Lecture 2: Simulation algorithms and stabilizer states

Computational aspects of quantum simulation
ENS de Lyon

2026

Abstract

We first revisit the simulation tasks from Lecture 1 through concrete circuits and solve the first assignment. We then introduce expectation-value estimation and compare it carefully with strong and weak simulation. Generic state-vector and path-enumeration algorithms lead to $\mathsf{BQP} \subseteq \mathsf{PSPACE}$. Finally, we introduce stabilizer states as a concise classical representation and work through basic examples.

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

- analyze strong and weak simulation on several multi-qubit examples;
- prove that exact access to arbitrary marginals gives an exact sampler;
- relate expectation-value estimation precisely to strong and weak simulation;
- implement and analyze generic state-vector simulation;
- prove the inclusion $\mathsf{BQP} \subseteq \mathsf{PSPACE}$;
- represent stabilizer states using commuting Pauli generators.

Suggested pacing for a two-hour lecture. Recap and examples: 25 minutes; exercise and solution: 15 minutes; expectation values: 15 minutes; generic simulation: 20 minutes; stabilizer states and examples: 45 minutes.

Contents

1 Recap and notation

Fix an N-qubit circuit C, a computational-basis input $x \in \{0,1\}^N$, and the final state

$$|\psi\rangle = U_C|x\rangle = \sum_{z\in\{0,1\}^N} \alpha_z|z\rangle.$$

A computational-basis measurement of every qubit produces the distribution

$$p_C^x(z) = |\alpha_z|^2.$$

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

$$p_{C,S}^x(y) = \sum_{z:z_S=y} p_C^x(z).$$

Lecture 1 distinguished state simulation, amplitude computation, strong simulation of a specified marginal, and weak simulation by sampling. We begin by applying these definitions to several circuits.

2 Worked examples

2.1 The GHZ circuit

Let C prepare

$$|\mathrm{GHZ}_N\rangle = \frac{|0^N\rangle + |1^N\rangle}{\sqrt{2}}$$

from $|0^N\rangle$. Its only nonzero amplitudes are

$$\langle 0^N|U_C|0^N\rangle = \langle 1^N|U_C|0^N\rangle = \frac{1}{\sqrt{2}}.$$

Thus the complete outcomes 0^N and 1^N each have probability $1/2$. For every nonempty subset S, the marginal on S is supported on $0^{|S|}$ and $1^{|S|}$, again with probabilities $1/2$. An exact weak simulator draws one fair bit b and returns b^N.

This example has a concise description despite involving N qubits: the state is determined by two basis strings and two amplitudes. Such "sparse" states can be concisely represented.

2.2 A purely classical reversible circuit

Consider a circuit containing only Toffoli gates and supplied with a computational-basis input $|x\rangle$. Every Toffoli gate permutes basis states, so the circuit implements a permutation

$$F_C : \{0,1\}^N \longrightarrow \{0,1\}^N, \qquad U_C|x\rangle = |F_C(x)\rangle.$$

The string $F_C(x)$ is computed by applying the L classical reversible updates in sequence, in time $O(L)$ and space $O(N)$. Consequently:

- a succinct state representation is the single string $F_C(x)$;
- $\langle z|U_C|x\rangle$ is 1 when $z = F_C(x)$, and 0 otherwise;
- every marginal probability is either 0 or 1, determined by comparing the requested bits with $F_C(x)$;
- an exact weak simulator returns $F_C(x)$ deterministically.

Thus all four simulation tasks from Lecture 1 are efficient for this input model.

2.3 Efficient simulation of all circuits would decide BQP

The acceptance probability of a BQP circuit is a one-qubit marginal, and its acceptance bit is part of a sample from the complete output distribution. This gives the following two consequences.

Proposition 2.1. *Suppose there is a classical polynomial-time algorithm applicable to every polynomial-size quantum circuit of a fixed finite gate set.*

(i) If it strongly simulates the acceptance probability to additive error η, for some fixed constant $\eta < 1/2$, with success probability at least $2/3$, then BQP $\subseteq$ BPP.

