Quantum singular-value decomposition of nonsparse low-rank matrices

Patrick Rebentrost,[1,2,*] Adrian Steffens,[1,3] Iman Marvian,[1] and Seth Lloyd[1,4,†]
[1]*Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*
[2]*Xanadu, 372 Richmond St W, Toronto, Canada M5V 2L7*
[3]*Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany*
[4]*Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

(Received 13 November 2017; published 24 January 2018)

We present a method to exponentiate nonsparse indefinite low-rank matrices on a quantum computer. Given access to the elements of the matrix, our method allows one to determine the singular values and their associated singular vectors in time exponentially faster in the dimension of the matrix than known classical algorithms. The method extends to non-Hermitian and nonsquare matrices via matrix embedding. Moreover, our method preserves the phase relations between the singular spaces allowing for efficient algorithms that require operating on the entire singular-value decomposition of a matrix. As an example of such an algorithm, we discuss the Procrustes problem of finding a closest isometry to a given matrix.

DOI: 10.1103/PhysRevA.97.012327

I. INTRODUCTION

Matrix computations are central to many algorithms in optimization and machine learning [1–3]. At the heart of these algorithms regularly lies an eigenvalue or a singular-value decomposition of a matrix, or a matrix inversion. Such tasks could be performed efficiently via phase estimation on a universal quantum computer [4], as long as the matrix can be simulated (exponentiated) efficiently and controllably as a Hamiltonian acting on a quantum state. Reference [5] paved the way for such a simulation of quantum systems by introducing an efficient algorithm for exponentiating Hamiltonians with tensor product structure—enabling applications such as in quantum computing for quantum chemistry [6]. Step by step, more general types of quantum systems were tackled and performance was increased: Aharonov and Ta-Shma [7] showed a method for simulating quantum systems described by sparse Hamiltonians, while Childs *et al.* [8] demonstrated the simulation of a quantum walk on a sparse graph. Berry *et al.* [9] reduced the temporal scaling to approximately linear via higher-order Suzuki integrators and further improvements in the sparsity scaling were presented in Ref. [10]. Beyond sparse Hamiltonians, quantum principal component analysis (qPCA) was shown to be capable of efficiently handling nonsparse positive-semidefinite low-rank Hamiltonians [11] given multiple copies (or samples) of the Hamiltonian as a quantum density matrix, as opposed to the requirement of quantum oracles as in prior works. This method offers applications in quantum process tomography and state discrimination [11], as well as in quantum machine learning [12–18], specifically in curve fitting [19] and support vector machines [20]. In an oracular setting, Refs. [10,21,22] showed the simulation of nonsparse Hamiltonians via discrete quantum walks. The scaling in terms of the simulated time t is $t^{3/2}$ or even linear in t.

In the spirit of Ref. [11], we provide an alternative method for efficiently simulating nonsparse matrices in an oracular setting that requires only *one*-sparse simulation techniques. Compared to Ref. [11], the matrices are not restricted to be positive semidefinite and the samples required are simple uniform superpositions instead of the actual Hamiltonian. We achieve a run time in terms of the matrix maximum element and a t^2 scaling, and discuss a class of low-rank matrices which can be efficiently simulated.

In order to effectively treat a general non-Hermitian nonquadratic matrix, we make use of an indefinite "extended Hermitian matrix" that incorporates the original matrix. With such an extended matrix, we are able to efficiently determine the singular-value decomposition of dense nonsquare, low-rank matrices. As one possible application of our method, we discuss the Procrustes problem [1] of finding a closest isometric matrix.

II. METHOD

We have been given an $N \times N$ dense (nonsparse) Hermitian indefinite matrix $A \in \mathbb{C}^{N\times N}$ via efficient oracle access to the elements of A. For the more general case of non-Hermitian matrices, see Sec. V below. The oracle either performs an efficient computation of the matrix elements or provides access to a storage medium for the elements such as quantum RAM [23,24]. Our method simulates $e^{-i(A/N)t}$ on an arbitrary quantum state for arbitrary times t. Note that the eigenvalues of A/N are bounded by $\pm\|A\|_{\max}$, where $\|A\|_{\max} = \max_{mn}|A_{mn}|$ denotes the maximal absolute value of the matrix elements of A. This means that there exist matrices A for which the unitary $e^{-i(A/N)t}$ can be far from the identity operator for a time of the order of $\|A\|_{\max}^{-1}$, i.e., an initial state can evolve to a perfectly distinguishable state. For such times, the unitary $e^{-i(A/N)t}$ can

*patrick@xanadu.ai
†slloyd@mit.edu

be well approximated by a unitary generated by a low-rank matrix.

