Lecture 1: Quantum circuits and simulation tasks

Computational aspects of quantum simulation
ENS de Lyon

2026

Abstract

These notes introduce the pure-state circuit model of quantum computation. We give a formal definition of a circuit over a gate set, explain how a uniform family computes a decision problem, and define the complexity class BQP. We then distinguish state, amplitude, strong, and weak simulation, including their exact and approximate forms. Particular attention is paid to the output requested from a simulator: evaluating point probabilities, computing arbitrary marginals, and sampling from the output distribution are different forms of access and should not be conflated.

Learning objectives. After this lecture, students should be able to:

- calculate the state and measurement statistics of a small quantum circuit;
- give a formal description of a circuit as gates acting on ordered tuples of a qubit register;
- state the definition of a uniform quantum circuit family and of BQP;
- distinguish state, amplitude, strong, and weak simulation;
- state the relevant error criteria for strong and weak simulation;
- explain why access to arbitrary marginals is stronger than access only to complete-output probabilities.

Course introduction

Quantum simulation asks how to predict the behaviour of a quantum system from a compact description of its dynamics or energy. The word "simulation" hides several distinct computational tasks. We may want to evolve a state, estimate a measurement involving a few qubits, sample a measurement outcome, approximate the lowest possible energy, or describe equilibrium at nonzero temperature. A central habit of this course will be to state the input, output, error criterion, and computational resource explicitly before discussing an algorithm.

The course is organized around three regimes:

1. **Dynamics and circuits.** We first study the generic exponential state-vector algorithm and then identify structure that makes classical simulation efficient, such as restrictions on the allowed gates, limited correlations between groups of qubits, or a simple interaction pattern.
2. **Minimum-energy states.** We study states having or approximating the lowest possible energy, the computational difficulty of finding them, and algorithms that exploit additional structure. The goal is to understand both worst-case hardness and tractable families.
3. **Finite temperature.** We study how energy determines an equilibrium distribution, how to compute its normalization, and classical and quantum algorithms for estimating or sampling from it.

Across these topics, the recurring question is not merely whether a quantum system is large, but whether the relevant object has a succinct classical description. Different algorithms exploit different resources:

Representation	Favourable structure	Typical obstruction
Full list of amplitudes	few qubits	exponentially many basis states
Compact algebraic description	restricted operations	description growth
Decomposition into small pieces	weak correlations across cuts	correlation growth
Simplified optimization problem	weak or dense interactions	approximation size
Random samples	rapid convergence and nonnegative weights	slow convergence or cancellations

The course combines mathematical analysis with small computational experiments. Each lecture ends with one assignment, asking for a proof, an algorithm, or a numerical comparison. A student project may be a critical reading of a paper, a reproducibility study, a comparison of simulation methods, or a carefully delimited open question. The objective is to learn how to connect a structural theorem to an algorithm one could actually run, and to recognize precisely where its guarantee ceases to apply.

Suggested pacing for a two-hour lecture. Course introduction and Sections 1–2: 35 minutes; Section 3: 30 minutes; Section 4: 25 minutes; Sections 5.1–5.4: 25 minutes; assignment discussion and recap: 5 minutes. Several proofs may be left for independent reading.

Contents

1 Why simulate a quantum circuit?

A circuit is a compact description of a linear transformation acting on an exponentially large vector space. If a circuit acts on n qubits, its state is described naively by 2^n complex amplitudes, even when the circuit itself has only $\mathrm{poly}(n)$ gates. This mismatch is both the source of the potential power of quantum computation and the central difficulty of classical simulation.

There is no single computational problem called “simulate the circuit.” One may want a single amplitude, a probability, a sample, an expectation value, or a complete classical representation of the state. These outputs have very different sizes and may have very different computational complexities. A first task in any simulation result is therefore to specify exactly what is given, what must be returned, and what approximation is permitted.

Throughout the lecture, our basic input will be a circuit

$$C = U_m U_{m-1} \cdots U_1$$

on n qubits, where each U_j acts on at most a constant number of qubits. Unless stated otherwise, the initial state is a computational-basis state $|x\rangle$, usually $|0^n\rangle$.

