Computing in the quantum world

Christian Paquin

Université de Montréal
paquin@iro.umontreal.ca

10 July 1998

Abstract

The current computing model is based on the laws of classical physics. But the world is not classical, it follows the laws of quantum mechanics. A quantum computer is a model of computation based on quantum mechanics. It has been proved that such a model more powerful than its classical counterpart, meaning that it can do the same computations as a classical computer (in approximately the same time) but there exist some problems for which the quantum computer is much faster. In this paper I will explain what is quantum information, why a quantum computer would be useful, what are the problems to build a quantum computer and how it will work.

1 What is quantum information?

Computer science and quantum mechanics are two of the most fabulous theories of the 20th century. They merge nicely to form a new field: quantum computation theory, in which we ask ourselves what a computer could do if it followed the laws of quantum mechanics.

Before we study the theory of quantum computation, we must define the formalism in which we are going to work. All we need to represent quantum states and transformations are vectors and operators in a complex vector space. Here are some definitions and laws of quantum mechanics.
Note: we will be using the braket notation to represent quantum states. A *ket* $|\psi\rangle$ is noting more than a vector $\vec{\psi}$ in some vector space.

Definition 1 (Hilbert space) An *Hilbert space* is a complete complex vector space. If a complex vector space has finite dimension n (the only interesting case here) then it is complete, hence it is an Hilbert space. We note the Hilbert space by $\mathcal{H}^n$.

The state of a quantum system is represented by a vector in some vector space $\mathcal{H}^n$. We will only study the case where the state is *pure* (as opposed to *mixed* [1], but we don't need to worry about those in this paper) so this vector representation is sufficient.

A classical bit has two possible values, either 0 or 1. Its quantum analogue will be in a superposition of those two values. The quantum bit is a quantum system represented by a vector in $\mathcal{H}^2$.

Let $\{|0\rangle, |1\rangle\}$ be an orthonormal basis of $\mathcal{H}^2$. So any vector $|\psi\rangle \in \mathcal{H}^2$ can be written as a linear combination $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where the complex numbers α and β are called the amplitudes of the state.

We can now define the basic information unit of quantum information theory.

Definition 2 (Quantum bit) A *quantum bit*, or *qubit* is a normalized vector $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ in $\mathcal{H}^2$, i.e. $||\alpha||^2 + ||\beta||^2 = 1$.

The normalization condition will be important later on. The basis vectors represent the classical bits 0 and 1. We can think of a qubit as being in superposition of those two values, meaning that it's both 0 *and* 1 at the same time. It may be difficult to imagine a qubit, but quantum mechanics allows such a behavior (we will see later some physical examples of qubits).

We will now explain how to extract information from a qubit. A classical bit can be measured without trouble, but the qubit is more sensitive: measuring it may destroy it. We will restrict the measurement to the standard basis, which means that the only answer we can get is either the classical bit 0 or the classical bit 1.

Law 1 (Qubit measurement) The measurement of a qubit in the state $\alpha|0\rangle + \beta|1\rangle$ yields the classical value 0 with probability $||\alpha||^2$ or the classical value 1 with probability $||\beta||^2$. If the answer 0 is observed, the state *collapses* (i.e. transforms) to $|0\rangle$; if the answer is 1, the state becomes $|1\rangle$.

When we measure a qubit, we actually project the vector on one of the basis vectors chosen randomly depending on the amplitudes. A qubit must be normalized for the probabilities to sum to one. We could measure in a different (orthonormal) basis, but the standard $\{|0\rangle, |1\rangle\}$ basis is sufficient for our needs. We can see that measuring a qubit twice will give the same answer. Indeed, suppose we measure the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and that we obtain the answer 0, so the new state is $|0\rangle$. Now if we measure again, the measurement law states that we will obtain 0 with probability 1.

To do useful computation, we will need more than one qubit. To represent a group of qubits, we must *combine* their vector spaces using a mathematical tool called a tensor product. If we group qubits in $\mathcal{H}^m$ and qubits in $\mathcal{H}^n$ then the resulting space will be $\mathcal{H}^{mn} = \mathcal{H}^m \otimes \mathcal{H}^n$. The definition of a quantum register follows.