Let σ and ρ be N-dimensional density matrices. The state σ is the target state on which the matrix exponential of A/N is applied, while multiple copies of ρ are used as ancillary states. Our method embeds the N^2 elements of A into a Hermitian sparse matrix $S_A \in \mathbb{C}^{N^2\times N^2}$, which we call "modified swap matrix" because of its close relation to the usual swap matrix, but with each column of S_A containing a single element of A. The modified swap matrix between the registers for a single copy of ρ and σ is defined as

$$S_A = \sum_{j,k=1}^{N} A_{jk}\, |k\rangle\langle j| \otimes |j\rangle\langle k| \in \mathbb{C}^{N^2\times N^2}. \tag{1}$$

This matrix is one-sparse in a quadratically bigger space and reduces to the usual swap matrix for $A_{jk} = 1$ and $j,k = 1,\ldots,N$. Given efficient oracle access to the elements, we can simulate a one-sparse matrix such as S_A with a constant number of oracle calls and negligible error [7–9,25]. We discuss the oracle access below in Sec. IV. The resulting matrix exponential of S_A is applied to a tensor product of a uniform superposition and an arbitrary state. Performing S_A for small Δt leads to a reduced dynamics of σ when expanded to terms of second order in Δt as

$$\begin{aligned}&\mathrm{tr}_1\{e^{-i\,S_A\Delta t}\,\rho\otimes\sigma\, e^{i\,S_A\Delta t}\}\\ &\quad=\sigma - i\,\mathrm{tr}_1\{S_A\,\rho\otimes\sigma\}\Delta t + i\,\mathrm{tr}_1\{\rho\otimes\sigma\,S_A\}\Delta t + O(\Delta t^2).\end{aligned} \tag{2}$$

Here, tr_1 denotes the partial trace over the first register containing ρ. The first $O(\Delta t)$ term can be rewritten as $\mathrm{tr}_1\{S_A\,\rho\otimes\sigma\} = \sum_{j,k=1}^{N} A_{jk}\langle j|\rho|k\rangle |j\rangle\langle k|\,\sigma$. Choosing $\rho = |\vec{1}\rangle\langle\vec{1}|$, with the uniform superposition $|\vec{1}\rangle := \frac{1}{\sqrt{N}}\sum_k |k\rangle$, leads to $\mathrm{tr}_1\{S_A\ \rho\otimes\sigma\} = \frac{A}{N}\,\sigma$. This choice for ρ contrasts with the qPCA method, where ρ is proportional to the simulated matrix [11]. Analogously, the second $O(\Delta t)$ term becomes $\mathrm{tr}_1\{\rho\otimes\sigma\,S_A\} = \sigma\,\frac{A}{N}$. Thus, for small times, evolving with the modified swap matrix S_A on the bigger system is equivalent to evolving with A/N on the σ subsystem,

$$\begin{aligned}\mathrm{tr}_1\{e^{-i\,S_A\Delta t}\,\rho\otimes\sigma\, e^{i\,S_A\Delta t}\} &= \sigma - i\,\frac{\Delta t}{N}\,[A,\sigma] + O(\Delta t^2)\\ &\approx e^{-i\,\frac{A}{N}\Delta t}\,\sigma\, e^{i\,\frac{A}{N}\Delta t}.\end{aligned} \tag{3}$$

Let ϵ_0 be the trace norm of the error term $O(\Delta t^2)$. We can bound this error by $\epsilon_0 \leqslant 2\,\|A\|_{\max}^2\,\Delta t^2$ (see the Appendix). Note that $\|A\|_{\max}$ coincides with the largest absolute eigenvalue of S_A. The operation in Eq. (3) can be performed multiple times in a forward Euler fashion using multiple copies of ρ. For n steps, the resulting error is $\epsilon = n\,\epsilon_0$. The simulated time is $t = n\,\Delta t$. Hence, fixing ϵ and t,

$$n = O\left(\frac{t^2}{\epsilon}\|A\|_{\max}^2\right) \tag{4}$$

steps are required to simulate $e^{-i\,(A/N)\,t}$. The total run time of our method is $n\,T_A$: the number of steps n is multiplied with the matrix oracle access time T_A (see Sec. IV below).

