1. Deutsch–Jozsa Algorithm

Goal:

Determine whether a function $f(x): \{0,1\}^n \to \{0,1\}$is **constant** or **balanced** with a single query.

Idea:

Classically, $2^{n-1}+1$evaluations are needed.

Quantumly, only **one** query.

Math Concept:

- Prepare uniform superposition of all inputs using Hadamard gates.
- Apply the **oracle** $U_f \mid x, y\rangle = \mid x, y \oplus f(x)\rangle$.
- Apply another layer of Hadamards to interfere all paths.

If $f$is constant → output all zeros

If $f$is balanced → output non-zero pattern.

2. Bernstein–Vazirani Algorithm

Goal:

Find a secret bit string $s \in \{0,1\}^n$in the function

$$f(x) = s \cdot x \pmod 2$$

Idea:

The oracle encodes the hidden string $s$via inner product.

Quantum parallelism + interference reveals $s$in one query.

Math Example:

For $s = 101$,

$$f(011) = 1 * 0 \oplus 0 * 1 \oplus 1 * 1 = 1$$

Circuit:

- Apply **Hadamard** to all qubits.
- Query the **oracle** U_f.
- Apply **Hadamard** again.
- Measure → output gives s.

3. Grover's Search Algorithm

Goal:

Search an unsorted database of $N = 2^n$elements in $O(\sqrt{N})$

steps.

Idea:

- Initialize all states equally.
- Use **oracle** to mark the desired solution by flipping its phase.
- Use **diffusion operator** to amplify its probability.

Math Concept:

$$\mid \psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \mid x\rangle$$

Each iteration rotates the state vector toward the solution by an angle 2θ.

4. Quantum Teleportation

Goal:

Send an *unknown quantum state* $\mid \psi\rangle$from Alice to Bob using **entanglement** and **2 classical bits**.

Steps:

1. Create EPR pair between Alice and Bob.
2. Alice performs Bell measurement.
3. Alice sends 2 classical bits.
4. Bob applies X/Z corrections.
5. **Math:**

$$|\psi\rangle_A \otimes |\Phi^+\rangle_{BC} \xrightarrow{\text{Bell Measure}} |ij\rangle_{AB} \otimes \sigma_{ij} |\psi\rangle_C$$

Algorithm	**Problem Solved**	**Key Quantum Feature**	**Classical Cost**	**Quantum Cost**
Deutsch–Jozsa	Constant vs balanced	Interference	$O(2^n)$	O(1)
Bernstein–Vazirani	Hidden bit string (s)	Phase encoding	O(n)	O(1)
Grover's Search	Search unstructured database	Amplitude amplification	O(N)	$O(\sqrt{N})$
Teleportation	State transfer via entanglement	Entanglement + measurement	Not possible	2 classical bits

5. Quantum Fourier Transform (QFT)

Goal:

Compute the discrete Fourier transform of amplitudes in a quantum state.

Use:

Core step in Shor's factoring algorithm and phase estimation.

Definition:

$$|x\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x k/N} |k\rangle$$

Circuit:

Series of Hadamards and controlled-phase rotations, followed by bit-reversal.

Complexity:

Classical DFT $\rightarrow O(N^2)$

Quantum QFT $\rightarrow O(n^2)$

6. Quantum Phase Estimation (QPE)

Goal:

Estimate the eigenphase ϕ of a unitary U where $U|\psi\rangle = e^{2\pi i\phi}|\psi\rangle$.

Use:

- Foundation for algorithms like Shor's factoring, HHL, and quantum chemistry.
- Finds eigenvalues of matrices or Hamiltonians.

Steps:

1. Apply Hadamards to counting qubits.
2. Apply controlled-U^{2^k}.
3. Apply inverse QFT.
4. Measure to get binary digits of phase.

7. Shor's Factoring Algorithm

Goal:

Factor a composite number N efficiently.

Idea:

Reduces factoring to finding the **period** of a modular exponential function $f(x) = a^x \bmod N$.
Uses **QFT** to find that period.

Complexity:

Classical: Exponential
Quantum: Polynomial $O((\log N)^3)$

8. Quantum Approximate Optimization Algorithm (QAOA)
Goal:
Approximate solutions to combinatorial optimization problems (e.g., MaxCut, SAT).
Idea:
Alternate between problem Hamiltonian H_C and mixing Hamiltonian H_B using tunable parameters (γ, β).
Output:
A quantum state whose measurement gives near-optimal solutions.

9. Variational Quantum Eigensolver (VQE)
Goal:
Find the ground-state energy of a molecule or Hamiltonian.
Idea:
Use a **parameterized quantum circuit** + **classical optimizer** to minimize

$$E(\theta) = \langle \psi(\theta) \mid H \mid \psi(\theta) \rangle$$

Use:
Quantum chemistry, materials simulation.

