Strengths and Weaknesses of Quantum Computing

Charles H. Bennett
IBM Research

Ethan Bernstein
UC Berkeley

Gilles Brassard
Université de Montréal

Umesh V. Vazirani
UC Berkeley

December 1, 1994

1 Introduction

Quantum computational complexity is an exciting new area that touches upon the foundations of both theoretical computer science and quantum physics. Quantum mechanical computers appear to violate a foundational principle of complexity theory, the extended Church-Turing thesis, which asserts that any reasonable model of computation can be simulated by a probabilistic Turning Machine (PTM) with at most a polynomial factor simulation overhead. Two polynomially equivalent [Yao93] quantum physical models of computation, the quantum Turing Machine (QTM) [Deu85] and quantum computational networks [Deu89] have been proposed which seem to be more powerful than a PTM and yet still seem to be physically reasonable. In terms of complexity classes, the class **BQP**, of languages that can be accepted in polynomial time with bounded error on a QTM, seems to be larger than the class **BPP**.

One requirement for a physically reasonable model of computation is that it must be a discrete model in that it does not require arbitrary accuracy in it's specification. One might be concerned that the power of quantum computers is due to some violation of this principle. After all, it is well known that a standard model of a computer, when allowed to compute with exponentially precise real numbers, can solve an **NP**–completeproblem. It has been shown previously [BV93] that any language in **BQP** is accepted in polynomial time and with bounded error by a QTM where each constant of is specified to only a polynomial number of bits of accuracy. A polynomial number of bits seems quite reasonable for the workspace of a computer since it only requires a polynomial amount of extra machinery. However, a polynomial number of bits for machine specification may not be physically reasonable since it might require an exponential amount of work to set each amplitude to lie within one of exponentially many ranges. The reason is that forcing a physical quantity (like say a voltage) into one of 2^k possible ranges seems to be 2^k times as hard as forcing it into one of two ranges, rather than just k times as hard. In this paper, we prove that any language in **BQP** is accepted by a QTM which need only be specified with a logarithmic number of bits of accuracy, so that each constant is chosen from only a polynomial number of ranges.

A second concern of this paper is the power of quantum computers. Several results [BV93, Sim94, Sho94] have used the quantum computer's ability to take large Fourier transforms to give evidence that quantum computers are more powerful than PTM's. The most impressive among these is a result by Shor [Sho94], which shows that the the discrete log and factoring problems, both widely believed to be classically intractable, can be solved in polynomial time on a quantum computer. In view of these successes it is tempting to conjecture that the quantum computer is so powerful that it can simulate the full power of classical nondeterminism, by solving some NP-complete problem, or even the full power of $\sharp$**P**, by solving a problem such as computing the permanent. We present evidence against these conjectures, not by refuting them directly (which would entail a resolution of the $\mathbf{P} = \mathbf{PSPACE}$ question), but rather by showing that that there is a type of task that **BQP** machines have difficulty with. We show that a **BQP** machine running with an oracle O on a particular input is sensitive to at most polynomially many oracle bits. In other words, for all but polynomially many of the oracle bits, if we flip a single oracle bit to get oracle O', and run the same machine with the same input but with O' instead of O, then it will compute the same output. This allows us to prove that with a random oracle, quantum computers cannot simulate classical nondeterminism: with probability 1 when R is a random oracle, there are problems in $\mathbf{NP}^R$ that do not belong to $\mathbf{BQP}^R$.

Among the obstacles that had to be overcome while constructing a universal QTM [BV93, Yao93] is that given two QTMs M_1 and M_2, it is not clear how to obtain a QTM M whose action corresponds to M_1 followed by M_2. A lemma from [BV94] shows that this "dovetailing" is always possible. Another problem along these lines arises when a QTM tries to use another QTM as a subroutine. The subroutine must not leave around any bits beyond it's computed answer, because computational paths with different residual information can no longer interfere. This is easily achieved for deterministic subroutines since any classical deterministic computation can be carried out reversibly so that only the input and the answer remain. However, this leaves open the more general question of whether a **BQP** machine can be used as a subroutine. Our final result in this paper, is to show how any **BQP** machine can be modified into a tidy **BQP** machine whose final superposition consists almost entirely of a tape configuration containing just the input and the single bit answer. Since these tidy **BQP** machines can be safely used as subroutines, this allows us to show that $\mathbf{BQP} = \mathbf{BQP}^{\mathbf{BQP}}$.