We now discuss for which Hermitian matrices the algorithm runs efficiently. At a simulation time t, only the eigenvalues of A/N with $|\lambda_j|/N = \Omega(1/t)$ matter. Let the number of these eigenvalues be $r \leqslant N$. Thus, effectively, a matrix A_r/N is simulated for which the following lower bound holds: $\mathrm{tr}\{A_r^2/N^2\} = \sum_{j=1}^{r}\lambda_j^2/N^2 = \Omega(r/t^2)$. It also holds that $\mathrm{tr}\{A_r^2/N^2\} \leqslant \mathrm{tr}\{A^2/N^2\} = \|A\|_F^2/N^2 \leqslant \|A\|_{\max}^2$, with $\|A\|_F$ the Frobenius norm of A. Combining the upper and the lower bounds, we find that the rank of the effectively simulated matrix is $r = O(\|A\|_{\max}^2 t^2)$.

For the algorithm to be efficient in terms of matrix oracle calls, we require that the number of simulation steps n is $O(\mathrm{poly}\log N)$. Let the desired error be $1/\epsilon = O(\mathrm{poly}\log N)$. Assuming $\|A\|_{\max} = \Theta(1)$, meaning a constant independent of N, we have from Eq. (4) that we can only exponentiate for a time $t = O(\mathrm{poly}\log N)$. For such times, only the large eigenvalues of A/N with $|\lambda_j|/N = \Omega(1/\mathrm{poly}\log N)$ matter. Such eigenvalues appear if the matrix is dense enough, for example, A/N has $\Theta(N)$ nonzeros of size $\Theta(1/N)$ per row. For the rank of the simulated matrix in this case, we find that $r = O(\mathrm{poly}\log N)$, and thus effectively a low-rank matrix is simulated. To summarize, we expect the method to work well for low-rank matrices A that are dense with relatively small matrix elements.

A large class of matrices satisfies these criteria. Draw a random unitary matrix $U \in \mathbb{C}^{N\times N}$ from the Haar measure and r suitable eigenvalues of size $|\lambda_j| = \Theta(N)$ and multiply them as $U\,\mathrm{diag}_r(\lambda_j)\,U^\dagger$ to construct A. Here, $\mathrm{diag}_r(\lambda_j) \in \mathbb{C}^{N\times N}$ is the diagonal matrix with the r eigenvalues on, e.g., the first r diagonal entries and zero otherwise. The entries of a typical random normalized vector have absolute size $O(1/\sqrt{N})$, and the entries of the outer product of such a vector with itself have absolute matrix elements of size $O(1/N)$. Each eigenvalue of absolute size $\Theta(N)$ is multiplied with such an outer product and the r terms are summed up. Thus, a typical matrix element of A will be of size $O(\sqrt{r})$ and $\|A\|_{\max} = O(r)$.

III. PHASE ESTIMATION

Phase estimation provides a gateway from unitary simulation to many interesting applications. For the use in phase estimation, we extend our method such that the matrix exponentiation of A/N can be performed conditioned on additional control qubits. With our method, the eigenvalues λ_j/N of A/N can be both positive and negative. The modified swap operator S_A corresponding to a Hermitian matrix A with eigendecomposition $A = \sum_j \lambda_j |u_j\rangle\langle u_j|$ is augmented as $|1\rangle\langle 1|\otimes S_A$, which still is a one-sparse Hermitian operator. The resulting unitary $e^{-i\,|1\rangle\langle 1|\otimes S_A\Delta t} = |0\rangle\langle 0|\otimes\mathbb{1} + |1\rangle\langle 1|\otimes e^{-i\,S_A\Delta t}$ is efficiently simulatable. This operator is applied to a state $|c\rangle\langle c|\otimes\rho\otimes\sigma$, where $|c\rangle$ is an arbitrary control qubit state. Sequential application of such controlled operations allows one to use phase estimation in order to prepare the state [25],

$$|\phi\rangle = \frac{1}{\sqrt{\sum_j |\beta_j|^2}}\sum_{|\lambda_j|/N\geqslant\epsilon}\beta_j|u_j\rangle|\lambda_j/N\rangle, \tag{5}$$

from an initial state $|\psi\rangle|0\ldots0\rangle$ with $O(\lceil\log(1/\epsilon)\rceil)$ control qubits forming an eigenvalue value register. Here, $\beta_j := \langle u_j|\psi\rangle$ and ϵ is the accuracy for resolving the eigenvalues. To achieve this accuracy, phase estimation is run for a total time $t = O(1/\epsilon)$. Thus, $O(\|A\|_{\max}^2/\epsilon^3)$ queries of the

oracle for A are required, which is of the order of $O(\text{poly}\log N)$ under the low-rank assumption for A discussed above.