Definition 3 (Quantum register) A *quantum register* of n qubits is a normalized vector in $\mathcal{H}^{2^n}$. It can be expressed as a linear combination of the basis vectors $|0\rangle, \ldots, |2^n - 1\rangle$:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$$

with the restriction that $\sum_{i=0}^{2^n-1} ||\alpha_i||^2 = 1$.

As a notational convenience, we often omit the tensor product symbol. These notations are all equivalent:

$$|u\rangle \otimes |v\rangle \equiv |u\rangle|v\rangle \equiv |uv\rangle.$$

To give an example of the tensor product of two qubits, suppose we have two vectors $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, $|\phi\rangle = \gamma|0\rangle + \delta|1\rangle \in \mathcal{H}^2$. The resulting product state is

$$|\psi\phi\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$$

and lies in $\mathcal{H}^4$. The standard basis of $\mathcal{H}^4$ is $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. The measurement law can be generalized to many qubits. In this case, measuring $|\psi\phi\rangle$ yields two classical bits depending on the probabilities (for example, yields 00 with probability $||\alpha\gamma||^2$).

To do computation, we will need something to operate on the qubits. What we need are unitary operations defined as follows.

Definition 4 (Unitary transformation) A *unitary transformation* is an unitary operator acting on a vector space $\mathcal{H}^n$. Unitarity means that the following properties are satisfied:

$$\begin{array}{lll} 1) & U : |\psi\rangle \longrightarrow |\phi\rangle & |\psi\rangle, |\phi\rangle \in \mathcal{H}^n \\ 2) & \langle\phi|\psi\rangle = \langle U\phi|U\psi\rangle & \forall_{|\psi\rangle,|\phi\rangle \in \mathcal{H}^n} \\ 3) & \exists_{U^{-1}} \ni UU^{-1} = \mathbb{1} & \end{array}$$

where $\langle u|v\rangle$ is the inner product in $\mathcal{H}^n$.

Law 2 (Evolution) The evolution of a quantum system $|\psi\rangle$ must be unitary.

An operator U of $\mathcal{H}^n$ can be represented by an $n \times n$ matrix $\mathcal{M}(U)$. Then an operator is unitary if its corresponding matrix is unitary, i.e. if $\mathcal{M}(U)^{-1} = \mathcal{M}(U)^\dagger$ (where $\mathcal{M}(U)^\dagger$ is the conjugate transpose of $\mathcal{M}(U)$).

One very useful transformation is the one qubit Walsh-Hadamard transform. We define a transformation by its action on the basis vectors.

$$H : \begin{array}{l} |0\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{array}$$

How does H transform a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$? By linearity of $\mathcal{H}^2$ we have

$$\begin{aligned} H|\psi\rangle &= \alpha H|0\rangle + \beta H|1\rangle \\ &= \tfrac{\alpha}{\sqrt{2}}(|0\rangle + |1\rangle) + \tfrac{\beta}{\sqrt{2}}(|0\rangle - |1\rangle) \\ &= \tfrac{\alpha+\beta}{\sqrt{2}}|0\rangle + \tfrac{\alpha-\beta}{\sqrt{2}}|1\rangle \end{aligned}$$

The corresponding matrix is

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ if } |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and we can check that it is unitary.

When we group qubits together to form a quantum register, we can calculate the resulting vector by applying the tensor product on the qubits. But if a unitary transformation is applied to the state, then it may happen that we can't separate the state, i.e. we can't represent it by a tensor product of qubits.

Definition 5 (Entanglement) We say that a quantum register is *entangled* if it can't be written as a tensor product of its parts.

If a quantum system is entangled, it means that there are correlations between the subsystems. What we do with one part of the system will influence the other part. Entanglement is very useful for computation as we will see later.

To give an example of an entangled quantum system, consider the following state in $\mathcal{H}^4$:

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).$$

This state is entangled because we cannot find two pure states $|\psi_1\rangle$ and $|\psi_2\rangle$ such that $|\Psi^-\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$. Entanglement has strange properties. Suppose that $|\Psi^-\rangle$ represents the state of two qubits, one of which is on earth and the other one is on a planet near Alpha Centauri. Since the qubits are entangled, their individual states cannot be written as vectors in $\mathcal{H}^2$ (but this is not important here).

If we measure our qubit (the one on earth), we will obtain a random answer (0 or 1), but we will be sure that if someone measure the other qubit, he will obtain the opposite answer. Although the answer we receive is random, the other answer is always its opposite. This strange "non-local" property led Einstein to criticize quantum mechanics in the famous EPR [2] paper.