2 Pure-state quantum formalism

2.1 States and Dirac notation

For this course all Hilbert spaces are finite-dimensional complex inner-product spaces. A pure state of a d-dimensional system is a unit vector $|\psi\rangle \in \mathbb{C}^d$. Vectors differing by a global phase describe the same physical state:

$$|\psi\rangle \quad \text{and} \quad e^{i\theta} |\psi\rangle$$

are operationally indistinguishable.

The symbol $\langle\psi|$ denotes the conjugate transpose of $|\psi\rangle$, and $\langle\phi|\psi\rangle$ denotes their inner product. The norm is $\|\psi\|_2 = \sqrt{\langle\psi|\psi\rangle}$. The outer product $|\psi\rangle\langle\phi|$ is the linear operator mapping $|v\rangle$ to $\langle\phi|v\rangle |\psi\rangle$.

2.2 Qubits and computational basis states

A qubit has state space $\mathbb{C}^2$, with computational basis

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

An arbitrary qubit state is

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \qquad |\alpha|^2 + |\beta|^2 = 1.$$

The state space of n qubits is

$$(\mathbb{C}^2)^{\otimes n} \cong \mathbb{C}^{2^n}.$$

For $x = x_1 \cdots x_n \in \{0,1\}^n$, write

$$|x\rangle = |x_1\rangle \otimes \cdots \otimes |x_n\rangle .$$

The 2^n states $\{|x\rangle : x \in \{0,1\}^n\}$ form an orthonormal basis, so every pure state has a unique expansion

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle , \qquad \sum_x |\alpha_x|^2 = 1.$$

2.3 Product states and entanglement

A two-part state is a product state if it can be written $|\phi\rangle_A \otimes |\eta\rangle_B$. Otherwise it is entangled. For example,

$$\left|\Phi^+\right\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

is entangled. Indeed, if it were $(a|0\rangle + b|1\rangle)(c|0\rangle + d|1\rangle)$, then the vanishing coefficients of $|01\rangle$ and $|10\rangle$ would imply $ad = bc = 0$, which is incompatible with both $ac \neq 0$ and $bd \neq 0$.

Entanglement is not by itself a criterion for classical hardness. Later lectures will identify restricted algebraic behaviour and limited correlations across a partition of the qubits as two mechanisms leading to efficient classical simulation.

2.4 Unitary evolution

The evolution of a closed quantum system is described by a unitary operator U, satisfying $U^\dagger U = UU^\dagger = \mathbb{I}$. It acts linearly:

$$|\psi\rangle \longmapsto U|\psi\rangle .$$

Unitarity preserves inner products and therefore normalization.

Important one-qubit gates include

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

and

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \qquad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.$$

The controlled-NOT gate is defined by

$$\mathrm{CNOT}\,|a,b\rangle = |a, a\oplus b\rangle .$$

2.5 Measurements

A computational-basis measurement of $|\psi\rangle = \sum_x \alpha_x |x\rangle$ returns x with probability

$$p(x) = |\alpha_x|^2.$$

After observing x, the post-measurement state is $|x\rangle$.

More generally, a projective measurement is specified by mutually orthogonal projectors Π_a satisfying $\sum_a \Pi_a = \mathbb{I}$. Outcome a has probability

$$p(a) = \langle\psi|\, \Pi_a \,|\psi\rangle \,,$$

and, when this probability is nonzero, the conditional post-measurement state is

$$\frac{\Pi_a \,|\psi\rangle}{\sqrt{p(a)}}.$$

For example, measuring only the first qubit and obtaining $b \in \{0, 1\}$ uses $\Pi_b = |b\rangle\langle b| \otimes \mathbb{I}$.

The amplitudes α_x are not probabilities. They are complex numbers that can cancel through interference. Probabilities arise only after taking squared absolute values or, more generally, evaluating the Born rule.