IV. MATRIX ORACLE AND RESOURCE REQUIREMENTS

To simulate the modified swap matrix S_A, we employ the methods developed in Refs. [8,9]. First, we assume quantum oracle access to the original matrix A,

$$|j\,k\rangle|0\cdots0\rangle \mapsto |j\,k\rangle|A_{jk}\rangle. \tag{6}$$

This operation can be provided by quantum random access memory (qRAM) [23,24] using $O(N^2)$ storage space and quantum switches for accessing the data in $T_{\mathrm{A}} = O(\log^2 N)$ operations. Alternatively, matrices whose elements are efficiently computable have, by definition, $T_{\mathrm{A}} = O(\text{poly}\log N)$. The unitary operation for the simulation of the one-sparse matrix S_A with the sparse methods [8,9] can be simply constructed from the oracle in Eq. (6) and is given by

$$|(j,k)\rangle|0\cdots0\rangle \mapsto |(j,k)\rangle|(k,j),(S_A)_{(k,j),(j,k)}\rangle. \tag{7}$$

Here, we use (j,k) as the label for the row and the column index of the modified swap matrix.

In the following, we will compare the required resources with those of other methods for sparse and nonsparse matrices: For a general $N \times N$ and s-sparse matrix, $O(s\,N)$ elements need to be stored. In certain cases, the sparse matrix features more structure and its elements can be computed efficiently [9,25]. For nonsparse matrices and the qPCA method [11], only multiple copies of the density matrix, as opposed to an operation as in Eq. (6), are required for applications such as state tomography. For machine learning via qPCA [11,20], the density matrix is prepared from a classical source via quantum RAM [23,24] and requires $O(N^2)$ storage. In comparison, the requirements of the method in this work are, in principle, not higher than these sparse and nonsparse methods, both in the case of qRAM access and in the case when matrix elements are computed instead of stored.

V. NONSQUARE MATRICES

Our method enables us to also effectively establish properties of general nonsquare low-rank matrices. To determine the singular-value decomposition of a matrix $A = U\,\Sigma\,V^\dagger \in \mathbb{C}^{M\times N}$ with rank r, simulating the positive-semidefinite matrices $AA^\dagger$ and $A^\dagger A$ via qPCA already yields the correct singular values and vectors. However, essential information is missing, leading to ambiguities in the singular vectors that become evident when inserting diagonal matrices into the singular-value decomposition of $AA^\dagger$ that change the relative phases of the singular vectors,

$$AA^\dagger = U\Sigma^2U^\dagger = U\Sigma D^\dagger V^\dagger\ V D\Sigma U^\dagger =: \hat{A}\hat{A}^\dagger, \tag{8}$$

with $D := \mathrm{diag}(e^{-i\vartheta_j})$, and ϑ_j being arbitrary phases. If $Av_j = \sigma_j u_j$ for each $j = 1,\ldots,r$, then

$$\hat{A}v_j = U\Sigma D^\dagger V^\dagger v_j = \sigma_j e^{i\vartheta_j}u_j := \sigma_j\hat{u}_j, \tag{9}$$

which means there are different phase relations between left and right singular vectors in $\hat{A}$ from those in A. Although A and $\hat{A}$ still share the same singular values and even the same singular vectors up to phase factors, $\|A-\hat{A}\|_F$ will, in general (with the exception of positive-semidefinite matrices, where $U = V$), not be zero or even be small: The matrix A cannot be reproduced this way—a singular value decomposition is more than a set of singular values and normalized singular vectors. This affects all kinds of algorithms that require the appropriate phase relations between each left singular vector u_j and the according right singular vector v_j. Such applications determine the best low-rank approximation of a matrix, the signal processing algorithms discussed in Ref. [26], or the nearest isometric matrix, related to the unitary Procrustes problem, of a non-Hermitian matrix.

