Quantum Computing

Unit 5: Analysis Tools, Discrete Logarithms & Schmidt Decompositions

Dr. Gopal Chandra Jana

Dept. of CSE, Sharda University

Quantum Computing (CSE063) — Unit 5

August 23, 2026

Course Page:

https://www.gcjana.in/courses/shardauniversity/2501/quantum-computing/

Outline

Learning Outcomes of Unit 5

After this unit, a student will be able to...

1. **Apply** Markov, Chebyshev, Chernoff/Hoeffding and the union bound to analyse algorithms.
2. **Boost** a randomised algorithm's success probability and count the repetitions needed.
3. **Describe** Shor's discrete-logarithm algorithm and the congruence it produces.
4. **Handle** the case where the order r of a is *composite*, recovering t via CRT.
5. **Compute** a Schmidt decomposition of a bipartite state using the SVD (or reduced density matrix).
6. **Read off** the Schmidt rank and coefficients to detect and quantify entanglement.

Mapping to the Syllabus

A. Tools for analysing probabilistic algorithms. **B.** Discrete logarithms when the order of a is composite. **C.** Computing Schmidt decompositions.

Why These Tools?

The setting

Randomised and quantum algorithms succeed only with some *probability*. To turn "often right" into "right with near-certainty" we need quantitative **tail bounds**.

The toolbox

Expectation & variance $\rightarrow$ **Markov** $\rightarrow$ **Chebyshev** $\rightarrow$ **Chernoff/Hoeffding** $\rightarrow$ **union bound** $\rightarrow$ *amplification*.

- How likely is a random variable to stray far from its mean?
- How many repetitions make the error negligible?
- How many samples estimate a probability accurately?

Recurring use

These underlie BPP/BQP error reduction and the post-processing of every algorithm in this course.

Expectation and Variance (Recap)

Definitions

$$\mathbb{E}[X] = \sum_x x \Pr[X = x],$$
$$\mathrm{Var}(X) = \mathbb{E}\big[(X - \mathbb{E}[X])^2\big].$$

Also $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$, and $\sigma = \sqrt{\mathrm{Var}(X)}$.

Linearity (always)

$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$, *even if* X, Y *are dependent.*

Independence helps variance

If $X_1, \ldots, X_n$ are *independent*,

$$\mathrm{Var}\left(\sum_i X_i\right) = \sum_i \mathrm{Var}(X_i).$$

Geometric trials

If each trial succeeds independently with probability p, the expected number of trials until the first success is $\boxed{1/p}$.

Markov's Inequality

Statement

For a **non-negative** random variable X and any $a > 0$,

$$\boxed{\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}}$$

Reading it

A non-negative quantity is rarely much larger than its average — the weakest but most general tail bound (needs only the mean).

Example

If the expected running time is $\mathbb{E}[T] = 100$, then $\Pr[T \geq 1000] \leq 100/1000 = 0.1$.

Chebyshev's Inequality

Statement

For any random variable with mean μ and variance σ^2, and any $a > 0$,

$$\boxed{\Pr\left[\,|X-\mu| \geq a\,\right] \;\leq\; \frac{\sigma^2}{a^2}} \qquad \text{equivalently} \qquad \Pr\left[\,|X-\mu| \geq k\sigma\,\right] \leq \frac{1}{k^2}.$$

Stronger than Markov

Uses the *variance*: concentration improves as σ shrinks. Proof: apply Markov to $(X-\mu)^2$.

Sampling use

To estimate a probability p to accuracy ε from n samples, $\mathrm{Var}(\bar{X}) = p(1-p)/n$, so $n = O(1/\varepsilon^2)$ samples suffice by Chebyshev.

Chernoff / Hoeffding Bounds

Exponential concentration for sums of independent variables

Let $X = \sum_{i=1}^{n} X_i$ with independent $X_i \in \{0,1\}$ and $\mu = \mathbb{E}[X]$. For $0 < \delta < 1$,

$$\Pr[X \geq (1+\delta)\mu] \leq e^{-\mu\delta^2/3}, \qquad \Pr[X \leq (1-\delta)\mu] \leq e^{-\mu\delta^2/2}.$$

(Hoeffding, for $X_i \in [0,1]$ and the mean $\bar{X}$: $\Pr[|\bar{X} - \mathbb{E}\bar{X}| \geq \varepsilon] \leq 2e^{-2n\varepsilon^2}$.)

The key gain

The failure probability decays **exponentially** in n — far stronger than Chebyshev's $1/a^2$.

