1 Introducing the block-encoding

The original motivation for quantum computers is simulation of quantum systems. One of the simplest such simulation tasks we could ask for is that of Hamiltonian simulation.

Problem (Hamiltonian simulation)**.** Let H be a Hamiltonian made up of m Pauli terms, meaning that

$$H = \sum_{a=1}^{m} \lambda_a E_a \text{ where } E_a \text{ is a tensor product of Pauli matrices.}$$

Find an algorithm implementing a unitary U close to $e^{-\mathrm{i}Ht}$, so that $\|U - e^{\mathrm{i}Ht}\| \leq \varepsilon$.

If you haven't seen Hamiltonians before: how you should think about H is as a sum of interaction terms E_a, where λ_a dictates the strength of the interaction. Original solutions proceeded by using Trotter approximations: for r large enough,

$$e^{-\mathrm{i}Ht} \approx (e^{-\mathrm{i}E_1 t/r} e^{-\mathrm{i}E_2 t/r} \cdots e^{-\mathrm{i}E_m t/r})^r,$$

However, this solution is far from optimal, notably because implementing this approximation requires $\mathrm{poly}(1/\varepsilon)$ gate complexity. Improved algorithms [BCCKS17; LC17; LC19] eventually developed into the framework I will present now [GSLW19].

This framework proceeds by:

1. Defining a type of quantum circuit called a "block-encoding";
2. Showing that, given λ_a and E_a, we can construct an efficient block-encoding of H;
3. Showing that we can get a block-encoding of (an approximation of) $e^{-\mathrm{i}Ht}$ with few uses of the block-encoding to H;
4. Using this block-encoding to apply our approximation to a state.

1.1 Block-encodings

Definition 1.1 (Variant of [GSLW19, Definition 43], [Ral20, Definition 1])**.** Given $A \in \mathbb{C}^{r\times c}$, we say $U \in \mathbb{C}^{d\times d}$ is a Q-block encoding of A if U is implementable with $\mathcal{O}(Q)$ gates and

$$B_{\mathrm{L},1}^\dagger U B_{\mathrm{R},1} = A, \tag{1}$$

where $B_{\mathrm{L},1} \in \mathbb{C}^{d\times r}, B_{\mathrm{R},1} \in \mathbb{C}^{d\times c}$ are the first r and c columns of the identity matrix. Equivalently,

$$U = \begin{pmatrix} A & \cdot \\ \cdot & \cdot \end{pmatrix}, \tag{2}$$

where $\cdot$ denotes arbitrary elements of U. We denote $\Pi_{\mathrm{L}} = B_{\mathrm{L},1}B_{\mathrm{L},1}^\dagger$, $\Pi_{\mathrm{R}} = B_{\mathrm{R},1}B_{\mathrm{R},1}^\dagger$ to be the corresponding projections onto the spans of $B_{\mathrm{L},1}$ and $B_{\mathrm{R},1}$, respectively.

We'll often consider d, r, and c as powers of two so that we can write everything in terms of qubits. The equation in Definition 1.1 then becomes

$$(\langle 0|^{\otimes a_{\mathrm{L}}} \otimes I) U (|0\rangle^{\otimes a_{\mathrm{R}}} \otimes I) = A. \tag{3}$$

In the literature, you'll often see block-encodings defined with an accuracy parameter ε and a rescaling parameter α, allowing for approximation:

$$\|A/\alpha - B_{\mathrm{L},1}^\dagger U B_{\mathrm{R},1}\| \leq \varepsilon.$$

In these lecture notes, we usually drop the (ε, α) parameters, and instead say that we have a 0-accurate 1-scaled block-encoding of $\tilde{A}/\alpha$ for $\|\tilde{A}/\alpha - A/\alpha\| \leq \varepsilon/\alpha$. Definitions sometimes also allow $B_{\mathrm{L},1}$ and $B_{\mathrm{R},1}$ to be arbitrary isometries [GSLW19, Definition 11]; this is not any more general, since then $B_{\mathrm{L}}^\dagger U B_{\mathrm{R}}$ is a block-encoding in the sense above, where $B_{\mathrm{L}}, B_{\mathrm{R}} \in \mathbb{C}^{d\times d}$ are unitary completions of $B_{\mathrm{L},1}$ and $B_{\mathrm{R},1}$.

We can view the block-encoding as a generalization of a unitary quantum circuit.

Lemma 1.2. *A quantum circuit implementing the unitary U with Q gates is a Q-block encoding of U.*

In the way that we apply a circuit implementing U to perform the map $|\psi\rangle \mapsto U|\psi\rangle$, a block-encoding of A can be used to perform the map $|\psi\rangle \mapsto A|\psi\rangle$, with some chance of failure. This allows us to perform more general types of linear algebraic operations than what unitary circuits offer.