In order to overcome this issue, consider the "extended matrix,"

$$\tilde{A} := \begin{bmatrix} 0 & A \\ A^\dagger & 0 \end{bmatrix}, \tag{10}$$

which was introduced for singular-value computations in Ref. [27] and, recently, in sparse quantum matrix inversion in [25]. The nonzero eigenvalues of $\tilde{A}$ correspond to $\{\pm\sigma_j\}$, with $\{\sigma_j\}$ being the singular values of A for $j = 1,\ldots,r$. The corresponding eigenvectors are proportional to $(u_j, \pm v_j) \in \mathbb{C}^{M+N}$ (see the Appendix). The left and right singular vectors of A can be extracted from the first M and last N entries, respectively. Since $\tilde{A}$ is Hermitian, its eigenvectors can be assumed to be orthonormal: $\|(u_j,v_j)\|^2 = \|u_j\|^2 + \|v_j\|^2 = 1$, and $(u_j,v_j)\cdot(u_j,-v_j)^\dagger = \|u_j\|^2 - \|v_j\|^2 = 0$, from which follows that the norm of each of the subvectors u_j and v_j is $1/\sqrt{2}$, independent of their respective lengths M and N. The important point is that the eigenvectors of the extended matrix preserve the correct phase relations between the left and right singular vectors since $(e^{i\vartheta_j}u_j, v_j)$ is only an eigenvector of $\tilde{A}$ for the correct phase $e^{i\vartheta_j} = 1$.

The requirements for our quantum algorithm can be satisfied also for the extended matrix. For randomly sampled left and right singular vectors, the matrix elements have maximal size of $O(\sum_{j=1}^r \sigma_j/\sqrt{MN})$, and thus $\sigma_j = O(\sqrt{MN})$. In addition, an $1/(M+N)$ factor arises in the simulation of the extended matrix from the ancillary state $\rho = |\vec{1}\rangle\langle\vec{1}|$ as before, which leads to the requirement $\sigma_j = \Theta(M+N)$. These two conditions for σ_j can be satisfied if the matrix A is not too skewed, i.e., $M = \Theta(N)$. In summary, by simulating the corresponding Hermitian extended matrices, general complex matrices of low rank can be simulated efficiently, yielding the correct singular-value decomposition.

VI. PROCRUSTES PROBLEM

The unitary Procrustes problem is to find the unitary matrix that most accurately transforms one matrix into another. It appears in many fields, such as in image analysis and statistics [1]. As an application for our method in the case of nonsquare matrices, we will discuss the more general Procrustes problem of finding the *isometry* W that most accurately transforms a matrix B into a matrix C: Formally, minimize $\|WB - C\|_F$ among all isometries $W \in \mathbb{C}^{M\times N}$, $W^\dagger W = \mathbb{1}$, with $B \in \mathbb{C}^{N\times K}$ and $C \in \mathbb{C}^{M\times K}$, where $M > N$. The problem is equivalent to the problem of finding the nearest isometric matrix $W \in \mathbb{C}^{M\times N}$ to a matrix $A \in \mathbb{C}^{M\times N}$ by taking $A = CB^\dagger$. Since our quantum algorithm is restricted to low-rank matrices, let $A = CB^\dagger$ be low rank with rank r and

singular-value decomposition $A = U\,\Sigma\,V^\dagger$ with $U \in \mathbb{C}^{M\times r}$, $\Sigma \in \mathbb{R}^{r\times r}$, and $V \in \mathbb{C}^{N\times r}$. The optimal solution to the Procrustes problem is $W = U\,V^\dagger$ [1], setting all singular values to one, in both the low-rank and the full-rank situation. Since A is assumed to be low rank, we find a *partial* isometry with $W^\dagger W = \mathbb{P}_{\mathrm{col}(V)}$, with $\mathbb{P}_{\mathrm{col}(V)}$ being the projector into the subspace spanned by the columns of V. Thus, W acts as an isometry for vectors in that subspace (see the Appendix).