The three results in this paper use similar techniques, which depend on the fact that the computation of any QTM is unitary and hence preserves angles between superpositions.

A more extensive survey of quantum computation literature will be included in a later version.

2 Are Quantum Turing Machines Discrete?

An important requirement for a physically reasonable model of computation is that any constants in the specification should only be required to be accurate up to a small degree of accuracy. Bernstein and Vazirani [BV93] and Yao [Yao93] showed that any QTM can be simulated by a universal QTM which has a single state with transition amplitudes other than $0, 1, -1$. Moreover, it was shown that this single non-trivial rotation can be assumed to be 2-dimensional and can be specified only to within $O(T)$ bits of accuracy to simulate T steps of computation. Lipton [Lip94] pointed out that in order to claim that a model of computation is truly discrete, it is necessary that any constants in the specification should only be required to be accurate up to $O(logT)$ bits of accuracy. This is because a constant that is accurate up to k bits can lie in one of 2^k distinct ranges, and even the task of verifying that the machine meets the specifications seems to require effort that is proportional to 2^k. The following theorem shows that we need only $O(\log T)$ bits of precision in the specification of each of the complex numbers defining the single-step transition function of a **BQP** machine.

Definition A QTM M is a **BQP** machine if M halts in polynomial time and for every input x, M either accepts with probability at least $2/3$ or at most $1/3$. We say that $x \in L(M)$ or equivalently $M[x] = 1$ if M accepts x with probability at least $2/3$ and we say that $x \notin L(M)$ or $M[x] = 0$ otherwise.

Theorem 1 *Given any* **BQP** *machine M that runs in time $p(n)$, then there is a constant c with the following property. If $\widehat{M}$ is any QTM with the same state set and tape alphabet as M, such that for every transition amplitude λ of M, the corresponding transition amplitude $\hat{\lambda}$ of $\widehat{M}$ satisfies $|\hat{\lambda} - \lambda| \leq 2^{-c \log n}$, then for all n-bit inputs x, $\widehat{M}[x] = M[x]$.*

Proof. Let such a QTM $\widehat{M}$ be given for some constant c.

Then, let x be an n-bit string, let U be the unitary transformation specified by M, and let $\hat{U}$ be the unitary transformation specified by $\widehat{M}$.

We will first prove that for any superposition $|\phi\rangle$, $U|\phi\rangle$ is close to $\hat{U}|\phi\rangle$

Now, let

$$|\phi\rangle = \sum_i \alpha_i |c_i\rangle$$

be some superposition. Then,

$$U|\phi\rangle - \hat{U}|\phi\rangle = \sum_i \left(\sum_{j \in P(i)} (\lambda_{j,i} - \hat{\lambda}_{j,i})\alpha_j \right) |c_i\rangle$$

where $P(i)$ is the set of j such that configuration $|c_j\rangle$ can lead to $|c_i\rangle$ in a single step of M or $\widehat{M}$. Note that for any i, $|P(i)| \leq c'$ where c' is some constant depending only on the number of states and tape symbols used in the definition of M.

So, we have

$$\begin{array}{rcl} |U|\phi\rangle - \hat{U}|\phi\rangle|^2 & = & \sum_i \left|\sum_{j \in P(i)} (\lambda_{j,i} - \hat{\lambda}_{j,i})\alpha_j\right|^2 \\ & \leq & \sum_i c' \sum_{j \in P(i)} \left|(\lambda_{j,i} - \hat{\lambda}_{j,i})\alpha_j\right|^2 \\ & \leq & c' \sum_i \sum_{j \in P(i)} |\lambda_{j,i} - \hat{\lambda}_{j,i}|^2 |\alpha_j|^2 \\ & \leq & c' \sum_i \sum_{j \in P(i)} 2^{-c \log n} |\alpha_j|^2 \\ & \leq & c' 2^{-c \log n} \sum_i \sum_{j \in P(i)} |\alpha_j|^2 \\ & \leq & c' 2^{-c \log n} c' \sum_j |\alpha_j|^2 \\ & = & (c')^2 2^{-c \log n} \end{array}$$

Now, if c is large compared to c', then this will be very small. We will use this to prove that M and $\widehat{M}$ accept the same n-bit strings.

