Sevag Gharibian

Lecture 4: Linear systems of equations and a BQP-complete problem

"The method of Gaussian elimination appears in the Chinese mathematical text Chapter Eight: Rectangular Arrays of The Nine Chapters on the Mathematical Art ... parts of it were written as early as approximately 150 BCE ... Carl Friedrich Gauss in 1810 devised a notation for symmetric elimination that was adopted in the 19th century by professional hand computers to solve the normal equations of least-squares problems. The algorithm that is taught in high school was named for Gauss only in the 1950s as a result of confusion over the history of the subject."
— Wikipedia, `https://en.wikipedia.org/wiki/Gaussian_elimination#History`

Contents

Introduction. A classic problem with applications across just about any technical field is that of solving linear systems of equations. The task here is, given a set of N equations of the form $a_{i1}x_1 + \cdots + a_{iN}x_N = b_i$, for inputs $\{a_{ij}, b_i\} \in \mathbb{C}$ and variables $x_i \in \mathbb{C}$, find a simultaneous solution $\mathbf{x} \in \mathbb{C}^N$ to all equations. More compactly, one is given as input $A \in \mathcal{L}(\mathbb{C}^N)$ and target vector $\mathbf{b} \in \mathbb{C}^N$, and asked to find $\mathbf{x} \in \mathbb{C}^N$ satisfying $A\mathbf{x} = \mathbf{b}$.

In this lecture, we study a quantum algorithm for "solving" certain linear systems exponentially faster than known classically. We put the word "solving" in quotes, as the algorithm does not solve the system in the typical fashion of outputting $\mathbf{x} \in \mathbb{C}^N$; in fact, since the algorithm runs in time only *logarithmic* in N, we cannot even hope to output all of $\mathbf{x} \in \mathbb{C}^N$! Along the way, we shall see that the study of this algorithm reveals a natural problem complete for BQP — that of *matrix inversion*.

1 The basic idea: Quantum eigenvalue surgery

The basic principle behind the linear systems algorithm is elegantly simple, and allows for broader applications than just solving linear systems.

Ingredients. The ingredients we will need are as follows:

- The quantum phase estimation algorithm,
- a quantum Hamiltonian simulation algorithm, and
- postselection.

Problem specification. Suppose we have a Hermitian operator $A \in \text{Herm}(\mathbb{C}^N)$ in mind, and a function $f : \mathbb{R} \mapsto \mathbb{R}$. For example, f might be $f(x) = x^2$ or, in the case of linear systems solvers, $f(x) = x^{-1}$. We also have a vector $\mathbf{b} \in \mathbb{C}^N$. Our goal is to compute $f(A) \cdot \mathbf{b} \in \mathbb{C}^N$, where f is acting as an operator function. Classically, this computation would require at least $\Omega(N)$ time, since just writing out the solution vector takes $\Omega(N)$ time.

Now let us change the rules and make some assumptions. Suppose I have an efficient algorithm for preparing $\mathbf{b}$ as a *quantum state*, i.e. $|b\rangle = \sum_{i=1}^{N} b_i |i\rangle \in \mathbb{C}^N$. Note that since $|b\rangle$ lives in an N-dimensional space, we require only $\lceil \log N \rceil + 1$ qubits to represent it. This also means the output vector I compute will presumably live in the same space, i.e. we shall compute

$$|x\rangle = f(A)|b\rangle \in \mathbb{C}^N,$$

which is also an $O(\log N)$-qubit state. Since I already assumed the ability to prepare $|b\rangle$ efficiently, it thus remains to simulate $f(A)$ via a quantum circuit. Before moving forward, however, some important observations:

1. The operator $f(A)$ might not be unitary. Thus, we can only hope to simulate its implementation *probabilistically*, and this is where the technique of postselection plays a role.

2. The output $|x\rangle$ may not be normalized as written above. However, any quantum state is implicitly normalized.

3. The operator $A \in \text{Herm}(\mathbb{C}^N)$ is exponentially large in the number of qubits. We will therefore need to assume some appropriate form of "succinct access" to it, rather than writing it all down.

4. Finally, the output $|x\rangle$ is a *quantum* state, meaning we cannot just read all its entries. Rather, once we prepare $|x\rangle$, we can next perform any efficient measurement we like on $|x\rangle$. Thus, we may learn global properties of the state $|x\rangle$ quickly, but not all of its entries.