In a quantum algorithm, intrinsically consisting of quantum operations acting on quantum states, let the task be to apply the nearest low-rank isometry to a quantum state $|\psi\rangle$. The state $|\psi\rangle$ is assumed to be in or close to the subspace spanned by the columns of V. We assume that the extended matrix for A in Eq. (10) is given in oracular form and that A is not too skewed such that $\sigma_j/(M+N) = \Theta(1)$ and $\|A\|_{\max} = \Theta(1)$. We perform phase estimation on the input state $|0,\psi\rangle|0\ldots0\rangle$ and, analogous to Eq. (5), obtain a state proportional to

$$\sum_{\frac{\sigma_j}{M+N}\geqslant\epsilon} \beta_j^{\pm}\,|u_j,\pm v_j\rangle\Big|\pm\frac{\sigma_j}{M+N}\Big\rangle, \tag{11}$$

with $\beta_j^{\pm} = \langle u_j,\pm v_j|0,\psi\rangle = \pm\langle v_j|\psi\rangle/\sqrt{2}$. The sum has $2r$ terms corresponding to the eigenvalues of the extended matrix with absolute value greater than $(M+N)\epsilon$. Performing a σ_z operation on the qubit encoding the sign of the respective eigenvalue and uncomputing the eigenvalue register yields a state proportional to $\sum_j \beta_j|u_j,\pm v_j\rangle$. Projecting onto the u_j part (with success probability $1/2$) results in a state proportional to

$$\sum_{\frac{\sigma_j}{M+N}\geqslant\epsilon} |u_j\rangle\langle v_j|\psi\rangle \propto U\,V^\dagger|\psi\rangle. \tag{12}$$

This procedure leads to the preparation of the desired state for the nonsquare low-rank Procrustes problem with accuracy ϵ in runtime $O[\|A\|_{\max}^2\log^2(N+M)/\epsilon^3]$. In contrast, performing the singular-value decomposition of a low-rank A classically requires, in general, without further structural assumptions, a runtime $O(N^3)$.

VII. CONCLUSION

The method presented here allows nonsparse low-rank nonpositive Hermitian $N\times N$ matrices A/N to be exponentiated for a time t with accuracy ϵ in runtime $O(\frac{t^2}{\epsilon}\|A\|_{\max}^2\,T_A)$, where $\|A\|_{\max}$ is the maximal absolute element of A and T_A is the data-access time. If the matrix elements are accessed via quantum RAM or computed efficiently and the significant eigenvalues of A are $\Theta(N)$, our method can achieve a runtime of $O(\mathrm{poly}\log N)$ for a large class of matrices. Our method allows non-Hermitian and nonsquare matrices to be exponentiated via extended Hermitian matrices.

We have shown how to compute the singular-value decomposition of a non-Hermitian nonsparse matrix on a quantum computer directly while keeping the relative phase information. The numerous potential applications of the quantum singular-value decomposition include determining the pseudoinverse of a matrix or its closest isometry exponentially faster than any known classical algorithm. In addition, the present method has been modified to a continuous-variable setting as a subroutine for Gaussian process regression [28]. In addition, by using a (possibly unknown) ancillary state different from the uniform superposition, the oracular setting of the present work and the tomography setting of Ref. [11] can be combined.

ACKNOWLEDGMENTS

We acknowledge support from DARPA, NSF, and AFOSR. A.S. thanks the German National Academic Foundation (Studienstiftung des deutschen Volkes) and the Fritz Haber Institute of the Max Planck Society for support.

APPENDIX

1. Norms

Denote the maximum absolute element of a matrix $A\in\mathbb{C}^{N\times N}$ with $\|A\|_{\max} = \max_{j,k}|A_{jk}|$. The Frobenius or Hilbert-Schmidt norm is given by $\|A\|_F = \sqrt{\sum_{j,k}|A_{jk}|^2}$ and its nuclear norm by $\|A\|_* = \sum_{i=1}^r \sigma_i$, where r is the rank and σ_j are the singular values.

2. Modified swap matrix

The modified swap matrix is defined as

$$S_A = \sum_{j,k=1}^{N} A_{jk}|k\rangle\langle j|\otimes|j\rangle\langle k| \in \mathbb{C}^{N^2\times N^2}. \tag{A1}$$

Taking $A_{jk} = 1$ leads to the original swap matrix $S = \sum_{j,k=1}^N |k\rangle\langle j|\otimes|j\rangle\langle k| \in \mathbb{C}^{N^2\times N^2}$. The N^2 eigenvalues of S_A are

$$A_{11}, A_{22}, \ldots, A_{NN}, A_{12}, -A_{12}, \ldots, A_{j,k>j}, -A_{j,k>j}, \ldots, \tag{A2}$$

where $k>j$ denotes an index k greater than j. The maximal absolute eigenvalue of S_A is thus $\max_{j,k}|A_{jk}| \equiv \|A\|_{\max}$, corresponding to the maximal absolute matrix element of A. The square of the modified swap matrix is

$$(S_A)^2 = \sum_{j,k=1}^{N} |A_{jk}|^2\,|k\rangle\langle k|\otimes|j\rangle\langle j| \leqslant \|A\|_{\max}^2\,\mathbb{1}. \tag{A3}$$

Its eigenvalues are $|A_{jk}|^2$ and the maximal eigenvalue is $\|A\|_{\max}^2$. This already points to the result that the second-order error of our method naturally scales with $\|A\|_{\max}^2$, which we will now derive.