10. Harrow–Hassidim–Lloyd (HHL) Algorithm
Goal:
Solve linear equations $A\vec{x} = \vec{b}$
Steps:

1. Encode $\vec{b}$
2. Perform **phase estimation** on A.
3. Apply rotation proportional to $1/\lambda_i$.
4. Uncompute to get $\mid x\rangle$.

Complexity:

Classical: $O(N^3)$
Quantum: $O(\log N)$ under certain conditions.

11. Quantum Counting
Goal:
Estimate number of solutions to a function $f(x)$ using **Grover's** oracle.
Idea:
Combines Grover's iterations with **phase estimation** to infer count of marked states.

12. Simon's Algorithm

Goal:
Find the secret string $s$in a function $f(x)$ where

$$f(x) = f(x \oplus s)$$

for all x.
Use:
Demonstrates **exponential speedup** over classical algorithms.
Complexity:
Classical: $O(2^{n/2})$
Quantum: $O(n)$

Variational Quantum Eigensolver (VQE)

You want to find the **smallest eigenvalue** of a Hamiltonian H:

$$E_0 = \min_{|\psi(\theta)\rangle} \langle \psi(\theta) \mid H \mid \psi(\theta) \rangle$$

where

- $|\psi(\theta)\rangle$= a *parameterized quantum state* (ansatz)
- H= Hamiltonian (energy operator)
- θ= set of classical parameters that control the quantum circuit

---

Working Steps

Step 1 – Define Hamiltonian

The Hamiltonian represents total energy:

$$H = \sum_i \quad c_i P_i$$

where each $P_i$is a **Pauli string** (e.g., Z_0Z_1, X_0X_1, I_0Z_1) and $c_i$are coefficients from chemistry calculations.
Example (H_2 molecule):

$$H = -1.05Z_0 - 0.39Z_1 + 0.18Z_0Z_1 + 0.39I$$

Step 2 – Choose an Ansatz

A parameterized quantum circuit creates the trial state:

$$|\psi(\theta)\rangle = U(\theta) \mid 0\rangle$$

Example circuit:

- Apply rotation gates $R_y(\theta_i)$
- Add entangling gates (CNOT)

Step 3 – Measure Energy

Run the quantum circuit to get:

$$E(\theta) = \langle \psi(\theta) \mid H \mid \psi(\theta) \rangle$$

This is done by measuring expectation values of each P_i.

Step 4 – Classical Optimization

A classical optimizer (e.g., COBYLA, SPSA, or Adam) updates $\theta$to minimize $E(\theta)$.

$$\theta_{new} = \theta_{old} - \eta \nabla_\theta E(\theta)$$

Repeat until energy converges.

Input:

1. **Hamiltonian (H):** Represents the system you want to study — e.g., a molecule or physical system.
   Example:
   $$H = a_0 I + a_1 Z_0 + a_2 Z_1 + a_3 Z_0 Z_1 + a_4 X_0 X_1 + a_5 Y_0 Y_1$$
2. **Parameterized quantum circuit (ansatz):** Describes the trial wavefunction $|\psi(\theta)\rangle$using parameters θ.
   Example: rotation gates $R_y(\theta_1), R_y(\theta_2)$, and entangling gate CX.
3. **Classical optimizer:** Minimizes the expectation value of $H$by updating parameters θ.

Process:

1. Prepare state $|\psi(\theta)\rangle$on the quantum circuit.
2. Measure expectation value $E(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle$.
3. Use a classical optimizer (like COBYLA or SPSA) to update θ.
4. Repeat until $E(\theta)$converges to minimum (ground state energy).

Output:

- **Ground state energy** E_{min}— the lowest eigenvalue of H.
- **Optimal parameters** θ^*— define the quantum state closest to the ground state $|\psi(\theta^*)\rangle$.

Mathematical Flow

1. Prepare circuit: $|\psi(\theta)\rangle$
2. Measure each term of H
3. Compute $E(\theta)$
4. Update θ
5. Repeat → minimum energy = **ground state**

Applications

- Quantum chemistry (molecular energies)
- Material science
- Optimization problems
- Nuclear structure simulation

Step-by-step VQE with the math you need and the circuit for each step.

What VQE does (one line)

Finds the ground-state energy E_0 of a Hamiltonian H by minimizing $E(\theta) = \langle\psi(\theta) \mid H \mid \psi(\theta)\rangle$ with a parameterized quantum circuit and a classical optimizer.

Inputs and output (quick)

Input: Hamiltonian $H = \sum_j \quad c_j P_j$ (Pauli strings), ansatz circuit $U(\theta)$, optimizer.
Output: approximate ground energy $E_{\min}$ and optimal parameters θ^* (and the state $\mid \psi(\theta^*)\rangle$).

