

My computer wants to be quantum when it grows up

Javier Rodríguez-Laguna¹

¹*Dto. Física Fundamental, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain.*

(Dated: March 31, 2016)

This text constitutes a crash course on the classical simulation of quantum computers, assuming no previous knowledge of quantum mechanics. These notes are written as supplementary material for a graduate school seminar at the Computer Science School of Universidad Complutense de Madrid (UCM), Spain.

I. INTRODUCTION

The ideal way to teach quantum computation would be to build a quantum computer. Barring that, we will choose the second best way: we will *simulate* the behavior of a quantum computer on a classical one. In fact, the proposed simulations will require little computational power, and can run even in a small laptop. Our intention is simply to help understand the principles involved.

We have written some sample code in C++, which you can find at <http://github.com/jvrlag/qtoys>. The reader is encouraged to download the code, understand it and modify it. In the final section there are instructions in order to compile and run the sample code.

II. THE DIMENSIONALITY CURSE

Quantum computation [1] was born out of a difficulty. Quantum mechanics, the physical theory required to study very small *or* very cold systems, is really hard. Much harder than classical mechanics. Why? Because it takes place in a *much larger space*.

Let us give an example. Imagine a single particle moving on a line segment. For simplicity, let us *discretize space* and assert that it can be in one out of N cells. How much information do you need in order to describe its state? Just one number: $x \in \{1 \dots N\}$. But now let us go quantum. The most salient difference with classical mechanics is that the particle *can be at several sites at once*. The state must be described giving N complex numbers, $\psi(x)$, for $x \in \{1 \dots N\}$, such that the probability of finding it at each site is given by its squared modulus, $|\psi(x)|^2$. If complex numbers scare you, do not worry too much, think of reals.

That is bad enough, but it gets worse. If we move to several particles, the classical description is given by their positions: $x_1 \dots x_n$. So, n numbers. But in quantum mechanics we must provide a complex number for *each possible configuration*: $\psi(x_1 \dots x_n)$, such that the probability of finding the particles at those positions is $|\psi(x_1 \dots x_n)|^2$. Thus, we need to know N^n numbers!! The number of components needed to determine the state grows *exponentially* with the number of particles! This is what we call the *dimensionality curse*.

But the crisis of one guy is the opportunity of another.

Ask Goldman Sachs. In 1982, one of the physics gurus, Richard Feynman proposed to turn the problem around [2]. Classical simulation of quantum systems of many particles takes a huge amount of resources. Yet, mother Nature seems to do it fast. Therefore, Nature must have really fast computers, at the Nature Computer Center, somewhere in the Bay Area, near the Google headquarters. Can we *harness* them?

III. SEDUCING NATURE INTO DOING OUR WORK

A. Optimization

Many scientific and engineering problems reduce to optimization problems. Optimal allocation of resources for society welfare, optimal distribution of elements in a circuit, optimal shipping route for commodities... An optimization problem asks for the minimization (or maximization) of a certain *target function* of many variables. Typical optimization problems include:

- The traveling salesman problem (TSP), in which the optimal route must be found in order to visit a set of cities.
- The knapsack problem, in which one is given a set of items, with a given weight and value, and is asked to select the optimal subset of them, maximizing the total value and with a certain bound on the total weight.

In fact, many seemingly unrelated problems can be mapped into an optimization problem. For example, the problem of sorting the numbers $\{x_1 \dots x_n\}$ can be rewritten as a minimization problem, with a target function over the space of permutations P :

$$S(P) = \sum_{i=1}^{n-1} \theta(P(x_i) - P(x_{i+1})), \quad (1)$$

where $\theta(x)$ is the Heaviside function, $\theta(x) = 1$ if $x > 0$ and 0 otherwise. Another example is the factorization problem, i.e. given an integer N , finding integers n_1 and n_2 such that $N = n_1 n_2$, can be solved by minimizing the target function

$$E(x, y) = |N - xy|, \quad (2)$$

if we restrict it to $x, y \in \mathbb{N}$ and $x, y > 1$. To have an efficient machine to solve generic optimization problems would be a real technological revolution.

B. Algorithmic Complexity

Are all optimization problems equally hard? By no means! The most important criterion to determine the difficulty of a problem is the time required to solve it, depending on the size of the input. If a problem can be solved in polynomial time, then it is said to belong to **P**. If a candidate solution can be evaluated in polynomial time, then it is said to belong to **NP**. As you probably know, one of the problems of the millenium is to find out whether **P=NP** or not.

I would like to give a cautionary remark: rigorously, classes **P** and **NP** are not defined for optimization problems, but for *decision problems*, which are solved by stating Yes or No. For example: is 29 a prime number? But optimization problems can be mapped to decision problems easily, so we will be a bit loose in the terminology.

For example, the sorting problem is **P**. As you know, it can be solved in time $O(N \log(N))$, so less than N^2 . What about the factorization problem? What is the *size of the input*? Typically, the number of bits in the binary expansion of the input integer. With this definition of the input size, nobody knows whether there is or not an algorithm to solve the factorization problem in polynomial time, but we do not expect it to exist. On the other hand, the problem is *easy* (i.e. polynomial) to check: if I give you one candidate factor, you can find whether it is or not good really fast. So, the problem is for sure in **NP**.

There are some special problems which are known as **NP**-complete. These are **NP** problems with a really special feature: if you *ever* find a polynomial algorithm for one of them, then all the **NP** problems will have a polynomial algorithm. The generality and power of this result is simply amazing, and is known as Cook's theorem.

Which optimization problems are **NP**-complete? The traveling salesman problem (TSP), in which we are given a set of cities in a map and we are asked to find the shortest route passing through all of them. Or the knapsack problem, in which we are given a set of objects characterized by a weight and a value, and we are asked to find the set of them with maximum value and below a certain maximal weight.

C. Nature optimizes

Also pure scientists are interested in optimization. Why? Because Nature optimizes all the time. Soap films

minimize area, which means that they minimize the energy. Do you want to *predict* which shape will a soap film acquire, when constrained between this wire and this wall? OK, find in your computer the surface of minimal area with your given constraints, et voilà! Do you want to know what kind of lattice will be formed by atoms of this and that kind? Then a good approximation can be found in this way. First, determine the interaction energy between atoms, as a function of their relative position, $V(\vec{u})$. This function typically has a minimum at a certain equilibrium distance, d_c , becomes very large for small d and tends to zero for very large d . Now, minimize the following energy function:

$$E(\vec{r}_1 \cdots \vec{r}_n) = \sum_{i < j} V(\vec{r}_i - \vec{r}_j). \quad (3)$$

Then you can get very different kinds of crystal structures, depending on the shape of V ! If you combine several different functions V , then there is an amazing richness of minimal energy configurations.

Can we use this fact, that Nature likes to minimize energy, in order to solve computational problems? Sure! Those constitute *analogue computers*, and are a fascinating topic. For example,

Why does Nature minimize energy? I am surprised at how seldom this question shows up. We all assume a principle of *economy of Nature* to hold, which nobody has ever stated in detail. In fact, an old good principle of physics is the *conservation of energy*, which looks completely opposed to the idea of *energy minimization*. Well, let us not forget another old good principle of physics: *entropy grows*, also known as the second principle of thermodynamics. A system in contact with an environment will tend to dissipate energy in the form of heat. An equilibrium configuration is found when all “fast dissipation” processes have already finished, but the “slow dissipation” ones have not yet started. In the soap film example, an initial configuration with a lot of “extra area” will dissipate the extra energy into the environment very fast, due to the contact to the air and the walls. Further energy can be dissipated by “breaking” the film, but that process will take much longer (if we are careful).