(ii) If it weakly simulates the output distribution to total-variation error τ, for some fixed constant $\tau < 1/2$, then BQP $\subseteq$ BPP. *Here, for distributions p and q on a finite set Ω, their total-variation distance is*

$$d_{\mathrm{TV}}(p,q) = \frac{1}{2}\sum_{z\in\Omega} |p(z) - q(z)| = \max_{A\subseteq\Omega} |p(A) - q(A)|.$$

Proof. For the implication under consideration, let $\xi = \eta$ in case (i) and $\xi = \tau$ in case (ii), and choose a constant ϵ satisfying $0 < \epsilon < 1/2 - \xi$. Standard BQP error reduction replaces the original uniform circuit family by a polynomial-size uniform family whose acceptance probability p satisfies

$$p \geq 1 - \epsilon \quad \text{in a yes instance,} \qquad p \leq \epsilon \quad \text{in a no instance.}$$

In case (i), whenever the strong simulator succeeds, its estimate is at least $1 - \epsilon - \eta > 1/2$ in a yes instance and at most $\epsilon + \eta < 1/2$ in a no instance. Comparing the estimate with $1/2$ therefore decides the language with probability at least $2/3$.

In case (ii), let q be the acceptance probability under the distribution produced by the weak simulator. By the event characterization of total-variation distance, applied to the event that the output bit is one,

$$|p - q| \leq \tau.$$

Consequently $q \geq 1 - \epsilon - \tau > 1/2$ in a yes instance and $q \leq \epsilon + \tau < 1/2$ in a no instance. The distance of q from $1/2$ is a positive constant, so polynomially many independent samples, followed by a majority vote, distinguish the two cases with bounded error. Uniformity ensures that the amplified circuit can be generated in polynomial time. □

The statement is one-way. A BPP decision algorithm for every language in BQP returns one bit and need not reproduce the output distribution of an arbitrary circuit.

2.4 Many-path interference: a parity phase oracle

For $s \in \{0,1\}^N$, suppose a phase oracle acts as

$$O_s|z\rangle = (-1)^{s\cdot z}|z\rangle,$$

where the inner product is modulo two. Consider $C_s = H^{\otimes N} O_s H^{\otimes N}$. Starting from $|0^N\rangle$, the amplitude of $|y\rangle$ is

$$\langle y|U_{C_s}|0^N\rangle = 2^{-N} \sum_{z\in\{0,1\}^N} (-1)^{(s\oplus y)\cdot z} = \begin{cases} 1, & y = s, \\ 0, & y \neq s. \end{cases}$$

All 2^N paths contribute, but pairwise cancellation makes the amplitude zero for every incorrect output. The final distribution is concentrated at s.

Remark 2.2. *Note that, in the oracle model where the objective is to find s, this says that a weak-simulation oracle can be stronger than a strong-simulation oracle that returns complete-output probabilities. One call to the weak-simulation oracle returns s, whereas a strong-simulation query at y only tests whether $y = s$.*

3 Exercise from Lecture 1

Problem

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 $Z_1, \ldots, Z_N$.

(b) Prove that its output distribution is exactly p, including when some prefixes have zero probability.

(c) Bound the number of calls to the strong simulator and the running time.

Solution

Start with the empty prefix $u = \varnothing$. At step k, query $m(u0)$ and $m(u1)$, and set

$$r_b = \frac{m(ub)}{m(u0) + m(u1)}, \qquad b \in \{0,1\}.$$

Draw the next bit according to (r_0, r_1), append it to u, and continue until N bits have been generated.

We prove inductively that every prefix reached by the algorithm has positive mass. The empty prefix has mass one. If $m(u) > 0$, then

$$m(u) = m(u0) + m(u1) > 0,$$

so the denominator is nonzero, and a zero-mass child is selected with probability zero. Hence no division by zero occurs along a realized execution.

For $z = z_1 \cdots z_N$ with $p(z) > 0$, the probability of generating z is

$$\prod_{k=1}^{N} \frac{m(z_1 \cdots z_k)}{m(z_1 \cdots z_{k-1})} = \frac{m(z)}{m(\varnothing)} = p(z).$$

If $p(z) = 0$, let k be the first index for which $m(z_1 \cdots z_k) = 0$. The transition into that prefix has probability zero, so the algorithm never outputs z. The output distribution is therefore exactly p.

There are two marginal queries at each of the N steps, hence at most $2N$ calls. All remaining operations are polynomial-time arithmetic and Bernoulli sampling under the assumption in the problem. The complete algorithm therefore runs in polynomial time.

