Adiabatic quantum computation and algorithmic equivalences

Follow an easy ground state along a changing Hamiltonian and learn why the gap controls the difficulty.

Research 21 min lesson Dr. Charith Jayasekara

By the end, you'll be able to

- Calculate the gap in a one-qubit interpolation.
- Separate model equivalence from practical optimisation speed.
- Identify schedule, preparation and readout assumptions.

Begin with a landscape, not a gate list

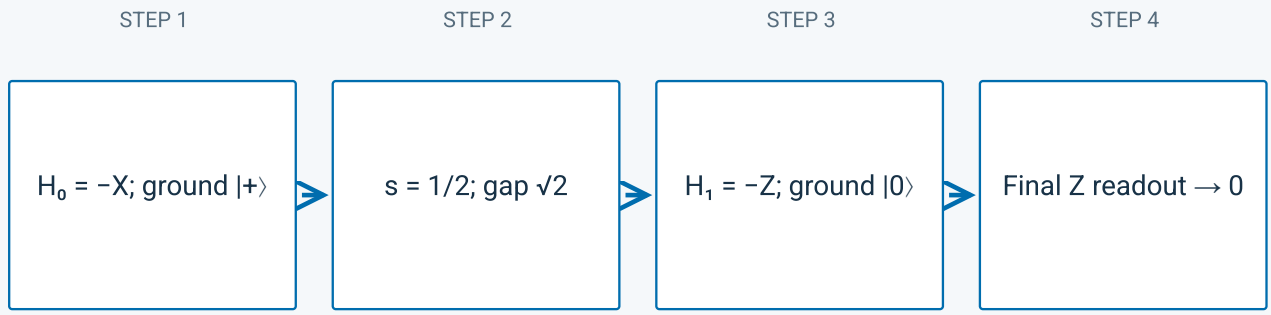
Imagine a ball resting at the lowest point of a bowl. If the bowl changes shape gently enough, the ball can follow its lowest point. Adiabatic quantum computation uses a related mathematical idea: prepare a ground state of an easy Hamiltonian, then change the Hamiltonian toward one whose ground state encodes the desired answer. The analogy explains the intention, but a quantum state is not a classical ball, and slow motion does not automatically guarantee efficient computation.

A Hamiltonian is a Hermitian matrix describing energy and generating time evolution. A ground state is an eigenvector with the smallest eigenvalue. The gap is the difference between the ground energy and the next relevant energy level. A narrow gap can require a very long evolution. The crucial question is therefore not just whether the final energy landscape contains an answer, but whether the chosen path lets us reach it with acceptable resources.

Before continuing, review matrices and vectors, phase and interference and algorithmic complexity. The first pass needs only the idea of eigenvectors: special directions that a matrix scales without mixing into other directions.

WORKED DIAGRAM · AA01

An easy ground-state path



For H(s) = −(1−s)X − sZ, the gap is 2√((1−s)²+s²). It is 2 at the endpoints and √2 at the midpoint, so it never closes.

State the computational setup

The input is a description of an initial Hamiltonian, a final problem Hamiltonian and a path between them. We require an efficiently preparable initial ground state and a way to implement the path's interactions. The output may be a sampled candidate solution or a final quantum state. A correctness claim needs a target success probability and conditions on the path and evolution time.

A common interpolation is

$$H(s) = (1 - s)H_0 + sH_1, \qquad 0 \leq s \leq 1.$$

The parameter s changes with physical time. Choosing a schedule means deciding how quickly to move through different parts of the path. The adiabatic theorem has several forms; runtime conditions depend on gaps, derivatives, boundary behaviour and error tolerance. "Time equals inverse gap squared" is a useful warning about narrow gaps in simple bounds, not a universal formula for every schedule and theorem.

Work a one-qubit example

Take the initial Hamiltonian to be negative X and the final Hamiltonian to be negative Z. The initial ground state is plus, while the final ground state is zero. Along the linear path, the two energies are the positive and negative square root of the sum of the squares of s and one minus s. The gap is twice that square root.

At the start, the gap is two. At the midpoint, both coefficients are one half, and the gap is square root of two. At the end, it returns to two. The gap never closes. In this small example, the ground-state Bloch direction moves from positive x to positive z. A sufficiently slow ideal evolution approximately follows that direction, so a final Z measurement returns zero with high probability.

This is an instructional state-preparation example, not an optimisation advantage. A single H gate can also turn plus into zero. Its value is that every energy and gap can be calculated explicitly, making the meaning of a path and schedule concrete before many-qubit examples hide those quantities.

The interpolation and its exact gap explain what the adiabatic schedule must follow.

$$H(s) = -(1-s)X - sZ, \qquad \Delta(s) = 2\sqrt{(1-s)^2 + s^2}, \qquad \min_{0 \le s \le 1} \Delta(s) = \sqrt{2}.$$

s is the path parameter and Δ is the energy difference between excited and ground states. The minimum occurs at $s = 1/2$. A real schedule also specifies how s changes with physical time.

Follow the procedure

First choose the final objective and convert it into a problem Hamiltonian. Next choose an easy initial Hamiltonian with a known ground state. Specify the interpolation and schedule, estimate or bound its relevant gap, and choose an evolution time consistent with the desired error. Prepare the initial state, evolve, measure in the required basis and verify any classical candidate. Repeat according to a declared success criterion.

The gap analysis is often the difficult part. Writing down a cost function as a diagonal Hamiltonian does not prove that its ground state can be reached quickly. An exponentially small gap may erase the hoped-for advantage. A proposed schedule also has implementation limits: the device must realise the required interactions and maintain adequate control during the entire evolution.

What model equivalence means

Aharonov and colleagues established polynomial equivalence between an appropriate adiabatic model and the conventional circuit model. A computation in one can be simulated by the other with polynomial overhead. This is a statement about computational power under specified mathematical models. It does not say every short adiabatic path becomes a practical short circuit, or that a restricted annealing device implements every universal adiabatic construction. Read the equivalence paper.

Equivalence also does not make arbitrary optimisation problems easy. Universal gate computation does not solve every hard search problem efficiently, and translating a circuit into a Hamiltonian construction does not change that limitation. The theorem connects models; it does not remove their complexity constraints.

Practice with answers

Exercise 1. At s equal to one half in the one-qubit example, calculate the two energies and their difference.

> Check the worked answer
>
> The coefficient length is the square root of one quarter plus one quarter, or one over square root of two. The energies are its positive and negative values, so their difference is square root of two. The gap remains nonzero.

Exercise 2. A report encodes a difficult scheduling objective in a final Hamiltonian and concludes that slow evolution solves it efficiently. What is missing?

Check the worked answer

It needs an implementable path, gap and schedule analysis, preparation and readout costs, error tolerance and total runtime. A final ground state encoding the optimum establishes an objective, not an efficient route to it. Compare against classical optimisation at the same solution quality and account for repetitions and verification.

The central habit is to distinguish expressing an answer as a ground state from preparing that state efficiently. The same distinction recurs in annealing, Hamiltonian simulation and variational energy minimisation.

Check your understanding

PAUSE & THINK

What is the midpoint gap in the worked interpolation?

- [ ] 0
- [ ] √2
- [ ] 2

Answer: √2. The two energies are ±1/√2, whose difference is √2. A nonzero gap here does not prove every optimisation path is efficient.

Sources

- Read the equivalence paper