Lemma 1.3. *Given $U \in \mathbb{C}^{d\times d}$, a Q-block encoding of $A \in \mathbb{C}^{r\times c}$, and a state $|\psi\rangle \in \mathbb{C}^c$, there is a quantum circuit with $\mathcal{O}(Q)$ gates that produces the state $\frac{A|\psi\rangle}{\|A|\psi\rangle\|}$ with probability $\|A|\psi\rangle\|^2$.*

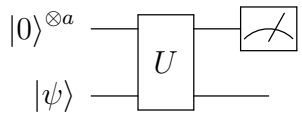

Figure 1: A basic block-encoding circuit. If U is a block-encoding of the matrix $A \in \mathbb{C}^{r\times r}$, then provided the outcome of the measurement on the first wire is $|0\rangle^{\otimes a}$, then the output of the circuit is $A|\psi\rangle$.

Proof. The circuit is shown in Fig. 1: we can take the state $|\psi\rangle$ and add a_{R} qubits initialized to $|0\rangle$. Then, we apply the block-encoding U and measure the first a_{L} qubits. If they all have outcome 0, then by Eq. (3), the resulting state is $A|\psi\rangle$. This occurs with probability $\|A|\psi\rangle\|^2$. □

1.2 Extensibility properties of block-encodings

The question now becomes: when can we produce an efficient block-encoding of a matrix? In fact, we can re-cast the problem of Hamiltonian simulation as follows: given 1-block encodings of $\{E_a\}_{a\in[m]}$ defining the Hamiltonian $H = \sum_{a=1}^m \lambda_a E_a$, can we get a block-encoding of (an approximation of) $e^{-\mathrm{i}Ht}$? Block-encodings enjoy several *extensibility properties*: that is, given block-encodings of A and B, we can get block-encodings of AB and $c_0 A + c_1 B$, whenever this makes sense. This will allow us to get a block-encoding of H/α for some rescaling constant α.

Lemma 1.4 (Multiplication of block-encodings)**.** *Let U and V be Q_U- and Q_V-block-encodings of $A \in \mathbb{C}^{r\times s}$ and $B \in \mathbb{C}^{s\times t}$, respectively. Then we can construct a $(Q_U + Q_V)$-block encoding of AB.*

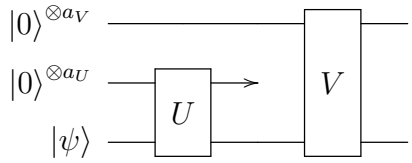

Figure 2: If U is a block-encoding of A and V is a block-encoding of B, then this circuit is a block-encoding of AB, shown being applied to input $|\psi\rangle$. Here, a_U and a_V are the padding needed for the respective block-encodings.

Proof. The circuit implementing AB is shown in Fig. 2. We can see that this is a block-encoding of AB by inspection, as this is a composition of two of the circuits in Fig. 1. □

We can construct block-encodings of linear combinations of block-encodings using the *Linear Combination of Unitaries* (LCU) algorithm.

Lemma 1.5 (Linear combination of block-encodings)**.** *Let $U^{(i)}$ be a $Q^{(i)}$-block-encoding of $A^{(i)} \in \mathbb{C}^{r\times c}$ for all $i = 0, \ldots, k-1$. Then we can construct a $(k + \sum_{i=0}^{k-1} Q^{(i)})$-block encoding of $\sum \alpha_i U^{(i)}$, for $\alpha_i \in \mathbb{C}$ such that $\sum|\alpha_i| \leq 1$.*

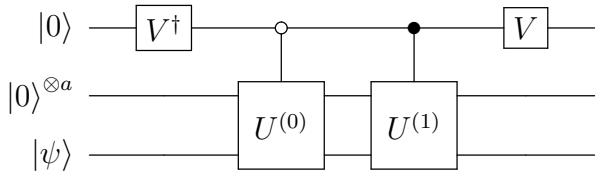

Figure 3: If $U^{(1)}$ and $U^{(2)}$ are block-encodings of $A^{(1)}$ and $A^{(2)}$, then this circuit is a block-encoding of $|V_{0,0}|^2 A^{(0)} + |V_{0,1}|^2 A^{(1)}$, shown being applied to input $|\psi\rangle$. Here, the gate blocks containing $U^{(0)}$ and $U^{(1)}$ denote conditioning on $|1\rangle$ and conditioning on $|1\rangle$.

