18

Quantum linear system solvers

The authors are grateful to Dong An for reviewing this chapter.

Rough overview (in words)

The goal is to solve linear systems of equations with quantum subroutines. More precisely, a *quantum linear system solver* (QLSS) takes as input an $N \times N$ complex matrix A together with a complex vector b of size N, and outputs a pure quantum state $|\tilde{x}\rangle$ that is an ε-approximation of the normalized solution vector of the linear system of equations $Ax = b$. In basic versions, QLSSs do so by loading the normalized entries of the matrix A and the normalized entries of the vector b into a unitary quantum circuit, either from a quantum random access memory (QRAM) data structure, or—if the structure of A and b allows for this—by efficiently computing the corresponding entries on the fly.

Crucially, the number of algorithmic qubits of the linear system solver itself is only roughly $\log_2(N)$, which is exponentially smaller than the matrix size. While for general systems the number of QRAM qubits still scales with the matrix/vector size, QRAM encodings can be made more space efficient for sparse systems or can even be avoided when the corresponding entries are efficiently computable. The complexity of QLSSs depends on the condition number $\kappa(A) = \left\|A^{-1}\right\| \cdot \|A\|$ of the matrix A, and one then aims to give circuits with minimal quantum resource costs—such as ancilla qubits, total gate count, circuit depth, etc.—in terms of $\kappa(A)$ and the desired accuracy $\varepsilon \in (0, 1)$.

Rough overview (in math)

There are different standard input models on how the classical data from (A, b) is loaded into the quantum processing unit, which are equivalent up to small polylogarithmic overhead for general matrices. We state the complexities in

terms of query access of a unitary U_b preparing the $n = \lceil \log_2(N) \rceil$-qubit pure quantum state $|b\rangle = \|b\|^{-1} \cdot \sum_{i=1}^{N} b_i |i\rangle$ for $b = (b_1, \ldots, b_N)$, where $\|\cdot\|$ for vector arguments denotes the standard Euclidean norm, together with an $(\alpha, a, 0)$-block-encoding U_A of the matrix A. The QLSS problem is then stated as follows: for a triple (U_A, U_b, ε) as above, the goal is to create an n-qubit pure quantum state $|\tilde{x}\rangle$ such that

$$\left\| |\tilde{x}\rangle - |x\rangle \right\| \leq \varepsilon$$

with

$$|x\rangle = \frac{\sum_{i=1}^{N} x_i |i\rangle}{\left\| \sum_{i=1}^{N} x_i |i\rangle \right\|} \text{ defined by } Ax = b \text{ with } x = (x_1, \ldots, x_N), \tag{18.1}$$

by employing as few times as possible the unitary operators $U_A, U_b, U_A^\dagger, U_b^\dagger$, controlled versions of $U_A, U_b, U_A^\dagger, U_b^\dagger$, and additional quantum gates on potentially additional ancilla qubits. An alternative (and closely related) error metric studied in some works is based on the trace norm, requiring $\frac{1}{2} \| |x\rangle\langle x| - |\tilde{x}\rangle\langle\tilde{x}| \|_1 \leq \varepsilon$.

One way to think of the QLSS problem is that we seek the matrix inverse A^{-1}, and that this can be implemented by, for example, quantum singular value transformation (QSVT) acting on A (via its block-encoding) with a polynomial approximation of the inverse function on the interval $[\|A\|/\kappa(A), \|A\|]$. The complexity of the corresponding scheme thereby depends on the degree of the polynomial needed for a good approximation of the inverse function on the relevant interval, and as such on the condition number $\kappa(A)$, the normalization factor α, and the approximation error ε of the resulting QLSS. In fact, it turns out that the complexity of most quantum algorithms depends on the following combined quantity

$$\kappa'(A) := \kappa(A) \cdot \frac{\alpha}{\|A\|} = \alpha \cdot \|A^{-1}\|,$$

which is no smaller than $\kappa(A)$, because $\alpha \geq \|A\|$ due to the unitarity of the block-encoding. Note that in QRAM-based implementations for dense matrices A, one naturally gets $\alpha = \|A\|_F$, which then leads to linear complexity dependence on the Frobenius norm $\|A\|_F$.