Algorithm sketch. Suppose A has spectral decomposition $A = \sum_j \lambda_j |\psi_j\rangle\langle\psi_j|$. We first sketch the algorithm in the simple case where $|b\rangle = |\psi_j\rangle$, such that the goal is to compute

$$|x\rangle = f(A)|\psi_j\rangle = f(\lambda_i)|\psi_j\rangle. \tag{1}$$

As this expression reminds us, operator function f is just a function of the eigenvalues of A. Thus, to simulate $f(A)$, we shall manually "extract" these eigenvalues, "process" them to simulate application of f, and then "reinsert" them where we found them.

Step 1: Eigenvalue extraction. Since A is Hermitian and not necessarily unitary, we cannot implement it directly. However, recall that $U = e^{iA}$ *is* unitary, and has eigenvalues $e^{i\lambda_j}$. In principle, we may hence compute $U|\psi_j\rangle = e^{i\lambda_j}|\psi_j\rangle$. But to "process" λ_j, we need to extract it out of the phase and into a register, i.e. to instead compute $|\lambda_j\rangle|\psi\rangle$, so that we can next try to map this to $|f(\lambda_j)\rangle|\psi\rangle$.

Any time one wishes to "bring a quantity to or from a phase", a helpful tool tends to be the Quantum Fourier Transform (QFT). Indeed, using the QFT in a clever way, and given the ability to apply high powers U in a controlled fashion, the Quantum Phase Estimation (QPE) algorithm yields the mapping

$$|\psi_j\rangle \mapsto |\lambda_j\rangle|\psi_j\rangle.$$

(Above, it is implicitly assumed ancillae initialized to $|0\rangle$ are added over the course of the mapping to store λ_j.) Brief reviews of the QFT and QPE are given in Section 1.1.

Exercise. Is there an *a priori* bound on how many bits λ_j requires to represent? What would be an ideal way to cram, say, an irrational λ_j into a finite-size register using bit strings?

Step 2: Eigenvalue processing. With the eigenvalue λ_j sitting in a register, we now simply coherently compute f via a classical algorithm to map

$$|\lambda_j\rangle|\psi_j\rangle \mapsto |f(\lambda_j)\rangle|\lambda_j\rangle|\psi_j\rangle. \tag{2}$$

Step 3: Eigenvalue reinsertion. This step is actually a sequence of steps, since we need to uncompute any auxilliary data we produced along the way.

As a first step, recall that our aim in Equation (1) was to compute $f(\lambda_i)|\psi_j\rangle$ (as opposed to $|f(\lambda_i)\rangle|\psi_j\rangle$). Having accomplished the latter, we simulate the former as follows: Conditioned on the register containing $|f(\lambda_j)\rangle$ we may produce state

$$|\lambda_j\rangle|\psi_j\rangle \mapsto |\lambda_j\rangle|\psi_j\rangle \left(\sqrt{1-f(\lambda_j)^2}|0\rangle + f(\lambda_j)|1\rangle\right),$$

where the last register is a single qubit. This is just a single-qubit rotation on the third register, conditioned on the contents of the first register.

Exercise. What is the difference between $f(\lambda_i)|\psi_j\rangle$ and $|f(\lambda_i)\rangle|\psi_j\rangle$?

Exercise. What range of $f(\lambda_i)$ is permitted in the equation above?

If we now measure the last register in the standard basis and obtain 1, we collapse our state onto essentially what we wanted:

$$f(\lambda_j)|\lambda_j\rangle|\psi_j\rangle|1\rangle.$$

All that is left is to "uncompute the garbage". Namely, we apply a Pauli X to $|1\rangle$ to return it to state $|0\rangle$, and we invert the phase estimation algorithm on $|\psi_j\rangle$ to map $|\lambda_j\rangle|\psi_j\rangle$ back to $|0\cdots 0\rangle|\psi_j\rangle$. We may now safely discard the registers which have been reinitialized back to all zeroes, obtaining Equation (1).

Applying the algorithm to general states $|b\rangle \in \mathbb{C}^N$. In Equation (1), we assumed $|b\rangle$ was an eigenstate of H for simplicity. To now extend this to arbitrary $|b\rangle$, we give Nature a "high 5" and thank it for being linear, because our analysis actually extends trivially due to the facts that (1) quantum mechanics is linear and (2) we may write

$$|b\rangle = \sum_{j=1}^{N} \beta_j |\psi_j\rangle,$$

since recall $\{|\psi_i\rangle\} \subseteq \mathbb{C}^N$ is an orthonormal basis for $\mathbb{C}^N$. Thus, by linearity the algorithm will correctly map

$$\sum_{j=1}^{N} \beta_j |\psi_j\rangle \mapsto \sum_{j=1}^{N} \beta_j f(\lambda_j)|\psi_j\rangle,$$

where we stress the right-hand side is *not* normalized as written.