2 Why would a quantum computer be useful?

2.1 What is a quantum computation?

Now that we have introduced the basic definitions and laws of quantum information theory, we can explain how to compute using quantum states and quantum

operations. A quantum computation works as follows; we prepare a quantum register in a known state, we apply quantum gates on the register, i.e. we apply unitary operations on some qubits in a precise order, and we measure the final state of the register (at the end) to learn its content. It is important to note that a quantum computation is probabilistic in nature, because although the evolution of the states is deterministic, the measurement step gives probabilistic answers.

One important result is that every computable function can be implemented by unitary transformations. This implies that the quantum computer can do whatever the classical computer does. Also, there is no significant increase in the time (number of steps) required to compute those functions with a quantum computer.

To give an example of a quantum computation, suppose we want to implement a fair coin toss. We will do a one qubit computation. We first prepare our qubit in the state $|0\rangle$. We then apply the Walsh-Hadamard transform on the qubit, yielding the superposition

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

We then measure the qubit in the standard basis to obtain the answer 0 with probability 1/2 or 1 with probability 1/2. The result we obtain is a *true* random number. It would be impossible to generate such a number using a classical, deterministic computer.

2.2 Some limitations

Quantum mechanics imposes limitations on what we can do with qubits. Two important results are presented.

Theorem 1 (No accurate measure) Given an arbitrary qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, we can't learn the amplitudes by measuring the qubit.

This means that if we have a qubit $|\psi\rangle$ in an unknown state, there is no way we can tell that this qubit is in the state, say $1/2|0\rangle + \sqrt{3/4}|1\rangle$. One thing we can do is to distinguish between orthogonal states, meaning that if we know that the qubit is either in the state $|\psi\rangle$ or in the orthogonal state $|\phi\rangle$ (such that $\langle\phi|\psi\rangle = 0$) then we could apply a measure to tell in which one of *those* two states the qubit is.

Theorem 2 (No cloning) Given an arbitrary qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, there does not exist an operator A and a state $|a\rangle$ such that $A(|\psi\rangle \otimes |a\rangle) = |\psi\rangle \otimes |\psi\rangle$.

This implies that we can't clone (or copy) arbitrary quantum states. We will omit the proof of those theorems because they are not necessary for this paper.

2.3 Quantum parallelism

We have seen that a qubit can be in a superposition of the classical bits 0 and 1. Now suppose we have a function $f : \{0,1\} \to \{0,1\}$ and we want to compute its outputs. Classically, we have to compute $f(0)$ and $f(1)$ separately. If we use quantum computation, we can do this in one step, by computing $f(|0\rangle + |1\rangle) = f(|0\rangle) + f(|1\rangle)$ (f acts on $\mathcal{H}^2$, the qubit is not normalized to simplify the notation). The two values were computed in parallel. This can be generalized to a superposition of any number of states and gives an exponential speedup over classical computation. The problem is that we can't extract both answers. If we measure the resulting qubit, we will obtain the answer $f(|0\rangle)$ or $f(|1\rangle)$ at random.

2.4 Quantum interference

What makes quantum computation so powerful is that the different parallel computations can interfere, leading some answers to become more likely than others. Suppose we have the unitary transformation U defined by

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$

If we apply this transformation on $|1\rangle$ and then measure, we obtain 0 or 1 with equal probability because

$$U|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

But if we apply this transformation twice (without measuring in between) we are sure to obtain 0 if we measure because

$$\begin{aligned} UU|1\rangle &= U\tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \tfrac{1}{\sqrt{2}}(\tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) + \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) \\ &= \tfrac{1}{2}(|0\rangle - |1\rangle + |0\rangle + |1\rangle) = |0\rangle. \end{aligned}$$

We also have that $UU|0\rangle = -|1\rangle$. Applying U once yields complete random but applying it twice gives a deterministic answer: the negation of the input (up to a constant factor, called global phase). This is the work of quantum interference. In many quantum algorithms, interference plays a crucial role. The "good" computational paths interfere constructively and the "bad" ones interfere destructively so they will not be encountered. Interference, together with entanglement and parallelism are the weapons that give the quantum computer all his power.