3. Error analysis

In the following, we will estimate the error from the second-order term in Δt in Eq. (2). The nuclear norm of the operator part of the second-order error is

$$\epsilon_{\rho,\sigma} = \|\mathrm{tr}_1\{S_A\,\rho\otimes\sigma\,S_A\} - \tfrac{1}{2}\mathrm{tr}_1\{(S_A)^2\,\rho\otimes\sigma\} - \frac{1}{2}\mathrm{tr}_1\{\rho\otimes\sigma\,(S_A)^2\}\|_*. \tag{A4}$$

In Ref. [11], this error was equal to $\epsilon_{\rho,\sigma}^{\mathrm{qPCA}} = \|\rho-\sigma\|_* \leqslant 2$, which is achieved in the present algorithm by choosing A such that $A_{jk} = 1$ for each j,k. Here, our algorithm coincides with

the qPCA method for ρ chosen as the uniform superposition. For general low-rank A, we bound Eq. (A4) via the triangle inequality. Taking the nuclear norm of the first term results in

$$\begin{aligned}\|\mathrm{tr}_1\{S_A\rho\otimes\sigma S_A\}\|_* &\leqslant \|S_A\rho\otimes\sigma S_A\|_* \\ &\leqslant \|\rho\otimes\sigma\|_*\left\|S_A^2\right\|_* \leqslant \|A\|_{\max}^2. \end{aligned} \quad \text{(A5)}$$

The second and third term can be treated similarly. We obtain $\|\mathrm{tr}_1\{(S_A)^2\rho\otimes\sigma\}\|_* \leqslant \|A\|_{\max}^2$. Combining all terms yields the bound

$$\epsilon_{\rho,\sigma} \leqslant 2\|A\|_{\max}^2. \quad \text{(A6)}$$

4. Extended matrices

We define the Hermitian extended matrix $\tilde{A}$ of a general complex-valued, not necessarily square matrix $A\in\mathbb{C}^{M\times N}$ as

$$\tilde{A} = \begin{bmatrix} 0 & A \\ A^\dagger & 0 \end{bmatrix} \in \mathbb{C}^{(M+N)\times(M+N)}. \quad \text{(A7)}$$

Using block matrix identities for the determinant, we obtain its characteristic polynomial,

$$\chi_{\tilde{A}}(\lambda) = \lambda^{|M-N|}\,\det(\lambda\mathbb{1}+\sqrt{AA^\dagger})(\lambda\mathbb{1}-\sqrt{AA^\dagger}). \quad \text{(A8)}$$

The eigenvalues of $\tilde{A}$ are either zero or correspond to $\{\pm\sigma_j\}$, the singular values of A for $j=1,\ldots,r$ with an additional sign. Hence, if A has low-rank r, then $\tilde{A}$ has low-rank $2r$. The corresponding eigenvectors are proportional to $(u_j,\pm v_j)\in\mathbb{C}^{M+N}$ since

$$\begin{bmatrix} \mp\sigma_j\mathbb{1} & A \\ A^\dagger & \mp\sigma_j\mathbb{1} \end{bmatrix}\cdot\begin{bmatrix} u_j \\ \pm v_j \end{bmatrix} = 0, \quad \text{(A9)}$$

where u_j and v_j are the jth left and right singular vector of A, respectively. The important point is that the eigenvectors of the extended matrix preserve the correct phase relations between the left and right singular vectors since $(e^{i\vartheta_j}u_j,\pm v_j)$ is only an eigenvector of $\tilde{A}$ for the correct phase $e^{i\vartheta_j}=1$,

$$\begin{aligned}\begin{bmatrix} \mp\sigma_j\mathbb{1} & A \\ A^\dagger & \mp\sigma_j\mathbb{1} \end{bmatrix}\cdot\begin{bmatrix} e^{i\vartheta_j}u_j \\ \pm v_j \end{bmatrix} &= \begin{bmatrix} \mp\sigma_j e^{i\vartheta_j}u_j \pm Av_j \\ e^{i\vartheta_j}A^\dagger u_j - \sigma_j v_j \end{bmatrix} \\ &= (e^{i\vartheta_j}-1)\sigma_j\begin{bmatrix} \mp u_j \\ v_j \end{bmatrix}. \end{aligned} \quad \text{(A10)}$$

The right-hand side is only equal to zero for the correct phase $e^{i\vartheta_j}=1$.