3 Quantum circuits

3.1 Gates on selected qubits

We follow the sequence-of-local-operations definition used by Kitaev, Shen, and Vyalyi [1]. A *gate set* $\mathcal{A}$ is a collection of unitary operators, each with a specified arity. If $G \in \mathcal{A}$ has arity k, if $n \geq k$, and if $A = (a_1, \ldots, a_k)$ is an ordered tuple of distinct elements of $[n]$, then $G[A]$ denotes the n-qubit unitary that applies G to qubits $a_1, \ldots, a_k$, in that order, and acts as the identity on the other qubits.

More explicitly, for $x \in \{0, 1\}^n$, let $x_A = (x_{a_1}, \ldots, x_{a_k})$, and let $x[A \leftarrow y]$ be obtained by replacing these entries by $y \in \{0, 1\}^k$. Then

$$G[A]\,|x\rangle = \sum_{y\in\{0,1\}^k} \langle y|\, G \,|x_A\rangle \,|x[A \leftarrow y]\rangle \,.$$

Definition 3.1 (Quantum circuit)**.** *An n-qubit quantum circuit over $\mathcal{A}$ is a finite sequence*

$$C = \big(G_1[A_1], G_2[A_2], \ldots, G_L[A_L]\big),$$

where $G_j \in \mathcal{A}$ and A_j is an ordered tuple of $\mathrm{arity}(G_j)$ *distinct qubits. The unitary realized by the circuit is*

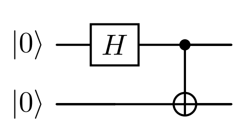
$$U_C = G_L[A_L] \cdots G_2[A_2]G_1[A_1].$$

The integer L is the size *of C. If its gates are grouped into layers whose supports are pairwise disjoint within each layer, the number of layers is the* depth *of that layered realization.*

The circuit description is classical: it specifies the name of each gate and the ordered qubits on which it acts. The order in the displayed product reflects the convention that $G_1[A_1]$ acts first.

3.2 Example: Bell and GHZ state preparation

On two qubits, take $G_1 = H[1]$ and $G_2 = \mathrm{CNOT}[1, 2]$. The circuit diagram is



and $U_C = \mathrm{CNOT}[1,2]H[1]$. Consequently,

$$|00\rangle \xmapsto{H[1]} \frac{|00\rangle + |10\rangle}{\sqrt{2}} \xmapsto{\mathrm{CNOT}[1,2]} \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

The same construction gives the n-qubit GHZ state. Define

$$U_{\mathrm{GHZ},n} = \mathrm{CNOT}[1,n] \cdots \mathrm{CNOT}[1,3]\,\mathrm{CNOT}[1,2]H[1].$$

Then

$$U_{\mathrm{GHZ},n}\,|0^n\rangle = \frac{|0^n\rangle + |1^n\rangle}{\sqrt{2}}.$$

This realization has size n. A binary tree of CNOT gates gives a different preparation with size $O(n)$ and depth $O(\log n)$.

3.3 Example: a reversible classical majority circuit

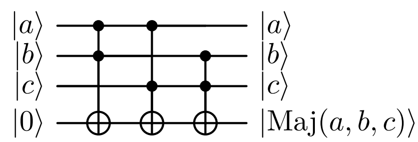
The Toffoli gate is the permutation unitary

$$\mathrm{TOF}\,|a,b,t\rangle = |a,b,t \oplus ab\rangle\,.$$

Three Toffoli gates compute the majority of three bits into an initially zero target, because

$$\mathrm{Maj}(a,b,c) = ab \oplus ac \oplus bc.$$

The circuit is



Every gate merely permutes computational-basis states, so on basis-state inputs the computation is classical and deterministic. Retaining the input bits makes the map reversible.

3.4 A fixed finite gate set

Complexity statements use circuits over one fixed finite gate set, so every circuit has a finite classical encoding. A standard choice is $\{H, T, \mathrm{CNOT}\}$, which can approximate arbitrary finite-qubit unitaries to any prescribed operator-norm accuracy. Replacing it by another efficiently universal finite set changes circuit size by at most polynomial overhead. We assume throughout that the matrix entries of the fixed gates can be computed to b bits in space polynomial in b. We will not need the quantitative approximation theorem here.