Moreover, Nature can make mistakes! Typical crystals are *not* perfect. They have defects: atoms which are misplaced, vacancies... In fact, those defects are rather important in practice: a perfect metallic crystal is fragile. Steel is hardened by filling it up with defects. The lesson that I want to insist upon is: if someone proposes to you a computational method based on the fact that Nature minimizes the energy, do not forget to ask *how* this minimization proceeds. You will avoid misunderstandings.

D. Landscapes

Let us take a closer look at the defect problem. Typical real-life optimization problems are high-dimensional, but

let us go down to one dimension. Imagine a real smooth function $f : \mathbb{R} \mapsto \mathbb{R}$, which you are asked to minimize. You pick up a random point and follow the slope downwards until you reach the closest minimum. Is that the *absolute* minimum of the function? How can we know, unless we explore further? A usual term for that situation is to say that we have a *complex landscape*.

The algorithm described, just to follow the slope downwards, is called a *greedy* algorithm, and is known to fail very often if the problem has many *false minima*. We physicists call them *metastable states*, because they can take really long to *dissipate* their extra energy to the environment and fall into the absolute minimum. For example, a crystal with a single defect in the middle will take a really long time to fall into the perfect crystal. And correspondingly, any greedy numerical minimization algorithm of the energy function will tell us “dude, it’s ok, we gotta minimum!”, and stop working.

But now the question comes... is Nature’s algorithm for optimization greedy? It depends. Sometimes it is, sometimes it is not. As I told you before: never take it for granted!

IV. BRAVE QUANTUM WORLD

In this section we will give a fast introduction to quantum mechanics, but always using concrete code to reinforce the learning.

A. Schrödinger’s cats (quantum states)

We all have heard about Schrödinger’s cat. At least, we heard it from Sheldon Cooper in *The Big Bang Theory*. A cat in a box with a vial of poison which can be broken by a single particle being emitted from a radioactive sample. If the poison spreads, the cat dies. Until we open the box, the cat is in a superposition of dead-cat and alive-cat.

But this explanation is wrong! Worse than wrong: it is misleading. A crucial question to ask: why is *quantum* important? If we simply toss a coin such that a machine will kill the cat if it comes heads, is there any difference? The answer is *yes*. The state “50% probabilities to be alive, and 50% to be dead” is very different from the state “alive + dead”. Let us explain why. And allow me to stop mistreating cats by now. It is really simpler if I use atoms.

Atoms usually have a *spin*, related to angular momentum, which makes them behave like little magnets. The simplest case is when they have *spin-1/2*. In this case, the atom can be in two states, which we will call \uparrow and \downarrow . In quantum people jargon, they are usually written with these funny symbols called *kets*:

$$|\uparrow\rangle, \quad |\downarrow\rangle. \quad (4)$$

But spin is a vector quantity. This means that we can measure its component *along any axis*. For example, let us choose the positive Z -axis. When we measure, only two outcomes are possible: $+1$ and -1 (in certain units). As expected, if we measure the Z -component of the magnetic moment on state $|\uparrow\rangle$, the result is always $+1$, and if it is $|\downarrow\rangle$, then we get -1 for sure.

But here comes the novelty of quantum mechanics. We can have states of the form

$$\alpha|\uparrow\rangle + \beta|\downarrow\rangle, \quad (5)$$

which are interpreted as follows. If we measure the Z -component of the spin, we get $+1$ with probability $|\alpha|^2$, and -1 with probability $|\beta|^2$. But, you may say, as long as this goes, forgetting about the funny squares, this is just the description of a classical probabilistic system.

What is the state in which up and down are equally likely? It is not hard to realize that it is

$$\frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle). \quad (6)$$

And here comes the magic. That state is *pointing to the right*, and we will call it $|\rightarrow\rangle$:

$$|\rightarrow\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle). \quad (7)$$

That means that, if we measure now the X -component of the spin, we obtain $+1$ with certainty. But if you ask about its component along the Z -direction, it does not know what to say... and it gives a random answer! What about the state that is pointing to the left? Not much imagination is needed:

$$|\leftarrow\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle - |\downarrow\rangle). \quad (8)$$

So that is the difference between the “dead+alive” state and the “50% dead and 50% alive” state. The first is called a *pure state*, the second is a *mixed* or probabilistic state. If the state is mixed, it will be probabilistic no matter which question we ask. If the state is pure, it may seem probabilistic when you ask some questions. But *there are always some other question* which will give a definite answer, with certainty. Please, notice how *weird* it is!

We can already give the first important definition, that of **qubit**. Choose two *basis states*, either $|\uparrow\rangle$ and $|\downarrow\rangle$, or $|0\rangle$ and $|1\rangle$. Their name is immaterial. And then, build any linear combination of them, with arbitrary (complex) coefficients: $\alpha|\uparrow\rangle + \beta|\downarrow\rangle$. That is a qubit state. The important thing is that *there is always* a question which the qubit knows how to answer with certainty. In the spin interpretation, there is always a direction in space, \vec{n} , such that if you measure the spin-component along that

direction, you obtain either $+1$ or -1 with certainty. In all other cases, you get random outcomes, with probabilities related to α and β . **A qubit only knows how to answer one question, but the question is, in principle, unknown.** If asked about what it does not know, its answer is random... but consistent. If you repeat the question very fast, the qubit is clever enough to know that it must give to you the same answer. Weird, eh?

So the mixed (probabilistic, classical) state and the pure state are different because they give different answers when asked some questions. But the difference is deeper than that. A mixed/probabilistic state has an *epistemological nature*. With this expression we mean that the state can be really thought to be in one of the classical states, only we do not know which one. In the quantum pure state, the system is in *both* states really at once. This will be one of the keys of quantum computation: we will try the system to do many computations at once, one for each term in the linear combination composing the pure state.

In general terms, the jump from classical to quantum is given by the *superposition principle*. If a classical system can be described by selecting a state from a certain configuration space, $\mathcal{S} = \{s_1 \cdots s_n\}$, then the quantum state is described by any *linear combination of them*. In proper terms: we *upgrade* each state, by writing them as kets, $\{|s_1\rangle \cdots |s_n\rangle\}$, and then allow for states of the form $(1/2)|s_1\rangle - (3/4)|s_2\rangle$. Or, more generally:

$$|\psi\rangle = \sum_{i=1}^n \alpha_i |s_i\rangle, \quad (9)$$

where the α_i are usually called *amplitudes*. We will assume them to be *normalized*:

$$\sum_{i=1}^n |\alpha_i|^2 = 1, \quad (10)$$

Why? Because we would like to interpret $|\alpha_i|^2$ as the *probability* to find the state s_i when we measure. Quantum measurements are random, but the outcome probabilities are controlled. Is that all we need to know to be able to predict the outcomes of measurements? Unfortunately, no.

B. Mistaking states (scalar products)

What happens if I have a quantum state $|\uparrow\rangle$ and I want to measure its component along the X -direction? Remember that, for the state $|\rightarrow\rangle$, given by Eq. (7), measurements of the X -component are certain. We need a way to *compare* states, to find out how similar they are. The idea is that, if they are very similar, the *probability of mistaking* one for the other will be high.