2.5 Shor's algorithm

In 1994, Peter Shor [3] gave a quantum algorithm to factor integers efficiently (i.e. in polynomial time). This discovery drew a lot of attention to the field of quantum computation because the factoring problem is at the core of modern cryptography. Indeed, there is no known way to factor large integers efficiently on a classical computer. In 1994, 1600 computers working together over a period of 8 months, succeeded to factor a 129-digit number. Later, a 130-digit number was factorized in a shorter time, but still required an incredible amount of work.

At this rate, factoring a 2000-digit number would be unimaginable. As Vazirani puts it [4], "Even if you imagine that every particle in the Universe is a [classical] computer and was computing at full speed for the entire life of the Universe, that would be insufficient to factor that number." Many cryptographic systems base their security on the assumption that the factoring problem is difficult. But Shor's algorithm could break such systems very effectively. This was the first practical problem which hinted the superiority of quantum computer.

2.6 Grover's algorithm

Suppose we have a function $f : \{0,1\}^n \to \{0,1\}$ such that $\exists_x f(x) = 1$ and $\forall_{y \neq x} f(y) = 0$. If we want to find the x such that $f(x) = 1$, classically we must choose inputs and compute the function until we find x. In complexity theory, we say that this takes a time in $\Theta(n)$[1]. Grover [5] developed a quantum algorithm that solve this problem in time $\Theta(\sqrt{n})$. This is not an exponential speedup, but the gain is substantial considering that this algorithm has many applications. Indeed, we can use this algorithm to search for an item in an unsorted database or to find collisions of a function. Many other algorithms that use Grover's as a subroutine were thereafter created.

2.7 Quantum teleportation

Quantum mechanics allows more than just computing functions. In 1993, Bennett *et al.* [6] discovered what is now called quantum teleportation. They proved that Alice can "teleport" an unknown qubit to Bob by sending only two classical bits, provided they share entanglement prior to the experiment. Recall that Alice can't just obtain the amplitude and send them to Bob. Teleportation works as follows. First, Alice and Bob have to share parts of an entangled pair (like $|\Psi^-\rangle$). Then, all Alice has to do is to entangle her part of the entangled pair with the qubit she wants to teleport (say $|\psi\rangle$) and measure both particles she has in hands. She then communicates her two classical measurement results to Bob, who can finally transform his part into the original state $|\psi\rangle$. Telepor-

[1] An algorithm is in $\Theta(n)$ if its number of steps is bounded between $c_1 n$ and $c_2 n$ (for some constants c_1,c_2) when $n \to \infty$.

tation was experimentally implemented in 1997 [7]. Brassard *et al.* [8] gave a simple quantum circuit that implements teleportation.

2.8 Quantum cryptography

Shor's algorithm poses a threat to the security of some cryptographic systems. The venue of a quantum computer would put a lot of secret data in danger. Ironically, quantum mechanics allows to communicate with unconditional security. The scheme of quantum cryptography was invented by Bennett and Brassard [9] who built on the work of Wiesner[10]. The technique is used to exchange a secret key that can be used for cryptographic purposes. The idea is to use non-orthogonal quantum states to encode bits. If a spy tries to eavesdrop on a data exchange, he will disturb the quantum states, revealing his presence. Many prototypes have been built and are working properly (see [11]).

3 Can we build a quantum computer?

3.1 Qubit?

To perform quantum computation we need qubits. But where do we find qubits? Fortunately, many elements in nature have the behavior required to act as qubits. For example, a qubit could be encoded in the polarization of a photon, where any two perpendicular directions could serve as a basis for the qubit state. Another example is the spin of an electron: the states spin-up and spin-down may represent $|0\rangle$ and $|1\rangle$ respectively. Quantum mechanics says that the spin of an electron can be in superposition of being up and down, so this would make a good qubit.

3.2 Errors

Until now, we supposed that all the preparations, transformations and measurements of qubits could be done reliably (i.e. without errors). Unfortunately, a quantum computation is doomed to introduce errors. Typically, a quantum system can't be totally isolated from its environment. This (slight) interaction causes the qubits to entangle themselves with the environment. This correlation damages the qubits, we say that they decohere. To this we must add the technical problem of applying transformation which form a continuum. If we transform the qubit into $(\alpha+\epsilon_1)|0\rangle+(\beta+\epsilon_2)|1\rangle$ instead of $\alpha|0\rangle+\beta|1\rangle$, the small errors build up into larger ones and this can ruin the computation.