Exercise. Why did we need to uncompute the garbage at the send of Step 3 earlier? How might omitting this uncomputation cause a problem when we move to the setting of general $|b\rangle$?

1.1 Aside: Brief review of the QFT and QPE

Since the procedure we've sketched crucially uses the QFT and QPE, let us briefly review how these components work. (We will only use them as black boxes in this lecture, but they are used sufficiently frequently to warrant a refresher.)

The Quantum Fourier Transform (QFT). Recall the *Fourier transform* is a unitary operation, meaning it is just a change of basis. A powerful change of basis it is, however, as it can be viewed as mapping the "time domain" to the "frequency domain". For this reason, it is ubiquitous in areas such as signal processing (meaning you can thank the Fourier transform for making mp3 sound files possible), and even allows one to multiply pairs of n-bit integers in essentially $O(n \log n)$ time (as opposed to $O(n^2)$ time via the grade school multiplication algorithm). The *Quantum* Fourier Transform (QFT) similarly finds key uses in quantum algorithms, particularly when one wishes to move a quantity from a register up into a phase, or vice versa.

We may completely specify the N-dimensional $\mathrm{QFT_N}$ via its action on the standard basis:

$$|j\rangle \xrightarrow{\mathrm{QFT_N}} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} |k\rangle.$$

As previously noted, observe that it lifts string j into the phase.

Exercise. Prove that $H = \mathrm{QFT}_2$. In other words, you have already been using the QFT.

A full recap of the QFT circuit would be distracting for the purposes of this lecture; we refer the reader to Chapter 5 of Nielsen and Chuang for a full exposition. We shall simply note that the basic implementation of the QFT requires quadratically many gates in the number of qubits; for us, this means a runtime of $O(\log^2 N)$, where recall N is our dimension.

Quantum Phase Estimation (QPE). With the QFT in hand, we may perform QPE. Specifically, we assume we have a black-box implementation of some unitary $U \in \mathcal{U}(\mathbb{C}^N)$, and an eigenvector $|\psi_j\rangle$ of U with eigenvalue $e^{2\pi i \phi_j}$ for some $\phi_j \in [0, 1)$ (without loss of generality). Our goal is to compute the n most significant bits of ϕ_j. Actually, we require a stronger assumption than just the ability to run U — we must be able to perform a controlled-U^k operation, which applies U k times to a target register, conditioned on a control register being set to $|1\rangle$. This is *not* a trivial assumption.

Exercise. Suppose $U \in \mathcal{U}((\mathbb{C}^2)^{\otimes n})$ has a polynomial-size quantum circuit implementation of 1- and 2-qubit gates. Is it true that for all such U, controlled-U^k can be implemented efficiently for $k \in \Theta(2^n)$? (Hint: Try to embed a brute force search algorithm over all assignments to a given 3-SAT formula $\phi : \{0,1\} \mapsto \{0,1\}$ into U^k for $k = 2^n$.)

The algorithm for QPE is now sufficiently simple that we include it for completeness:

1. Start with initial state $|\eta_0\rangle = |0^t\rangle|\psi_j\rangle$, for $t = n + \lceil \log(2 + \frac{1}{2\epsilon}) \rceil$. Here, ϵ will dictate the success probability of the procedure.
2. Applying $H^{\otimes n}$ to the first register, we obtain state $|\eta_1\rangle = \left(\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle \right) |\psi_j\rangle$.
3. Apply the controlled-U operation, with register 1 as control and register 2 as the target, to obtain

$$|\eta_2\rangle = \left(\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{2\pi i k \phi_j} |k\rangle \right) |\psi_j\rangle.$$

   Exercise. To what power do we need to raise U in this step in the worst case?

4. This expression now has our desired phase ϕ_j encoded in the phase; in fact, if ϕ_j can be encoded exactly using n bits, then the first register is just the QFT_{2^t} of $|2^t \phi_j\rangle$ (and in this case we could have set $t = n$). Therefore, applying $\mathrm{QFT}_{2^t}^\dagger$ yields $|\eta_3\rangle = |2^t \phi_j\rangle|\psi_j\rangle$, from which we can compute ϕ_j.