Let x be any n-bit string x. Then, we will show that the final superpositions of M and $\widehat{M}$, $U^{p(n)}|x\rangle$ and $\hat{U}^{p(n)}|x\rangle$, are close together.

$$\begin{array}{rcl} \hat{U}^{p(n)}|x\rangle & = & \hat{U}^{p(n)-1}(\hat{U}|x\rangle) \\ & = & \hat{U}^{p(n)-1}(U|x\rangle + |E_1\rangle) \\ & = & \hat{U}^{p(n)-1}(U|x\rangle) + \hat{U}^{p(n)-1}|E_1\rangle \\ & = & \hat{U}^{p(n)-2}(U^2|x\rangle) + \hat{U}^{p(n)-2}|E_2\rangle + \hat{U}^{p(n)-1}|E_1\rangle \\ & = & U^{p(n)}|x\rangle + \sum_{i=1}^{p(n)} \hat{U}^{p(n)-i}|E_i\rangle \end{array}$$

where each $|E_i\rangle$ has $|E_i|^2 \leq (c')^2 2^{c \log n}$. Therefore,

$$\begin{array}{rcl} |U^{p(n)}|x\rangle - \hat{U}^{p(n)}|^2 & \leq & |\sum_{i=1}^{p(n)} \hat{U}^{p(n)-i}|E_i\rangle|^2 \\ & \leq & p(n) \sum_{i=1}^{p(n)} |\hat{U}^{p(n)-i}|E_i\rangle|^2 \\ & = & p(n) \sum_{i=1}^{p(n)} |E_i|^2 \\ & \leq & p^2(n)(c')^2 2^{c \log n} \end{array}$$

Now if

$$c > \frac{2\log p(n) + 2\log c' + \log 3}{\log n}$$

then this is less than 1/3, and so $M[x] = \widehat{M}[x]$ ∎

3 QTMs can't always simulate non-determinism

We now show that an oracle QTM machine M^O run on a particular input x is sensitive to at most polynomially many oracle answers. In other words, for all but polynomially many of the oracle bits, if we flip a *single* oracle bit to get oracle O', then $|P[M^{O'}$ accepts $x]$ - $P[M^O$ accepts $x]| \leq \epsilon$. Note that a **BQP** machine can detect whether a large number of oracle answers in a certain pattern are flipped, which is not possible for a classical **BPP** machine — this is the basis of the positive results on quantum computation.

Theorem 2 *Given* **BQP** *machine M that runs in time $p(n)$, oracle O, and n-bit string x, then for all but $2p^2(n)/\epsilon$ strings y, the oracle O' where $O'(z) = O(z)$ iff $z \neq y$, gives $|P[M^{O'}$ accepts $x]$ - $P[M^O$ accepts $x]| \leq \epsilon$.*

Proof. Let S be a set of k strings.

Consider the $p(n)$ intermediate superpositions produced during the computation of M^O on input x. Each superposition consists of some configurations which query a string in S and some which do not. We could express this by saying that $|\phi_t\rangle$, the time t superposition of M^O, can be written as

$$|\phi_t\rangle \;=\; \alpha_{t,y_1}|\psi_{t,y_1}\rangle \;+\; \alpha_{t,y_2}|\psi_{t,y_2}\rangle \;+\; \ldots \;+\; \alpha_{t,y_k}|\psi_{t,y_k}\rangle \;+\; \alpha_t|\psi_t\rangle$$