Proof. First, consider when taking the linear combination of $k = 2$ block-encodings. The circuit implementing a linear combination is shown in Fig. 3. The controlled-$U^{(0)}$ and controlled-$U^{(1)}$ apply the unitary

$$\begin{pmatrix} U^{(0)} & \\ & I \end{pmatrix}\begin{pmatrix} I & \\ & U^{(1)} \end{pmatrix} = \underbrace{\begin{pmatrix} U^{(0)} & \\ & U^{(1)} \end{pmatrix}}_{(|0\rangle\langle 0|)\otimes U^{(0)} + (|1\rangle\langle 1|)\otimes U^{(1)}} \tag{4}$$

So, the full circuit is performing

$$\underbrace{\begin{pmatrix} V_{0,0}I & V_{0,1}I \\ V_{1,0}I & V_{1,1}I \end{pmatrix}^\dagger \begin{pmatrix} U^{(0)} & \\ & U^{(1)} \end{pmatrix}\begin{pmatrix} V_{0,0}I & V_{0,1}I \\ V_{1,0}I & V_{1,1}I \end{pmatrix}}_{(V^\dagger|0\rangle\langle 0|V)\otimes U^{(0)} + (V^\dagger|1\rangle\langle 1|V)\otimes U^{(1)}.} \tag{5}$$

The top-right corner of this matrix, which is where the block-encoding should be, equals

$$|V_{0,0}|^2 U^{(0)} + |V_{1,0}|^2 U^{(1)}.$$

So, for any non-negative real α_0, α_1 summing to one, we can find some one-qubit unitary V whose first column is $\sqrt{\alpha_0}, \sqrt{\alpha_1}$, giving the desired block-encoding. If, say, α_0 was

negative, we could use the circuit for $|\alpha_0|$, but use a controlled unitary of $-U^{(0)}$ instead of $U^{(0)}$ to negate it in the block-encoding.

The general version is of the following form. First, if U is a block-encoding of A then so is $I \otimes U$, so without loss we can pad the dimension until all $U^{(i)}$'s are all the same size, $d \times d$. Second, without loss we can pad our linear combination until k is a power of two by adding $U^{(i)} = I$ and $\alpha_i = 0$ to the linear combination. Let $V \in \mathbb{C}^{k \times k}$ be a unitary such that

$$V|0\rangle = \sum_{i=0}^{k} \sqrt{|\alpha_i|}\,|i\rangle$$

and let $U \in \mathbb{C}^{kd \times kd}$ be the unitary

$$\sum_{i=0}^{k-1} (|k\rangle\langle k|) \otimes (\frac{\alpha_i}{|\alpha_i|}) U^{(i)}).$$

Then $(V^\dagger \otimes I)U(V \otimes I)$ is a block-encoding of $\sum \alpha_i U^{(i)}$. The cost of applying V is $\mathcal{O}(k)$, and assuming that the cost of applying the controlled version of $U^{(i)}$ is only a constant factor larger than the cost of applying $U^{(i)}$ itself, the cost of U is $\mathcal{O}(k + \sum_i Q^{(i)})$. □

We will need a final assertion about block-encodings. This assumes that the cost of a circuit is equal to the cost of its inverse.

Lemma 1.6. *If U is a Q-block encoding of A, then $U^\dagger$ is a Q-block encoding of A.*

1.3 The "fundamental theorem" of block-encodings

These extensibility theorems are powerful: one might notice that we can combine them to get block-encodings of polynomials of A. "Polynomials of A" has a clear meaning when A is Hermitian: for a function $f : \mathbb{R} \to \mathbb{C}$, $f(A)$ is defined to be the function that applies f to the eigenvalues of A: for $A = \sum \lambda_i u^{(i)} (u^{(i)})^\dagger$ the unitary eigendecomposition of A, $f(A) = \sum f(\lambda_i) u^{(i)} (u^{(i)})^\dagger$.

Definition 1.7 ([GSLW19, Definition 16])**.** Let $f : \mathbb{R} \to \mathbb{C}$ be even or odd, and let $A \in \mathbb{C}^{r \times c}$ have SVD $A = \sum_{i \in [\min(r,c)]} \sigma_i u_i v_i^\dagger$. Then we define

$$f^{(\mathrm{SV})}(A) = \begin{cases} \sum_{i \in [\min(r,c)]} f(\sigma_i) u_i v_i^\dagger & f \text{ is odd} \\ \sum_{i \in [c]} f(\sigma_i) v_i v_i^\dagger & f \text{ is even} \end{cases}$$

where σ_i is defined to be zero for $i > \min(r, c)$.