Recall that in general, we have no guarantee that ϕ_j can be expressed in n bits. This is intuitively why we require $t > n$ ancilla bits; an appropriate analysis shows that choosing $t = n + p$ yields failure probability at most $[2(2^p - 1)]^{-1}$.

Exercise. Suppose the input state to the QPE algorithm is not an eigenvector $|\psi_j\rangle$ of U, but rather an arbitrary state $|\psi\rangle \in \mathbb{C}^N$. Sketch the action of each step of the QPE algorithm on $|\psi\rangle$.

2 A quantum algorithm for linear systems of equations

We now show how to instantiate the algorithm sketch of Section 1 in the setting of solving linear systems $A\mathbf{x} = \mathbf{b}$. Let us assume for simplicity that A is Hermitian and full rank (the algorithm can be modified to work even without these assumptions). Then, recall from elementary Linear Algebra that the system has the unique solution $\mathbf{x} = A^{-1}\mathbf{b}$. But now note A^{-1} is an operator function $f(x) = x^{-1}$ applied to Hermitian operator A. Thus, we can try to apply the "eigenvalue surgery" technique of Section 1 to "manually invert" the eigenvalues of A. Before proceeding, we must discuss a key quantity known as the *condition number* of A, which is relevant to both classical and quantum linear systems solvers.

2.1 Condition numbers of matrices

Whereas in theory, any full rank matrix A can be inverted, *in practice* this cannot necessarily be done reliably. For some choices of A, the computation of A^{-1} is very sensitive to numerical error, and this is captured by the notion of *condition number*.

Formally, suppose we are only able to represent $\mathbf{b}$ via a numerical approximation, i.e. we numerically compute $\mathbf{b}' = \mathbf{b} + \epsilon$ for some error vector ϵ. We are interested in understanding how $A^{-1}\mathbf{b}'$ compares with $A^{-1}\mathbf{b}$. By linearity, this depends on how $A^{-1}\mathbf{b}$ compares with $A^{-1}\epsilon$ — if the former is not much larger than the latter, the error will be quite noticeable. So let us look at the ratio between these two, normalized by the ratio of $\mathbf{b}$ and ϵ themselves, and where we employ the Euclidean norm:

$$\frac{\left\| A^{-1}\epsilon \right\|_2}{\| A^{-1}\mathbf{b} \|_2} \cdot \frac{\| \mathbf{b} \|_2}{\| \epsilon \|_2} = \frac{\left\| A^{-1}\epsilon \right\|_2}{\| \epsilon \|_2} \cdot \frac{\| \mathbf{b} \|_2}{\| A^{-1}\mathbf{b} \|_2}.$$

To maximize this ratio over *all* $\mathbf{b}$ and ϵ, we maximize the left quantity (giving us $\left\| A^{-1} \right\|_\infty$ by definition of the spectral norm) and minimize the right quantity (giving us $\| A \|_\infty$). This worst case ratio is precisely how we define the condition number,

$$\kappa(A) := \left\| A^{-1} \right\|_\infty \| A \|_\infty .$$

Thus, $\kappa(A)$ roughly captures the worst-case sensitivity of A, over all choices of $\mathbf{b}$, to encoding errors in $\mathbf{b}$.

Exercise. In a previous lecture, we defined the spectral norm slightly differently — instead of dividing by $\| \epsilon \|_2$ as in $\left\| A^{-1}\epsilon \right\|_2 / \| \epsilon \|_2$ above, we required ϵ to be a unit vector. Prove that both definitions of the spectral norm are equivalent.

Exercise. Prove that the minimum of $\left\| A^{-1}\mathbf{b} \right\|_2$ over all vectors $\mathbf{b}$ is indeed $\| A \|_\infty$. Hint: One easy way to see this is to note that $\| A \|_\infty$ is the largest singular value of A.

Exercise. What is $\kappa(A)$ for any unitary A?

Exercise. What is $\kappa(A)$ for rank deficient A? How does this support the idea that A is by definition non-invertible?

2.2 Assumptions for the algorithm

The following assumptions are required for the algorithm (in addition to the non-crucial assumptions that A is Hermitian and full rank):

1. That $\| A \|_\infty = 1$, and hence $\left\| A^{-1} \right\|_\infty = \kappa(A)$. Under this assumption, once the postselection step passes, the algorithm is guaranteed to be correct. This assumption can be relaxed, in exchange for allowing further errors (i.e. we would need to also consider the "ill-conditioned" subspace of A).