where $|\psi_{t,i}\rangle$ is a superposition of configurations about to query string y_i, and $|\psi_t\rangle$ is a superposition of configurations which are querying strings outside of S or are not querying the oracle at all. Then since

$$\sum_i |\alpha_{t,y_i}|^2 \leq 1$$

we must have

$$\sum_{i,t} |\alpha_{t,y_i}|^2 \leq p(n)$$

Therefore, there must be some $y \in S$ such that

$$\sum_t |\alpha_{t,y}|^2 \leq \frac{p(n)}{k}$$

Now, let's think about flipping this query, and running $M^{O'}$ on x where $O'(z) = O(z)$ iff $z \neq y$. Let U be the unitary transformation corresponding to M^O, and U' the unitary transformation corresponding to $M^{O'}$. Then U and U' are almost identical; they differ only in columns corresponding to configurations that query y. Moreover, we know that y is queried with very small squared amplitude on each step of the computation of M^O. By linearity, changing the answer on y at one step of the computation does not perturb the final superposition by much. We have to be careful however, since changing the answer to y at the first step of the computation can change the amplitude with which y is queried at subsequent steps. Potentially, this could cause these amplitudes to increase exponentially. The main point in the rest of the proof is to show that this does not occur.

The superpositions produced by M^O can now be written more compactly as

$$|\phi_t\rangle \;=\; \alpha_{t,y}|\psi_{t,y}\rangle \;+\; \alpha_t|\psi_t\rangle$$

where $|\psi_{t,y}\rangle$ is a superposition of configurations which query y and $|\psi_t\rangle$ is a superposition of configurations which do not query y. Then, since $U|\phi_t\rangle = |\phi_{t+1}\rangle$, we have

$$U\left(\alpha_{t,y}|\psi_{t,y}\rangle \;+\; \alpha_t|\psi_t\rangle\right) \;=\; \alpha_{t,y}U|\psi_{t,y}\rangle \;+\; \alpha_t U|\psi_t\rangle \;=\; \alpha_{t+1,y}|\psi_{t+1,y}\rangle \;+\; \alpha_{t+1}|\psi_{t+1}\rangle$$

Therefore,

$$\alpha_t U|\psi_t\rangle \;=\; \alpha_{t+1,y}|\psi_{t+1,y}\rangle \;+\; \alpha_{t+1}|\psi_{t+1}\rangle \;-\; \alpha_{t,y}U|\psi_{t,y}\rangle$$

Now, since U and U' only differ on configurations which query y, they perform the same transformation on ψ_t. Therefore

$$\begin{aligned}
U'|\phi_t\rangle &= U'\left(\alpha_{t,y}|\psi_{t,y}\rangle \;+\; \alpha_t|\psi_t\rangle\right) \\
&= \alpha_{t,y}U'|\psi_{t,y}\rangle \;+\; \alpha_t U'|\psi_t\rangle \\
&= \alpha_{t,y}U'|\psi_{t,y}\rangle \;+\; \alpha_t U|\psi_t\rangle \\
&= \alpha_{t,y}U'|\psi_{t,y}\rangle \;+\; \alpha_{t+1,y}|\psi_{t+1,y}\rangle \;+\; \alpha_{t+1}|\psi_{t+1}\rangle \;-\; \alpha_{t,y}U|\psi_{t,y}\rangle \\
&= |\phi_{t+1}\rangle \;+\; \alpha_{t,y}U'|\psi_{t,y}\rangle \;-\; \alpha_{t,y}U|\psi_{t,y}\rangle
\end{aligned}$$

This means that

$$U'|\phi_t\rangle = |\phi_{t+1}\rangle + |E_{t+1}\rangle$$

where $|E_{t+1}|^2 \leq 2|\alpha_{t,y}|^2$. Therefore, the final superposition of $M^{O'}$ can be written as

$$\begin{aligned}
(U')^{p(n)}|\phi_0\rangle &= (U')^{p(n)-1}|\phi_1\rangle \;+\; (U')^{p(n)-1}|E_1\rangle \\
&= (U')^{p(n)-2}|\phi_2\rangle \;+\; (U')^{p(n)-2}|E_2\rangle \;+\; (U')^{p(n)-1}|E_1\rangle \\
&= |\phi_{p(n)}\rangle \;+\; \textstyle\sum_{t=1}^{p(n)}(U')^{p(n)-t}|E_t\rangle
\end{aligned}$$