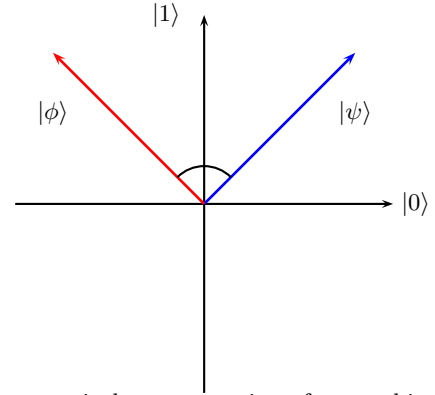


FIG. 1. A geometrical representation of two qubits, $|\psi\rangle$ and $|\phi\rangle$. The squared cosine of the angle between them gives the probability of mistaking them.

The recipe is better understood in geometric terms. Each qubit state can be regarded as a vector on the plane (see Fig. 1). So, if $|\psi\rangle = \psi_0 |0\rangle + \psi_1 |1\rangle$, the vector is $\vec{\psi} = (\psi_0, \psi_1)$. Since the state is normalized, $|\psi_0|^2 + |\psi_1|^2 = 1$: vectors lie on the unit circle. Now, let us introduce a second state, $|\phi\rangle = \phi_0 |0\rangle + \phi_1 |1\rangle$, so $\vec{\phi} = (\phi_0, \phi_1)$. How would you compare how similar the two points on the circumference are? Easy: measure the angle between them. For two unit vectors $\vec{\psi}$ and $\vec{\phi}$, the angle between them is given by the *scalar product*:

$$\vec{\psi} \cdot \vec{\phi} \equiv \bar{\psi}_0 \phi_0 + \bar{\psi}_1 \phi_1 = \cos(\alpha), \quad (11)$$

(The bars are introduced in case of complex numbers.) We get the cosine of the angle, instead of the angle. Shall we take the arc cosine? No!!! The cosine (squared) is better: it gives a number between zero and one, so it can act as a probability just by itself.

So, quantum mechanics proposes the following rule: **the probability of mistaking one state for another is found by obtaining the scalar product between them, as vectors, and squaring that.** In math,

$$\text{Prob}(\phi, \psi) = |\vec{\phi} \cdot \vec{\psi}|^2 \equiv |\langle \phi | \psi \rangle|^2, \quad (12)$$

where we introduce a new notation for the scalar product: $\langle \phi | \psi \rangle$. The state which is *reversed* is called a *bra*, because when you put it together with the *ket*, you get a *bra-ket*. Yes, pun intended. Yes, physicists laugh a lot at their own jokes. Yes, nobody else does.

A space in which elements can be added and multiplied by numbers is called a *vector space*. If you also have a scalar product, we call it a *Hilbert space*, all praise be given to David Hilbert, father of the famous list of mathematical problems to be solved in the XX century, and who thought that his abstract geometrical ideas would never find application in real life. Mwahaha.

So, returning to our original problem, if the real state is $|\uparrow\rangle$ and we measure the X -component of spin, then the probability of obtaining $+1$ coincides with the probability of mistaking state $|\uparrow\rangle$ and the special state which points

to the right, given by Eq. (7). So, let us find their scalar product.

$$\langle \uparrow | \rightarrow \rangle = (1, 0) \cdot \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right) = \frac{1}{\sqrt{2}}, \quad (13)$$

and that is the cosine of the angle between them. But we want it squared, so we get a probability:

$$|\langle \uparrow | \rightarrow \rangle|^2 = 1/2. \quad (14)$$

Evidently, the situation is symmetrical: if you try to measure the X -component on a spin pointing in the Z -direction, the result is $+1$ with probability $1/2$, and -1 with probability $1/2$ too.

C. Many-cat theory (tensor basis)

One cat is fun. Many cats is exponentially more fun.

Let us put two cats inside a box, one white and one black. But they are killer cats of the jungle, which will not stop fighting until only one of them is alive. Then, in classical probabilistic terms, we can describe the final state as “50% probabilities that white cat is alive and black cat is dead & 50% probabilities that black cat is alive and white cat is dead.” Let us give a more concise notation: $|01\rangle$ will mean “white cat is dead, black cat is alive”. So we get “50% $|01\rangle$ and 50% $|10\rangle$ ”. That is a mixed state.

But can we have a pure state for two cats? Yes, of course. In a more sophisticated language, which will allow you to communicate with quantum computer scientists, we are asking about a two-qubit state. For example, we can have $|00\rangle$, in which both cats are dead. Or we can have something funnier, like

$$\frac{1}{\sqrt{2}} (|01\rangle + |10\rangle). \quad (15)$$

Of course, instead of $+$ we can have a $-$ sign between them (or any phase, since the coefficients are complex!) But the important thing is that both configurations exist simultaneously, as we discussed above.

The most general pure state for two qubits is, therefore:

$$|\Psi\rangle = \alpha |00\rangle + \beta |01\rangle + \gamma |10\rangle + \delta |11\rangle. \quad (16)$$

If we can remember the order of the basis states, $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, then the state can be written as a regular vector, an array with the four amplitudes:

$$\Psi = (\alpha, \beta, \gamma, \delta). \quad (17)$$

For example, the vector corresponding to the Eq. (15) is

$$\Psi = \left(0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0 \right). \quad (18)$$

This is the convenient way to store quantum states in the computer.

Now, let us consider the situation in which you have two spins pointing to the right. How do we write that state? Well, we can say that the state is $|\rightarrow\rightarrow\rangle$. But we do not know how to “compare” this state with states written in the $\uparrow - \downarrow$ basis. So, let us rewrite that expression into our usual basis! The trick is to understand the “spin composition” (putting two spins together) as a *product*, which we call the *tensor product* and symbolize with a sign \otimes :

$$\begin{aligned} |\rightarrow\rightarrow\rangle &= |\rightarrow\rangle \otimes |\rightarrow\rangle \\ &= \frac{1}{\sqrt{2}} (|\uparrow\rangle + |\downarrow\rangle) \otimes \frac{1}{\sqrt{2}} (|\uparrow\rangle + |\downarrow\rangle), \end{aligned} \quad (19)$$

and now we apply the distributive property:

$$|\rightarrow\rightarrow\rangle = \frac{1}{2} (|\uparrow\uparrow\rangle + |\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle + |\downarrow\downarrow\rangle). \quad (20)$$

As you may have noticed, the state $|\rightarrow\rightarrow\rangle$ is a linear combination, with equal probabilities, of *all* possible configurations of two qubits! It is a rather democratic state.

What if we have more than two qubits? No real difference. For three qubits, the most general state is:

$$\begin{aligned} |\Psi\rangle &= \alpha |000\rangle + \beta |001\rangle + \gamma |010\rangle + \delta |011\rangle \\ &\quad + \epsilon |100\rangle + \zeta |101\rangle + \eta |110\rangle + \theta |111\rangle. \end{aligned} \quad (21)$$

So, you need 2^3 numbers in order to specify it. For a general case of N qubits, you need 2^N . Some of them have special names:

$$|GHZ\rangle = \frac{1}{\sqrt{2}} (|000\dots 0\rangle + |111\dots 1\rangle), \quad (22)$$

is called the GHZ state (for Greenberger, Horne and Zeilinger). In it, either all qubits are 0 or all 1, we only do not know which. Once one of them is measured, we are sure of the answer. Also we like the *Néel* state,

$$|N\rangle = \frac{1}{\sqrt{2}} (|0101\dots\rangle + |1010\dots\rangle), \quad (23)$$

Other interesting states are the Dicke states $|D_{N,m}\rangle$, in which we have m 1's among N qubits, but we do not know where. For example,

$$|D_{4,2}\rangle = \frac{1}{\sqrt{6}}(|0011\rangle + |0101\rangle + |0110\rangle + |1001\rangle + |1010\rangle + |1100\rangle). \quad (24)$$

Thus, a many-qubit quantum (pure) state is a linear combination of certain basis states, which are called the *tensor basis*. They are 2^N states, which contemplate all possible combinations of 0's and 1's. You can see already how the *dimensionality curse* has made its appearance...