2. An efficient unitary implementation of $|b\rangle \in \mathbb{C}^N$. This is treated as a genuine black box.

3. An efficient Hamiltonian simulation algorithm to implement unitary e^{iAt}. Recall here $A \in \mathrm{Herm}(\mathbb{C}^N)$, whereas efficient means with respect to the number of qubits, $O(\log N)$. This is a highly non-trivial assumption.

   Exercise. Prove that the following is false: For any Hermitian A and evolution time $t \in \mathbb{R}$, there is a quantum circuit with size polylog(N) simulating e^{iAt}. (Hint: Recall the equivalence with arbitrary unitary operators $U \in \mathcal{U}(\mathbb{C}^N)$, and apply a basic counting argument to show the latter cannot have poly-size circuits for all U.)

   Luckily, there are some fairly broad classes of Hamiltonians which we *can* simulate efficiently — these include s-sparse Hamiltonians A, which have two important properties: (1) At most s non-zero entries per row, and (2) there exists a poly-time classical algorithm which, given a row index $r \in [N]$, outputs the non-zero entries in row r. In this case, there exist quantum algorithms for simulating e^{iAt} with error at most ϵ_H (with respect to trace distance) in time $\widetilde{O}(\log(N)s^2t)$, where the tilde means slower growing terms are omitted for simplicity. Note this runtime has been simplified using the assumption here that $\| A \|_\infty \leq 1$.

2.3 The algorithm

For simplicity in the analysis, we shall make the following additional assumptions: (1) The state $|b\rangle$ can be prepared without error. (2) $A \succeq 0$, which means that since we assumed $\| A \|_\infty = 1$ and $\left\| A^{-1} \right\|_\infty = \kappa(A)$ in Section 2.2, we have that $\lambda_{\max}(A) = 1$ and $\lambda_{\min}(A) = 1/\kappa(A)$. (3) All eigenvalues $1/\kappa(A) \leq \lambda_j \leq 1$ of A require at most n bits to represent, for some integer $n > 0$. (This n will shortly play a role in terms of run-time.) (4) Simulating e^{iAt} can be done without error for any time $t \geq 0$ (with some associated cost, of course).

Exercise. Prove that if $A \succeq 0$, then $\| A \|_\infty = \lambda_{\max}(A)$ and $\left\| A^{-1} \right\|_\infty = \lambda_{\min}(A)$.

The algorithm now follows the sketch of Section 1:

1. Use the assumed black box to prepare state

$$|b\rangle = \sum_{j=1}^{N} \beta_j |\psi_j\rangle \in \mathbb{C}^N,$$

   where $|\psi_j\rangle$ are the eigenvectors of A with eigenvalues λ_j.

2. Apply QPE (for unitary e^{iA}) with an n-qubit ancilla to our state $|b\rangle$ to obtain

$$\sum_{j=1}^{N} \beta_j |2^n \lambda_j\rangle |\psi_j\rangle \in (\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^N.$$

3. Conditioned on the first register, rotate a new single-qubit ancilla as follows:

$$\sum_{j=1}^{N} \beta_j |2^n \lambda_j\rangle |\psi_j\rangle \left(\sqrt{1 - \frac{1}{\lambda_j^2 \kappa^2(A)}} |0\rangle + \left(\frac{1}{\lambda_j \kappa(A)} \right) |1\rangle \right) \in (\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^N \otimes \mathbb{C}^2. \tag{3}$$

Note that the third register is now entangled with the first two.

Exercise. Which of our assumptions guarantee that $1/\kappa(A) \leq 1/(\lambda_j \kappa(A)) \leq 1$, and hence that the amplitudes above are well-defined?

4. Apply the inverse of the QPE algorithm to the first two registers to obtain

$$\sum_{j=1}^{N} \beta_j |0^n\rangle |\psi_j\rangle \left(\sqrt{1 - \frac{1}{\lambda_j^2 \kappa^2(A)}} |0\rangle + \left(\frac{1}{\lambda_j \kappa(A)} \right) |1\rangle \right) \in (\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^N \otimes \mathbb{C}^2.$$

We may hence now discard the first register safely, having uncomputed it.

Exercise. Why does applying the QPE^{-1} operation produce the state above, given that the third register in Equation (3) is entangled with the first two? (Hint: With respect to which basis did we define the action of QPE?)