Union bound (Boole)

For any events, $\Pr\left[\bigcup_i A_i\right] \leq \sum_i \Pr[A_i]$ — bound many bad events at once.

Probability Amplification (Boosting)

Majority voting

Suppose one run is correct with probability $\geq \frac{1}{2} + \gamma$. Run k times independently and take the **majority** answer. By Hoeffding,

$$\Pr[\text{majority wrong}] \;\leq\; e^{-2\gamma^2 k}.$$

- To reach error $\leq \delta$: take $k \geq \dfrac{\ln(1/\delta)}{2\gamma^2}$ repetitions.
- *One-sided* error p (never a false "yes"): k runs give error $\leq (1-p)^k$.

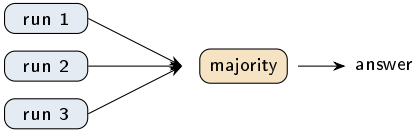

Figure: repeat and vote — error falls exponentially in k.

Worked Example — Boosting a $\frac{2}{3}$-Correct Algorithm

Setup

One run is correct with probability $\frac{2}{3}$, so $\gamma = \frac{2}{3} - \frac{1}{2} = \frac{1}{6}$. After k runs and a majority vote,

$$\Pr[\text{wrong}] \le e^{-2(1/6)^2 k} = e^{-k/18}.$$

How many runs for 1% error?

$$e^{-k/18} \le 0.01 \iff \frac{k}{18} \ge \ln 100$$

$$\Rightarrow k \ge 18 \ln 100 \approx 82.9 \Rightarrow \boxed{k = 83}.$$

Takeaway

A modest, *constant* number of repetitions crushes the error — amplification is cheap.

Quick Check — Round 1 (Analysis Tools)

[Q1]

State Markov's inequality and its one requirement on X.

[Q2]

A run is correct with probability $\frac{1}{2} + \gamma$. Bound the error of a k-run majority vote.

[Q3]

Using Chebyshev, how many samples estimate a probability to accuracy ε?

[Q4] True/False

"Chernoff bounds give an exponentially small failure probability, unlike Chebyshev."

Think for a minute — answers on the next slide.

Answers — Round 1

[A1]

$\Pr[X \geq a] \leq \mathbb{E}[X]/a$ for $a > 0$, requiring $X \geq 0$ (non-negative).

[A2]

$\Pr[\text{majority wrong}] \leq e^{-2\gamma^2 k}$ (Hoeffding).

[A3]

$n = O(1/\varepsilon^2)$ samples: $\mathrm{Var}(\bar{X}) = p(1-p)/n \leq 1/(4n)$, then $\Pr[|\bar{X} - p| \geq \varepsilon] \leq 1/(4n\varepsilon^2)$.

[A4]

True. Chernoff/Hoeffding decay like e^{-cn}; Chebyshev only like $1/a^2$.

The Discrete Logarithm Problem (DLP)

Definition

In a cyclic group $\langle a \rangle$ of *order* r, given $b \in \langle a \rangle$, find the unique $t \in \{0, 1, \ldots, r-1\}$ with

$$a^t = b, \qquad t = \log_a b.$$

Quantum promise

Shor's algorithm solves DLP in **polynomial** time. This section focuses on the subtlety when the order r is *composite*.

- Believed **hard** classically (basis of Diffie–Hellman, ElGamal, ECC).
- Prerequisite: the order $r = \mathrm{ord}(a)$, itself found by quantum *order-finding*.

Shor's Discrete-Log Algorithm — Setup

The idea (hidden-subgroup form)

Work over $\mathbb{Z}_r \times \mathbb{Z}_r$ with

$$f(x, y) = a^x b^y = a^{x+ty} \pmod{\cdot}.$$

f depends only on $x + ty \pmod{r}$: its "period" hides t.

Steps

1. Uniform superposition over $x, y \in \mathbb{Z}_r$.
2. Compute $|x\rangle\,|y\rangle\,|a^x b^y\rangle$.
3. QFT_r on both index registers.
4. Measure to obtain (l_1, l_2).

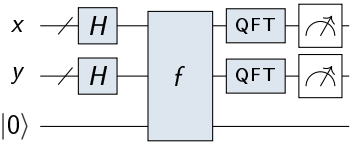

Figure: two index registers over $\mathbb{Z}_r$ plus a value register.

