CSCI3350 Introduction to Quantum Computing (2026 Spring)

Introduction: Physics and Computing

Instructor: Xiao Liang

Department of Computer Science and Engineering
The Chinese University of Hong Kong

(Nothing from this lecture will be tested on quizzes or exams!)

What Is Quantum Computing? — From Newton to Shrödinger

Core Idea: Computing Follows Physics

- Our computational power is bounded—and enabled—by our understanding of physics.
- Different physical laws suggest different ways to represent and process information.
- We will travel from classical mechanics (Newton) to electromagnetism (Maxwell) to quantum mechanics (Schrödinger).

Era I: Classical Mechanics and Mechanical Computing

Governing law for classical mechanics: Newton's second law

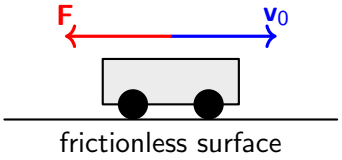
$$\mathbf{F} = m \cdot \mathbf{a} = \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t}$$

- $\mathbf{F}$: Force vector. The net force acting on an object (magnitude and direction).
- m: Mass (scalar). A measure of an object's inertia; how much it resists changes in motion.
- $\mathbf{a}$: Acceleration vector. The time rate of change of velocity: $\mathbf{a} = \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}$.
- $\mathbf{p}$: Momentum vector. Defined as $\mathbf{p} = m\mathbf{v}$ for a particle with constant mass m and velocity $\mathbf{v}$.
- $\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t}$: Time derivative of momentum. How momentum changes with time; equals the net force.

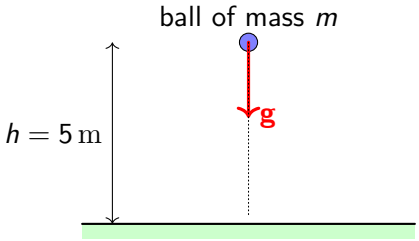
Governing law for classical mechanics: Newton's second law

$$\mathbf{F} = m \cdot \mathbf{a} = \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t}$$

Exemplary systems we can solve:



(a) Cart on a frictionless surface



(b) Ball dropped from height

If we can build gears, levers, and pulleys, we can compute with them.

Mechanical Computing Devices: Abacus

Abacus:

- rods with beads represent digits;
- compute by moving beads according to rules.

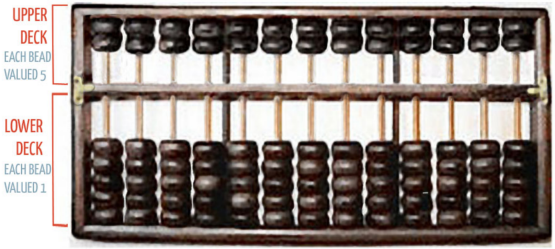
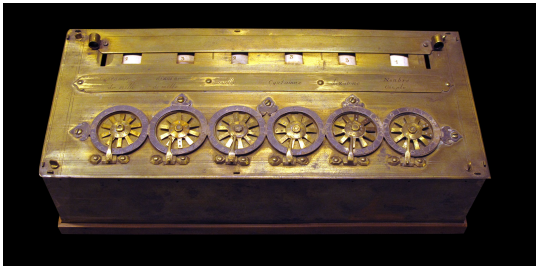


Figure: Abacus

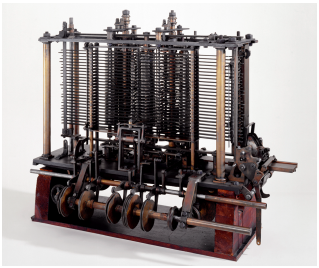
Mechanical Computing Devices: Gear calculators

Gear calculators (e.g., Pascaline, Babbage's analytical engine):

- gear positions encode numbers;
- addition/subtraction via gear rotations and carries.