4 Expectation-value estimation

4.1 Definition

Let O be a Hermitian operator with a succinct classical description. The quantity of interest is

$$\langle O \rangle_{C,x} = \langle x | U_C^\dagger O U_C | x \rangle.$$

If

$$O = \sum_a \lambda_a \Pi_a$$

is the spectral decomposition of O, measuring the projectors $\{\Pi_a\}_a$ produces a random variable equal to λ_a with probability

$$\langle x | U_C^\dagger \Pi_a U_C | x \rangle.$$

Its expectation is precisely $\langle O \rangle_{C,x}$, which explains the terminology.

An additive (ϵ, δ)-estimator for O is a randomized classical algorithm that outputs v such that

$$\Pr[|v - \langle O \rangle_{C,x}| \leq \epsilon] \geq 1 - \delta.$$

4.2 From expectation values to strong simulation

For a measured subset S and $y \in \{0,1\}^{|S|}$, define

$$\Pi_{S,y} = |y\rangle\langle y|_S \otimes \mathbb{I}_{\overline{S}}.$$

The state generated by the circuit on input x is

$$U_C |x\rangle = \sum_{z \in \{0,1\}^N} \alpha_z |z\rangle.$$

Then

$$\langle x | U_C^\dagger \Pi_{S,y} U_C | x \rangle = \sum_{z : z_S = y} |\alpha_z|^2 = p^x_{C,S}(y).$$

Consequently, an algorithm accepting every succinct Hermitian observable as part of its input implies strong simulation whenever the permitted descriptions include all projectors $\Pi_{S,y}$. Exact expectation evaluation gives exact strong simulation, and additive (ϵ, δ)-estimation gives the corresponding additive strong simulation.

4.3 From weak simulation to expectation values of diagonal observables

Consider a computational-basis diagonal observable

$$O_f = \sum_z f(z)|z\rangle\langle z|, \qquad \langle O_f\rangle_{C,x} = \mathbb{E}_{z\sim p_C^x} f(z).$$

As a result, suppose if we have a weak simulator produces independent samples $z^{(1)}, \ldots, z^{(M)}$ from p_C^x, a natural estimator for the expectation value is

$$\widehat{\mu} = \frac{1}{M}\sum_{j=1}^{M} f(z^{(j)}).$$

Hoeffding's inequality gives

$$\Pr[|\widehat{\mu} - \mathbb{E}_p f| > \epsilon] \leq 2\exp\left(-\frac{M\epsilon^2}{2B^2}\right).$$

assuming $f(x) \in [-B, B]$. Thus

$$M \geq \frac{2B^2}{\epsilon^2}\ln\frac{2}{\delta}$$

suffices for additive error ϵ with failure probability at most δ.

For example, if $T \subseteq [N]$ and $Z_T = \prod_{j\in T} Z_j$, then

$$\langle Z_T\rangle_{C,x} = \mathbb{E}_{z\sim p_C^x}\left[(-1)^{\sum_{j\in T} z_j}\right].$$

A general Hermitian observable need not be diagonal in the computational basis. Weak simulation applies when its measurement can be implemented efficiently: append a circuit that maps the relevant eigenbasis to the computational basis, weakly simulate the augmented circuit, and attach the corresponding eigenvalue to each sample. Without such a measurement procedure, computational-basis samples alone do not determine $\langle O\rangle$.

5 Generic classical simulation algorithms

5.1 State-vector evolution

The direct simulator stores

$$|\psi_t\rangle = U_t \cdots U_1|x\rangle = \sum_{z\in\{0,1\}^N} \alpha_z^{(t)}|z\rangle.$$

Initially, $\alpha_z^{(0)} = 1$ for $z = x$ and $\alpha_z^{(0)} = 0$ otherwise. Suppose the next gate $U_t = G = (g_{ab})_{a,b\in\{0,1\}}$ acts on qubit j. Fix an assignment $r \in \{0,1\}^{N-1}$ of all qubits other than j, and let $z^{(0)}$ and $z^{(1)}$ be the two full strings that agree with r and have bit j equal to 0 and 1, respectively. The simulator replaces the two old amplitudes simultaneously by

$$\alpha_{z^{(0)}}^{(t)} = g_{00}\alpha_{z^{(0)}}^{(t-1)} + g_{01}\alpha_{z^{(1)}}^{(t-1)},$$
$$\alpha_{z^{(1)}}^{(t)} = g_{10}\alpha_{z^{(0)}}^{(t-1)} + g_{11}\alpha_{z^{(1)}}^{(t-1)}.$$

Both right-hand sides must be evaluated from the amplitudes at time $t-1$ before either stored value is overwritten. Repeating this operation for every r updates all 2^N amplitudes, since each basis string occurs in exactly one pair.