The Measured Congruence

What a run tells us

Fourier sampling returns a pair (l_1, l_2), with l_1 (nearly) *uniform* on $\mathbb{Z}_r$, obeying the linear congruence

$$\boxed{l_2 \equiv t\, l_1 \pmod{r}}.$$

Recovering t

If $\gcd(l_1, r) = 1$ then l_1 is invertible and

$$t \equiv l_2\, l_1^{-1} \pmod{r}.$$

The catch

This needs l_1 **invertible** mod r. Whether that is guaranteed depends on whether r is prime or composite.

Prime Order — The Easy Case

If r is prime

Every $l_1 \in \{1, \ldots, r-1\}$ is coprime to r, hence invertible. So *any* run with $l_1 \neq 0$ yields t immediately:

$$t \equiv l_2 \, l_1^{-1} \pmod{r}.$$

Why prime is nice

$\mathbb{Z}_r$ is a *field* when r is prime — every nonzero element has an inverse, so there is nothing to fix.

- $\Pr[l_1 = 0] = 1/r$ is tiny; otherwise one run suffices.
- Expected number of runs ≈ 1.

Composite Order — The Subtlety

If r is composite

Some l_1 share a factor with r: $d = \gcd(l_1, r) > 1$. Then l_1 is **not** invertible and the congruence $l_2 \equiv t\, l_1 \pmod{r}$ pins down t only *modulo* r/d:

$$t \equiv \left(\frac{l_2}{d}\right)\left(\frac{l_1}{d}\right)^{-1} \pmod{r/d}.$$

Two ways to finish

1. **Repeat** until $\gcd(l_1, r) = 1$: probability $\varphi(r)/r$ per run, so expected $O(\log \log r)$ runs.
2. **Combine** several partial congruences by the *Chinese Remainder Theorem*.

Still efficient

A constant/poly-logarithmic number of runs plus classical CRT recovers t fully — the algorithm stays **polynomial time**.

Worked Example — Composite Order $r = 6$

Group of order 6, unknown $t = 4$

Take $a = 3$ in $\mathbb{Z}_7^*$ (order 6), $b = a^t$. Runs give (l_1, l_2) with $l_2 \equiv 4\, l_1 \pmod 6$.

- $l_1 = 2$: $l_2 = 2$, $d = 2 \Rightarrow t \equiv 1 \pmod 3$.
- $l_1 = 3$: $l_2 = 0$, $d = 3 \Rightarrow t \equiv 0 \pmod 2$.

Chinese Remainder Theorem

$$t \equiv 1 \pmod 3, \quad t \equiv 0 \pmod 2$$
$$\Longrightarrow \quad t \equiv 4 \pmod 6.$$

Check: $4 \bmod 3 = 1$, $4 \bmod 2 = 0$. ✓

Moral

Neither run alone gives t; **CRT stitches the pieces** across the prime-power factors of r.

Quick Check — Round 2 (Discrete Logarithms)

[Q1]

State the discrete logarithm problem.

[Q2]

What linear congruence does each run of Shor's DL algorithm produce?

[Q3]

Why is a *prime* order r easy, and what goes wrong when r is *composite*?

[Q4]

From $t \equiv 2 \pmod 3$ and $t \equiv 1 \pmod 5$, find $t \pmod{15}$.

Answers — Round 2

[A1]

Given a of order r and $b \in \langle a \rangle$, find t with $a^t = b$.

[A2]

A pair (l_1, l_2) with $l_2 \equiv t\, l_1 \pmod{r}$ and l_1 (nearly) uniform on $\mathbb{Z}_r$.

[A3]

Prime r: every $l_1 \neq 0$ is invertible, so $t \equiv l_2 l_1^{-1}$ in one run. Composite r: if $\gcd(l_1, r) = d > 1$ then l_1 is not invertible and only $t \bmod (r/d)$ is learned — combine runs by CRT.

[A4]

$t \equiv 2 \pmod{3}$, $t \equiv 1 \pmod{5} \Rightarrow t \equiv \mathbf{11} \pmod{15}$ ($11 \bmod 3 = 2$, $11 \bmod 5 = 1$).

Recap — The Schmidt Decomposition (Unit 2)

Statement

Any pure bipartite state has the form

$$|\psi\rangle = \sum_k \lambda_k \, |u_k\rangle_A \otimes |v_k\rangle_B , \qquad \lambda_k \geq 0, \ \sum_k \lambda_k^2 = 1,$$

with orthonormal $\{|u_k\rangle\}$, $\{|v_k\rangle\}$. The number of nonzero λ_k is the **Schmidt rank**.