Now finally

$$\begin{aligned}
|\textstyle\sum_{t=1}^{p(n)}(U')^{p(n)-t}E_t|^2 &\leq p(n)\textstyle\sum_{t=1}^{p(n)}|(U')^{p(n)-t}E_t|^2 \\
&= p(n)\textstyle\sum_{t=1}^{p(n)}|E_t|^2 \\
&\leq p(n)\textstyle\sum_{t=1}^{p(n)}2|\alpha_{t,y}|^2 \\
&\leq \tfrac{2p^2(n)}{k}
\end{aligned}$$

Therefore, if $k > 2p^2(n)/\epsilon$, then the final superpositions of $M^{O'}[x]$ and $M^{O}[x]$ have inner product greater than $1-\epsilon$. Therefore $|P[M^{O'}$ accepts $x]$ - $P[M^{O}$ accepts $x]| \leq \epsilon$. ∎

This theorem can be used to prove the following series of oracle results.

Theorem 3 *Relative to a random oracle R, with probability 1,* $\mathbf{NP}^R$ *includes languages not in* $\mathbf{BQP}^R$*.*

Theorem 4 *Relative to a random permutation oracle π, with probability 1,* $(\mathbf{NP} \cap \mathbf{co{-}NP})^\pi$ *includes languages not in* $\mathbf{BQP}^\pi$*.*

4 Using BQP machines as subroutines

Much of the power of quantum computers comes from the fact that interference can only happen between two configurations that differ only in the contents of cells which are currently active (i.e. where the read/write head is). This also creates a difficulty, since in order to simulate one QTM with another, we must make sure that there is no extraneous information on the tape that prevents interference. This effect shows up when we wish to make a subroutine call inside a QTM. If the subroutine is a **BQP** machine, then the tape of the **BQP** machine might contain intermediate results produced as a by-product of the computation. In general, these intermediate results will prevent interference between different branches of the computation. In this section, we first show how to modify any **BQP** machine into a tidy **BQP** machine whose final superposition consists almost entirely of tape configurations containing just the input and a single bit answer. The final result showing that **BQP** programs can be called as subroutines will follow from this fact together with a careful consideration of some timing issues, and a lemma about legal dovetailing of QTMs [BV94].

Theorem 5 *Let M be a* **BQP** *machine. Then there is a* **BQP** *machine $\widehat{M}$ that on input x and ϵ produces a final superposition in which the amplitude α of the configuration $|x, \epsilon, M[x]\rangle$ satisfies $|\alpha|^2 \geq 1-\epsilon$. Moreover the running time of $\widehat{M}$ on x, ϵ is $O(\log 1/\epsilon)$ times the running time of M on x.*

Proof. We start by boosting the success probability of M. This is done by running k copies of M in separate parts of the tape. The superposition produced is the result of tensoring the k superpositions produced by the individual copies. The majority of the k answers is now written on the tape, producing superposition $\sum_c \alpha_c |c, maj_c\rangle$. It is not hard to show that if this majority bit is measured then it is a random variable with exactly the same distribution as the majority of k independent coin flips, where each coin flip has bias equal to the success probability of a single run of M. Therefore, $\sum_{maj_c=1} |\alpha_c|^2 \geq 1-\epsilon/2$ provided that $k \geq b\log(1/\epsilon)$ for some constant b.