What physical role can many-qubit states play? If we read them as a combination of $|\uparrow\rangle$ and $|\downarrow\rangle$, then they can be regarded as *spin chains*. Atoms in a metal sometimes tend to align their spins to point in the same direction, and this means that the system will become *magnetized*. So, this is a key to understand the magnetic properties of materials. On the other hand, a $|0\rangle$ and $|1\rangle$ reading can be interpreted as *absence* or *presence* of a particle (e.g. an atom, or an electron) in a certain place. Then, combinations like $|100001\rangle$ means that there are two particles, one at the first site and another one at the last site. Thus, we can study systems in which many particles interact, such as electrons in a metal, collisions of particles at CERN...

D. Qubism

So, a many-qubit state can be seen as a vector on a 2^N -dimensional space. Of course, that is nearly impossible to visualize. Since we physicists love visualizing things, in our group we designed a visualization strategy which has some very nice properties, which we called **qubism** [3]. Yes, we think that we are hilarious, and our grandmothers agree.

Let us refer to Fig. 2. We draw a unit square and select a basis state for N qubits, such as 011001, with $N = 6$. Separate the bit string in pairs, and select the first one: 01. Now, divide the square in four smaller squares, and attach to each of them a pair of bits: 00 for the upper-left, 01 for the upper-right, 10 for the lower-left and 11 for the lower-right. So we select, in our case, the upper-right square. Now, divide this square again in four smaller squares, and repeat the selection using the next pair of bits, 10.

Once we know how to map each basis state into a tiny square, the next step is straightforward: paint each tiny square with a color related to its amplitude, as it is done in Fig. 3 and Fig. 4 for a few examples: the $|GHZ\rangle$ state with $N = 4$ qubits, the Dicke states $|D_{4,2}\rangle$ and $|D_{12,6}\rangle$ and a factorizable qubit state of the form

$$(\alpha|0\rangle + \beta|1\rangle)^{\otimes 12}, \quad (25)$$

which means that all $N = 12$ qubits are pointing in the same direction.

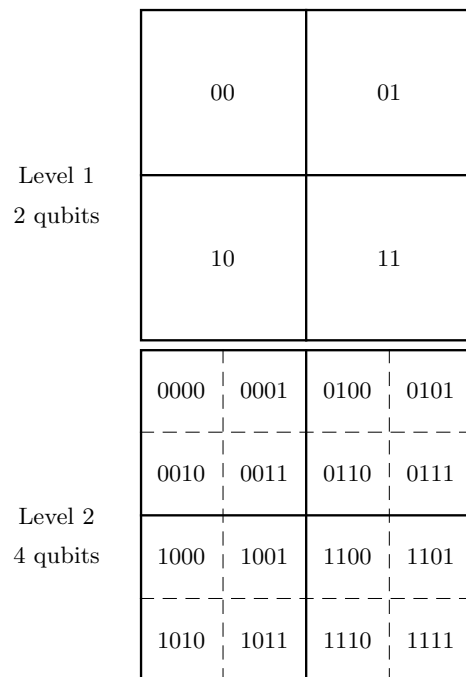


FIG. 2. Qubism visualization.

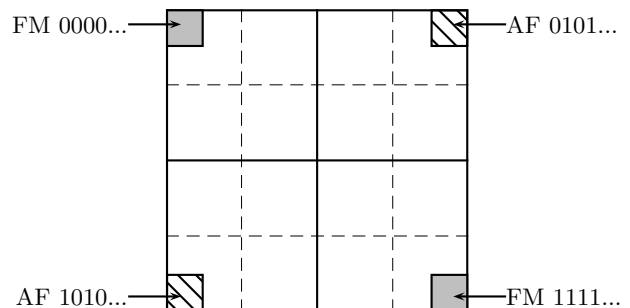


FIG. 3. Special points in a qubistic plot: the ferromagnetic and anti-ferromagnetic points.

Why bothering with those pictures? First of all, because they are cute. But, beyond that, because they are informative. At a glimpse, we can read physical properties of the system. Please, see [3] for further details.

E. Energy

What is the *energy* of a quantum state? If you know how to compute the energy of a (classical or quantum) system, it means that you know everything relevant about its interactions. For spins, the most relevant interactions are of two types:

- An external magnetic field will force the spin to prefer to be pointing its direction. This interaction affects the spins one by one, so it is said to be *single-*

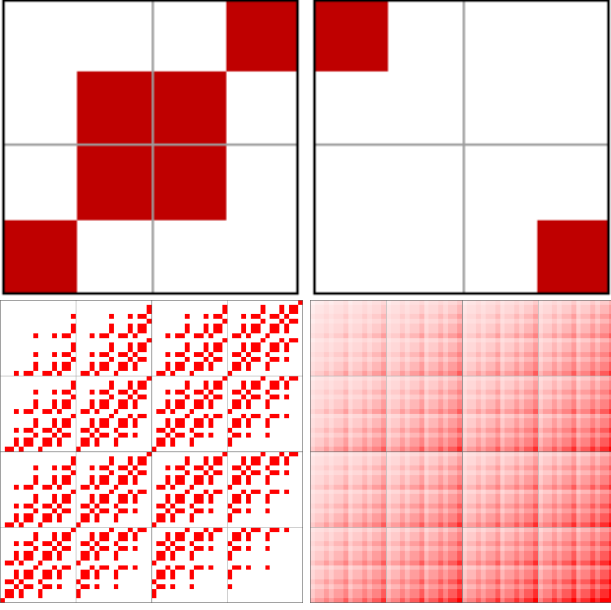


FIG. 4. Examples of qubistic plots. (A) The Dicke state, $|D_{4,2}\rangle$ of $N = 4$ qubits, which contains all states with 2 0 and 2 1. (B) The GHZ state with $N = 4$ qubits. (C) The Dicke state $|D_{12,6}\rangle$. (D) A factorizable state $(\alpha|0\rangle + \beta|1\rangle)^{\otimes 12}$.

body.

- Neighboring spins may want to be parallel or anti-parallel. This interaction affects the spins in pairs, so it is said to be *two-body* or *pair interaction*.

This is not a physics course, so I will not give details about their physical origin, I will only show you how to use them. So, let us start with an external magnetic field pointing in the Z -axis. The expression for the energy will be

$$E = \Gamma S^z, \quad (26)$$

where Γ is a constant, proportional to the magnetic field. So the energy of the state $|\uparrow\rangle$ is $+\Gamma$, and of the state $|\downarrow\rangle$ is $-\Gamma$. What about the state $|\psi\rangle = \alpha|\uparrow\rangle + \beta|\downarrow\rangle$? We can mistake it for $|\uparrow\rangle$, with probability $|\alpha|^2$, and obtain $+\Gamma$; or we can mistake it for $|\downarrow\rangle$, with probability $|\beta|^2$, and obtain $-\Gamma$. What will be the *expected value* of the energy, if we perform many times the experiment? For that we use the expression

$$\langle E \rangle = p(\uparrow)E(\uparrow) + p(\downarrow)E(\downarrow) = \Gamma (|\alpha|^2 - |\beta|^2). \quad (27)$$

Now let us bring more spins! Each one contributes an energy to the total equal to the value of its Z -component multiplied by Γ :

$$E = \Gamma \sum_{i=1}^N S_i^z, \quad (28)$$

where S_i^z is the value of the Z -component of the i -th spin. So, let us consider the state $|\uparrow\uparrow\uparrow\downarrow\rangle$. What is its energy? It is not hard: three spins \uparrow contribute 3Γ , one \downarrow contributes $-\Gamma$, so we get $E = 2\Gamma$ with certainty.