When $f(x) = p(x)$ is an even or odd polynomial, $p^{(\mathrm{SV})}(A)$ can be written as a polynomial in the expected way, e.g. if $p(x) = x^2 + 1$, $p^{(\mathrm{SV})}(A) = A^\dagger A + I$ and if $p(x) = x^3 + x$, $p^{(\mathrm{SV})}(A) = AA^\dagger A + A$.

Definition 1.8. A degree-d polynomial $p \in \mathbb{C}[x]$ is "achievable" if there is an explicit way to convert a block-encoding of A to a block-encoding of $p^{(\mathrm{SV})}(A)$.

Corollary 1.9 (Corollary of the extensibility properties)**.** *Polynomials of the form $p(x) = \sum_{k=0}^{d} a_k x^k$ are achievable, provided that $\sum |a_k| \leq 1$ and p is odd or even.*

With this definition in hand, we are now ready to state the main result of the QSVT framework, which states that all bounded polynomials with real coefficients are achievable.

Theorem 1.10 ([GSLW19, Theorem 17 and Corollary 18])**.** *If a polynomial with real coefficients $p \in \mathbb{R}[x]$ is even or odd and satisfies $|p(x)| \leq 1$ for all $x \in [-1, 1]$, then it is achievable.*

This is the most we could hope for, since we could never get a block-encoding of $p^{(\mathrm{SV})}(A)$ if $p^{(\mathrm{SV})}(A)$ has norm greater than one; this implies the boundedness constraint.

1.4 Wielding our tool

Hamiltonian simulation gives a nice view into how to use block-encodings and QSVT. As we discussed before, we can construct a m-block-encoding of $H/(\sum|\lambda_i|)$. Rescaling time, we can take $\sum|\lambda_i| = 1$ without loss of generality. Our goal is to get an (approximate) block-encoding of $f(H)$, where

$$f(x) = \exp(-\mathrm{i}xt) = \cos(tx) - \mathrm{i}\sin(tx). \tag{6}$$

Since $\cos(tx)$ and $\sin(tx)$ are bounded even and odd functions, respectively, we can find good polynomial approximations of them. That is, we can find c and s such that, for all $x \in [-1, 1]$,

$$|c(x) - \cos(tx)| \leq \varepsilon \qquad |s(x) - \sin(tx)| \leq \varepsilon$$

By Theorem 1.10, we can get block-encodings of $c^{(\mathrm{SV})}(H)$ and $s^{(\mathrm{SV})}(H)$. By Lemma 1.5, we can get a block-encoding of $\frac{1}{2}(c^{(\mathrm{SV})}(H) - \mathrm{i}s^{(\mathrm{SV})}(H)) \approx \frac{1}{2}e^{-\mathrm{i}Ht}$. This is enough if we wish to apply it to an input state $|\psi\rangle$, but to decrease the failure probability we can remove the $\frac{1}{2}$ through *oblivious amplitude amplification*, which can be done with QSVT.

I haven't yet discussed gate complexity or the error analysis, but we will see that the running time of the whole algorithm is dictated by how small the degree can be of polynomials approximating $\cos(tx)$ and $\sin(tx)$. Up to constant factors of wiggle room in the parameters, the number of times one needs to apply the block-encoding for H is equal to this degree, and we get optimal algorithms for Hamiltonian simulation by choosing the optimal polynomial approximations.

References

[BCCKS17] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. "Exponential improvement in precision for simulating sparse hamiltonians". In: *Forum of Mathematics, Sigma* 5 (2017), e8. DOI: 10.1017/fms.2017.2. arXiv: 1312.1414 [quant-ph] (page 1).

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. "Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics". In: *Proceedings of the 51st ACM Symposium on the Theory of Computing (STOC)*. ACM, June 2019, pp. 193–204. DOI: 10.1145/3313276.3316366. arXiv: 1806.01838 (pages 1, 2, 4, 5).

[LC17] Guang Hao Low and Isaac L. Chuang. "Optimal hamiltonian simulation by quantum signal processing". In: *Physical Review Letters* 118.1 (Jan. 2017), p. 010501. DOI: 10.1103/PhysRevLett.118.010501. arXiv: 1606.02685 [quant-ph] (page 1).

[LC19] Guang Hao Low and Isaac L. Chuang. "Hamiltonian simulation by qubitization". In: *Quantum* 3 (July 2019), p. 163. DOI: 10.22331/q-2019-07-12-163 (page 1).

[Ral20] Patrick Rall. "Quantum algorithms for estimating physical quantities using block encodings". In: *Physical Review A* 102.2 (Aug. 2020), p. 022408. DOI: 10.1103/physreva.102.022408. arXiv: 2004.06832 [quant-ph] (page 1).