4 What does it mean to compute with a quantum circuit?

4.1 Uniform circuit families

A circuit is a finite object and therefore handles only one input length. A computational model needs a family of circuits together with an effective rule that produces them.

Definition 4.1 (Polynomial-time uniform family)**.** *Let $\mathcal{A}$ be a fixed finite gate set. A family $\{C_n\}_{n\in\mathbb{N}}$ of circuits over $\mathcal{A}$ is* polynomial-time uniform *if a deterministic classical Turing machine, on input 1^n, outputs the complete encoding of C_n in time* $\mathrm{poly}(n)$*. It has polynomial size if $|C_n| \le \mathrm{poly}(n)$.*

Suppose C_n acts on $N(n) = n + a(n)$ qubits. On input $x \in \{0,1\}^n$, the remaining $a(n)$ ancillas start in $\left|0^{a(n)}\right\rangle$. Taking the first qubit as the output qubit, define

$$p^x_{C_n} = \left\langle x, 0^{a(n)} \right| U^\dagger_{C_n} (|1\rangle\langle 1| \otimes \mathbb{I}_{N(n)-1}) U_{C_n} \left| x, 0^{a(n)} \right\rangle .$$

Definition 4.2 (The class BQP). *A language $L \subseteq \{0,1\}^*$ is in BQP if there exist polynomially bounded $a(n)$ and a polynomial-time uniform, polynomial-size circuit family $\{C_n\}$ such that, for every $x \in \{0,1\}^n$,*

$$\begin{cases} p^x_{C_n} \geq 2/3, & x \in L, \\ p^x_{C_n} \leq 1/3, & x \notin L. \end{cases}$$

Remark 4.3 (Error reduction). *The constants $2/3$ and $1/3$ are conventional. If one run is correct with probability at least $1/2 + \gamma$, repeat independently r times and take the majority. Hoeffding's inequality bounds the new error by $e^{-2\gamma^2 r}$. Thus an inverse-polynomial gap can be amplified to error $2^{-\operatorname{poly}(n)}$ using polynomially many repetitions.*

Recall that BPP is defined in the same bounded-error way using a polynomial-time classical randomized Turing machine.

Theorem 4.4. $\mathsf{BPP} \subseteq \mathsf{BQP}$.

Proof. First make the deterministic part of a classical computation reversible, using Toffoli gates and polynomially many ancillas; Toffoli has a constant-size realization over the standard gate set above. Independent unbiased random bits are produced by applying H to fresh $|0\rangle$ qubits and then using those qubits as coherent controls. Deferred measurement allows all measurements to be placed at the end. The resulting uniform quantum circuit has only polynomial overhead and the same acceptance probability as the randomized computation. □

No unconditional separation between BPP and BQP is known. Also, BQP is a class of decision problems: it specifies one accept/reject bit, not the problem of reproducing an arbitrary circuit's full output distribution.

5 What does it mean to simulate a circuit?

Fix a circuit C on N qubits and a computational-basis input $x \in \{0,1\}^N$. Its full computational-basis output distribution is

$$p^x_C(z) = |\langle z| U_C |x\rangle|^2, \qquad z \in \{0,1\}^N.$$

For an ordered tuple $S = (s_1, \ldots, s_r)$ of distinct measured qubits and $y \in \{0,1\}^r$, define the marginal

$$p^x_{C,S}(y) = \sum_{z : z_S = y} p^x_C(z).$$

The encoding of C, x, S, y, and any requested accuracy parameters, is part of the input to the classical simulation algorithm.

5.1 State simulation

One may ask for a classical data structure representing $U_C |x\rangle$. This is not one formal problem until the admissible data structure and the queries it must support are specified. An explicit vector, for example, supports amplitude queries, marginals, and sampling, but has 2^N entries. A succinct data structure is useful only together with algorithms for the desired queries and explicit bounds on their running times and approximation errors.