What happens if we have two spins pointing in the right direction? The state is now $|\rightarrow\rightarrow\rangle$. This state can be rewritten in the $\{\uparrow, \downarrow\}$ basis as in Eq. (20). How much is the energy in this case? The result is not certain, but *probabilistic* again. If you mistake your state with $|\uparrow\uparrow\rangle$, the measurement will give 2Γ . But that will occur with probability $1/4$. Let us check all the cases:

$$|\uparrow\uparrow\rangle \quad p = 1/4, \quad E = 2\Gamma, \quad (29)$$

$$|\uparrow\downarrow\rangle \quad p = 1/4, \quad E = 0, \quad (30)$$

$$|\downarrow\uparrow\rangle \quad p = 1/4, \quad E = 0, \quad (31)$$

$$|\downarrow\downarrow\rangle \quad p = 1/4, \quad E = -2\Gamma. \quad (32)$$

So, what is the *expected value* of the result? You obtain 2Γ one fourth of the times, -2Γ one fourth of the time and zero one half. So, *in average*, the result will be zero. That is intuitively correct: the spins are pointing in the X -direction, so they do not care for a magnetic field which is orthogonal.

Now let us study the pair interaction term, which is much more fun. Consider the situation in which two spins interact so that their energy is lowest when they are aligned along the Z -axis:

$$E = -JS_1^z S_2^z, \quad (33)$$

this is the so-called Ising term, where J is some coupling constant. The energy of state $|\uparrow\uparrow\rangle$ is $-J$, while the energy of state $|\uparrow\downarrow\rangle$ is $+J$. Of course, the energy of the state $|\rightarrow\rightarrow\rangle$ is zero, as you can readily check.

F. Enter The Matrix (The Hamiltonian)

We physicists are often asked if we are friends of the mathematicians. No, no: we are brothers. You choose your friends. Mathematics is intrinsically attached to physics. The reason is that whenever we have visited our mathematician friends with a problem, the solution always gives much more in return than we asked for. And further problems that we didn't expect.

So, we explained to them our methods to compute the energy of a quantum state, and they explained to us a trick that might help us: associate the energy to a symmetric square matrix. Well, this is in layman terms. What they said is: every observable is represented by a self-adjoint linear endomorphism of the Hilbert space. More or less, the same thing.

For example, for the X -component and the Z -components of spin use

$$S^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad S^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (34)$$

But, what is the big deal? What do we do with these matrices? To start with, expectation values on a given state $|\psi\rangle = \alpha|\uparrow\rangle + \beta|\downarrow\rangle$ are found by computing

$$\langle\psi|S^x|\psi\rangle, \quad \langle\psi|S^z|\psi\rangle \quad (35)$$

Does this work? Let us check. The expected value of S^z on a state $|\psi\rangle = \alpha|\uparrow\rangle + \beta|\downarrow\rangle$ is $|\alpha|^2 - |\beta|^2$, as we showed earlier. With our trick, it is given by:

$$(\bar{\alpha}, \bar{\beta}) \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = (\bar{\alpha}, \bar{\beta}) \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} = |\alpha|^2 - |\beta|^2. \quad (36)$$

So, OK. But we can repeat now the calculation with S^x :

$$(\bar{\alpha}, \bar{\beta}) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = (\bar{\alpha}, \bar{\beta}) \begin{pmatrix} \beta \\ \alpha \end{pmatrix} = \bar{\alpha}\beta + \alpha\bar{\beta}. \quad (37)$$

And the result is obtained pretty fast.

But, as I promised to you, the idea gives further benefits! Assume we have a complex quantum system and somebody has given to us the matrix corresponding to its energy, which we call the *Hamiltonian*, for a certain Irish guy who used to engrave equations with a penknife in the walls of Dublin. The expectation value of the energy is given by

$$\langle E \rangle = \langle\psi|H|\psi\rangle. \quad (38)$$

But there are more to it. There are some states which are specially dear to any observable, those for which it behaves like a number:

$$H|\varphi_1\rangle = E_1|\varphi_1\rangle. \quad (39)$$

Such state is called an *eigenstate* of the Hamiltonian, and the number is called an *eigenvalue*. The expected value of the energy on those states is just the eigenvalue: $\langle\varphi_1|H|\varphi_1\rangle = \langle\varphi_1|E_1|\varphi_1\rangle = E_1$, because the state is normalized. But there is more to it: **if the state is an eigenstate, the outcome of a measurement is certain**. If the state is $|\varphi_1\rangle$, a measurement of the energy will always yield E_1 .

Let us give a couple of examples. If $H = S^z$, we have as eigenvectors $|\uparrow\rangle$ and $|\downarrow\rangle$, with eigenvalues $+1$ and -1 . Check it! If we have $H = S^x$, then we have as eigenvectors $|\rightarrow\rangle$ and $|\leftarrow\rangle$, with eigenvalues $+1$ and -1 again. In math:

$$S^x|\rightarrow\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} +1 \\ +1 \end{pmatrix} = \begin{pmatrix} +1 \\ +1 \end{pmatrix} = (+1) \cdot |\rightarrow\rangle \quad (40)$$

$$S^x|\leftarrow\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} +1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ +1 \end{pmatrix} = (-1) \cdot |\leftarrow\rangle \quad (41)$$

$$(42)$$

Are eigenstates rare? Or are there many of them? Many! That is the best part: as many as the dimension of the Hilbert space! Mathematicians tell us even more: eigenstates corresponding to different eigenvalues will be orthogonal (their scalar product is zero). This means that you can not *mistake* one for another.

So, any observable, and the Hamiltonian (=energy) too, can be *diagonalized*, i.e.: we can find N eigenstates $|\varphi_i\rangle$ and N eigenvalues, E_i . A measurement on a given state $|\psi\rangle$ will yield, randomly, one of the eigenvalues, E_i , with probability given by the *probability of mistaking your state with the corresponding eigenstate*:

$$P(E_i) = |\langle\varphi_i|\psi\rangle|^2. \quad (43)$$

The eigenvalues are assumed to be ordered $E_1 < E_2 < \dots < E_N$ (of course, some of them might coincide, but we will not go into that now). Therefore, the minimal energy which can be obtained is always E_1 , if the state is $|\varphi_1\rangle$, which is called the **ground state** of the system.

Given a (symmetric) matrix, obtaining the eigenstates and eigenvalues (typically called diagonalization) is a standard problem in linear algebra. Of course, I will not go into the details [4]. Suffice it to say that it can be efficiently solved, in time $O(N^3)$, and that the algorithms are available in free software, such as LAPACK.