(a) Pascaline



(b) Babbage's analytical engine

Era II: Electromagnetism and Electronic Computing

Governing laws: Maxwell's equations (unifying electricity and magnetism):

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0} \qquad \text{(Gauss's law)}$$

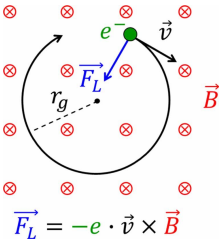
$$\nabla \cdot \mathbf{B} = 0 \qquad \text{(Gauss's law for magnetism)}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \qquad \text{(Faraday's law)}$$

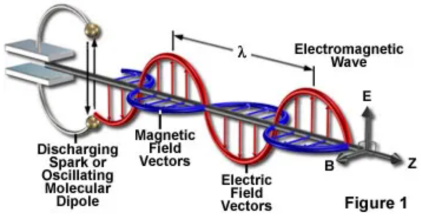
$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} \qquad \text{(Ampère–Maxwell law)}$$

Where $\mathbf{E}$ is the electric field, $\mathbf{B}$ is the magnetic field, ρ is the electric charge density, $\mathbf{J}$ is the current density, ε_0 is the vacuum permittivity, and μ_0 is the vacuum permeability.

Exemplary systems we can solve using Maxwell's equations:



(a) An electron in a magnetic field



(b) Propagation of an electromagnetic wave

If we can harness fields and waves (wires, antennas, circuits), we can compute with electrons and light.

Electronic Computing Devices

State-of-the-art devices: calculators, desktops, smartphones, cloud servers.

Encoding:

- Bits as high/low voltage levels.
- Wires carry bits; memory stores them as charge states or magnetic domains.

Computing:

- Logic gates (AND, OR, NOT, NAND, etc.) built from transistors implement Boolean functions.
- Synchronous circuits orchestrated by clocks compose gates into processors.
- Algorithms = sequences of logic operations on registers of bits.

Mechanical Computing vs Electronic Computing

Mechanical calculators are great for a few coupled motions (gears, cams, analog integrators). They struggle when a task needs:

- exploring an enormous number of possibilities,
- processing very large datasets with high precision,
- running millions to billions of steps quickly,
- flexible memory and branching logic.

Electronics handle these via fast arithmetic, memory, and massive parallelism. E.g.,

- High-Dimensional PDEs and Large-Scale Simulation
- Large-Scale Linear Algebra and Optimization
- Symbolic Computation and Automated Reasoning
- Data-Intensive Analytics

Era III: Quantum Mechanics and Quantum Computing

A New Revolution: **Quantum Mechanics**

- At microscopic scales, classical laws fail to explain experiments.
- Quantum mechanics provides new rules for how systems evolve and how measurements behave.

Concrete examples:

- Atoms and molecules (about 0.1–1 nanometer): Explains why elements have colors and why chemicals bond.
- Tunneling through thin barriers (a few nanometers): Used in flash memory and scanning tunneling microscopes.
- Quantum dots (1–10 nanometers): Tiny crystals that glow different colors in modern TV screens.

Shrödinger's Equation

Quantum Mechanics is ruled by Shrödinger's equation

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = \hat{H} |\psi(t)\rangle$$

where

- i: the imaginary root $i := \sqrt{-1}$; (we'll discuss more on why complex numbers are necessary later)
- $\hbar$: Planck's reduced constant[1]. You just need to know that it is some physics constant of the magnitude $\approx 1.055 \times 10^{-34} J \cdot s$.
- $|\psi(t)\rangle$: It's a vector (more discussion later)
- $\hat{H}$: It's a matrix (more discussion later)

[1] This is named after the renowned physicist Max Planck, recipient of the Nobel Prize in Physics in 1918.

An Inaccurate-yet-Helpful Analogy (1/3)

We don't need a fully rigorous understanding of this equation—this isn't a quantum mechanics course. But it helps to have an intuitive feel for it. We'll build that intuition by drawing an analogy with classical mechanics.

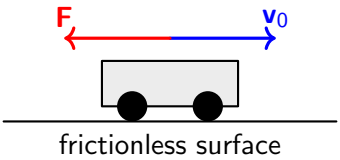


Figure: Cart on a frictionless surface

- You have a cart, an initial speed v_0, and a force F acting.
- If you know the setup, you can predict **where the cart will be** at time t.
- In short: knowing the rule $\mathbf{F} = m \cdot \mathbf{a}$ and the starting point lets us predict motion.