5. Measure the third register in the standard basis, postselect on outcome 1, and then discard the third register to obtain a state

$$\sum_{j=1}^{N} \beta_j \left(\frac{1}{\lambda_j \kappa(A)} \right) |\psi_j\rangle \equiv \sum_{j=1}^{N} \beta_j \left(\frac{1}{\lambda_j} \right) |\psi_j\rangle \propto A^{-1} |b\rangle \in \mathbb{C}^N.$$

Exercise. The use of "equivalence" $\equiv$ above is not an accident — why are the first two expressions equivalent for the purposes of quantum information?

Exercise. Prove that in Step 5, the probability of obtaining outcome 1 is at least $1/\kappa^2(A)$.

We conclude, by the exercise above, that the expected number of repetitions to obtain $|1\rangle$ in Step 5 is $O(\kappa^2(A))$. Using the technique of quantum amplitude amplification, this can be reduced to $O(\kappa(A))$ repetitions; we omit the details of this step here, but assume it has been implemented in our runtime analysis below.

Runtime. Let T_b denote the number gates required to prepare $|b\rangle$. The total runtime we have hence accrued is $\widetilde{O}(\kappa(A)(T_b + ts^2 \log(N)))$, where t is the time we run Hamiltonian simulation for in the QPE algorithm. Since we assumed all eigenvalues of A require at most n bits to represent exactly, this means QPE will simulate e^{iAt} for $t \in O(2^n)$ (in the notation of QPE from Section 1.1, there we would set $t = n$). Thus, the total runtime scales as

$$\widetilde{O}(\kappa(A)(T_b + 2^n s^2 \log(N))),$$

assuming we wish to compute the *exact* answer proportional to $A^{-1}|b\rangle$, and assuming all subroutines we have called run without error. Recalling that the number of qubits is $O(\log(N))$ for A an $N \times N$ matrix, this means that if $n \in O(\log \log N)$ (i.e. QPE approximates phases up to additive inverse polynomial error in the number of qubits), we would get runtime $\widetilde{O}(\kappa(A)(T_b + s^2 \log^2(N)))$. In the regime $\kappa(A), T_b, s \in \text{polylog}(N)$, this is exponentially faster than classical algorithms explicitly solving the entire $N \times N$ system.

Of course, we cannot assume $n \in O(\log \log N)$ *a priori* (since we know just about nothing about the spectrum of A), but a more involved algorithm and analysis can be used to show that if one wishes to output state $|\widetilde{x}\rangle$ satisfying $\| |x\rangle - |\widetilde{x}\rangle \|_2 \leq \epsilon$, it suffices to set $t \in O(\kappa(A)/\epsilon)$, obtaining a final runtime of

$$\widetilde{O}(\kappa(A)(T_b + \kappa(A)^2 s^2 \log(N))/\epsilon).$$

This more advanced algorithm can also handle e.g. non-Hermitian A, dropping the assumption that $\kappa(A) = \| A^{-1} \|_\infty$, and relaxing the simplifications that all subroutines run perfectly (e.g. they may fail with some probability).

On optimality. In order to maintain an efficient runtime in the number of qubits, $O(\log N)$, the error ϵ permitted above must be at most inverse polynomial in the number of qubits. This is generally sufficient for the purposes of BQP to distinguish between YES and NO cases of promise problems. Nevertheless, it is natural to ask: *Can the runtime of this algorithm be improved to something* polylogarithmic *in $\kappa(A)$ and ϵ?*

It turns out that if the goal is to *classically estimate* the quantity $\langle x|\Pi|x\rangle$ for some efficiently implementable projector Π, then this is highly unlikely. (For example, it would imply BQP = PSPACE if a runtime in $\kappa(A)^{1-\delta}$ for $\delta > 0$ a constant would be possible. More on such ideas in Section 3.) However, if we relax the requirements so that the goal is to produce the quantum state $|x\rangle$ (i.e. this is now a different beast altogether, as the output is no longer classical), then an improvement *is* possible. Namely, recent advances in Hamiltonian simulation techniques have led to an improved linear systems solver which has runtime polynomial in $\log(1/\epsilon)$, as opposed to $1/\epsilon$ here. In other words, exponentially good approximations to $|x\rangle$ can be prepared efficiently quantumly (assuming A is well-conditioned, etc).

Exercise. Can you see where the polynomial dependence on $1/\epsilon$ in the runtime must come in, i.e. which step of the algorithm we described here cannot be run to full precision in general, even assuming all subroutines work perfectly?