A good teacher knows when to stop, but I never claimed to be one! So, let me finish with the most magical property of the Hamiltonian. It *rules* the dynamics!! Seriously, in order to know how your state will evolve in time, the only thing you need to know is the Hamiltonian. The dynamics is given by Schrödinger's equation:

$$i\frac{d|\psi(t)\rangle}{dt} = H|\psi\rangle, \quad (44)$$

so the evolution is obtained by solving this differential equation. Yes, it contains complex numbers. No, you do not need to understand it to follow the rest of this tutorial. Yes, I am evil and I only wrote the equation to scare you out.

G. Writing Hamiltonians

Most of what a quantum physicist does is divided in two sections: (1) writing the Hamiltonian for a certain system; (2) finding the eigenstates and eigenvalues of a given Hamiltonian. The rest of the time is devoted to writing grant proposals if you have tenure, or finding a job if you are not. So, let us concentrate now on the idea of writing a Hamiltonian.

Hamiltonian-writing is a venerable art with only one rule: find how the operator acts on all the states of a basis, write the result as columns and you are done. How does this apply in practice? The only way to do it is via

an example. Consider the following Hamiltonian, acting on two qubits

$$H = S_1^z + S_2^x. \quad (45)$$

It is immaterial to use $\{\downarrow, \uparrow\}$ or $\{0, 1\}$ as our basic symbols, so we choose $\{0, 1\}$, and we write the lexicographic basis, $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. The action of S^z is simple: multiply by +1 if the corresponding qubit is 1 or by -1 if it is 0. The action of S^x is also simple: change the value of the qubit: $0 \leftrightarrow 1$.

Then, we write the action of the Hamiltonian on all basis states:

$$\begin{aligned} (S_1^z + S_2^x) |00\rangle &= -|00\rangle + |01\rangle = (-1, 1, 0, 0), \\ (S_1^z + S_2^x) |01\rangle &= -|01\rangle + |00\rangle = (1, -1, 0, 0), \\ (S_1^z + S_2^x) |10\rangle &= +|10\rangle + |11\rangle = (0, 0, 1, 1), \\ (S_1^z + S_2^x) |11\rangle &= +|11\rangle + |10\rangle = (0, 0, 1, 1). \end{aligned} \quad (46)$$

Now, we write the Hamiltonian matrix putting the four vectors as columns:

$$H = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad (47)$$

What if, instead of addition of operators we have a product? Like

$$H = S_1^x S_2^x, \quad (48)$$

In that case, we act *consecutively*. First, with S_2^x . Then, with S_1^x . In this case, the order is immaterial, but remember that operators do not commute in general! (It is not the same to study and take an exam, or take an exam and then study). So, let us see how the action goes:

$$\begin{aligned} S_1^x S_2^x |00\rangle &= |11\rangle = (0, 0, 0, 1), \\ S_1^x S_2^x |01\rangle &= |10\rangle = (0, 0, 1, 0), \\ S_1^x S_2^x |10\rangle &= |01\rangle = (0, 1, 0, 0), \\ S_1^x S_2^x |11\rangle &= |00\rangle = (1, 0, 0, 0). \end{aligned} \quad (49)$$

And we can write the full matrix:

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad (50)$$

The logic is the same if we work with a larger number of qubits, but with a larger basis. Remember the dimensionality curse! 2^N states... that is scary! But, if you

want to simulate a quantum computer in your classical computer, there is no choice!

At this stage, we know several important things:

- States of N qubits are vectors with 2^N components.
- The energy is represented by a $2^N \times 2^N$ matrix.
- To minimize the energy, find the eigenstate with lowest eigenvalue.

Now, your training as quantum padawans is finished. You can move on to the last room, where you will be given your jedi sword, in the form of computer code, and learn the not-so-ancient arts of quantum simulation.

V. QUANTUM ENGINEERING

This section is the core of this tutorial. We will learn what adiabatic quantum computation (AQC) is, and how to simulate it in your computer.

A. The spin-glass problem

As we have discussed above, Nature solves problems when it tries to minimize the energy of a system. We can use that to our advantage. We can prepare a quantum system such that its ground state corresponds to the solution of a problem.

For example, we can get a lot of spins-1/2 (aka qubits) and join them with wires. These wires are of two basic types: *ferromagnetic* wires are designed so that the spins want to be parallel. This means that, if they both point up or they both point down, the energy is minimal. *Anti-ferromagnetic* wires behave in the opposite way: anti-parallel spins minimize the energy.

If the wiring graph is simple, the problem of finding the ground state can be pretty simple. For example, if the wiring is *linear*, it is always a trivial problem: give any direction to the first spin, and then set all to either the same if the wire is ferro or the opposite as the previous if the link is anti-ferro.

But if the graph is complex enough, the problem can be tough (see Fig. 5). The reason is that using the previous algorithm can run into contradictions, which we call *frustration*. Through one path, we get to put a certain spin as up, while coming from another path we get to put it as down. Which one shall we choose? Also, wires can have different strengths, of course. And, moreover, we can give a different *absolute* energy to the spin up over the spin down, which amounts to a certain magnetic field in the Z -direction. The problem that we have described is called the *spin-glass* problem which, for generic graphs, is **NP**-complete.

All in all, we can understand why the spin-glass problem is difficult by assimilating it to the most relevant problem for all of us: to sort the priorities in life in

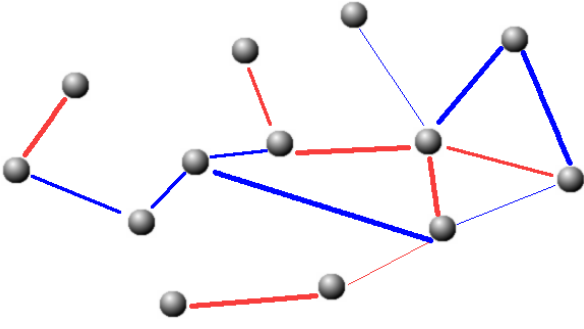


FIG. 5. Example of a spin glass, nodes represent qubits. Blue links are ferromagnetic (parallel) interactions, while red are antiferromagnetic.

order to maximize happiness. What does this mean? We all have certain aims in life, with different intrinsic values: to understand quantum computation, to have a {girl,boy}friend, to get wasted in Oktoberfest, to have children, to have friends, to have a lot of fun... Some of them reinforce each other, such as “going to Oktoberfest” and “having a lot of fun”. But some others are contradictory”, such as “having a lot of fun” and “having children”. The problem, of course, is to select a set of aims on which we should focus in order to maximize our happiness. If we focus on an aim, we say its *spin is up*, and if we decide to let it go, we say the *spin is down*. Then, we can easily convert our problem into an instance of the spin-glass problem.

Now, let us go formal. Consider a system of spins-1/2 (qubits) with the following Hamiltonian, known by the name of *Ising*:

$$H_I = \sum_{i,j} J_{ij} S_i^z S_j^z + \sum_i h_i S_i^z, \quad (51)$$

where the J_{ij} are the coupling constants, which can be either positive or negative. If the coupling is positive, spins minimize their energy by pointing in opposite directions, so it will be an *anti-ferromagnetic* coupling. The h_i are the *local magnetic fields*, and tell us what is the *natural trend* of each spin by itself. If it is positive, it will tend to go down.

Can we write the full matrix for this Hamiltonian? Yes, we can, provided that the system is not too large: remember that the dimension of the matrix grows as 2^N ! Writing the Hamiltonian matrix is not that hard. Remember that the action of the S^z operators is always *diagonal*: their action on each basis state is always proportional to itself.