More generally, if a k-qubit gate G acts on an ordered set of qubits J, fix an assignment r on the complementary qubits. For $u \in \{0,1\}^k$, write (r,u) for the full string whose bits on J are u. The corresponding block is updated by

$$\alpha^{(t)}_{(r,u)} = \sum_{v\in\{0,1\}^k} G_{u,v}\alpha^{(t-1)}_{(r,v)} \qquad \text{for every } u \in \{0,1\}^k.$$

There are 2^{N-k} disjoint blocks, each containing 2^k amplitudes.

Proposition 5.1 (State-vector simulation)**.** *For constant-arity gates, a size-L, N-qubit circuit can be simulated using $O(2^N)$ complex numbers and $O(L2^N)$ arithmetic operations.*

Once the final vector is stored, an amplitude can be read directly, a marginal can be obtained by summing selected $|\alpha_z|^2$, and an output sample can be drawn from the probability array. If $O = \sum_{j=1}^{M} O_j$, where every O_j acts on at most a constant number of qubits, then $\langle O\rangle$ can be computed in $O(M2^N)$ arithmetic operations.

5.2 Enumeration of computational paths

Writing $U_C = U_L \cdots U_1$ and inserting computational-basis identities between consecutive gates gives

$$\langle y|U_C|x\rangle = \sum_{z_1,\ldots,z_{L-1}} \prod_{t=1}^{L} \langle z_t|U_t|z_{t-1}\rangle,$$

where $z_0 = x$, $z_L = y$, and every intermediate string ranges over $\{0,1\}^N$. Every factor is a matrix element of a constant-size gate. The number of paths may be exponential, but the paths can be enumerated one at a time using counters of polynomial length; the full state vector need not be stored.

5.3 Application: BQP $\subseteq$ PSPACE

Definition 5.2 (The class PSPACE)**.** *A language lies in* PSPACE *if a deterministic Turing machine decides it using at most polynomially many work-tape cells in the input length. No polynomial time bound is imposed, but the machine must halt on every input.*

Theorem 5.3. BQP $\subseteq$ PSPACE*.*

Proof. Let $\{C_n\}$ be a polynomial-time uniform, polynomial-size circuit family deciding a language in BQP. On input $x \in \{0,1\}^n$, let $|\xi\rangle = |x, 0^{a(n)}\rangle$, let $N = n + a(n)$, and write $U_C = U_L \cdots U_1$. If $P = |1\rangle\langle 1| \otimes \mathbb{I}_{N-1}$ is the accepting projector, then

$$p_C(x) = \langle \xi | U_C^\dagger P U_C | \xi \rangle.$$

Put $z_0 = w_0 = \xi$ and $z_L = w_L = y$. Inserting computational-basis identities on both sides of P gives

$$p_C(x) = \sum_{\substack{y\in\{0,1\}^N \\ y_1=1}} \sum_{z_1,\ldots,z_{L-1}} \sum_{w_1,\ldots,w_{L-1}} \prod_{t=1}^{L} \langle z_t|U_t|z_{t-1}\rangle \overline{\langle w_t|U_t|w_{t-1}\rangle}.$$

There are at most

$$R = 2^{N(2L-1)} = 2^{\mathrm{poly}(n)}$$

summands. Nested binary counters enumerate them one at a time. The basis labels and counters use $N(2L-1) = \mathrm{poly}(n)$ bits. Uniformity allows each gate description to be produced as needed in polynomial space, and only the current product and a running complex accumulator must also be stored.

Every gate entry has magnitude at most one, so every exact summand does as well. The absolute value of a partial sum is at most R, requiring $\log R + O(1) = \mathrm{poly}(n)$ integer bits. Let $R = 2^{q(n)}$. Compute every gate entry and arithmetic operation with

$$b = q(n) + O(\log L) + O(1)$$

fractional guard bits. A telescoping bound for a product of $2L$ factors shows that the error in one summand is $O(L2^{-b})$. Summing at most R terms, including rounding in the accumulator, gives total error $O(RL2^{-b})$. The constant hidden in the final $O(1)$ can be chosen so that this error is less than $1/12$. The fixed-gate computability assumption from Lecture 1 ensures that gate entries can be generated to b bits in space polynomial in b. Hence the entire computation uses polynomial space.