Exercise. Why does the ability to prepare $|x\rangle$ efficiently to within 2^{-n} Euclidean distance not allow one to estimate the classical value $\langle x|\Pi|x\rangle$ to within 2^{-n} additive error efficiently (i.e. why does the improved algorithm not contradict the known lower bounds)? (Hint: Imagine we could even prepare $|x\rangle$ perfectly.)

3 A BQP-complete problem: Matrix inversion

At the end of the Section 2.3, we touched on the topic of optimality. It turns out that task of *matrix inversion*, which is the core subroutine of the linear systems algorithm, characterizes the complexity of BQP, in that it is a BQP-complete problem. Let us formalize it as follows.

Definition 1 (Matrix inversion problem (MI))**.** *The promise problem $MI = (A_{\text{yes}}, A_{\text{no}}, A_{\text{inv}})$ is as follows.*

- *Input: An $O(1)$-sparse invertible Hermitian matrix $A \in \text{Herm}(\mathbb{C}^N)$ satisfying $\kappa^{-1}(A) \preceq A \preceq I$ for $\kappa(A) \in \text{polylog}(N)$, specified via a polynomial-time Turing machine M which, given any row index $r \in [N]$ of A, outputs the $O(1)$ non-zero entries of A.*
- *Output: Let $|\widetilde{x}\rangle \propto A^{-1}|0^N\rangle$ be a unit vector, and $\Pi = |1\rangle\langle 1| \in \mathcal{L}(\mathbb{C}^2)$ a projector onto the first qubit of $|\widetilde{x}\rangle$.*
  - *(Completeness) If $\text{Tr}(\Pi|\widetilde{x}\rangle\langle\widetilde{x}|) \geq 2/3$, output YES.*
  - *(Soundness) If $\text{Tr}(\Pi|\widetilde{x}\rangle\langle\widetilde{x}|) \leq 1/3$, output NO.*
  - *(Invalid) Else, output YES or NO arbitrarily.*

Theorem 2. *MI is BQP-complete under polynomial-time many-one reductions.*

Proof. Containment of MI in BQP follows immediately from the linear systems algorithm. We show BQP-hardness. Let $V = V_m \cdots V_1$ be a BQP circuit acting on n qubits, and of size $m \in \text{poly}(n)$, for $n \in \log(N)$. Without loss of generality, we may assume m is a power of 2. We give a polynomial-time many-one reduction to a matrix A as in the definition of MI.

Let us begin with an idea that almost works. Define

$$U = \sum_{t=0}^{m-1} |t+1\rangle\langle t| \otimes V_{t+1} + \sum_{t=m}^{2m-1} |t+1 \bmod 2m\rangle\langle t| \otimes V_{2m-t}^\dagger \in \mathcal{U}((\mathbb{C}^2)^{\otimes \log m} \otimes (\mathbb{C}^2)^{\otimes n}),$$

where recall we implicitly assume V_t (which is a 2-qubit gate) is tensored with an identity on all $n-2$ qubits it does not act on.

Exercise. Show that U is indeed unitary.

Exercise. Show that $U^m|0^{\log m}\rangle|0^n\rangle = |0^{\log m}\rangle V|0^n\rangle$. Thus, measuring the first qubit of the second register simulates measuring the output qubit of the BQP computation V.

Exercise. For what values of k is $U^k|0\rangle|0^n\rangle = |0\rangle|0^n\rangle$ necessarily?

We would like to embed U in a matrix A, so that A^{-1} can be nicely expressed via U. The natural idea is to recall the Maclaurin series expansion

$$\frac{1}{1-x} = \sum_{l=0}^{\infty} x^l$$

which holds for $|x| < 1$. Applying this to U, we find

$$(I-U)^{-1} = \sum_{l=0}^{\infty} U^l.$$

Exercise. Technically, the last equation above is *not* correct (nevertheless, it provides the correct intuition; we will correct the error later). What requirement for the Maclaurin series expansion have we violated?

Defining $A = I - U$ and $|\widetilde{x}\rangle \propto A^{-1}|0^{\log m+n}\rangle$ a unit vector, we have

$$|\widetilde{x}\rangle \propto \sum_{l=0}^{\infty} U^l|0\rangle^{\log m}|0^n\rangle \propto |0\rangle|0^n\rangle + |1\rangle V_1|0^n\rangle + \cdots + |m\rangle V_m \cdots V_1|0^n\rangle.$$

Exercise. Again, technically, the last statement above is not exactly correct — not all terms on the right hand side will be equally weighted. Which two terms should be weighted slightly less than the others? (For pedagogical reasons, we nevertheless work with the statement above, as the discrepancy will not affect our high-level discussion other than to complicate the amplitudes involved.)