Let us give a thorough example, from Fig. 6. We have three spins, linked with anti-ferromagnetic (positive) couplings, $J_{12} = +1$, $J_{13} = J_{23} = +2$. Moreover, we only have a local field at spin 1: $h_1 = +1$, all the others are zero. Then, the Hamiltonian action on the basis states is simple:

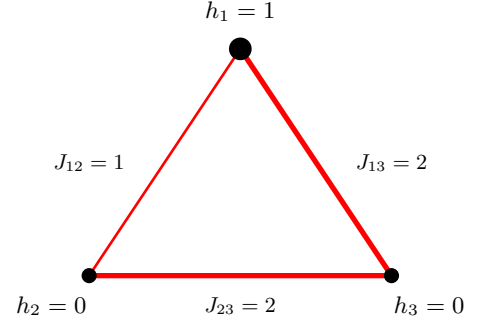


FIG. 6. Example of spin-glass, studied in detail in the text.

$$\begin{aligned} H_I |000\rangle &= (+1 + 2 + 2 - 1) |000\rangle = +4 |000\rangle, \\ H_I |001\rangle &= (+1 - 2 - 2 - 1) |001\rangle = -4 |001\rangle, \\ H_I |010\rangle &= (-1 + 2 - 2 - 1) |010\rangle = -2 |010\rangle, \\ H_I |011\rangle &= (-1 - 2 + 2 - 1) |011\rangle = -2 |011\rangle, \\ H_I |100\rangle &= (-1 - 2 + 2 + 1) |100\rangle = +0 |100\rangle, \\ H_I |101\rangle &= (-1 + 2 - 2 + 1) |101\rangle = +0 |101\rangle, \\ H_I |110\rangle &= (+1 - 2 - 2 + 1) |110\rangle = -2 |110\rangle, \\ H_I |111\rangle &= (+1 + 2 + 2 + 1) |111\rangle = 6 |111\rangle. \end{aligned} \quad (52)$$

So the full matrix is diagonal:

$$H_I = \begin{pmatrix} +4 & & & & & \\ & -4 & & & & \\ & & -2 & & & \\ & & & -2 & & \\ & & & & 0 & \\ & & & & & 0 \\ & & & & & & -2 \\ & & & & & & & +6 \end{pmatrix} \quad (53)$$

The ground state, of course, is the state $|001\rangle$, with energy -4 . So, if we write the Hamiltonian for the spin-glass problem, we already will know the solution, because we have to compute the energy of every single basis state, right? Yes. But remember, we are not trying to devise a *classical* technique to solve the spin-glass problem.

What happens if we go to the lab, prepare our complex spin system in Hamiltonian H_I and ask Nature to find the minimal energy state? The first idea one might have is to put it at finite temperature and then cool it down. Making a system cool down is the same as making it dissipate energy. So, in other terms, we are doing classical (thermal) annealing. The result, in the spin-glass problem, is rather poor: the system tends to get stuck in metastable configurations (fake minima) for a very long time.

In fact, this result is so important that we give it a name: we find *glassy behavior*. Why that name? Take a look at your window. Glass is an overcooled liquid, with a high viscosity. So, the minimal energy configuration for

your window glass would be to creep down to the floor. Eventually, it will do that, but it will take a quite long time. So, classical annealing of spin-glasses is not useful for the same reason that we can use glass in windows: it takes too long to dissipate the energy.

B. Transverse Field

So, if getting the ground state of Hamiltonian H_I ((51)) is so tough, what can we do? Let us study a different Hamiltonian, whose ground state is easy to obtain, to see if we can learn something in the procedure.

$$H_X = - \sum_i S_i^x, \quad (54)$$

In order to minimize the energy, all spins should point in the X -axis. This means that the state will be:

$$|\Psi(0)\rangle = \left(\frac{1}{\sqrt{2}}\right)^N (|0\rangle + |1\rangle) \otimes \cdots \otimes (|0\rangle + |1\rangle). \quad (55)$$

If we carry out the tensor products (remember, they behave like regular products), we realize that $|\Psi(0)\rangle$ contains *all* configurations with the same weight! In other terms, it is the *most democratic state*, which does not distinguish between configurations at all. The physical meaning is nice: S_k^z makes the k -th spin fluctuate: go from 0 to 1 to 0... So, all the spins fluctuate independently and randomly, and the final state is the democratic one.

What is the action of the transverse field Hamiltonian on quantum states? Let us write down the matrix for H_X explicitly in the $N = 3$ case. The action of the S^x operator is to *flip* the spin: 0 goes to 1 and 1 goes to 0. So, we have:

$$\begin{aligned} (S_1^x + S_2^x + S_3^x) |000\rangle &= |100\rangle + |010\rangle + |001\rangle \\ (S_1^x + S_2^x + S_3^x) |001\rangle &= |101\rangle + |011\rangle + |000\rangle \\ (S_1^x + S_2^x + S_3^x) |010\rangle &= |110\rangle + |000\rangle + |011\rangle \\ (S_1^x + S_2^x + S_3^x) |011\rangle &= |111\rangle + |001\rangle + |010\rangle \\ (S_1^x + S_2^x + S_3^x) |100\rangle &= |000\rangle + |110\rangle + |101\rangle \\ (S_1^x + S_2^x + S_3^x) |101\rangle &= |001\rangle + |111\rangle + |100\rangle \\ (S_1^x + S_2^x + S_3^x) |110\rangle &= |010\rangle + |100\rangle + |111\rangle \\ (S_1^x + S_2^x + S_3^x) |111\rangle &= |011\rangle + |101\rangle + |110\rangle. \end{aligned} \quad (56)$$

Or, in matrix form:

$$H_X = - \begin{pmatrix} & 1 & 1 & 1 \\ 1 & & 1 & 1 \\ 1 & 1 & 1 & \\ 1 & & 1 & 1 \\ & 1 & & 1 \\ & & 1 & 1 \\ & & & 1 \end{pmatrix} \quad (57)$$

Oh, come on! It is all zeroes and ones, you computer scientists should *love it*! For sure, you can write a program to get all the 0's and the 1's in the right places!!

The ground state of H_X is really trivial to find. Just get all spins pointing to the right! Unfortunately, we do not really want it for anything, since it does not solve any problem.

C. Adiabatic Quantum Computation

So, we have two different Hamiltonians: H_I , whose ground state we want to obtain but can not; and H_X , whose ground state we can obtain but do not want.

There is a theorem in quantum mechanics that I really like. It is called the *adiabatic theorem*. Let us start with a given Hamiltonian, and with the system in its ground state. If we change the Hamiltonian *infinitely slowly*, then the state of the system will adapt to be always in the ground state of the new Hamiltonian, *as long as* the ground state is always unique.

So, we can think of a possible strategy to solve our spin-glass problem: start with H_X , and get its ground state. Then, modify slowly the Hamiltonian until it becomes H_I , and then read the solution! Imagine that we install a *knob* on our system (we are engineers! we can do that!), and call it s :

$$\begin{aligned} H(s) &= sH_I + (1-s)H_X \\ &= s \left(\sum_{ij} J_{ij} S_i^z S_j^z + \sum_i h_i S_i^z \right) - (1-s) \sum_i S_i^x. \end{aligned} \quad (58)$$

This Hamiltonian, which is an interpolation between H_X and H_I , is known as the *Ising model in a transverse field*, or ITF. Now, all we have to do is:

- Start at $s = 0$, get the ground state.
- Increase s slowly, so that the system will stay in its ground state.
- Reach $s = 1$ and read the solution.