In a yes instance the estimate is at least $2/3 - 1/12 = 7/12$, whereas in a no instance it is at most $1/3 + 1/12 = 5/12$. Comparing with $1/2$ decides the language. The enumeration may take exponential time, which is allowed in PSPACE. □

6 Stabilizer states: a concise representation

6.1 The Pauli group

The one-qubit Pauli matrices are

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

They are Hermitian and unitary, and any two distinct nonidentity matrices among X, Y, Z anticommute. The N-qubit Pauli group is

$$\mathcal{P}_N = \{ i^r P_1 \otimes \cdots \otimes P_N : r \in \{0,1,2,3\},\ P_j \in \{I, X, Y, Z\} \}.$$

The phases $\{1, i, -1, -i\}$ are included because the Hermitian Pauli operators are not closed under multiplication. For example, X and Y are Hermitian, but $XY = iZ$ is not. The full Pauli group is therefore the natural object for multiplication and for the definition of the Clifford group. Stabilizer generators and Pauli observables, on the other hand, will always be Hermitian.

The operators X and Z already generate every Pauli up to an overall phase. More precisely, for $a, b \in \mathbb{F}_2^N$, the $2N$-bit string (a, b) labels

$$X^{a_1} Z^{b_1} \otimes \cdots \otimes X^{a_N} Z^{b_N}$$

up to a scalar in $\{1, i, -1, -i\}$. Two Paulis have the same binary label exactly when they differ only by such a scalar. We choose one canonical Hermitian representative of each label by setting

$$P(a, b) = i^{\sum_j a_j b_j} X^{a_1} Z^{b_1} \otimes \cdots \otimes X^{a_N} Z^{b_N}.$$

Indeed, $XZ = -iY$, so the factor i at every position where $a_j = b_j = 1$ converts the local factor XZ into Y. Thus the single-qubit labels correspond to

$$(0,0) \longleftrightarrow I, \qquad (1,0) \longleftrightarrow X, \qquad (0,1) \longleftrightarrow Z, \qquad (1,1) \longleftrightarrow Y,$$

and $P(a,b)$ is Hermitian and satisfies $P(a,b)^2 = I$.

There are nevertheless two Hermitian choices for a fixed label, namely $P(a,b)$ and $-P(a,b)$. Consequently every Hermitian Pauli is uniquely of the form

$$(-1)^\sigma P(a,b), \qquad \sigma \in \mathbb{F}_2.$$

The binary label (a,b) specifies the tensor product of I, X, Y, Z up to sign, while σ records the remaining sign. Multiplication by one additional factor of i gives the two non-Hermitian elements with the same label and hence recovers the full Pauli group.

On the binary label space, define

$$\omega((a,b),(a',b')) = a \cdot b' - b \cdot a' \pmod 2.$$

This is the symplectic inner product. It is nondegenerate: if its value vanishes against every (a',b'), choosing b' and then a' among the standard basis vectors forces $a = b = 0$. Two Paulis commute exactly when the symplectic inner product of their labels is zero.

6.2 Stabilizer groups and stabilizer states

The restriction to Hermitian elements has a direct meaning here. If $P = P^\dagger$ is a Pauli, then iP and $-iP$ have eigenvalues $\pm i$, so neither can satisfy $g|\psi\rangle = |\psi\rangle$ for a nonzero vector. Thus a Pauli that stabilizes a nonzero state with eigenvalue $+1$ is automatically Hermitian. Hermitian Paulis are also precisely the Paulis that can be used as observables with outcomes ± 1.

Definition 6.1 (Stabilizer group)**.** *A stabilizer group is an abelian subgroup $S \subset \mathcal{P}_N$ whose elements are Hermitian and which does not contain $-I$. A vector $|\psi\rangle$ is stabilized by S if*

$$g|\psi\rangle = |\psi\rangle \qquad \textit{for every } g \in S.$$

A set of generators is independent if no nonempty product of them is the identity.

