Lecture 1

Course webpage: teaching. Probability course useful: MATH 302 or STAT 302.

Bits and qubits Classical computers use bits that operate according to the rules of classical mechanics, quantum computers use "quantum bit" (qubits) that operate according to the rules of quantum mechanics.

Bits are usually instantiated as current being on or off in a transistor. Qubits can be instantiated in many ways, e.g.,

1. Atoms: ground and excited states of electrons in the atom. (QuEra).
2. (Dual-rail) photons: left and right (rail) spatial locations of the photon (PsiQuantum).
3. Superconducting circuits (artificial atoms): ground and excited states of the current in the circuit (Google, IBM).

The point to stress is:

> **Quantum computers are not "sped-up versions" of classical computers, e.g., by "making the current run faster", but a new type of computer that uses different (more general) laws of physics.**

What makes a good qubit? A good qubit needs to satisfy two requirements in tension with each other:

1. The qubit does not want to interact with the environment.
2. The qubit does want to interact with the experimenter wishing to perform a computation.

These are in tension because in practice it is hard for the experimenter to distinguish itself from the environment. Atoms, photons and other instantiations of qubits in nature score highly on the first point and less highly on the second point. For superconducting circuits the reverse is true.

States of bits and qubits. n bits can be in one of 2^n states. In contrast n qubits can be in one of those states but also states "in-between" that are called superposition states. As a brief preview: the state of n qubits is mathematically written as

$$\sum_{x\in\{0,1\}^n} \alpha_x |x\rangle \,, \tag{1}$$

where the α_xs are complex numbers.

But it is important to stress: by employing random coin flips, bits can also be in "classical superposition states" that can be written as

$$\sum_{x\in\{0,1\}^n} p_x \boxed{x}, \tag{2}$$

where the p_x are non-negative real numbers. For example, the state of n fair coin flips can be described as

$$\sum_{x\in\{0,1\}^n} \frac{1}{2^n} \boxed{x}, \tag{3}$$

The fact that there is a "superposition" is not what accounts for the difference between classical and quantum computation.

> **The main difference is that the α_x can be negative whereas the p_x are non-negative.**

Why study quantum computation?

For this course, the main answer will be: there are problems which quantum computers can solve faster than classical computers. There are other strong answers such as quantum cryptography, quantum science (e.g., physics, chemistry, materials), quantum sensing, philosophy. (We will touch on some of these too in this course.)

Quantum algorithms.

Example problems – classical below refers to randomized classical, the most general class of classical:

1. SATISFIABILITY: e.g., $(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_3 \vee \neg x_2)$ (the fact that's it's a large AND of small ORs is important; a large OR of small ANDs is easy!) Best-known classical 2^n, best-known quantum $2^{n/2}$.
2. factoring ($15 = 5 \times 3$, $221 = 13 \times 17$, $30743126349163 = 4210601 \times 7301363$ make sure you the appreciate this is hard!) Best-known classical $2^{n^{1/3}}$, best-known quantum n^3, where n is the number of digits in the number to be factored.
3. simulating quantum systems. Problem size is n= number of quantum particles, e.g., electrons. Want to compute some properties of the system after time t, e..g. electron population in a certain state: Best-known classical $t2^n$, best-known quantum $t\,\mathrm{poly}(n)$.