Does this work? Well, the theorem says *infinitely slowly*. In fact, in physics we use the word *adiabatic* for that. But, in practice, this can not be the case. What is, in practice, the minimum time that we can take to do it? The theory behind this is beyond the scope of these notes, but I think that I have a good way to explain it. It is called the *Landau-Zener formula*.

D. Landau-Zener Formula

Do you have small children? If you have, you know what happens when they fall asleep in the living room in front of the TV. You have to take them in your arms and move slowly (adiabatically) to their room. But, if you move too fast, the child might wake up. Of course, there is always a certain critical point, when you have to open the doorknob, when being slow and careful is crucial. The same happens in adiabatic quantum computation.

So, our aim is to keep the ground state (sleeping baby) as we move. The risk is to jump into the first excited state (asleep and crying baby). How slow do we have to go, in order to make sure that we will keep the computation adiabatic? We have an expression for it, given by Landau and Zener.

As s advances, from $s = 0$ to $s = 1$, let $|\Psi_0(s)\rangle$ be the ground state, with energy $E_0(s)$, while $|\Psi_1(s)\rangle$ will be the *first excited*, with energy $E_1(s)$. The difference between these two energies is called the (spectral) *gap*: $\Delta E(s) = E_2(s) - E_1(s)$. The gap is the risk. If the gap is small, the ground state can *spontaneously jump* into the first excited. But also the speed at which we move is important. We must decide the speed at which we are going to move from $s = 0$ to $s = 1$. Of course, we should *adapt* our speed, going faster when the gap is large, and more slowly when it is small. In general terms, the speed should be proportional to the gap squared:

$$v(s) \propto \Delta E(s)^2 \quad (59)$$

Of course, the minimal speed will provide the bottleneck. In order to compare the efficiency of classical algorithms and quantum adiabatic computation, one should check the time taken by both. And the time taken by an AQC can be found by

$$T_{AQC} \sim \int_0^1 ds \frac{1}{v(s)}. \quad (60)$$

More specifically, one should compare how T_{AQC} grows with the input size of our problem, for different types of problems. If it grows polynomially, then our problem is in class **BQP**, which stands for *bound-error quantum polynomial-time*. Is the quantum equivalent of **P**.

Fig. 7 shows a few *gap functions*, $\Delta E(s)$ for the AQC applied to four realizations of spin-glass problem with

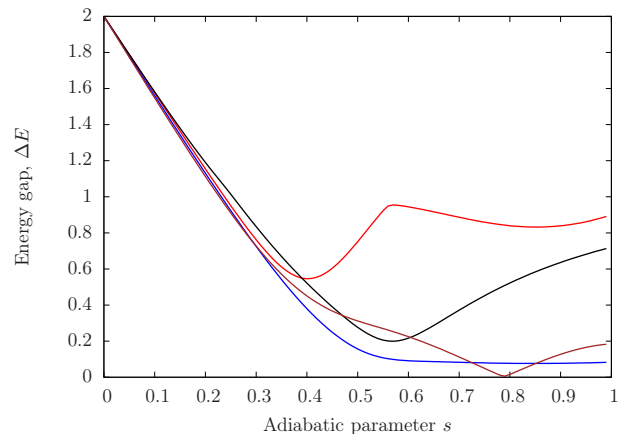


FIG. 7. Spectral gap along the AQC as a function of the adiabatic parameter s , for different realizations of the spin-glass problem.

$N = 8$ qubits. Notice that, usually, they have a minimum at a certain value of $s = s^*$. Thus, in order to characterize how well will AQC perform on a certain class of problems, we should establish how will the gap behave for them. That is an active area of research in quantum computation.

But, one might ask, is adiabatic quantum computation good enough? The answer is *yes*: AQC is *universal*, in the sense that it can reproduce the best results obtained by any other flavor of quantum computation.

Another question may rise. We did AQC interpolating between H_X and our desired Hamiltonian, H_I , through the most straightforward path, just $H(s) = sH_I + (1 - s)H_X$. If we find a bad bottleneck in our way, maybe we can *avoid* it by selecting a different path? The answer now is complicated.

During an AQC, a moment where the gap goes to zero with the system size is called a *quantum phase transition*. Somehow, quantum phase transitions constitute the *censors* which control the speed at which we can do computations in Nature. Although nobody has been able to prove this, we have the strong suspicion that there is always a quantum phase transition separating *easy* and *hard* Hamiltonians, in order to make sure that we do not solve hard problems really fast.

But the characterization of that quantum censorship has not been carried out. It is now your task to find out how much better we can do with quantum algorithms than with classical ones!

VI. NOW, LET US PLAY!

This text is conceived as a tutorial on classical simulation of quantum adiabatic computations, so that you can now write your own code to simulate a quantum adiabatic computation in your (classical, I guess) computer.

The structure of a simple simulation of an AQC is al-

ways the following:

- Select your target function, $T(s_1 \cdots s_N)$, where s_i is a bit.
- Write down your target Hamiltonian, H_I , which has as diagonal entries the evaluations of the target function.
- Write down a *simple* Hamiltonian, usually H_X .
- Compute, for all s , $H(s) = sH_I + (1-s)H_X$.
- Find the ground state and the excited state of $H(s)$.
- Find the bound for the speed at which you can do AQC for that value of s .
- Integrate, and find the total time required for the AQC.

You may wonder how difficult it will be to build the target Hamiltonian, H_I . At this level, we do not worry too much about that. Few-qubit interactions can be easily built, indeed. But our concern is more general: *if* the engineers are able to build H_I , will quantum computation be able to find its ground state, using AQC, in a short time?

A. Technicalities

We have prepared the code in the Github repository

<http://github.com/jvrlag/qtoys>

and, as a tar-ball, in the webpage

<http://mononoke.fisfun.uned.es/qtoys>

It is written in C++, which implements these ideas

in a simple situation. The code is prepared to run on a Linux or Mac box, but it is not hard to adapt to other systems. In order to be able to compile and run easily, you need the following libraries:

- BLAS (basic linear algebra system) and LAPACK (linear algebra package), standard algorithms for matrix manipulation, including diagonalization.
- Xlib, the basic graphics libraries of the X11 system.
- Imlib2, a pretty general image manipulation library.

The Xlib and Imlib2 dependencies can be removed easily, by cancelling the graphics output. The code, nonetheless, will not work without BLAS-LAPACK. How can I be sure that I have got them in my system? It depends on your Linux/Mac flavor. For example, in Debian/Ubuntu, run:

```
sudo apt-get install g++ libblas-dev
libblas3 liblapack-dev liblapack3 libX11-dev
libImlib2-dev
```

The code is, hopefully, self-explanatory. But, since that hope is beyond all realistic dreams, please feel free to contact the author [jvrlag AT gmail.com](mailto:jvrlag@gmail.com).

ACKNOWLEDGMENTS

The author wants to acknowledge the Computer Science School (Facultad de Informática) of UCM, Madrid, for their kind invitation to deliver this talk, particularly to I. Rodríguez-Laguna and N. Martí. This work was funded by the Spanish government through grants FIS-2012-33642 and FIS-2012-38866-C05-1.

-
- [1] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge Univ. Press (2001).
 - [2] R.P. Feynman, Int. J. Mod. Phys. **21**, 467 (1982).
 - [3] J. Rodríguez-Laguna, P. Migdal, M. Ibáñez-Berganza, M.

- Lewenstein, G. Sierra, New J. Phys. **14**, 053028 (2012).
- [4] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C*, Cambridge Univ. Press (1988).