Proposition 6.2 (Dimension of the stabilized subspace)**.** *If $S = \langle g_1, \ldots, g_r \rangle$ has r independent commuting generators, then its simultaneous $+1$ eigenspace has dimension 2^{N-r}. In particular, N independent generators determine a unique pure state. Such a state is called a stabilizer state.*

Proof. For each generator, set

$$Q_j = \frac{I + g_j}{2}.$$

Since g_j is Hermitian and $g_j^2 = I$ it has eigenvalues ± 1. Thus Q_j is the orthogonal projector onto the $+1$ eigenspace of g_j. The generators commute, so the projectors $Q_1, \ldots, Q_r$ commute as well. Their product is therefore an orthogonal projector. Moreover, if $|\psi\rangle$ is such that $g_k|\psi\rangle = |\psi\rangle$ for all k, then $\prod_k Q_k|\psi\rangle = |\psi\rangle$ and $|\psi\rangle$ is in the image of $\prod_k Q_k$. Conversely, as $Q_j \prod_k Q_k = \prod_k Q_k$, the image of $\prod_k Q_k$ is included in the image of Q_j. It follows that the projector onto the simultaneous $+1$ eigenspace is

$$\Pi_S = \prod_{j=1}^{r} \frac{I + g_j}{2} = \frac{1}{2^r} \sum_{A \subseteq \{1,\ldots,r\}} \prod_{j \in A} g_j.$$

Independence means that the 2^r subset products are distinct, and they are precisely the elements of S. Hence

$$\Pi_S = \frac{1}{2^r} \sum_{g \in S} g.$$

Because $-I \notin S$, the only scalar Pauli in S is I. Every other element of S has a nonidentity single-qubit Pauli factor and therefore has trace zero, while $\mathrm{Tr}(I) = 2^N$. Since the trace of an orthogonal projector equals the dimension of its image, we obtain

$$\dim(\mathrm{im}\, \Pi_S) = \mathrm{Tr}(\Pi_S) = 2^{N-r}.$$

□

A stabilizer state is represented by N generators rather than by 2^N amplitudes. Each generator needs $2N$ binary Pauli labels and a phase bit, so the complete representation has $O(N^2)$ bits. When $r = N$, the projector has rank one and gives the useful identity

$$|\psi\rangle\langle\psi| = \Pi_S = \frac{1}{2^N} \sum_{g \in S} g.$$

6.3 Examples

The basis state $|x\rangle$ is the unique state stabilized by

$$(-1)^{x_1} Z_1, \ldots, (-1)^{x_N} Z_N.$$

The Bell state

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

is stabilized by the two independent commuting operators X_1X_2 and Z_1Z_2. The projector formula is explicit in this case:

$$\Pi_S = \frac{1}{4}(I + X_1X_2)(I + Z_1Z_2) = \frac{1}{4}(I + X_1X_2 + Z_1Z_2 - Y_1Y_2) = |\Phi^+\rangle\langle\Phi^+|.$$

More generally, the GHZ state is stabilized by

$$X_1X_2 \cdots X_N, \quad Z_1Z_2, \quad Z_2Z_3, \quad \ldots, \quad Z_{N-1}Z_N.$$

These are N independent commuting generators. Stabilizer states can therefore be highly entangled even though their algebraic description is quadratic in N.

7 Exercise: a stabilizer and a non-stabilizer state

Consider the two-qubit states

$$|\psi\rangle = \frac{|00\rangle + i|11\rangle}{\sqrt{2}}, \qquad |\phi\rangle = \frac{|00\rangle + e^{i\pi/4}|11\rangle}{\sqrt{2}}.$$

(a) Verify that Z_1Z_2 and X_1Y_2 are independent commuting Hermitian Paulis and that both stabilize $|\psi\rangle$. Conclude that $|\psi\rangle$ is a stabilizer state, and list the four elements of the stabilizer group generated by these two operators.

(b) Prove that $|\phi\rangle$ is not a stabilizer state. As a first step, show that a Pauli stabilizing $|\phi\rangle$ must preserve the set of computational-basis strings $\{00, 11\}$. Up to sign, it must therefore be one of

$$II,\ ZI,\ IZ,\ ZZ,\ XX,\ XY,\ YX,\ YY.$$

Check which signed operators in this list stabilize $|\phi\rangle$, and show that there cannot be two independent stabilizer generators.