Step 0 — Notation & preliminaries

- n qubits, basis states $\mid x\rangle$.
- Hamiltonian in qubit form:

$$H = \sum_j \quad c_j P_j, P_j \in \{I, X, Y, Z\}^{\otimes n}.$$

- Ansatz (trial state): $\mid \psi(\theta)\rangle = U(\theta) \mid 0\rangle^{\otimes n}$.
- Objective: minimize $E(\theta) = \langle\psi(\theta) \mid H \mid \psi(\theta)\rangle$.

Step 1 — Build Hamiltonian (math)

1. From chemistry or model get integrals. Map to qubits via Jordan–Wigner / Parity / Bravyi–Kitaev.
2. Express as Pauli sum:

$$H = \sum_j \quad c_j P_j.$$

3. Example 2-qubit:

$$H = a\, Z_0 + b\, Z_1 + c\, Z_0 Z_1.$$

Step 2 — Choose an ansatz $U(\theta)$

Requirements: expressive enough to approximate ground state, low depth for NISQ.
Common choices:

- Hardware-efficient: layers of single-qubit rotations (e.g. $R_y(\theta)$) + entanglers (CNOT/CZ).
- Problem-inspired (UCCSD for chemistry).
  Circuit form example (2 qubits):

Step 3 — Evaluate $E(\theta)$ on the quantum device (math + measurement)

- Because $H$is sum of Pauli strings,
- $E(\theta) = \sum_j \quad c_j \langle \psi(\theta) \mid P_j \mid \psi(\theta) \rangle$.

- Each term $\langle P_j \rangle$is estimated by preparing $\mid \psi(\theta) \rangle$, measuring in the basis of $P_j$repeatedly (shots), and taking the average. For a Pauli string $P_j$the measurement outcomes $o \in \{\pm 1\}$give
- $\widehat{\langle P_j \rangle} = \frac{1}{S} \sum_{s=1}^{S} \quad o_s$.
  Then
- $\widehat{E(\theta)} = \sum_j \quad c_j \, \widehat{\langle P_j \rangle}$.

Step 4 — Classical optimizer updates θ

- Choose optimizer type:
- Gradient-free: COBYLA, Nelder-Mead, CMA-ES (robust to noise).
- Stochastic gradient or analytic gradient (faster with many parameters): use parameter-shift rule.
- Parameter-shift gradient (for a parameter $\theta_k$generated by a single-qubit rotation $e^{-i\theta_k G}$with eigenvalues $\pm\frac{1}{2}$):
- $\frac{\partial E}{\partial \theta_k} = \frac{1}{2}\left(E\left(\theta_k + \frac{\pi}{2}\right) - E\left(\theta_k - \frac{\pi}{2}\right)\right)$.

- Compute two evaluations per gradient component. Use this inside gradient-based optimizers (Adam, L-BFGS-B, SPSA).

Step 5 — Output & postprocessing

- Return $\theta^*$and $E_{\min} = E(\theta^*)$. Optionally:
- Reconstruct final state and compute other observables.
- Estimate error bars from repeated measurements and optimizer noise.
- **Small worked mathematical example (2-qubit Hamiltonian)**
- Hamiltonian:
- $H = aZ_0 + bZ_1 + cZ_0Z_1$.

- Ansatz: $U(\theta) = R_y(\theta_0) \otimes R_y(\theta_1)$followed by CNOT(0→1). Statevector $\mid \psi(\theta) \rangle$. Compute expectation:
- $E(\theta) = a\langle Z_0 \rangle + b\langle Z_1 \rangle + c\langle Z_0Z_1 \rangle$,

- where each expectation is computed from $\mid \psi(\theta) \rangle$analytically (if small) or numerically. In code you compute $\psi$and then $E = \psi^\dagger H \psi$.

Gradients & optimization details

Parameter-shift works for gates of form $e^{-i\theta\sigma/2}$(RY, RX). Use analytic gradients, less noisy than finite differences.

SPSA is efficient when many parameters and noisy cost evaluations (uses 2 measurements per iteration).

Use adaptive shot allocation: more shots near minimum.

Resource / complexity notes

Each energy evaluation needs as many circuit executions as there are measurement groups × shots.

Optimization iterations × evaluations per iteration (and gradient evaluations) determine total runtime.

Depth constraints: choose ansatz depth compatible with device coherence.

💡

Hybrid Advantage

VQE uses:

- **Quantum computer** → generates complex quantum states
- **Classical computer** → optimizes parameters efficiently

This makes it **robust to noise** on current (NISQ) quantum hardware.