2003. 1

[4] C. Gidney. Stim: a fast stabilizer circuit simulator. *Quantum*, 5:497, July 2021. 1, 3

[5] V. Kliuchnikov and S. Schonnenbeck. Stabilizer operators and barnes-wall lattices. *arXiv preprint arXiv:2404.17677*, 2024. 1

[6] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023. 3

[7] StackExchange. How do I check if a gate represented by unitary U is a Clifford gate?, 2020. https://quantumcomputing.stackexchange.com/questions/13157/how-do-i-check-if-a-gate-represented-by-unitary-u-is-a-clifford-gate. 1