5.2 Amplitude computation

Given $x, z \in \{0,1\}^N$, the amplitude problem asks for

$$A_C(x,z) = \langle z| U_C |x\rangle .$$

An additive ϵ-approximation is a complex number $\widetilde{A}$ satisfying $|\widetilde{A} - A_C(x,z)| \le \epsilon$.

5.3 Strong simulation

Strong simulation asks for a specified marginal probability $p^x_{C,S}(y)$. Common approximation guarantees are

$$|\widetilde{p} - p^x_{C,S}(y)| \le \epsilon \quad \text{(additive error)}$$

and

$$|\widetilde{p} - p^x_{C,S}(y)| \le \epsilon p^x_{C,S}(y) \quad \text{(relative error).}$$

For a randomized estimator, the chosen inequality is required with a stated success probability, say at least $1-\delta$. Exact strong simulation also requires a convention for writing probabilities exactly: for the chosen gate set, one must specify a symbolic number format and the cost of arithmetic and comparison in that format.

Relative error is much stronger for exponentially small probabilities. When $p = 0$, the displayed relative-error convention forces $\widetilde{p} = 0$. One must also state whether S is arbitrary, fixed, or equal to all output qubits.

5.4 Weak simulation

Weak simulation asks for samples. A randomized classical algorithm is an exact weak simulator if its output distribution is exactly p^x_C. It is an ϵ-approximate weak simulator if its output distribution q^x_C satisfies

$$\|p^x_C - q^x_C\|_{\mathrm{TV}} := \frac{1}{2} \sum_{z\in\{0,1\}^N} |p^x_C(z) - q^x_C(z)| \le \epsilon.$$

This is a guarantee on the distribution produced by the algorithm, not an estimate inferred from finitely many samples. For every event $E \subseteq \{0,1\}^N$, it implies $|p^x_C(E) - q^x_C(E)| \le \epsilon$.

6 Assignment

1. **Strong simulation with marginals implies weak simulation.** Let p be the computational-basis output distribution of an N-qubit circuit. Suppose that, for every prefix $u \in \{0,1\}^k$, $0 \le k \le N$, a polynomial-time strong simulator returns the exact marginal

   $$m(u) := \Pr_{Z\sim p}[Z_1 \cdots Z_k = u].$$

   Assume that the arithmetic model permits exact sampling of a Bernoulli random variable whose parameter is a ratio of probabilities returned by the simulator.

   (a) Give a sequential algorithm that samples $Z_1, Z_2, \ldots, Z_N$, using prefix marginals to determine the conditional distribution of each bit.

   (b) Prove that the probability that your algorithm outputs any $z \in \{0,1\}^N$ is exactly $p(z)$. Explain why no zero-mass prefix causes a division by zero.

   (c) Prove that the algorithm makes at most $2N$ calls to the strong simulator and runs in polynomial time under the stated arithmetic assumption.

References

[1] Alexei Yu. Kitaev, Alexander H. Shen, and Mikhail N. Vyalyi, *Classical and Quantum Computation*, Graduate Studies in Mathematics 47, American Mathematical Society, 2002; see Definition 6.1 for quantum circuits.

[2] John Watrous, *An Introduction to Quantum Information and Quantum Circuits*, 2011. https://cs.uwaterloo.ca/~watrous/Papers/IntroductionQuantumCircuits.pdf

[3] Ronald de Wolf, *Quantum Computing: Lecture Notes*, arXiv:1907.09415. https://arxiv.org/abs/1907.09415

[4] John Watrous, *Quantum Computational Complexity*, arXiv:0804.3401. https://arxiv.org/abs/0804.3401

[5] Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information*, 10th anniversary edition, Cambridge University Press, 2010.

[6] Maarten Van den Nest, *Classical Simulation of Quantum Computation, the Gottesman-Knill Theorem, and Slightly Beyond*, Quantum Information and Computation 10, 258–271 (2010). https://arxiv.org/abs/0811.0898