Unit 2 proved it exists — here we *compute* it. The engine is the singular value decomposition (SVD).

The Coefficient Matrix

From state to matrix

Expand in product bases:

$$|\psi\rangle = \sum_{i,j} c_{ij}\, |i\rangle_A |j\rangle_B .$$

Collect the amplitudes into the matrix $C = (c_{ij})$ (rows $= A$, columns $= B$).

Reduced states

$$\rho_A = \mathrm{Tr}_B\, |\psi\rangle\langle\psi| = CC^\dagger, \quad \rho_B = (C^\dagger C)^\top .$$

Their eigenvalues are the λ_k^2.

- Normalisation $\sum_{ij} |c_{ij}|^2 = 1$ means $\|C\|_F = 1$.
- Schmidt rank $=$ **rank**(C).

Method 1 — via the Singular Value Decomposition

Algorithm

1. Form $C = (c_{ij})$.
2. Compute the SVD $C = U\Sigma V^\dagger$, with U, V unitary and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \dots)$, $\sigma_k \geq 0$.
3. Then $|\psi\rangle = \sum_k \sigma_k \, |u_k\rangle_A \otimes |v_k^*\rangle_B$, where $|u_k\rangle = k$-th column of U and $|v_k^*\rangle =$ conjugated k-th column of V.

Read-off

Schmidt coefficients $\lambda_k = \sigma_k$ (the singular values); Schmidt rank = number of nonzero σ_k. Automatically $\sum_k \sigma_k^2 = \|C\|_F^2 = 1$.

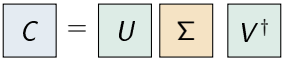

Figure: the SVD directly gives the Schmidt form.

Method 2 — via the Reduced Density Matrix

Algorithm

1. Compute $\rho_A = \mathrm{Tr}_B\, |\psi\rangle\langle\psi| = CC^\dagger$.
2. Diagonalise $\rho_A = \sum_k \lambda_k\, |u_k\rangle\langle u_k|$ (spectral theorem).
3. Set $\sigma_k = \sqrt{\lambda_k}$ and, for $\sigma_k > 0$,

$$|v_k\rangle_B = \tfrac{1}{\sigma_k}\big(\langle u_k|_A \otimes I_B\big)\,|\psi\rangle\,.$$

4. Assemble $|\psi\rangle = \sum_k \sigma_k\, |u_k\rangle\, |v_k\rangle$.

Same answer

Singular values of $C = \sqrt{\text{eigenvalues of } CC^\dagger}$, so both methods give identical Schmidt coefficients.

When to prefer it

Handy when ρ_A is already known (e.g. from an experiment) rather than the full state vector.

Worked Example — Schmidt via SVD

A "disguised" product state

$$|\psi\rangle = \tfrac{1}{\sqrt{2}}\big(\,|00\rangle + |01\rangle\,\big), \quad C = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}.$$

C has rank 1: one singular value $\sigma_1 = \|C\|_F = 1$, with $|u_1\rangle = |0\rangle$, $|v_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$.

$$|\psi\rangle = 1 \cdot |0\rangle \otimes |+\rangle \;\Rightarrow\; \text{rank 1 (product).}$$

A maximally entangled state

$$|\Phi^+\rangle = \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad C = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Singular values $\sigma_1 = \sigma_2 = \frac{1}{\sqrt{2}} \Rightarrow$ Schmidt rank 2 (entangled), already in Schmidt form.

Point

The SVD reveals the *natural* bases (here $\{|+\rangle, |-\rangle\}$ on B) and the rank — no guessing required.

Complexity and Remarks

Cost

The SVD of a $d_A \times d_B$ matrix costs $O(d_A d_B \min(d_A, d_B))$ — polynomial in the *matrix dimensions*.

Honest caveat

Those dimensions are $d_A = 2^{n_A}$, $d_B = 2^{n_B}$: **exponential** in qubit counts. Computing a Schmidt decomposition classically is efficient only when one side is *small*.

Why it matters

- Detects/quantifies entanglement (rank, entropy $-\sum \lambda_k^2 \log \lambda_k^2$).
- Underpins purification, tensor-network methods, and channel analysis.

Quick Check — Round 3 (Schmidt Decompositions)

[Q1]

Given $|\psi\rangle = \sum_{ij} c_{ij} |i\rangle |j\rangle$, how do you compute its Schmidt decomposition?

[Q2]

How are the Schmidt coefficients related to the singular values of C and to ρ_A?

[Q3]

What does Schmidt rank 1 tell you about the state?