As noted in [1039, 248], in general, we need not assume that A is invertible nor that it is a square matrix, but can instead use the Moore–Penrose pseudoinverse A^+ of the matrix to solve the regression problem Eq. (18.1) in a least-squares sense, in which case one needs to appropriately change the definition of $\kappa(A)$ to $\|A^+\| \cdot \|A\|$. In fact, the above QSVT-based approach directly solves this more general version of the problem [431].

Dominant resource cost (gates/qubits)

The performance of different QLSSs is typically compared based on how their query complexity (to U_A and U_b) grows with the condition number, where a lower bound of $\Omega(\kappa(A))$ is known; see [500, 814]. Methods achieving $O(\kappa'(A))$ dependence are termed "optimal" and methods achieving $\kappa'(A)$ polylog$(\kappa'(A))$ are termed "near-optimal."[1]

The first optimal method was given in [313] (for invertible matrices), which does not directly employ the QSVT for the inverse function. Instead, it is based on discrete adiabatic methods together with quantum eigenstate filtering based on the QSVT for a minimax polynomial [689]. In particular, the adiabatic portion prepares an "ansatz" state $|x_{\text{ans}}\rangle$ for which $|\langle x_{\text{ans}}|x\rangle|^2 \geq 1/2$, using at most $O(\kappa'(A))$ (controlled) queries to U_A and U_b. Then, the eigenstate filtering step refines this state by approximately projecting it onto $|x\rangle$: one obtains the state $|\tilde{x}\rangle$ that is ε-far from $|x\rangle$ at additional query cost $O(\kappa'(A)\log(1/\varepsilon))$. The projection succeeds with probability $p \geq 1/2$, so the whole procedure must be repeated no more than twice on average. Overall, the expected number of queries made by the algorithm is Q controlled queries to each of U_A and $U_A^\dagger$ and $2Q$ queries to each of U_b and $U_b^\dagger$, where

$$Q = \kappa'(A)\Big(C + D\ln(2\varepsilon^{-1})\Big) + o(\kappa'(A)) = O\left(\kappa'(A)\log(\varepsilon^{-1})\right). \tag{18.2}$$

Here, $o(\kappa'(A))$ denotes terms growing sublinearly in $\kappa'(A)$, and C, D are constants. The algorithm operates on $n + O(1)$ qubits ($n + 5$ in the case of [313]), plus the additional qubits used for the block-encoding, discussed in more detail below. There is an additional constant quantum gate complexity for each query to U_A and U_b. For the discrete adiabatic method in [313], the constant C can be rigorously bounded as $C \leq 117{,}235$[2] and the constant D is at most 2. Note that when C is this large, the corresponding term will actually dominate the $D\kappa'(A)\log(\varepsilon^{-1})$ term for practical scenarios.

Subsequent work has given alternative methods that achieve optimal asymptotic complexity [325, 327]. Reference [325] achieves this by a small modification to and improved analysis of the adiabatic path-following method of [964]. Meanwhile, [327] replaces the step of ansatz state preparation via adiabatic methods with a simpler norm-estimation step, where one seeks a constant-

[1] Regarding the optimal ε dependence, it has additionally been claimed [313] that a complexity of $O(\kappa\log(1/\varepsilon))$ is jointly optimal in both κ and ε, based on forthcoming work by Harrow and Kothari; see also [314, Appendix A].