Since each term $|j\rangle V_j \cdots V_1|0^n\rangle$ is a unit vector, this means

$$|\widetilde{x}\rangle = \frac{1}{\sqrt{m+1}} \left(|0\rangle|0^n\rangle + |1\rangle V_1|0^n\rangle + \cdots + |m\rangle V_m \cdots V_1|0^n\rangle\right).$$

Thus, if we measure the first register of $|\widetilde{x}\rangle$ in the standard basis, with probability $1/(m+1)$ we collapse onto state $|m\rangle V|0^n\rangle$. Let E_m (respectively, E_0) be the event we postselect in the first register on $|m\rangle$ (respectively, $|1\rangle, \ldots, |m-1\rangle$). Then, conditioned on E_m, the probability of measuring 1 in the first qubit of register reveals

the answer of the BQP computation with probabilty at least 2/3; otherwise, conditioned on E_0, we may assume[1] without loss generality that V rejects with probability 1. This implies (where recall $\Pi = |1\rangle\langle 1|$ is a single qubit projector onto the output qubit of the second register):

- If V denotes a YES instance, then $\langle \widetilde{x}|\Pi|\widetilde{x}\rangle \geq \frac{2}{3(m+1)}$.
- If V denotes a NO instance, then $\langle \widetilde{x}|\Pi|\widetilde{x}\rangle \leq \frac{1}{3(m+1)}$.

Exercise. Prove the probability bounds claimed above.

This is almost what we want, except for three things: (1) A must be $O(1)$-sparse. (2) We should have probability bounds 2/3 (completeness) and 1/3 (soundness) for MI. (3) A should be an invertible Hermitian matrix $A \in \mathrm{Herm}(\mathbb{C}^N)$ satisfying $\kappa^{-1}(A) \preceq A \preceq I$ for $\kappa(A) = \mathrm{polylog}(N)$.

Exercise. Prove that U is $O(1)$-sparse. Conclude that A is also $O(1)$-sparse.

Exercise. How can one boost the probability bounds from $2/(3(m+1))$ and $1/(3(m+1))$ to 2/3 and 1/3? (Hint: Two tricks can be used together. First, use error reduction for BQP. Second, what happens if before constructing U, we modify the circuit V by appending M time steps in which nothing is done?)

Exercise. Is A necessarily invertible?

Obtaining the third desired property is slightly trickier. We begin by bringing the condition number of A down to polylogarithmic in N, which we now discuss.

Exercise. What is the worst case condition number $\kappa(I - U)$ for a unitary U?

Exercise. Show that defining $A = I - \frac{1}{2}U$ has $\kappa(A) \in O(1)$. Where in the previous analysis does this choice of A cause problems, however?

To circumvent the problems in the previous exercise, we choose $A = I - e^{-1/m}U$ for scalar $e^{-1/m}$.

Exercise. Prove that $\kappa(A) \in O(m)$ for the new definition of A. (Hint: You only need to use the fact that U is unitary, not the specific definition of U. Also, use the fact that for normal operators A, the singular values of A are $\{|\lambda(A)|\}$ (why?).)

Exercise. Show that A is now invertible.

Exercise. Rerun the analysis with our new choice of A and show that we are still able to distinguish between YES and NO cases for V with constant completeness-soundness gap.

With a bounded condition number in place, to make A Hermitian we apply the same trick for dealing with non-Hermitian matrices U as in the linear systems algorithm (which we shall again omit; roughly, one creates the anti-block diagonal matrix

$$\begin{pmatrix} 0 & I - e^{-1/m}U \\ I - e^{-1/m}U^\dagger & 0 \end{pmatrix},$$

[1] One way to achieve this is to modify V so that in all but the last time step, V sets the output qubit to $|0\rangle$, and only in time step m does V perform a CNOT to copy over its answer to the output qubit.

which is clearly Hermitian, and whose inversion can be shown to also yield a similar final result to inverting $I - U$.) Finally, note that the requirement $\kappa^{-1} \preceq A \preceq I$ does not make sense until we map A to a Hermitian matrix, since the "$\preceq$" partial order is defined on the set of Hermitian matrices (just as the usual "$\leq$" total order is defined on the set of real numbers, but not on complex numbers). Once A is Hermitian, however, a global rescaling of A will suffice to satisfy this final constraint, which we also omit. $\square$