An Inaccurate-yet-Helpful Analogy (2/3)

We can use Shrödinger's equation (instead of Newton's) to solve the system as well:

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = \hat{H} |\psi(t)\rangle$$

- $|\psi(t)\rangle$: think of this as a **vector** that describes the system's overall state at time t.
- $\hat{H}$ (Hamiltonian): think of this as a **big rule-book or matrix** encoding how the system behaves (energies, forces, boundaries).
- The equation says: **change in the system's state over time** is determined by **the rule-book acting on the state**.

An Inaccurate-yet-Helpful Analogy (3/3)

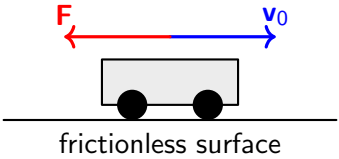


Figure: Cart on a frictionless surface

In more details:

- $|\psi(t)\rangle$: Think it as "where/what the system is like right now." It's like a vector you update over time.
- $\hat{H}$: Think it as "all the ingredients of the system"—mass, potential energy, constraints, fields, etc. It acts like a matrix that "pushes" the system's state forward in time.
- $i\hbar \dfrac{\partial}{\partial t}$: It reads as "how fast the state changes with time," scaled by a constant i times $\hbar$.

All You Need to Know for Shrödinger's Equation I

The following is all you need to know about Shrödinger's equation:

- Pretend the quantum state is like a vector that tells us where the system could be. The Hamiltonian is a big matrix that encodes the rules of the world. The Schrödinger equation says the matrix pushes the vector forward in time. If we know the rule and where we started, we can predict what we'll see later.

The previous analogy is not accurate. For example,

- The "vector" $|\psi\rangle$ is not a single, deterministic status of the system; it actually encodes **probabilities of many possible status**.
- $\hat{H}$ is **total energy operator** (kinetic + potential).

All You Need to Know for Shrödinger's Equation II

Indeed, there is an alternative formulation of classical mechanics, equivalent to Newton's, called **Hamiltonian mechanics**. The Schrödinger equation often feels more natural to those familiar with this formalism. If you are interested, you can easily find online resources on the topic, e.g., David Tong's course.

But again, our CSCI3350 is not a quantum mechanics course. We don't need a fully rigorous interpretation of the equation. Instead, we'll focus on its major implications for how we perform computation (i.e., **the Four Postulates** that we will discuss later).

We may return to its interpretation when needed—for example, when discussing Hamiltonian complexity or applications of quantum computing to quantum simulation and quantum chemistry, where the meaning of the Hamiltonian $\hat{H}$ becomes more relevant.

Quantum Computing: Our Focus (1/4)

With the quantum mechanics background introduced earlier, our goal is to understand how the Schrödinger equation enables us to:

- encode information, and
- perform computation.

Quantum Computing: Our Focus (2/4)

Once we have the basics, we will explore:

- Differences between classical and quantum computing:
  - Are there problems quantum computers can solve that classical computers cannot?
  - What features are uniquely quantum?
- The limits of quantum computing:
  - Can quantum computers solve all problems we care about?
  - Which problems remain hard even for quantum computers?
- Applications/opportunities at the intersection of quantum computing and other fields:
  - quantum machine learning, quantum networks, quantum cryptography, quantum chemistry, and more.

Quantum Computing: Our Focus (3/4)

No programming; all theory: This course is similar in style to CSCI3160 Design and Analysis of Algorithms, focusing on the algorithmic ideas, pseudocode, and theoretical analysis. It does not involve programming, real-world implementation, or software engineering considerations.

More than "algorithms": We also explore the fundamental principles of quantum information — including concepts like superposition, entanglement, and measurement. The course may also cover several important topics that are not algorithmic in nature but are central to the field of quantum computing, such as quantum error correction, fault-tolerant computation, proofs of quantumness, and non-local games.

Quantum Computing: Our Focus (4/4)

This course is NOT intended for students who:

- Want to learn how to build a quantum computer
- Want to practice quantum programming in a specific (quantum programming) language
- Expect to learn quantum physics

This arrangement isn't unique to CSCI3350. It's a common practice for quantum computing courses offered by most CS departments. You can explore similar courses at:

- COMS4281 at Columbia University
- CS498QC at UIUC
- 15-859BB at CMU
- CS358H at UT Austin