[2] This number is derived from applying [313, Theorem 9] with $\sqrt{2-\sqrt{2}} \times 44{,}864 \times \kappa$ steps, each of which incurs one call to the block-encoding, such that the output is guaranteed to have overlap at least $1/\sqrt{2}$ with the ideal state. Eigenstate filtering then succeeds with probability at least $1/2$; accounting for the need to repeat twice on average, one arrives at a constant 117,235, matching [572, Eq. (L2)].

factor approximation of the Euclidean norm $\|x\|$, following that with an eigenstate filtering–like step. An optimized version of this approach was reported to have complexity following Eq. (18.2) with $C = 56$ and $D = 1.05$. Additionally, this method does not require A to be invertible, but does require b to be in the column space of A [327].

Other known QLSSs with suboptimal asymptotic complexities are based on other versions of adiabatic ansatz state preparation [964, 31, 689], QSVT [431, 744], linear combination of unitaries (LCU) [282], or variable-time amplitude amplification (VTAA) [26, 29, 248]. While the known bounds on the asymptotic complexities of these methods are slightly worse, it remains open if finite-size performance could be competitive (see, e.g., [572, 327]). Moreover, to date, the VTAA-based algorithms are the only variants that are proven to solve the generic least-squares (pseudoinverse) problem while achieving a near-optimal asymptotic scaling [248].

Note that if the matrix A is given in a classical data structure in the computational basis, then standard ways to create the block-encoding U_A make use of a QRAM structure. For general (dense) matrices A, the requirement is then size $O(N^2)$ (number of qubits) with circuit depth $O(n)$ for each query—or alternatively, as few as $O(n)$ ancilla qubits could suffice, but at the expense of using $O(N^2)$ circuit depth [496, 296]. Initializing the depth-efficient QRAM data structure will in general also take $O(N^2)$ time. However, if A is sparse, either in the computational basis [349], Pauli basis [1011], or any orthonormal basis with efficiently implementable basis transformation, there are more efficient direct constructions for block-encoding A. Moreover, for Pauli basis access, there exist randomized QLSSs with complexity scaling as the ℓ_1-norm of the Pauli coefficients [1022], completely avoiding the use of block-encodings (and as such QRAM and ancilla qubits).

Caveats

QLSSs are an important subroutine for a variety of application areas of quantum algorithms. However, it is crucial to keep track of all the quantum and classical resources required and to compare these to state-of-the-art classical methods. In particular, the following factors should be taken into account:

- The classical precomputation complexities for the eigenstate filtering routine are neglected, but can be kept efficient in practice [356].
- The rigorous upper bound on the size of the complexity constant C has been reduced by several orders of magnitude [327] since the first optimal QLSS was given in [313], but nevertheless remains larger than ideal for usage in applications where QLSS plays a heavy role. However, numerical investiga-

tions of two adiabatic methods on small random matrices gave evidence that the empirical performance of those methods is significantly better than the rigorous upper bounds [314].