Now, we run k copies of the reverse of M on the separate parts of the tape. If we hadn't written the majority answer on the tape, then this would exactly reverse our earlier computation, once again giving us the input configuration $|x,\epsilon\rangle$. This means that, if our superposition were $\sum_c \alpha_c|c, M[x]\rangle$, we would compute $|x,\epsilon,M[x]\rangle$. Therefore, applying the fact that $\langle U\phi|U\phi'\rangle = \langle\phi|\phi'\rangle$ for any unitary transformation U and superpositions $|\phi\rangle$ and $|\phi'\rangle$, the result when operating on $\sum_c \alpha_c|c, maj_c\rangle$ has inner product at least $1-\epsilon/2$ with $|x,\epsilon,M[x]\rangle$. Therefore the amplitude α of $|x,\epsilon,M[x]\rangle$ satisfies $|\alpha|^2 \geq 1-\epsilon$. ∎

Theorem 6 *Let M be a* **BQP** *machine, and $L \in$* **BQP**. *Then $M^L \in$* **BQP**.

References

[Ben82] Benioff, P., "Quantum Hamiltonian models of Turing machines" *Journal of Statistical Physics*, vol. 29, 1982, pp. 515 – 546.

[BBBSS92] Bennett, C. H., Bessette, F., Brassard, G., Salvail, L. and Smolin, J., "Experimental quantum cryptography", *Journal of Cryptology*, vol. 5, no. 1, 1992, pp. 3 – 28.

[BBE92] Bennett, C. H., Brassard, G. and Ekert, A. K., "Quantum cryptography", *Scientific American*, October 1992, pp. 50 – 57.

[BV93] Bernstein, E. and Vazirani, U., "Quantum complexity theory", *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, 1993, pp. 11 – 20.

[BV94] Bernstein, E. and Vazirani, U., "Quantum complexity theory", final version of

[BB92a] Berthiaume, A. and Brassard, G., "The quantum challenge to structural complexity theory", *Proceedings of 7th Annual IEEE Symposium on Structure in Complexity Theory*, June 1992, pp. 132 – 137.

[BB92b] Berthiaume, A. and Brassard, G., "Oracle quantum computing", *Proceedings of the Workshop on Physics and Computation — PhysComp '92*, October 1992, IEEE Press, pp. 195 – 199.

[BB94] Berthiaume, A. and Brassard, G., "Oracle quantum computing", *Journal of Modern Optics*, to appear.

[Deu85] Deutsch, D., "Quantum theory, the Church–Turing principle and the universal quantum computer", *Proceedings of the Royal Society*, London, vol. A400, 1985, pp. 97 – 117.

[Deu86] Deutsch, D., in *Quantum Concepts in Space and Time* (R. Penrose and C. Isham, editors), Clarendon Press, Oxford, 1986, pp. 215 – 225.

[Deu89] Deutsch, D., "Quantum computational networks", *Proceedings of the Royal Society*, London, vol. A425, 1989, pp. 73 – 90.

[DJ92] Deutsch, D. and Jozsa, R., "Rapid solution of problems by quantum computation", *Proceedings of the Royal Society*, London, vol. A439, 1992, pp. 553 – 558.

[Fey82] Feynman, R., "Simulating physics with computers", *International Journal of Theoretical Physics*, vol. 21, nos. 6/7, 1982, pp. 467 – 488.

[Fey86] Feynman, R., "Quantum mechanical computers", *Foundations of Physics*, vol. 16, no. 6, 1986, pp. 507 – 531. (Originally appeared in *Optics News*, February 1985.)

[Lip94] Lipton, R., Personal communication, 1994.

[Lan91] Landauer, R., "Information is physical", *Physics Today*, vol. 44, May 1991, pp. 23 – 29.

[Llo93] Lloyd, S., "A potentially realizable quantum computer", *Science*, vol. 261, 17 September 1993, pp. 1569 – 1571.

[Sho94] Shor, P., "Algorithms for quantum computation: Discrete log and factoring", FOCS 1994.

[Sim94] Simon, D., "On the power of quantum computation", FOCS 1994.

[Unr94] Unruh, W., "Decoherence and Quantum Computers: a Problem", IEEE PhysComp-94.

[Yao93] Yao, A., "Quantum circuit complexity", *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science*, 1993.

[Zur84] Zurek, W., "Reversibility and stability of information processing systems", *Physical Review Letters*, vol. 53, 1984, pp. 391 – 394.