3.3 Error correction

In classical computers, we have a way to deal with errors. With error correction techniques, we can detect and correct errors. This is done by introducing

redundancy in the data, which is easy because we can read and copy bits. But qubits can be neither read or copied. Fortunately, error correcting codes were developed for qubits [12, 13]. The idea is to encode one qubit with many qubits, introducing the redundancy within the entanglement of the register. One could say that entangled information is hidden from the environment. Destroying one qubit is not enough to corrupt the state of the entire register. Recent results showed that if the error probability per step of computation is lower than a certain threshold, we can store and compute with quantum information for an unlimited time. See Preskill [14] for a nice review.

4 How will a quantum computer work?

Just a few years ago, many researchers believed that quantum computation was science-fiction. Recent results give hope that one day, quantum computers will be part of our lives. There are more and more laboratory experiments that try to bring the quantum computer to life. We will present two of the more promising models.

4.1 Ion trap

In 1995, Cirac and Zoller [15] proposed a scheme to implement a quantum computer using a linear array of trapped ions (the quantum register) and laser pulses (to apply operations). Each ion, maintained in place by an electromagnetic field, encodes a qubit. Its ground and excited states are used as the basis $\{|0\rangle, |1\rangle\}$. Laser pulses are applied to qubits to change their states, allowing the register to be placed in arbitrary superpositions.

4.2 NMR

The apparatus used in a NMR quantum computer is a liquid containing a large number of molecules of some type. The spin state of the nuclei of each atom in the molecule encodes a qubit. Each molecule is an independent quantum computer. We operate on qubits using nuclear magnetic resonance (NMR) techniques. This scheme was proposed by Chuang *et al* [16] in 1996 and is very promising considering the stability of the states. They recently succeed to implement Grover's algorithm on a four-state quantum system.

Acknowlegment

I thank Gilles Brassard for his encouragement and support; Sébastien Paquet, Frédéric Légaré and Julien Marcil for their useful comments on this paper.

References

[1] Peres, A., "Quantum Theory: Concepts and Methods", Kluwer Academic Publishers, Dordrecht (1993).

[2] Einstein, A., Podolsky, B. and Rosen, N., "Can Quantum-mechanical description of physical reality be considered complete?", *Physical Review*, Vol. 47 (1935), pp. 777–780.

[3] Shor, P.W., "Algorithms for quantum computation: discrete logarithms and factoring", *Proceedings 35th Annual Symposium on Foundations of Computer Science*, (1994), pp. 124–134.

[4] Vazirani, U., Quotation from a newspaper article by Tom Siegfried, Science Editor of the Dallas Morning News, (1994).

[5] Grover, L., "A fast quantum mechanical algorithm for database search", *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, (1996), pp. 212–219.

[6] Bennett, C., Brassard, G., Crépeau, C., Jozsa, R., Peres, A. and Wooters, W., "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels", *Physical Review Letters*, Vol. 70 (1993), pp.1895–1899.

[7] Bouwmeester, D., Pan, J.W., Mattle, K., Eibl, M., Weinfurter H. and Zeilinger, A., "Experimental Quantum Teleportation", *Nature*, Vol. 390 (December 1997), p. 575.

[8] Brassard, G., Braunstein, S., Cleve, R., "Teleportation as quantum computation", *Physica D*, (1998), in press.

[9] 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.

[10] Wiesner, S., "Conjugate coding", *Sigact News*, Vol. 15, no. 1 (1983), pp. 78–88.

[11] Marand, C. and Townsend, P.H., "Quantum key distribution over distances as long as 30km", *Optics Letters*, Vol. 20 (15 August 1995).

[12] Shor, P.W., "Scheme for reducing decoherence in quantum computer memory", *Physical Review A*, Vol. 52 (October 1995), pp.2493–2496.

[13] Steane A., "Multiple particle interference and quantum error correction", *Proceedings of the Royal Society*, Series A 452, 2551 (1996)

[14] Preskill, J., "Reliable quantum computers", quant-ph/9705031.

[15] Cirac, J. and Zoller, P., "Quantum computation with cold trapped ions", *Physical Review Letters*, Vol. 74 (1995), pp.4091–4094.

[16] Gershenfeld, N.A., Chuang, I. and Lloyd, S., "Bulk quantum computation", *Proceedings of the 4th Workshop on Physics an Computation*, PhysComp96 (1996), p.134.