- When needed, the QRAM cost can be prohibitive, if it requires the full overhead of quantum error correction and fault tolerance [496], especially for QRAMs of maximum size $O(N^2)$ qubits, required for general (dense) matrices.
- In the formulation of the QLSS problem, the pure quantum state $|x\rangle$ corresponds to the normalized solution vector of the linear system $Ax = b$. While the normalization factor $\|x\|$ can be obtained as well, this comes at the price of added query complexity scaling as $O(\kappa'(A)\varepsilon^{-1}\log(\varepsilon^{-1}))$ [327] (see also [248, Corollary 32]). This nearly achieves the lower bound of $\Omega(\kappa(A)\varepsilon^{-1})$ [327] (note that norm estimation necessarily has worse ε dependence than the QLSS itself).
- QLSSs do not produce a classical description of the solution vector x or an approximation thereof, but rather the pure quantum state $|\tilde{x}\rangle$. In order to obtain a classical approximation of the vector x, one needs to combine QLSSs with pure state quantum tomography, which can be performed using $O(N\varepsilon^{-2})$ samples. If poly(n) query-cost QRAM is also available, then the complexity can be quadratically improved in terms of the precision using optimized pure state tomography [49], or alternatively the overall complexity may be further improved using *iterative refinement* to $O(Ns^2 + Ns\kappa^2(A)/\|A\|) \cdot \text{polylog}(N/\varepsilon)$, as described in [772], where s is the maximum number of nonzero elements of A in any row or column. In the special case of Laplacian, or more generally symmetric, weakly diagonally dominant (SDD) matrices, [50] gives a quantum algorithm with complexity $\widetilde{O}(\sqrt{Ns}/\varepsilon)$ that outputs an ε-approximate solution $\tilde{x}$ with respect to the A-induced norm. (Measuring error in this norm enables their algorithm not to have a condition number dependence.) The algorithm uses QRAM and provides a subquadratic speedup compared to the classical complexity $O(N\log(1/\varepsilon))$, but uses rather different techniques compared to standard QLSS algorithms [500].
- The overall complexities $\widetilde{O}(N\kappa'(A)\varepsilon^{-1})$ and $O(Ns^2 + Ns\kappa^2(A)/\|A\|) \cdot \text{polylog}(N/\varepsilon)$ (where we generously allow poly(n) query-cost QRAM) to obtain a classical description of the solution can be compared to classical textbook Gaussian elimination–based computation, which leads to complexity $O(N^3)$ or more precisely $O(N^\omega)$ with $\omega \in [2, 2.372)$ denoting the matrix multiplication exponent. Further, QLSSs should also be compared with state-of-the-art randomized solvers. For example, the randomized Kaczmarz method [959] with standard classical access to the matrix elements returns

an ε-approximation of the vector x, while scaling as $O(s\kappa_F^2(A)\log(\varepsilon^{-1}))$ for s row-sparse matrices and $\kappa_F(A) = \left\|A^{-1}\right\| \cdot \|A\|_F$. Moreover, if A is s-sparse and positive semidefinite (PSD), then using the conjugate gradient method one can obtain a solution in time $\widetilde{O}(Ns\sqrt{\kappa(A)}\log(\varepsilon^{-1}))$ [481, Chapter 10.2], which can be generalized to the least-squares problem (and thus non-Hermitian matrices) at the cost of a quadratically worse condition number dependence $O(Ns\kappa\log(\kappa(A)/\varepsilon))$ by considering the modified equation $A^\dagger Ax = A^\dagger b$. As such, it seems that the QLSS may not provide a superquadratic speedup when a full classical solution is to be extracted, and even subquadratic speedups seem to be limited to a narrow parameter regime.

- Quantum-inspired methods [271, 433] that start from a classical data structure intended to mimic QRAM—allowing one to sample from probability distributions with probabilities proportional to the squared magnitudes of elements in a given row of A—give samples from an ε-approximation to the solution vector in (N-independent) complexity $O(\kappa_F^4(A)\kappa^2(A)\varepsilon^{-2})$ [924, 433], and can be used to compute an approximate solution by repeated sampling. Note that while the required data structure is classical, it might still be prohibitively expensive to build when the matrix A is huge.
- When it comes to classical methods, solvers that depend on the condition number are useful in practice whenever combined with preconditioners [888]. However, the performance of preconditioners in the quantum setting (see, e.g., [295, 925, 990, 83]) is often only heuristic, or applies only to restricted situations. This topic would benefit from further exploration.

Example use cases

- Quantum interior point methods in convex optimization and corresponding applications [610, 771].
- Quantum machine learning applications [1039, 866].
- Solving differential equations and corresponding applications, for example, for the finite element method that does not require a tomography step [777].

Further reading

- Original QLSS (termed HHL) [500].
- For an overview discussion of QLSS, see [31].
- Optimal-in-κ QLSSs are given in [313, 327, 325].
- There are also known (polynomial) speedups in case one needs a full classical description of the output vector in linear equation solving and in some regression variants [772, 266].