[Q4] MCQ

The Schmidt coefficients equal:
(a) eigenvalues of C (b) singular values of C (c) diagonal of C (d) traces of C

Answers — Round 3

[A1]

Form $C = (c_{ij})$, take its SVD $C = U\Sigma V^\dagger$; then $|\psi\rangle = \sum_k \sigma_k |u_k\rangle |v_k^*\rangle$ with $|u_k\rangle, |v_k\rangle$ the singular vectors.

[A2]

$\lambda_k = \sigma_k$ (singular values of C), and λ_k^2 are the eigenvalues of $\rho_A = CC^\dagger$.

[A3]

The state is a **product** (separable): $|\psi\rangle = |u_1\rangle \otimes |v_1\rangle$, no entanglement.

[A4]

(b) singular values of C.

Summary of Unit 5

What we covered

1. **Markov** $\Pr[X \geq a] \leq \mathbb{E}[X]/a$ and **Chebyshev** $\Pr[|X - \mu| \geq a] \leq \sigma^2/a^2$.
2. **Chernoff/Hoeffding** — exponentially small tails; the **union bound**.
3. **Amplification** — majority voting gives error $\leq e^{-2\gamma^2 k}$.
4. **Discrete log** — each run yields $l_2 \equiv t\, l_1 \pmod{r}$.
5. **Prime** r — recover $t \equiv l_2 l_1^{-1}$ in one run.
6. **Composite** r — non-invertible l_1 gives $t \bmod (r/d)$; finish by CRT.
7. **Schmidt via SVD** — $C = U\Sigma V^\dagger$; coefficients = singular values; rank = $\mathrm{rank}(C)$.
8. **Reduced density matrix** route and the (exponential-dimension) cost caveat.

Probability bounds tame randomness; number theory and the SVD do the rest.

Practice Questions (Exam Oriented)

Short answer (2–5 marks)

1. State and compare Markov, Chebyshev and Chernoff bounds.
2. Derive the number of repetitions to boost success $\frac{1}{2} + \gamma$ to error δ.
3. State the congruence produced by Shor's discrete-log algorithm and how t is recovered.
4. Explain why composite order needs the CRT; illustrate with $r = 6$.
5. Give the SVD recipe for a Schmidt decomposition and state what the rank means.

Long answer (8–10 marks)

6. Use Chebyshev to bound the samples needed to estimate a probability to accuracy ε.
7. Work through the discrete-log recovery for a composite r of your choice using CRT.
8. Compute the Schmidt decomposition of a given two-qubit state by SVD and by ρ_A; compare.
9. Show $\rho_A = CC^\dagger$ and that its eigenvalues are the squared Schmidt coefficients.
10. Discuss why classical Schmidt computation is efficient only for small subsystems.

Think & Explore (Take-Home)

Mini task (by hand)

- Boost a 0.6-correct algorithm to error ≤ 0.001; find k.
- For $r = 12$, list which l_1 are invertible and finish a composite-order recovery by CRT.
- Schmidt-decompose $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle)$ via SVD.

Coding (self-learning)

In `NumPy`: `svd` for Schmidt coefficients; simulate majority-vote boosting and compare with the Hoeffding bound.

Reading & reflection

- Kaye–Laflamme–Mosca, Appendices A.1, A.2, A.7.
- Mitzenmacher & Upfal, *Probability and Computing*.
- Prepare for **Unit 6**: applications and error correction.

Reminder

Assignment 3 covers Unit 5 — handwritten, step-by-step, submitted as a single PDF via the course page.

References

Textbooks (as per the course page)

1. P. Kaye, R. Laflamme & M. Mosca, *An Introduction to Quantum Computing*, Oxford Univ. Press (Appendices A.1, A.2, A.7). **[Main text]**
2. M. Mitzenmacher & E. Upfal, *Probability and Computing*, Cambridge.
3. R. Motwani & P. Raghavan, *Randomized Algorithms*, Cambridge.

Seminal papers

4. P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms...," *SIAM J. Comput.*, 1997.
5. M. A. Nielsen & I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge (SVD & Schmidt).

NPTEL: *Quantum Computing* · Course page: https://www.gcjana.in/courses/shardauniversity/2501/quantum-computing/

Thank You

Any Questions?

"Probability theory is nothing but common sense reduced to calculation."

— Pierre-Simon Laplace

Post your doubts on the course page →

https://www.gcjana.in/courses/shardauniversity/2501/quantum-computing/#Post-doubts