5. Low-rank Procrustes

Let the isometry be $W=UV^\dagger$ with $U\in\mathbb{C}^{M\times r}$ and $V\in\mathbb{C}^{N\times r}$. Assume that $M>N$, leading to orthogonal columns in the full-rank Procrustes problem ($r=N$). We find for the low-rank (partial) isometry that

$$W^\dagger W = VU^\dagger UV^\dagger = VV^\dagger = \sum_{j=1}^{r}\vec{v}_j\vec{v}_j^\dagger. \quad \text{(A11)}$$

Pick an arbitrary vector $\vec{x}=\sum_{j=1}^{r}\alpha_j\vec{v}_j+\vec{x}^\perp=\vec{x}^\parallel+\vec{x}^\perp$, where $\vec{x}^\perp$ denotes the part orthogonal to the orthonormal vectors $\vec{v}_j$. Then,

$$W^\dagger W\,\vec{x} = \sum_{j=1}^{r}\alpha_j\vec{v}_j = \vec{x}^\parallel. \quad \text{(A12)}$$

Thus, $W^\dagger W$ acts as the identity operator in the low-rank subspace and projects out the space perpendicular to that subspace.

[1] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Vol. 3 (JHU Press, Baltimore, MD, 2012).

[2] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004).

[3] K. P. Murphy, *Machine Learning: A Probabilistic Perspective* (MIT Press, Cambridge, MA, 2012) .

[4] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010).

[5] S. Lloyd, Science **273**, 1073 (1996).

[6] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Science **309**, 1704 (2005).

[7] D. Aharonov and A. Ta-Shma, in *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing, STOC '03*, edited by L. L. Larmore and M. X. Goemans (ACM, New York, 2003), pp. 20–29.

[8] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, in *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing, STOC '03*, edited by L. L. Larmore and M. X. Goemans (ACM, New York, 2003), pp. 59–68.

[9] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Commun. Math. Phys. **270**, 359 (2007).

[10] D. W. Berry and A. M. Childs, Quantum Info. Comput. **12**, 29 (2012).

[11] S. Lloyd, M. Mohseni, and P. Rebentrost, Nat. Phys. **10**, 631 (2014).

[12] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum Inf. Comput. **15**, 0316 (2015).

[13] N. Wiebe, A. Kapoor, and K. M. Svore, arXiv:1412.3489.

[14] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Phys. Rev. A **94**, 022308 (2016).

[15] M. Schuld, I. Sinayskiy, and F. Petruccione, Phys. Lett. A **379**, 660 (2015).

[16] M. Schuld, I. Sinayskiy, and F. Petruccione, Phys. Rev. A **94**, 022342 (2016).

[17] I. Kerenidis and A. Prakash, arXiv:1603.08675.

[18] H.-K. Lau, R. Pooser, G. Siopsis, and C. Weedbrook, Phys. Rev. Lett. **118**, 080501 (2017).

[19] G. Wang, Phys. Rev. A **96**, 012335 (2017).

[20] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014).

[21] A. Childs, Comm. Math. Phys. **294**, 581 (2010).

[22] A. Childs and R. Kothari, in *Theory of Quantum Computation, Communication, and Cryptography*, Lecture Notes in Computer Science Vol. 6519, edited by W. van Dam, V. Kendon, and S. Severini (Springer, Berlin, 2011), p. 94.

[23] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 160501 (2008).

[24] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. A **78**, 052310 (2008).

[25] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[26] A. Steffens, P. Rebentrost, I. Marvian, J. Eisert, and S. Lloyd, New J. Phys. **19**, 033005 (2017).

[27] G. Golub and W. Kahan, J. Soc. Ind. Appl. Math., Ser. B: Numer. Anal. **2**, 205 (1965).

[28] S. Das, G. Siopsis, and C. Weedbrook, arXiv:1707.00360.